Multispectral Classification of Landsat-Images Using Neural Networks

H. Bischof, W. Schneider, and A. J. Pinz, *Member, IEEE*

Abstract— **Recent progress in neural network research has demonstrated the usefulness of neural networks in a variety of areas. In this work we report the application of three-layer back-propagation networks for classification of Landsat TM data on a pixel-by-pixel basis. The results are compared to Gaussian maximum likelihood classification. First, we show that the neural network is able to perform better than the maximum likelihood classifier. Secondly, in an extension of the basic network architecture we show that textural information can be integrated into the neural network classifier without the explicit definition of a texture measure. Finally, we examine the use of neural networks for postclassificational smoothing.**

I. Introduction

The renewed interest in artificial neural networks (henceforth called neural networks) is mainly due to the discovery of new powerful learning algorithms and advances in analog VLSI technology. In this paper, the application of neural networks for multispectral image analysis is discussed. Traditionally, this is performed by a Bayesian classifier [9] (usually a Gaussian classifier when the underlying distribution is assumed to be Gaussian), which assigns the most likely class to the observed data. It is well known that the Bayesian classifier is theoretically optimal if the assumptions about the probability density functions (PDF's) are correct. The need to have a specific probabilistic model is a major limitation of the Bayesian approach and poor performance may be obtained if the true PDF's are different from those assumed by the model.

Neural networks can be seen as intermediate between statistical and structural methods though they resemble the former more than the latter. The ability of learning in neural networks provides an interesting alternative to the Bayesian classifier. It is especially interesting that no assumptions about the probabilistic model need to be made, so that neural networks can be called universal [34]. Recent results have shown that several classifiers are special cases of neural networks. Yau and Manary [35] have shown the equivalence between Gaussian classifiers and sigma–pi neural networks [10]. They showed that any Gaussian classifier can be implemented in a neural network. Yair and Gersho pointed out in a recent paper [34] that maximum *a posteriori* classifiers (classifiers which choose the class with the highest *a posteriori* probability) are a special case of the Boltzmann perceptron network. Two recent papers [28], [32] have shown that minimizing the mean (summed) squared error of the outputs of multilayer neural networks approximates the *a posteriori* probabilities of the various classes. Neural networks may therefore be considered as a nonparametric method for estimating *a posteriori* probability distributions. For these reasons we choose multispectral image classification as a test case to compare Gaussian classifiers with simple neural network architectures. We show that neural networks may yield better classification results than conventional multispectral classification methods (e.g., maximum likelihood classification [9]). As an example, images from Landsat TM (thematic mapper) are used, but the approach is general enough to be applied to other multispectral data.

Recently, work in applying neural networks to remotely sensed data has been reported. Decatur [8] has used three-layer back-propagation networks to classify synthetic aperture radar (SAR) data. He compares his results to maximum likelihood classifiers. Lee *et al.* [17] used four-layer back-propagation networks for cloud classification of Landsat MSS data. They also compared the neural network classifier with various statistical classifiers. McCellan *et al.* [20] in a preliminary study used three-layer back-propagation networks for mapping the seven-dimensional Landsat TM color space to the three-dimensional RGB color space. Benediktsson *et al.* [3] made an empirical comparison of neural networks and statistical techniques in the classification of multisource remote sensing (Landsat MSS) and geographic (elevation, slope, aspect) data. All these investigations point out aspects of the usefulness of neural networks in the classification of remotely sensed data.

In this paper, we first discuss several properties of neural networks and methods to apply neural networks to multispectral classification. In the following sections the networks used for Landsat TM classification are described. Classification results are given and compared to those of maximum likelihood classification of the same scene. Next, we propose an improved network architecture which is able to extract texture information for classification. Further, a postprocessing network is described. Finally, possible future improvements of our method are discussed.

II. Neural Networks

Neural networks or connectionist models or Parallel Distributed Processing (PDP) models are information processing systems which consist of a large number of very simple yet highly interconnected processing elements called units. Neural networks can be called biologically inspired, following the idea of using general organizational principles found in brains

Manuscript received January 2, 1991; revised November 4, 1991.

H. Bischof and A. Pinz are with the Institute for Automation, Department for Pattern Recognition and Image Processing, Technical University of Vienna, Treitlstr. 3/1832 A-1040, Vienna, Austria.

W. Schneider is with the Institute of Surveying and Remote Sensing, University of Bodenkultur, Peter Jordanstr. 82 A-1190, Vienna, Austria.

IEEE Log Number 9107671.

0196–2892/92$03.00

to build more powerful computing machines. The principles found in the brain and used in neural networks are parallel and distributed processing which means that information is not processed serially and is not stored at one fixed location. These features give neural networks a fault tolerant behavior which is generally referred to as "graceful degradation" [21]. This means that such systems can operate successfully in noisy environments or when some of their components are damaged [19].

A unit or processing element in a neural network can be considered as a simple processor with many different input connections from other units in the network, and one output which is sent to many other units. Every connection in the network has a numerical value attached to it which is called weight (w_{ij} denotes the weight of the connection from unit i to unit j). A unit i computes a net input (net_i) from the outputs o_j of other units and from the weights w_{ij} of the connections (usually by a weighted sum). In most models a numerical value called bias (b_i) is added to the net input. A function f is applied to this value (called activation function) yielding the output o_i of the unit. Formally, this can be stated as

$$\begin{aligned} \mathrm{net}_i &:= \sum\nolimits_j w_{ij} * o_j + b_i \\ o_i &:= f(\mathrm{net}_i). \end{aligned} \tag{1}$$

The idea of neural networks is not a new one. In the late 1950's and early 1960's there was great interest in neural networks. But the early models (the most famous of which was the Perceptron [27]) had some severe limitations as pointed out by Minsky and Papert [22]. Since these models consist of only two layers of processing units, they can only realize linear decision functions. To overcome these limitations of the early models and to be able to represent nonlinear decision boundaries one has to use either high order networks [10] which also have multiplicative connections, or networks with more than two layers of processing elements (multilayer perceptrons). Lippmann [19] showed that a four-layer network with threshold functions can represent any decision function. In [15], it was even shown that a three-layer network with sigmoid functions in the hidden units can approximate any Borel measurable function to any desired degree of accuracy (if enough hidden units are available).

Using high order networks, one can generalize the learning rules for two layer networks. There is, however, the danger of running into a "combinatorial explosion" even for small problems. For multilayer perceptrons one needs new learning algorithms which are able to train the hidden units (i.e., units which are neither input units nor output units).

One of the most popular learning schemes for multilayer perceptrons is the back-propagation learning algorithm. It was discovered by Werbos [33] and rediscovered independently by Parker [23], Le Cun [16], and Rumelhart *et al.* [29]. Back-propagation in its original version is designed to reduce an error between the actual and the desired output of a network in a gradient descent manner. The usual error measure, the summed squared error (*SSE*), is defined as

$$SSE = \frac{1}{2} \sum_p \sum_k (o_{pk} - t_{pk})^2 \tag{2}$$

where p ranges over all vectors in the training set and k denotes the output unit. o_{pk} indicates the output of unit k when the input vector p is applied to the network, and t_{pk} is the corresponding target output. The change of the weights is done in a gradient descent manner by

$$\Delta w_{ij} = -K \frac{\partial SSE}{\partial w_{ij}} \tag{3}$$

where Δw_{ij} is the weight change and K is the learning factor. A detailed derivation of the equations for the back-propagation algorithm can be found in [29].

Examples where back-propagation has been used successfully come from such diverse fields as text to phoneme conversion [30], protein structure prediction [26], backgammon [31], shape from shading [18], sonar target identification [11], tree species recognition [5], [24] and many others besides.

III. Data

The data we used for training and testing of the classification accuracy of the neural network were selected from a section (512 × 512 pixels) of a Landsat TM scene of the surroundings of Vienna. In Fig. 1, channel 4 of this scene is shown.

The aim of the classification with the neural network was to distinguish between the four categories: built-up land, agricultural land, forest, and water. The resulting thematic map was to be compared with the Gaussian classification. The Gaussian classifier assumes a normal distribution of the data. Preliminary analyses indicated that this requirement was not fulfilled in the case of the four categories. Therefore, these categories were split into 12 subcategories with spectral data of approximately (not exactly) normal distributions: Built-up land was subdivided into continuous urban fabric, discontinuous urban fabric (with rather large deviations from a normal distribution), industrial area, roads, and graveyards. Agricultural land was subdivided into soil without vegetation, soil with vegetation, and meadow. Forest was subdivided into forest and swamp forest and water was subdivided into water courses and water bodies.

Two thematic maps of this scene were prepared by visual classification of the Landsat image, using auxiliary data from maps, aerial photographs, and field work. These two thematic maps showing 4 and 12 landcover classes were considered to represent the "true" classification of the scene and were used for obtaining both training information and test data for assessment of the accuracy of the automatic classification results.

From this scene, a small set of samples was selected. About half of the pixels of this set were chosen for the training of the neural network and the Gaussian classifier ("training set", Table I), the rest was used for classification and testing of the result ("classification set"). This small set of pixels for testing the classification accuracy of the neural network was extremely

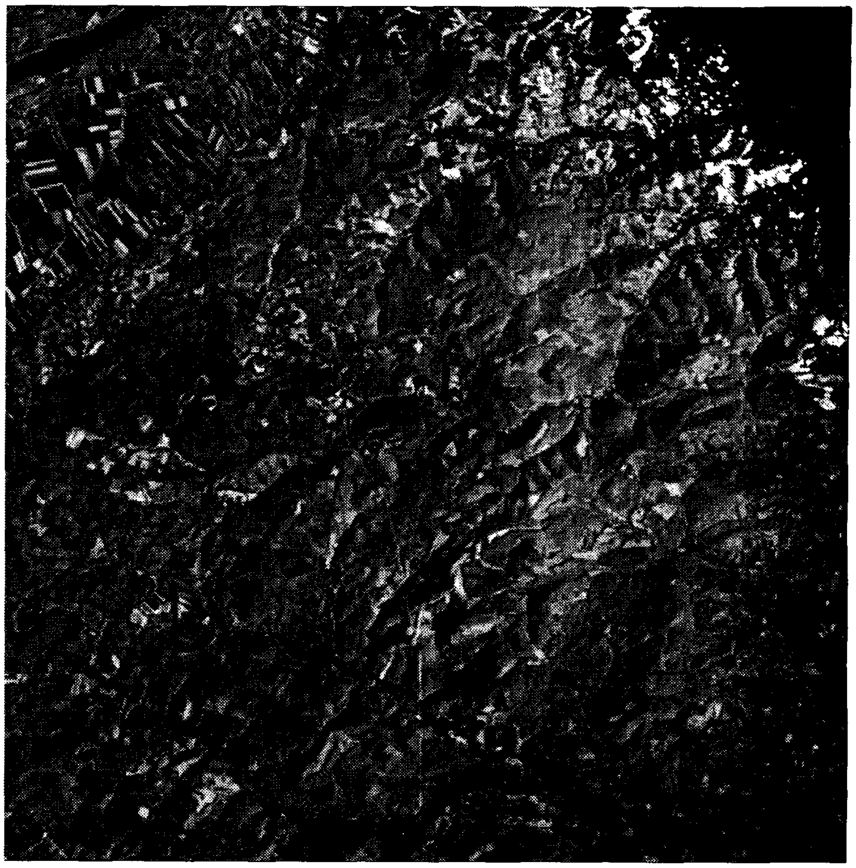


Fig. 1. Channel 4 of Landsat TM image.

TABLE I
CATEGORIES USED FOR CLASSIFICATION AND NUMBER OF TRAINING PIXELS

built-up land	continuous urban fabric	248	
	discontinuous urban fabric	734	
	industrial area	36	
	roads	52	
	graveyards	210	1280
forest	forest	3792	
	swamp forest	870	4662
water	water courses	790	
	water bodies	132	922
agricultural land	soil with vegetation	409	
	soil without vegetation	1271	
	meadow	435	2115

useful because many different parameter combinations could be tested very fast without classifying a whole image.

For maximum likelihood classification, the training data define the parameters of the (normal) probability distributions of the categories. It follows that representative samples of the categories should be selected for training. In feature space, the training samples should mainly lie in the interior of the classification region of the corresponding category. On the other hand, for neural network classifiers the training data should be selected to define the decision borders between the various categories accurately [31], so that rather atypical examples might be better for neural network training. Since our training data were selected according to the requirements of maximum likelihood classification, they are not optimal for training a neural network.

IV. Data Representation

Classification of multispectral data is a decision problem of assigning one of $M > 1$ possible classes to a vector (pixel) $p \in V^n$ (where $V \subseteq \mathbb{N}$ is a discrete set of greyscale values and n is the number of spectral channels, e.g., for Landsat TM:$V = \{0 \ldots 255\}$ and $n = 7$).

Using neural networks for classification, the spectral values have to be mapped onto a set of unit activations. This is referred to as data representation or coding. In a similar way, an output representation of the various classes has to be defined.

Input representations for the various bands can be considered independently. Therefore, a method for representing numerical values (greyscale values in a certain spectral band) in a neural network has to be found. There are numerous ways of achieving this [12]. One could use one unit per pixel (greyscale value) of each spectral channel and activate the unit proportionally to the value to be coded. Another form of coding, which is referred to as local coding, uses one unit for every value to be coded. The activation of unit i is set to 1 only when value i is to be represented, and all other units are set to 0. These two different forms of coding have several advantages and disadvantages. When using one unit per spectral channel, only a few units are needed, but the network is not able to separate similar pixel values. In the case of the other extreme (local representation), many units are used to represent one value, and the network can easily separate similar values. A problem associated with this form of coding is its inability to generalize: Even if the number to be coded changes by only one, a completely different activation pattern is presented to the network, which has little in common with the previous pattern (successful application of Gray coding is also reported [3], but in [12] it was shown that continuous codings are preferable to discrete codings).

Another kind of coding somewhat intermediate between these two forms is called coarse coding [13]. This form provides enough differences to separate similar values if necessary. At the same time, similar values also have a similar code, favoring the generalization properties of neural networks. Coarse coding can be considered a kind of interpolation. One uses n units to represent the numbers in the interval $[A, B]$. These n units are distributed (usually, but not necessarily) uniformly in this interval. This means that every unit i is assigned a fixed location z_i. When coding the numerical value m, the output of unit i at location z_i is given by a Gaussian response function (other forms are also possible):

$$o_i(m) = e^{\frac{-(m-z_i)^2}{\sigma^2}} \tag{4}$$

where σ is a parameter characterizing the width of the curve. Coarse coding is illustrated schematically in Fig. 2.

Usually the output for a classification task is coded locally (though other forms are possible). Every class is represented by one output unit which should be fully activated when the input corresponds to that class, while all other units should have zero activation. An input pattern is assigned class i if output unit i has the highest activation of all output units. A pattern is classified as unknown if the activations of all output units are below a prespecified threshold or the activations of two or more output units are above threshold and similar.

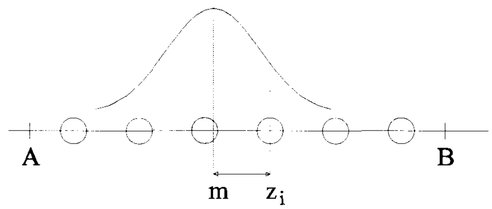


Fig. 2. Coarse coding.

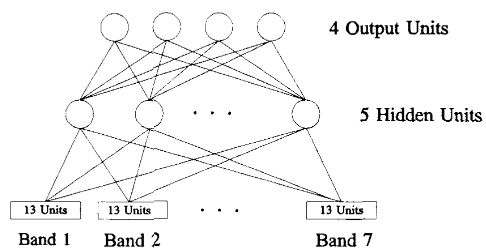


Fig. 3. Network architecture.

In this work, two separate channels are generated when classifying an image. The first channel represents the classification result and a second one is a measure of confidence for the classification. The measure used is the difference of the activation of the most activated unit and the unit of the next lower activation. The number thus obtained is scaled between 0 and 255 giving 255 for an absolutely certain and 0 for a completely uncertain classification. This channel can be used to generate a rejection class. If the confidence of classification of a pixel is below a certain threshold, the pixel is classified as unknown.

V. Network Architecture

All network simulations described in this paper were implemented using the neural network tool NNT [4]. The basic network architecture used for classification is shown in Fig. 3. The activation function of the hidden and output units is a sigmoid function given by:

$$o_i = f(\text{net}_i) = \frac{1}{1 + e^{-\text{net}_i}} \tag{5}$$

The network is trained with the back-propagation algorithm [29], using the SSE (see (2)) as error measure which is minimized.

For each of the seven Landsat TM spectral channels (bands) we use 13 units with a coarse coding scheme to represent the interval of [0, 255]. As preliminary studies have shown, the network is not very sensitive to small variations of the number of units per channel. The parameter σ in (3) is set to 23. The network is not very sensitive to variations of this parameter either.

The number of output units is four, corresponding to four classes: built-up land, forest, water, and agricultural area.

TABLE II
PERCENTAGES OF CORRECTLY CLASSIFIED PIXELS VERSUS NUMBER OF HIDDEN UNITS. THE VALUES ARE OBTAINED AFTER 50 EPOCHS OF LEARNING

hidden units	training set	classification set
2	80.0%	86.6%
3	97.4%	97.5%
5	**98.2%**	**98.1%**
8	97.8%	97.8%
10	97.1%	97.1%
15	97.4%	97.6%

The class membership of the pixel presented at the input is determined by choosing the output unit with the highest activation (we did not use any thresholds).

The number of hidden units can be a crucial question in network design, as was pointed out in [1]. We use five hidden units, but the classification accuracy is hardly affected by a variation of this number of hidden units. We tested networks with from 2 up to 15 hidden units. The performance is about the same for all these networks with a number of hidden units between 3 and 15 (see Table II). This may seem to be in contradiction to experiments from previous projects [5] which indicated a decrease of network performance for large numbers of hidden units. An explanation for this phenomenon may be found in the large number of training examples used here. While the network is able to learn the problem with a rather small number of hidden units there are sufficient training examples for a correct determination of additional weights.

VI. RESULTS

The various parameters of the network (number of hidden units, width of the Gaussian curve for input coarse coding, number of input units per spectral band, learning rate) could be varied within certain limits without influencing the classification performance too much (see Table II for results on varying the number of hidden units). Peak performance was achieved with 13 units per channel, using all seven channels, 5 hidden units, $\sigma = 23$ (see (4)) and learning factor $K = 0.2$ (see (3)). In this case, 98.2% of the training set and 98.1% of the classification set were correctly classified after 50 epochs (complete presentations of the the training set) of learning. Examining the learning curve (Fig. 4) one can see rapid convergence in the first few steps and only slight improvements and oscillatory behavior for the following steps. Most of the errors are made classifying the agricultural class as built-up land and vice versa.

Applying the network to the whole image leads to a classification accuracy of 85.9%, which is slightly more than the Gaussian classifier achieved with 84.7% correctly classified pixels (training and classification with 12 classes, merging into four classes after classification). In Table III the results of the two classification procedures are shown. In Fig. 5 the result of the neural network classification is compared to the "true" (visual) classification. Comparing the results one can see that both methods have similar problems (misclassifications of built-up land and agricultural area). The neural network makes less errors for all categories except for the agricultural class where maximum likelihood outperforms the network.

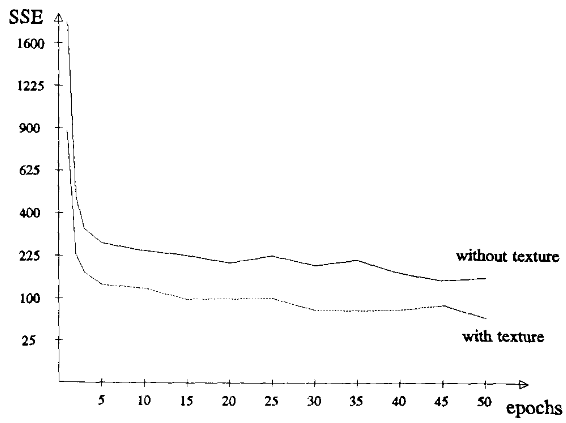


Fig. 4. Learning curves for networks.

It is well known that from a statistical point of view maximum likelihood classification is optimal. The question then arises why the neural classifiers can perform even better. We believe that the small but (in spite of the subdivision of the categories) existing deviations of the spectral data from normal distributions are responsible for this phenomenon. This assumption is supported by the fact that the neural network outperforms the Gaussian classifier especially for the built-up land category, where the deviation from a normal distribution is noticable most.

VII. WEIGHT VISUALIZATION CURVES (WV-CURVES)

To examine the mode of operation of neural networks, a method for visualizing the weights by so-called WV-diagrams has been developed [7], [25] which has proved extremely useful in the past. The WV-diagrams were modified to the WV-curves, in order to better represent the spectral character of the data. In Fig. 6 the WV-curves for a specific network are shown. Each of the five hidden units receives input from seven spectral channels. Therefore, seven WV-curves are associated with one hidden unit. A single WV-curve visualizes the weights of the connections from 13 input units—the pixel grey value is coarse coded using 13 units to represent the interval [0, 255]. Positive values indicate positive weights from that particular input unit. In the top row of Fig. 6 the weights from the hidden units to the output units are visualized in a Hinton diagram-like manner [14]. There is one square for each output unit, with black squares indicating negative weights, and the size of the squares corresponding to the magnitude of the weight.

Using such diagrams the network behavior can be interpreted more easily. The internal representation developed by the network and represented by these diagrams can, at least in part, be explained in terms of the reflection curves and temperatures of the various ground cover types. For example, the curve for hidden unit 5 channel 4 (near infrared), which is positively connected to the output units 1 (built-up land) and 3 (water) and negatively connected to the output units 2 (forest) and 4 (agricultural land) indicates a positive contribution for low channel 4 values (no vegetation) and a

TABLE III
RESULTS OF CLASSIFICATION PROCEDURES

Neural Network

		classified as				correctly classified
	true category	1	2	3	4	
1	built-up land	42821	980	240	4898	87.5%
2	forest	8783	145541	1091	6557	89.9%
3	water	46	29	2492	37	95.7%
4	agricultural area	10855	3283	135	34356	70.6%
	overall accuracy					85.9%

Maximum Likelihood

		classified as				correctly classified
	true category	1	2	3	4	
1	built-up land	38286	986	18	9649	78.2%
2	forest	6824	145403	181	9564	89.7%
3	water	195	5	2208	196	84.7%
4	agricultural area	8689	3841	43	36059	74.1%
	overall accuracy					84.7%

Neural Network with Texture Input

		classified as				correctly clasified
	true category	1	2	3	4	
1	built-up land	40720	1122	93	5564	85.7%
2	forest	6640	145426	597	6229	91.5%
3	water	75	41	2135	45	93.0%
4	agricultural area	6801	3231	75	37238	78.7%
	overall accuracy					88.1%

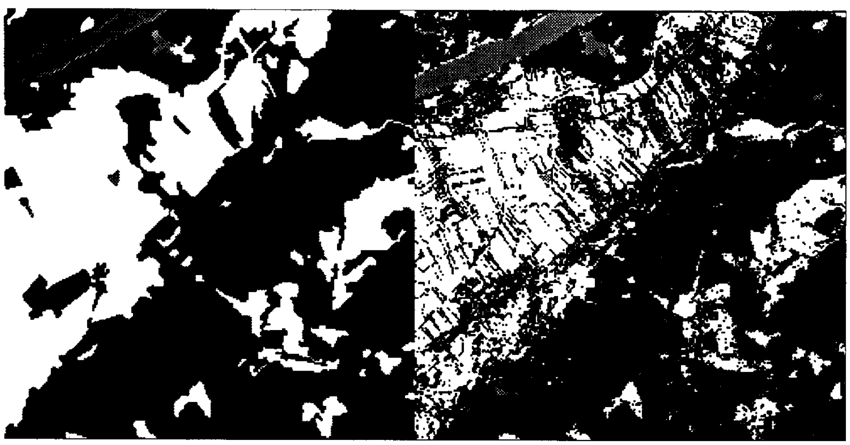


Fig. 5. "True" classification (left) vs. neural network classification (right) for upper left part of Landsat TM scene, black: built-up land (1), light gray: forest (2), dark gray: water (3), white: agricultural land (4).

negative contribution for high channel 4 values (vegetation). For areas without vegetation this unit will, therefore, favor a decision toward built-up land and/or water, while a decision toward forest and/or agricultural land is favored for areas with vegetation. Similarly, the curve for channel 6 (thermal infrared) and hidden unit 2 indicates that low temperatures favor a decision toward category 3 (water), while medium temperatures favor a decision toward category 4 (agricultural land).

WV-curves are also useful in feature selection. The weights of the blue channel (no. 1) and the green channel (no. 2) can be seen as very similar. This means that these two channels are highly correlated so that one of these two channels can be eliminated without degrading the classification accuracy. There are some spectral channels in a few hidden units where the sign of the weights changes for nearby input units (e.g., channel 2 of hidden unit 2). This fact indicates that these channels are hardly used by that hidden unit and may therefore be eliminated.

VIII. INCLUDING TEXTURE

One can see from the error matrix of the network classification that most of the errors occur in the agricultural category. This can be explained by the fact that in this category there are pixels with spectral signatures similar to those of built-up areas. These pixels, therefore, cannot be

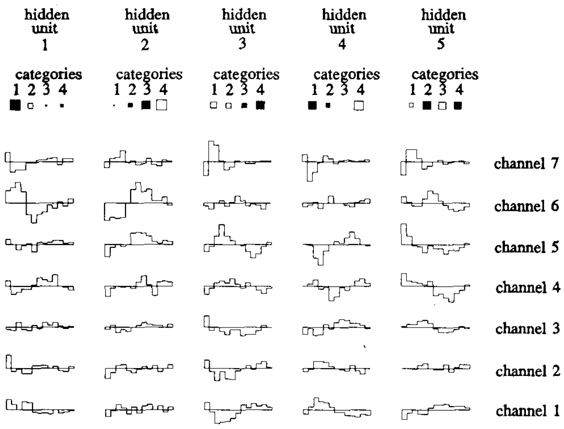


Fig. 6. WV-curves for network.

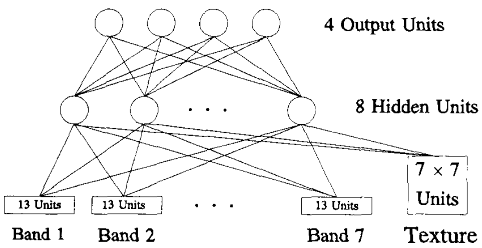


Fig. 7. Extended network architecture.

separated solely on the basis of multispectral data. Visual inspection reveals, however, that the two categories are clearly distinguishable on the basis of the spatial distribution of pixels of different spectral signatures. The use of textural features in the automatic classification process should therefore improve the classification result. Texture information can be included in a neural network classifier by adding input units for a window of $n \times n$ pixels around the pixel to be classified. We use a 5×5 and a 7×7 window in channel 5, coding the activation of the 25 or 49 units proportional to their greyvalue. In Fig. 7 the extended network architecture is shown. The number of hidden units was raised from 5 to 8 in order to give the network the chance to cope with this additional information. In Table III the results of the classification with texture (using a 7×7 window) are given. In Fig. 4 the learning curve for the network with texture is shown. One can see that the network is able to learn faster and more effectively.

Comparing this result with the results of the network which does not use texture (Table III), one can see that the categories forest and agriculture are predicted more accurately. The classification accuracy of the categories built-up land and water dropped, but the overall accuracy is raised mainly due to improvements of the agricultural class where most of the problems occurred in classification without texture information.

This example shows that neural networks are able to integrate other sources of knowledge in the classification procedure (e.g., texture) and use them for classification. It seems particularly important that no explicit measure of texture has to be defined, so that we believe the method to be applicable to a wider variety of problems.

IX. Postclassification Smoothing

After classification on a pixel-by-pixel basis, the result usually has a "salt-and-pepper" appearance (i.e., isolated, in many cases misclassified pixels of one category are dispersed within the area of another category). "Smoothing" of the classification result is desirable to remove isolated misclassified pixels and to obtain a certain degree of generalization, as is common practice in thematic mapping. For the smoothing of a classification result a majority filter can be used, assigning every pixel to the majority category within an $n \times n$window surrounding the pixel. Other techniques based on the same or similar principles may be employed (e.g., weighting, preservation of landcover regions). We use a special neural network for postclassificational smoothing. The input to this network is a $n \times n$ window scanning the classified image. For each pixel of the window, the classification result with local coding of the category and the confidence information generated during classification are input. The idea is that the degree of smoothing should depend on the confidence of classification of the individual pixels. We use two-layer networks for this smoothing procedure because nonlinear decision surfaces are not needed. For training the network the same training data as for the classification network are used.

Using a 5×5 input window for this network, smoothing the image which was classified without texture input raised the classification accuracy to 90.8%. Using a weighted majority filter the classification accuracy was 89.1%. The application of a smoothing network for the classification of an image classified with texture input for the neural network raised the classification accuracy to 91.0%. The gain in classification accuracy is less for a network which uses texture input, because by using texture for classification the salt-and-pepper effect is already reduced dramatically (see Fig. 8 for the comparison of smoothed classification without texture versus unsmoothed classification with texture input). Therefore, using texture as an additional input for the network also has the result of reducing the salt-and-pepper effect of the classification.

Using neural networks for postclassificational smoothing has the advantage that small elements, for which the classification is certain, do not disappear. On the other hand, large objects which have a low confidence measure for the classification may well be removed. In this respect, the above method outperforms conventional filters.

A similar method of postclassificational processing may also be used for a combination of two or more classification results (e.g., combination of maximum likelihood and neural network classification), where the network selects the "better" classification result out of a set of possible results.

X. Conclusion

We have shown that using neural networks for multispectral image classification gives results that are comparable to maximum likelihood classifiers but are not sensitive to the

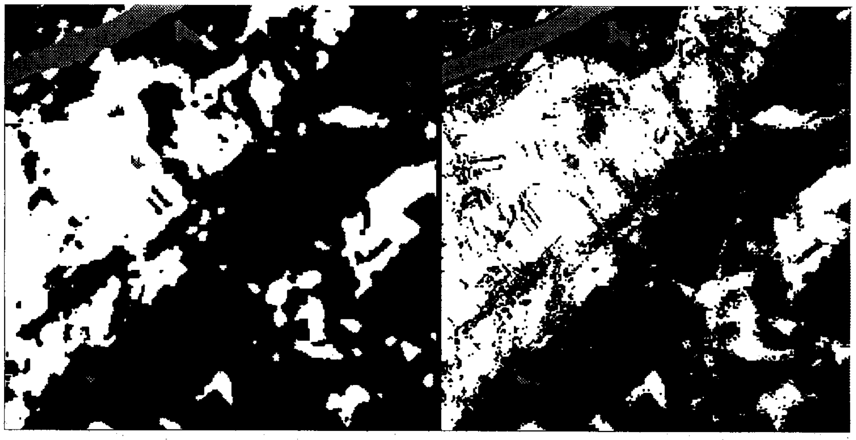


Fig. 8. Smoothed classification (left) versus unsmoothed classification with texture (right) for upper left part of Landsat TM scene, black: built-up land (1), light gray: forest (2), dark gray: water (3), white: agricultural land (4).

form of the underlying probability density function. It has been shown that texture information can be easily integrated and used by neural network classifiers. A method based on neural networks for postclassificational smoothing was also presented and shown to be superior to conventional majority based filters. The weights developed by the neural network and visualized in "WV-curves" have been shown to have a physical interpretation in this particular example.

A point of further investigation will be the incorporation of this method into conventional knowledge based systems for image interpretation, especially the Vision Station VS [2]. Some preliminary results have already been obtained [6]. VS will use the neural network described here to generate hypotheses about the image under consideration, and will apply further operators to the image (especially when the measure of confidence is low) to get more information. VS will also use special neural networks which are trained to recognize specific objects. Using such methods one will get a hybrid system where conventional and neural systems are combined into one system and work together cooperatively.

REFERENCES

[1] V. V. Anshelerich, A. V. Amirikian, A. V. Lukashin, and M. D. Frank-Komenetskii, "On the ability of neural networks to perform generalization by induction," *Biol. Cybernetics*, vol. 61, pp. 125–128, 1989.

[2] R. Bartl, A. Pinz, "Allgemeines Konzept einer Ablaufsteuerung für ein Bildverstehendes System," in *Image Acquisition and Real-Time Visualization* (OCG Schriftenreihe 56), Bernroider and Pinz, Eds. Oldenbourg, 1990, pp. 51–75.

[3] J. A. Benediktsson, P. H. Swain, and O. K. Ersoy, "Neural network approaches versus statistical methods in classification of multisource remote sensing data," *IEEE Trans. Geosci. Remote Sensing*, vol. 28, no. 4, pp. 540–552, July 1990.

[4] H. Bischof and A. Pinz, "A neural network tool as an additional feature for a computer vision system," in *Image Acquisition and Real-Time Visualization* (OCG Schriftenreihe 56), Bernroider and Pinz, Eds. Oldenbourg, 1990, pp. 37–50.

[5] H. Bischof and A. Pinz, "Verwendung von neuralen Netzwerken zur Klassifikation natürlicher Objekte am Beispiel der Baumerkennung aus Farb-Infrarot-Luftbildern," in *Konnektionismus in Artificial Intelligence und Kognitionsforschung*(Informatik Fachberichte 252), Dorffner, Ed. Berlin: Springer Verlag, 1990, pp. 112–120.

[6] H. Bischof, R. Bartl, A. Pinz, and W. Schneider, "AI-methods in remote sensing: Neural networks and knowledge based systems," to appear in the *Proc. of the 11th EARSEL Symposion Graz*, pp. 14–22, July 1991.

[7] H. Bischof and A. Pinz, "Visualization methods for neural networks," in *Konnektionismus: Beiträge aus Theorie und Praxis*,H. G. Ziegeler, Ed., Schriftenreihe der ÖGAI, Bd. 3, 1–14, 1991.

[8] S. E. Decatur, "Applications of neural networks to terrain classification," in *Proc. IJCNN'89 (Washington DC)*, vol. I, pp. 283–288, 1989.

[9] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* New York: Wiley, 1973.

[10] L. C. Giles and T. Maxwell, "Learning, invariance, and generalization in high-order neural networks," *Appl. Opt.*, vol. 26, pp. 4972–4978, 1987.

[11] P. Gorman and T. Sejnowski, "Learned classifications of sonar targets using a massively parallel network," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 36, pp. 1135–1140, 1988.

[12] P. J. Hancock, "Data representation in neural nets: an empirical study," in *Proc. of the 1988 Connectionist Models Summer School, San Mateo*, Touretzky, Hinton, and Sejnowski, Eds. Morgan Kaufmann, 1988, pp. 11–20.

[13] G. E. Hinton, J. L. McCelland, and D. E. Rumelhart, "Distributed representations," in Rumelhart and McCelland, Eds., *Parallel Distributed Processing, Vol I.* Cambridge, MA: MIT Press, 1986.

[14] G. Hinton, "Connectionist learning procedures," *Artificial Intelligence*, vol. 40, pp. 185–234, 1989.

[15] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[16] Y. Le Cun, "Learning processes in an assymetric threshold network," in *Disordered Systems and Biological Organization*, Bienenstock, Fogelman Souli, and Weisbruch, Eds. Berlin: Springer-Verlag, 1986.

[17] J. Lee, R. C. Weger, S. K. Sengupta, and R. M. Welch, "A neural network approach to cloud classification," *IEEE Trans. Geosci. Remote Sensing*, vol. 28, no. 5, pp. 846–855, 1990.

[18] S. R. Lehky and T. J. Sejnowski, "Network model of shape-from-shading: Neural function arises from both receptive and projective fields," *Nature*, vol. 333, pp. 452–454, 1988.

[19] R. L. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, vol. 4, pp. 4–22, 1987.

[20] G. E. McCellan, R. N. DeWitt, L. N. Matheson, and G. O. Moe, "Multispectral image-processing with a three-layer backpropagation network," *in Proc. IJCNN'89 (Washington DC)*, Vol. I, pp. 151–153, 1989.

[21] J. L. McCelland, D. E. Rumelhart, and G. E. Hinton, "The appeal of parallel distributed processing," in *Parallel Distributed Processing, Vol. I*, Rumelhart and McCelland, Eds. Cambridge, MA: MIT Press, 1986.

[22] M. Minsky and S. Papert, *Perceptrons, Expanded Edition 1988.* Cambridge, MA: MIT Press, 1969.

[23] D. B. Parker, "Learning-logic," Tech. Rep. TR-47, Sloan School of Management, MIT, Cambridge, MA 1985.

[24] A. Pinz and H. Bischof, "Constructing a neural network for the interpretation of the species of trees in aerial photographs," in *Proc. 10. ICPR Atlantic City*, IEEE Computer Society Press, pp. 755–757, 1990.

[25] A. Pinz and H. Bischof, "The IVF Weight-Visualization Diagrams (WV-Diagrams)," IVF Rep., Insitute of Surveying and Remote Sensing, Univ. of Bodenkultur, Vienna, July 1990.

[26] N. Qian and T. J. Sejnowski, "Predicting the secondary structure of globular proteins using neural network models," *J. Mol. Biol.*, vol. 202, pp. 865–884, 1988.

[27] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 386–408, 1958.

[28] D. W. Ruck, S. K. Rogers, M. Kabrisky, M. E. Oxely, and B. W. Suter, "The multilayer perceptron as an approximation to a Bayes optimal discriminant function," *IEEE Trans. Neural Networks*, vol. 1, pp. 296–298, 1990.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[30] T. J. Sejnowski and C. R. Rosenberg, "NETtalk: A parallel network that learns to read aloud," Tech. Rep. JHU/EECS-86/01, John Hopkins University, Baltimore, MD, 1986.

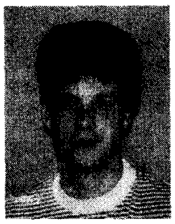
[31] G. Tesauro and T. J. Sejnowski, "A parallel network that learns to play backgammon," *Artificial Intelligence*, vol. 39, pp. 357–390, 1989.

[32] E. A. Wan, "Neural network classification: A Bayesian interpretation," *IEEE Trans. Neural Networks* vol. 1, pp. 303–305, 1990.

[33] P. J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," Ph. D. thesis, Harvard University, Cambridge, MA, 1974.

[34] E. Yair and A. Gersho, "Maximum *a posteriori* decision and evaluation of class probabilities by Boltzmann perceptron classifiers," *Proc. IEEE*, vol. 78, no. 10, 1990.

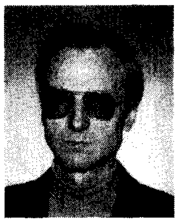
[35] H-C. Yau and M. T. Manary, "Iterative improvement of Gaussian classifier," *Neural Network*, vol. 3, pp. 437–443, 1990.



Horst Bischof was born in Saanen, Switzerland, in 1967. He received the M.Sc. degree in computer science in 1990 from the Technical University of Vienna.

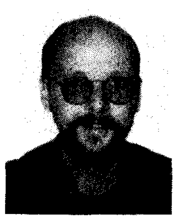
From 1989 to 1991 he was with the Institute of Surveying and Remote Sensing, University of Bodenkultur, Vienna. Since 1991 he has been with the Institute for Automation, Technical University of Vienna, where he is working toward the Ph.D. degree in computer science. His research interests include neural networks, image understanding, fractals, and cognitive science.

He is a member of the International Neural Network Society (INNS), International Association forPattern Recognition (IAPR), and Austrian Association for Artificial Intelligence (ÖGAI).



Werner Schneider received the Ph.D. degree in technical physics from the Technical University of Vienna in 1976.

He is an assistant professor at the University of Bodenkultur, Vienna, where he heads the Remote Sensing Department of the Institute of Surveying and Remote Sensing. His research interests cover various aspects of radiometric calibration, atmospheric correction, and automated thematic analysis of remote sensing image data.



Axel J. Pinz (S'86–M'87) was born in Vienna, Austria, in 1958. He received the M.Sc. degree in electrical engineering in 1983 and the Ph..D degree in computer science in 1988 from the Technical University of Vienna.

From 1983 to 1990 he was with the Institute of Surveying and Remote Sensing, University of Bodenkultur, Vienna. Since 1990 he has been with the Institute for Automation, Technical University of Vienna. His research interests include image understanding, multisource data integration, and neural networks for pattern recognition with applications in remote sensing and medical image processing.