



ELSEVIER

Pattern Recognition Letters 17 (1996) 1331–1341

---

---

Pattern Recognition  
Letters

---

---

# An experimental comparison of neural and statistical non-parametric algorithms for supervised classification of remote-sensing images

S.B. Serpico <sup>a,\*</sup>, L. Bruzzone <sup>a</sup>, F. Roli <sup>b</sup>

<sup>a</sup> *Department of Biophysical and Electronic Engineering, University of Genoa, Via all' Opera Pia, 11A, I-16145, Genova, Italy*

<sup>b</sup> *Department of Electrical and Electronic Engineering, University of Cagliari, Piazza d'Armi, I-09123, Cagliari, Italy*

Received 18 July 1996

---

## Abstract

An experimental analysis of the use of different neural models for the supervised classification of multisensor remote-sensing data is presented. Three types of neural classifiers are considered: the Multilayer Perceptron, a kind of Structured Neural Network, proposed by the authors, that allows the interpretation of the network operation, and a Probabilistic Neural Network. Furthermore, the  $k$ -nearest neighbour statistical classifier is also considered in order to evaluate the validity of the aforementioned neural networks, as compared with that of classical statistical methods. The results provided by the above classifiers are compared.

*Keywords:* Artificial neural networks; Supervised classifiers; Remote-sensing image classification

---

## 1. Introduction

Classical statistical methods, like the maximum likelihood and the  $k$ -nearest neighbour classifiers, have been traditionally used to perform classification of remote-sensing images (Swain and Davis, 1978). However, in recent years, the remote-sensing community has focused attention on the neural-network approach to data classification, mainly because this approach does not require any a priori knowledge of the statistical distribution of data and is characterized by intrinsic parallelism and fast classification time

(Bischof et al., 1992). On the other hand, some difficulties arise when using neural networks, related to the choice of the neural model and of the network architecture, to the dependence of classification results on training conditions, and to the difficulty with interpreting the network behaviour.

Several works dealt with neural networks for remote-sensing data classification. Lee et al. (1990), Bischof et al. (1992), Hwang et al. (1993), Salu and Tilton (1993), and Paola and Schowengerdt (1995a) applied neural networks to the classification of images acquired with a Multispectral Scanner and a Thematic Mapper of Landsat. Decatur (1989) and Azimi-Sadjadi et al. (1993) used the neural approach to classify synthetic aperture radar (SAR) data. Neu-

---

\* Corresponding author. E-mail: vulcano@dibe.unige.it.

ral classifiers were applied to multisource remote-sensing data by Benediktsson et al. (1990), Ersoy and Hong (1990), Serpico and Roli (1995). Recently, Benediktsson et al. (1995) applied neural networks to the classification of very-high-dimensional data provided by an Airborne Visible-Infrared Imaging Spectrometer (AVIRIS).

This paper reports the results of an experimental investigation into multisensor (optical and SAR) remote-sensing image classification by using two non-parametric approaches, i.e., the neural-network and the statistical approaches. The choice of non-parametric methods is due to the difficulty with representing the distributions of multisensor data by a parametric model. Three neural models were considered: the Multilayer Perceptron, a kind of Structured Neural Network, proposed by the authors, that allows the interpretation of the network operation, and a Probabilistic Neural Network. A brief description of such neural classifiers and the definition of the architectures used in our investigation are reported in Section 2. Experimental results are presented and compared with those obtained with one of the most widely used non-parametric statistical classifiers, i.e., the  $k$ -nearest neighbour classifier ( $k$ -nn) (Section 3). Results are discussed in Section 4, where conclusions are also drawn.

## 2. Neural models and architectures

### 2.1. Multilayer perceptron

Multilayer Perceptrons (MLPs) (Rumelhart et al., 1986; Hush and Horne, 1993) are the most widely used neural networks for the classification of remote-sensing images (Paola and Schowengerdt, 1995b). They are multilayer feedforward networks composed of multiple layers of neurons: an ‘‘input layer’’, one or more ‘‘hidden layers’’ and an ‘‘output layer’’.

In our experiments, we used as many neurons in the input layer as the number of attributes that characterized each pixel to be classified; the output layer consisted of a number of neurons equal to the number of data classes. Input neurons just propagate input attribute vectors to the next layer. As activation function for the neurons of the hidden layers and of

the output layer, we used the sigmoid function (Hertz et al., 1991). The classification of each pixel was carried out by assigning each pixel to the class corresponding to the output unit with the highest output (Winner-Takes-All (WTA) decision rule), with no threshold margin between the maximum output and the other outputs. Thanks to an appropriate data representation and to an appropriate learning of the network (Richard and Lippmann, 1991), this decision rule corresponds to the selection of the most probable class, as estimated by the network.

A problem with the use of MLPs is the selection of the optimal architecture of the network (Moody, 1991), i.e., the number of hidden layers, the number of neurons per layer and the connections among them. According to a well-known theoretical result (Hertz et al., 1991), one hidden layer is sufficient to approximate any continuous function. However, this result is difficult to use in applications, as it does not suggest how many neurons per layer should be included; moreover, this number may be large. A heuristic approach suggests selecting the number of hidden layers and the number of neurons per layer according to a tradeoff between complexity of representation and generalization ability (Musavi et al., 1994) of the net. The complexity of representation depends on both the number of hidden layers and the number of neurons per layer: each neuron provides a different nonlinear transformation, and each hidden layer increases the complexity of the nonlinear transformations of inputs. On the other hand, the capability of generalization of the network decreases with the increase in the number of parameters (i.e., the number of neurons and connections).

In our experiments, we used fully connected networks. To choose the number of parameters  $W$ , we used a simplified version of the rule suggested in (Baum and Haussler, 1989), i.e.,

$$W < \varepsilon m, \quad (1)$$

where  $m$  is the number of training pixels and  $\varepsilon$  is the desired error probability in the classification of unknown samples. Using the aforementioned simplified rule, we experimented with different architectures, with one or two hidden layers and various numbers of units per layer. Several learning techniques were proposed for MLPs (Vogl et al., 1988; Hinton, 1989; Tollenaere, 1990; Hertz et al., 1991).

In our experiments, we used the classic error back-propagation algorithm (Hertz et al., 1991).

## 2.2. Structured neural network

The Structured Neural Networks (SNNs) we utilized in our comparative investigation are MLPs with structured architectures tailored to multisensor classification problems (Serpico and Roli, 1995). Their main feature lies in the possibility of obtaining a simplified network representation that allows a quantitative and detailed interpretation of the network operation. Such a representation may be utilized by photointerpreters for the validation of the neural classifier.

SNNs have one-net–one-class architectures, that is, a separate network is devoted to each class. Each Class-Related Network (CRN) has a not fully connected architecture with an input layer, two hidden layers, and an output layer (Fig. 1). The input layer has as many neurons as the number of sensor channels; each neuron of the first hidden layer imposes a constraint on the intensity in a sensor channel. A neuron is included in the second hidden layer for each sensor to combine the results of the application of constraints related to the channels of that sensor. The output layer is composed of one neuron which combines the output of sensor-related neurons of the second hidden layer; such an output neuron computes an estimate of the probability that an input pixel belongs to the class associated with the CRN considered. A WTA decision block compares the outputs of the different CRNs and makes the final decision on classification. Each CRN is trained separately (in our experiments, we adopted the same learning procedure as for the MLPs, i.e., the back-propagation algorithm), then it is used to classify data.

For the purpose of interpreting the network behaviour, two transformations (which save the input–output response of the network) are then applied to each CRN in order to obtain a simplified representation of the network (Serpico and Roli, 1995). The first transformation makes the importance of the contribution of a neuron to the input to a neuron of the next layer correspond directly to the weight of the connection between the two neurons. The second transformation provides a piecewise-linear approxi-

mation of neuron activation functions and channel constraints in order to make the comprehension of the network behaviour easier. After these transformations, the simplified representation of a CRN (see, for example, Fig. 2) allows one to interpret the network as a hierarchical arrangement of committees that judge on the hypothesis that a pixel belongs to a given class. In particular, each couple of neurons of the first hidden layer is transformed into an equivalent neuron that provides its judgment on the basis of a constraint on the intensity values in a channel; this equivalent neuron is a member of a “Sensor-Related Committee” (SRC). Each neuron of the second hidden layer is a “Vote-Taking Unit” (VTU) of an SRC, that is, it combines the votes of the members of its SRC. In addition, the members of the second

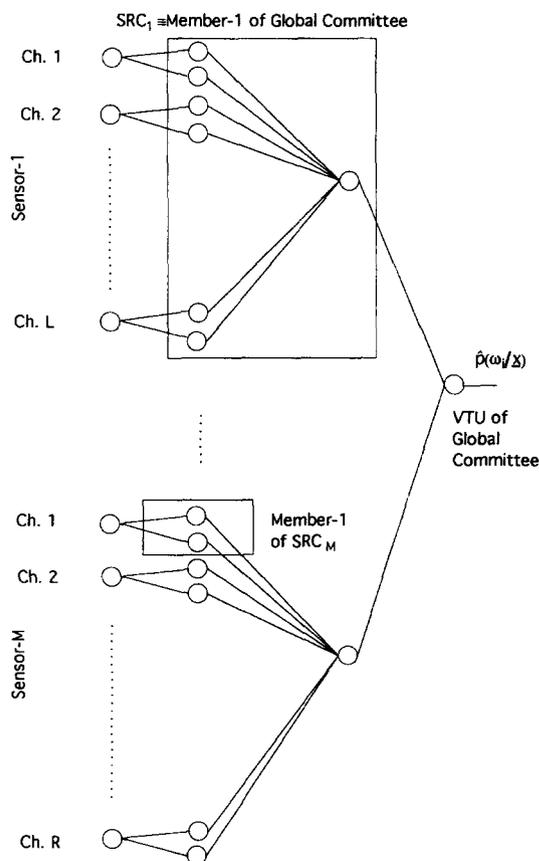


Fig. 1. Architecture of a Structured Neural Network (SRC stands for Sensor-Related Committee and VTU stands for Vote-Taking Unit).



hidden layer are members of the “global committee” whose votes are taken by the neuron of the output layer to compute the final decision about the probability that a pixel belongs to a given class. The importance of both channels and sensors for the output of each CRN depends on their “voting power” in the related committee, i.e., on the values (normalized to 1000) of the weights associated with the related connections. The decisions are made by each VTU on the basis of majority rules applied to the sums of the votes of all members of the related committee. These rules are defined by piecewise-linear functions in the simplified representation (Fig. 2). Such a representation allows a photointerpreter to understand if the network operation is in agreement with his a priori knowledge and with the visual analysis of the input image. If there is no agreement, the network may be trained again with different random starting weights. An example of the network interpretation is given in the next section.

### 2.3. Probabilistic neural network

Probabilistic Neural Networks (PNNs) are a model for supervised classification based on multivariate probability estimation (Specht, 1990). They are founded on an extension of the Parzen approach to univariate probability estimation (Fukunaga, 1990). Given a set of  $N$  samples  $X_i$  drawn from a statistical distribution  $p(X)$ , the Parzen approach provides an asymptotic, unbiased and consistent estimate  $\hat{p}(X)$  of the related probability density function by using an appropriate “kernel function”  $k(\cdot)$  which is applied to each sample considered, i.e.,

$$\hat{p}(X) = \frac{1}{N} \sum_{i=1}^N k(X - X_i). \quad (2)$$

PNNs are founded on an extension of such an approach to the multivariate case (Cacoullos, 1966), based on the use of the Gaussian kernel function.

The typical architecture of a PNN is shown in Fig. 3. The network is composed of an input layer, one hidden layer and an output layer. For our experiments, also in this architecture the number of neurons in the input layer is equal to the number of attributes. The hidden layer has as many neurons as the number of training patterns; as a kernel function,

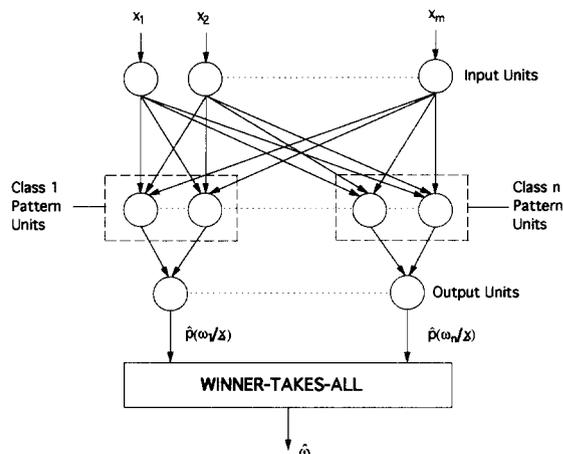


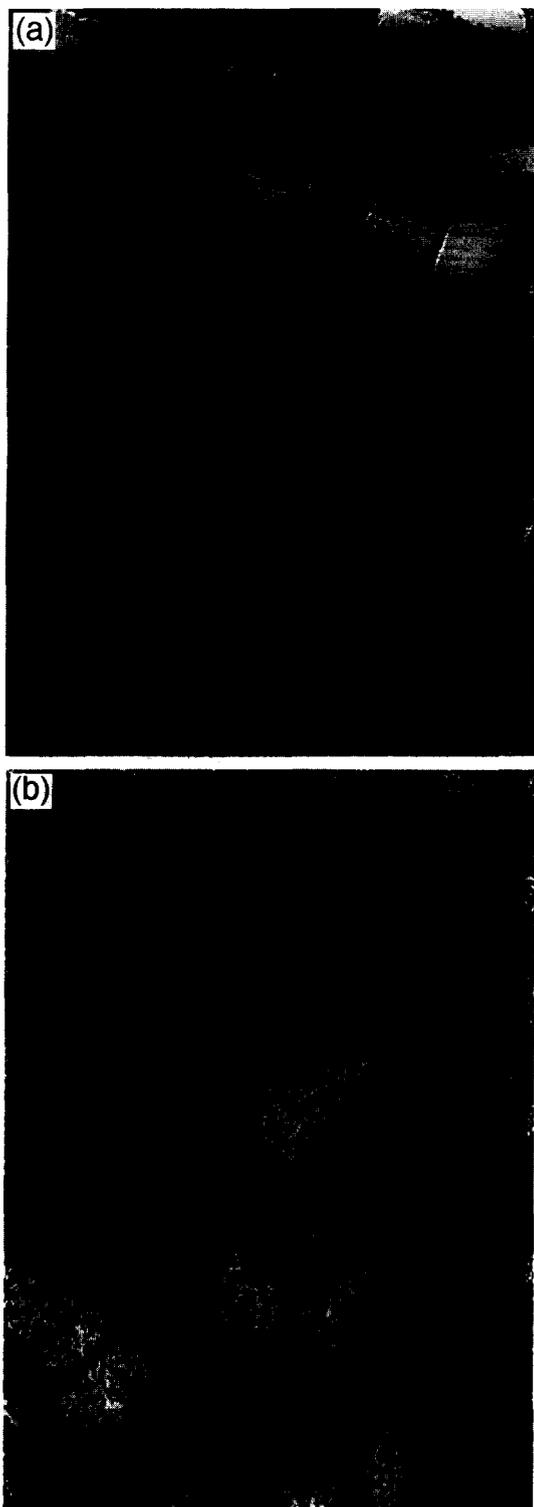
Fig. 3. Typical architecture of a Probabilistic Neural Network.

each neuron has an activation function of the Gaussian type, and is centred on the attribute vector of the corresponding training pixel. The output layer has as many neurons as the number of data classes considered; the activation function of each output neuron computes the sum of the inputs to the neuron. The neurons of the hidden layer propagate their outputs only to the neuron of the output layer corresponding to the class the training pixel belongs to. Given the attribute vector of an unknown pixel as input to the net, the neurons of the output layer provide the estimates of the probability that the unknown pixel belongs to the corresponding data classes. The classification is carried out by using the WTA decision rule to identify the most probable class. Training PNNs consists in the optimization of the Gaussian kernel by trials with different values of the “smoothing parameter” (Specht, 1990) which tunes the width of the Gaussian function.

## 3. Experimental results

### 3.1. Data set description

The considered data set refers to an agricultural area near the village of Feltwell (UK). We selected a section ( $250 \times 350$  pixels) of the scene acquired with two imaging sensors installed on an airplane: a Daedalus 1268 Airborne Thematic Mapper (ATM)



scanner and a PLC-band, fully polarimetric, NASA/JPL SAR sensor; as an example, Fig. 4(a) shows channel 9 of the ATM sensor and Fig. 4(b) shows channel L-HV (band L, polarization HV) of the SAR sensor. Images were registered by using the radar image as a reference. The ground truth was used to prepare a reference map to assess the classification accuracy (Fig. 5(a)). We considered the five numerically most representative agricultural classes (55657 pixels). The agricultural fields were randomly subdivided into two disjoint sets; 5124 training pixels were taken from the fields of one set, and 5820 test pixels from the fields of the other set. Fifteen channels were selected to form an “attribute vector” for each pixel. We selected the six ATM channels corresponding to TM channels in the visible and in the infrared spectrum, and the nine SAR channels in the PLC-band and HH, HV and VV polarizations. The noise affecting the intensity values was reduced by applying a simple running mean filtering to both the ATM ( $5 \times 5$  window) and the SAR ( $9 \times 9$  window) channels.

### 3.2. Results

Five MLPs with one or two hidden layers were used (Table 1), all with fifteen input units and five output units as the numbers of attributes and data classes, respectively. The number of neurons per hidden layer were chosen such that the condition in Eq. (1) holds, with  $\varepsilon$  set to 0.15. Training was carried out by the error backpropagation learning procedure using different “learning rates”  $\eta$  (i.e., 0.05 and 0.01). As an indication of the computational cost of learning, the best architecture on the training set (15–30–5,  $\eta = 0.05$ ) converged in 264 epochs to a mean square error value below a threshold set to 0.005. Table 1 gives the accuracies provided by the considered MLPs on both the training and test sets. The third column in Table 2 shows the class-by-class accuracies provided by the best architecture on the test set (i.e., 5–8–15,  $\eta = 0.01$ ). The MLP with this architecture was then applied to the

Fig. 4. Multisensor image utilized for experiments: (a) channel 9 of the ATM sensor and (b) channel L-HV (band L, polarization HV) of the SAR sensor.

Table 1

Percentages of correctly classified pixels by using MLPs. Different architectures with one or two hidden layers and various numbers of units per layer were used. Two “learning rates” ( $\eta$ ) were used for learning

Architecture	Training set ( $\eta = 0.01$ )	Test set ( $\eta = 0.01$ )	Training set ( $\eta = 0.05$ )	Test set ( $\eta = 0.05$ )
15–15–8–5	98.7%	79.9%	98.5%	86.3%
15–7–5–5	96.9%	82.3%	98.1%	76.1%
15–15–5	97.3%	87.9%	98.7%	86.0%
15–30–5	97.2%	88.2%	98.8%	86.2%
15–8–5	96.5%	89.6%	98.0%	82.3%
Mean value	97.3%	85.6%	98.4%	83.4%

whole image. Considering only the pixels belonging to the five classes selected, we obtained the classification map shown in Fig. 5(b) with a classification accuracy of 88.4%.

With regard to the SNNs, five CRNs were defined, i.e., one for each class; each CRN had the same architecture as the network depicted in Fig. 2. The fourth column of Table 2 gives the classification results provided by the SNNs on the test set. The SNNs were then applied to the whole image: the classification accuracy was equal to 86.5%; the classification map obtained is shown in Fig. 5(c).

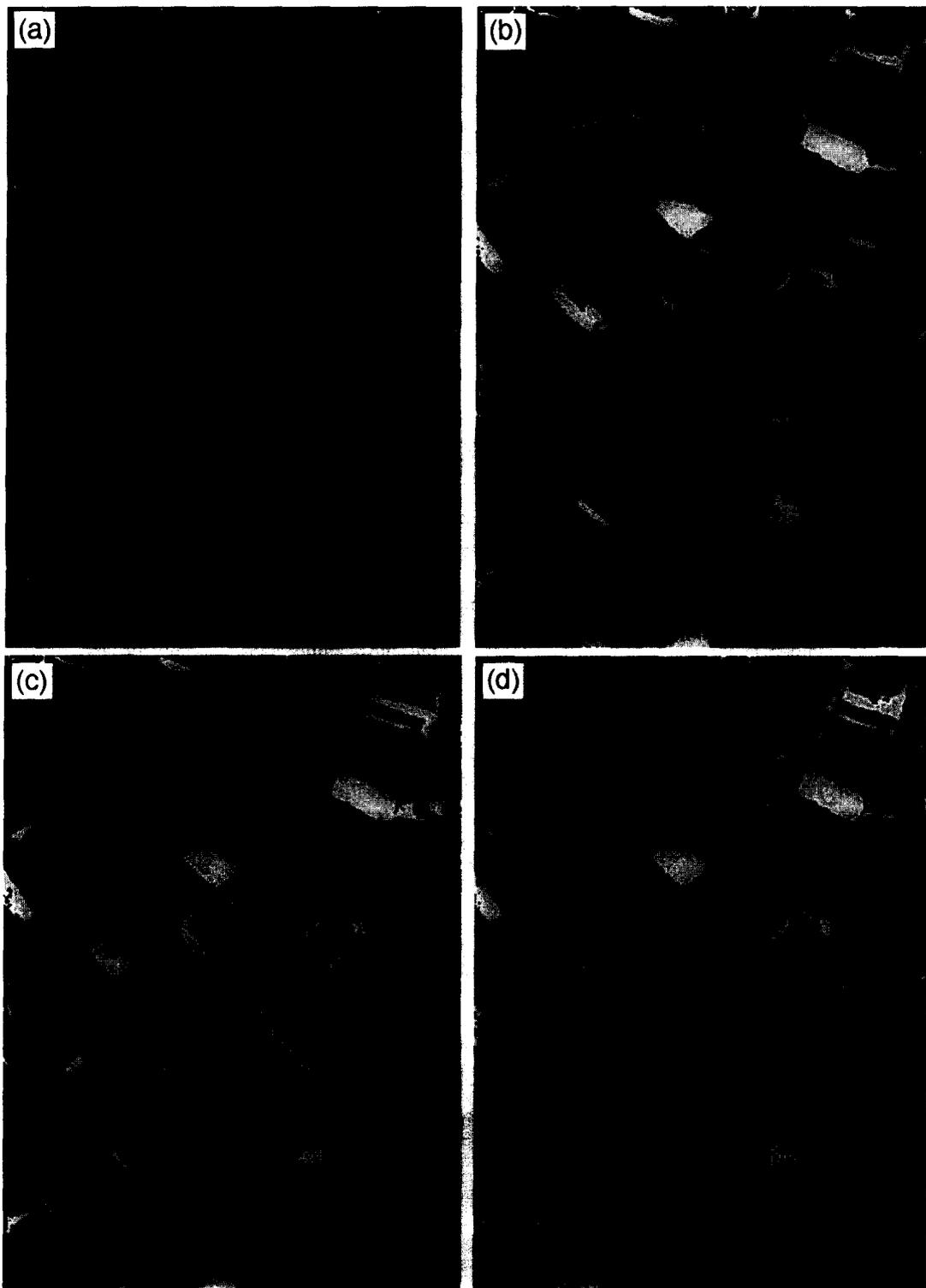
As an example of network interpretation, let us consider the simplified representation of the CRN depicted in Fig. 2, which relates to the “sugar beets” class. The importance of the ATM and SAR sensors in determining the output of the network is provided by the values of the weights between each sensor-related neuron and the output neuron. In the case of sugar beets, the two sensors have similar importance, as voting powers equal to 550 for the ATM sensor and to 450 for the SAR sensor were obtained. In order to understand the contribution of

each channel to the output of its Sensor-Related Committee, it is necessary to consider the voting power indicated on the connection starting from the related equivalent neuron of the first hidden layer. Concerning the ATM-Related Committee, channel 9 is the most important in the identification of sugar beets (voting power equal to 527), while channels 3 and 10 are not practically useful to identify such a class (voting powers equal to 2 and 19, respectively). The other channels all have an intermediate importance. Analogously, for the SAR-Related Committee, the most important channels are the band L with polarization HV and the bands P and C with polarization HH (voting powers equal to 277, 223 and 213, respectively), whereas the least important channels are the band C with polarizations HV and VV (voting powers equal to 11 and 15, respectively). It is also interesting to consult the piecewise-linear representation of constraints on the spectral response of each class in each spectral channel. For example, in the case of the ATM sensor and of the CRN of sugar beets, a pixel should be light in channels 5 and 10, from medium gray level to light in channel 2,

Table 2

Class-by-class accuracies in the classification of test pixels by using the best MLP on the test sets, SNNs, PNNs and the  $k$ -nn classifier. The overall accuracy is also provided that is given in terms of the percentage of correctly classified pixels and of the Kappa coefficient

Class	Number of pixels	MLP 15–8–5, $\eta = 0.01$	SNNs	PNNs	$k$ -nn
Sugar beets	2043	99.8%	99.5%	97.8%	97.4%
Stubble	1371	84.6%	85.9%	82.4%	88.4%
Bare soil	555	80.9%	79.3%	79.6%	76.0%
Potatoes	884	80.9%	74.2%	81.8%	86.4%
Carrots	967	88.2%	75.2%	89.3%	87.1%
Correct classification		89.6%	86.5%	88.6%	89.8%
Kappa coefficient		0.863	0.820	0.850	0.869



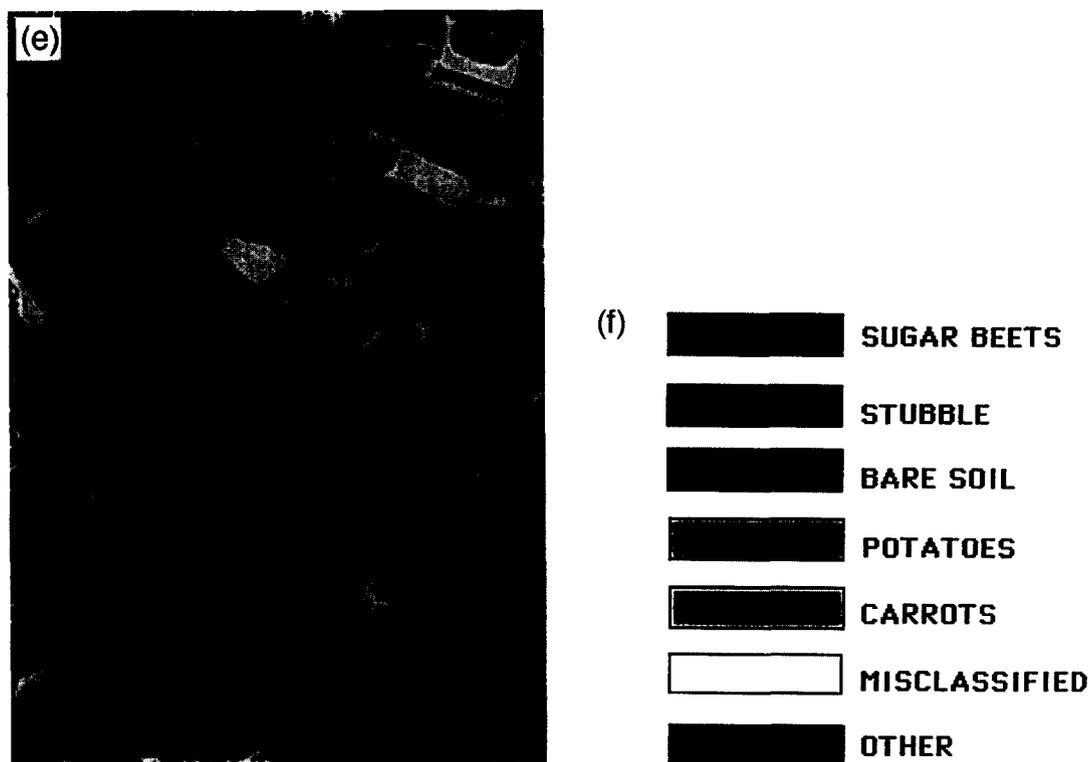


Fig. 5. Classification maps of the scene in Fig. 4: (a) reference map; (b) classification results of the MLP with the 15–8–5 architecture, trained with  $\eta = 0.01$ ; (c) classification results of the SNNs; (d) classification results of the PNNs; (e) classification results of the  $k$ -nn classifier; (f) legend.

and dark in channel 9, in order to give rise to a high output value of each equivalent neuron of the first hidden layer.

Concerning the PNNs, a three-layer network was defined, with a number of input units, a number of pattern units, and a number of output units that were equal to the numbers of attributes (15), of training pixels (5124), and of classes (five), respectively. A value 0.1 of the smoothing parameter of the Gaussian kernel function was selected experimentally as a result of the training phase. The classification accuracies on the test set that were obtained by using the PNNs are summarized, class by class, in the fifth column of Table 2. A classification accuracy of 88.7% was obtained on the whole image; the classification map provided by the classifier is shown in Fig. 5(d).

The results provided by the aforementioned three neuronal classifiers were then compared with those

provided by the  $k$ -nn classifier. The multisensor data set was classified using different  $k$  values, from  $k = 3$  up to  $k = 50$ . The best classification accuracy on the test set (i.e., 89.8%) was obtained by  $k = 25$  (Table 2). By this value of  $k$ , a classification accuracy of 89.9% was achieved on the whole image (the classification map is shown in Fig. 5(e)).

Finally, in order to better compare the accuracies provided by all the classifiers, the Kappa coefficient (Congalton et al., 1983; Rosenfield and Fitzpatrick-Lins, 1986) was computed (Table 2) as an effective measure to assess the accuracies of the different types of classifiers.

#### 4. Discussion and conclusions

The results reported in the previous section show that the best classification accuracy on the test set,

among the neural classifiers considered, was provided by the MLP with the architecture 15–8–5, trained with  $\eta = 0.01$ . The PNNs performed a little worse than such an MLP and a little better than the SNNs (there were differences of about 1% and 2% on the test set, respectively). However, the SNNs had the advantage of using classification criteria that are intelligible.

By analyzing the classification maps provided by the different classifiers (Fig. 5), one can see that some classification errors were common to all the classifiers. This is the case of some errors located on boundary pixels, due to the presence of both spurious ground coverings (e.g., trees, lanes) and mixed pixels between neighbouring fields. In addition, as both the training and test sets were taken from disjoint regions, fields of the test set with spectral responses not represented in the training set were likely to be completely misclassified by all the classifiers. A strong resemblance is evident in the errors produced by the PNNs and the  $k$ -nn classifier. This can be explained by the theoretical relationship existing between PNNs and  $k$ -nn, as also the  $k$ -nn classifier is based on the Parzen approach to density estimation (Fukunaga, 1990). On the other hand, also the maps provided by the SNNs and by the MLP exhibit strong similarities, due to the fact that SNNs are MLPs with particular architectures.

The SNNs and the PNNs had the advantage of requiring a simpler phase of architecture design than the MLPs, as they did not need experiments with different architectures.

From the viewpoint of processing time, the SNNs and the MLPs were slower than the PNNs in the training phase. The training of the PNNs was fast, as it required only a search in a one-dimensional space for the selection of the smoothing parameter. On the other hand, for the PNNs the size of the network and the classification time grow proportionally to the size of the training set. Therefore, with very large training sets, more complex techniques for the definition of the network are required (Burrascano, 1991). However, for our data set, all the network classifiers considered were quite fast in the classification phase.

Concerning the  $k$ -nn classifier, it provided a classification accuracy a little higher than those of all the neural classifiers considered. However, whereas the  $k$ -nn classifier was faster than the MLPs and the

SNNs in the training phase, it was much slower than all the considered neural networks in the classification phase. Depending on the application, speed may be more important in one or the other phase.

Considering the Kappa coefficient (Table 2), the differences among the accuracies provided by the various classifiers slightly increase. However, the above conclusions are substantially confirmed.

All the neural classifiers considered provided similar classification accuracies, only slightly worse than that of the  $k$ -nn classifier. This substantial equivalence of neural and statistical classifiers from the viewpoint of performances, which is basically in agreement with the results reported in the literature, was also confirmed by other experiments we carried out on different data sets (Roli, 1996). As a specific contribution, this paper should help to choose, on the basis of the requirements for the application at hand, from among different neural models, including the considered SNNs, which are interesting when an interpretation of the network operation is desired.

Finally, further research regarding SNNs is in progress: in particular, we are applying the concept of neural-network ensembles in order to improve the classification accuracy and the reliability of SNNs.

## Acknowledgements

This research was carried out in part within the framework of the research project "Integrated assessment of environmental degradation connected with forest fires in European areas", which was funded by the EEC (Environment Programme II, Contract N. EV5V-CT94-0481). The support is gratefully acknowledged.

## References

- Azimi-Sadjadi, M.R., S. Ghaloum and R. Zoughi (1993). Terrain classification in SAR images using principal components analysis and neural networks. *IEEE Trans. Geoscience Remote Sensing* 31, 511–515.
- Baum, E.B. and D. Haussler (1989). What size network gives valid generalization. *Neural Computation* 1, 151–160.
- Benediktsson, J.A., P.H. Swain and O.K. Ersoy (1990). Neural network approaches versus statistical methods in classification

- of multisource remote-sensing data. *IEEE Trans. Geoscience Remote Sensing* 28, 540–552.
- Benediktsson, J.A., J.R. Sveinsson and K. Arnason (1995). Classification and feature extraction of AVIRIS data. *IEEE Trans. Geoscience Remote Sensing* 33, 1194–1205.
- Bischof, H., W. Schneider and A.J. Pinz (1992). Multispectral classification of Landsat-images using neural networks. *IEEE Trans. Geoscience Remote Sensing* 30, 482–490.
- Burrascano, P. (1991). Learning vector quantization for the probabilistic neural network. *IEEE Trans. Neural Networks* 2, 458–461.
- Cacoullos, T. (1966). Estimation of multivariate density. *Ann. Inst. Stat. Math.* 18, 179–189.
- Congalton, R.G., R.G. Oderwald and R.A. Mead (1983). Assessing Landsat classification accuracy using discrete multivariate analysis statistical techniques. *Photogrammetric Engineering & Remote Sensing* 49, 1671–1678.
- Decatur, S.E. (1989). Applications of neural networks to terrain classification. *Proc. Internat. Joint Conf. on Neural Networks*, Washington DC, Vol. 1, 283–288.
- Ersoy, O.K. and D. Hong (1990). Parallel, self-organizing, hierarchical neural networks. *IEEE Trans. Neural Networks* 1, 167–178.
- Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press, New York, 2nd edition.
- Hertz, J., A. Krogh and R.G. Palmer (1991). *Introduction to the Theory of Neural Computation*. Addison-Wesley, The Advance Book Program, Reading, MA.
- Hinton, G.E. (1989). Connectionist learning procedures. *Artificial Intelligence* 40, 185–234.
- Hush, D.R. and B.G. Horne (1993). Progress in supervised neural networks. *IEEE Signal Process. Mag.* 10, 8–39.
- Hwang, J.N., S.R. Lay and R. Kiang (1993). Robust construction neural networks for classification of remotely sensed data. *Proc. World Congress on Neural Networks*, Portland, OR, Vol. 4, 580–584.
- Lee, J., R.C. Weger, S.K. Sengupta and R.M. Welch (1990). A neural network approach to cloud classification. *IEEE Trans. Geoscience Remote Sensing* 28, 846–855.
- Moody, J.E. (1991). Note on generalization, regularization, and architecture selection in nonlinear learning systems. *Proc. 1st IEEE Workshop on Neural Networks for Signal Processing*, IEEE Press, New York, 1–10.
- Musavi, M.T., K.H. Chan, D.M. Hummels and K. Kalantri (1994). On the generalization ability of neural network classifier. *IEEE Trans. Pattern Anal. Mach. Intell.* 16, 659–663.
- Paola, J.D. and R.A. Schowengerdt (1995a). A detailed comparison of backpropagation neural network and maximum-likelihood classifiers for urban land use classification. *IEEE Trans. Geoscience Remote Sensing* 33, 981–996.
- Paola, J.D. and R.A. Schowengerdt (1995b). A review and analysis of backpropagation neural networks for classification of remotely-sensed multi-spectral imagery. *Internat. J. Remote Sensing* 16, 3033–3058.
- Richard, M.D. and R.P. Lippmann (1991). Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Computation* 3, 461–483.
- Roli, F. (1996). Multisensor image recognition by neural networks with understandable behaviour. *Internat. J. Pattern Recognition Artificial Intelligence*, in press.
- Rosenfield, G.H. and K. Fitzpatrick-Lins (1986). A coefficient of agreement as a measure of thematic classification accuracy. *Photogrammetric Engineering & Remote Sensing* 52, 223–227.
- Rumelhart, D.E., J.L. McClelland and the PDP Research Group (1986). *Parallel Distributed Processing*, Vols. 1–2, MIT Press, Cambridge, MA.
- Salu, Y. and J. Tilton (1993). Classification of multispectral image data by the binary diamond neural network and by nonparametric, pixel-by-pixel methods. *IEEE Trans. Geoscience Remote Sensing* 31, 606–617.
- Serpico, S.B. and F. Roli (1995). Classification of multisensor remote-sensing images by structured neural networks. *IEEE Trans. Geoscience Remote Sensing* 33, 562–578.
- Specht, D.F. (1990). Probabilistic neural networks. *Neural Networks* 3, 109–118.
- Swain, P.H. and S.M. Davis (1978). *Remote Sensing: The Quantitative Approach*. McGraw-Hill, New York.
- Tollenaere, T. (1990). SuperSAB: fast adaptive back propagation with good scaling properties. *Neural Networks* 3, 561–573.
- Vogl, T.P., J.K. Mangis, A.K. Rigler, W.T. Zink and D.L. Alkon (1988). Accelerating the convergence of the back-propagation method. *Biol. Cybernet.* 59, 257–263.