Urban mapping using coarse SAR and optical data: outcome of the 2007 GRSS data fusion contest

Fabio Pacifici, Fabio del Frate, William J. Emery, Paolo Gamba, Jocelyn Chanussot

HAL Id: hal-00348851
https://hal.science/hal-00348851v1

Submitted on 22 Dec 2008

Urban Mapping Using Coarse SAR and Optical Data: Outcome of the 2007 GRSS Data Fusion Contest

Fabio Pacifici, *Student Member, IEEE*, Fabio Del Frate, *Member, IEEE*, William J. Emery, *Fellow, IEEE*, Paolo Gamba, *Senior Member, IEEE*, and Jocelyn Chanussot, *Senior Member, IEEE*

Abstract—The 2007 Data Fusion Contest that was organized by the IEEE Geoscience and Remote Sensing Data Fusion Technical Committee was dealing with the extraction of a land use/land cover maps in and around an urban area, exploiting multitemporal and multisource coarse-resolution data sets. In particular, synthetic aperture radar and optical data from satellite sensors were considered. Excellent indicators for mapping accuracy were obtained by the top teams. The best algorithm is based on a neural classification enhanced by preprocessing and postprocessing steps.

Index Terms—Data fusion, European Remote Sensing satellite (ERS), image classification, Landsat, neural networks (NNs), urban remote sensing.

TABLE I
DATA FUSION CONTEST DATA SET

Sensor	Acquisition date	Image ID	Mean	St. Dev.
ERS-1	August 13, 1992	SAR Date 1	1247.8	750.6
ERS-1	October 22, 1992	SAR Date 2	1475.7	740.1
ERS-1	June 24, 1993	SAR Date 3	1337.4	774.4
ERS-1	November 11, 1993	SAR Date 4	1457.5	727.0
ERS-1	October 3, 1994	SAR Date 5	1480.0	720.0
ERS-1	November 9, 1994	SAR Date 6	1500.1	807.5
ERS-1	July 22, 1995	SAR Date 7	100.2	80.9
ERS-2	July 23, 1995	SAR Date 8	115.5	92.9
ERS-2	August 27, 1995	SAR Date 9	118.1	90.9
Landsat-5	April 7, 1994	Landsat Date 1		
Landsat-7	October 8, 2000	Landsat Date 2		

I. INTRODUCTION

IN THE past two decades, monitoring urban centers and their peripheries at a regional scale has become an increasingly relevant topic for public institutions to keep track of the loss of agricultural land and natural vegetation due to urban development. In recent years, significant attention has focused on multisensor data fusion for remote sensing applications and, more specifically, for land cover mapping. Data fusion techniques combine information from multiple sources, providing potential advantages over a single sensor in terms of classification accuracy [1].

The characteristics of data acquired by optical and synthetic aperture radar (SAR) sensors greatly differ. Multispectral satellites such as Landsat provide information on the energy scattered and radiated by the Earth's surface in different wavelengths, from the visible to the thermal infrared, providing the ability to discriminate between different land cover classes such as vegetated areas, water surfaces, and urban centers. SAR sensors such as the European Remote Sensing satellites (ERS) 1/2 provide measurements in amplitude and phase related to the interaction of the Earth's surface with microwaves. These acquisitions (C-band) are characterized by high returns from buildings in urban areas and low and very low values from vegetated areas and water surfaces, respectively. Within residential areas, further discrimination is achievable because the low-density

areas are generally characterized by lower backscattering, given the wide streets and the presence of trees. This means that SAR sensors provide information that may not be obtained from optical sensors, and therefore, data fusion potentially provides improved results in the classification process compared to the conventional single-source classification results [2].

Data fusion may be accomplished at different information levels such as signal, pixel, feature, or decision. A signal-based fusion combines data from different sensors, creating a new input signal with improved characteristics over the original (e.g., a better signal-to-noise ratio). Information from different images can be merged in the pixel-based fusion to improve the performance of the processing tasks. A feature-based fusion combines features extracted from different signals or images, whereas a decision-level fusion consists of merging very dissimilar data at a higher level of abstraction [3].

In the data fusion literature, many alternative methods have been proposed for combining multisensor decisions by weighting the influence of each sensor. A common approach to multisource classification is to concatenate the data in a stacked vector and treat it as a unique set of measurements [4], but statistical classifiers can be difficult to deal with because it is not always possible to formulate reasonable assumptions about the distribution of features. Contextual information from neighboring pixels improves the accuracy of a pixel-based classification. For instance, the reliability of each information source can be estimated for each pixel using spatial features and can be integrated in a fuzzy-logic-based fusion scheme [5]. Markov random fields (MRFs) also provide a powerful methodological framework for modeling spatial and temporal context, allowing the images from different sensors and map data to be merged in a consistent way [6], [7]. Nonparametric approaches such as neural networks (NNs) [8], [9] or support vector machines [10] can be exploited because they do not require any specific probabilistic assumptions for class distribution. Hybrid approaches combining parametric methods and NNs have been proposed by Benediktsson *et al.* [11] by first
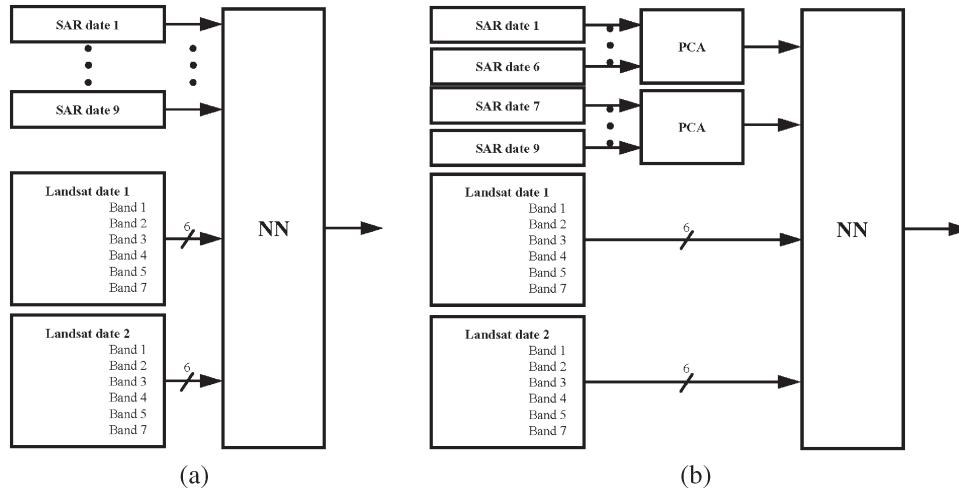
Fig. 1.   Classification schemes. (a) All 21 inputs feed the NN. The feature reduction is obtained by applying (b) the PCA only to the SAR imagery and considering the first component of dates $(1:6)$ and $(7:9)$.

separately treating each data source using statistical methods and then using an NN to obtain the consensus decision.

This letter is organized as follows. The 2007 Data Fusion Contest is presented in Section II, whereas Section III introduces the methodology of the winning team, discussing how it was designed. Finally, in Section IV, the classification results are presented and discussed.

II. 2007 DATA FUSION CONTEST

The Data Fusion Contest has been organized by the Data Fusion Technical Committee (DFTC) of the Geoscience and Remote Sensing Society (GRSS) of the IEEE and has been annually proposed since 2006. It is a contest open not only to DFTC members, but to everyone. The aim of the Data Fusion Contest is to evaluate existing methodologies at the research or operational level to solve remote sensing problems using data from different sensors. The main aim of this contest is to provide a benchmark to the researchers interested in a class of data fusion problems, starting with a contest and then allowing the data and results to be used as reference for the wider community, inside and outside the DFTC. The first issue of the contest was devoted to pansharpening [12]. In 2007, the contest was related to urban mapping using radar and optical data. A set of satellite radar and optical images (ERS amplitude data and Landsat multispectral data) was made available, and the task was to obtain a classified map as accurate as possible, relative to ground reference data. The reference data depicted land cover and land use classes for the urban area of interest. The final results led to an award for the methodology proposed in this letter, and the award was announced at the 2007 International Geoscience and Remote Sensing Symposium (IGARSS07) held in July 2007 in Barcelona, Spain.

The data set contains different SAR amplitude images, referring to the urban area of Pavia, Northern Italy (45.11 N, 9.09 E), acquired by ERS 1 and ERS 2 sensors during the period between 1992 and 1995 and two Landsat images acquired in 1994 and 2000, respectively, as reported in Table I [13]. The site has been chosen because it is typical of the diversity of urban land covers, uses, and features. Pavia is a small town with a very densely built center, some residential areas, industrial suburbs, and the Ticino river running through it [14].

The data sets and the resulting evaluations were all provided using a web portal specifically designed for the 2007 Data Fusion Contest. The portal allowed anyone to register, read the contest rules, understand the task for the contest, download the data, and upload the results. It also provided immediate feedback on the quality of the uploaded map by computing a few quantitative indices and correspondingly upgrading the *top ten ranking list*. In particular, because a classification map was the final output of the algorithms proposed by the contest's participants, this map was evaluated against ground reference data (unknown to the contestants), and the *kappa index* was used as a ranking value. The top three classification maps of the final list provided a *kappa index* larger than 0.93, an extremely good result, showing the excellence of the teams and the methodologies prompted by the contest task.

The contest attracted considerable attention in the remote sensing research community. More than 70 individuals registered to download the data sets and try their own approach. In the end, nine different teams uploaded more than 100 classification maps, most of them continuously refining their algorithm performance. The best result, awarded with a Certificate of Recognition during the Technical Committees' Dinner at IGARSS07, is presented in Section III.

III. WINNING ALGORITHM

The winning algorithm is based on an NN approach. The classification procedure can be divided into three steps: 1) preprocessing; 2) NN classification; and 3) postprocessing.

A. Preprocessing

Different classification schemes may be developed to achieve the desired classification accuracy. The first step to be considered regards the choice of inputs, which will successively feed the NN. The simplest scheme, shown in Fig. 1(a), feeds an NN with 21 inputs (9 SAR and $6 + 6$ optical), which requires a long training time. Therefore, the reduction of the input dimensionality is generally desirable. The principal component analysis (PCA) was applied to decrease the number of inputs used to train the NN. The PCA maps image data into a new uncorrelated coordinate system in which the data have greatest variance
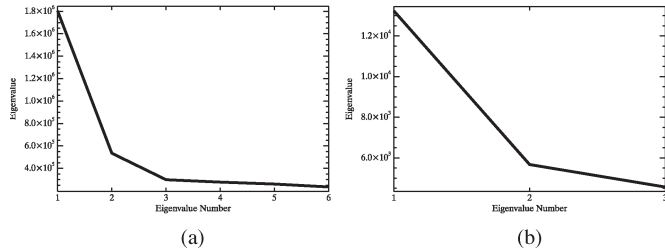
Fig. 2. PCA eigenvalues for dates (a) (1 : 6) and (b) (7 : 9) of the SAR imagery. The difference in magnitude of the eigenvalues in (a) and (b) is due to the different value distributions of the data sets that are considered.

along its first axis, the next largest variance along a second mutually orthogonal axis, and so on. The later principal components would be expected, in general, to show little variance [15].

The input space reduction can be applied to both SAR and optical imagery, but a loss of useful information may be encountered during this processing if the input variables significantly differ in magnitude because the PCA favors those variables that show the greatest variance, which generally are those with the larger absolute values.

Considering the different value distributions and characteristics of the data, the PCA was first applied to similar features, resulting in eight inputs: two from SAR, using the first component from dates 1 to 6 (1 : 6) and from 7 to 9 (7 : 9), as reported in Table I, and six from optical imagery, using for each pair of bands only the first PCA component. In Fig. 2, the PCA eigenvalues for the cases relative to dates (1 : 6) and (7 : 9) of the SAR imagery are shown. As expected, only the first principal component shows a large variance. However, the obtained eight inputs did not appear to contain enough information because a poor classification accuracy resulted (k-coeff of 0.6091 with respect to the validation set obtained by visual inspection). The successive experiment was to apply the PCA method only to SAR imagery, resulting in 14 inputs: two from SAR data [again, considering the first component of dates (1 : 6) and (7 : 9)] and six + six optical, as shown in Fig. 1(b). The resulting classification accuracy was k-coeff = 0.816. A further alternative might have been to exploit the correlation matrix (instead of the covariance matrix) in the PCA processing, but this resulted in a lower classification accuracy than the one obtained with the methodology described above.

B. NN Classification

An artificial NN may be viewed as a mathematical model composed of nonlinear computational elements named neurons operating in parallel and connected by links characterized by different weights. An NN is specified by its *neuron structure*, *architecture*, and *learning algorithm*.

1) Neuron Structure: The basic building block of an NN is the neuron. As described by many models over the years [16]–[18], a single neuron is an information processing unit generally characterized by several inputs and one output. Each neuron consists of three main parts: 1) the synaptic weight vector; 2) the net function; and 3) the activation function. The net function used is a weighted linear combination of inputs such as

$$o = \sum_{i=1}^{N} x_i w_i + \theta \qquad (1)$$

TABLE II
CLASSES OF INTEREST WITH RELATIVE COLOR MAP AND THE
NUMBER OF TRAINING AND VALIDATION SAMPLES

Class	Color	Tr. Set	Val. Set
City Center (CC)	Yellow	3783	4000
Residential Areas (RA)	Red	7572	8000
Sparse Buildings (SB)	Blue	5994	8000
Water (WA)	Cyan	893	1000
Vegetation (VE)	Green	591	10000

where o denotes the net function output $x_1, x_2, \ldots, x_N$ and $w_1, w_2, \ldots, w_N$ are the neuron inputs and the components of the synaptic weight vector, respectively, whereas θ is called the bias or threshold. The neuron output is achieved using the activation function related to the net function output through a linear or nonlinear transformation. The following Sigmoid activation function was used:

$$y(o) = \frac{1}{1 + e^{-o}}. \qquad (2)$$

2) Network Architecture: The multilayer perceptron (MLP) is the most widely used NN for solving decision-making problems for many different applications [19], [20]. It approximates an unknown input–output relationship, providing a nonlinear mapping between its inputs and outputs.

The architecture consists of different layers of neurons, whereas the interconnections are provided only between neurons of successive layers of the network. The first layer merely distributes the inputs to the internal stages of the network—there is no processing at this level. The last layer is the output, which provides the data that are processed. The layers between the input and the output are called hidden layers. The number of neurons that compose the input and output layers are directly related to the dimension of the input space and to the dimension of the desired output space, respectively. To limit saturation within the network, we scaled all the values of the input vector to the range [−1;+1]. At the same time, the component of the output vector corresponding to the true class has been set to 1, whereas the others are set to 0.

The five classes of interest considered for the contest are: 1) *City Center*; 2) *Residential Areas*; 3) *Sparse Buildings*; 4) *Water*; and 5) *Vegetation*. The number of training and validation samples are reported in Table II. The training pixels used were part of the larger set provided, whereas the validation samples were selected by a careful visual inspection of the scene and used only to have a rough estimation on the performances of the networks during the designing process.

Once the input and the output of the network are established, the critical step is to find the optimal number of units to be considered in the hidden layers. In fact, if the number of these neurons is too small, the input–output associative capabilities of the network are too weak. At the same time, this number should not be too large, otherwise, these capabilities might show a lack of generality being too narrowly tailored to the training set, and the computational complexity of the algorithm would be increased in vain [21]. Therefore, a compromise has to be found to select the most suitable number of hidden neurons for the optimal NN topology.

Two different approaches can be used to find the best architecture:

1) the *growing method* in which the starting network is small and the neurons are subsequently added until the optimization criteria is reached;

2) the *pruning method* in which the starting network is relatively large and the neurons are subsequently removed until the optimization criterion is satisfied.

The latter approach is used here. Therefore, after a reasonable evaluation in terms of classification accuracy, the chosen topology was 14-200-200-5. This estimation involved the analysis of the output variance of different topologies characterized by an increasing number of hidden neurons (by a factor of 40) starting from 14-40-40-5. In general, we expect that increasing the number of hidden neurons is effective up to a given number, after that, the overall error value does not significantly change.

3) Learning Algorithm: During the training phase, the network learns how to approximate an unknown input–output relation by adjusting the weight connections. This is done by minimizing an error function that quantifies the difference between the actual outputs and the desired outputs. There are several learning algorithms designed to minimize this error function. The learning algorithm used here is the scaling conjugate gradient method [22]. This is a member of the class of conjugate gradient methods, general-purpose second-order techniques that help minimize the goal functions of several variables. Second order indicates that such methods use the second derivatives of the error function, whereas a first-order technique, like standard backpropagation, only uses the first derivatives. To decide on when the training has to be interrupted, the *early-stopping* criterion based on the validation set was used [16].

A central problem in pattern recognition consists of minimizing the system complexity. In NNs, this issue often consists of minimizing the number of connection weights [23]. Pruning algorithms try to make NNs smaller by pruning unnecessary links or units. The advantages are the following.

1) The cost of an NN can be reduced (runtime, memory, and cost for hardware implementation).
2) The generalization of the NN can be improved.
3) Unnecessary input units can be pruned to give evidence of the relevance of input values.

To decide which links or units are less important and, thus, are candidates for removal, it is necessary to assess the relative importance of weight, namely its *saliency*. There are different approaches to determine the saliency of an element. The magnitude-based pruning is the simplest weight pruning algorithm [24], which is based on deleting links with small saliency. After the standard training, the link having the smallest magnitude value is removed (thus, the saliency of a link is just the absolute size of its weight). The network is then retrained, and the process is iteratively repeated until the training error reaches a specified limit. Although this method is very simple, it rarely yields worse results than the more sophisticated algorithms such as optimal brain damage, optimal brain surgeon, or skeletonization [25], [26].

Based on the validation set, the pruning process tried to remove neurons and connections following two distinct goals:

1) to reach the minimum of the classification error (on the validation set). The network is then retrained and the pruning is iteratively repeated until the classification error does not increase;
2) to reach the minimum number of connections taking into account a maximum decrease of the classification

TABLE III
ACCURACY DETAILS OF THE DIFFERENT TOPOLOGIES

Net ID	Number of Connections	Overall Accuracy (%)	k-coeff	Epochs
Full	43800	86.4	0.816	4227
net1	43684	86.9	0.824	10027
net2	43657	89.7	0.861	11377
net3	43600	84.3	0.787	14227
net4	42470	75.4	0.667	70727

accuracy of about 10% with respect to the fully connected network. Note that this task is only academic and does not address the contest.

After 4227 epochs of training, the fully connected NN (43 800 connections) correctly classified 86.4% of the validation patterns, as shown in Table III, which lists the accuracy details of the different network topologies.

The number of retraining epochs of the pruning phase has been set to 50. This means that the network was retrained for 50 cycles after the pruning of each connection. The fully connected NN has been pruned, reaching the minimum classification error with 43 657 connections (overall accuracy of 89.7%). Therefore, less than 0.5% of the initial connections were removed, for an improvement, in terms of classification accuracy, of 3.3%. However, no neuron was removed by the procedure. Successively, the pruning continued to further reduce the number of connections, expecting a decrease in the classification accuracy. As shown in Table III, 57 connections were sufficient to decrease the classification accuracy from 89.7% to 84.3%. The decrease in the classification accuracy of about 10% with respect to the fully connected network was reached with 42 470 connections (3.0% of the initial connections). These accuracies were based on the validation set obtained by the visual inspection of the scene and were only used to have a rough estimation on the performances of the different topologies during the designing process.

C. Postprocessing

Classified images often suffer from a lack of spatial coherence, which results in speckle or holes in homogeneous areas. This noise phenomenon appears as many isolated pixels or small groups of pixels whose classifications are different from those of their neighbors [27]. Postclassification processing techniques can be applied to further increase the classification accuracy. This is often achieved by analyzing the neighborhood for each pixel and removing the isolated pixels (*sieve* process) and then merging the small groups of pixels together to make more continuous and coherent units (*clump* process) [28]. These *sieve* and *clump* procedures were used here to reduce the effect of isolated pixels removing all regions smaller than the designated cluster dimension.

The *trial-and-error* strategy was used to define the optimal size with respect to the validation set, starting with a small cluster dimension (10 pixels). The highest classification accuracy was reached using a dimension of 142 pixels. Therefore, even though the smaller clusters may have been correctly classified, they were considered unreliable.

The classification map obtained after the postclassification processing reached an accuracy of 0.9698 in terms of k-coeff, relative to the validation set obtained by visual inspection.
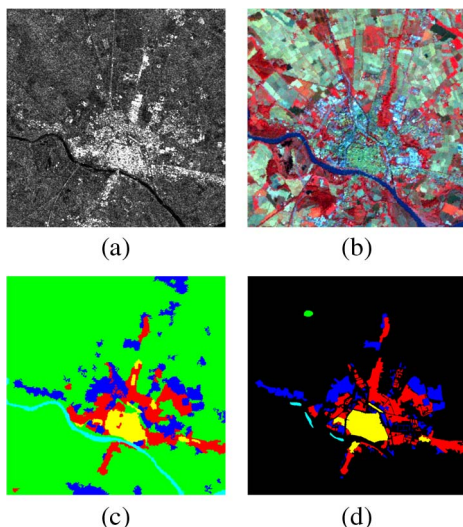
(a)　　　　　　　　(b)

(c)　　　　　　　　(d)

Fig. 3.　City of Pavia imaged by (a) SAR (backscattering amplitude) and (b) optical (bands RGB-431) sensors. In (c) and (d), the final classification map and the ground reference data are shown. The color code is in Table II.

TABLE IV
CONFUSION MATRIX WITH RESPECT TO THE
CONTEST GROUND REFERENCE DATA

	CC	RA	SB	WA	VE	%
CC	14174	1114	49	0	408	90.02
RA	700	31692	410	54	825	94.09
SB	66	472	31932	0	1246	94.71
WA	1	2	105	4174	245	92.20
VE	205	813	536	51	98199	98.39
%	93.58	92.96	96.67	97.55	97.30	96.11
k-coeff.=0.9393						

IV. DISCUSSIONS AND CONCLUSION

One of the major advantages of NNs with respect to statistically based classifiers is that NNs directly draw their own input–output discriminant relations from the data and do not require that a particular form of a probability density function be assumed [21].

The design of an NN-based framework for data fusion of ERS 1/2 and Landsat data set has been presented in Section III, highlighting the careful analysis that is required to find the most suitable network topology. The final classification map, shown in Fig. 3, reached an accuracy of 0.9393 in terms of k-coeff with respect to the unknown ground reference data that were used to rank the contest's results (whose confusion matrix is shown in Table IV).

Vegetated areas, sparse buildings, and water surfaces have shown higher classification accuracies (stemming from a higher class separability) with respect to the other two classes, which is an expected result because the responses of the areas characterized by high or moderate density of buildings (such as city center and residential, respectively) are quite similar in both optical and SAR sensors.

ACKNOWLEDGMENT

The authors would like to thank G. Pulina for designing and managing the 2007 contest website and implementing the automatic evaluation software.

REFERENCES

[1] D. Hall, *Mathematical Techniques in Multisensor Data Fusion*. Boston, MA: Artech House, 1992.
[2] A. H. Schistad Solberg, A. K. Jain, and T. Taxt, "Multisource classification of remotely sensed data: Fusion of Landsat TM and SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 4, pp. 768–778, Jul. 1994.
[3] M. Fauvel, J. Chanussot, and J. A. Benediktsson, "Decision fusion for the classification of urban remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 10, pp. 2828–2838, Oct. 2006.
[4] 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 Sens.*, vol. 28, no. 4, pp. 540–552, Jul. 1990.
[5] J. Chanussot, G. Mauris, and P. Lambert, "Fuzzy fusion techniques for linear features detection in multi-temporal SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 3, pp. 1292–1305, May 1999.
[6] A. H. Schistad Solberg, T. Taxt, and A. K. Jain, "A Markov random field model for classification of multisource satellite imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 34, no. 1, pp. 100–113, Jan. 1996.
[7] A. H. Schistad Solberg, "Contextual data fusion applied to forest map revision," *IEEE Trans. Geosci. Remote Sens.*, vol. 37, no. 3, pp. 1234–1243, May 1999.
[8] A. V. Bogdanov, S. Sandven, O. M. Johannessen, V. Y. Alexandrov, and L. P. Bobylev, "Multisensor approach to automated classification of sea ice image data," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 7, pp. 1648–1664, Jul. 2005.
[9] M. Q. Nguyen, P. M. Atkinson, and H. G. Lewis, "Superresolution mapping using a Hopfield neural network with fused images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 3, pp. 736–749, Mar. 2006.
[10] S. Zhao, "Remote sensing data fusion using support vector machine," in *Proc. IGARSS*, Sep. 20–24, 2004, vol. 4, pp. 2575–2578.
[11] J. A. Benediktsson, J. R. Sveinsson, and P. H. Swain, "Hybrid consensus theoretic classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 35, no. 4, pp. 833–843, Jul. 1997.
[12] L. Alparone, L. Wald, J. Chanussot, C. Thomas, P. Gamba, and L. M. Bruce, "Comparison of pansharpening algorithms: Outcome of the 2006 GRS-S data fusion contest," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 10, pp. 3012–3021, Oct. 2007.
[13] P. Gamba, "A collection of data for urban area characterization," in *Proc. IGARSS*, Sep. 20–24, 2004, vol. 1, pp. 69–72.
[14] F. Dell'Acqua, P. Gamba, and G. Lisini, "Improvements to urban area characterization using multitemporal and multiangle SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 9, pp. 1996–2004, Sep. 2003.
[15] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis*. Berlin, Germany: Springer-Verlag, 2006.
[16] C. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1995.
[17] N. K. Kasabov, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*. Cambridge, MA: MIT Press, 1996.
[18] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models*. Cambridge, MA: MIT Press, 2001.
[19] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 683–697, Sep. 1992.
[20] F. Del Frate, A. Petrocchi, J. Lichtenegger, and G. Calabresi, "Neural networks for oil spill detection using ERS-SAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 5, pp. 2282–2287, Sep. 2000.
[21] F. Del Frate, F. Pacifici, G. Schiavon, and C. Solimini, "Use of neural networks for automatic classification from high-resolution images," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 4, pp. 800–809, Apr. 2007.
[22] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Netw.*, vol. 6, no. 4, pp. 525–533, 1993.
[23] B. Hassibi, D. G. Stork, and G. Wolf, "Optimal brain surgeon and general network pruning," in *Proc. IEEE Int. Conf. Neural Netw.*, San Francisco, CA, Apr. 1993, pp. 293–300.
[24] T. Kavzoglu and P. M. Mather, "Pruning artificial neural networks: An example using land cover classification of multi-sensor images," *Int. J. Remote Sens.*, vol. 20, no. 14, pp. 2787–2803, Sep. 1999.
[25] A. Zell *et al.*, *SNNS, Stuttgart Neural Network Simulator, User's Manual v.4.2*. Stuttgart, Germany: Univ. Stuttgart.
[26] T. Ragg, H. Braun, and H. Landsberg, "A comparative study of neural network optimization techniques," in *Proc. ICNNGA*, Norwich, U.K., 1997, pp. 343–347.
[27] H. Huang, J. J. Legarsky, S. Gudimetla, and C. H. Davis, "Post-classification smoothing of digital classification map of St. Louis, Missouri," in *Proc. IGARSS*, Sep. 2004, vol. 5, pp. 3039–3041.
[28] R. A. Schowengerdt, *Techniques for Image Processing and Classification in Remote Sensing*. New York: Academic, 1983.