TOR VERGATA UNIVERSITY

Department of Computer, Systems and Production

GeoInformation PhD Programme

XXII Cicle

NOVEL NEURAL NETWORK-BASED ALGORITHMS FOR URBAN CLASSIFICATION AND CHANGE DETECTION FROM SATELLITE IMAGERY

Fabio Pacifici

Supervisor: **William J. Emery**

Supervisor: **Domenico Solimini**

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT
TOR VERGATA UNIVERSITY
ROME, ITALY

March 2010

TOR VERGATA UNIVERSITY

Department of Computer, Systems and Production

GeoInformation PhD Programme

XXII Cicle

The examining committee has read the thesis entitled "**Novel Neural Network-Based Algorithms for Urban Classification and Change Detection from Satellite Imagery**" by Fabio Pacifici and recommend it in partial fulfillment of the requirements for the degree of Doctory of Philosophy.

Date: _____

Chair:

_____
William J. Emery

Examining Committee:

_____
Paolo Gamba

_____
Martino Pesaresi

Additional Members

_____
Mihai Datcu

_____
Victor H. Leonard

_____
Domenico Solimini

TOR VERGATA UNIVERSITY

Department of Computer, Systems and Production
GeoInformation PhD Programme
XXII Cicle

Author: **Fabio Pacifici**

Title: **Novel Neural Network-Based Algorithms for Urban Classification and Change Detection from Satellite Imagery**

Fabio Pacifici

Dedicated to my family and Eidelheid

One closes behind one the little gate of mere boyishness
and enters an enchanted garden.
Its very shades glow with promise.
Every turn of the path has its seduction.
And it isn't because it is an undiscovered country.
One knows well enough that all mankind had streamed that way.
It is the charm of universal experience
from which one expects an uncommon or personal sensation,
a bit of one's own.

Joseph Conrad

Table of Contents

Abstract

Human activity dominates the Earth's ecosystems with structural modifications. The rapid population growth over recent decades and the concentration of this population in and around urban areas have significantly impacted the environment. Although urban areas represent a small fraction of the land surface, they affect large areas due to the magnitude of the associated energy, food, water, and raw material demands. Reliable information in populated areas is essential for urban planning and strategic decision making, such as civil protection departments in cases of emergency.

Remote sensing is increasingly being used as a timely and cost-effective source of information in a wide number of applications, from environment monitoring to location-aware systems. However, mapping human settlements represents one of the most challenging areas for the remote sensing community due to its high spatial and spectral diversity. From the physical composition point of view, several different materials can be used for the same man-made element (for example, building roofs can be made of clay tiles, metal, asphalt, concrete, plastic, grass or stones). On the other hand, the same material can be used for different purposes (for example, concrete can be found in paved roads or building roofs). Moreover, urban areas are often made up of materials present in the surrounding region, making them indistinguishable from the natural or agricultural areas (examples can be unpaved roads and bare soil, clay tiles and bare soil, or parks and vegetated open spaces) [1].

During the last two decades, significant progress has been made in developing and launching satellites with instruments, in both the optical/infrared and microwave regions of the spectra, well suited for Earth observation with an increasingly finer spatial, spectral and temporal resolution. Fine spatial sensors with metric or sub-metric resolution allow the detection of small-scale objects, such as elements of residential housing, commercial buildings, transportation systems and utilities. Multi-spectral and hyper-spectral remote sensing systems provide additional discriminative features for classes that are spectrally similar, due to their higher spectral resolution. The temporal component, integrated with the spectral and spatial dimensions, provides essential information, for example on vegetation dynamics. Moreover, the delineation of temporal homogeneous patches reduces the effect of local spatial heterogeneity that often masks larger spatial patterns.

Nevertheless, higher resolution (spatial, spectral or temporal) imagery comes with limits and challenges that equal the advantages and improvements, and this is valid for both optical and synthetic aperture radar data [2].

This thesis addresses the different aspects of mapping and change detection of human settlements, discussing the main issues related to the use of optical and synthetic aperture radar data. Novel approaches and techniques

are proposed and critically discussed to cope with the challenges of urban areas, including data fusion, image information mining, and active learning. The chapters are subdivided into three main parts. Part I addresses the theoretical aspects of neural networks, including their different architectures, design, and training. The proposed neural networks-based algorithms, their applications to classification and change detection problems, and the experimental results are described in Part II and Part III.

Specifically:

Part I

neural network models: the mathematical formalisms of feed-forward and feed-back architectures are discussed in Chapter 2, while Chapter 3 introduces a novel Pulse Coupled model. Chapter 4 addresses the mathematical aspects of neural networks for the feature selection problem

Part II

classification of urban areas: the diverse multi-spatial, multi-temporal, and multi/hyper-spectral aspects of urban mapping are discussed in detail in Chapter 6, Chapter 7, and Chapter 8, respectively. The problem of data fusion between sensors is then addressed in Chapter 9, whereas the concepts of image information mining and active learning are discussed in Chapter 10 and Chapter 11, respectively

Part III

change detection of urban areas: a parallel approach based on an neural architecture is discussed in Chapter 13, while a change detection application of Pulse Coupled Neural Networks is addressed in Chapter 14

Fabio Pacifici
Longmont, CO, U. S. A.
February 7, 2010

Part I

Neural network models

Chapter 1

Neural networks in remote sensing

Following their resurgence as a research topic in the second half of the eighties [3][4], neural networks expanded rapidly within the remote sensing community. By the end of the same decade a group of researchers started looking at them as an effective alternative to more traditional methods for extracting information from data collected by airborne and space-borne platforms [5][6]. Indeed, the possibility of building classification algorithms without the assumptions required by the Bayesian methods on the data probability distributions seemed rather attractive to researchers who have worked with multi-dimensional data.

For the case of parameter retrieval, neural networks had the advantage of determining the input-output relationship directly from the training data with no need to seek for an explicit model of the physical mechanisms, which were often nonlinear and poorly understood [7]. Moreover, it was shown that multi-layer feed-forward networks formed a class of universal approximators, capable of approximating any real-valued continuous function provided a sufficient number of units in the hidden layer were considered [8]. There were enough reasons to explore the potential of such neuro-computational models, also known as associative memory based models, for a wide range of remote sensing applications. This actually happened throughout the nineties and, with less emphasis, is continuing today.

Image classification, from synthetic aperture radar imagery to the latest hyper-spectral data, has probably been one of the most investigated fields in this context. However, the use of neural networks to retrieve bio-geophysical parameters from remotely sensed data has been an active research area. In particular, the synergy between these algorithms and radiative transfer electromagnetic models represented a new and effective way to replace the empirical approaches, often based on limited seasonal or regional data with small generalization capability. The combined use of electromagnetic models and neural networks is a topic treated in many published studies taking into account the most diverse scenarios: from the inversion of radiance spectra to infer atmospheric ozone concentration profiles [9], to the retrieval of snow parameters from passive microwave measurements [10] or to the estimation of sea water optically active parameters from hyper-spectral data [11].

As far as the network topology is concerned, the feed-forward multi-layer perceptron is probably the most commonly used for remote sensing applications, but the choice is widely spread over the numerous alternatives which include Radial Basis Function, recurrent architectures, as well as the unsupervised Kohonen Self Organizing Maps [12] and Pulse Coupled Neural Networks. This extensive research activity confirmed that neural networks, besides representing a new and easy way of machine learning, possessed particularly interesting properties, such as the capability of capturing subtle dependencies among the data, an inherent fault tolerance due to their parallel and distributed structure, and a capability of positively merging pieces of information stemming from different sources.

Do neural networks have only advantages? Obviously not. The conventional back-propagation learning algorithm can be stuck in a local minimum. Moreover, the choice of the network architecture (i.e. number of hidden layers and nodes in each layer, learning rate as well as momentum), weight initialization and number of iterations required for training may significantly affect the learning performance. Addressing these issues is one of the dominant themes of current research. Some authors suggest that Support Vector Machines may have accuracies similar to those obtained with neural networks without suffering from the problem of local minima and with limited effort for architecture design (although their training involves nonlinear optimization, the objective function is convex, and so the solution of the optimization problem is relatively straightforward [13]). Other authors remain focused on neural models, aiming at automating the selection of the parameters characterizing the algorithm or searching for improvements in the learning procedures [14]. A different ongoing line of research is more sensitive to the opportunities that should be provided by the data of the last generation Earth observation missions. Hence, it is more dedicated to exploit the aforementioned neural networks potential for new and more complex applications.

The following chapters illustrate the fundamental characteristics of neural networks. In particular, feed-forward and feed-back architectures are introduced and discussed in detail in Chapter 2, while Chapter 3 introduces a novel Pulse Coupled model. Chapter 4 addresses the theoretical aspects of neural networks for the feature selection problem, a central issue in many remote sensing data analyses. Further, in the same chapter, a support vector model is discussed for comparison purposes.

Chapter 2

Feed-forward and feed-back networks

Part of this Chapter's contents is extracted from:

1. F. Pacifici, F. Del Frate, C. Solimini, W. J. Emery, "Neural Networks for Land Cover Applications", in *Computational intelligence for remote sensing*, M. Graña, R.Duro, Eds. Studies in Computational Intelligence, Springer, vol. 133, pp. 267-293, Springer, ISBN: 978-3-540-79352-6, 2008

An artificial neural network (NN) is a system based on the operation of biological neural system. It may be viewed as a mathematical model composed of non-linear computational elements, named neurons, operating in parallel and connected by links characterized by different weights. Different NN models exist, which consequently requires different types of algorithms. NN models are mainly specified by:

- neuron structure
- network topology
- training or learning rules

Details of these three elements follow.

2.1 Neuron structure

The basic building block of a NN is the neuron. As described by many models over the years [15], [16], [17], [18], [19], a single neuron is an information processing unit generally characterized by several inputs and one output. Each unit performs a relatively simple job: it receives input from neighbors or external sources and uses this to compute an output signal which is propagated to other units. As shown in Figure 2.1, there are three basic components in the functional model of the biological neuron:

- weight vector
- network function
- activation function

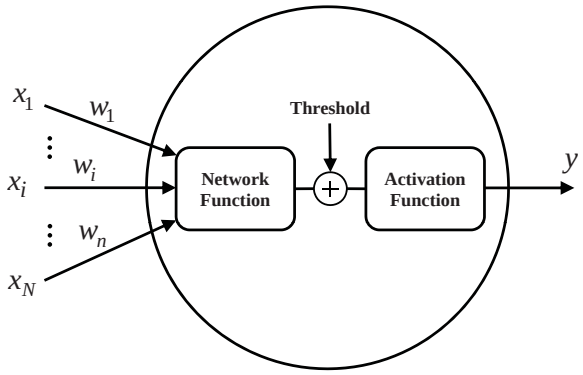


Figure 2.1: Neuron model: the input vector is combined with the weight vector through the network function. Then, the activation function is applied, producing the neuron output.

Table 2.1: Network functions.

Network Functions	Formula
Linear	$y = \sum_{i=1}^{N} x_i w_i + \theta$
Higher order (2^{nd} order)	$y = \sum_{i=1}^{N} \sum_{j=1}^{N} x_i x_j w_{ij} + \theta$
Delta ($\sum - \prod$)	$y = \prod_{i=1}^{N} x_i w_i$

The synapses of the neuron are modeled as weights. The strength of the connection between an input and a neuron is noted by the value of the weight. Negative weight values reflect inhibitory connections, while positive values designate excitatory connections. The next two components model the actual activity within the neuron cell. The network function sums up all the inputs modified by their respective weights. This activity is referred to as a linear combination. Finally, an activation function controls the amplitude of the output of the neuron. An acceptable range of output is usually between 0 and 1, or -1 and 1.

An example of a commonly used network function is the weighted linear combination of inputs such as:

$$y = \sum_{i=1}^{N} x_i w_i + \theta \tag{2.1.1}$$

where y denotes the network function output, $x_1, x_2, ..., x_N$ and $w_1, w_2, ..., w_N$ are the components of the neuron input and synaptic weight vectors, while θ is called the bias or threshold. The biological interpretation of this threshold is to inhibit the firing of the neuron when the cumulative effect of the inputs does not exceed the value of θ. Many other network functions have been proposed in the literature, which vary with the combination of inputs. A brief list of network functions is shown in Table 2.1.

The neuron output is achieved by the activation function related to the network function output through a linear or non-linear transformation. In general, these activation functions are characterized by saturation at a minimum and a maximum value and by being non-decreasing functions. In the most common network architectures, the error minimization yields an iterative non-linear optimization algorithm. The performance of the learning algorithms is related to their abilities to converge in a relatively small number of iterations to a minimum of the global error function which depends on the network weights. Many of these optimization methods are based on the gradient descent algorithm which requires that the activation function is continuously differentiable. The fundamental types of activation functions are shown in Table 2.2.

Table 2.2: Neuron activations functions.

Activation Functions	Neuron Output
Sigmoid	$f(x) = \frac{1}{1+e^{-x}}$
Hyperbolic tangent	$f(x) = tanh\frac{x}{T}$
Inverse tangent	$f(x) = \frac{2}{\pi} tan^{-1}\frac{x}{T}$
Threshold	$f(x) = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \end{cases}$
Gaussian radial basis	$f(x) = exp\Big\{-||x-m||^2/\sigma\Big\}$
Linear	$f(x) = ax + b$

2.2 Network topology

Although a single neuron can solve simple information processing functions, the advantage of the neural computation approach comes from connecting neurons in networks, where several units are interconnected to form a distributed architecture. In general, the way in which the units are linked together is related to the learning algorithm used to train the network. Neurons can be fully connected, which means that every neuron is connected to every other one, or partially connected, where only particular connections between units in different layers are allowed.

The feed-forward and the feed-back (or recurrent) architectures are the two types of connection patterns that can be distinguished. In a feed-forward architecture there are no connections back from the output, while in a feed-back architecture there are connections from output to input neurons. The multi-layer perceptron (MLP) network, described in Section 2.2.1, is one of the most popular feed-forward models among all the existing paradigms. Generally speaking, feed-forward architectures are *static*, since they provide only one set of output values rather than a sequence of values from a given input. This means that these architectures are memory-less in the sense that their response to an input is independent of the previous network status. On the other hand, recurrent networks are *dynamic* systems and these networks remember the previous states and the actual state does not only depend on the input signals. The Elman network, described in Section 2.2.2, is of this type.

2.2.1 Multi-layer perceptron

Due to its high generalization ability, the MLP is the most widely used neural network for solving decision-making problems for many different applications [20]. An MLP is a more complex version of the original perceptron model proposed in the early 50's which consisted of a single neuron that utilized a linear network function and a threshold activation neuron function, respectively (see Table 2.1 and Table 2.2). To overcome the linear separability limitation of the perceptron model, the MLP neural networks were designed to have continuous value inputs and outputs, and nonlinear activation functions. In fact, in many practical applications, such as pattern recognition, the major problem of the simple perceptron model is related to the learning algorithm, which applies only to linearly separable samples. An MLP approximates an unknown input-output relationship providing a nonlinear mapping between its inputs and outputs. The MLP architecture is a feed-forward type, therefore there are no connections back from the output to the input neurons and each neuron of a layer is connected to all neurons of the successive layer without feed-back to neurons in the previous layer. The architecture consists of different layers of neurons, while the interconnections are provided only between neurons of successive layers of the network. The

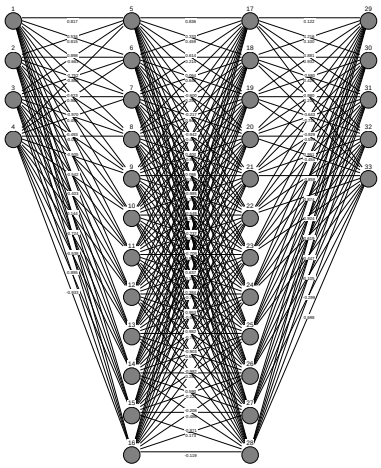


Figure 2.2: Two hidden layers MLP network. Each circle represents a single neuron.

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 final result. The layers between the input and the output are called hidden layers. The number of neurons which 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. Even if the MLP is one of the most used neural network models, there are several problems in designing the whole topology of the network since it is necessary to decide the number of units in the hidden layers, which in general may have any arbitrary number of neurons. Therefore, the network topology providing the optimal performance should be selected to achieve the desired behavior. If the number of neurons is too small, the input-output associative capabilities of the network are too weak. On the other hand, this number should not be too large; otherwise, these capabilities can show a lack of generality for networks that are tailored too much to the training set. Thus, the computational complexity of the algorithm would be increased in vain. In Figure 2.2 is illustrated a MLP network with two hidden layers.

2.2.2 Elman neural network

The MLP model discussed in the previous section is a static architecture which, after the training phase provides an output that only depends on the vector input to the network. In this sense, the MLP model has no memory since it does not take into account the dependence of an individual input on the others processed previously. In general, a classification algorithm should be capable of exploiting the information embedded in multi-temporal measurements, since time series provide significant contributions to understanding processes in engineering applications. In many remote sensing cases, it is necessary to deal with objects that continuously evolve their state following precise dynamical conditions. For example, the characteristics of an agricultural field change during the year due to the phenological cycle of the plants which cover sprouting, growth and the senescence or harvest of the vegetation. In other periods of the year, when soil characteristics dominate, variations are mainly due to soil conditions, such as moisture effects. As a consequence, little difference can be observed between samples belonging to the same scene and year, while year-to-year variations may be large. If time series information is available, these measurements show temporal correlations between samples (each extracted at a different time position) which can be considered as additional information.

Recurrent networks can be used when the temporal variability of the inputs contains useful information since they have feed-back connections from the neurons in one layer to neurons in a previous layer. The successive state

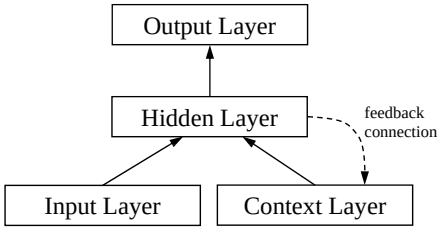


Figure 2.3: Architecture of the Elman NN. Context units can be considered as additional input units.

of the network depends not only on the actual input vector and connection weights, but also on the previous states of the network. This means that, given a constant input vector, recurrent networks do not necessarily provide a constant output vector, as the static model would, having memory of the previous inputs. The main feature of recurrent network based modeling is that it leads to the discovery of information on the time series variations by training only a limited number of connection weights compared with the MLP approach, resulting in a faster analysis. Given a time-series $x(t), x(t-1), x(t-2), ..., x(t-N)$ the MLP model creates inputs $x_0, x_1, x_2, ..., x_N$ which represent the last $N+1$ values of the input vector (only a time window of the input vector is used to feed the network). Therefore, the disadvantage is that the input dimensionality of MLP is multiplied by $N+1$, leading to a very large network, which is, slow and difficult to train. Depending on the architecture of the feed-back connections, there are two general models:

1. fully recurrent neural networks, in which any node is connected to any other node
2. partially recurrent neural networks, in which only specific feed-back connections to the input layer are established

The partially recurrent networks proposed by Elman at the beginning of the 90's for string recognition consist of four main layers, as illustrated in Figure 2.3:

- input
- hidden
- output
- context

The weights assigned to the feed-back connections are not all adjustable. In fact, hidden neurons are feed-back connected to the context units with fixed weights (dashed line in Figure 2.3), while only feed-forward connections require training (solid line in Figure 2.3).

The basic structure of the Elman NNs is reported in Figure 2.4. As discussed previously, for each hidden neuron, an extra context unit is added: there are recurrent connections from the hidden to the context layers by means of the unit delay. Furthermore, the context neurons are fully forward connected with all hidden units. Since the context layer consists of supplementary input units whose values are feed-back from the hidden layer, this architecture is considered as a more general feed-forward neural network with additional memory units and local feed-back [21]. In fact, a simple feed-forward network remains after removing all recurrent links of a recurrent network.

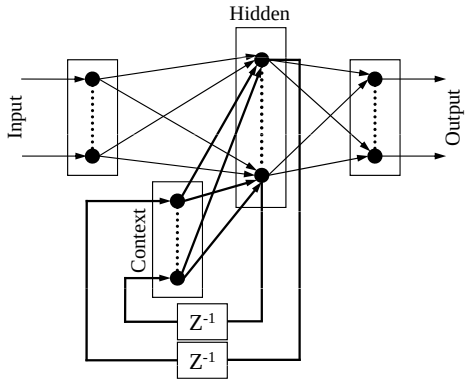


Figure 2.4: Feed-back connections between layers in the Elman architecture (Z^{-1} is the unit delay).

2.3 Learning algorithms

2.3.1 Back-propagation

During the training phase, the network learns how to approximate an unknown input-output relation by adjusting the weight connections. This is done by receiving both the raw data as inputs and the target as outputs which are used to minimize an error function. There are several learning algorithms designed to minimize this error function related to the differences between the inputs $x_1, ..., x_N$ and the target outputs $t_1, ..., t_N$. One of the most widely used is the back-propagation algorithm which minimizes the error function by changing the vector weights. To achieve this, it is assumed that the error function may be written as the sum over all training patterns, defined by:

$$E = \sum_{p=1}^{N} E_p \tag{2.3.1}$$

and differentiable with respect to the output variables. A suitable function with these behaviors is the Sum-of-Squares Error (SSE) defined by:

$$E_p(w) = \frac{1}{2} \sum_{i=1}^{N} [t_i(p) - y_i(p)]^2 \tag{2.3.2}$$

There are numerous nonlinear optimization algorithms available to minimize the above error function. Basically, they adopt a similar iterative formulation:

$$w(t+1) = w(t) + \Delta w(t) \tag{2.3.3}$$

where $\Delta w(t)$ is the correction made to the current weights $w(t)$. Each time the weights are updated is called an epoch. Many algorithms differ in the form of $\Delta w(t)$. Among them, the well-known *gradient descent procedure* changes the weight vector by a quantity η proportional to the negative gradient as in:

$$\Delta w = -\eta \nabla E_p(w) \tag{2.3.4}$$

where η governs the overall speed of weight adaptation. In general, η should be sufficiently large to avoid oscillations within a local minimum of E_p. In fact, the algorithm may overshoot from a different direction during

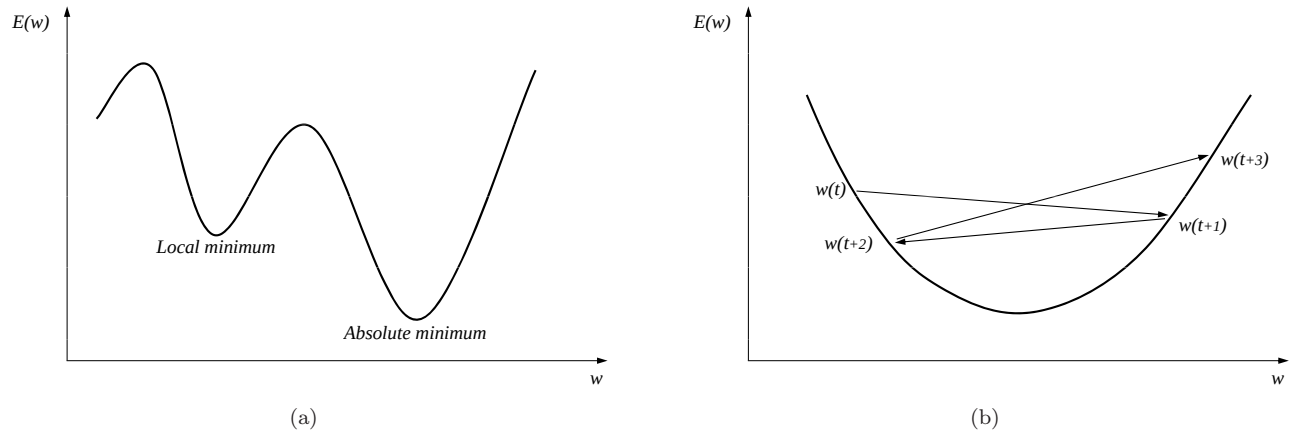


Figure 2.5: Sum-of-square error as a function of weights: (a) a suitable choice of η permits to reach the minimum of the function jumping out from local minimum, otherwise (b) the weights change their direction oscillating around the minimum.

the learning phase, resulting in weights that are changed to values around the minimum, but without reaching it. This problem often occurs for deep and narrow minima of the error function because, in this case, ∇E_p is large and, therefore, Δw is also large. At the same time, η should be sufficiently small so that the minimum does not jump over the correct value. These concepts are illustrated in Figure 2.5.

The back-propagation algorithm uses a step-by-step procedure [22]. During the first phase, after an appropriate value of η has been chosen, the weights are initialized to random, small values and the training patterns are propagated forward through the network. Then, the outputs obtained are compared with the desired values, computing the preliminary sum-of-square cost function for the data. During the second phase, a backward pass through the network permits the computation of appropriate weight changes. The Conjugate Gradient Method (CGM) belongs to the second-order techniques that minimize the error function. Second order indicates that these methods exploit the second derivative of the error function instead of a first derivative such as back-propagation. This latter method, always proceeds down the gradient of the error function, while CGM proceeds in a direction which is conjugate to the directions of the previous steps. This approximation gives a more complete information about the search direction and step size. The Scaling Gradient (SCG) method [23] adds to the CGM a scaling approach to the step size in order to speed up the algorithm. Basically, in most application problems, the error function decreases monotonically towards zero for a increasing number of learning iterations.

2.3.2 Back-propagation through time

From the operative point of view, a standard MLP training algorithm (e.g. the ordinary back-propagation algorithm) can be used to adjust the connection weights of an Elman network. In fact, the context units can be considered as input units and, therefore, the total network input vector consists of two stages. The first is the actual input vector, which is the only input of the feed-forward network. The second is the context vector, which is given through the next context output vector in every step. In this way, an Elman network can be trained by a simple feed-forward network algorithm that receives the total network input as part of the actual input vector. In particular, the back-propagation algorithm for recurrent networks consists of:

- initializing the context units
- presenting the input patterns to the input layer

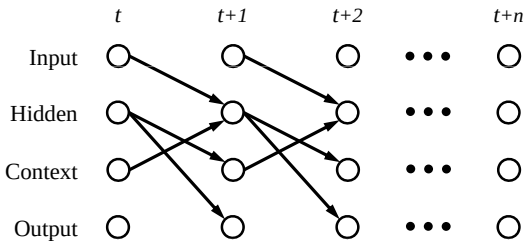


Figure 2.6: Equivalent discrete time model of the original recurrent network.

- running the recurrent network for a determined number of iterations
- comparing the results with the desired output in terms of error
- propagating backward the obtained errors for the same number of iterations
- updating the weights

Many practical methodologies based on the back-propagation algorithm have been proposed in the literature for recurrent networks [24][22]. One popular technique, known as Back-propagation Through Time (BPTT), has been developed to train recurrent networks that are updated in discrete time steps. In fact, the original recurrent network is transformed to a virtual network in discrete time by replication of every processing unit in time as shown in Figure 2.6. The connections are established between a unit at time t to another at time $t+1$. The value of each unit at $t+1$ is a function of the state of the units at the previous times. Basically, this is done by running the recurrent network for different time steps and successively unrolling the network in time. This process results in an equivalent network with a number of layers equal to the product of the original number of layers and the number of time steps. Then, the back-propagation algorithm is applied to the virtual network and the result is used to update the weights of the original network. More specifically, let N be the recurrent network to be trained with a temporal series starting at instant t and finishing at instant $t+n$, and N^* the multi-layer network resulting from N. The relation between N and N^* can be summarized as:

- for each temporal step within the interval $[t, t+n]$, the network N^* has a layer composed by k units, where k is the number of layers contained in N
- in each layer of the network N^*, there is a copy of each neuron of the network N
- for each temporal step $l = 1, ..., n$ within the interval $[t, t+n]$, the connection from the unit i of the layer l, to the unit j of the layer $l+1$ of the network N^* is a copy of the connection from the unit i to the unit j of the network N

The standard back-propagation algorithm needs to train only the input/output pairs, which is different from the back-propagation through time method that needs to memorize the input and output vectors of the whole training data set before the learning phase. The buffer used to memorize the n input/output pairs before the back-propagation of the error through the network increases the computation time, which makes the algorithm harder to use for long training sequences. To avoid this problem, it is possible to memorize only a shorter sequence $n' < n$ of the input/output pairs as illustrated in Figure 2.7.

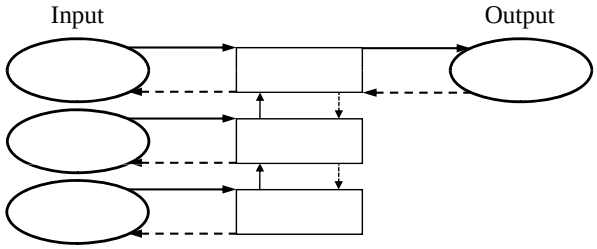


Figure 2.7: Shorter sequence $n' < n$ of the input/output pairs with $n' = 3$.

The appropriate dimension of n' depends mostly on the particular application. In general, n' should be set at least as the number of steps required so that an input can be propagated to the output. In many practical cases, n' can be set to 10.

Chapter 3

Pulse coupled neural networks

Part of this Chapter's contents is extracted from:

1. F. Pacifici and F. Del Frate, "Automatic change detection in very high resolution images with pulse-coupled neural networks", *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 1, pp. 58-62, January 2010

2. F. Pacifici, W. J. Emery, "Pulse Coupled Neural Networks for Automatic Urban Change Detection at Very High Spatial Resolution", in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, E. Bayro-Corrochanpo, J. Eklundh, Eds. Lecture Notes in Computer Science, Springer, vol. 5856, pp. 929-942, Springer, ISBN: 978-3-642-10267-7, 2009

Pulse Coupled Neural Networks (PCNNs) entered the field of image processing in the nineties, following a publication by Eckhorn *et al.* [25] of a new neuron model based on the mechanisms underlying the visual cortex of small mammals.

Interesting results have been already shown by several authors in the application of this model in image segmentation, classification and thinning [26][27], including the use of satellite data [28][29]. Hereafter, the main concepts underlying the behavior of PCNNs are briefly recalled. For a more comprehensive introduction to image processing using PCNNs refer to [30].

3.1 The pulse coupled model

A PCNN is a neural network algorithm that, when applied to image processing, yields a series of binary pulsed signals, each associated to one pixel or to a cluster of pixels. It belongs to the class of unsupervised artificial neural networks in the sense that it does not need to be trained. The network consists of nodes with spiking behavior interacting with each other within a pre-defined grid. The architecture of the network is rather simpler than most other neural network implementations: there are no multiple layers that pass information to each other. PCNNs only have one layer of neurons, which receives input directly from the original image, and form the resulting *pulse* image.

The PCNN neuron, shown schematically in Figure 3.1, has three compartments. The *feeding* compartment receives both an external and a local stimulus, whereas the *linking* compartment only receives the local stimulus. The third compartment is represented by an active threshold value. When the internal activity becomes larger than the threshold the neuron fires and the threshold sharply increases. Afterwards, it begins to decay until once again the internal activity becomes larger. Such a process gives rise to the pulsing nature of the PCNN.

More formally, the system can be defined by the following expressions:

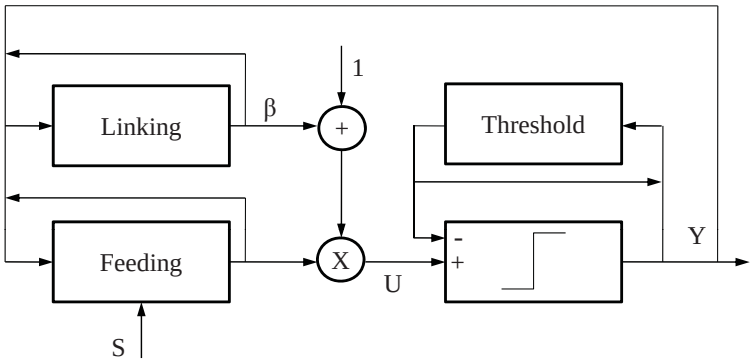


Figure 3.1: Schematic representation of a PCNN neuron.

$$F_{ij}[n] = e^{-\alpha_F} \cdot F_{ij}[n-1] + S_{ij} + V_F \sum_{kl} M_{ijkl} Y_{kl}[n-1] \tag{3.1.1}$$

$$L_{ij}[n] = e^{-\alpha_L} \cdot L_{ij}[n-1] + V_L \sum_{kl} W_{ijkl} Y_{kl}[n-1] \tag{3.1.2}$$

where S_{ij} is the input to the neuron (ij) belonging to a 2D grid of neurons, F_{ij} is the value of the feeding compartment and L_{ij} is the corresponding value of the linking compartment. Each of these neurons communicates with neighboring neurons (kl) through the weights given by M and W respectively. M and W traditionally follow very symmetric patterns and most of the weights are zero. Y indicates the output of a neuron from a previous iteration $[n-1]$. All compartments have a memory of the previous state, which decays in time by the exponent term. The constant V_F and V_L are normalizing constants. The state of the feeding and linking compartments are combined to create the internal state of the neuron, U. The combination is controlled by the linking strength, β. The internal activity is given by:

$$U_{ij}[n] = F_{ij}[n] \left\{1 + \beta\, L_{ij}[n]\right\} \tag{3.1.3}$$

The internal state of the neuron is compared to a dynamic threshold, Θ, to produce the output, Y, by:

$$Y_{ij}[n] = \begin{cases} 1 & \text{if } U_{ij}[n] > \Theta_{ij}[n] \\ 0 & \text{otherwise} \end{cases} \tag{3.1.4}$$

The threshold compartment is described as:

$$\Theta_{ij}[n] = e^{-\alpha_\Theta} \cdot \Theta_{ij}[n-1] + V_\Theta Y_{ij}[n] \tag{3.1.5}$$

where V_Θ is a large constant generally more than one order of magnitude greater than the average value of U.

The algorithm consists of iteratively computing Equation 3.1.1 through Equation 3.1.5 until the user decides to stop. Each neuron that has any stimulus will fire at the initial iteration, creating a large threshold value. Then, only after several iterations the threshold will be small enough to allow the neuron to fire again. This process is the beginning of the *auto-waves* nature of PCNNs. Basically, when a neuron (or group of neurons) fires, an auto-wave emanates from that perimeter of the group. auto-waves are defined as normal propagating waves that do not reflect or refract. In other words, when two waves collide they do not pass through each other.

For each unit, i.e. for each pixel of an image, PCNNs provide an output value. The *time signal*, computed by:

$$G[n] = \frac{1}{N} \sum_{ij} Y_{ij}[n] \tag{3.1.6}$$

is generally exploited to convert the pulse images to a single vector of information. In this way, it is possible to have a *global* measure of the number of pixels that fire at epoch $[n]$ in a sub-image containing N pixels. The signal associated to $G[n]$ was shown to have properties of invariance to changes in rotation, scale, shift, or skew of an object within the scene [30].

PCNNs have several parameters that may be tuned to considerably modify the outputs. The linking strength, β, together with the two weight matrices, scales the feeding and linking inputs, while the three potentials, V, scale the internal signals. The time constants and the offset parameter of the firing threshold can be exploited to modify the conversions between pulses and magnitudes. The dimension of the convolution kernel directly affects the speed of the auto-waves. The dimension of the kernel allows the neurons to communicate with neurons farther away and thus allows the auto-wave to advance farther in each iteration. The pulse behavior of a single neuron is directly affected by the values of α_Θ and V_Θ. The first affects the decay of the threshold value, while the latter affects the height of the threshold increase after the neuron pulses [30]. The auto-wave created by PCNN is greatly affected by V_F. Setting V_F to 0 prevents the auto-wave from entering any region in which the stimulus is also 0. There is a range of V_F values that allows the auto-wave to travel but only for a limited distance.

The network also exhibits some synchronizing behavior. In the early iterations, segments tend to pulse together. However, as the iterations progress, the segments tend to *desynchronize*. The synchronization occurs by a *pulse capture*. This occurs when one neuron is close to pulsing ($U < \Theta$) and its neighbor fires. The input from the neighbor provides an additional input to U allowing the neuron to fire prematurely. Therefore, the two neurons synchronize due to their linking communications. The loss of synchronization occurs in more complex images due to residual signals. As the network progresses, the neurons begin to receive information indirectly from other non-neighboring neurons. This alters their behavior and the synchronization begins to fail.

There are also architectural changes that can alter the PCNN behaviour. One such alteration is the *quantized linking* where the linking values are either 1 or 0 depending on a local condition (γ). In this system the, linking compartment is computed by:

$$L_{ij}[n] = \begin{cases} 1 & \text{if } \sum_{kl} W_{ijkl} Y_{kl} > \gamma \\ 0 & \text{otherwise} \end{cases} \tag{3.1.7}$$

The distinctive behavior of quantized links is the reduction of the auto-waves. In the original implementation of PCCNs, auto-waves decay on the edges. In other words a wave front tends to lose its shape near its outer boundaries. Quantized linking was observed in [30] to maintain the wave-fronts shape.

Another variation of the original implementation of PCCNs is the *fast linking*. This allows the linking waves to travel faster than the feeding waves. It basically iterates the linking and internal activity equations until the system stabilizes. A detailed description of can be found in [30].

3.2 Application of PCNN to toy examples

PCNNs were applied to two toy examples to illustrate the internal activity of the model. Figure 3.2 shows the first 49 iterations of the algorithm with two images of 150 by 150 pixels. The original inputs ($n = 1$) contain

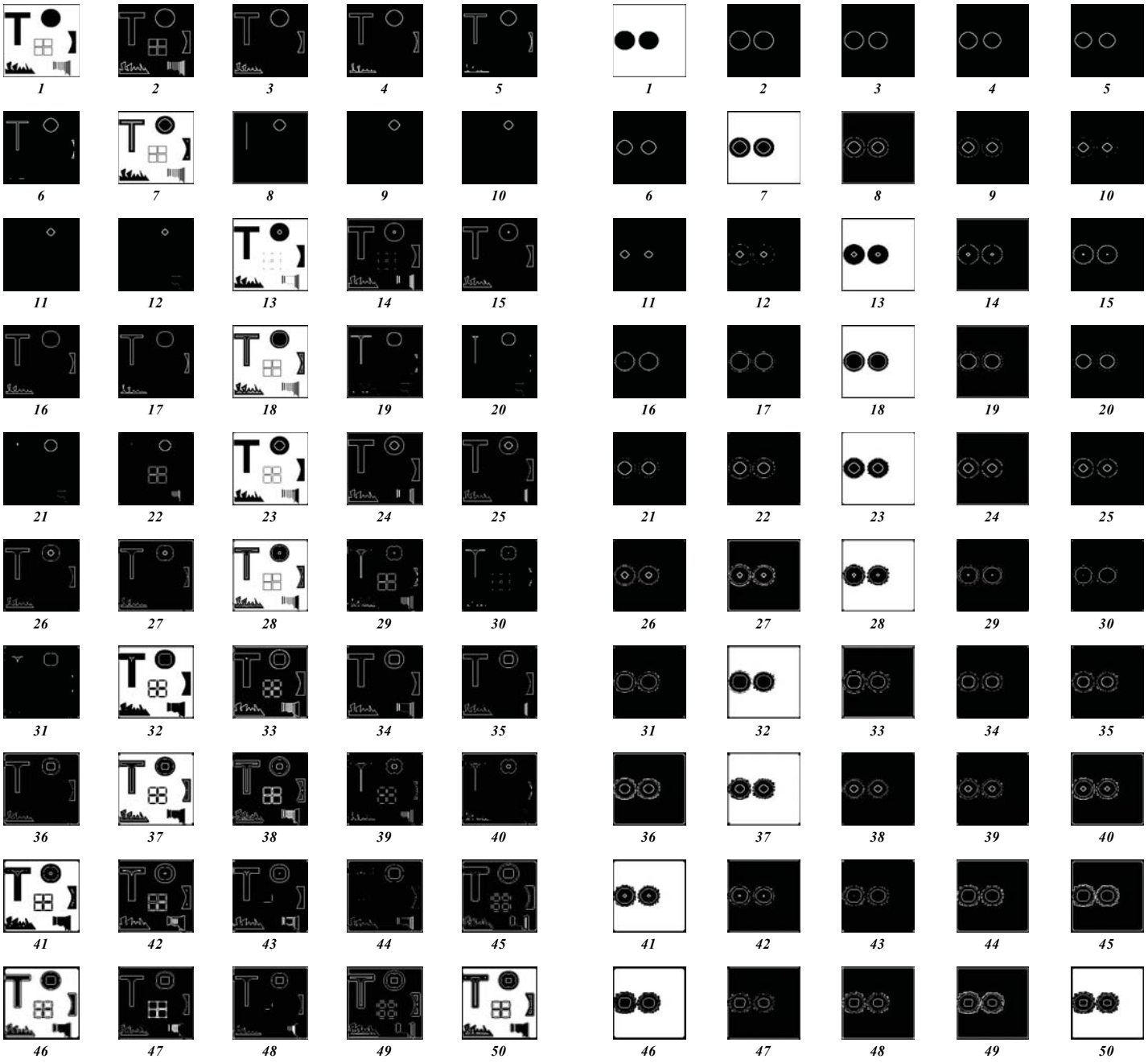
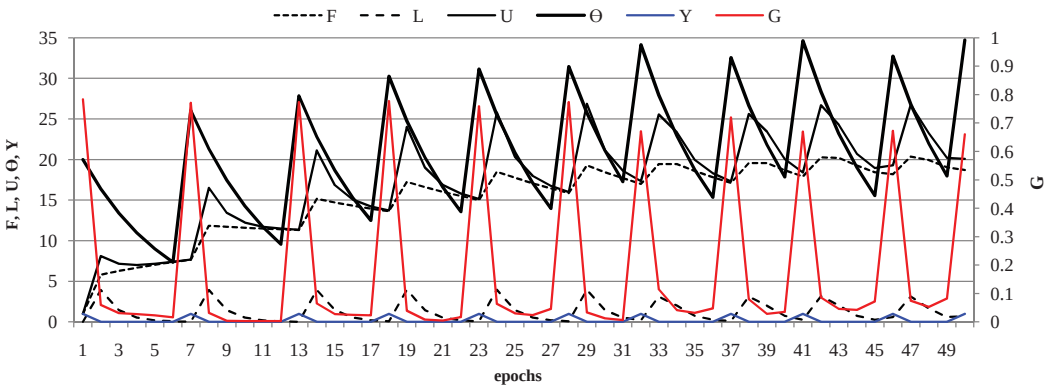


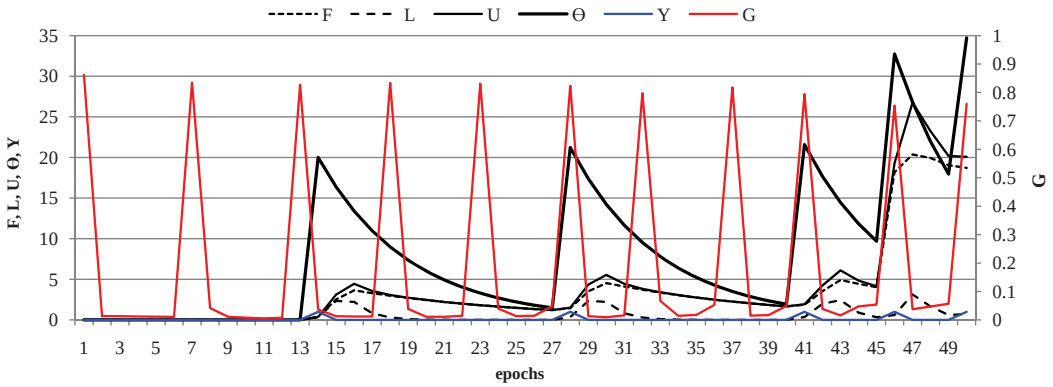
Figure 3.2: Iterations of the PCNN algorithm applied to toy examples of 150 by 150 pixels. As the iterations progress ($n > 1$), the auto-waves emanate from the original pulse regions and the shapes of the objects evolve through the epochs due to the pulsing nature of PCNNs.

objects with various shapes, including "T", squares and circles. As the iterations progress ($n > 1$), the auto-waves emanate from the original pulse regions and the shapes of the objects evolve through the epochs due to the pulsing nature of PCNNs.

Figure 3.3a and Figure 3.3b illustrate the progression of the states of a single neuron and trend of $G[n]$ (see Equations 3.1.1–3.1.6) for the toy examples in Figure 3.2a and Figure 3.2b, respectively. As shown, the internal activity U rises until it becomes larger than the threshold Θ and the neuron fires ($Y = 1$). Then, the threshold



(a)



(b)

Figure 3.3: Progression of the states of a single neuron (in this example, the central pixel) and trend of G for the toy example in Figure 3.2a and Figure 3.2b, respectively.

significantly increases and it takes several iterations before the threshold decays enough to allow the neuron to fire again. Moreover, F, L, U and Θ maintain values within comparable ranges. It is important to note that the threshold Θ reflects the pulsing nature of the *single* neuron, while $G[n]$ gives a *global* measure of the number of pixels that fired at epoch $[n]$.

Chapter 4

Feature selection

Part of this Chapter's contents is extracted from:

1. F. Pacifici, M. Chini and W. J. Emery, "A neural network approach using multi-scale textural metrics from very high resolution panchromatic imagery for urban land-use classification", *Remote Sensing of Environment*, vol. 113, no. 6, pp. 1276-1292, June 2009

2. D. Tuia, F. Ratle, F. Pacifici, M. F. Kanevski and W. J. Emery "Active Learning Methods for Remote Sensing Image Classification", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 7, pp. 2218-2232, July 2009

A large number of features can be used in remote sensing data analysis. Generally, not all of these features are equally informative. Some of them may be redundant, noisy, meaningless, correlated or irrelevant for the specific task. Therefore, if on one hand more information may be helpful, on the other hand, the increasing number of input features may introduce additional complexity related to the increase of computational time and the *curse of dimensionality* [15], overwhelming the expected increase in class separability associated with the inclusion of additional features. Intuitively, any classification approach should include only those features that make significant contributions, which should result in better performance [31]. As a result, it is necessary to reduce the number features, as for example, using principal component analysis to diminish the number of inputs [32].

Under this scenario, *feature extraction* and *feature selection* methods aim at selecting a set of features which is relevant for a given problem. The importance of these methods is the potential for speeding up the processes of both learning and classifying since the amount of data or processes is reduced.

In general, feature extraction results from the reduction of dimensionality of data by combination of input features. The aim of these methods is to extract new features while maximizing the discrimination between classes. Linear feature extraction methods have been studied in [33][34][35], where algorithms such as Decision Boundary Features Extraction [36] and Nonparametric Weighted Feature Extraction [37] have been shown to effectively reduce the size of the feature space. Despite the effectiveness of these methods, the extracted features are combinations of the original features and all of them are necessary to build the new feature set. On the contrary, feature selection (FS) is performed to select informative and relevant input features by analyzing the relevancy of each of the input features for a certain classification problem.

Feature selection algorithms can be divided in three classes: *filters*, *wrappers* and *embedded methods* [38]. Filters rank features based on feature relevance measures. Such measures can be computed either using a similarity measure, such as correlation, or using a measure of distance between distributions (for instance Kullback-Leibler (KL) divergence between the joint and product distribution). Then, a search strategy such as forward or backward selection is applied in order to select the relevant subset of features. Filters may be considered as a preprocessing

step and are generally independent from the selection phase. Remote sensing applications of filters can be found in [39][40][41][42][43][44].

Wrappers utilize the predictor as a black box to score subsets of features according to their predicting power. In a wrapper selector, a set of predictors receives subsets of the complete feature set and gives feed-back on the quality of the prediction (represented, for instance, by accuracy or Kappa statistics [45][46]) using each subset.

Feature selection and learning phases interact in embedded methods where the feature selection is part of the learning algorithm and the feature selection is performed by using the structure of the classifying function itself. This is different from wrappers where, as pointed out, the learning algorithm is accessed as a black box and only the output is used to build the selection criterion.

Several specific methods have been proposed, either wrappers or embedded, depending on the degrees of interaction between the classifier and the feature selection criterion. A well known embedded backward selection algorithm is the Recursive Feature Elimination (RFE) that uses the changes in the decision function of the Support Vector Machines (SVMs) as criterion for the selection [47][48]. In [49], an embedded feature selection is proposed. The algorithm, based on boosting, finds an optimal weighting and eliminates the bands linked with small weights. In [50], a genetic algorithm is used to select an optimal subset of features for a successive SVM classification. An example of feature selection with multi-layer perceptron neural networks can be found in [51].

The feature selection approach using neural networks and the formulation to establish the importance (*contribution*) of each input feature are illustrated in Section 4.1, while the RFE algorithm is described in Section 4.2.

4.1 Neural network pruning

As discussed in Chapter 2, a neural network with a small architecture may not be able to capture the underlying data structure. In fact, if the network has an insufficient number of free parameters (weights), it underfits the data, leading to excessive biases in the outputs. When the number of neurons in the hidden layers increases, the network can learn more complex data patterns by locating more complex decision boundaries in feature space. However, a neural network with an architecture too large may fit the noise in the training data, leading to good performance on the training set but rather poor accuracy relative to the validation set, resulting in large variance in the predicted outputs and poor accuracy. Generally speaking, the larger the network, the lower its generalization capabilities, and the greater the time required for training networks [52].

Unfortunately, the optimal architecture is not known in advance for most real-world problems. Consequently, there are two common ways to find the desired network size: *growing* and *pruning* approaches. The first consists of starting with a small network and adding neurons until the optimization criteria are reached [53]. However, this approach may be sensitive to initial conditions and become trapped in local minima. The second approach consists of beginning with a large network and then removing connections or units that are identified as less relevant. This approach has the advantage that the network is less sensitive to initial conditions. Moreover by reducing network size, it improves its generalization capabilities when applied to new data [54].

Feature selection with neural networks can be seen as a special case of architecture pruning, where input units are pruned rather than hidden units or single weights. Many pruning procedures for removing input features have been proposed in the literature [55][56][51]. Although many different pruning methods have been proposed in the literature, the main ideas underlying most of them are similar. They all establish a reasonable relevance measure, namely saliency, for the specific element (unit or weight) so that the pruning process has the least effect on the

performance of the network. Pruning methods can be divided into three wide groups in terms of decision criteria for the removing of weights or nodes:

1. sensitivity-based
2. penalty-term approaches
3. others, which may include interactive pruning, bottleneck method, or pruning by genetic algorithms

In the first method, the sensitivity is estimated by the error function after the removal of a weight or unit. Then, the less significant element (weight or unit) is removed. An example of these methods is the Magnitude-base (MB) pruning. In penalty-term methods, one adds terms to the objective function that rewards the network for choosing efficient solutions. There is some overlap in these two groups since the objective function could include sensitivity terms [57]. The pruning problem has been also formulated in terms of solving a linear equation system [58]. Suzuki *et al.* [59] proposed a pruning technique on the basis of the influence of removing units in order to determine the structures of both the input and the hidden layers. Reed [60] has given detailed surveys of pruning methods.

4.1.1 Magnitude-base pruning

Among the neural network pruning techniques, a sensitivity-based method proved to be the most popular. Magnitude-based pruning is the simplest weight-pruning algorithm which is based on removing links with the smallest magnitude value. Thus the saliency of a link is simply the absolute value of its weight. Tarr [61] explained this concept considering that when a weight is updated, the learning algorithm moves the weight to a lower value based on the classification error. Thus, given that a particular feature is relevant to the problem solution, the weight would be moved in a constant direction until a solution with no error is reached. If the error term is consistent, the direction of the movement of the weight vector will also be consistent (a consistent error term is the result of all points in a local region of the decision space belonging to the same output class). If the error term is not consistent, which can be the case of a single feature of the input vector, the movement of the weight attached to that node will also be inconsistent. In a similar fashion, if the feature does not contribute to a solution, the weight updates would be random. In other words, useful features would cause the weights to grow, while weights attached to non-salient features simply fluctuate around zero. Consequently, the magnitude of the weight vector serves as a reasonable saliency metric.

Although MB is very simple, it rarely yields worse results than the more sophisticated algorithms, such as Optimal Brain Damage, Optimal Brain Surgeon or Skeletonization [22]. Kavzoglu and Mather [54] investigated different pruning techniques using synthetic aperture radar and optical data for land-cover mapping. They found that the MB pruning technique generally yielded better results despite the simplicity of the algorithm. Moreover, their results show that pruning not only reduces the size of neural networks, but also increases overall network performance.

The network pruning technique provides a reduced set of features and at the same time optimizes the network topology. However, the resultant input space may have more than a reasonable number of input features. A trade-off between classification accuracy and computational time should be determined. The so-called *extended pruning* technique [51] is the process of eliminating iteratively (by successive pruning phases) the least contributing inputs until the training error reaches a specified limit. This process identifies a sub-optimal feature set (sub-optimal

from the classification accuracy point of view). In fact, this further input reduction results in a decrease in the classification accuracy. Thus, after the optimization (from the classification accuracy point of view) of the input features and the network topology by pruning, the extended pruning technique can be used to identify a reduced input set containing only the most contributing features.

4.1.2 Computing the feature saliency

Considering how weights in a neural network are updated, they can be used to calculate the feature saliency of the MB approach. Once the network has been pruned (or extended pruned), a general method for determining the relative significance of the remaining input features has been suggested by Tarr [61]. Input units whose weighted connections have a large absolute value are considered to be the most important. He proposes the following saliency metric to define the relevance for every weight between the input i and hidden unit j of the network:

$$S_i = \sum_{j}^{N_h} w_{ij}^2 \tag{4.1.1}$$

which is simply the sum of the squared weights between the input layer and the first hidden layer. This formulation may not be completely representative when dealing with pruned networks, since several connections between the input and the first hidden layer may be missing. Yacoub and Bennani [62] exploited both weight values and network structure of a multi-layer perceptron network in the case of one hidden layer. They derived the following criterion:

$$S_i = \sum_{j \in H} \left(\frac{|w_{ji}|}{\sum_{i' \in I} |w_{ji'}|} \cdot \sum_{k \in O} \frac{|w_{kj}|}{\sum_{j' \in H} |w_{kj'}|} \right) \tag{4.1.2}$$

where I, H, O denote the input, hidden and output layer, respectively. For the two hidden layers case, the importance of variable i for output j is the sum of the absolute values of the weight products over all paths from unit i to unit j, and it is given by:

$$S_{ij} = \sum_{k \in H1} \left[\frac{|w_{ik}|}{\sum_{k' \in H1} |w_{ik'}|} \cdot \sum_{x \in H2} \left(\frac{|w_{kx}||w_{xj}|}{\sum_{x' \in H2} |w_{kx'}| \cdot \sum_{x' \in H2} |w_{x'j}|} \right) \right] \tag{4.1.3}$$

where $H1$ and $H2$ denote the first and the second hidden layers, respectively. Then, the importance of variable i is defined as the sum of these values over all the outputs classes N_{cl}.

$$S_i = \sum_{j=1}^{N_d} S_{ij} \tag{4.1.4}$$

4.2 Recursive feature elimination

Recursive feature elimination is an embedded feature selector whose selection criterion is based upon the analysis of the classification function of the predictor (SVMs are here exploited as classification scheme as comparison to the neural-based pruning). Two versions of the algorithm have been proposed in the literature so far. The first one takes into account linearly separable data, while the other is suitable for linearly non-separable data [47][48]. Following, both versions of the RFE algorithm are briefly summarized.

4.2.1 RFE for linearly separable data

In a linearly separable case and for a binary problem, the SVM decision function for a Q-dimensional input vector $\mathbf{x} \in \mathbb{R}^Q$ is given by:

$$D(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b \tag{4.2.1}$$

with:

$$\mathbf{w} = \sum_k \alpha_k y_k \mathbf{x}_k \tag{4.2.2}$$

The weight vector $\mathbf{w}$ is a linear combination of the training points, $\boldsymbol{\alpha}$ are the support vector coefficients and $\mathbf{y}$ the labels. In this case and for a set of variables Q, the width of the margin is $2/||w||^2 = 2/\sqrt{\sum_{i=1}^{Q} w_i^2}$. By computing the weight vector $c_i = (w_i)^2$ for each feature in the features set Q, it is possible to rank the features by their contribution to the total decision function and evaluate the selection criterion f as in:

$$f = \arg\min_{i \in Q} |c_i| \tag{4.2.3}$$

In this way, the feature that provide the smallest change in the objective function is selected and removed.

4.2.2 RFE for nonlinearly separable data

In a non-linearly separable case, it is not possible to compute the weight vectors c_i. This can be explained as follows. The decision function is defined in a feature space, which is the mapping of the original patterns $\mathbf{x}$ in a higher dimensional space. The mapping ϕ is not computed explicitly, but only expressed in terms of dot products in the feature space by the kernel function $K(\mathbf{x}_k, \mathbf{x}_l)$. This way, only distances between the mapped patterns $\phi(\mathbf{x})$ are used and their position in the feature space is not computed explicitly. Therefore, it is not possible to apply Equation 4.2.2 directly. In order to take into account linearly inseparable datasets, it has been proposed in [47] to use the quantity $W^2(\boldsymbol{\alpha})$ expressed as:

$$W^2(\boldsymbol{\alpha}) = ||\mathbf{w}||^2 = \sum_{k,l} \alpha_k \alpha_l y_k y_l K(\mathbf{x}_k, \mathbf{x}_l) \tag{4.2.4}$$

Such a quantity is a measure of the predictive ability of the model and is inversely proportional to the SVM margin. Using this property and assuming that the $\boldsymbol{\alpha}$ coefficients remain unchanged by removing the less informative feature, it is possible to compute $W^2_{(-i)}(\boldsymbol{\alpha})$ for all the feature subsets counting for all the features minus the considered feature i. This quantity is computed without re-training the model. Successively, the feature whose removal minimizes the change of the margin is removed:

$$f = \arg\min_{i \in Q} |W^2(\boldsymbol{\alpha}) - W^2_{(-i)}(\boldsymbol{\alpha})| \tag{4.2.5}$$

For the sake of simplicity, the RFE algorithms described above (Equations 4.2.3-4.2.5) were discussed for binary problems only. To take into account multi-class data, the quantities $W^2(\boldsymbol{\alpha})$ and $W^2_{(-i)}(\boldsymbol{\alpha})$ are computed separately for each class. Then, as proposed in [48], the selection criterion of Equation 4.2.5 is evaluated for the sum over the classes of the $W^2(\boldsymbol{\alpha})$'s:

$$f = \arg\min_{i \in Q} \sum_{cl} |W^2_{cl}(\boldsymbol{\alpha}_{cl}) - W^2_{cl,(-i)}(\boldsymbol{\alpha}_{cl})| \tag{4.2.6}$$

RFE runs iteratively, removing a single feature at each epoch. Therefore, a prior knowledge about the number of features to select is required.

Part II

Classification of urban areas

Chapter 5

Introduction to the urban classification problem

Human activity truly dominates the Earth's ecosystems with structural modifications. The rapid population growth over recent decades and the concentration of this population in and around urban areas have significantly impacted the environment. Although urban areas represent a small fraction of the land surface, they affect large areas due to the magnitude of the associated energy, food, water and raw material demands. Urban areas are undergoing dynamic changes and, as a consequence, are facing new spatial and organizational challenges as they seek to manage local urban development within a global community. Sub-urbanization refers to a highly dynamic process where rural areas, both close to, but also distant from, city centers become enveloped by, or transformed into, extended metropolitan regions. These areas are a key interface between urban and rural areas due to the provision of essential services [63]. Reliable information in populated areas is essential for urban planning and strategic decision making, such as civil protection departments in cases of emergency. Lack of information contributes to problems such as ineffective urban development programs and activities, unplanned investment projects, poor functioning land markets, and disregard of the environmental impact of developments [64].

Satellite remote sensing is increasingly being used as a timely and cost-effective source of information in a wide number of applications. However, mapping human settlements represents one of the most challenging areas for the remote sensing community due to its high spatial and spectral diversity. From the physical composition point of view, several different materials can be used for the same built-up element (for example, building roofs can be made of clay tiles, metal, grass, concrete, plastic, or stones). On the other hand, the same material can be used for different built-up elements (for example, concrete can be found in paved roads or building roofs). Moreover, urban areas are often made of materials present in the surrounding region, making them not distinguishable from the natural or agricultural areas (examples can be unpaved roads and bare soil, clay tiles and bare soil, or parks and vegetated open spaces) [1].

During the last two decades, significant progress has been made in developing and launching satellites with instruments, in both the optical/infrared and microwave regions of the spectra, well suited for Earth observation with an increasingly finer spatial, spectral and temporal resolution [65][66][67][68]. Fine spatial optical and synthetic aperture radar (SAR) sensors, with metric or sub-metric resolution, allow the detection of small-scale objects, such as elements of residential housing, commercial buildings, transportation systems and utilities. Multi-spectral and hyper-spectral remote sensing systems provide additional discriminative features for classes that are spectrally similar, due to their higher spectral resolution. The temporal component, integrated with the spectral and spatial dimensions, provides essential information on vegetation dynamics. Moreover, the delineation of temporal

homogeneous patches reduces the effect of local spatial heterogeneity that often masks larger spatial patterns.

The current offering of imagery does not match the customer real need, which is for *information*. Users in all domains require information or information-related services that are focused, concise, reliable, low-cost, timely, and which are provided in forms and formats compatible with the user own activities. However, the information extraction process is generally too complex, too expensive and too dependent on user conjecture to be applied systematically over an adequate number of scenes. Therefore, there is the need to develop automatic or semi-automatic techniques.

In the following, the diverse multi-spatial, multi-temporal, and multi/hyper-spectral aspects for the classification of urban areas are discussed in detail in Chapter 6, Chapter 7, and Chapter 8, respectively. The problem of data fusion between sensors is then addressed in Chapter 9. This part of the thesis is concluded with the concepts of image information mining and active learning, outlined in Chapter 10 and 11.

Chapter 6

Exploiting the spatial information

Part of this Chapter's contents is extracted from:

1. F. Pacifici, M. Chini and W. J. Emery, "A neural network approach using multi-scale textural metrics from very high resolution panchromatic imagery for urban land-use classification", *Remote Sensing of Environment*, vol. 113, no. 6, pp. 1276-1292, June 2009

2. D. Tuia, F. Pacifici, M. F. Kanevski, W. J. Emery, "Classification of very high spatial resolution imagery using mathematical morphology and support vector machines", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 11, pp. 3866-3879, November 2009

The successful launches of new optical and SAR systems, such as WorldView-1, WorldView-2, TerraSAR-X and CosmoSkyMed, have made major contributions towards the advancement of the remote sensing industry by providing sensors with higher spatial resolution, more frequent revisits and greater imaging flexibility with respect to the precursors, such as QuickBird, IKONOS, Satellite Pour l'Observation de la Terre (SPOT), Landsat Enhanced Thematic Mapper (ETM) or RADARSAT, European Remote Sensing satellites (ERS-1/2) and Environmental Satellite (ENVISAT), just to mention a few of them. These new systems have a potential for more detailed and accurate mapping of the urban environment with details of sub-meter or meter ground resolution. At the same time, the higher spatial resolution presents additional problems for the classification of the urban environment, especially when single-band panchromatic or SAR data is exploited. Urban areas are composed of a wide range of elements, such as concrete, asphalt, metal, plastic, glass, shingles, water, grass, shrubs, trees and soil, arranged by humans in complex ways to build housing, transportation systems, utilities, commercial buildings or recreational areas. Therefore, even a simple isolated building may appear as a complex structure with many architectural details surrounded by gardens, trees, roads, social and technical infrastructure and many temporary objects.

Panchromatic imagery has often been fused with multi-spectral data using pansharpening methods [69][70][71] in order to exploit the spectral information at higher spatial resolution. However, as stated by Gong *et al.* [72], improved spatial resolution data increases within-class variances, which results in high interclass spectral confusion. Further, several pixels may be representative of objects, which are not part of land-use classes defined. Cars can be used as a representative example, as they do not belong to any land-use class. However cars may be present in related classes, such as roads and parking lots. In addition, cars may create schematic patterns, for example in parking lots, and this effect may be measured and utilized for filtering them out. It is evident that this problem is intrinsically related to the spatial resolution of the sensor and it cannot be solved by increasing the number of spectral channels.

In SAR imagery, the complexity of the electromagnetic urban environment poses even more severe limitations to the analysis of very high spatial resolution imagery. In general, the backscattered signal of an isolated building

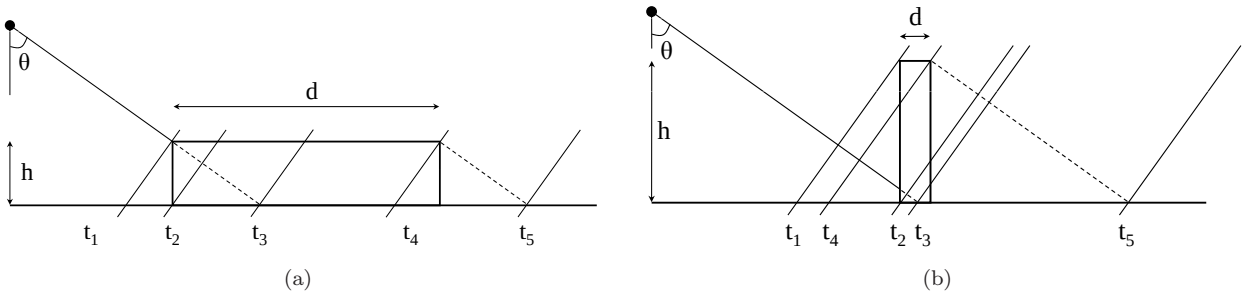


Figure 6.1: Geometry of the SAR acquisition on a simulated building.

can be determined by geometrical considerations concerning the times of arrival of the echoes to the sensor, given by:

$$t_i = \frac{2}{c} r_i, \qquad i = 1, \ldots, 5 \tag{6.0.1}$$

which correspond to the distances in time between the sensor and objects located on the equal-range curves. In Figure 6.1a, until time t_1, the building does not contribute to the return signal which arises from the rough terrain. Within the interval $t_1 - t_2$, the vertical walls and the roof of the building contribute to increase the backscattered field, giving rise to layover effect. Double and triple reflections are expected in t_2 and within the interval $t_2 - t_3$, respectively. The roof return characterizes the backscattered field of the interval $t_2 - t_4$, while shadowing is expected in the interval $t_4 - t_5$. Finally, after t_5, the radar return corresponds again to the field backscattered from the terrain. The presented geometrical scenario may vary drastically depending on a different acquisition angle or on the heights, widths, or distances of the buildings. For example, as shown in Figure 6.1b, when $t_4 < t_2$, the roof contribution is located before the double reflection.

Building 14 of the European Space Research Institute (ESRIN) located in Frascati (Italy) is considered here to analyze in detail such a complex electromagnetic environment in a relatively easy scenario. The reference image was acquired on October 5, 2005 by the L-band E-SAR of the German aerospace agency (Deutsches Zentrum für Luft und Raumfahrt - DLR) in a fully polarimetric mode with a spatial resolution of about 2m. Figure 6.2a shows the polarimetric color composite image of the test site. The structure is analyzed by means of two perpendicular cuts, orthogonal (58 pixels) and parallel (17 pixels) to the flight track, as shown by the corresponding optical views in Figure 6.2b and Figure 6.2c. Considering the orthogonal cut (left column of Figure 6.3), a first high-level signal (pixels 55-57) in correspondence of the parking lots. For pixels 53-54, the asphalted surface scattering produces a relatively low signal before the high return caused by the east side of the building. Pixels 48-52 are the brightest in the image due to the double bounce reflection mechanism: the typical foreshortening distortion is clearly visible. Pixels within the interval 18-52 are characterized by a rather constant return from the roof. At the end of the roof (pixels 9-18) the lower backscattering corresponds to the shadow of the building and to the asphalt surface. The remaining pixels can be associated with the adjacent building with the same scattering mechanism of pixels 55-57. Considering the parallel cut (right column of Figure 6.3), the most interesting behavior is the high level signal of the north and south walls of the building (pixels 3-5 and 11-14): the presence of vertical structures creates a double bounce, thus enhancing the radar return.

With this simple example, it appears clear that very high spatial resolution SAR imagery is strongly affected by the complexity and the variety of scattering mechanisms, even for single isolated buildings. This behavior is dramatically different compared to previous decametric SAR systems where the variety of electromagnetic and

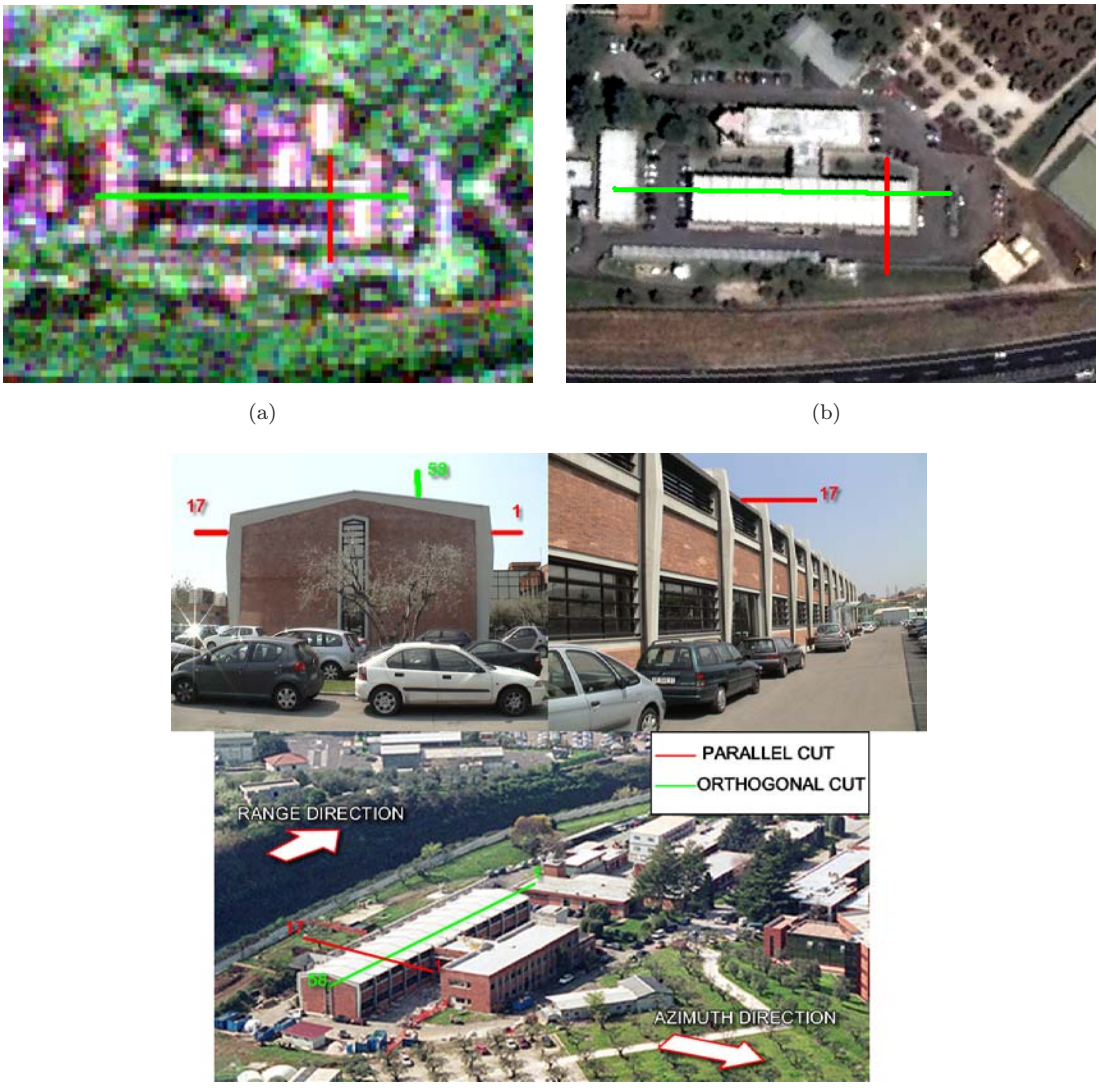


Figure 6.2: Building 14 of the ESRIN site imaged in (a) polarimetric color composite (R: HH, G:HV, B:VV), (b) and (c) optical views of the orthogonal and parallel cuts.

geometric effects tend to be smeared by the coarser spatial resolution. Therefore, very high spatial resolution single band systems, either optical or SAR, are generally not suitable for land-use mapping of complex urban scenes. Hence, it is necessary to extract additional information in order to recognize objects within the scenes.

Spatial analysis plays an increasingly important role in satellite image processing and interpretation. In the past, adding spatial information, such as textural or morphological features, for the classification of satellite images has been shown to overcome the lack of multi-band information [39]. Texture is the term used to characterize the tonal or gray level variations in an image. A different way to integrate contextual information is to extract the shape of single objects using mathematical morphology [73][74].

In this chapter, the extraction and application of textural and morphological features to very high spatial resolution optical and SAR imagery are illustrated in Section 6.1 and Section 6.2, respectively.

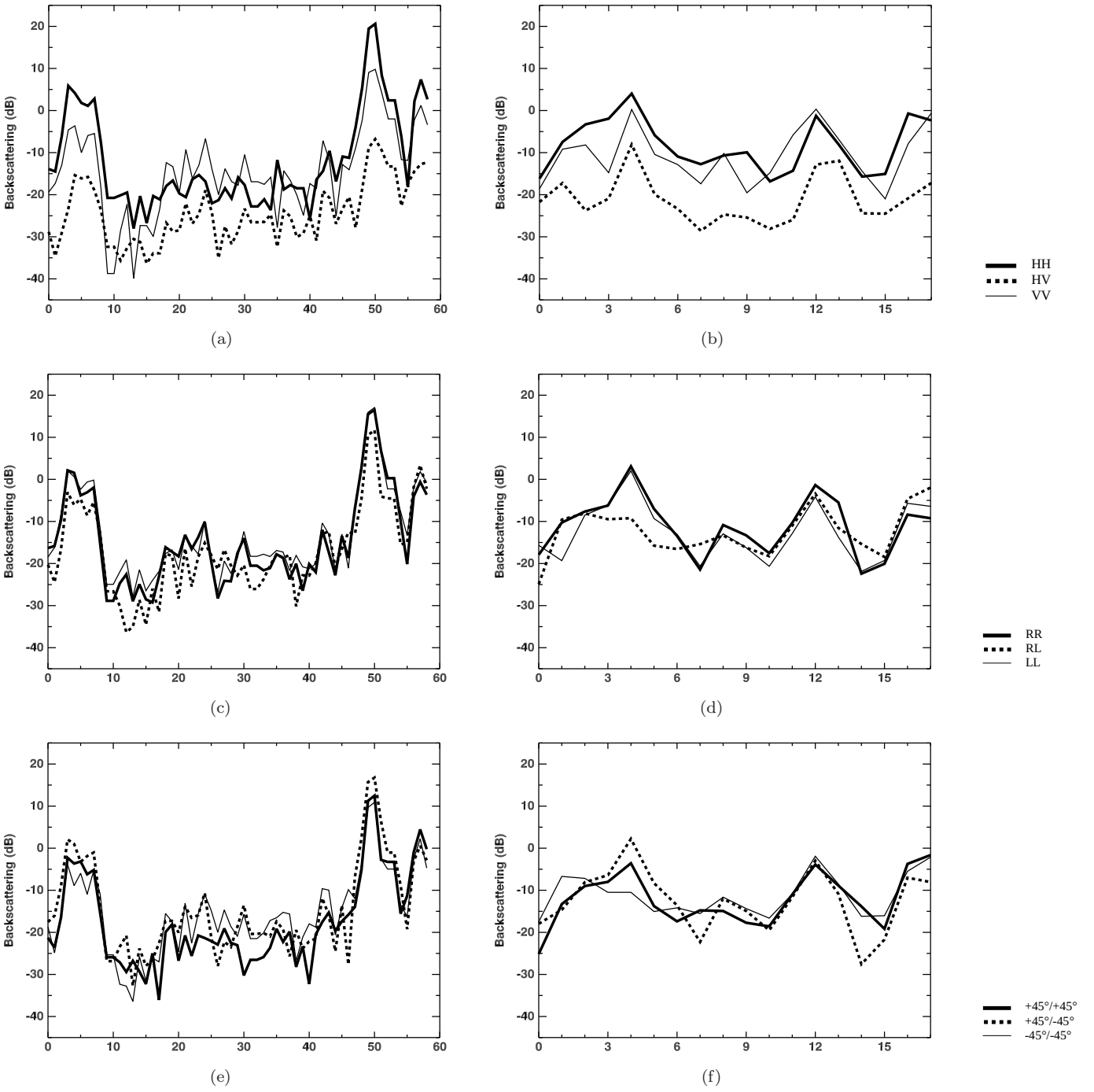


Figure 6.3: Orthogonal (left column) and parallel (right column) cuts of the building 14 of the ESRIN site: (a) and (b) represent the H/V polarizations, (c) and (d) the circular R/L polarizations, and (e) and (f) the $\pm 45^{\circ}$ linear polarizations.

6.1 Textural analysis

Numerous texture features exist as attested by the numerous papers appeared in literature in the past years. Tuceryan and Jain [75] identified four major textural categories: *statistical*, such as those based on the computation of the Gray-Level Co-occurence Matrix (GLCM) [76], *geometrical*, *model-based*, such as Markov Random Fields (MRF), and *signal processing*. Shanmugan *et al.* [77] pointed out that textural features derived from GLCM are

Table 6.1: Characteristics of the data sets.

Location	Dimension (pixels)	Satellite	Date	Spatial res. (m)	View angle (°)	Sun elev. (°)
Las Vegas	755x722	QuickBird	May 10, 2002	0.6	12.8	65.9
Rome	1,188x973	QuickBird	July 19, 2004	0.6	23.0	63.8
Washington D. C.	1,463x1,395	WorldView-1	Dec. 18, 2007	0.5	27.8	24.9
San Francisco	917x889	WorldView-1	Nov. 26, 2007	0.5	19.6	29.6

the most useful for analyzing the contents of a variety of imagery in remote sensing, while, according to Treitz *et al.* [78], statistical texture measures are more appropriate than the geometrical ones in land-cover classification. Clausi *et al.* [79] demonstrate that the GLCM method has an improved discrimination ability relative to MRFs with decreasing window size. Six GLCM parameters are considered to be the most relevant [80] among the 14 ones that can be computed, some of which are strongly correlated with the other. In their investigation, Baraldi and Parmiggiani [81] concluded that Energy and Contrast are the most significant parameters to discriminate between different textural patterns.

Many other examples of the use of textural parameters have been proposed for the extraction of quantitative information of building density [82] or for the recognition of different urban patterns [83]. In most cases, texture increased the per-pixel classification accuracy, especially in urban areas where the images are more heterogeneous [84]. Chen *et al.* [85] stated that this increase in terms of classification accuracy is dependent on the geometrical resolution of the scene. In fact, the improvement is greater for higher spatial resolution images. Textural analysis is also widely accepted for land-cover classification with SAR data, as shown in Kurosu *et al.* [86], Kurvonen and Hallikainen [87], and Arzandeh and Wang [88]. In particular, Dell'Acqua and Gamba [89] shown how the coarse spatial resolution of ERS-1 and ERS-2 allowed the recognition of dense, residential, and suburban areas.

It is important to point out that the majority of the studies that appeared in the literature in the past years have dealt with decametric spatial resolution imagery. Consequently, the resulting classification maps are generally representative of different terrain patterns and not of single objects within the image, such as buildings or trees [90].

In the following, the textural characteristics of very high spatial resolution panchromatic imagery are systematically analyzed to classify the land-use of different urban environments. To account for the spatial setting of cities, textural parameters were computed over five different window sizes, three different directions and two different cell shifts for a total of 191 input features. Neural network pruning and saliency measurements made it possible to determine the most important textural features for sub-metric spatial resolution optical imagery of urban scenes. The data sets are described in Section 6.1.1, while Section 6.1.2 deals with methodology, introducing the multi-scale textural analysis. Experimental results and the analysis of the textural feature contributions are discussed in Section 6.1.3. Final conclusions follow in Section 6.1.4.

6.1.1 Data sets

The data sets used considers four different cities with diverse architectural urban structures: Las Vegas (U. S. A.), Rome (Italy), Washington D. C. (U. S. A.) and San Francisco (U. S. A.). The first two scenes were acquired by QuickBird in 2002 and 2004, respectively, and the other two by WorldView-1 in 2007. Details of the images are reported in Table 6.1.

6.1.1.1 Description of the scenes

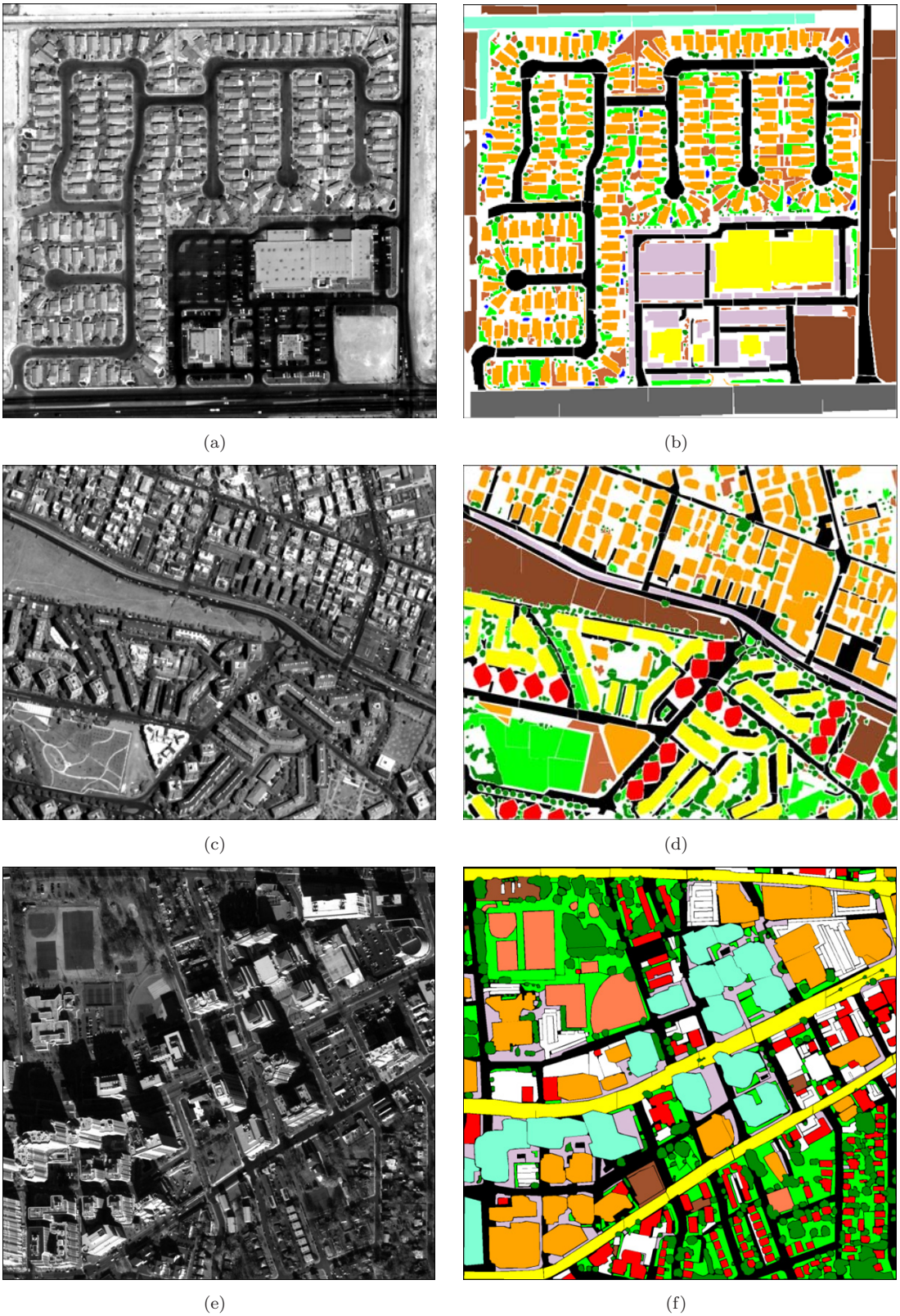
The Las Vegas scene, shown in Figure 6.4a, contains regular crisscrossed roads and examples of structures with similar heights (about one or two stories) but different dimensions, from small residential houses to large commercial buildings. This first scene was chosen for two reasons. First, its simplicity and regularity allowed an easier analysis and interpretation of the textural features. Second, it represents well a common American sub-urban landscape, including small houses and large roads, which is different from the European style of old cities built with more complex structures. To take into account this last situation, a second area of the sub-urban of Rome (see Figure 6.4c) was used. This scene shows a more elaborate urban lattice with buildings having a variety in heights (from four stories to twelve), dimensions and shapes including apartment blocks and towers. In particular, this area has two completely different urban architectures separated by a railway. The area located in the upper right of the scene was built during the 60s: buildings are very close to each other and have a maximum of five stories, while roads are narrow and usually show traffic jams due to the presence of cars and buses. The other side of the railway was developed during the 80s and 90s: buildings have a variety of architectures, from apartment blocks (eight stories) to towers (twelve stories), while roads are wider than those on the other side of the railroad tracks. The Washington D. C. scene, shown in Figure 6.4e, contains elements that characterize the other two, but imaged with a higher spatial resolution (0.5 m). Buildings have different heights, dimensions and shapes, varying from small residential houses to large structures with multiple stories (more than twenty), while asphalt surfaces include roads with different widths (e.g. residential and highways) and parking lots. The image of San Francisco presents regular structures, such as a highway, residential roads, two different type of buildings, commercial/industrial and residential, some sparse trees and vegetated areas. This scene was used for validation purposes only and it will be described better later more in detail.

6.1.1.2 Classes, training and validation set definition

Several different surfaces of interest were identified many of which are particular to the specific scene. For the Las Vegas case, whose ground reference is shown in Figure 6.4b, one goal was to distinguish the different uses of the asphalt surfaces, which included Roads (i.e. roads that link different residential houses), Highways (i.e. roads with more than two lanes) and Parking Lots. An unusual structure within the scene was a Drainage Channel located in the upper part of the image. This construction showed a shape similar to roads, but with higher brightness since it was built with concrete. A further discrimination was made between Residential Houses and Commercial Buildings due to the different size, and between Bare Soil (terrain with no use) and Soil (generally, backyards with no vegetation cover). Finally, more traditional classes, such as Trees, Short Vegetation and Water were added for a total of eleven classes of land-use. The areas of shadow were very limited in the scene due to the modest heights of buildings and relative sun elevation.

Due to the dual nature of the architecture of the Rome test case and the high off-nadir angle (about 23°), the selection of the classes was designed to investigate the potential of discriminating between structures with different heights, including Buildings (structures with a maximum of 5 stories), Apartment Blocks (rectangular structures with a maximum of 8 stories) and Towers (more than 8 stories). As for the previous case, other surfaces of interest were recognized, including Roads, Trees, Short Vegetation, Soil and the peculiar Railway for a total of nine classes. Different from the previous case, shadow occupies a larger portion of the image in this scene. The ground reference for this area is shown in Figure 6.4d.

The Washington D. C. scene has features in common with the previous two. In particular, it is possible to



(a) (b) (c) (d) (e) (f)

Figure 6.4: Original image (left) and ground reference (right) of (a) and (b) Las Vegas, (c) and (d) Rome, and (e) and f) Washington D. C., respectively. Color codes are in Table 6.2.

distinguish different uses of asphalt surfaces, such a Roads, Highways and Parking Lots, while buildings show a variety of dimensions and heights, from small Residential Houses in the bottom-right of the scene, to Tall Buildings with multiple stories in the center of the area. An interesting feature is the role of the class Trees. The image was acquired in December and most of the plants were without leaves. Therefore, these objects were not associated with the class Trees, but were assigned to a wider class Vegetation (including short vegetation and trees without leaves), and only trees with leaves were recognized as belonging to Trees. Finally, the classes Sport Facilities and Sidewalks were added for a total of 11 classes. The relative ground reference is shown in Figure 6.4f. It is important to highlight that, as for the Rome case, the image was acquired with a high off-nadir angle of about 28°, and with a sun elevation (about 25°), which caused large shadows.

The ground references of each scene were obtained by careful visual inspection of separate data sources, including aerial imagery, cadastral maps and in situ inspections (for the Rome scene only). An additional consideration regards objects within shadows that reflect little radiance because the incident illumination is occluded. Textural features have the potential to characterize these areas as if they were not within shadow. Therefore, these surfaces were assigned to one of the corresponding classes of interest described above.

When classifying imagery at sub-meter spatial resolution, many of the errors may occur in the boundaries between objects. On the other hand, it is also true that the nature of the objects is fuzzy and often it is not possible to correctly identify an edge. To investigate this effect, the first two ground references (Las Vegas and Rome) were created not including boundary areas, while the other (Washington D. C.) was generated minimizing the extensions of these regions.

In order to select training and validation samples, having both statistical significance and avoiding the correlated neighboring pixels, the stratified random sampling (SRS) method was adopted, ensuring that even small classes, such as water or trees, were adequately represented. In SRS, the population of N pixels is divided into k subpopulations of sampling units $N_1, N_2, \ldots, N_k$, which are termed *strata*. Therefore, the pixels in each of those classes have randomly sampled according to their extension in area, based on the ground reference.

The number of pixels used for training may influence the final classification accuracy. To investigate this, about 5% and 10% of the total pixels for the Las Vegas and Rome scenes (same spatial resolution) were used, respectively, and for comparison about 10% for the Washington case (higher spatial resolution). Details of the number of samples used as training (TR) and validation (VA) are reported in Table 6.2 for the Las Vegas, Rome and Washington D. C. areas, respectively. The Kappa coefficient was used to evaluate the accuracies of the classifiaction maps [46][91].

6.1.2 Multi-scale texture analysis

Two first-order and six second-order textural features derived from the GLCM have been exploited for the classification of scenes. First-order statistics can be computed from the histograms of pixel intensities in the image. These depend only on individual pixel values and not on the interaction or co-occurrence of neighboring pixel values. The first-order parameters used are Mean and Variance. The former is the average gray-level in the local window and the latter is the gray-level variance in the local window (high value when there is a large gray-level standard deviation in the local region).

The mathematical formulation of the six second-order textural features is given by:

$$Homogeneity = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \frac{p(i,j)}{1+(i-j)^2} \tag{6.1.1}$$

Table 6.2: Classes, training and validation samples, and color legend.

Location	Classes	TR	VA	Color
Las Vegas	Bare Soil	4,255	44,675	dark brown
	Commercial Buildings	1,822	19,126	yellow
	Drainage Channel	1,143	12,001	cyan
	Highways	2,836	29,774	gray
	Parking Lots	2,257	23,695	purple
	Residential Houses	7,007	73,563	orange
	Roads	6,098	64,023	black
	Short Vegetation	1,793	18,823	green
	Soil	1,472	15,437	brown
	Trees	1,043	10,945	dark green
	Water	118	1,236	blue
	Total	**29,844**	**313,298**	
Rome	Bare Soil	4,127	38,572	brown
	Apartment Blocks	20,472	44,672	yellow
	Buildings	27,188	77,034	orange
	Railway	2,606	6,727	purple
	Roads	35,531	69,002	black
	Soil	3,506	5,776	brown
	Tower	9,187	19,365	red
	Trees	13,632	38,624	dark green
	Short Vegetation	10,443	29,587	green
	Total	**126,692**	**329,359**	
Washington D. C.	Buildings	24,178	76,159	orange
	Highways	17,985	56,653	yellow
	Parking Lots	17,019	53,611	white
	Residential	14,195	44,714	red
	Roads	20,618	64,946	black
	Sidewalks	12,203	38,439	purple
	Soil	2,553	8,043	dark brown
	Sport Facilities	8,270	26,051	coral
	Tall Buildings	21,047	66,297	cyan
	Trees	18,535	58,386	dark green
	Vegetation	23,403	73,720	green
	Total	**180,006**	**567,019**	

$$Contrast = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} p(i,j) \cdot (i-j)^2 \tag{6.1.2}$$

$$Dissimilarity = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} p(i,j) \cdot |i-j| \tag{6.1.3}$$

$$Entropy = -\sum_{i=1}^{N-1} \sum_{j=1}^{N-1} p(i,j) \cdot \log(p(i,j)) \tag{6.1.4}$$

$$Second\ Moment = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} p(i-j)^2 \tag{6.1.5}$$

$$Correlation = \sum_{i=1}^{N-1} \sum_{j=1}^{N-1} \frac{(i \cdot j) \cdot p(i,j) - \mu_i \cdot \mu_j}{\sigma_i \cdot \sigma_j} \tag{6.1.6}$$

where σ and μ are mean and standard deviation; (i,j) are the gray-tones in the windows, which are also the coordinates of the co-occurrence matrix space; $p(i,j)$ are the normalized frequencies with which two neighboring resolution cells (separated by a fixed shift) occur on the image, one with gray tone i and the other with gray tone j; N is the dimension of the co-occurrence matrix, which has a gray value range of the original image.

Table 6.3: Input space resulting from panchromatic band, first- and second-order textural features.

Input Features	Texture	Cell Size (pixel)	Step (pixel)	Direction (°)	Number of Inputs
Panchromatic	-	-	-	-	1
Mean Variance	First-order	3×3 7×7 15×15 31×31 51×51	-	-	10
Homogeneity Contrast Dissimilarity Entropy Second Moment Correlation	Second-order		15 30	0 45 90	180
				Total Features	**191**

Homogeneity assumes higher values for smaller digital number differences in pair elements. Therefore, this parameter is more sensitive to the presence of near diagonal elements in the GLCM. Contrast takes into account the spatial frequency, which is the difference in amplitude between the highest and the lowest values of a contiguous set of pixels. This implies that a low contrast image is not necessarily characterized by a low variance value, but the low contrast image corresponds to low spatial frequencies. Unlike Contrast where the weights increase exponentially as one moves away from the diagonal, for Dissimilarity the weights increase linearly. This parameter measures how different the elements of the co-occurrence matrix are from each other and it is high when the local region has a high contrast. Entropy measures the disorder in an image. When the image is not uniform, many GLCM elements have very small values, which implies that Entropy is very large. Considering a window with completely random gray tones, the histogram for such a window is a constant function, i.e., all $p(i, j)$ are the same, and Entropy reaches its maximum. The Second Moment measures textural uniformity, i.e., pixel pairs repetitions. Indeed, when the image segment under consideration is homogeneous (only similar gray levels are present) or when it is texturally uniform (the shift vector always falls on the same (i, j) gray-level pair), a few elements of GLCM will be greater than 0 and close to 1, while many elements will be close to 0. Correlation is expressed by the correlation coefficient between two random variables i and j, and it is a measure of the linear-dependencies between values within the image. High Correlation values imply a linear relationship between the gray levels of pixel pairs. Thus, Correlation is uncorrelated with Energy and Entropy, i.e. to pixel pair repetitions [81].

Textural information is valuable for the discrimination of different classes that have similar spectral responses. At the same time, it is also necessary to exploit a multi-scale approach to better deal with objects having different spatial coverage in an area. For this purpose, the eight features defined previously were computed with five different window sizes 3×3, 7×7, 15×15, 31×31, 51×51 pixels (about 1.5-1.8 m, 3.5-4.2 m, 7.5-9.0 m, 15.5-18.6 m and 25.5-30.6 m, using WorldView-1 and QuickBird imagery), three different directions 0°, 45° and 90° and two different cell shift values of 15 and 30 pixels, for a total of 191 textural features, as reported in Table 6.3. The dimensions of the windows and the values of the shift have been based on the analysis of a previous work of Small [92] who estimated the characteristic length scale of 6357 sites in 14 urban areas around the World, showing that the majority of sites have characteristic length scales between about 8.0 m and 24.0 m (see Figure 6.5).

Figure 6.6 shows an example of Homogeneity computed over Las Vegas with three different directions (same step and window size). As expected, the direction highlights different structural patterns within the area, such as vertical or horizontal roads, and parking lots. In fact, the highway, which is a horizontal structure, has its highest value in the 15_0 and 30_0 directions, while parking area show a distinct behavior (with respect to the other classes) in the diagonal direction, due to its wide structure. Note, the notation "a×b_c_d" has the following

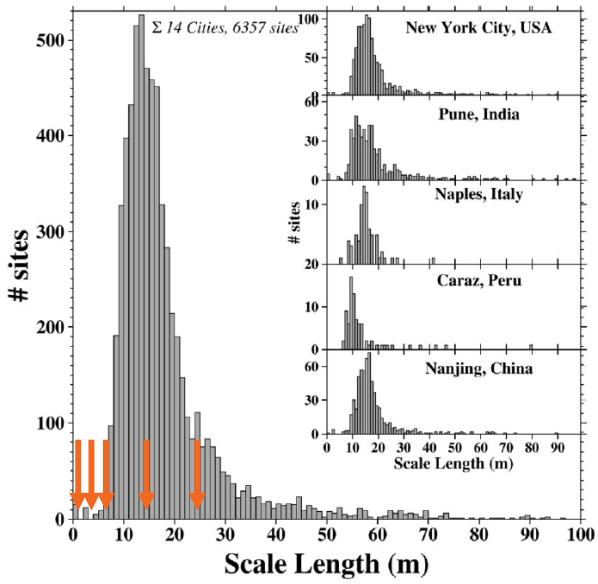


Figure 6.5: Characteristic length scale of 6,357 sites in 14 urban areas around the World. The orange arrows indicate the window sizes chose in this study. In particular, three window sizes delimit the bell-shaped curve, while the two smaller window sizes are exploited to extract information of small objects. Adapted from [92].

meaning: (a,b) are the dimensions of the window size, while (c,d) represent the Cartesian components of direction and shift. For example, the feature 3×3_15_0 is computed with a 3×3 window size, 15 pixels of shift and horizontal direction.

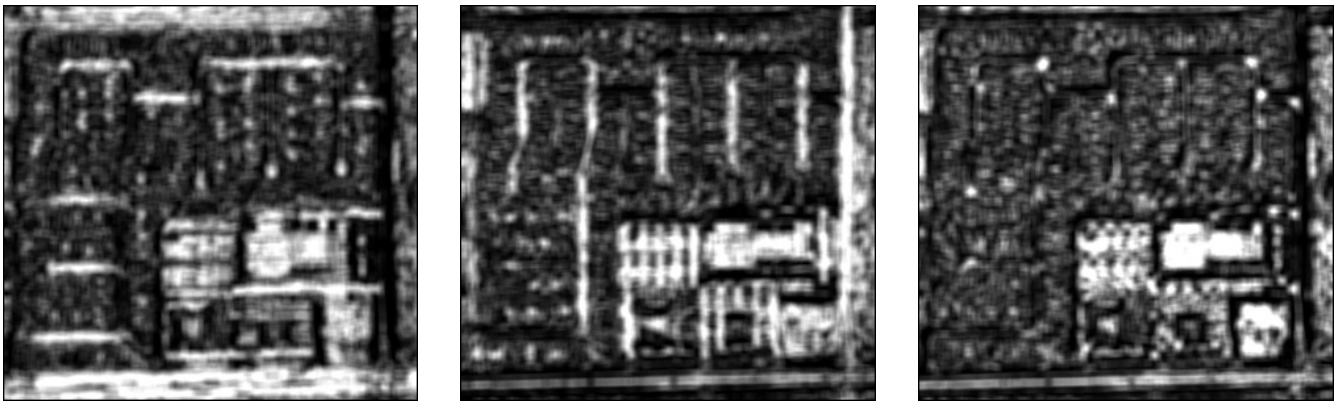
6.1.3 Results

The results obtained with the multi-scale textural approach to produce land-use maps are discussed here. Particularly, in Section 6.1.3.1 pruning is exploited to optimize the input feature space and the network topologies. Then, in Section 6.1.3.2, the analysis of most effective input features is discussed. A detailed analysis of the best 10 features is illustrated in Section 6.1.3.3. In addition, the independent test of San Francisco is discussed showing the value of the selected features for mapping an urban scenario not included in the feature extraction phase. The analysis of the texture properties of shadowed areas follows in Section 6.1.3.4.

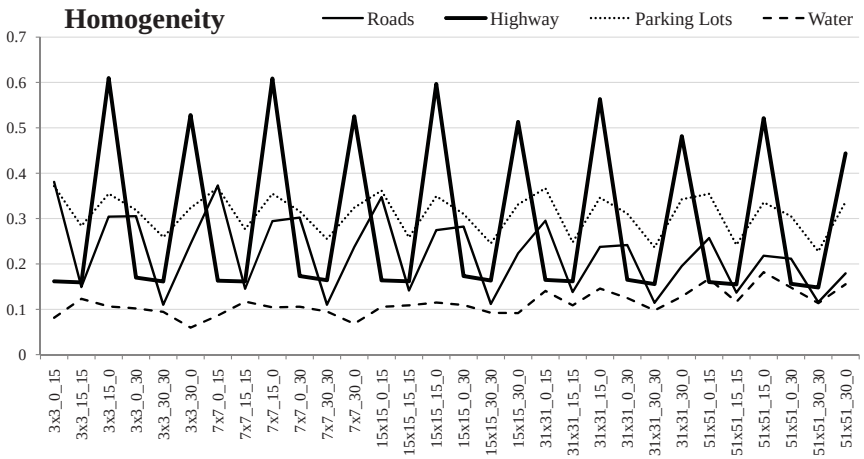
6.1.3.1 Optimization of the feature space and network topology

The first exercise was to produce land-use maps using only panchromatic information. As expected, the results obtained appeared to be really poor for the three test cases in terms of classification accuracy, as shown in Figure 6.7. Several of the defined classes were not recognized. For example, for the Las Vegas scene, only Bare Soil, Residential Houses, Roads and Short Vegetation were identified. In this case, the digital number (DG) values of the image can be grouped together into four separate sets, as shown in Figure 6.8. Even though water and asphalt are different, they show similar values in panchromatic data. The obtained Kappa coefficient values are 0.378 for Las Vegas, 0.184 for Rome and 0.187 for Washington D. C..

A considerable increase in classification accuracy was obtained using the entire set of 191 textural features. More precisely, Kappa coefficient values of 0.916 for Las Vegas, 0.798 for Rome and 0.838 for Washington D. C. have been obtained. With respect to the previous implementation, all classes were identified in the classification maps. On the other hand, as mentioned, a large input space rarely yields high classification accuracies due to



(a) (b) (c)

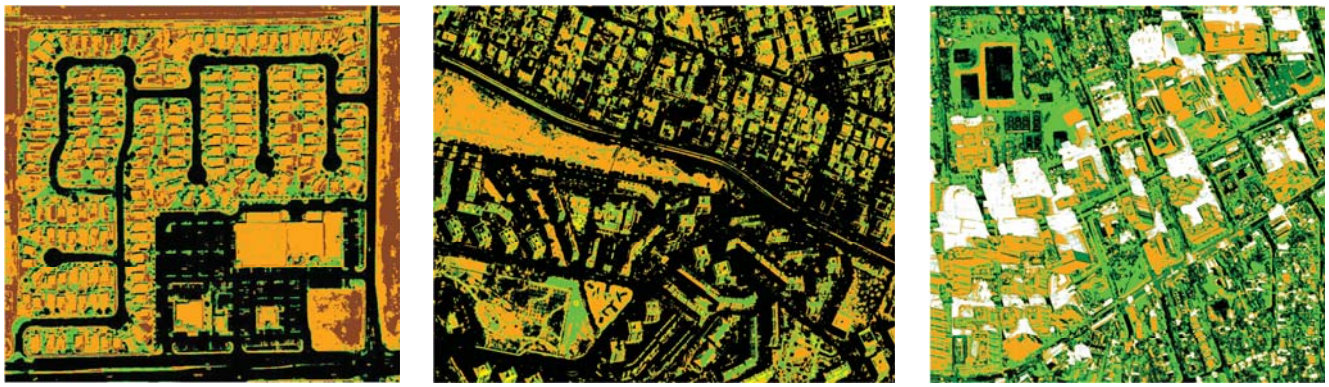


(d)

Figure 6.6: In (a), (b) and (c) is shown the Homogeneity parameter computed over Las Vegas using the same step and window size, but different directions (horizontal, vertical and diagonal, respectively). The directional information highlights different structural patterns within the area, such as vertical or horizontal roads. In (d) is shown the homogeneity values for classes Roads, Highway, Paring Lots and Water computed for all directions, window sizes and shifts. In particular, the highway (which is a horizontal structure) assumes the highest values with respect to the other classes using horizontal parameters while the parking area shows a distinguishing behavior (with respect to the other classes) in the diagonal directions.

information redundancy. This results in the necessity of estimating the contribution of each parameter in order to reduce and optimize the input space. To this end, neural network pruning was exploited to eliminate the weakest connections, optimizing at the same time the network topology. Generally, this process increases the classification accuracy by eliminating features that do not contribute to the classification process, but instead only introduce redundancy.

After the pruning phase, the remaining inputs are 169 for Las Vegas, 140 for Rome and 152 for Washington D. C., respectively. This relatively small feature elimination resulted in a further increase of classification accuracy. In particular, the Kappa coefficient values increased to 0.920 for Las Vegas, 0.941 for Rome and 0.904 for Washington D. C., whose classification maps are illustrated in Figure 6.9 and the accuracies are summarized in Table 6.4 for the reader's convenience. Taking into account the extension of boundary areas between objects, a slight decrease of the classification accuracy for the Washington D. C. case (whose ground reference included boundary areas) can be noted.



(a) (b) (c)

Figure 6.7: Classification maps of (a) Las Vegas, (b) Rome and (c) Washington D. C. using only the panchromatic information. Color codes are in Table 6.2.

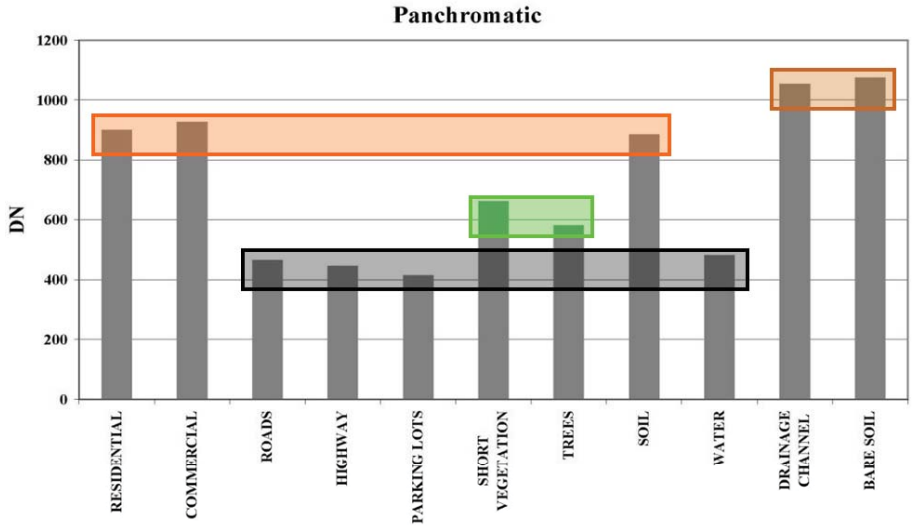
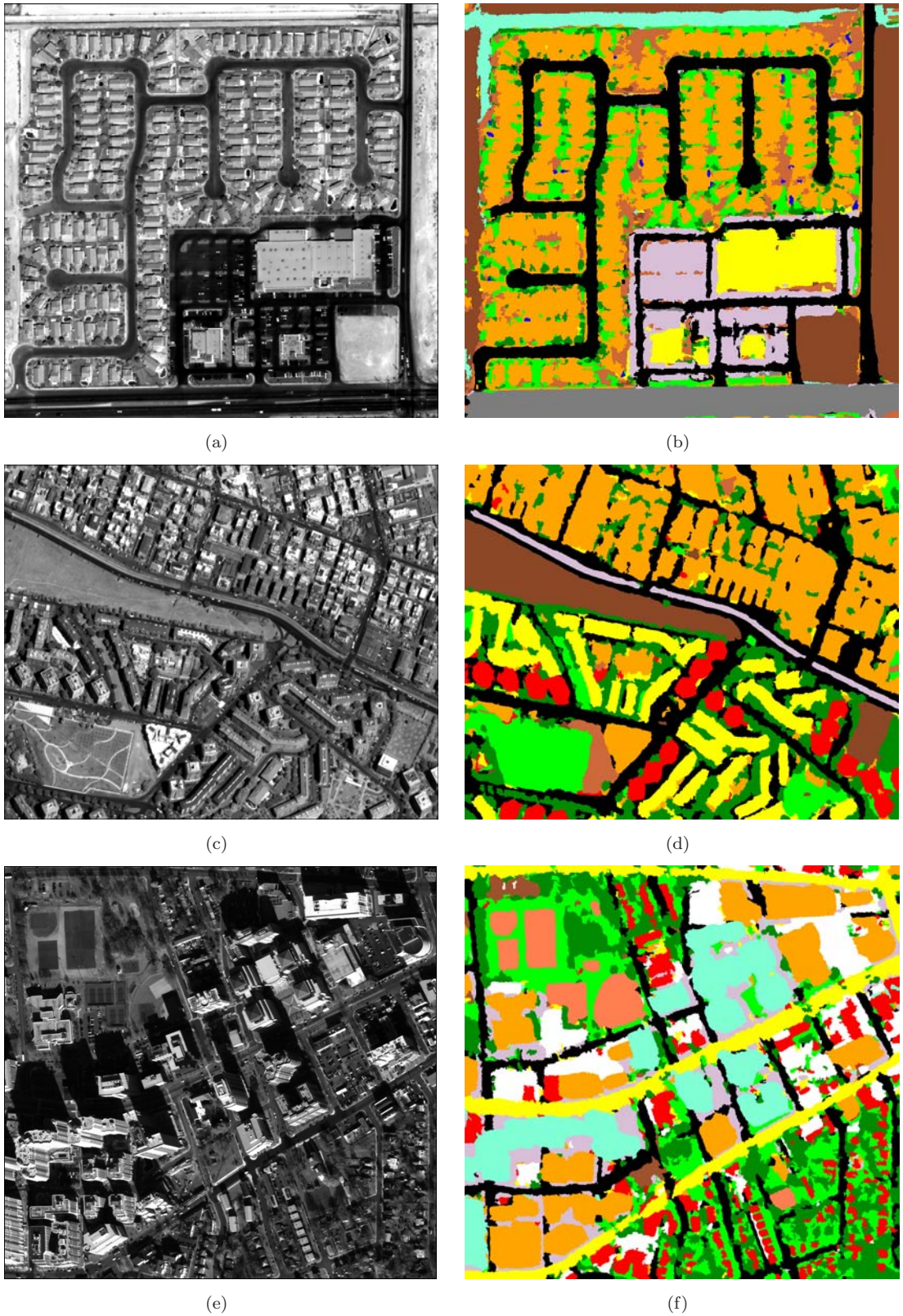


Figure 6.8: Mean digital number of the eleven classes of Las Vegas. Only four groups were identified using the panchromatic band and represented by horizontal blocks (colors are consistent with Table 6.2). Particularly, water was grouped with roads, highways and parking lots, while soil appeared to be closer to residential and commercial buildings than to bare soil.

Table 6.4: Classification accuracies for the Las Vegas, Rome and Washington D. C. cases at the three classification stages.

	Las Vegas			Rome			Washington D. C.		
	Cl. Err. (%)	*Kappa coeff.*	*Num. Inputs*	*Cl. Err. (%)*	*Kappa coeff.*	*Num. Inputs*	*Cl. Err. (%)*	*Kappa coeff.*	*Num. Inputs*
Panchromatic	50.2	0.378	1	66.0	0.184	1	68.6	0.187	1
Full NN	7.1	0.916	191	16.9	0.798	191	14.5	0.838	191
Pruned NN	6.8	0.920	169	5.0	0.941	140	8.6	0.904	155

The obtained classification maps not only discriminated different asphalt surfaces, such as roads, highways and parking lots, but also distinguished traffic patterns in the parking lots due to the different textural information content. This approach also made it possible to differentiate building architectures, sizes and heights, such as residential houses, apartment blocks and towers. It is important to note that shadowed areas did not influence any of the maps obtained. The accuracies obtained with the optimization of the network topology are considered here as an upper bound of the classification accuracies derived from the multi-scale approach.



(a) (b) (c) (d) (e) (f)

Figure 6.9: Original image and classification map of (a) and (b) Las Vegas, (c) and (d) Rome, and (e) and (f) Washington D. C. obtained after the pruning phase. The maps discriminated different asphalt surfaces, such as roads, highways and parking lots due to the different textural information content. Shadowed areas did not influence any of the maps. Color codes are in Table 6.2.

6.1.3.2 Features selection by extended pruning technique

In the previous section, the pruning of the network provided a reduced set of textural features and an optimized topology to obtain the most accurate classification maps. However, the input space is not close to a minimum number of features and a trade-off between classification accuracy and computational time should be found. The extended pruning technique was exploited to identify a minimal sub-optimal textural feature set. The resulting classification is sub-optimal from the classification accuracy point of view, since this further input reduction results in a decrease in the classification accuracy. Particularly, the criterion chosen to stop the extended pruning phase was to reach a classification accuracy of about 0.800 in terms of Kappa coefficient.

After the extended pruning phase, the remaining inputs are 59 for Las Vegas, 61 for Rome and 59 for Washington D. C., with accuracies (Kappa coefficient) decreased to 0.859, 0.820 and 0.796, respectively. In Figure 6.10 is shown the relative contributions of the input features (including the panchromatic image itself), which are not eliminated by the extended pruning. The saliency metric used to compute the feature contributions is illustrated in Chapter 4. As shown, the contribution of each input varies from city to city, due to the architectural peculiarities (and diversity) of them. The analysis of the mean values of these contributions, shown in red, clearly indicates that many of the remaining inputs have a smaller influence on the classification process compared to other features. This means that using certain textural features (including different cell sizes and directions) may have more significance than others.

To analyze the importance of textural parameters regardless of the choice of cell sizes and directions, the feature contributions were computed, for each of them, as sums over the different cell sizes and directions. As shown in Figure 6.11a, the panchromatic band, which does not contain any information on cell sizes and directions, has the smallest contribution. First-order textural features, which do not contain any information on directions, have smaller contributions than second-order features. Dissimilarity appears to be the most informative texture parameter, even if it is similar to Contrast. This may be related to the linear weighting of the gray tone levels of the scene (to be compared to the exponential weight of Contrast). In the same way, the importance of the cell sizes, regardless of the choice of textural parameters and directions was analyzed. This is illustrated in Figure 6.11b, where larger cell sizes (31×31 and 51×51) show higher contributions. This may be related to the very high spatial resolution data. In fact, it is reasonable that textural information is contained in the spatial range of 15.0-25.0 m. This result is consistent with [92].

In Figure 6.11c, the three directions (in black) and the two-step sizes (in gray) have high and similar contributions, meaning that both direction sand step sizes are relevant for the classification phase. This last result points out the necessity of having directional information to better capture differences in textural patterns.

6.1.3.3 Analysis of the best 10 textural features

Many of the remaining inputs, after the extending pruning phase, have a smaller influence on the classification process compared to other features. To further investigate individual feature contributions, the frequency of the input features with respect to the feature contribution is shown in Figure 6.12, which highlights that only very few inputs show a relative contribution greater than 0.30. The best ten features are reported in Table 6.5 with the corresponding values of contribution averaged over three cities and different land-uses.

To understand the contribution of a single class to these ten features, the different land-use classes were merged together into five groups, which are common to the three scenes. The resultant common five classes are: Buildings, Roads, Soil, Trees and Vegetation. The contributions per single class of the ten best features with

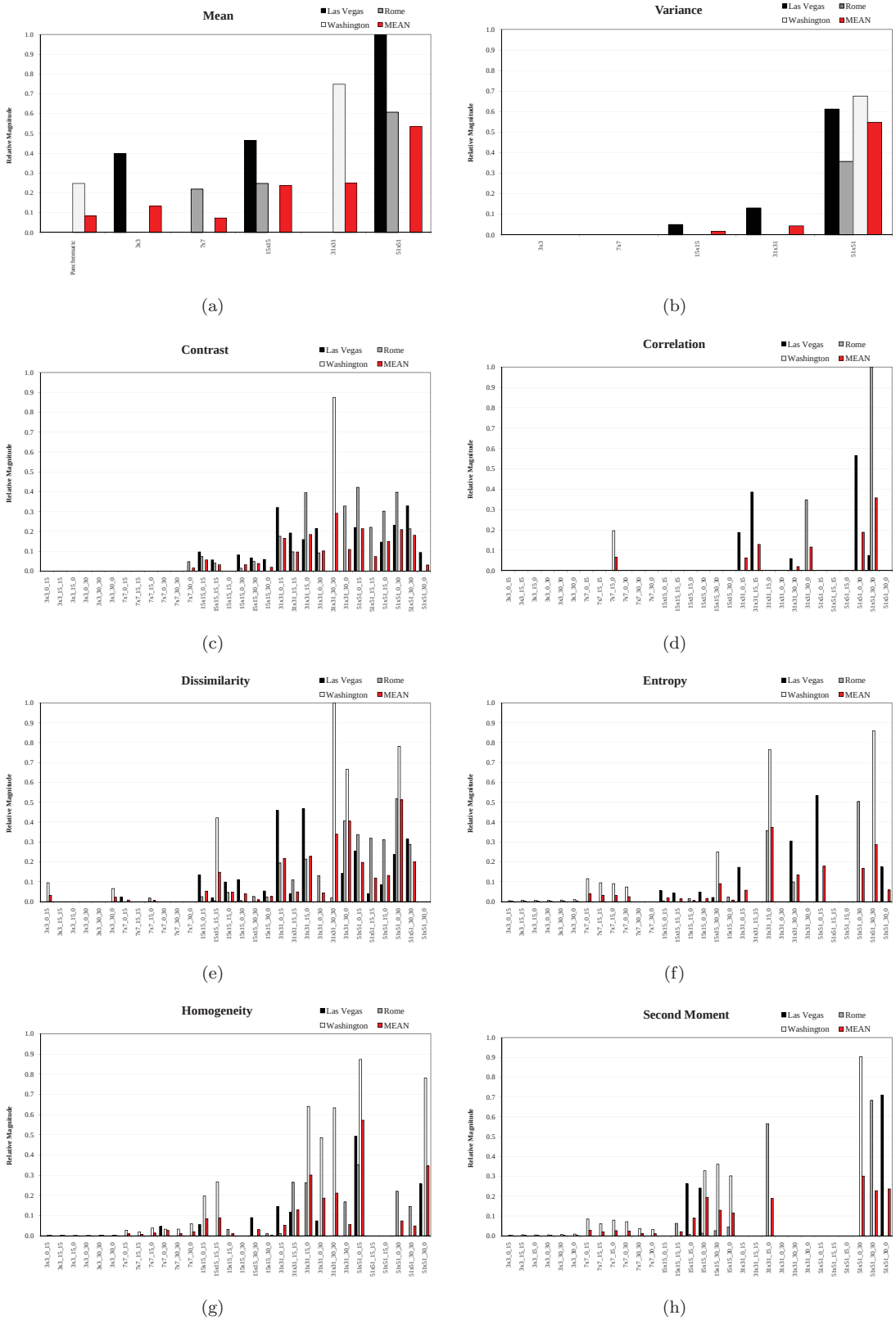
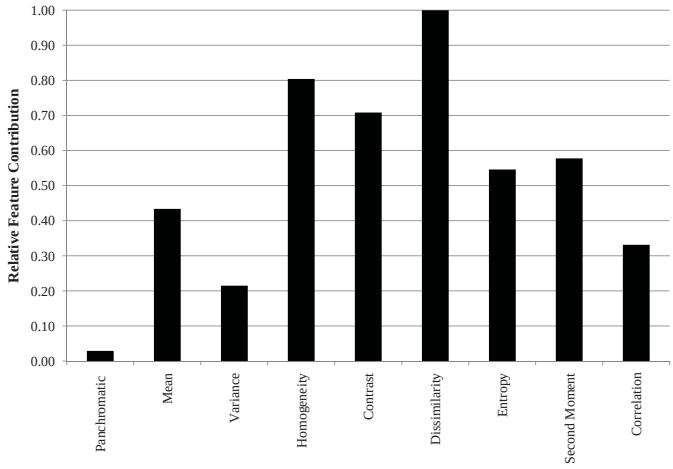
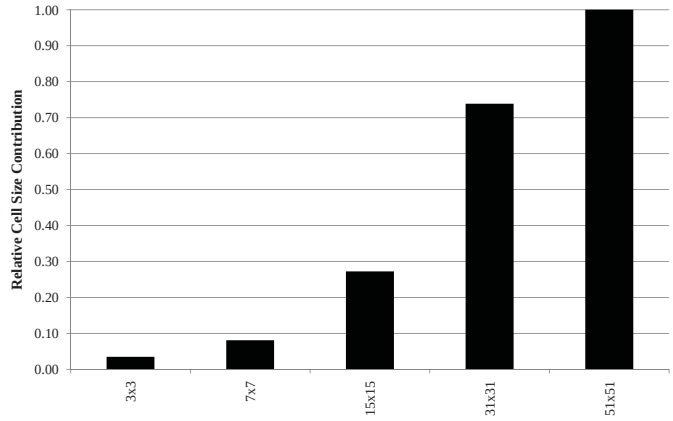


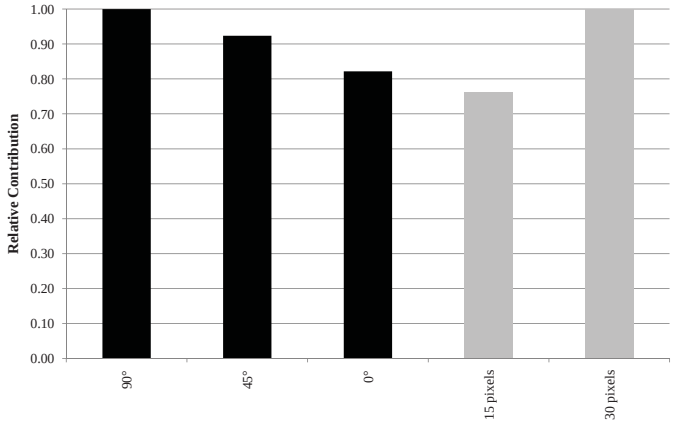
Figure 6.10: Relative feature contribution of the input features not eliminated by the extended pruning of (a) Panchromatic and Mean, (b) Variance, (c) Contrast, (d) Correlation, (e) Dissimilarity, (f) Entropy, (g) Homogeneity and (h) Second Moment computed over all window size, directions and shifts.



(a)



(b)



(c)

Figure 6.11: Feature contributions with respect to textural parameters, window sizes and directions. In (a) is shown the contributions of textural parameters regardless of the choice of cell sizes and directions. In (b) is illustrated the importance of the cell sizes, regardless of the choice of textural parameters and directions. In (c) is shown the contribution of the three directions (in black) and the two-step size (in gray).

Table 6.5: Best ten features and corresponding contribution values averaged over three cities and different land-uses classes.

Best 10 features	Feature Contribution
Mean 51×51	0.535
Variance 51×51	0.547
Homogeneity 51×51_0_15	0.572
Homogeneity 51×51_30_0	0.345
Dissimilarity 31×31_3_30	0.339
Dissimilarity 31×31_30_0	0.404
Dissimilarity 51×51_0_30	0.512
Entropy 31×31_15_0	0.374
Second Moment 51×51_0_30	0.301
Correlation 51×51_3_30	0.357

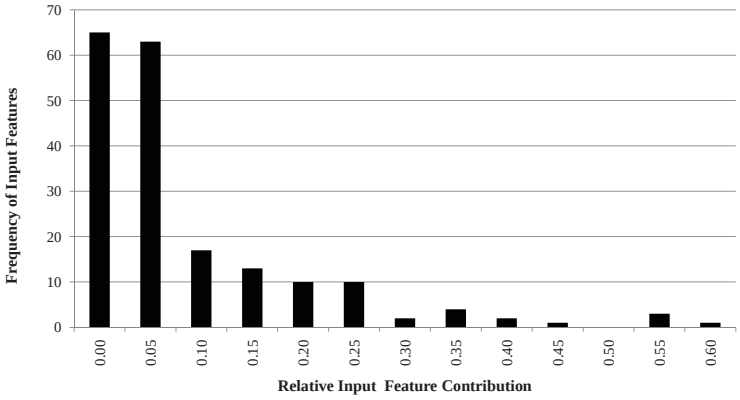


Figure 6.12: Frequency of the input features with respect to the feature contribution. Only very few inputs show a relative contribution greater than 0.30.

respect to these five classes are illustrated in Figure 6.13. The first-order Mean 51×51 seems to be appropriate for the discrimination of roads, while Dissimilarity 51×51_0_30 appears to be valuable for the detection of trees.

With this drastic reduction of input features (from about 60 to 10), a further decrease of the classification accuracy with respect to the extended pruning results was expected. On the other hand, this effect was compensated by the reduction of the output classes (from about 11 to 5). In particular, the Washington D. C. scene was re-classified using only the ten best textural inputs, obtaining an accuracy of 0.861. This result is particularly relevant since it was obtained after a generalization (mean) of results obtained over all of the cities with different architectures.

Even though these considerations show similarities, especially in terms of computational time and generalization capability for different urban scenarios, it is necessary to emphasize the importance of exploiting the entire textural feature data set, including all different spatial scales and directions. Starting from the same textural features and using network pruning made it possible to classify different urban scenarios with high accuracies, showing both the efficiency and the robustness of the multi-scale approach used here.

6.1.3.3.1 The San Francisco test case The feature contributions as mean values over three different test sets corresponding to Las Vegas, Rome and Washington D. C. were discussed in the previous sections. A reduced set of ten features has been identified as valuable for urban classification. Now the question is: how well do these ten features classify a new urban scene? To answer this question, these ten features were used to classify an independent data set of a portion of San Francisco. Note that different combinations of texture metrics might produce more accurate results for this particular test case. However, this experiment was to investigate

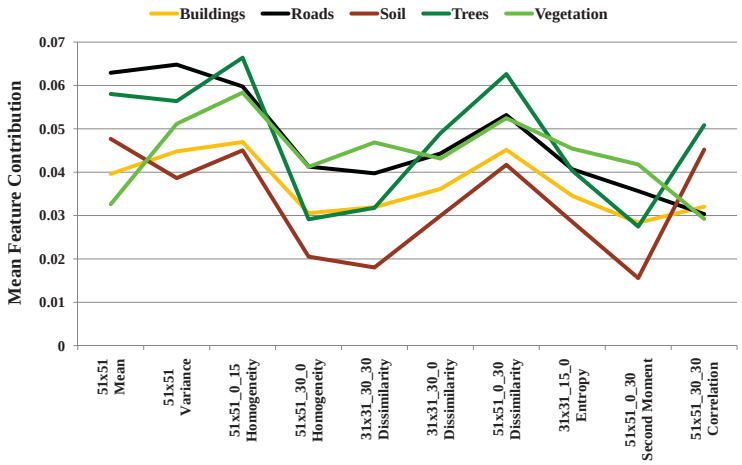


Figure 6.13: Contributions per single class of the best ten features with respect to the five common classes defined. The first-order Mean and Variance 51×51 seem to be more appropriate for the discrimination of roads than Correlation or Second Moment, while Dissimilarity 51×51_0_30 appears to be valuable for the detection of trees.

Table 6.6: Classes, training and validation samples, and color legend.

Classes	TR	VA	Color
Buildings	36,689	85,240	orange
Roads	49,436	114,857	black
Soil	6,524	15,158	brown
Trees	7,745	17,993	dark green
Vegetation	1,465	3,404	green
Total	**101,859**	**236,652**	

the capability of the selected features (obtained as an average of 3 different conditions) when applied to a new scenario. In this sense, these ten features are not an optimal combination, but simply a set of inputs that may potentially increase the classification accuracy when applied to very high spatial resolution imagery.

The scene, shown in Figure 6.14a, was acquired by WorldView-1 with an off-nadir angle of 19.6°. Further, long shadows are caused by a sun elevation of 29.6°. This neighborhood may be considered a representative architecture for many cities, making it suitable for validation purposes. The same five common classes defined previously were investigated. Training and validation pixels were selected using SRS (see Table 6.6). The corresponding ground reference map is shown in Figure 6.14b.

The panchromatic image was first classified alone, obtaining a Kappa coefficient of about 0.224. Successively, the map shown in Figure 6.14c was produced using only the reduced set of ten textural features with an accuracy of about 0.917.

6.1.3.4 Analysis of texture properties of shadowed areas

Shadow effects are often ignored when using decametric spatial resolution images, such as Landsat. In these cases, shadowed pixels may be located on an object's boundaries where there is a mixture of radiances caused by different surfaces. Vice versa, shadows have a huge impact on classification with metric or sub-metric spatial resolution images. In urban areas, shadows are mainly caused by buildings, tress or bridges and may potentially provide additional geometric and semantic information on the shape and orientation of objects. On the other hand, shadowed surfaces require more consideration since they may cause a loss of information or a shape distortion of objects.

(a) (b) (c)

Figure 6.14: (a) Panchromatic image and (b) ground reference of San Francisco. In (c) is shown the classification map obtained using the reduced set of ten textural features. Color codes are in Table 6.6.

Table 6.7: Normalized textural values (and their standard deviation) of the panchromatic band and the ten most contributing features of the Rome case for shadowed (SH) and non-shadowed (NON-SH) pixels.

Panchromatic band and best 10 features	Blocks		Roads		Vegetation	
	SH	*NON-SH*	*SH*	*NON-SH*	*SH*	*NON-SH*
Panchromatic	0.08 (0.13)	0.58 (0.26)	0.05 (0.07)	0.25 (0.16)	0.07 (0.11)	0.65 (0.14)
Mean 51×51	0.25 (0.22)	0.41 (0.24)	0.21 (0.21)	0.27 (0.20)	0.28 (0.21)	0.72 (0.23)
Variance 51×51	0.61 (0.17)	0.52 (0.20)	0.52 (0.26)	0.40 (0.25)	0.67 (0.19)	0.23 (0.23)
Homogeneity 51×51_0_15	0.52 (0.25)	0.42 (0.26)	0.54 (0.27)	0.55 (0.25)	0.65 (0.21)	0.80 (0.19)
Homogeneity 51×51_30_0	0.37 (0.21)	0.43 (0.26)	0.38 (0.28)	0.58 (0.26)	0.22 (0.20)	0.73 (0.27)
Dissimilarity 31×31_30_30	0.55 (0.27)	0.49 (0.24)	0.57 (0.28)	0.43 (0.28)	0.70 (0.23)	0.26 (0.23)
Dissimilarity 31×31_30_0	0.63 (0.18)	0.54 (0.23)	0.63 (0.26)	0.38 (0.25)	0.78 (0.21)	0.28 (0.25)
Dissimilarity 51×51_0_30	0.55 (0.25)	0.55 (0.23)	0.50 (0.28)	0.42 (0.27)	0.56 (0.24)	0.23 (0.23)
Entropy 31×31_15_0	0.46 (0.19)	0.51 (0.19)	0.44 (0.23)	0.33 (0.20)	0.45 (0.23)	0.24 (0.21)
Second Moment 51×51_0_30	0.51 (0.20)	0.46 (0.20)	0.51 (0.24)	0.60 (0.24)	0.42 (0.21)	0.77 (0.24)
Correlation 51×51_30_30	0.31 (0.17)	0.35 (0.22)	0.36 (0.27)	0.50 (0.28)	0.32 (0.21)	0.70 (0.23)

In the literature, shadow is generally dealt with as an additional class (see for example Benediktsson *et al.* [34]; Bruzzone and Carlin [93]). In this study, the ground references have been defined regardless of the presence of shadowed areas, leading to classification maps, which do not contain any pixels of shadow. To further investigate this, the shadowed pixels of the Rome scene were extracted and their textural values (normalized between 0 and 1) analyzed with respect to the non-shadowed pixels of the same class. As illustrated in Figure 6.15, where continuous-lines represent pixels of shadow and dotted-lines represent the non-shadowed pixels of the corresponding class, only panchromatic information appears to not be able to adequately separate the classes whose pixels are covered by shadow. In fact, they are mainly concentrated in the lower part of the scale values, acting more as a unique class. Mean values (and their standard deviation) are also reported in Table 6.7 for Blocks, Roads and Vegetation. Even though these surfaces are different, the (normalized) panchromatic values of shadowed areas are very similar: 0.08, 0.05 and 0.07, respectively. In contrast, shadowed areas have their own texture properties, allowing the discrimination between different shadowed classes.

To make Figure 6.15 clearer, only the class Buildings (including both Apartment Blocks and Towers) was considered. Shadows of buildings are mainly due to two different effects: i) shadows of the building on itself and ii) shadows of other objects, such as other buildings or trees, on a building. As illustrated in Figure 6.16, buildings show a sort of characteristic textural signature. For example, the Variance feature of shadowed buildings is slightly higher than non-shadowed buildings. This may be interpreted by considering the smaller extension in the area of

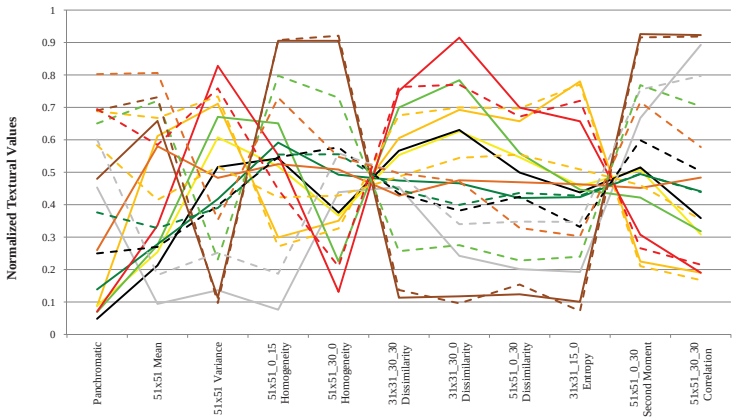


Figure 6.15: Normalized textural values of the ten most contributing features and panchromatic band of the Rome case for shadowed (continuous-lines) and non-shadowed (dotted-lines) pixels. The only panchromatic information appears to not separate sufficiently the classes whose pixels are covered by shadow, since the result mainly concentrated in the lower part of the scale values. In contrary, shadowed areas show their own texture properties, allowing the discrimination between different shadowed classes. Color codes are in Table 6.2.

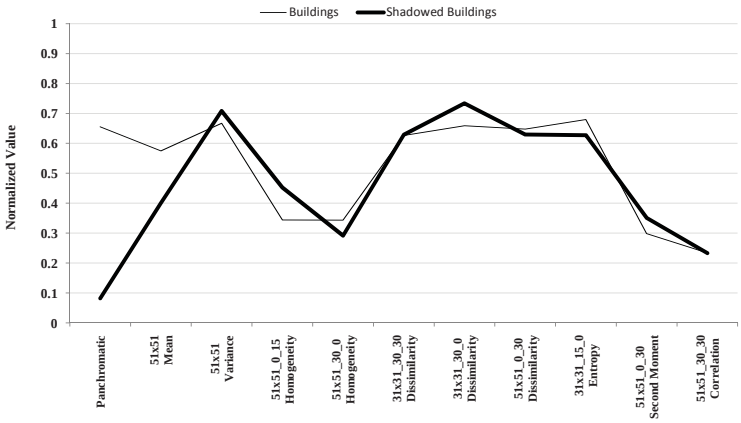


Figure 6.16: Normalized textural values of the ten most contributing features and panchromatic band of the Rome case for shadowed and non-shadowed pixels of buildings (including also apartment blocks and towers). The Variance of shadowed buildings is slightly higher than non-shadowed buildings, while Homogeneity shows an inversion of the trend between shadowed and non-shadowed buildings using a larger step size.

shadow with respect to buildings (for the scene considered) and the higher contrast in terms of gray-tone levels on the boundary between shadowed and non-shadowed pixels. Smaller objects with higher contrast lead to higher spatial variability, thus larger variance values. Similar considerations may be given to the Homogeneity feature, which shows an inversion of the trend between shadowed and non-shadowed buildings due to the larger step size (30 instead of 15 pixels), but equal window size of 51×51 pixels.

6.1.4 Summary

In this section, the potential of very high resolution panchromatic QuickBird and WorldView-1 imagery was investigated to classify the land-use of urban environments. Spectral-based classification methods may fail with the increased geometrical resolution of the available data. In fact, finer spatial resolution data increases within-class variances, which results in higher interclass spectral confusion. This problem is intrinsically related to the sensor resolution and it cannot be solved by increasing the number of spectral channels. To overcome the spectral

information deficit of panchromatic imagery, it is necessary to extract additional information to recognize objects within the scene.

The multi-scale analysis exploited the contextual information of first- and second-order textural features to characterize land-use. For this purpose, textural parameters were systematically investigated computing the features over five different window sizes, three different directions and two different cell shifts for a total of 191 inputs. Neural network pruning and saliency measurements allowed the optimization of network topology and gave an indication of the most important textural features for sub-metric spatial resolution imagery and urban scenarios.

Network pruning appeared to be necessary in a neural-net based classification. As summarized in Table 6.4, using the full neural network, the classification accuracies were modest (about 0.8 in terms of Kappa coefficient). After network pruning, the maps of the three data set exploited for the texture analysis, i.e. Las Vegas, Rome and Washington D. C., showed higher land-use classification accuracies, above 0.90 in terms of Kappa coefficient computed over more than a million independent validation pixels. The network pruning greatly improved the classification accuracies due to two main effects: (1) the network topology was optimized; (2) the smaller set of input features reduced the effect of the *curse of dimensionality*.

The identification of the most effective textural features was carried out using the extended pruning technique with saliency measurements. Fist-order textural features, which do not contain any information on direction, had smaller contributions than second-order features. Dissimilarity appeared to be the dominant texture parameter. For the spatial resolution and test cases considered, bigger cell sizes, such as 31×31 and 51×51 pixels, had higher contributions than smaller cell sizes (regardless of the choice of textural parameters and directions), making it clear that there is a need to exploit the entire directional information. However, after the extending pruning phase many of the inputs remaining had a smaller influence on the classification process compared to other features. Specifically, only very few inputs showed a relative contribution greater than 0.30, as reported in Table 6.5. As expected, the drastic reduction of input features (from about 60 to 10) decreased the classification accuracy of the Washington D. C. scene to 0.861 (Kappa coefficient) with respect to the extended pruning result. Nevertheless, the simple selection of these features resulted in remarkable results compared to those obtained using only the panchromatic information. Furthermore, the result obtained for the independent test case of San Francisco, i.e. not included in the multi-scale textural analysis carried out in the first part of this analysis, indicated the potentiality of the reduced set of textural features for mapping a common urban scene.

Since the textural analysis was carried out as an average of three different environments (and validated over more than a million independent samples), this approach may efficiently be extended to large areas, such as an entire city. In this sense, the San Francisco scene may represent a non exhaustive, but significant, example.

To conclude, only panchromatic information appeared not to be able to adequately separate classes whose pixels are covered by shadow (these values can be grouped together in Figure 6.8 and Figure 6.15, acting as a unique class). On the contrary, the multi-scale textural analysis proved that it is possible to distinguish different shadowed areas, since they have their own texture properties.

6.2 Mathematical morphology

As shown in the previous section, the use of spatial features for the classification of satellite images can overcome the lack of spectral information of single-band sensors. In particular, it was shown the use of texture features for classifying very high resolution QuickBird and WorldView-1 urban scenes with neural networks. Other techniques

to extract spatial information have been proposed in the literature, such as Markov Random Fields [94][95][79][96] or Gabor Filters [97][98].

A different way to integrate contextual information is to extract shapes of single objects using *mathematical morphology*. Mathematical morphology provides a collection of image filters (called operators) based on set theory. The effective results obtained in [99][74] with panchromatic imagery using basic operators such as *opening* and *closing* have focused the attention of the scientific community on the use of morphological analysis for image classification. Morphological operators were used to classify remote sensed images at decametric and metric resolutions and have been highlighted as very promising tools for data analysis, as reported in [100][101]. In [102], the authors used morphological operators for image segmentation. Pesaresi and Benediktsson [100] proposed building Differential Morphological Profiles (DMP) to account for differences in the values of the morphological profiles at different scales. These profiles were exploited in [100][103] for the segmentation of Indian Remote Sensing Satellite 1C (IRS-1C) panchromatic imagery, using the maximal derivative of the DMP. In [33], complete DMP was applied to IRS-1C and IKONOS panchromatic imagery. Specifically, the authors exploited reconstruction DMP with neural network classifiers and two linear feature extraction methods to reduce the redundancy in information. More recently, the extraction of morphological information from hyper-spectral imagery was addressed. In [34][104], the first principal component was used to extract the morphological images. In [105], extended opening and closing operators by reconstruction were investigated for target detection on both Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) and Reflective Optics System Imaging Spectrometer (ROSIS) data. In [35], multi-channel and directional morphological filters were used for hyper-spectral image classification, showing its ability to account simultaneously for the scale and orientation of objects.

In the following section, results of an extensive analysis of different morphological operators are reported with the goal of investigating the relevancy of the most contributing features when applied to very high spatial resolution optical and SAR data for land-cover classification of urban scenes. To account for the spatial setting of different cities, these parameters have been calculated over a range of window scales. The effects of different filters, as well as their scale, are addressed and discussed in detail. Section 6.2.1 recalls the basic theory of mathematical morphology. Then, morphology is applied to very high spatial resolution optical and SAR data in Section 6.2.2 and Section 6.2.3, respectively.

6.2.1 Morphological operators

Morphological operators are a collection of filters based on set theory. These operators are applied to two ensembles, the image g to analyze and a structuring element B, which is a set with known size and shape that is applied to the image as a filter. When centered on a pixel x, $\mathbf{B}$ is a vector that takes into account all the values x_b of the pixels of g covered by the structuring element B.

Morphological operators can be summarized into two fundamental operations: *erosion* $\epsilon_B(g)$ and *dilation* $\delta_B(g)$ [74], whose principles are illustrated in Figure 6.17 for binary images. Basically, erosion deletes all pixels whose neighborhood can not contain a certain structuring element (SE), i.e. it performs an intersection between the binary image g and B. On the contrary, dilation provides an expansion by addition of the pixels contained in the SE, i.e. a union between g and B. Mathematically, erosion and dilation can be represented as:

$$\epsilon_B(g) = \bigcap_{b \in B} g_{-b} \qquad \delta_B(g) = \bigcup_{b \in B} g_{-b} \tag{6.2.1}$$

Binary morphology can be extended to gray-scale images by considering them as a topographic relief, where

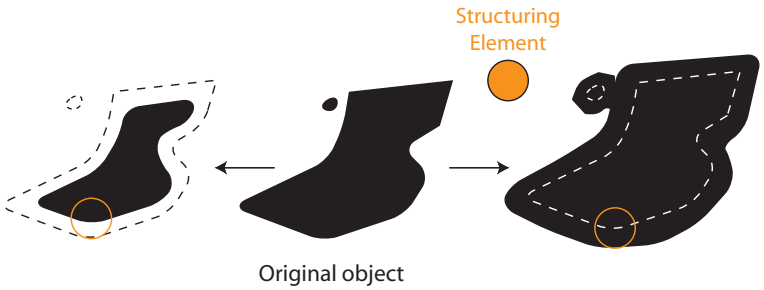


Figure 6.17: Erosion (left) and dilation (right) using a circular structuring element.

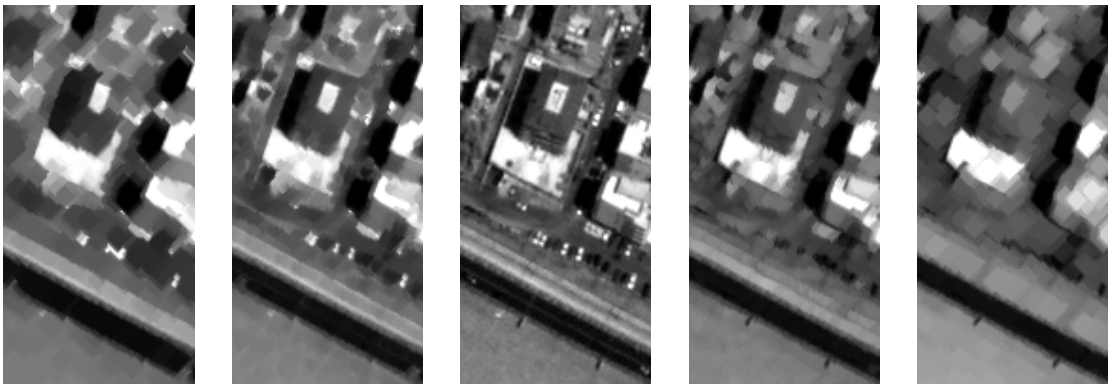


Figure 6.18: Progressive opening and closing operators using a diamond-shaped structuring element.

brighter tones correspond to higher elevation [99][35]. In gray-scale morphology intersection $\cap$ and union $\cup$ become the pointwise minimum $\wedge$ and maximum $\vee$ operators [101].

In the following, the morphological filters are briefly discussed as combinations of erosion and dilation operators.

6.2.1.1 Opening and closing

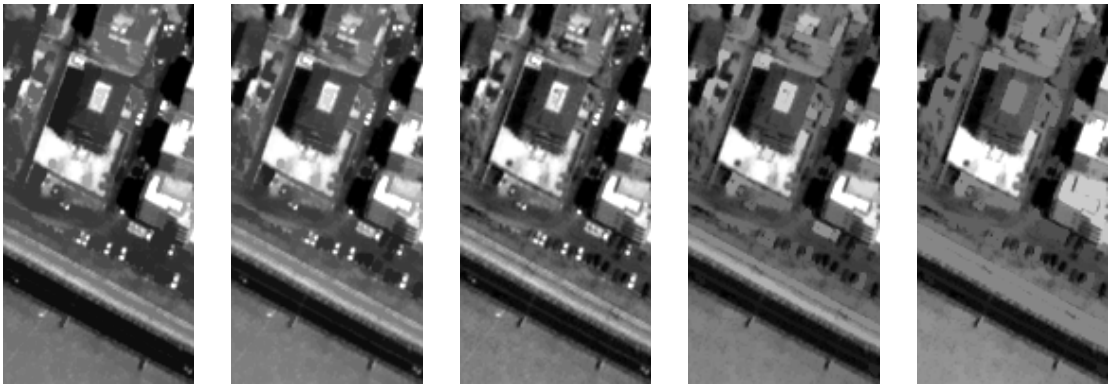
Two of the most common morphological filters are *opening* $\gamma_B(g)$ and *closing* $\phi_B(g)$ operators. Opening is the dilation of an eroded image and is widely used to filter brighter (compared to surrounding features) structures in gray-scale images. On the contrary, closing is the erosion of a dilated image and allows one to filter out darker structures. Opening and closing operators can be represented as:

$$\gamma_B(g) = \delta_b \circ \epsilon_B(g) \qquad \phi_B(g) = \epsilon_B \circ \delta_B(g) \tag{6.2.2}$$

Figure 6.18 illustrates a series of openings and closings using structuring elements with increasing sizes. It is possible to note that the shapes of the objects in the images are not preserved. In general, this is not a desirable behavior in image classification. To preserve original shapes, Crespo *et al.* [106] proposed to use opening and closing *by reconstruction* operators instead of simple opening and closing operators.

6.2.1.2 Reconstruction filtering

Reconstruction filters (see Figure 6.19) provide an iterative reconstruction of the original image g starting from a mask I. If the mask I is the erosion $\epsilon_B(g)$, the original brighter features are filtered by geodesic dilation



Closing Rec. 11 pixels | Closing Rec. 5 pixels | Panchromatic | Opening Rec. 5 pixels | Opening Rec. 11 pixels

Figure 6.19: Progressive opening and closing by reconstruction operators using a diamond-shaped structuring element.

(opening by reconstruction). On the contrary, if the mask I is the dilation $\delta_B(g)$, the original darker features are filtered by geodesic erosion (closing by reconstruction). A geodesic dilation (respectively erosion) is the pointwise minimum (maximum) between the dilation (erosion) of the marker image and the original image. Equation 6.2.3 and Equation 6.2.4 illustrate opening and closing by reconstruction, respectively:

$$\rho^{\delta}[\epsilon_B(g)] = \rho^{\delta}(I) = \min\left\{x^g, \delta_B^k(I)\right\} | \delta_B^k(I) = \delta_B^{k-1}(I) \tag{6.2.3}$$

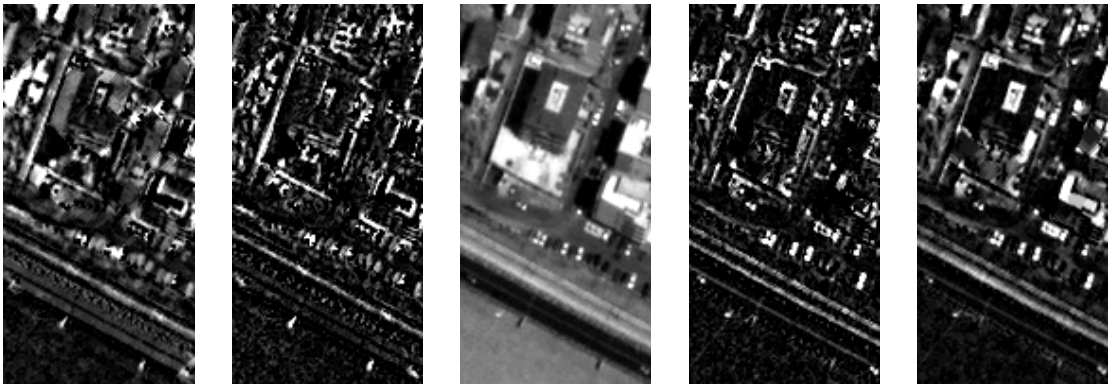
$$\rho^{\epsilon}[\delta_B(g)] = \rho^{\delta}(I) = \max\left\{x^g, \epsilon_B^k(I)\right\} | \epsilon_B^k(I) = \epsilon_B^{k-1}(I) \tag{6.2.4}$$

The reconstruction process is iterated until the reconstructed image at iteration k is identical to the image obtained at iteration $k-1$ (i.e. $\delta_B^k(I) = \delta_B^{k-1}(I)$ in Equation 6.2.3 and $\epsilon_B^k(I) = \epsilon_B^{k-1}(I)$ in Equation 6.2.4, respectively). By comparison with Figure 6.18, the difference is rather obvious. Using opening by reconstruction, the shape of the objects is preserved and the progressive increase of the size of the structuring element results in a progressive disappearance of objects whose pixels are eliminated by erosion using larger structuring elements. These objects can not be recovered during the reconstruction. The size of the structuring element used in the reconstruction controls the strength of the reconstruction. Using larger structuring elements, peaks and valleys of the marker are filled (or taken out, respectively) quickly and only very bright (or dark, respectively) structures are recovered. On the contrary, using smaller size structuring elements in the reconstruction phase, the reconstruction is reached gradually and gray structures are also recovered.

6.2.1.3 Top hat

Top-hat operators (see Figure 6.20) are the residuals of an opening (or a closing) image, when compared to the original image. Therefore, top-hat images show the peaks (opening by top-hat, OTH) or valleys (closing by top-hat, CTH) of the image. This correspond to the following:

$$\begin{aligned} \text{OTH} &= g - \gamma_B(g) \\ \text{CTH} &= \phi_B(g) - g \end{aligned} \tag{6.2.5}$$



CTH 11 pixels | CTH 5 pixels | Panchromatic | OTH 5 pixels | OTH 11 pixels

Figure 6.20: Progressive opening and closing top hat operators using a diamond-shaped structuring element.

Table 6.8: Ground reference for the Las Vegas image and number of samples used for training, validation and test.

Class	Color	Training	Validation	Test
Residential	orange	7,066	5,971	74,553
Commercial	red	1,816	1,463	19,485
Road	black	6,089	5,063	65,068
Highway	dark gray	2,858	2,372	30,220
Parking Lots	light gray	2,291	1,929	23,990
Short vegetation	light green	1,815	1,505	19,090
Trees	dark green	1,006	877	11,157
Soil	light brown	1,484	1,246	15,670
Water	blue	152	91	1,227
Drainage channel	cyan	1,098	958	12,224
Bare soil	dark brown	4,325	3,525	45,339
Total		**30,000**	**25,000**	**318,023**

Table 6.9: Ground reference for the Rome image and number of samples used for training, validation and test.

Class	Color	Training	Validation	Test
Buildings	orange	11,646	6,995	162,613
Apartment blocks	yellow	7,033	4,323	98,464
Roads	black	10,645	6,209	146,676
Railway	gray	1,049	638	14,373
Vegetation	light green	4,408	2,747	62,465
Trees	dark green	5,883	3,532	81,465
Bare soil	dark brown	5,306	3,179	72,785
Soil	light brown	929	569	13,562
Towers	red	3,101	1,808	43,008
Total		**50,000**	**30,000**	**695,411**

The OTH operator represents the bright peaks of the image. On the contrary, the CTH operator represents the dark peaks (or valleys) of the image.

6.2.2 Morphology applied to very high spatial resolution optical imagery

In this section, the relevancy of morphological operators for the classification of urban land-use using sub-metric panchromatic imagery is investigated. Two panchromatic QuickBird images were used for the morphological analysis. The first image was taken over Las Vegas in 2002, while the second was acquired over Rome in 2004. Both scenes were previously described in Section 6.1.1. The reference ground surveys are illustrated in Figure 6.4, while the number of samples used as training, validation and test are reported in Table 6.8 and Table 6.9.

Eight experiments were investigated using different morphological sets as reported in Table 6.10. Specifically, the following six morphological filters were considered:

Table 6.10: Morphological feature sets investigates.

Code	Panchromatic	O	C	OR	CR	OTH	CTH	Num. Features
Panchromatic	x							1
OC	x	x	x					19
OCR	x			x	x			19
TH	x					x	x	19
OC–OCR	x	x	x	x	x			37
OC–TH	x	x	x			x	x	37
OCR–TH	x			x	x	x	x	37
OC–OCR–TH	x	x	x	x	x	x	x	55
RFE–#†	x	x	x	x	x			†

† the number of features is selected after RFE on the OC–OCR set (see discussion)

- opening (O)
- closing (C)
- opening by reconstruction (OR)
- closing by reconstruction (CR)
- opening top-hat (OTH)
- closing top-hat (CTH)

For each of these filters, a SE whose dimensions increased from 9 to 25 pixels with steps of 2 pixels was used, resulting in 9 morphological features. The size of the structuring elements was chosen according to the image resolution. For the Las Vegas data set, a square structuring element was exploited to take into account the major directions of the objects on the image, which are 0° and 90°. The Rome data set, being characterized by an overall 45° angle in the disposition of the objects, a diamond-shaped was used for a better reconstruction of the borders of the objects. The reconstruction was performed using a small (3-pixels diameter) structuring element.

As for texture features, the use of morphological operators may result in a very high input space dimensionality, since it is possible to extract many morphological features using different filters or by changing the size and the shape of the structuring elements. Therefore, also in this case, feature extraction was a central problem. Here, the SVMs-based recursive feature elimination method, discussed in Section 4.2, was exploited for comparison to NNs.

The classification was performed using a one-against-all SVM. A Radial Basis Function (RBF) kernel was used for all experiments. Kernel parameters $\theta = \{\sigma, C\}$ were optimized by a grid search in the ranges $\sigma = \{0.01, \ldots, 1.3\}$, $C = \{1, \ldots, 51\}$ based on previous experiments. Labels of a model selection set were predicted by each model and the one showing the higher accuracy was retained. The optimal model was then used to predict a new unseen data set: the data test.

To evaluate the performance of the models, both percentage accuracy and Kappa coefficient were used. Since overall accuracies of the models may often be similar, statistical significance of the difference between the results were assessed using the McNemar test. This test compares the classification results for related samples by assessing the standardized normal test statistic z for two thematic maps [46]. For a confidence interval of $\alpha = 5\%$, the first map is significantly different than the second when $|z| > 1.96$. In Table 6.11, Table 6.12, Table 6.13 and Table 6.14 the sign "*" is added alongside of the overall accuracy when the result is significantly different from the best result reported in the table.

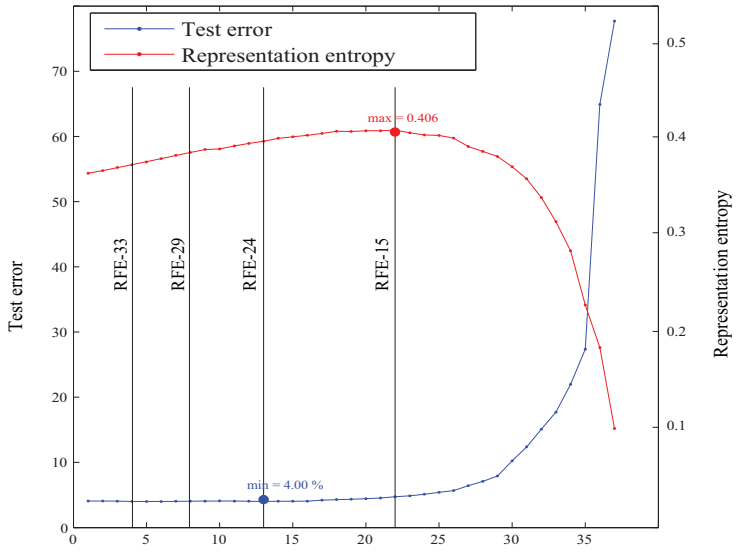


Figure 6.21: Application of the three criteria for the selection of the optimal number of features (example of Las Vegas - 37 initial features). Prior knowledge results in the sets RFE–33, RFE–29. Test error minimum (blue) results in the RFE–24 set. Representation entropy maximum (red) results in the RFE–15 set.

One drawback of the RFE is that it does not provide a straightforward stopping criterion. This means that the algorithm runs until all the features are removed. Thus, a prior knowledge about the desired number of features is necessary. The selection of the number of features was based on the following three criteria:

- Prior knowledge: the number of selected features was decided *a priori*. In the experiments, 4 and 8 features were removed. These solutions were conservative, because they preserved most of the features

- Test error: a test error was computed at each iteration using the new features set. The feature set related to the minimal test error was selected

- Representation entropy: at each RFE iteration, the d eigenvalues $\tilde{\lambda_i} = \lambda_i / \sum_{i=1}^{d} \lambda_i$ of the covariance matrix of the current d-dimensional feature set provide the distribution of the information between the d features. If the distribution of the eigenvalues is uniform, the maximal degree of compression possible using these features is achieved [107]. An entropy function can be computed to evaluate the degree of compression [108]:

$$H_R = -\sum_{i=1}^{d} \tilde{\lambda_i} log \tilde{\lambda_i} \tag{6.2.6}$$

  This quantity is called *representation entropy*. It has a minimum (zero) when all the eigenvalues except one are zero and has a maximum when all the eigenvalues are equal (uniform distribution)

The application of the three criteria for the selection of the optimal number of features is shown in Figure 6.21. For a comparison, classical Principal Components Analysis (PCA) extraction of 10 features (accounting for 99.5% of the original information in both cases) was added.

Table 6.11: Classification accuracies (in percent) and Kappa coefficient for the Las Vegas data set.

Class	Pan	OCR	TH	OC	OC–TH	OCR–TH	OC–OCR	OC–OCR–TH
Residential	70.60	87.08	95.96	**96.02**	95.42	95.01	**96.10**	95.27
Commercial	4.38	93.02	91.81	**96.02**	96.40	94.13	**97.41**	96.70
Roads	82.56	86.12	96.35	**97.03**	96.82	96.86	**97.11**	96.67
Highway	0.50	86.26	94.55	**97.14**	97.81	95.85	**98.31**	97.89
Parking Lots	0.34	55.27	86.91	**90.97**	89.68	89.11	**91.90**	90.98
Short vegetation	0.01	71.50	86.12	**92.36**	90.69	90.22	**92.02**	91.35
Trees	2.17	55.17	77.75	**85.88**	84.59	84.14	**87.26**	86.04
Soil	1.21	67.44	74.99	**86.31**	84.37	81.60	**89.68**	87.28
Water	0.08	80.21	90.15	**93.32**	93.97	92.02	**94.79**	**94.79**
Drainage Channel	0.07	82.75	87.26	**94.63**	96.83	91.80	**97.23**	97.11
Bare soil	73.64	95.44	96.12	**98.18**	99.26	97.37	**99.52**	99.36
Overall Accuracy	44.42*	82.73*	92.37*	**95.14***	94.95*	93.84*	**95.93**	95.27*
Kappa coefficient	0.321	0.797	0.911	**0.943**	0.941	0.928	**0.952**	0.945

* significant difference from the OC–OCR result by the McNemar test

(a) (b) (c)

Figure 6.22: Classification maps for the Las Vegas image using (a) only the panchromatic band, (b) the OC set, and (c) the RFE–24 set. Color codes are in Table 6.8.

6.2.2.1 Results

6.2.2.1.1 Las Vegas For this scene, the single panchromatic image can not correctly classify the eleven classes resulting in an overall accuracy of 44.42% with a Kappa coefficient of 0.321, as reported in Table 6.11. Only the classes Residential, Roads and Bare soil were recognized by the SVM, as shown in the classification map of Figure 6.22a.

The addition of morphological features improved the classification results for all the feature sets considered. Specifically, all the land-use classes were detected. The reconstruction filters resulted in an overall accuracy of 82.73% with a Kappa coefficient of 0.797. Even if improved results were obtained with respect to the panchromatic image for the class Commercial (93.02%), the class Parking Lots was substantially confused with roads (55.27%). Major confusion was also observed between the classes Soil and Residential buildings. The class Trees showed the smallest accuracy (55.17%), being confused with residential buildings, roads and vegetation. This can be explained by the fact that reconstruction filtering smoothes small structures whose reflectance is not sharply darker than the surrounding objects. Top-hat filters led to better global results. Residential buildings, Roads and Highway results were improved using these features, achieving an overall accuracy of 92.37% (Kappa coefficient: 0.911). Parking Lots accuracy improved by 25% because of the better separability obtained using large structuring element closing top hat filters. Surprisingly, the use of simple opening and closing filtering led to the best results

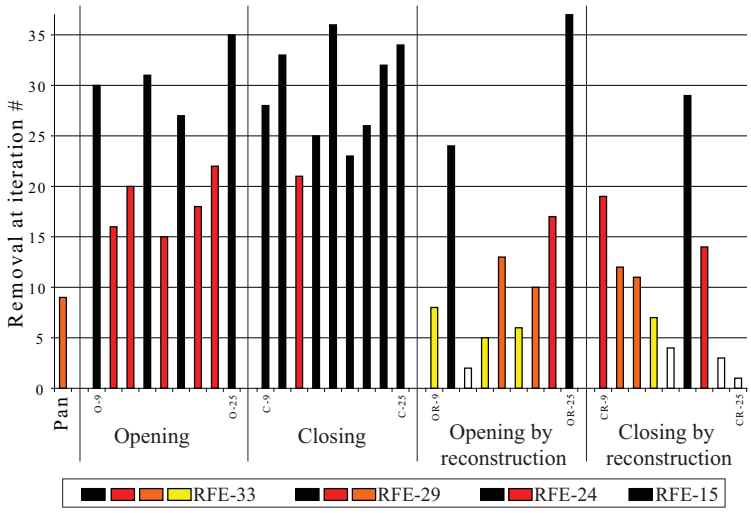


Figure 6.23: Feature selection by the RFE algorithm for the Las Vegas image. Each bar represents the iteration when the feature has been removed. White bars represent features removed during iterations 1-4; yellow bars during iterations 5-8; orange bars during iterations 9-13; red bars during iterations 14-22; black bars are the features maintained in each RFE result.

for the single filter-sets with an overall accuracy of 95.14% and a Kappa coefficient of 0.943. These simple filters allowed a significant improvement for the classes badly treated by the previous operators. Results for classes Short vegetation, Trees and Soil are improved by 10-15%, and the best result was reached for all the classes. The classification map obtained is shown in Figure 6.22b.

Combination of the sets (OC–TH, OCR–TH, OC–OCR and OC–OCR–TH) led to overall accuracies around 94–95%, equaling the OC results. The OC–OCR set showed the best performance, slightly improving the OC results for each class (the class Short vegetation is the only one showing a small decrease in accuracy). This set achieves an overall accuracy of 95.93% with Kappa coefficient of 0.952. The McNemar's test showed the improvement of this result with respect to all others. Summing up, adding the sets into a stacked vector improves the results (with best results obtained for the set combining OC and OCR features), but also added noise and redundancy that prevents the combination of improving significantly the solution found using the simple feature sets.

To reduce such a redundancy, RFE feature selection was applied. Four different stages of the feature selection were considered, accounting for different sizes of the final feature set. Solutions after removal of 4 (RFE–33), 8 (RFE–29), 13 (RFE–24) and 22 (RFE–15) were considered here. The RFE–24 feature set was chosen by taking into account the minimum test error criterion (see the blue curve in Figure 6.21a) and the RFE–15 was chosen by taking into account the maximum representation entropy (red curve in Figure 6.21a). As stated above, this last solution was optimal in terms of information compression. In general, none of the results obtained outperformed significantly the OC–OCR result (by the McNemar's test), even if small increases in the accuracies can be observed in Table 6.12. The SVM is already robust relative to the problems of dimensionality. Therefore, the benefits of RFE should be interpreted in terms of image compression more than in terms of improvement of the classification accuracy.

The sequential removal of features by RFE is illustrated in Figure 6.23. During the first iterations (RFE–33, RFE–29), only features from the OCR set were removed and in particular CR features with large structuring elements and OR features with small elements. By looking at these features, the changes between them are minor

Table 6.12: Classification accuracies (in percent) and Kappa coefficient for RFE experiments using the Las Vegas data set. In bold the results outperforming the OC–OCR set.

Class	OC-OCR	RFE–33	RFE–29	RFE–24	RFE–15	PCA
Residential	96.10	**96.15**	**96.82**	**96.71**	**97.36**	89.19
Commercial	97.41	97.38	97.13	97.10	96.52	**97.78**
Roads	97.11	97.06	**97.26**	**97.30**	**97.34**	95.19
Highway	**98.31**	98.25	98.14	98.30	97.94	92.67
Parking Lots	91.90	**92.03**	91.68	91.71	90.59	87.28
Short vegetation	92.02	**92.37**	**92.36**	**92.72**	91.22	82.84
Trees	87.26	**87.64**	**87.42**	**88.04**	84.77	74.42
Soil	89.68	**89.98**	88.68	89.09	86.59	84.92
Water	**94.79**	**94.79**	93.49	93.49	90.47	88.16
Drainage Channel	97.23	97.19	**97.78**	**97.48**	96.49	94.61
Bare soil	**99.52**	**99.52**	99.35	99.38	98.90	98.33
Overall Accuracy	95.93	**95.98**	**96.05**	**96.11**	95.67	90.10*
Kappa coefficient	0.952	**0.953**	**0.954**	**0.955**	0.949	0.901

* significant difference from the OC–OCR result by the McNemar test

Table 6.13: Classification accuracies (in percent) and Kappa coefficient for the Rome data set.

Class	Pan	OCR	TH	OC	OC-TH	OCR-TH	OC-OCR	OC-OCR-TH
Buildings	27.65	79.17	86.94	**88.89**	89.33	88.80	89.33	**89.62**
Blocks	0.00	66.97	66.70	**74.75**	69.48	76.66	**80.80**	77.55
Roads	54.61	83.79	83.11	**86.92**	84.89	88.12	**89.39**	88.29
Railway	0.01	**94.46**	76.40	85.52	79.48	93.34	**94.98**	93.37
Vegetation	0.00	67.83	70.17	**77.92**	74.59	83.28	**84.80**	83.38
Trees	48.49	48.29	72.29	**78.99**	74.74	77.52	**78.93**	78.59
Bare soil	**97.20**	84.69	89.95	92.18	90.85	94.33	**95.29**	94.60
Soil	0.01	70.00	74.03	**81.95**	77.49	82.53	**86.54**	83.88
Tower	0.00	**74.60**	58.04	66.86	59.94	70.26	**77.79**	70.03
Overall Accuracy	33.84*	74.21*	78.10*	**83.10***	80.46*	84.52*	**86.48**	85.05*
Kappa coefficient	0.229	0.694	0.738	**0.799**	0.767	0.816	**0.839**	0.822

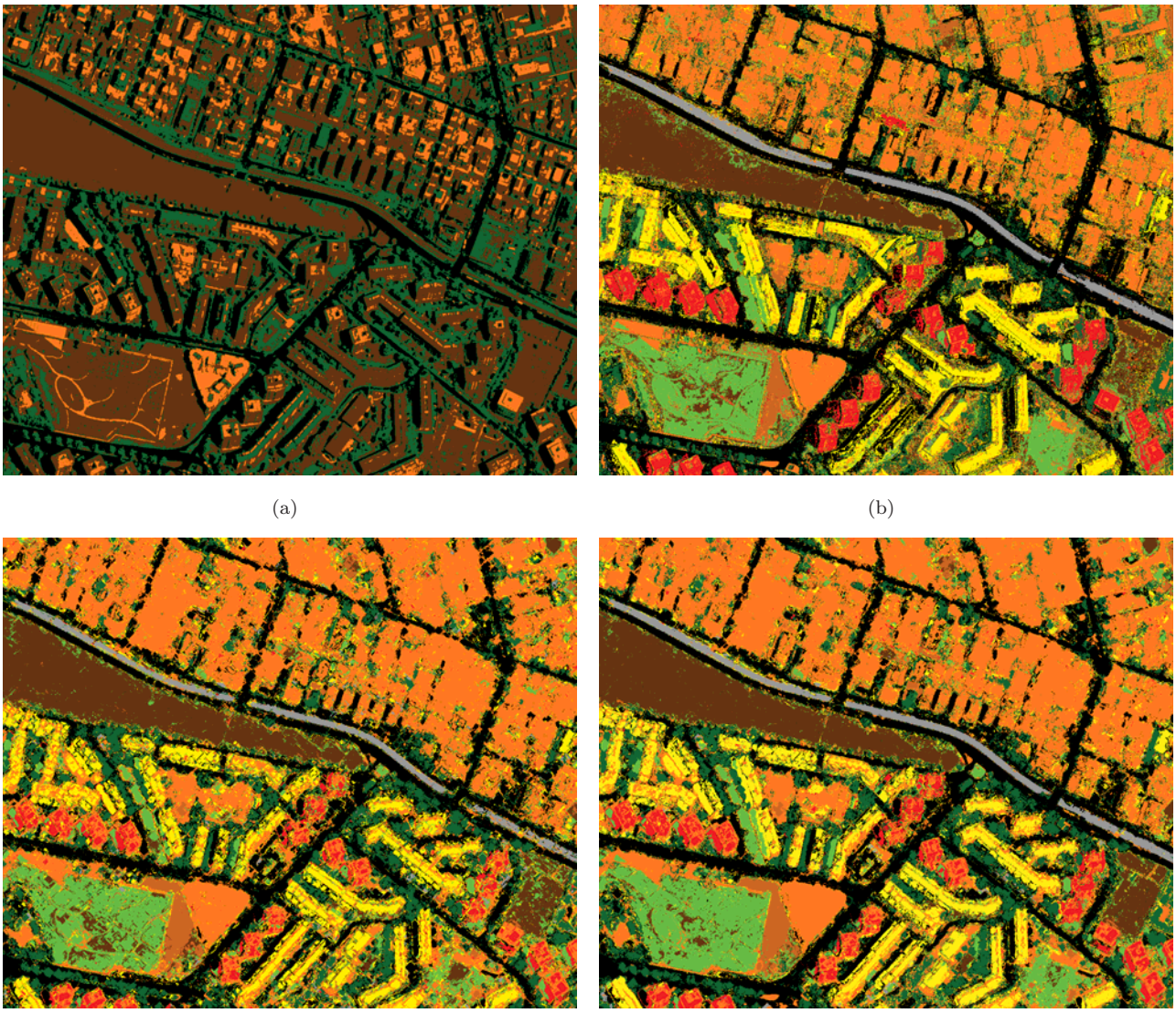
* significant difference from the OC–OCR result by the McNemar test

and their redundancy is strong. RFE–33 and RFE–29 columns in Table 6.12 show small improvements in the results of several classes.

Successively (iteration 9, RFE–24), the panchromatic image was removed, being the band with the higher spatial information (including noise). In this sense, O9 and C9 features were very similar to the panchromatic band and appeared as a smoothing filter. Moreover, RFE–24 showed the removal of OCR features related to small structuring elements. Thus, the algorithm selected OCR features adding information to the OC features. RFE–24 provided the best results, showing an overall accuracy of 96.11% with Kappa coefficient of 0.955.

At (RFE–15), OC features started to be removed. Figure 6.23 illustrates the removal of opening features (in red). From this figure it is possible to note that the features were removed at regular intervals and that features O–9, O–15, O–19 and O–25 were preserved. This removal can be interpreted as the redundancy between features extracted using overly similar structuring elements. In particular, a two-pixels step for the extraction of features appeared to be a short interval. Thus, only a small number of features carrying very different information was kept. Nonetheless, a decrease in the performance of the SVM was obtained (accuracy: 95.67%, Kappa coefficient: 0.949). The representation entropy was a criterion useful for information compression and did not account for classification accuracy. Therefore, the RFE–15 set represented the best compromise for the reduction of the size of the data set, even if it showed a small decrease in the performance. In fact, after the removal of 22 features, the SVM result was degraded only by 0.003 in terms of Kappa coefficient.

Regarding the computational burden, most of the computational complexity of the algorithm was taken by the SVM model selection. The generation of the morphological features took only a few seconds, while the SVM showed a complexity which was quadratic with respect to the number of examples. For the Las Vegas case and for



(a) (b)

(c) (d)

Figure 6.24: Classification maps for the Rome image using (a) only the panchromatic band, (b) OCR set, (c) OC set and (d) RFE–33. Color codes are in Table 6.9.

the most complex experiment (the OC–OCR–TH), such a calibration with 25 parameter sets took about 16 hours. The complete RFE relied on the evaluation of $Q \cdot (Q-1)$ SVMs. Nonetheless, the optimal model parameters of the OC–OCR set were kept, so that the model selection computational burden was avoided. Each iteration sped up the SVM evaluation because a feature was removed, but the overall computation burden still remained heavy. Note that, in this scenario, the definition of a task-based stopping criterion for the RFE becomes important.

6.2.2.1.2 Rome The results obtained for the Las Vegas scene were confirmed also for this second test case. In fact, using only panchromatic information did not distinguish the nine classes defined in the ground reference. This resulted in an overall accuracy of 33.84% with a Kappa coefficient of 0.229, as reported in Table 6.13. Only the classes Buildings, Roads, Trees and Bare soil were retained by the classifier. Nonetheless, the results for the panchromatic set provided the best results in terms of accuracy for the class Bare soil (97.20%). This is due to the fact that the model classified most of the pixels as Bare Soil, including the ones that actually were bare soil,

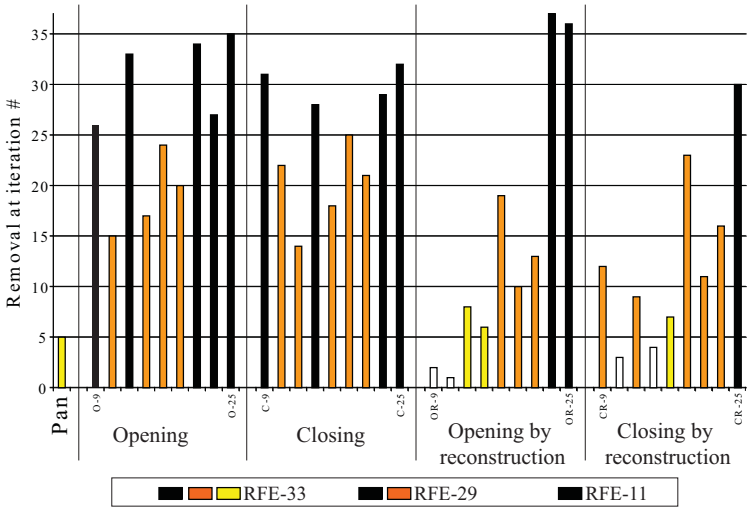


Figure 6.25: Feature selection by the RFE algorithm for the Rome image. Each bar represents the iteration when the feature has been removed. White bars represent features removed during iterations 1-4; yellow bars during iterations 5-8; orange bars during iterations 9-26; black bars are the features maintained in each result.

as shown in the classification map of Figure 6.24a. Since the overall accuracy criterion does not take into account commission errors, the accuracy for this class is really high.

Similarly to what was observed in the previous scene, the OCR set showed better results than the panchromatic experiment (accuracy of 74.21% and Kappa coefficient of 0.694). Moreover, the difference in overall performance with the TH experiment was smaller. This can be interpreted as follows. The OCR set provided the best performance for the classes Railway and Tower because the OCR features preserved the shape of the objects (see the classification map of Figure 6.24b). Specifically, for Towers (note that only this set can handle this class correctly), the reproduction of the shape was far more accurate than the one obtained using other sets (see the results for the class Tower for the OC and TH sets and the classification maps of Figure 6.24b and 6.24c). Nonetheless, looking at the per class accuracy of Table 6.13, the OCR model confirmed the poor performance for the classes Trees (accuracy 48.29%, strongly confused with Roads) and Vegetation. Moreover, OCR provided a more noisy solution than the OC set in the residential building areas. TH set provided overall better results (accuracy of 78.10% and a Kappa coefficient of 0.738). Even if the overall result was better for this data set, the TH set did not recognize the class Towers, poorly classified with 58.04% of accuracy. The best results for single sets were obtained again by the OC set (accuracy of 83.10% and Kappa coefficient of 0.799). The building area reconstruction was less noisy and the class Trees was slightly confounded with the class Road. Despite the higher accuracies, the lower half of the classification map of Figure 6.24c showed a less desirable result for Apartment blocks than the one obtained with the OCR set.

Mixed sets (OC–TH, OCR–TH, OC–OCR and OC–OCR–TH reported in Table 6.13) showed general improvement of the results obtained by the non-mixed sets. Again, OC–OCR sets provided the best results (overall accuracy: 86.48%, Kappa coefficient: 0.839), as for the Las Vegas scene. Results for all the classes were improved with respect to the results obtained by the single sets.

RFE feature selection is illustrated in Figure 6.25. Only three subsets were evaluated because the minimum for test error was achieved by the RFE–33 set, when 4 features were removed.

Table 6.14 shows the per-class and global accuracies for the Rome data set. At first glance, the feature selection did not improve significantly the classification accuracy for this data set. The best result achieved was provided

Table 6.14: Classification accuracies (in percent) and Kappa coefficient for RFE experiments using the Rome data set. In bold the results outperforming the OC–OCR set.

Class	OC-OCR	RFE–33	RFE–29	RFE–12	PCA
Buildings	89.33	**91.21**	**90.52**	87.90	70.82
Blocks	**80.80**	79.56	79.65	77.62	64.95
Roads	**89.39**	88.95	89.03	87.02	51.64
Railway	**94.98**	94.94	94.69	93.93	80.29
Vegetation	84.80	**85.26**	**85.48**	82.20	74.42
Trees	78.93	**80.26**	**79.98**	78.31	37.70
Bare soil	**95.29**	95.16	95.12	93.96	83.39
Soil	86.54	**86.58**	86.09	84.63	86.01
Tower	**77.79**	72.98	73.87	73.50	70.92
Overall Accuracy	86.48	**86.54**	86.43	84.43*	64.10*
Kappa coefficient	0.839	**0.840**	0.838	0.815	0.57

* significant difference from the OC–OCR result by the McNemar test

by the RFE–33 set, which achieved an overall accuracy of 86.54% with a related Kappa coefficient of 0.840. This can be explained by the good generalization capabilities of the SVM, being able to easily handle high-dimensional spaces up to several tens of dimensions. In terms of classification accuracy, the OC–OCR was already an optimal choice and the feature selection should be seen here as a way to rank the features in terms of information. Note that the RFE–33 and RFE–29 results, even if accounting for less features, were not significantly different to the OC–OCR result (McNemar's test).

As observed for the previous scene, most of the features removed during the first iteration were part of the OCR set. Opening and closing by reconstruction features related to small structuring elements were removed at the RFE–33 stage. For the Rome data set, the panchromatic image was removed at iteration 5 (RFE–29), showing again its redundancy with the OC features extracted using small structuring elements. CR features extracted using small structuring elements were removed rapidly, mainly because they highlighted small shadowed areas, smoothing the differences between other structures in the image. Finally, OR features were also rapidly removed. These features filtered small scale structures such as details in the roofs, leaving the main structures of the image unchanged. On the contrary, features with larger structuring elements took into account large scale structures such as entire buildings. These features showed higher variability and were more valuable for land-use classification because they provided the information necessary for the recognition of the towers.

The set selected by the representation entropy criterion (RFE–12) was the set resulting in the best compression rate. All the CR information was collected into a single feature, the CR–25. Also for the OR features only OR–23 and OR–25 were selected. Regarding the O and C features, the same scheme observed for the Las Vegas image is found: small, medium and large size structuring elements were selected and intermediate steps were removed from the data set. In terms of classification accuracy, a decrease of 2% of the overall accuracy was obtained. By the McNemar's test, the result was significantly smaller with respect to the OC–OCR result. By keeping only 12 features, the classification result still outperformed all the results obtained by the single sets.

Summing up, the same feature selection was observed, giving more importance to OC features and selecting them in regular structuring element size intervals. Few OCR features appeared to contain all the OCR information.

6.2.2.2 Summary

In this section, morphological features were used for the classification of land-use from panchromatic very high spatial resolution satellite images. Six types of filters were considered: opening, closing, opening by reconstruction, closing by reconstruction, opening top hat and closing top hat. Each of them highlighted a different aspect of the information contained in the image. Different spatial scales were considered in order to produce a classification

result dependent on the object size. Support vector machines were used for the classification phase of the morphological features. Moreover, RFE feature selection was exploited in order to decrease the dimensionality of the input information.

Two QuickBird images were exploited with spatial resolutionS of about 0.6 m, containing respectively 11 and 9 classes of land-use. In both experiments, the simple feature sets obtained with opening and closing operators showed the best classification accuracies. Nonetheless, each set of operators showed specific peculiarities that made the use of a mixed set suitable. The mixed set sharply improved the results, but could not fully take advantage of the added features for the classes where a single set failed. RFE feature selection was used to remove features that increased the redundancy,resulting in optimal sets of features either in terms of classification accuracy or data redundancy.

Even if characterized by large differences between urban structures, the same feature sets were found to be the most valuable for the classification of the images. Moreover, the RFE selection led to the same conclusions for both scenes in terms of importance of the features, showing the possibility to define a family of features which is optimal for the classification of land-use using very high spatial resolution panchromatic imagery.

At present, the method suffers two major drawbacks. First, in light of the high complexity of the problems, a consequent training set has to be provided to the machine. In these experiments, 30,000 and 50,000 pixels were used for training, which is a very large amount even if it represents only 5% of the available ground reference. In this sense, active learning methods (see Chapter 11) may provide a solution to this issue.

The second problem is related to the feature selection routine used. Even if resulting in good performance, RFE is a greedy method, whose computational cost heavily relies to the number of support vectors found by the SVM. A solution for this problem may be again a technique that selects the most relevant training samples. Otherwise, faster feature selection methods can be considered or designed [38].

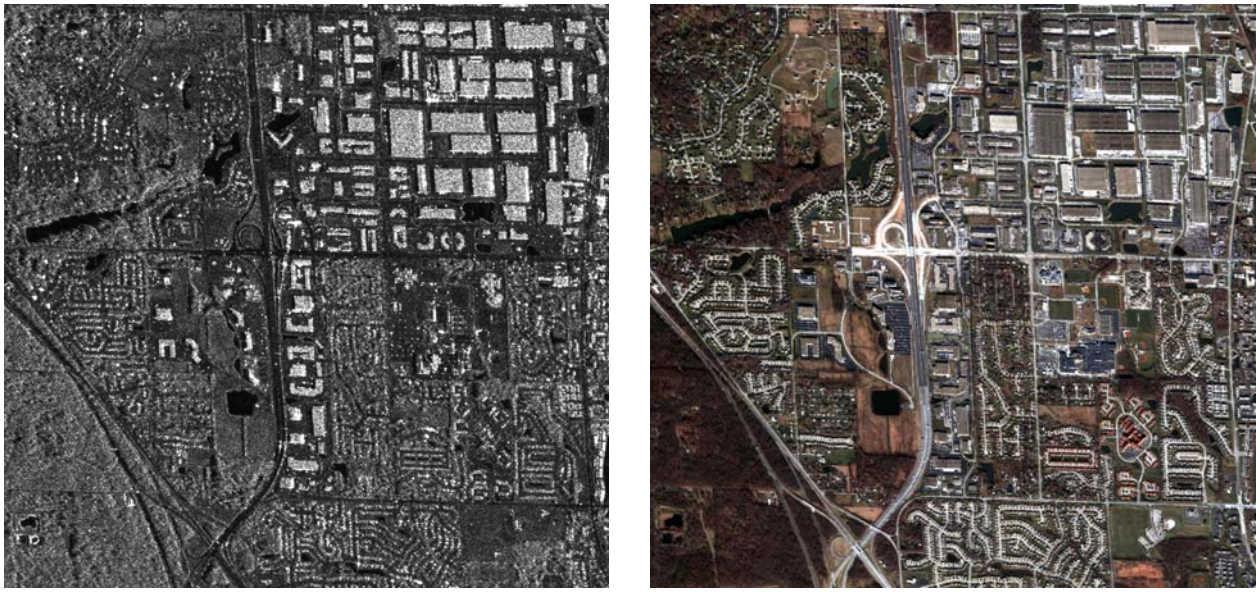
6.2.3 Morphology applied to very high spatial resolution X-band SAR imagery

Very high resolution synthetic aperture radar data available today from satellite represent a powerful tool to monitor urban areas. The availability of images with resolution comparable to that of airborne sensors, but acquired in a continuous and regular way from satellites, have dramatically increased the number of applications of remote sensing data in urban areas. The presence of more than one X-band constellation, such as TerraSAR-X and COSMO-SkyMed, will increase the revisit time of a scene.

Previous generations of SAR sensors, with decametric spatial resolution comparable with the dimension of single buildings, were capable of providing useful information about urban areas. In particular, these sensors were able to detect urban texture, building densities, main road directions and so on [109]. This information was useful for demographic and statistical analysis, local transport management, urban growth monitoring [89][110] and damage detection in urban areas caused by destructive events [111]. More details on the use of decametric spatial resolution SAR sensors for urban classification can be found in Chapter 7.

The advent of the new SAR satellites with sub-meter spatial resolution produced a radical change in this field, since the spatial details of the image are below the dimension of a single building or the dimension of typical target objects. However, they are still not fully exploited for urban classification due their complex interpretation, the presence of speckle [112] and the low information content in the backscattering intensity image for the characterization of objects having dimensions comparable to the geometric resolution.

As shown previously, morphological operators are useful for extracting object attributes related to their dimensions and geometry. These filters are widely exploited for very high spatial resolution optical images, but



(a) (b)

Figure 6.26: Sub-urban area of Indianapolis imaged by (a) TerraSAR-X and (b) QuickBird.

Table 6.15: Number of selected training and validation samples per class.

Classes		TR	VA	Color
BS	Bare soil	498	1,991	brown
IB	Industrial buildings	1,613	6,450	red
F	Forestry	484	1,935	dark green
G	Grass	743	2,971	light green
R	Roads	1,459	5,836	black
RH	Residential houses	231	922	yellow
W	Water	766	3,064	blue
Total pixels		**5,794**	**23,169**	

there is a lack in literature in the use mathematical morphology for the extraction of contextual information from very high spatial resolution SAR data. In this section, the extraction of contextual information from very high spatial resolution SAR backscattering data is discussed. Anisotropic morphological filters were applied to the backscattering image using a multi-scale approach. A range of different spatial domains was investigated using neural network pruning to analyze the different spatial characteristics.

The data set consists of one stripmap TerraSAR-X image taken on July 1, 2007 over the sub-urban area of Indianapolis (U. S. A.). This scene, shown in Figure 6.26a, is composed of roads, vegetated areas, including parks and forests, and structures with a variety of dimensions and architectures, such as residential housing, commercial buildings, utilities and industrial buildings. The image was acquired in both polarization, HH and VV, but only the HH polarization was used. The geometric resolution is 6 meters and the incidence angle is about 31°. A 2.4 m multi-spectral QuickBird image (shown in Figure 6.26b) taken on July 11, 2007, was used as ground reference to identify seven classes of land-use, including Industrial buildings, Water, Roads, Houses, Grass, Bare soil and Forestry. Training and validation samples (reported in Table 6.15) for the classification and validation phases were selected by visual inspection of the QuickBird image.

Section 6.2.3.1 introduces the anisotropic morphological filters. In Section 6.2.3.2, the results are analyzed and discussed. Final conclusions follow in Section 6.2.3.3.

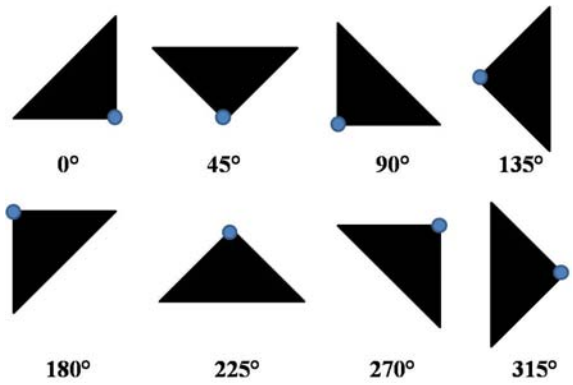


Figure 6.27: Eight different anisotropic triangular structuring elements.

6.2.3.1 Anisotropic morphological filters

The main characteristic of the filtering process provided by the opening and closing operators is that not all structures within the original image are recovered when these operators are subsequently applied. The size of the SE, with respect to the size of the structures actually shown in the scene, influences the output of the filtering operation. In fact, some structures in the images may have a high response for a given selected size and a lower response for other sizes. As discussed previously, in urban areas, the elements that compose the scene have different dimensions, so it is necessary to use SE with different sizes and shapes for characterizing the different objects.

Because structural classes from mathematical morphology depend on SE size, it is important to define the most suitable shape and dimension for classifying a certain type of target. The dimension of SE is directly related to pixel resolution and to the objects in the scene. In this section, an anisotropic SE with triangular shape was exploited. Anisotropic means that the filtering window is not centered on the center of the right triangle but in the right angle. Figure 6.27 illustrates the eight different triangles implemented, each one oriented in a particular direction.

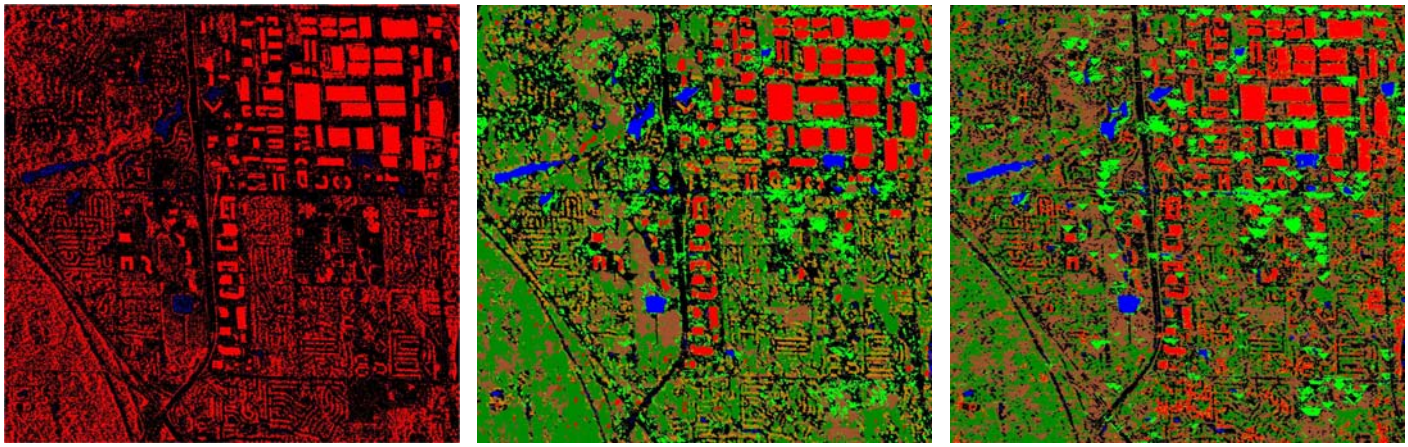
Eight different triangular SE with different sizes of hypotenuse, spanning from 3 to 41 pixels, were investigated, for a total of 321 inputs, composed of eight different morphological profiles (each morphological profile contains 40 bands), including the original intensity backscattering of the HH polarization. The triangular shape was chosen because buildings can be decomposed into this particular shape (i.e. they can be seen as composition of more than one of these triangles). The selection of the anisotropy characteristic of the filter was done to investigate if there are some distinctive directions which have more information content with respect to the others.

The morphological profiles obtained from the backscattering image are the input space of a multi-layer perceptron neural network. The analysis of the most effective morphological profiles was carried out by network pruning as described in Chapter 4.

6.2.3.2 Results

The use of only the backscattering information did not provide satisfactory accuracies for the classification of the land-use, as shown in Figure 6.28a where only a few classes were detected.

The pruning of the 321 profiles reduced the input space to only 50 bands, producing a land-use map, shown in Figure 6.28b, with a Kappa coefficient of 0.91 (the confusion matrix is reported in Table 6.16). Within all inputs, only two of them showed values of relative saliency close to one (maximum contribution), while all other values were close to zero and less than 0.2, as shown in Figure 6.29a.



(a) (b) (c)

Figure 6.28: Land-use map of Indianapolis using (a) only the backscattering information, (b) 50 input features and (c) the two most contributing input features. For the color classes see Table 6.15.

Table 6.16: Confusion Matrix using fifty most contributing inputs (%)

class	BS	IB	F	G	RH	R	W	Error
BS	83.83	0.0	1.14	3.47	1.51	0.33	0.13	8.15
IB	0.0	99.69	0.0	0.0	0.0	2.60	0.0	27.86
F	2.46	0.12	96.80	0.07	0.51	3.58	0.0	8.61
G	3.42	0.0	0.0	85.19	3.70	0.33	0.10	12.18
R	10.20	0.0	0.41	11.21	91.45	0.54	7.87	26.44
RH	0.10	0.19	1.65	0.07	0.02	92.62	0.0	3.90
W	0.0	0.0	0.0	0.0	2.81	0.0	91.91	12.86
Kappa coefficient = 0.92								

The two most important features were in the 225° direction, with the hypotenuses values of 5 and 35 and the saliency values of 0.8 and 1, respectively. Both were obtained using the closing filter. In Figure 6.29b is summarized the contribution of all the features, grouped for the eight different directions, and opening and closing filters. The closing filters gave an important contribution only in the 225° direction with the two most important inputs, while, in all other directions, they had no contributions. The 0° direction did not give any contribution, neither using closing nor opening filters. It is important to note that the direction 0° is very close to the shadow side of the objects, related to the look angle of the sensor.

Successively, only the two most contributive inputs were used again to classify the scene. The resulting map is shown in Figure 6.28c. The classification accuracy showed a small decrease in terms of Kappa coefficient (0.87). Analyzing the confusion matrix reported in Table 6.17, the capability of classifying small houses decreased drastically, being confused with industrial buildings. The percentage classification accuracy of small houses was reduced of about 50% with respect to the previous case. Therefore, it is evident that even if some inputs have small saliency values, they may be crucial for obtaining reliable land-use maps of complex urban scenes.

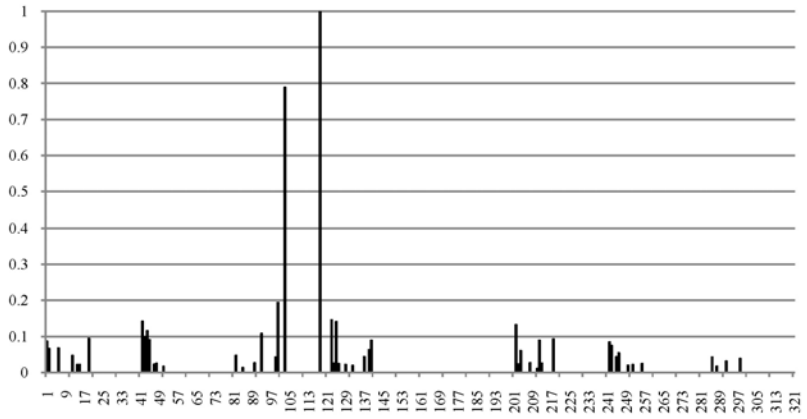
6.2.3.3 Summary

In this section, the use of anisotropic morphological filters with TerraSAR-X backscattering images for the classification of urban land-use was investigated. The anisotropic multi-scale analysis, coupled with the pruning network as a feature selection tool, was able to provide urban land-use maps with accuracies of about 0.90 in terms of Kappa coefficient.

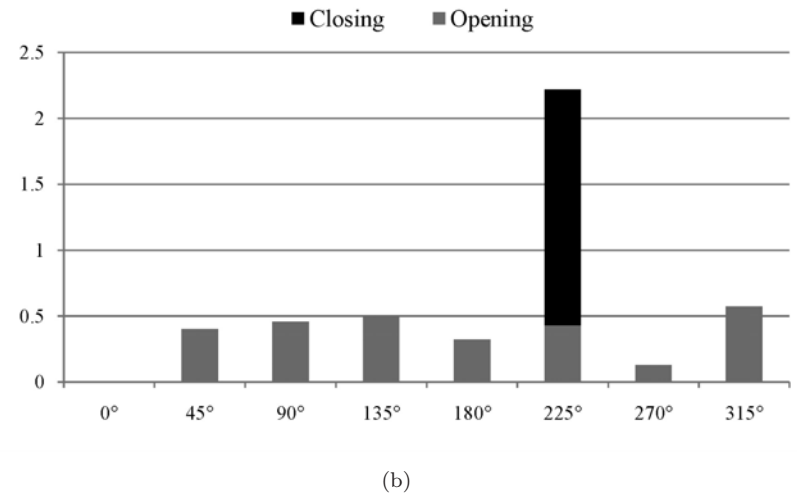
The analysis of the most effective morphological filters indicated that only a small number of these inputs

Table 6.17: Confusion Matrix using the two most contributing inputs (%)

class	BS	IB	F	G	RH	R	W	Error
BS	82.22	0.08	3.20	1.85	3.05	0.33	0.20	8.49
IB	0.0	96.45	1.45	0.0	0.02	49.24	0.0	28.94
F	2.61	0.43	93.95	0.64	0.84	7.48	0.0	8.78
G	1.00	0.0	0.0	84.32	2.66	0.0	0.16	11.59
R	13.16	0.0	0.21	13.19	92.22	0.22	4.77	26.71
RH	0.0	3.04	1.14	0.0	0.10	45.52	0.0	2.66
W	0.0	0.0	0.05	0.0	1.11	0.22	94.88	12.84
Kappa coefficient = 0.87								



(a)



(b)

Figure 6.29: (a) relative feature contribution of the input features not eliminated by the extended pruning, and (b) sum of the relative feature contribution values of each input with respect to the eight different directions (b).

contain valuable information. In particular, it was highlighted that only one direction (225°) was the most relevant in term of information content for this specific data set, being related to the look angle of the sensor.

6.3 Conclusions

In this chapter, the use of spatial information from optical panchromatic and SAR very high spatial resolution imagery was investigated to classify the land-use of urban environments. To overcome the multi-channel information deficit of single-band imagery, it was necessary to extract additional information to recognize objects within the

scene. Textural and morphological features were systematically investigated computing them over different sizes and directions.

A few remarkable outcomes are listed below:

- network pruning appeared to be necessary in a neural-net based classification, since a relevant increase in accuracy (Kappa coefficient ranged from 0.8 to more than 0.9) was observed with respect to a fully connected NN topology
- fist-order textural features, which do not contain any information on direction, yelled smaller contributions than second-order features. Dissimilarity appeared to be the dominant texture parameter. For the spatial resolutions and test cases considered, larger cell sizes, such as 31×31 and 51×51 pixels, had higher contributions than smaller cell sizes (regardless of the choice of textural parameters and directions), making it clear that there is a need to exploit the entire directional information
- the simple feature sets obtained with Opening and Closing operators showed the best classification accuracies with respect to the other morphological filters considered. Nonetheless, each set of operators showed specific characteristics that made the use of a mixed sets suitable
- only panchromatic information appeared not to be able to adequately separate classes whose pixels are covered by shadow. On the contrary, the multi-scale analysis carried out with textural or morphological features proved that it is possible to distinguish different shadowed areas
- the anisotropic multi-scale analysis (applied to SAR data) showed that only one direction (225°) was the most relevant in term of information content for this specific data set, being related to the look angle of the sensor
- textural features and NNs outperformed the results obtained with morphological filters and SVMs only for the Rome case (95.0% and 86.5%, respectively), and achived comparable accuracies (93.2% and 96.1%, respectively) for the Las Vegas area (which corresponds to a less complex urban scenario)

Chapter 7

Exploiting the temporal information

Part of this Chapter's contents is extracted from:

1. F. Del Frate, F. Pacifici and D. Solimini, "Monitoring urban land-cover in Rome, Italy, and its changes by single-polarization multi-temporal SAR images", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 1, no. 2, pp. 87-97, June 2008

In this chapter, the identification of a set of features derived from multi-temporal single-polarization synthetic aperture radar data is investigated. The C-band SAR images provided over the past decades by ERS-1 and ERS-2, and currently ENVISAT, are systematically available at relatively low price. Together with Landsat, they provide a long-term history of urban areas, hence their value should not be overlooked. In particular, the long ERS SAR image time series provides a unique, systematic means of periodically tracking, retrieving and understanding the frequently dramatic changes undergone by the land-cover of large cities in many parts of the world in the past 20 years.

Remote sensing in the optical band is a well established tool for producing maps of urban land-use and monitoring changes, but it can suffer from atmospheric limitations, especially where clouds systematically occur or when unpredictable, abnormally long periods of cloud cover affect normally clear-sky regions. Hence, when a systematic, timely and reliable survey of an urban area is required, the use of SAR imagery [113] might be suitable. Moreover, the management of emergencies over large areas relies on near-real time information, irrespective of the time of day and of the cloud cover: to this purpose the availability of SAR acquisitions is essential.

However, the ERS SAR data contain a minimum of information, having a single polarization. Moreover, due to the decametric size of the resolution cells on the ground, the shapes of the structures cannot be represented in detail and mixed pixels are likely to occur, especially in a sub-urban landscapes, where heterogeneous land-covers coexist over short distances. These limitations bound the data potential to fully identify the spatial features of land-cover. However, very high resolution images may prove sub-optimal in monitoring very large urban areas, both for the computational burden and for the unnecessary detail they depict. Indeed, for a global characterization of the dynamics of large areas over extended periods of time, images at decametric resolution may often turn out to be a useful compromise. Analogous considerations hold for the near-real time mapping for the management of emergencies eventually affecting large cities. In such circumstances, prompt, i.e. irrespective of cloud cover and time of day, information on the global land-cover evolution over large (e.g., hundreds of square kilometers) areas can be crucial to the decisional process.

The method proposed here exploits three different partially independent sources of information generated from a limited number of SAR acquisitions. This approach might be required in particular events affecting an urban

area or to improve the cost-efficiency of the data base. Many studies of SAR land-cover classification did not consider more than two types of image features at the same time [89][114].

The backscattering intensity, its textural properties, and the interferometric coherence were identified as the set of features containing the pieces of information embedded in both the amplitude and the phase of the scattered wave, with a minimum of two SAR acquisitions (the minimum needed to generate an interferometric image). The *short-term* classification (days or possibly less) utilized 4 quantities extracted from the image pair, namely the amplitude of the backscattering coefficient, two textural parameters and the degree of coherence. For the *long-term* scheme (inter-annual), the information on the seasonal variations of backscattering and of coherence was added. A minimum set of five late winter/early summer SAR scenes turned out to be suitable for attaining the desired classification accuracy. In this case, 6 quantities relative to three types of image features were exploited: the amplitude of the backscattering coefficient averaged over the five images and its standard deviation, the two degrees of interferometric coherence of two pairs of seasonal images, and two backscattering textural parameters.

The decision-making process was performed by a multi-layer perceptron NN classifier [15]. This algorithm satisfied the requirements mentioned above, since, once trained, it runs in real time and has considerable ease in using multi-domain data sources. Several studies appeared in the literature dealt with classification of SAR images using neural networks. They mainly refer to agricultural or forest studies. In particular, Chen and Mcnairn [115] reported on rice monitoring using RADARSAT-1, while Gimeno *et al.* [116] investigated burnt areas in the Mediterranean region using ERS-2 SAR time series. In both cases, the neural algorithm showed an overall classification accuracy over 90%. The use of SAR data in land-cover classification of urban areas is relatively limited, given the complex imaging geometry, the interactions between urban features and radar waves and the presence of speckle noise. These effects make it generally difficult to attain high classification accuracies by using single-channel single-polarization SAR images. To this end, neural network approaches, making use of multi-scale textural parameters [89] or backscattering temporal variations and long-term coherence [114] were proposed.

The rest of this chapter is organized as follows. The data set is described in Section 7.1. Section 7.2 deals with the extraction of features from multi-temporal imagery. The design of the neural network is discussed in Section 7.3. Experimental results of the classification phase are reported and accuracies are analyzed in Section 7.4 and Section 7.5. Discussion and conclusions follow in Section 7.6.

7.1 Data set

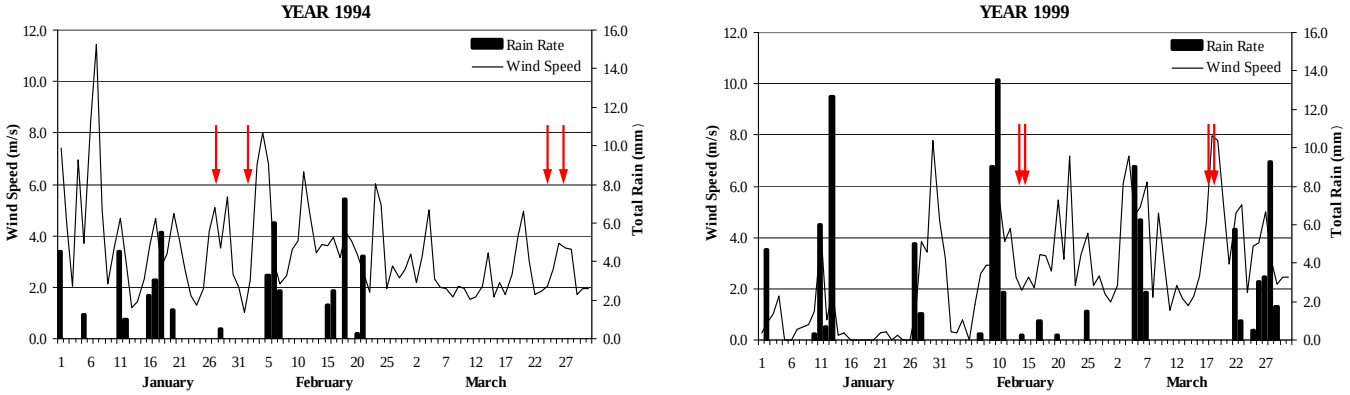
The study area included the city of Rome, Italy, and its outskirts for an overall extension of about 836 square kilometers (992×995 pixels). The data set was composed of Single Look Complex (SLC) SAR images acquired in winter, early spring and early summer by ERS-1 in 1994 and by the ERS-1/2 tandem mission in 1999, with 5 acquisitions each year as reported in Table 7.1.

Since meteorological and climatic aspects may be relevant for understanding the involved physical phenomena, the precipitation and wind speed data recorded by Aeronautica Militare Italiana at the Ciampino Airport (South-East Rome) were acquired. As shown in Figure 7.1, a light rainy period preceded the first winter ERS acquisitions in the years 1994 and 1999, while no precipitation was recorded immediately before the March acquisitions. However, the measured values of backscattering and coherence did not appear to be affected by the recorded meteorological conditions.

For the long-term classification, each set of 5 images indicated in Table 7.1 yielded the classified map for the corresponding year, while only the two images acquired in March of each year were used in the near-real time

Table 7.1: Data set relative to years 1994 and 1999. B_p refers to the perpendicular component of the baseline.

Acquisition Date	Satellite	B_p (m)
January 25, 1994	ERS 1	89
January 31, 1994	ERS 1	
March 26, 1994	ERS 1	157
March 29, 1994	ERS 1	
July 13, 1994	ERS 1	-
February 13, 1999	ERS 1	211
February 14, 1999	ERS 2	
March 20, 1999	ERS 1	65
March 21, 1999	ERS 2	
July 4, 1999	ERS 2	-



(a) (b)

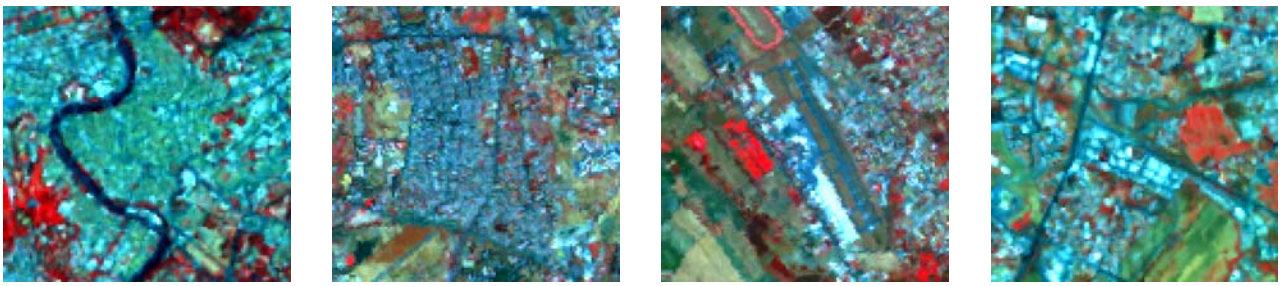
Figure 7.1: Daily average wind speed (light line) and total rain (histogram) recorded at Ciampino Airport from (a) January 1994 to March 1994, and (b) January 1999 to March 1999. The red arrows indicate the acquisition date of the SAR images.

classification exercise.

7.2 The classification problem

A systematic subdivision of land-cover types is proposed in the CORINE project of the European Environment Agency [117]. The basic land-covers (Level 1) include 4 classes (artificial surfaces, agricultural areas, forests and wetlands), while Level 2 refers to 13 cover types, including also mine and dump sites, pasture areas and costal wetland. Given the peculiarities of the Rome urban area, the purpose of this study, and the use of images at decametric resolution, 7 land-cover classes were chosen, being more suitable to urban analysis, including water surfaces (WS), vegetation (VE), arboreous (FO), asphalt/concrete (AS), industrial/commercial buildings (IB) and high/low density continuous urban fabric (HD/LD). Examples of some of these classes imaged in false colors composite (Bands 431) are shown in Figure 7.2. The high density urban fabric is mainly found in the oldest (middle age to 18^{th} century) sections of the city, while the low density urban fabric is typical of mixed areas with small buildings, gardens and narrow streets with trees, mainly developed in the first half of the 20^{th} century. Asphalted (and concrete) surfaces correspond to large roads, parking lots and airport runways. More recent isolated large residential, commercial or industrial buildings are found on the borders of the city.

Classification of urban areas by SAR data in the literature refers to only four [114] or five [89] classes. The intent of this classification exercise is to discriminate among the above 7 land-cover classes utilizing the information stored in the SAR acquisitions. Hence, those image features carrying effective information need to be identified.



(a) (b) (c) (d)

Figure 7.2: Landsat false colors composite (Bands 431) of (a) high density continuous urban fabric, (b) low density continuous urban fabric, (c) asphalt/concrete surfaces and (d) industrial/commercial buildings.

The rationale for the choice of the features is discussed in the following.

The first two inputs to the long-term classification algorithm were the mean and the standard deviation of the backscattering coefficient σ^0 computed for each pixel over the multi-temporal (winter, spring, and summer) data set. Only the single-date (spring) intensity was used in the short-term classification algorithm.

Urban backscattering typically reaches high values when resulting from single, double-bounce and trihedral reflections from relatively large man-made plane surfaces. The imaging geometry and the structure of the built up area involves the observation azimuth angle and the orientations of buildings, and can produce different backscattering values [118]. However, the backscattering from man-made structures is only partially sensitive to the different seasons of the year. In contrast, the backscattering intensity of natural (parks) and agricultural areas, which include bare soil and surfaces with trees and low vegetation, may vary significantly with the seasons, according to the changing geometric (growth, blooming stage and farming activities) and dielectric (moisture) conditions. Hence, when the near-real time response is not required, the seasonal behavior of the backscattering intensity can also be exploited.

The winter and late spring/summer interferometric short-term coherence (one day repeat-pass in 1999 or three days in 1994) were considered as the third and the fourth input of the long-term classification scheme, while only the spring value was used for the short-term classification.

The degree of interferometric coherence γ [119] is an indicator of both the temporal and the spatial phase stability of a target according to its geometrical and dielectric properties. The degree of coherence depends on sensor parameters, such as wavelength and system noise, imaging geometry, such as baseline and look angle, and target features. Moreover, volume scattering and temporal changes contribute to vary the coherence which results in the product of independent factors mainly related to the time delay between image pairs ($\gamma_{temporal}$), the difference in signals between images due to different positions in space ($\gamma_{spatial}$) and other factors (γ_{system}) which arise during the data acquisition (e.g., thermal noise or different atmospheric path delays) and processing (e.g., imperfect image registration.). Therefore, the degree of coherence can be estimated by taking into account these different correlation factors as long as the sources of correlation are statistically independent:

$$\gamma = \gamma_{temporal}\gamma_{spatial}\gamma_{system} \tag{7.2.1}$$

As reported in [120], the perpendicular baseline has a large influence on the coherence values for given surface height standard deviation, look angle, wavelength and sensor-target distance:

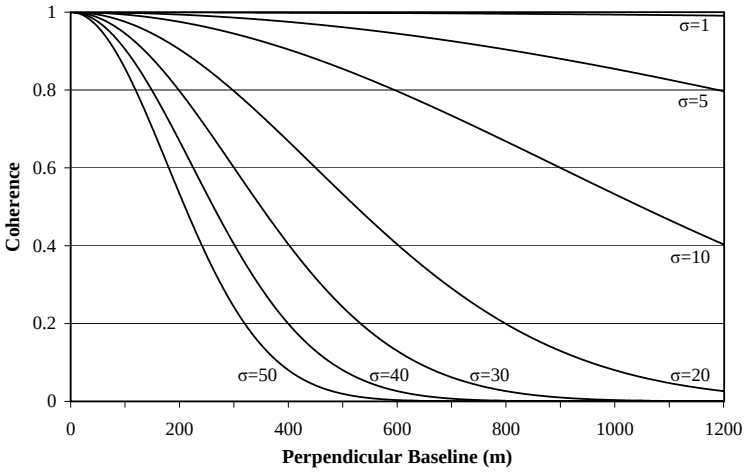


Figure 7.3: Theoretical relationship between coherence and perpendicular baseline for a range of buildings height variance values (σ).

$$\gamma = \exp\left[-\frac{1}{2}\left(\frac{4\pi\sigma\sin(\theta)B_p}{\lambda r}\right)^2\right] \tag{7.2.2}$$

where σ is the surface height standard deviation, θ is the look angle, B_p is the perpendicular baseline, λ is the wavelength and r is the sensor-target distance. In areas containing large buildings or small residential houses (for which the surface roughness has values of σ in the range 30 to 50 m and 10 to 20 m, respectively), coherence drops off more rapidly with increasing baseline than for flat bare surfaces, such as parking lots or wide roads as shown in Figre 7.3.

Several authors have [120][121][122][123][124] exploited coherence for land-cover discrimination. In particular, Bruzzone *et al.* [114] found that the coherence features proved to increase the classification accuracy by more than 16% when added to a multi-temporal data set. They pointed out the effectiveness of coherence to significantly reduce the confusion between both forest and urban areas, and agricultural fields and urban areas. In fact, in high density urban environments, coherence remains high even for imagery characterized by long-scale time given the phase stability of man-made structures, while low density residential areas exhibit lower coherence because gardens and parks can cover a considerable portion of the surface. In fact, vegetated surfaces are significantly influenced by temporal decorrelation and lose coherence within few days (or weeks) due to growth, movement of scatterers, harvest and changing moisture conditions.

Diverse studies demonstrated the importance of the long-term coherence with respect to the short-term coherence, showing the higher accuracy in classifying urban land-cover. In fact, for long acquisition time intervals, stable permanent scatterers show high coherence values: in temperate regions, buildings and man-made structures are almost exclusively stable targets, as reported in [124]. However, the use of long-term coherence is not appropriate for early (near-real time, ideally) operations. In the following, the analysis of the accuracy attainable by including only one or two short-term coherence images into the classification algorithm is carried out. In this case, only single-pass or short-term measurements, as the one reported in [125] or those foreseen by the new generation satellite constellations, are usable.

The last two inputs to the algorithm were computed from textural features of the intensity image. In particular, GLCM [76] were used to characterize the stochastic properties of the spatial distribution of gray-levels in the intensity images. As shown in the previous Chapter 6, in their investigation, Baraldi and Parmiggiani [81] concluded

Table 7.2: Number of SAR images, inputs and parameters used for the long- and short-term scheme.

Scheme	SAR Images	Inputs	Parameters
Long-term	5	6	Mean Int.
			Int. St. Dev.
			Winter Coh.
			Spring Coh.
			Contrast
			Energy
Short-term	2	4	Mean Int.
			-
			-
			Spring Coh.
			Contrast
			Energy

Table 7.3: Performance of different topologies on the 1994 test set.

Topology	Overall Error (%)	St. Deviation
6-12-12-9	21.3	1.08
6-22-22-9	19.3	2.43
6-28-28-9	13.1	1.96
6-40-40-9	11.6	1.59
6-60-60-9	7.1	0.73
6-80-80-9	6.5	1.29
6-100-100-9	6.1	0.79

that Energy and Contrast are the most significant parameters to discriminate between different textural patterns. Here, two different methods (reported in [76] and [126]) were used to generate these two textural features. Moreover, two additional textural parameters were considered to further investigate textural features, the Large Number Emphasis and the Second Moment suggested by [127]. These six textural features were computed using different window sizes (7×7, 11×11 and 15×15 pixels) and gray-levels (16, 32, 64), thus generating a total of 54 texture images. The class separability was subsequently computed for the textural images based on the Wilk's lambda as reported in [128]. The two parameters which yielded the maximum value were Energy and Contrast (consistent with [76]), with a window size of 7×7 pixels and 16 gray levels. In particular, Energy appeared to be valuable especially in separating high density from low density residential areas and asphalt from water, while Contrast contributed to solving the ambiguity between low vegetation and low density residential housing.

To summarize, the classification scheme exploited averages and standard deviations of backscattering intensity, coherence and average textural parameters computed by formulas known in the literature, e.g., [114] and [76]. In Table 7.2 is reported the number of SAR images and the parameters used for the short-term and the long-term classification schemes, while an example of data input relative to 1994 is illustrated in Figure 7.4.

7.3 Neural network design

As discussed, three different sources of information were used, contributing heterogeneous inputs to the classification algorithm. The classification algorithm is based on a multi-layer perceptron network with two hidden layers. Several different classification accuracies resulted from a varying number of hidden neurons, starting from a small topology (6-12-12-9) to end with a large one (6-100-100-9), as reported in the example for year 1994 in Table 7.3. The variance of the accuracy for different initializations of the weights was also computed to monitor the stability of the algorithm. The configuration that maximized the accuracy in each year without instability was retained.

A pruning procedure was then applied to thin the net, taking advantage of the reduction of runtime and memory, and generalization capability [129]. In particular, the fully connected NN was pruned with the MB

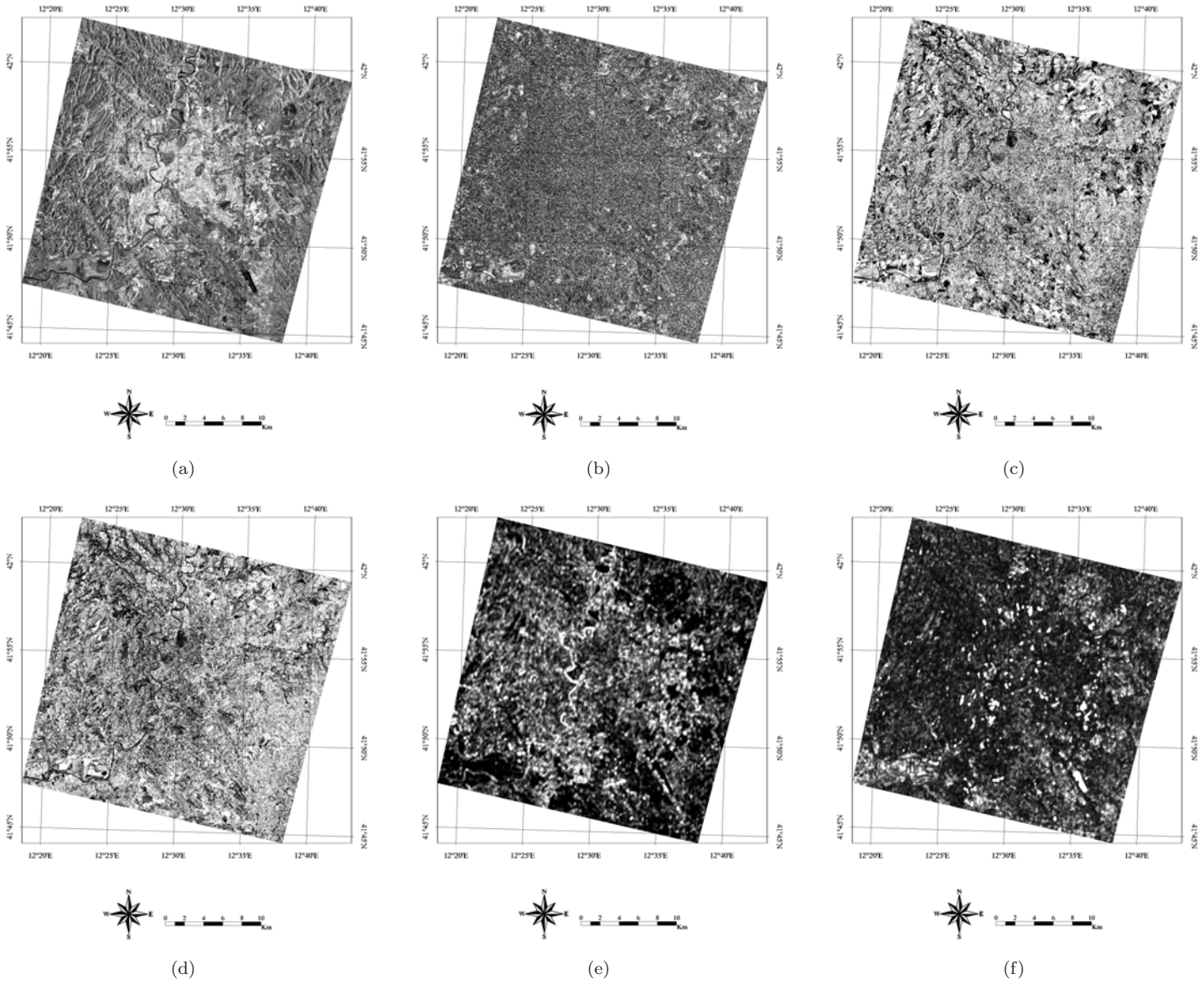


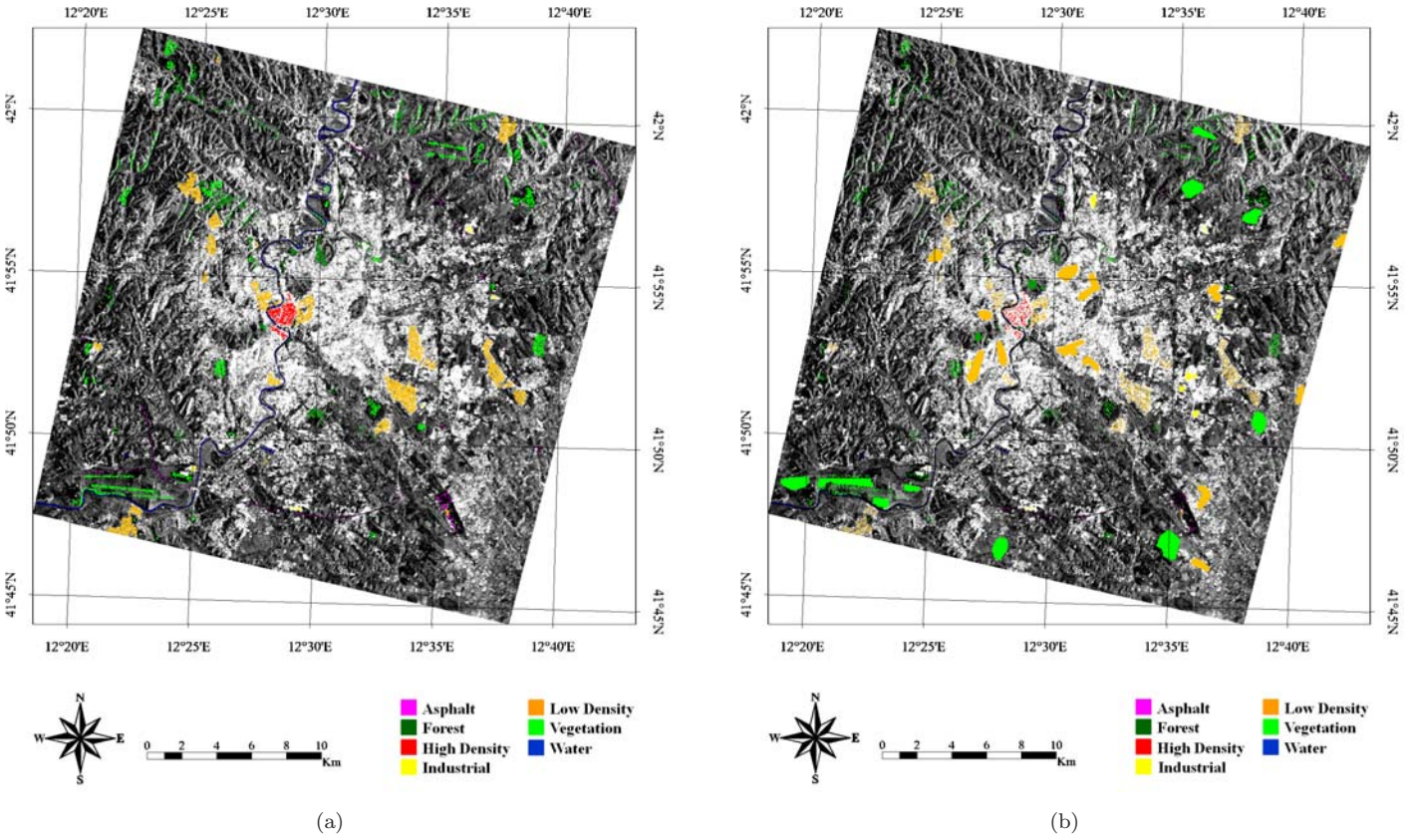
Figure 7.4: The six features used as input of the long-term classification algorithm: (a) Mean Intensity, (b) Intensity Standard Deviation, (c) Winter Coherence, (d) Spring Coherence, (e) Contrast and (f) Energy for year 1994.

pruning [54]. Training of the algorithm was carried out by the scaled conjugate gradient method [23].

The selection of pixels both for the training and validation phase was particularly critical. On one side, the training process impacts on the performance of the algorithm and, on the other, a biased test set can lead to fallacious evaluation of the results. The training samples were mainly selected by visual inspection of co-registered optical imagery taken by the Landsat 5/7 satellites in 1991 and in 2001, a time range which encompasses the dates of the radar images. Moreover, subsequent very high spatial resolution (QuickBird) multi-spectral imagery and *in situ* inspections were used to identify or validate ambiguous ground references. Since the area of the different surface types varied considerably (e.g., water is much less abundant than urban fabric), particular care was exerted in including a balanced number of pixels belonging to the each class. Stratified random sampling, which allows the inclusion of samples also of the less likely classes, was used to ensure a balanced representation of all classes. Table 7.4 reports the number of pixels of each class that were selected for the training and validation

Table 7.4: Number of selected training and validation samples per class.

	Output Classes	TR	VA
1	Asphalt/Concrete (AS)	511	219
2	Forest (FO)	2,592	1,326
3	High Density (HD)	648	278
4	Industrial Buildings (IB)	122	433
5	Low Density (LD)	4,900	6,130
6	Vegetation (VE)	3,535	6,008
7	Water Surfaces (WS)	892	382
	Total pixels	**13,200**	**14,776**



(a) (b)

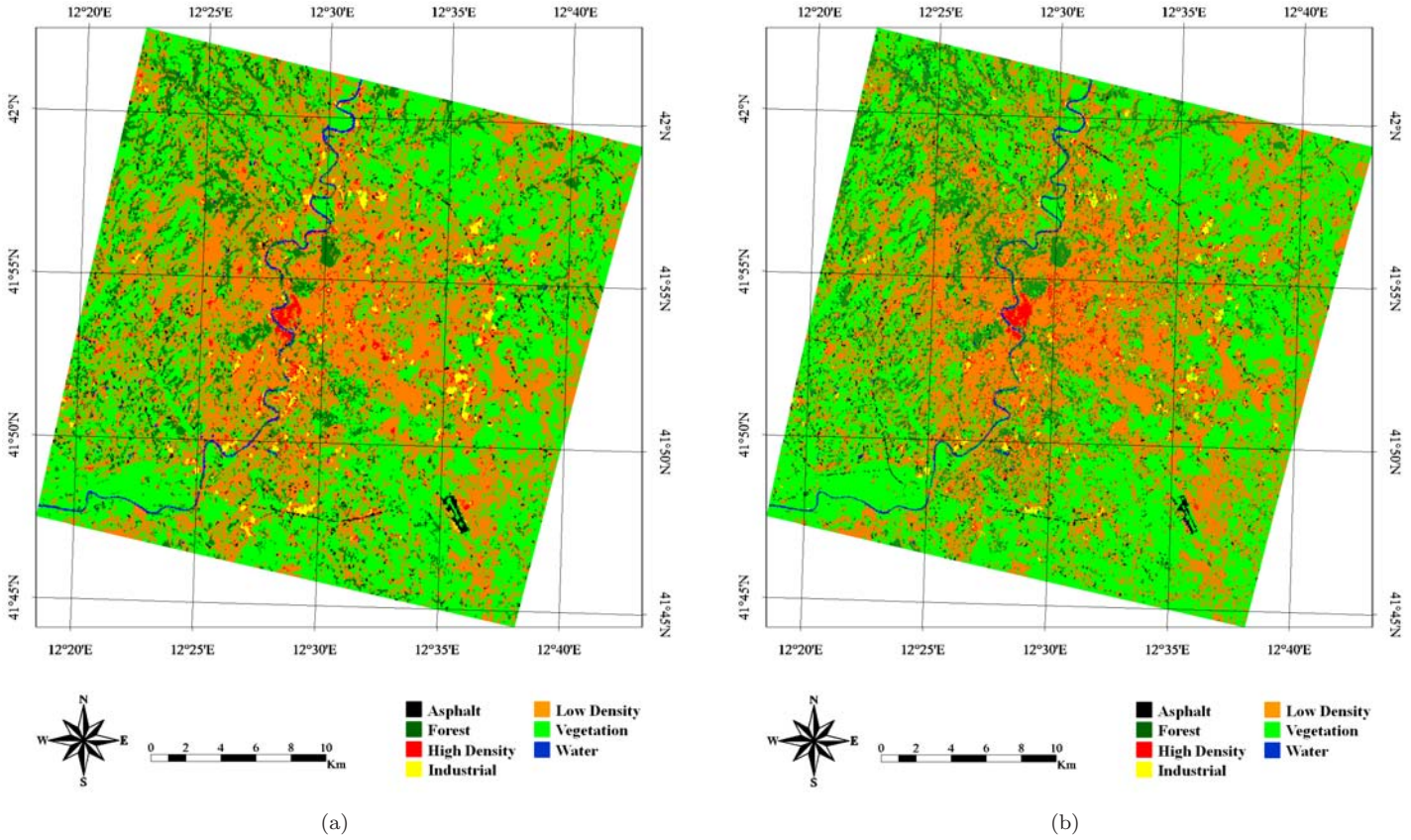
Figure 7.5: Training (a) and validation (b) sets superimposed to the Mean Intensity image of 1999.

sets (illustrated in Figure 7.5). The neural nets were trained by 13,200 total samples and the classification accuracy evaluated on 14,776 samples selected independently from the training ones.

To further investigate the influence of the number of both training and test pixels on the results, sets of different size were considered. In addition, the sets were exchanged to use the validation set as training and vice versa. These results are reported in the next section. Both short-term and long-term classifications were carried out with algorithms trained and tested by the largest sets, while the effect of the training and test sizes was studied for the long-term case.

7.4 Results

The SAR images acquired in 1994 and 1999 were processed using NNs to obtain the classification of the Rome land-cover. The 1994 and 1999 land-cover maps provided by the long-term algorithm are shown in Figure 7.6(a)

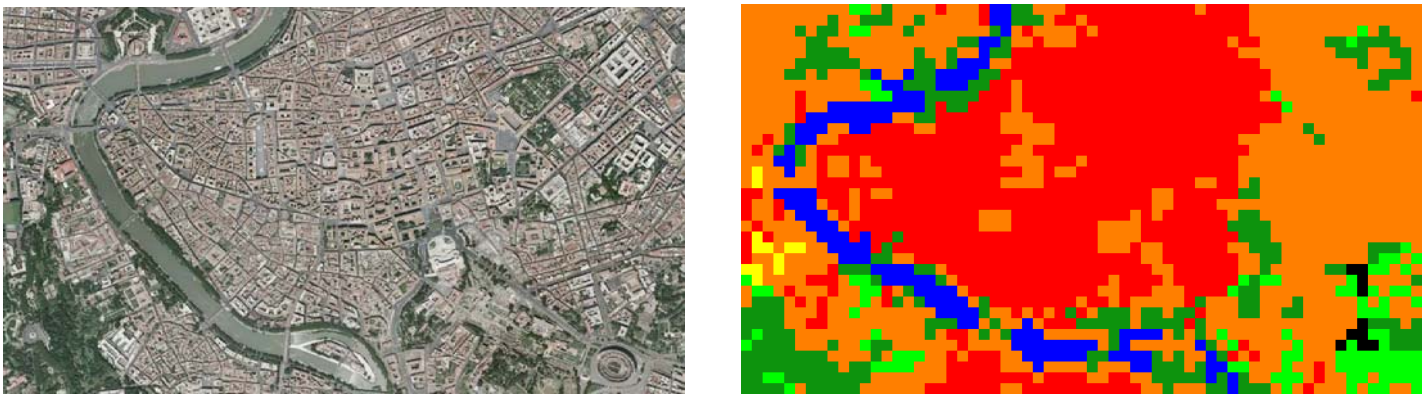


(a) (b)

Figure 7.6: Long-term classification map for year (a) 1994 and (b) 1999. The main large built area was identified with good accuracy, as well as some specific structures, such as the compact old part of the city, observable in red in the central part of the image, the Tiber river, the Ciampino airport (near the bottom-right corner) and the parks inside the city.

and Figure 7.6(b), respectively. In both cases, the main large built up area was identified, as well as some specific structures, such as the compact old part of the city, observable in red in the central part of the images, the Tiber river, the Ciampino airport (in black near the bottom-right corner) and the parks inside the city. Figure 7.7 displays a detail of the old part of the city relative to the long-term classification map for year 1999. In spite of the decametric resolution of the SAR acquisitions, several features, even of relatively small dimensions, were captured. These include the trees along the river and within the *Quirinale gardens* (located in the upper-right of the image), areas with lawns and plants such as *Piazza Venezia*, *Via dei Fori Imperiali*, *Foro Romano* and *Colle Oppio* (all located in the lower-right part of the image) and squares such as *Piazza Navona* and *Torre Argentina* (both correctly classified as low-density residential areas in the middle of the image).

The accuracy of the short-term classification, utilizing only the two March images, ranges from 92.0% (Kappa coefficient: 0.86) for 1994 to 89.3% (Kappa coefficient: 0.83) for 1999. The addition of the seasonal information from a second interferometric pair and a fifth single image increases the respective accuracies of the (long-term) classifications to 96.0% (Kappa coefficient: 0.92) for 1994 and 94.0% (Kappa coefficient: 0.91) for 1999. The detailed performance of the algorithm in discriminating the considered types of land-cover can be appreciated in the confusion matrix shown for the 1994 short-term case in Table 7.5. As expected from physical considerations, the classification errors mainly consisted in misclassification between high- and low-density continuous urban areas, between asphalt/concrete and low vegetation, and between parks and low-density residential areas or low



(a) (b)

Figure 7.7: Detail of the Rome historical area: (a) optical image and (b) long-term classified image of year 1999.

Table 7.5: Short-term classification confusion matrix for year 1994.

classes	AS	FO	HD	IB	LD	VE	WS
AS	74.18	2.75	0.00	0.00	0.00	19.23	3.85
FO	0.00	81.39	0.00	0.00	16.71	10.05	1.49
HD	0.00	0.00	36.32	1.42	61.32	0.94	0.00
IB	0.00	0.00	1.12	96.21	2.68	0.00	0.00
LD	0.07	0.95	0.23	0.11	96.99	1.65	0.00
VE	0.39	1.08	0.00	0.00	7.66	93.50	0.01
WS	1.22	1.52	0.00	0.00	0.30	0.30	93.92
Overall Error = 8.12%				**Kappa coefficient = 0.863**			

Table 7.6: Long-term classification confusion matrix for year 1994.

class	AS	FO	HD	IB	LD	VE	WS
AS	94.51	0.00	0.00	0.55	0.55	4.40	0.00
FO	0.00	93.07	0.00	0.00	2.85	3.80	0.27
HD	0.00	0.00	87.74	0.00	11.32	0.94	0.00
IB	0.00	0.00	3.35	93.75	2.68	0.22	0.00
LD	0.00	0.34	1.67	0.08	95.39	2.52	0.00
VE	0.43	2.47	0.00	0.15	1.80	95.16	0.00
WS	0.00	0.30	0.00	0.00	0.00	0.00	96.96
Overall Error = 4.01%				**Kappa coefficient = 0.923**			

vegetation. From the confusion matrix in Table 7.6 relative to the 1994 long-term case, it can be noted that the classification accuracy improved when six parameters were used rather than four, i.e., when including winter coherence and standard deviation of the backscattering coefficient. This could be expected, since the long-term classification algorithm exploited a richer input data set. Moreover, in this case the amplitude and texture parameters were averaged over five images rather than over just two, resulting in a more stable estimation of their mean values. Hence, the increase of accuracy can be seen as a consequence of the combined effect of adding another coherence image, another amplitude image and of the averaging. As before, misclassification mainly occurred between high-density and low-density urban regions, but misclassification between parks and low-density residential areas, as well as between asphalt/concrete and low vegetation, appeared reduced.

The figures vary slightly when reducing the number of pixels in the training and test sets. The 1994 long-term classification reaches an accuracy of 95.7% (Kappa coefficient: 0.94) when trained on 3,507 and tested on 5,733 samples; exchanging the sets yields 93.2% (Kappa coefficient: 0.91). Analogously, training on 8,182 and testing on 13,376 samples lowers the 1999 long-term classification accuracy to 93.5% (Kappa coefficient: 0.91); exchanging the sets yields 92.2% (Kappa coefficient: 0.89). Taken into account the variability of the surface from year to year,

Table 7.7: Relevance of the six features per single class.

	AS	FO	HD	IB	LD	VE	WS
Mean Int.	0.113	0.156	0.111	0.190	0.210	0.073	0.147
Int. St. Dev.	0.062	0.087	0.059	0.106	0.122	0.037	0.080
Winter Coh.	0.070	0.097	0.070	0.117	0.128	0.047	0.092
Spring Coh.	0.074	0.100	0.075	0.123	0.131	0.048	0.096
Contr.	0.083	0.114	0.080	0.137	0.155	0.054	0.108
Energy	0.096	0.129	0.095	0.161	0.176	0.063	0.126

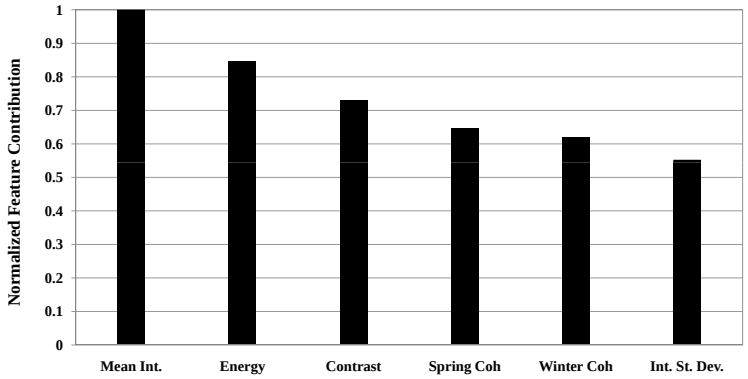


Figure 7.8: Normalized feature contribution of the six inputs.

which ensues from the single seasonal evolution and from the meteorological conditions in the days preceding the SAR acquisitions or during them, the above figures seems to confirm the essentially consistent performance of the algorithm, within its expected numerical fluctuations.

An interesting point is the assessment of the relevance of the six partially independent channels in the classification scheme. The feature contribution per single class is reported in Table 7.7. Globally, the backscattering intensity carried the maximum information, followed by energy, contrast and the two short-term coherence features, while the standard deviation of intensity contributed the least as shown in Figure 7.8. This result seems to confirm previous results on urban classification from multi-temporal SAR data [114], where the authors reported a classification accuracy of 65% using only temporally filtered images which rose to about 81% when the long-term coherence was added. Hence, the contribution of coherence is important, since it increases the overall classification accuracy, but the other features are also quite effective. This result was obtained for the particular urban scenario, land-cover classes and SAR acquisitions. However, it suggests that backscattering amplitude and texture might contribute considerable information when classifying areas with abundant urban features.

Finally, it is interesting to note that the short-term classification scheme exploited just the four quantities contributing more information, although the contribution of the winter coherence is quite close to that of the spring one.

7.5 The fully automatic mode

The new satellite missions, such as the Canadian RADARSAT-2, the German TerraSAR-X and the Italian COSMO-SkyMed, will make available large archives of images. In principle, the monitoring of a specific area with time-series acquisitions with a small temporal scale will be feasible. The results reported in the previous sections have shown how SAR imagery and neural networks can be effective in producing classification maps. The accuracies obtained were rather high and encourage the implementation of the methodology in a more applicative

Table 7.8: Data set relative to year 1996. B_p refers to the perpendicular component of the baseline.

Acquisition Date	Satellite	B_p (m)
February 24, 1996	ERS 1	12
February 25, 1996	ERS 2	
March 30, 1996	ERS 1	106
March 31, 1996	ERS 2	
July 14, 1996	ERS 2	-

Table 7.9: Confusion matrix for year 1996.

class	AS	FO	HD	IB	LD	VE	WS
AS	76.24	0.61	0.00	0.00	0.00	21.07	2.06
FO	0.25	83.50	0.00	0.00	61.15	3.93	6.15
HD	0.00	0.00	71.33	2.73	27.74	1.19	0.00
IB	0.33	0.00	4.31	91.36	1.66	2.35	0.00
LD	0.05	2.28	7.99	0.00	82.56	7.02	0.09
VE	0.79	2.39	0.67	0.04	7.05	88.40	0.63
WS	2.47	2.09	0.00	0.00	0.00	1.73	93.70
Overall Error = 15.7%				**Kappa coefficient = 0.789**			

context. However, each image was processed by its own network, which has to be trained off-line. This might not meet the need of a fully automatic scheme for a fast processing chain.

Starting from these motivations, the same methodology was extended to be exploited in a fully automatic mode. This means the design of an unique neural algorithm capable of classifying images whose pixels have not been considered at all during the training phase, to stress the generalization characteristics of NNs. For this purpose a different set of images was used over the same test site, but corresponding to a different year, 1996 (see Table 7.8 for details). The classification procedure followed the same long-term scheme illustrated before in terms of the six physical quantities to be considered as input and classes to be discriminated, but it was significantly diverse in terms of the generation of the training set. In particular, a set consisting of 26,400 pixels was created derived only from the union of the 1994 and 1999 samples, i.e. no samples from the 1996 image were used for the training of this neural network.

The resulting confusion matrix in shown in Table 7.9. The overall accuracy is about 84.3% and Kappa coefficient is about 0.79. This is about ten points less than the accuracies obtained using a single network for each year. The origin of most of the errors resulted from in the misclassification of HD as LW, which, given the contiguity of the two classes, can be recognized as a minor drawback at this spatial resolution. On the other hand, merging these two classes, the overall accuracy reached 89.2% which represents a satisfactory target for this type of automatic application and a benchmark for successive studies.

7.6 Conclusions

The global dynamics of large urban areas is hard to monitor over time with images at metric spatial resolution. Moreover, the short reaction time allowed by emergencies makes radar acquisitions an essential element of the decision-making process. The archived ERS SAR images provide a valuable source of information on the evolution of human settlements and urban land-use.

Single-polarization decametric SAR data was investigated by discussing the extraction of suitable features for producing land-cover maps with the aim of joint use intensity, coherence and texture information. In particular, the NN algorithm behaved quite satisfactorily in handling such a heterogeneous data set and in yielding reasonable results. The accuracy in discriminating the 7 types of surface considered from a single interferometric acquisition exceeded 86%, with a Kappa coefficient larger than 0.78. The accuracy increased to about 88% (Kappa coefficient

above 0.80) when the information of different seasons was eploited by acquiring a second interferometric image pair and by adding a fifth image. However, the algorithm required care in designing the topology, in scaling input and output quantities, and in training and pruning procedures. When running in a fully automatic mode, the procedure performs well, although the results showed some difficulties in discriminating between low and high density residential areas. Backscattering intensity, Energy and Contrast, and spring coherence turned out to be the most effective parameters for classifying the particular landscape

Chapter 8

Exploiting the spectral information

Part of this Chapter's contents is extracted from:

1. G. Licciardi, F. Pacifici, D. Tuia, S. Prasad, T. West, F. Giacco, J. Inglada, E. Christophe, J. Chanussot, P. Gamba, "Decision fusion for the classification of hyperspectral data: outcome of the 2008 GRS-S data fusion contest", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 11, pp. 3857-3865, November 2009

Multi-spectral and hyper-spectral imagery observe the same scene at different wavelengths. Generally speaking, every pixel of the image is represented by several values, each corresponding to a different wavelength. These values correspond to a sampling of the continuous spectrum emitted by a pixel [130].

The main differences between multi-spectral and hyper-spectral imagery are in the number of bands (usually, 100-200 bands for hyper-spectral sensors) and the spectral width of these bands. The process of data acquisition is also different. In multi-spectral sensors, the separation between the different bands is generally done using filters or distinct acquisition systems whereas in the hyper-spectral case, the light is sent through a dispersive element (for example, a grating) to separate the different wavelength components. The light from one ground pixel is projected on a charge-coupled device line. Each element of the line simultaneously receives a narrow wavelength band that corresponds to this ground pixel [131].

The wealth of spectral information provided by hyper-spectral sensors has opened a ground-breaking perspective in many remote sensing applications, including environmental modeling and assessment, target detection for military and defense/security deployment, urban planning and management studies, risk/hazard prevention and response including wild-land fire tracking, biological threat detection, monitoring of oil spills [132].

However, the characteristics of hyper-spectral data sets pose different processing problems, such as classification and segmentation or spectral mixture analysis. In particular, conventional supervised classification techniques for hyper-spectral imagery were originally exploited under the assumption that the classes to be separated are discrete and mutually exclusive, i.e., it was assumed that each pixel vector is pure and belongs to a single spectral class. Often, however, this is not a realistic assumption. In particular, most of the pixels collected by imaging instruments contain the resultant mixed spectra from the reflected surface radiation of various constituent materials at a sub-pixel level. The presence of mixed pixels is due to several reasons. First, the spatial resolution of the sensor is generally not high enough to separate different pure signature classes at a macroscopic level, and the resulting spectral measurement can be a composite of individual pure spectra (often called endmembers in hyper-spectral analysis terminology) which corresponds to materials that jointly occupy a single pixel. Second, mixed pixels also result when distinct materials are combined into a microscopic mixture, independent of the spatial resolution of

Figure 8.1: The city of Pavia, Italy, imaged by ROSIS-03.

the sensor [132].

In this chapter, two state-of-the-art algorithms for the classification of very high resolution hyper-spectral imagery are discussed and compared. These algorithms ranked, respectively, the first and second position at the 2008 Institute of Electrical and Electronics Engineers (IEEE) Data Fusion Contest. The first algorithm, discussed in Section 8.2, uses different standard classifiers (neural networks and maximum likelihood) and performs a majority voting between the different outputs. The second algorithm, illustrated in Section 8.3, uses both spectral and spatial features. The spectral features are a 6-PCA extraction of the initial pixel vector values. The spatial information is extracted using morphological operators. These features are classified by combining several SVMs results using majority voting.

8.1 Data set

A hyper-spectral data set was distributed to every participant of the contest and the task was to obtain a classified map as accurate as possible with respect to the reference data, depicting land-cover and land-use classes. The ground reference was kept secret, but training pixels could have been selected by the participants using photo-interpretation to apply supervised methods. The data set consisted of an airborne image from the ROSIS-03 optical sensor with spatial resolution of 1.3 m. The flight over the city of Pavia, Italy, was operated by DLR in the framework of the HySens project, managed and sponsored by the European Union. The scene is illustrated in Figure 8.1. According to specifications, the number of bands of the ROSIS-03 sensor is 115 with spectral coverage ranging from 0.430 to 0.860μm. In total, 13 noisy bands were removed. For the contest, five classes of interest were considered, namely: Buildings, Roads, Shadows, Vegetation and Water. The corresponding spectral signatures are shown in Figure 8.2.

8.2 Neural network and maximum likelihood

The analysis of hyper-spectral imagery usually necessitates the reduction of the data set dimensionality to decrease the complexity of the classifier and the computational time required with the aim of preserving most of the relevant information of the original data according to some optimal or sub-optimal criteria [133].

The pre-processing procedure exploited in this section divides the hyper-spectral signatures into adjacent regions of the spectrum and approximates their values by piecewise constant functions. This technique was shown in [134] to reduce effectively the input space by applying piecewise constant functions instead of higher order polynomials. This simple representation outperformed most of the feature reduction methods proposed in the

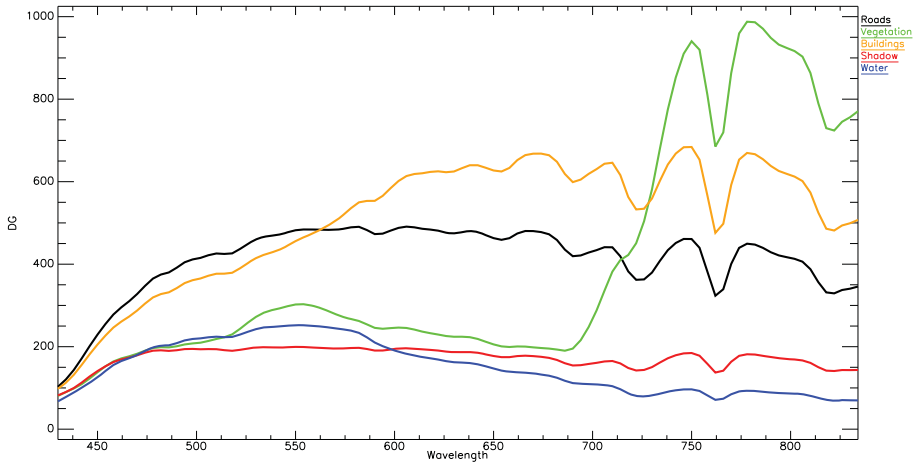


Figure 8.2: Spectral signatures of the five class of interest.

Table 8.1: Details of the resulting sub-bands.

	Sensor bands		Wavelength (μm)	
	from	*to*	*from*	*to*
B1	1	15	430	486
B2	16	35	490	566
B3	36	65	570	686
B4	66	75	690	726
B5	78	82	730	766
B6	86	90	770	786
B7	91	95	790	834

literature, such as principal component transforms, sequential forward selection or decision boundary feature extraction [135].

Assume S_{ij} to be the value of the ith pixel in the jth band, with a total of N pixels. The spectral signatures of each class extracted from ground reference is partitioned into a fixed number of I_k contiguous intervals with constant intensities to minimize:

$$H = \sum_{k=1}^{K} \sum_{i=1}^{N} \sum_{j \in I_k} (S_{ij} - \mu_{ik})^2 \tag{8.2.1}$$

where K represents the number of breakpoints and μ_{ik} the mean value of each pixels interval between breakpoints. For the data set considered, a number of $K = 7$ breakpoints was found to be a reasonable compromise between model complexity and computational time. The resulting seven sub-bands are reported in Table 8.1.

In [136], it was demonstrated that combining the decisions of independent classifiers lead to better classification accuracies. This combination can be implemented using a variety of strategies, among which majority voting (MV) is the simplest, and it was found to be as effective as more complicated schemes [137].

Majority voting was used on five independent maps resulting from two different methods, i.e. three neural networks and two maximum likelihood classifiers. For each method, the input space was composed by the seven features obtained from the reduction of the sensor bands, while the outputs were the five class of interest. Three different training sets were defined varying the number of samples to train the supervised classifiers, as reported in Table 8.2. In the following, the classification methods exploited are briefly discussed.

The topology of the multi-layer perceptron networks was designed through an optimization of the number of hidden layers and units, based on results that appeared in the literature and on previous experience. Two

Table 8.2: Number of selected training pixels for the NN classification.

	Buildings	Roads	Shadows	Vegetation	Water
Set 1	132,369	18,914	20,356	53,065	43,104
Set 2	33,168	6,525	3,260	14,323	26,816
Set 3	45,268	5,210	1,524	17,485	20,367

Table 8.3: Classification accuracies on the training set for NN, ML and MV.

	NN 1 *(set 1)*	NN 2 *(set 2)*	NN 3 *(set 3)*	ML 1 *(set 1)*	ML 2 *(set 2)*	MV
Acc. (%)	95.6	95.4	95.1	95.0	94.9	96.3
Kappa coefficient	0.936	0.932	0.929	0.927	0.925	0.946

Table 8.4: Confusion matrix for the NN classification.

class	Buildings	Roads	Shadows	Vegetation	Water
Buildings	213,359	391	155	203	0
Roads	246	10,430	0	71	0
Shadows	143	27	16,245	2	0
Vegetation	2	5	1	24,480	0
Water	0	0	0	0	10,961
Kappa coefficient = 0.9884					

hidden layers appeared to be a suitable choice, while the number of hidden neurons was found using a growing method increasing the number of elements. The variance of the classification accuracy for different initializations of the weights was computed to monitor the stability of the topology. The configuration 7-25-25-5 maximized the accuracy and minimized the instability of the results. Successively, three independent NNs were trained with sets 1, 2 and 3 (see Table 8.2), obtaining three different maps.

Maximum likelihood is a well known parametric classifier, which relies on the second-order statistics of a Gaussian probability density function for the distribution of the feature vector of each class. ML is often used as a reference for classifier comparisons because it represents an optimal classifier in case of normally distributed class probability density functions. ML classification was performed using sets 1 and 2 (see Table 8.2), obtaining two different maps.

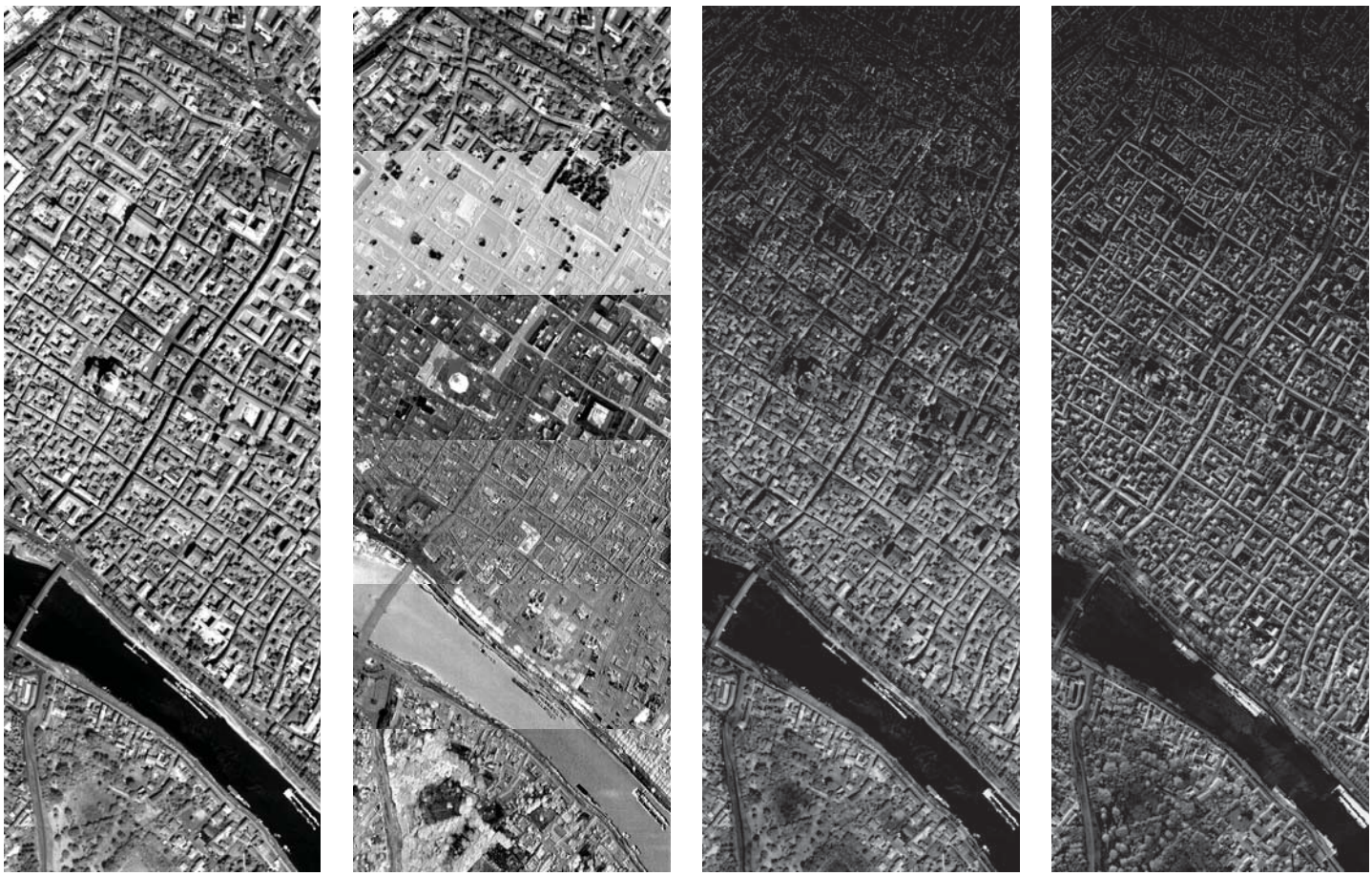
The results from the five classification maps were combined using majority voting implemented following two simple rules:

- a class is the winner if it recognized from the majority of the classifiers
- in case of a balance voting, the winner class is the one with the highest Kappa coefficient

The analysis of the classification accuracy based on the training sets, reported in Table 8.3, shows that majority voting increased the precision of the single classifier by about 1-2%. The confusion matrix (based on the unknown graound truth) is shown in Table 8.4. The resulting Kappa coefficient is about 0.9884.

8.3 Morphological features and SVM classifier

Principal component analysis was used to reduce the dimensionality of the original image. Specifically, the six first principal components, shown in the components composition of Figure 8.3a and Figure 8.3b, were retained to exploit the spectral information. These features counted for 99.9% of the variance contained in the original hyper-spectral bands. Moreover, the first principal component was also used to extract morphological features. In particular, 28 spatial features were extracted by applying opening and closing top-hat operators using diamond shaped SE with increasing diameter size (from 3 to 29 pixels), as shown in Figure 8.3c and Figure 8.3d, respectively.



(a) (b) (c) (d)

Figure 8.3: Principal component analysis applied to the data set: (a) first principal component and (b) the first six principal components retained (from top to bottom). Morphological (c) opening and (d) closing top-hat features – the size of the structuring element increases from 3 pixels (top) to 29 pixels (bottom).

Table 8.5: Labeled pixels for the SVM classfication.

class	Labeled pixels	Training	Validation	Test
Buildings	84,305	13,000	12,484	58,821
Roads	17,495	7,000	1,840	8,655
Shadow	11,375	7,000	758	3,617
Vegetation	49,730	5,000	7,770	36,960
Water	43,104	2,000	7,148	33,956
Total	**206,009**	**34,000**	**30,000**	**142,009**

A total of 206,009 labeled pixels were identified by careful visual inspection of the hyper-spectral image. Successively, these samples were divided into a training set, validation set (for model selection), and test containing the remaining pixels, as shown in Table 8.5.

Each input feature was converted to standard scores and stacked in a single 34-dimensional input vector. A RBF kernel was exploited, while the model selection was performed by grid search to find optimal parameters σ and C. Also in this case, the classification maps that improved the final solution were combined using majority voting.

The confusion matrix (based on the unknown graound truth) is shown in Table 8.6. The resulting Kappa coefficient is about 0.9858.

Table 8.6: Confusion matrix for the SVM classification.

class	Buildings	Roads	Shadows	Vegetation	Water
Buildings	213,351	385	260	107	5
Roads	414	10,296	12	25	0
Shadows	223	35	16,158	1	0
Vegetation	52	5	1	24,430	0
Water	0	0	0	0	10,961
Kappa coefficient = 0.9858					

Table 8.7: Final confusion matrix obtained as decision fusion of the five best individual results of the contest.

class	Buildings	Roads	Shadows	Vegetation	Water
Buildings	213,600	229	248	31	0
Roads	199	10,539	2	7	0
Shadows	71	43	16,301	1	1
Vegetation	8	9	1	24,470	0
Water	0	0	0	0	10,961
Kappa coefficient = 0.9921					

8.4 Conclusions

The data fusion contest was open for three months. At the end of the contest, 21 teams uploaded over 2,100 classification maps. The contest provided some interesting conclusions and perspectives as summarized:

- dimension reduction: most of the proposed methods used a dimension reduction as pre-processing. Most of them used the Principal Component Analysis, retaining a varying numbers of components. However, this step, with PCA or other methods, seems to be a *must*
- spatial and spectral features: several algorithms used both kinds of features. While the spectral information is easily extracted from the original spectra (directly or after some sort of dimension reduction), the spatial information remains a more tricky issue. Textural analysis and mathematical morphology provides some answers. Other ways to extract such meaningful information are currently being investigated. Mixing the spectral and spatial information appears as a clear direction for future researches
- support vector machines: almost all the best methods used some SVMs-based classifiers. SVMs appeared as extremely well suited for hyper-spectral data, thus confirming the results presented in the recent abundant literature
- neural networks: it must be emphasized that neural networks provided the best individual performance

To further investigate majority voting, the five best individual classification maps of the contest were fused together. The decision fusion of these individual results (the best two of those are described in the previous two sections) was achieved using a simple majority vote. Table 8.7 shows the corresponding final confusion matrix. The Kappa coefficient is about 0.9921. Even though the final score is less than 1% higher than the best algorithm, it remains the best. As a conclusion, decision fusion is indeed a promising way in order to actually solve the problem of classification in hyper-spectral imagery.

Chapter 9

Data fusion

Part of this Chapter's contents is extracted from:

1. F. Pacifici, F. Del Frate,W. J. Emery, P. Gamba and J. Chanussot, "Urban mapping using coarse SAR and optical data: outcome of the 2007 GRS-S data fusion contest", *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 3, pp. 331-335, July 2008

As a consequence of the increasing availability of multi-source remote sensing imagery, significant attention has been given by the scientific community to data fusion techniques. These approaches proved to offer better performance over a single-sensor approach by efficiently exploiting the complementary information of the different sensor types [138]. For example, characteristics of data acquired by optical and SAR sensors differ greatly. Multi-spectral satellites, such as QuickBird or Landsat, provide information on the energy scattered and radiated by the Earth 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. Synthetic aperture radar sensors, such as TerraSAR-X or ERS-1/2, provide measurements in amplitude and phase related to the interaction of the Earth surface with microwaves. In the case of C-band sensors, these acquisitions are characterized by high returns from buildings in urban areas, low and very low values from vegetated areas and water surfaces, respectively. Within residential areas, further discrimination is achievable, since the low density areas are generally characterized by lower backscattering, given the wide streets and trees. This means that SAR sensors provide information that may not be obtained from optical sensors alone and therefore data fusion potentially provides improved results in the classification process compared to the conventional single-source classification results [139].

Data fusion may be accomplished at different information levels such as signal, pixel, feature or decision: 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 using pixel-based fusion to improve the performance of the processing tasks. Feature-based fusion combines features extracted from different signals or images, while decision-level fusion consists of merging very dissimilar data at a higher level of abstraction [140].

In the data fusion literature, many alternative methods were proposed for combining multi-sensor decisions by weighting the influence of each sensor. A common approach to multi-source classification is to concatenate the data in a stacked-vector and treat it as a unique set of measurements [5], but statistical classifiers can become difficult to deal with since 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 integrated in a

Table 9.1: Data fusion contest data set.

Sensor	Acquisition date	Image ID	Mean	St. Dev.
ERS-1	August 13, 1992	Date 1	1,247.8	750.6
ERS-1	October 22, 1992	Date 2	1,475.7	740.1
ERS-1	June 24, 1993	Date 3	1,337.4	774.4
ERS-1	November 11, 1993	Date 4	1,457.5	727.0
ERS-1	October 3, 1994	Date 5	1,480.0	720.0
ERS-1	November 9, 1994	Date 6	1,500.1	807.5
ERS-1	July 22, 1995	Date 7	100.2	80.9
ERS-2	July 23, 1995	Date 8	115.5	92.9
ERS-2	August 27, 1995	Date 9	118.1	90.9
Landsat-5	April 7, 1994	Date 10	55.1	15.5
Landsat-7	October 8, 2000	Date 11	59.1	13.2

Table 9.2: Classes of interest with relative color mapping, number of training and validation samples.

Class	Color	TR	VA
City Center (CC)	Yellow	3,783	4,000
Residential Areas (RA)	Red	7,572	8,000
Sparse Buildings (SB)	Blue	5,994	8,000
Water (WA)	Cyan	893	1,000
Vegetation (VE)	Green	591	10,000

fuzzy logic-based fusion scheme [141]. Markov Random Fields also provide a powerful methodological framework for modeling spatial and temporal contexts allowing the images from different sensors and map data to be merged in a consistent way [142][143]. Nonparametric approaches, such as Neural Networks [144][145] or Support Vector Machines [146], can be exploited since they do not require specific probabilistic assumptions for class distribution. Hybrid approaches combining parametric methods and neural networks were proposed by Benediktsson *et al.* [147] by first treating each data source separately using statistical methods, and then using neural networks to obtain the final decision.

In this chapter, a NN-based data fusion framework, which ranked the first place at the 2007 IEEE Data Fusion Contest, is investigated. A set of satellite SAR and optical images, presented in Section 9.1, was made available by the contest organizers with the aim of obtaining a classified map as accurate as possible relative to the unknown (to the participants) ground reference. The methodology exploited and the results obtained are discussed in Section 9.2. Conclusions follow in Section 9.3.

9.1 Data set

The data set included the urban area of Pavia, Northern Italy, acquired by ERS-1 and ERS-2 during 1992 and 1995, and Landsat in 1994 and 2000, as shown in Figure 9.4a and Figure 9.4b, respectively. Details of the data set are reported in Table 9.1. The site was 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 [148]. The five classes of interest considered for the contest are City Center, Residential Areas, Sparse Buildings, Water and Vegetation. The number of training and validation samples are reported in Table 9.2. The training pixels used were part of the larger set provided by the contest organizers, while the validation samples (shown in Figure 9.4c) were selected by careful visual inspection of the scene and used only to have a rough estimation of the network performance during the designing process.

The results were evaluated using a web portal specifically designed for the contest that provided immediate feedback on the accuracy of the uploaded map by computing the confusion matrix. Again, the ground reference was kept secret to the contestants.

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 approaches. In the end, 9 different teams uploaded more than 100 classification maps, most of them continuously refining their algorithm performance. The best result is presented in the following section.

9.2 Neural networks for data fusion

The fusion procedure (based on a neural network approach) can be divided into three steps:

1. dimensionality reduction
2. classification phase
3. spatial filtering

As discussed for hyper-spectral imagery, the reduction of the input dimensionality is generally desirable when dealing with large data sets, as in this data fusion exercise. Principal component analysis was applied to decrease the number of inputs used to train the NN. PCA maps image data into a new, uncorrelated coordinate system in which the data have greatest variance along a first axis, the next largest variance along a second mutually orthogonal axis, and so on. The higher order components would be expected, in general, to have little variance [91]. The PCA eigenvalues for the SAR images relative to dates [1:6] and [7:9] are shown in Figure 9.1. As expected, only the first principal component shows large variance.

Theoretically, the input space reduction can be independently applied to both SAR and/or optical imagery, but a loss of useful information may be encountered during this processing if the input variables differ significantly in magnitude. This fact 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 set (see Table 9.1), two different experiments were investigated.

PCA was first applied to statistically similar images, as shown in Figure 9.2a, resulting in 8 inputs: 2 from SAR data, using the first component from date [1:6] and [7:9], and 6 from optical data, using for each pair of bands only the first PCA component, as reported in Table 9.1. The obtained 8 inputs produced a map with poor classification accuracy (Kappa coefficient of 0.6091 with respect to the validation set obtained by visual inspection).

In the successive experiment, PCA was applied only to statistically similar SAR images (see Table 9.1), resulting in 14 inputs: 2 from SAR data (again, considering the first component of dates [1:6] and [7:9]) and 6+6 from optical data, as shown in Figure 9.2b. This scheme produced a more accurate map than the previous one, resulting in a Kappa coefficient of 0.816.

A further alternative might have been to exploit the correlation matrix (instead of the covariance matrix) in the PCA processing, but this approach resulted in lower classification accuracy.

A multi-layer perceptron neural network was used to fuse the SAR and optical images. Once the input and the output neurones of the network are established (corresponding to the number of input features and the number of desired classes, respectively), the critical step was to find the optimal number of units to be considered in the hidden layers. As discussed, two different approaches can be used to find the best architecture:

- *growing*, in which the starting network is small and the neurons are subsequently added until the optimization criteria is reached

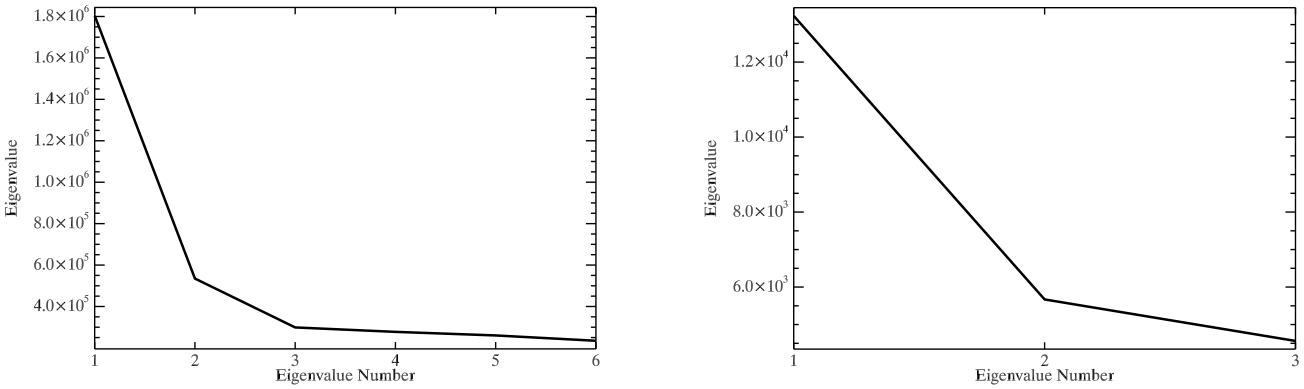


Figure 9.1: PCA eigenvalues for dates (a) [1:6] and (b) [7:9] of the SAR imagery. The difference in magnitude of the Eigenvalues in (a) and (b) is due to the different values distributions of the data sets considered.

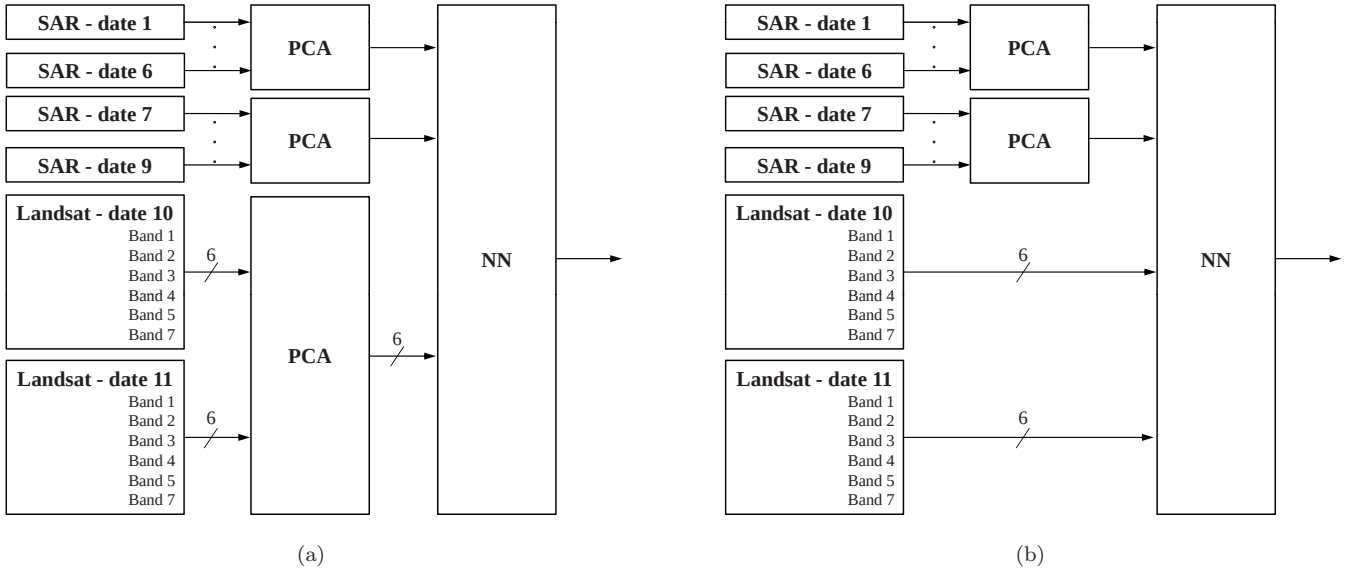


Figure 9.2: The features reduction is obtained by applying PCA to (a) statistically similar images and (b) statistically similar SAR images.

- *pruning*, in which the starting network is relatively large and the neurons are subsequently removed until the optimization criteria is satisfied

The latter approach was used here. After a reasonable evaluation in term 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, increasing the number of hidden neurons is effective up to a given number, after that the overall error value does not change significantly. Network pruning was then applied to minimizing the number of connections.

Based on the validation set produced by visual inspection of the scene, the progressive removal of neurons and connections followed two distinct goals:

- to reach the minimum of the classification error (pruning)

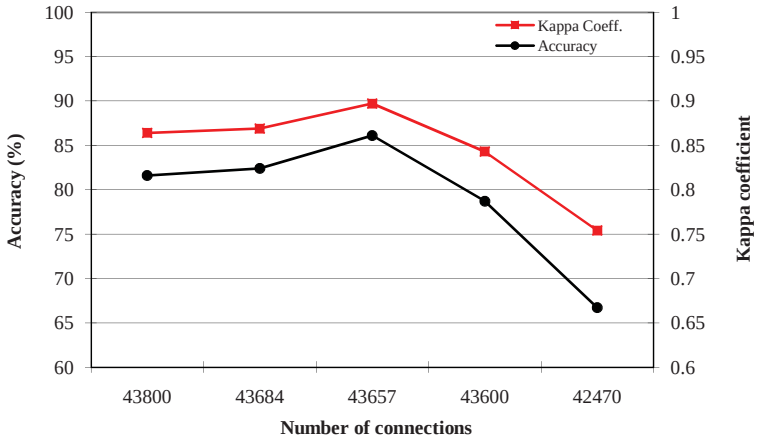


Figure 9.3: Classification accuracy with respect of the number of connections.

Table 9.3: Accuracy details of the of the different topologies.

Net ID	Connections	Accuracy (%)	Kappa coefficient	Epochs
Full	43,800	86.4	0.816	4,227
net1	43,684	86.9	0.824	10,027
net2	43,657	89.7	0.861	11,377
net3	43,600	84.3	0.787	14,227
net4	42,470	75.4	0.667	70,727

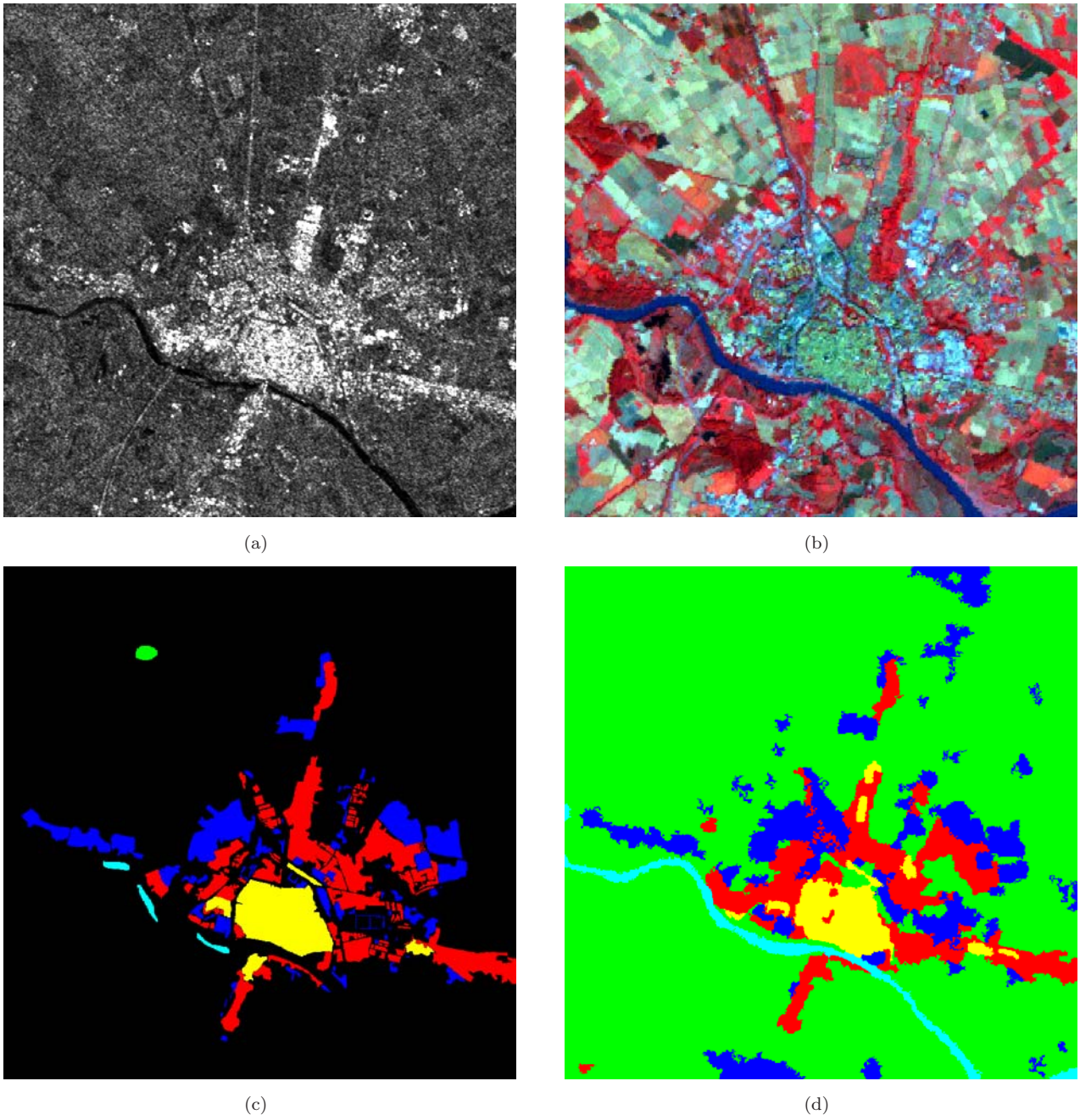
- to reach the minimum number of connections taking into account a maximum decrease of the classification accuracy of about 10% with respect to the fully connected network (extended pruning)

After 4,227 epochs of training, the full connected neural network (43,800 connections) correctly classified 86.4% of the validation patterns as in Figure 9.3 which shows the classification accuracy with respect of the number of connections.

The full connected neural network was pruned reaching the minimum classification error with 43,657 connections (accuracy 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 of the classification accuracy. As shown in Table 9.3, 57 connections were sufficient to decrease the classification accuracy from 89.7% to 84.3%. The decrease of the classification accuracy of about 10% with respect of the full connected network was reached with 42,470 connections (3.0% of the initial connections). These accuracies were based on the validation set obtained by visual inspection of the scene and were only used to have a rough estimation on the performance of the different topologies during the designing process.

Classified images often suffer from a lack of spatial coherence, which results in speckle or holes in homogeneous areas. This noise phenomenon appears as isolated pixels or small groups of pixels whose classifications are different from those of their neighbors. Therefore, following the classification phase, spatial filtering was applied to reduce the lack of spatial coherence in homogeneous areas. The spatial filtering is often achieved by analyzing the neighborhood for each pixel and removing isolated pixels or cluster (*sieve* process), and then merging the small groups of pixels together to make more continuous and coherent units (*clump* process) [149]. Sieve and clump were used here to reduce the effect of isolated pixels removing all regions smaller than the designated MMU.

The *trial-and-error* strategy was used to define the optimal size with respect to the unknown test set, starting with a small cluster dimension (10 pixels). The highest classification accuracy was reached using a dimension of



(a) (b)

(c) (d)

Figure 9.4: City of Pavia imaged by (a) ERS-1 and (b) Landsat (Bands 431) sensors. In (c) and (d) are shown validation samples and the final classification map. The color codes are in Table 9.2.

142 pixels. Therefore, even though smaller clusters were correctly classified, they were considered unreliable. The classification map obtained after the spatial filtering reached the accuracy of 0.9698 in terms of Kappa coefficient relative to the validation set obtained by visual inspection.

The final classification map, shown in Figure 9.4c, reached the accuracy of 0.9393 in terms of Kappa coefficient with respect to the unknown ground reference data used to rank the contest results. The corresponding confusion matrix is reported in Table 9.4.

Table 9.4: Confusion matrix with respect to the contest ground reference data.

class	CC	RA	SB	WA	VE	Acc. (%)
CC	14,174	1,114	49	0	408	90.02
RA	700	31,692	410	54	825	94.09
SB	66	472	31,932	0	1,246	94.71
WA	1	2	105	4,174	245	92.20
VE	205	813	536	51	98,199	98.39
Acc. (%)	93.58	92.96	96.67	97.55	97.30	96.11
Kappa coefficient = 0.9393						

As expected, vegetated areas, sparse buildings and water surfaces showed higher classification accuracies (stemming from a higher class separability) with respect to the other two classes. In fact, the responses of areas characterized by high or moderate density of buildings (such as city center and residential) are quite similar in both optical and SAR sensors.

9.3 Conclusions

The design of a NN-based framework for data fusion of ERS1/2 and Landsat data was discussed. A few outcomes are listed in the following:

- the application of PCA to statistically similar images provided worse results than applying PCA only to statistically similar SAR images
- the network pruning improved the results of about 3%, even though less than 0.5% of the connection were removed
- as expected, vegetated areas, sparse buildings and water surfaces showed higher separability than high or moderate urban density
- similarly to the 2008 Data Fusion Contest (see Chapter 8), neural networks provided the best individual performance compared to more sophisticated approaches

Chapter 10

Image information mining

Part of this Chapter's contents is extracted from:

1. F. Del Frate, F. Pacifici, G. Schiavon, C. Solimini, "Use of neural networks for automatic classification from high-resolution images", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 4, pp. 800-809, April 2007

Over the past years satellite sensors have acquired a huge amount of data that was systematically collected, processed and stored. In the near future, the access to image archives will become even more difficult due to the enormous data acquired by the new generations of optical and SAR satellite sensors. As a consequence, new technologies are needed to easily and selectively access the information content of image archives [150].

Information mining and the associated data management are changing the paradigms of user/data interaction by providing simpler and wider access to data archives. Today, satellite images are retrieved from archives based on attributes, such as geographical location, time of acquisition and type of sensor, which provide no insight into the actual image information content. Then, experts interpret the images to extract information using their own personal knowledge, and the service providers and users combine that extracted knowledge with information from other disciplines in order to make or support decisions. However, the information extraction process is generally too complex, too expensive and too dependent on user conjecture to be applied systematically over an adequate number of scenes. Therefore, there is a need to develop automatic or semi-automatic information mining systems.

Many studies have proven the effectiveness of multi-layer perceptron networks as a tool for the classification of remotely sensed images. In the previous chapters, different examples of NN-based applications were discussed. However, the classification problem was normally focused on the use of a single NN for classifying and/or extracting specific features from a single image, namely the image where the training examples were taken.

A first attempt to overcome this limitation was discussed in Section 7.5. However, the capabilities of a single network of performing automatic classification and feature extraction over a collection of archived images have not yet been explored. This network might be used to retrieve all the images from an archive that contain or do not contain a specific class of land-cover, or where the ratio between areas corresponds to different classes within/out predefined ranges. In other words, the network allows the identification of high-level (object or region of interest) spatial features from the low-level (pixel) representation contained in a raw image or image sequence, hence addressing the image information mining field [151][152].

In this chapter, the generalization capabilities of this type of algorithm is investigated with the purpose of using NNs as a tool for fully automatic classification of collections of satellite images. In particular, applications

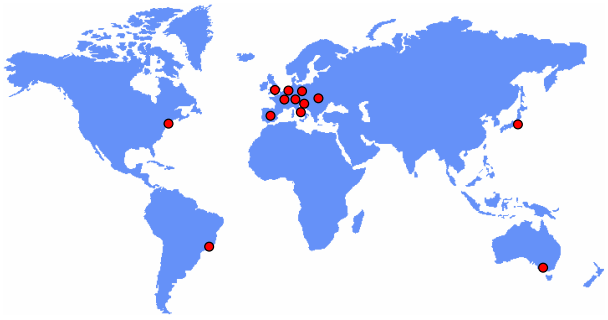
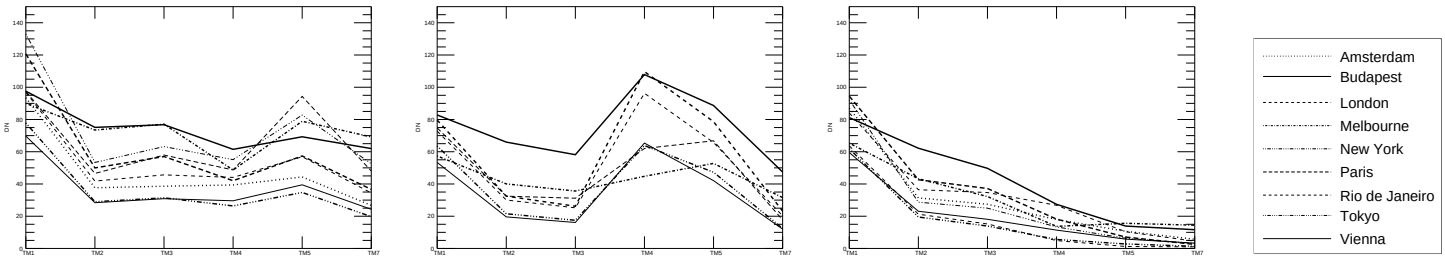


Figure 10.1: The Landsat data set contains urban areas located throughout the all five continents.



(a) (b) (c) (d)

Figure 10.2: Spectral signature of the Landsat images for the classes (a) high-density residential, (b) forest, and (c) water for different cities. Color code in (d).

to urban area monitoring are addressed, with the aim of distinguishing between areas with artificial coverage (sealed surfaces), including asphalt or buildings, open spaces, such as bare soil or vegetation, and water surfaces.

10.1 Neural networks for automatic retrieve of urban features in data archive

A neural network was designed for the automatic retrieval of urban features in a data archive of Landsat imagery collected over urban areas located throughout four different continents as illustrated in Figure 10.1.

The spectral signatures of the main classes of urban land-cover were first analyzed. Despite the considerable distances among the geographic locations, a good stability of the spectral information was observed, as shown in Figure 10.2 which reports the analysis of the classes high-density residential, forest, and water. For these three classes, the spectral signatures were computed starting from about 25,000 pixels, distributed over seven different geographic areas (see Figure 10.2d). Within the same class, the shape of the signatures are in general similar, and only a bias value seems to characterize the different plots. On the other hand, different classes have quite dissimilar spectral shapes.

The analysis carried out on other classes typical of urban and suburban land-covers confirmed the discrimination possibilities, especially if similar classes, such as forest and short vegetation, or high-density and low-density residential areas, were grouped together. The sealed fraction of an urban area is indeed one of the primary indexes for monitoring the urbanization process. However, many big cities are characterized by large amounts of water surfaces, belonging to rivers, lakes, or sea. Therefore, the addition of the class water is important to obtain a better monitoring.

The final classification problem aimed at discriminating between these three classes:

Table 10.1: Location and dates of the Landsat images used for the generation of the training set. The rightmost column indicates which classes were considered for the specific image.

City	Acquisition date	Classes
Amsterdam, The Netherlands	May 22, 1992	all
Barcelona, Spain	Aug 10, 2000	unsealed
Berlin, Germany	Aug 14, 2000	unsealed
Budapest, Hungary	Aug 09, 1999	all
London, U. K.	May 20, 1992	all
Melbourne, Australia	Oct 05, 2000	all
New York, U. S. A.	Sep 28, 1989	all
Paris, France	May 09, 1987	all
Rio de Janeiro, Brazil	Jan 18, 1988	all
Rome, Italy	Aug 03, 2001	all
Rome (2), Italy	Jan 16, 2001	all
Tokyo, Japan	May 21, 1987	all
Udine, Italy	Aug 16, 2000	sealed, water
Vienna, Austria	Sep 10, 1991	all

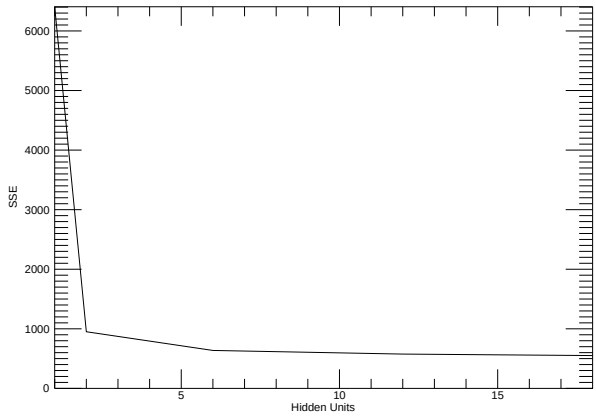


Figure 10.3: SSE values calculated over the test set varying the number of hidden neurons in a two hidden layers topology for the classification of the collection of Landsat images. The number of units is the same in both layers.

- *sealed*: surfaces with dominant human influence but non agricultural use, such as continuous and discontinuous urban fabric, industrial, commercial and transportation areas including all asphalted surfaces
- *unsealed*: arable lands, permanent crops, pasture, forestal and semi-natural areas
- *water*: all water surfaces

More than 56,000 patterns were selected to train the NN with samples extracted from an overall set of 14 images including urban areas of 12 large cities from 12 countries. Details of the data set exploited for the extraction of training samples are reported in Table 10.1 in terms of the images and the classes considered.

The design of the network was carried out using particular care in the selection of the number of hidden units. To this end, different network topologies were investigated. The plot in Figure 10.3 illustrates the sum-of-square error over test sets corresponding to different numbers of hidden units. Considering both the SSE and the network complexity, the best compromise was obtained with a 6-9-9-3 topology. Indeed, the increase in the number of hidden units did not change the SSE significantly.

The resulting classification maps of London, Melbourne and Rio de Janeiro are shown in Figure 10.4. At visual inspection, water bodies were detected rather precisely as the major parts of the urban lattice, including roads, bridges and airport runways. On the other hand, some inaccuracies could be noted in areas which appeared as

low residential areas at image visual inspection, but were labeled as unsealed in the classification map. A more quantitative validation follows in the next section.

10.2 Comparison of neural networks and knowledge-driven classifier

This section reports the accuracies obtained with the network developed above when applied to a set of unknown Landsat images. Moreover, the results achieved by the Knowledge-driven Information Mining (KIM) [151][153] over the same independent test sets are compared and discussed.

Since 2000, the German aerospace agency (DLR) has been developing the KIM prototype system, recognized by various organizations, such as the European Space Agency (ESA), the National Aeronautics and Space Administration (NASA) and the Centre National d'Etudes Spatiales (CNES), as the most advanced image information mining system in its class. KIM represents a theoretical framework of collaborative methods for extracting the content of large volumes of images and establishing the link between the user needs and the information content of images. As a classifier, KIM is based on a learning algorithm able to select and combine features, allowing the user to interactively enhance the quality of the queries. This computation is based on a few positive and negative training samples. The class discrimination is computed on-line and the complete image archive can be searched for images that contain the defined cover type.

For comparison between NN and KIM, the confusion matrix and the overall accuracy of each algorithm were computed. To this purpose a set of control points was collected over independent images not considered for the training of the algorithms. The number of the control points was taken according to the SRS. Therefore a different number of samples was collected in each class. It is important to remark that it was not possible to train the NN and the KIM classifiers under the same conditions due to the different inner structures of the two learning algorithms. In fact, the KIM system works with positive and negative examples for each class of interest. These samples were manually extracted starting from the reference pixels previously used for the NN training phase.

Further, a Minimum Mapping Unit (MMU) was defined. The MMU was taken equal to the optimal sampling size, which is dependent on the pixel size and geometric accuracy, according to the following formula:

$$A = [P(1 + 2G)]^2 \tag{10.2.1}$$

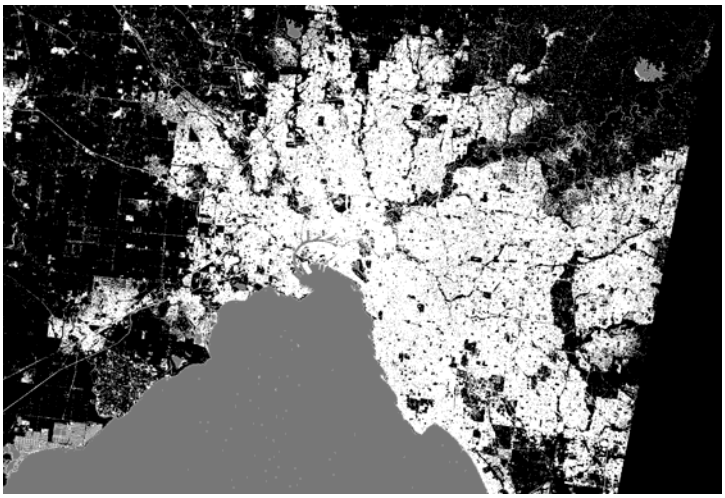
where A is the area to be sampled, P is the pixel size and G is the geometric accuracy (in pixels). This means that for Landsat images (assuming a geolocation accuracy of 0.5 pixel) the MMU should be at least of 60×60 m, corresponding to a square of 2×2 pixels.

The following urban areas were used as independent data sets for the validation of the two algorithms:

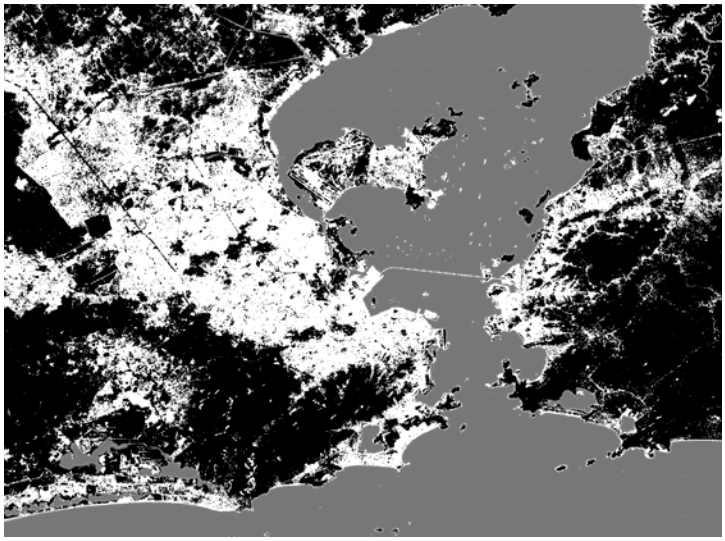
- Copenhagen, Denmark
- San Francisco, U. S. A.
- Washington D. C., U. S. A.
- Prague, Czech Republic

The classification maps of these four independent test sets, together with the Landsat image, are illustrated in Figure 10.5, Figure 10.6, Figure 10.7 and Figure 10.8, where sealed areas are displayed in white, unsealed in

(a)

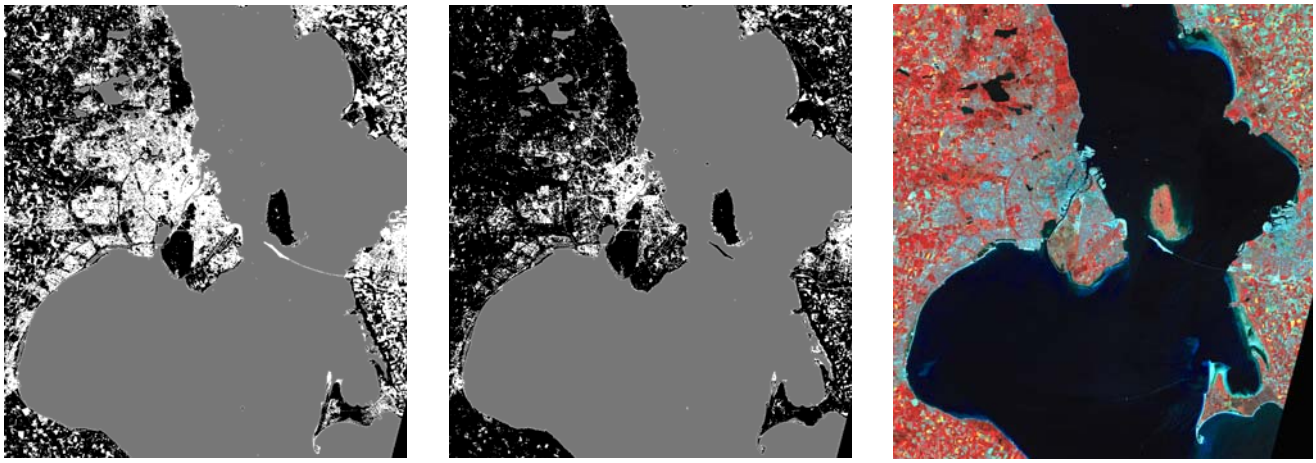


(b)



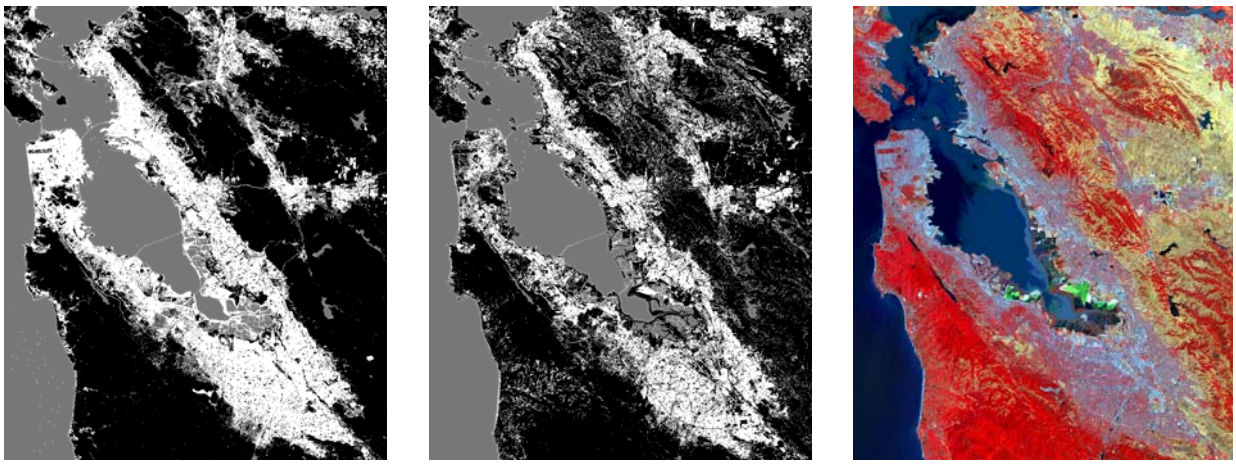
(c)

Figure 10.4: Examples of classification maps of (a) London, (b) Melbourne, and (c) Rio de Janeiro. Sealed areas are displayed in white, unsealed in black, and water in gray.



(a) (b) (c)

Figure 10.5: Copenhagen: classification maps provided by (a) NN and (b) KIM, and (c) Landsat image (Bands 431). Sealed areas are displayed in white, unsealed in black, and water in gray.

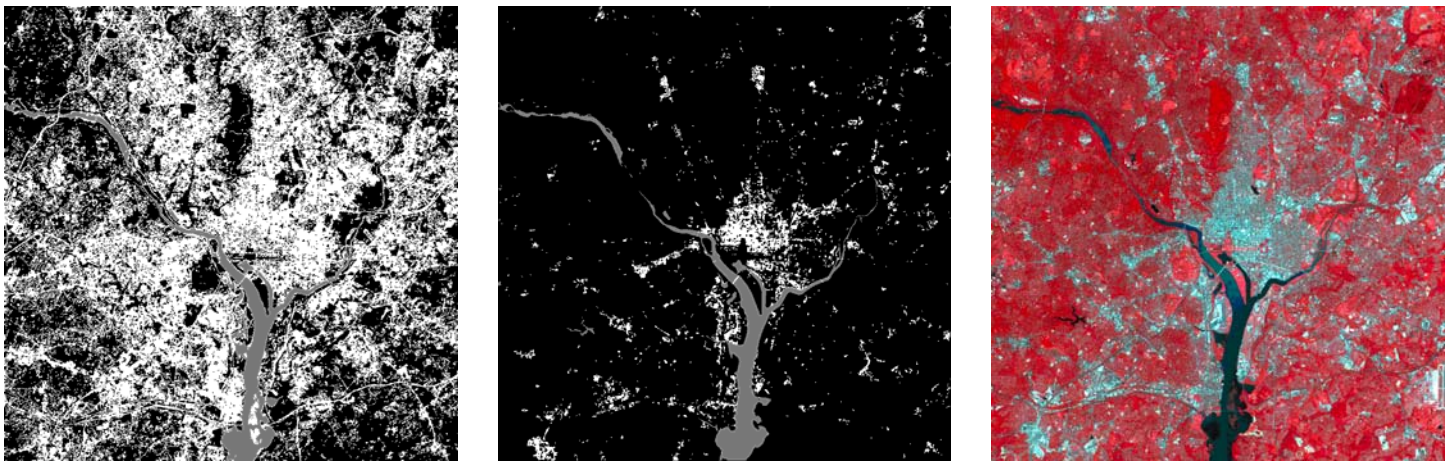


(a) (b) (c)

Figure 10.6: San Francisco: classification maps provided by (a) NN and (b) KIM, and (c) Landsat image (Bands 431). Sealed areas are displayed in white, unsealed in black, and water in gray.

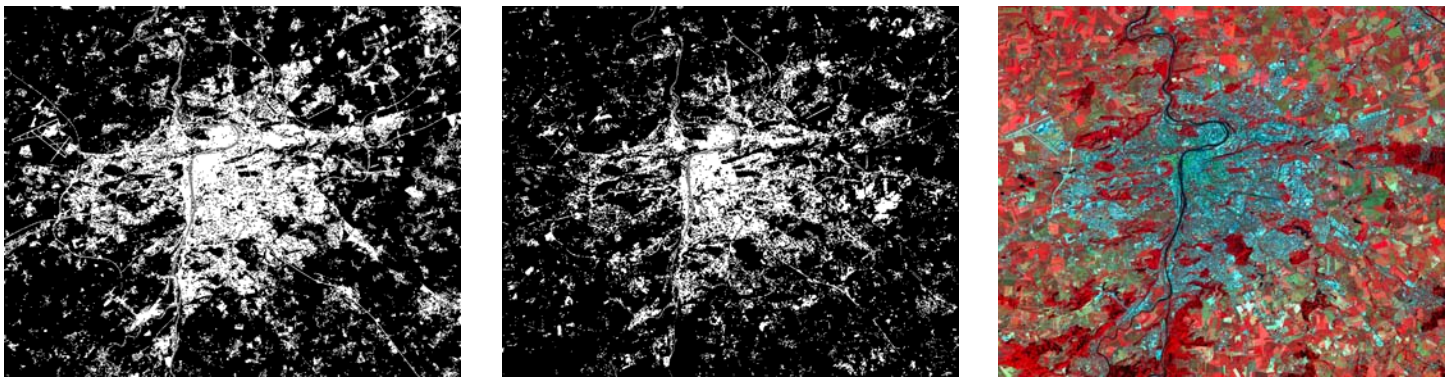
black, and water in gray. More quantitative results for both algorithms are reported in the confusion matrices in Table 10.2.

These results demonstrate the better performance of neural algorithm compared to KIM. Indeed both algorithms seem in general to underestimate the sealed areas in favor of the unsealed ones. In the case of KIM, this type of inaccuracy assumes higher values. This is clearly the case for the Washington D. C. image where the inclusion error of KIM for the class unsealed is almost of 90%, which means that most of the pixels actually belonging to the class sealed were erroneously assigned to the unsealed class. On the other hand, all pixels actually belonging to the unsealed class, were classified correctly. A possible explanation for this particular behavior may stem from the definition of the sealed class which included both high density and low density residential areas. These areas can be very different in the various parts of the World, as shown in Figure 10.9a and Figure 10.9b, where examples of Washington D. C. and Rome are presented. This contributed to incorrectly classifying many pixels belonging to moderately dense urban areas where a percentage of vegetation was present and the spectral signature was more



(a) (b) (c)

Figure 10.7: Washington D. C.: classification maps provided by (a) NN and (b) KIM, and (c) Landsat image (Bands 431). Sealed areas are displayed in white, unsealed in black, and water in gray.



(a) (b) (c)

Figure 10.8: Prague: classification maps provided by (a) NN and (b) KIM, and (c) Landsat image (Bands 431). Sealed areas are displayed in white, unsealed in black, and water in gray.

ambiguous.

Moreover, to operate in optimal conditions, KIM requires standard Landsat products: CEOS format, radiometric correction and nearest neighbor geometric correction. Neural networks did not require any kind of radiometric correction or pre-processing. In the severe case of Washington D. C., the neural network inclusion error for the sealed case is about 27% compared to 88% for KIM. As far as the retrieval of water surfaces and unsealed areas, the two methods appear to have comparable performance. As expected, the results obtained for the images considered in the training phase are better than those obtained for test images. The two algorithms show a similar behavior: for KIM the decrease of the overall accuracy considering test images instead of training images is about 4%, while for NN of about 3%, as reported in Table 10.3.

From a more general point of view, it is important to note that KIM is a user-oriented system which offers much wider opportunities for extracting information from remote sensing image archives, while the NN algorithm was designed specifically for the classification problem considered. In other words, KIM is not a dedicated tool for discriminating artificial from non artificial areas. In this context, the performance of KIM as a classifier can be recognized as good (accuracy 77.3% over training test and 73.4% over test areas, in comparison to 91.0% and

Table 10.2: Confusion matrices of Copenhagen, San Francisco, Washington D. C., and Prague for NN and KIM algorithms. The quantity k represents the Kappa coefficient.

(a) Neural Network confusion matrix of Copenhagen.

$k = 0.83$	*Sealed*	*Unsealed*	*Water*	*Err. (%)*
Sealed	14	4	0	4(22.2)
Unsealed	5	20	0	5(20.0)
Water	1	0	57	1(1.7)
Err. (%)	6(30.0)	4(16.7)	0(0.0)	10(9.9)

(b) KIM confusion matrix of Copenhagen.

$k = 0.74$	*Sealed*	*Unsealed*	*Water*	*Err. (%)*
Sealed	6	12	0	12(66.7)
Unsealed	1	24	0	1(4.0)
Water	2	0	56	2(3.4)
Err. (%)	3(33.3)	12(33.3)	0(0.0)	15(14.8)

(c) Neural Network confusion matrix of San Francisco.

$k = 0.95$	*Sealed*	*Unsealed*	*Water*	*Err. (%)*
Sealed	25	2	0	2(7.4)
Unsealed	0	51	0	0(0.0)
Water	1	0	23	1(4.2)
Err. (%)	1(3.8)	2(3.8)	0(0.0)	3(2.9)

(d) KIM confusion matrix of San Francisco.

$k = 0.72$	*Sealed*	*Unsealed*	*Water*	*Err. (%)*
Sealed	18	9	0	9(33.0)
Unsealed	8	43	0	8(15.7)
Water	1	0	23	1(4.2)
Err. (%)	9(33.3)	9(17.3)	0(0.0)	18(17.8)

(e) Neural Network confusion matrix of Washington D. C..

$k = 0.62$	*Sealed*	*Unsealed*	*Water*	*Err. (%)*
Sealed	43	15	1	16(27.1)
Unsealed	5	35	0	5(12.5)
Water	0	0	3	0(0.0)
Err. (%)	5(10.4)	15(30.0)	1(25.0)	21(20.6)

(f) KIM confusion matrix of Washington D. C..

$k = 0.62$	*Sealed*	*Unsealed*	*Water*	*Err. (%)*
Sealed	43	15	1	16(27.1)
Unsealed	5	35	0	5(12.5)
Water	0	0	3	0(0.0)
Err. (%)	5(10.4)	15(30.0)	1(25.0)	21(20.6)

(g) Neural Network confusion matrix of Prague.

$k = 0.77$	*Sealed*	*Unsealed*	*Water*	*Err. (%)*
Sealed	21	4	0	4(16.0)
Unsealed	5	70	0	5(6.7)
Water	0	0	1	0(0.0)
Err. (%)	5(19.2)	4(5.4)	0(0.0)	9(8.9)

(h) KIM confusion matrix of Prague.

$k = 0.65$	*Sealed*	*Unsealed*	*Water*	*Err. (%)*
Sealed	16	9	0	9(36.0)
Unsealed	4	71	0	4(5.3)
Water	0	0	1	0(0.0)
Err. (%)	4(20.0)	9(11.2)	0(0.0)	13(12.9)

(a) (b)

Figure 10.9: Examle of residential urban areas in (a) Washington D. C. and (b) Rome.

87.7% of NN), especially when true and false examples are directly taken over the image one wants to classify, i.e. the case of the image training set.

Table 10.3: Accuracy and Kappa coefficient for the training and test sets.

Algorithm	Training Set		Test Set	
	Acc. (%)	*Kappa coeff.*	*Acc. (%)*	*Kappa coeff.*
NN	90.9	0.84	87.7	0.80
KIM	77.4	0.57	73.4	0.55

10.3 Conclusions

In this chapter, a neural network was designed for the automatic retrieval of urban features in data archive of Landsat imagery. A few conclusions follow:

- today, Earth observation data are retrieved from archives based on different attributes which provide no insight into the actual image content. Experts then interpret the images to extract information using their own personal knowledge
- a neural network was successfully designed to retrieve urban features that contain or do not contain a specific class of land-cover. The performance appeared to be satisfactory, given the automatic nature of the procedure
- the classification maps showed better performance for neural algorithms compared to KIM
- KIM requires standard Landsat products to operate in optimal conditions, while neural networks do not need any kind of radiometric correction or pre-processing
- the results obtained may be considered as a first step in demonstrating how neural networks can contribute to the development of image information mining in Earth observation

Chapter 11

Active learning

Part of this Chapter's contents is extracted from:

1. D. Tuia, F. Ratle, F. Pacifici, M. F. Kanevski and W. J. Emery "Active Learning Methods for Remote Sensing Image Classification", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 7, pp. 2218-2232, July 2009

Any supervised classifier relies on the quality of the labeled data used for training. Theoretically, the training samples should be fully representative of the surface type statistics to allow the classifier to find the correct solution. This constraint makes the generation of an appropriate training set a difficult and expensive task which requires extensive manual (and often subjective) human-image interaction.

Manual training set definition is usually done by visual inspection of the scene and the successive labeling of each sample. This phase is highly redundant, as well as time consuming. In fact, several neighboring pixels carrying the same information are included in the training set. Such a redundancy, though not harmful for the quality of the results if performed correctly, slows down the training phase considerably. Therefore, the training set should be kept as small as possible and focused on the pixels that really help to make the models as efficient as possible. This is particularly important for very high spatial resolution images that may easily reach several millions of pixels. In this sense, there is a need for procedures that automatically, or semi-automatically, define a suitable (in terms of information and computational costs) training set for satellite image classification, especially in the context of complex scenes such as urban areas.

In the machine learning literature this problem is known as *Active Learning*. A predictor trained on a small set of well-chosen examples can perform as efficiently as a predictor trained on a larger number of examples randomly chosen, while being computationally smaller [154][155][156]. Following this idea, active learning exploits the user-machine interaction, decreasing simultaneously both the classifier error, using an optimized training set, and the user effort to build this set. Such a procedure, starting with a small and non-optimal training set, presents to the user the pixels whose inclusion in the set improves the performance of the classifier. The user interacts with the machine by labeling such pixels. The procedure is then iterated until an optimal criterion is reached.

In the active learning paradigm, the model has control over the selection of training examples between a list of candidates. This control is given by a problem-dependent heuristic, for instance the decrease in test error if a given candidate is added to the training set [155]. The active learning framework is effective when applied to learning problems with large amounts of data. This is the case of remote sensing images, for which active learning methods are particularly relevant since, as stated above, the manual definition of a suitable training set is generally costly and redundant.

Several active learning methods were proposed in the literature so far. They can be grouped into three different classes. The first class of active learning methods relies on SVMs specificities [157][158][159] and were widely applied in environmental science for monitoring network optimization [160], species recognition [161], in computer vision for image retrieval [162][163] and in linguistics for text classification and retrieval [164]. These active methods take advantage of the geometrical features of SVMs. For instance, the margin sampling (MS) strategy [157][158] samples the candidates lying within the margin of the current SVM by computing their distance to the dividing hyperplane. This way, the probability of sampling a candidate that will become a support vector is maximized. Tong and Koller [164] proved the efficiency of these methods. Mitra et al. [165] discussed the robustness of the method and proposed confidence factors to measure the closeness of the SVM found to the optimal SVM. Recently, Ferencatu and Boujemaa [163] proposed adding a constraint of orthogonality to the margin sampling, resulting in maximal distance between the chosen examples.

The second class of active learning methods relies on the estimation of the posterior probability distribution function of the classes, i.e. $p(\cdot|\cdot)$. The posterior distribution is estimated for the current classifier and then confronted with n data distributions, one for each of the n candidate points individually added to the current training set. Thus, unknown examples have to be estimated for as many posterior probability distribution functions as there are. In [166], uncertainty sampling was computed for a two-class problem and the selected samples were those that provided the closest class membership with a probability of 0.5. In [167], a multi-class algorithm was proposed. The candidate that maximized the KL divergence (or relative entropy [168]) between the distributions was added to the training set. These methods can be adapted to any classifier giving probabilistic outputs, but they are not well suited for SVMs classification, given the high computational cost involved.

The last class of active methods is based on the query-by-committee paradigm [169]. A committee of classifiers using different hypotheses about parameters is trained to label a set of unknown examples (the candidates). The algorithm selects the samples where the disagreement between the classifiers is maximal. The number of hypotheses to cover becomes quickly computationally intractable for real applications [170] and approaches based on multiple classifier systems have been proposed [171]. In [172], methods based on boosting [173] and bagging [174] were described as adaptations of the query-by-committee. In [172] the problem was applied solely to binary classifications. In [175], results obtained by query-by-boosting and query-by-bagging were compared using several batch data sets, and showed excellent performance of the methods proposed. In [176], expectation-maximization and a probabilistic active learning method based on query-by-committee were combined for text classification. In this application, the disagreement between classifiers was computed by the KL divergence between the posterior probability distribution function of each member of the committee and the mean posterior distribution function.

Despite both their theoretical and experimental advantages, active learning methods can rarely be found in remote sensing image classification. Mitra *et al.* [177] discussed a SVM margin sampling method similar to [157] for object-oriented classification. The method was applied successfully to a 512×512 multi-spectral four band image of the IRS satellite with a spatial resolution of 36.25 meters. Only a single pixel was added at each iteration, requiring several re-trainings of the SVM, resulting in high computational cost. Rajan *et al.* [178] proposed a probabilistic method based on [167] using maximum likelihood classifiers for pixel-based classification. This method showed excellent performance on two data sets. The first was a 512×614 AVIRIS spectrometer at 18 m resolution, while the second was a Hyperion $1,476 \times 256$ image (30 m spatial resolution). Unfortunately, the approach proposed cannot be applied to SVMs, again because of the computational cost. Jun and Ghosh [179] extended this approach, proposing to use boosting to weight pixels that were previously selected, but no longer relevant for the current classifier. Zhang *et al.* [180] proposed information-based active learning for target detection of buried objects.

Recently, this approach was extended by Liu *et al.* [181], who proposed a semi-supervised method based on active queries. In this study, the advantages of active learning to label pixels and of semi-supervised learning to exploit the structure of unlabeled data were fused to improve the detection of targets.

In this chapter, two variations of existing active learning models are discussed and compared with the aim at improving the adaptability and speed of these methods. In the first algorithm, the margin sampling by closest support vector (MS-cSV) is an extension of margin sampling [157] and aims at solving the problem of simultaneous selection of several candidates addressed in [163]. The original heuristic of margin sampling is optimal when a single candidate is chosen at every iteration. When several samples are chosen simultaneously, their distribution in the feature space is not considered and therefore, several samples lying in the same region close to the hyperplane, i.e. possibly providing the same information, are added to the training set. A modification of the margin sampling heuristic is here proposed to take this effect into account. In fact, by adding a constraint on the distribution of the candidates, only one candidate per region of the feature space is sampled. Such a modification allows sampling of several candidates at every iteration, improving the speed of the algorithm and conserving its performance.

The second algorithm, named entropy query-by-bagging (EQB), is an extension of the query-by-bagging algorithm presented in [172]. An entropy-based heuristic is exploited to obtain a multi-class extension of this algorithm. The disagreement between members of the committee of learners is therefore expressed in terms of entropy in the distribution of the labels provided by the members. A candidate showing maximum entropy between the predictions is poorly handled by the current classifier and is added to the training set. Since this approach belongs to the query-by-committee algorithms, it has the fundamental advantage of being independent from the classifier used and can be applied with any other method, such as neural networks.

Both methods, MS-cSV and EQB, are compared to classical margin sampling on three different test cases, including very high resolution optical imagery and hyper-spectral data. For each data set, the algorithm starts with a small number of labeled pixels and adds pixels iteratively from the list of candidates. SVMs were used instead of NNs to provide a fair comparison between the various active methods.

The chapter is organized as follows. Section 11.1 introduces the margin sampling approach and the two algorithms proposed. Section 11.2 illustrates the data sets, while the experimental results are discussed in Section 11.3. Final conclusions are in Section 11.4.

11.1 Active learning algorithms

Consider the synthetic illustrative example shown in Figure 11.1. The training set is composed of n labeled examples consisting of a set of points $X = \{x_1, x_2, \ldots, x_n\}$ and corresponding labels $Y = \{y_1, y_2, \ldots, y_n\}$ (see Figure 11.1a). The algorithm adds a series of examples to the training set from a set of m unlabeled points $Q = \{q_1, q_2, \ldots, q_m\}$ (see Figure 11.1b), with $m >> n$. In particular, X and Q have the same features. The examples are not chosen randomly, but by following a problem-oriented heuristic that aims at maximizing the performance of the classifiers. Figure 11.1c illustrates the training set obtained by a random selection of points on the artificial data set, while Figure 11.1d shows the training set obtained using an active learning method. In this case, the algorithm concentrates on difficult examples, i.e., the examples lying on the boundaries between classes. This is due to the fact that the classifier has control over the sampling and avoids taking examples in regions that are already well classified. This mean that the classifier favors samples that lie in regions of high uncertainty.

These considerations hold when $p(y|x)$ is smooth and the noise can be neglected. In the case of very noisy data, an active learning algorithm might include in the training set noisy and uninformative examples, resulting

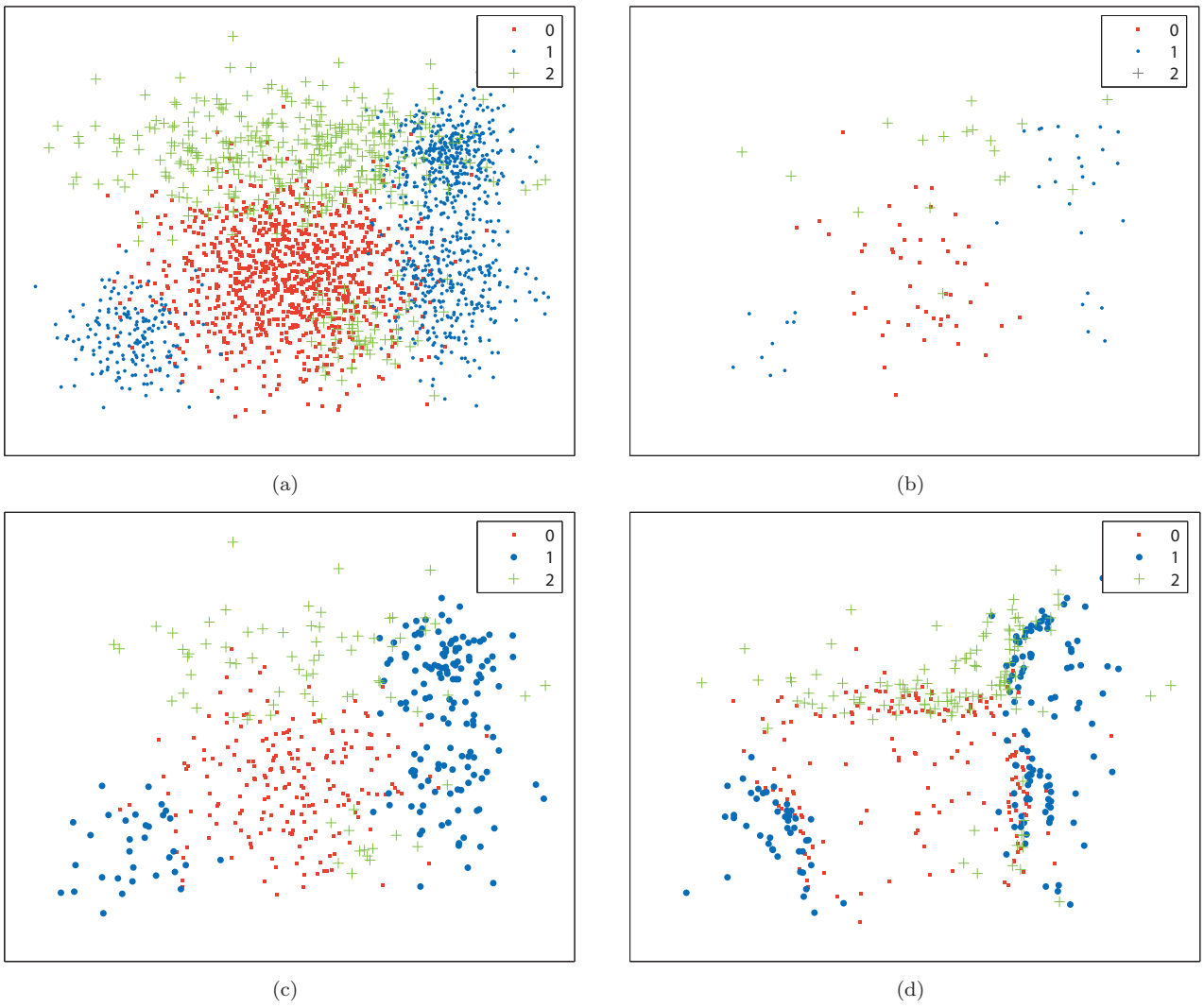


Figure 11.1: Example of active selection of training pixels: (a) initial training set X (labeled), (b) unlabeled candidates Q, (c) random selection of training examples, and (d) active selection of training examples.

in a selection equivalent to random sampling. In remote sensing, such an assumption about noise holds for multi-spectral and hyper-spectral imagery, but it does not for synthetic aperture radar imagery, where the algorithms discussed below can hardly be applied.

The margin sampling algorithm is discussed in Section 11.1.1, while the two proposed active learning approaches are illustrated in Section 11.1.2 and Section 11.1.3, respectively.

11.1.1 Margin sampling

Margin sampling is a SVM-specific active learning algorithm that takes advantage of SVM geometrical properties [157]. Assuming a linearly separable case, where the two classes are separated by a hyper-plane given by the SVM classifier (Figure 11.2a), the support vectors are the labeled examples that lie on the margin at a distance exactly 1 from the decision boundary (filled circles and diamonds in Figure 11.2). Assuming an ensemble of unlabeled candidates ("×" in Figure 11.2), the most interesting candidates are the ones that fall within the margin of the current classifier, as they are the most likely to become new support vectors (Figure 11.2b).

Consider the decision function of the two-class SVM:

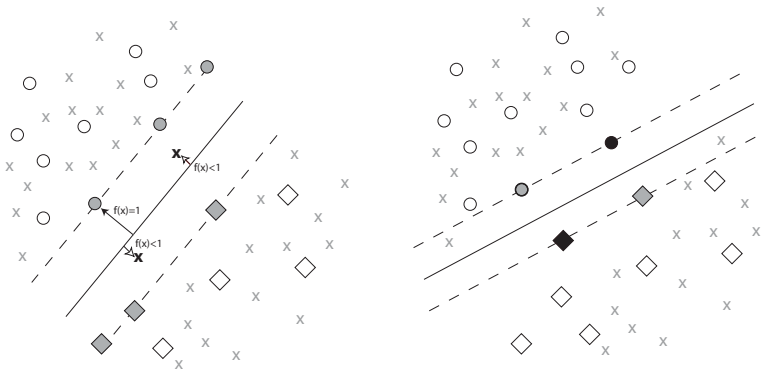


Figure 11.2: Margin sampling active learning: (left) SVM before inclusion of the two most interesting examples, and (right) new SVM decision boundary after inclusion of the new training examples.

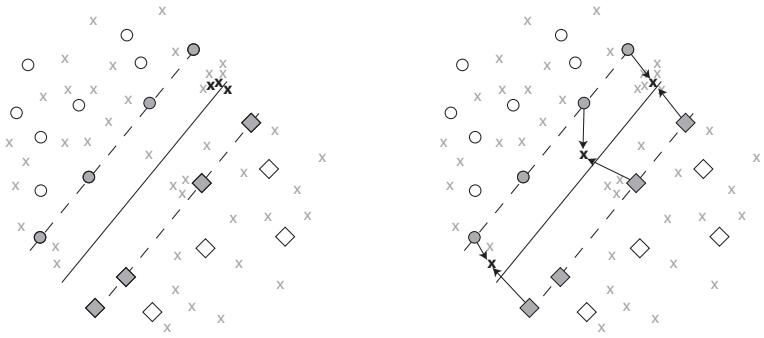


Figure 11.3: Margin sampling active learning: (left) candidates chosen by the margin sampling, and (right) candidates chosen taking into account the support vectors distribution.

$$f(q_i) = sign\left(\sum_{j=1}^{n} y_j \alpha_j K(x_j, q_i)\right) \tag{11.1.1}$$

where $K(x_j, q_i)$ is the kernel matrix, which defines the similarity between the candidate q_i and the j support vectors; α are the support vector coefficients and y_i their labels of the form $\{\pm 1\}$. In a multi-class context and using a one-against-all SVM, a separate classifier is trained for each class cl against all the others, giving a class-specific decision function $f_{cl}(x_i)$. The class attributed to the candidate q_i is the one that minimizes $f_{cl}(x_i)$.

Therefore, the candidate included in the training set is the one that respects the condition:

$$\hat{x} = \arg\min_{q_i \in Q} |f(q_i)| \tag{11.1.2}$$

In the case of remote sensing imagery classified with SVM, the inclusion of a single candidate per iteration is not optimal. Considering the computational cost of the model (cubic with respect to the observations), inclusion of several candidates per iteration is preferable. MS provides a set of candidates *Npts* at every iteration. However, margin sampling is not designed for this purpose and such a straightforward adaptation of the method is not optimal on its own. The left side of Figure 11.3 illustrates the effect of a non-uniform distribution of candidates when several neighboring examples lie close to the margin. If the margin sampling algorithm chooses three examples (right side of Figure 11.3) in a single run, three candidates from the same neighborhood will be chosen.

11.1.2 Margin sampling by closest support vector

As stated above, one of the drawbacks of the margin sampling is that the method is optimal only when a single candidate is chosen per iteration. In order to take into account the distribution in the feature space of the candidates, the margin sample algorithm needs to be modified. The position of each candidate with respect to the current support vectors is stored and this information is used to choose the most interesting examples.

The SVM solution provides a list of support vectors:

$$SV = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \tag{11.1.3}$$

with $\alpha \neq 0$. For every candidate q_i, the algorithm selects the closest support vector:

$$cSV = \arg \min_{x_j \in SV} K(x_j, q_i) \tag{11.1.4}$$

The heuristic of Equation 11.1.2 can be modified in order to include an additional constraint: the algorithm only takes into account the candidate associated with the minimal distance to the margin when confronted with several candidates located in the vicinity of the same support vector. In other words, no points added can share the closest support vector at each iteration, as shown by Equation 11.1.5.

$$\hat{x}_h = \arg \min_{q_i \in Q}(|f(q_i)|) \cap cSV_h \neq cSV_l \tag{11.1.5}$$

where $l = \{1, \ldots, h-1\}$ are the indexes of the already selected candidates.

It is important to notice that if only one candidate is added at each iteration, the MS-cSV algorithm is identical to margin sampling. Adding only one sample makes the cSV constraint useless, since the point chosen is the one that minimizes the distance to the margin between all the unique regions in the feature space. This point is simply the one that minimizes the distance to the margin over the candidates.

11.1.3 Entropy-based query-by-bagging

The query-by-bagging approach is quite different from the approaches discussed previously. The algorithm belongs to the query-by-committee algorithms, for which the choice of a candidate is based on the maximum disagreement between a committee of classifiers. First, k training sets built on bootstrap samples [182] are defined (i.e. a draw with replacement of the original data). Then, each set is used to train a SVM classifier and to predict the class membership of the m candidates. At the end of the bagging procedure, k possible labelings of each candidate are provided.

The approach proposed in [172] was discussed for binary classification. The candidates added to the training set were those for which the predictions are the most evenly split:

$$\hat{x} = \arg \min_{q_i \in Q} ||\{s \leq k | f_t(q_i) = 1\}| - |\{s \leq k | f_t(q_i) = 0\}|| \tag{11.1.6}$$

where t was one of the k classifiers and the binary labels were of the form {0,1}. If the classifiers agree with a certain classification, Equation 11.1.6 is maximized. On the contrary, uncertain candidates yield small values.

Here, the heuristic of Equation 11.1.6 is replaced by a multi-class form based on the maximum entropy of the distribution of the predictions of the k classifiers (Equation 11.1.8). By considering the k labels of a given candidate q_i, it is possible to compute the entropy of the distribution of the labels $H(q_i)$:

Table 11.1: Data set considered.

Location	Rome	Las Vegas	KSC
Dimension (pixels)	706 x 729	755 x 722	614 x 512
Satellite	QuickBird	QuickBird	NASA AVIRIS
Acquisition Date	May 29, 2002	May 10, 2002	March 23, 1996
Spatial resolution (m)	2.4	0.6	18.0

$$H(q_i) = \sum_{cl} -p_{i,cl} \log(p_{i,cl}) \tag{11.1.7}$$

$H(q_i)$ is computed for each candidate in Q and then the candidates that satisfy the heuristic:

$$\hat{x} = \arg\max_{q_i \in Q} H(q_i) \tag{11.1.8}$$

are added to the training set, where $p_{i,cl}$ is the probability of having the class cl predicted for the candidate i.

Entropy maximization gives a naturally multi-class heuristic. One candidate for which all the classifiers in the committee agree is associated with null entropy. Such a candidate is already correctly labeled by the classifiers and its inclusion does not bring additional information. On the contrary, a candidate with maximum disagreement between the classifiers results in maximum entropy, i.e., a situation where the predictions given by the k classifiers are the most evenly split. Therefore, the parallels with the original query-by-bagging formulation are strong.

Entropy-based query-by-bagging does not depend on SVM characteristics, but on the distribution of k class memberships resulting from the committee learning process. Therefore, it depends on the outputs of the classifiers only and can be applied to any type of classifier (maximum likelihood, neural networks, etc.).

Specific considerations can be done about the computational cost of the method depending on the classifier used. When using an SVM, the cost remains competitive compared to margin sampling, because the training phase scales linearly with respect to the number of k models (when all the training set are drawn in the bootstrap samples) compared to MS that exploits the entire training set. For smaller draws of the bootstrap samples, the additional computational burden becomes less than linear. When using probabilistic classifiers, and in comparison with models based on posterior probability distribution function estimation, entropy-based query-by-bagging implies k trainings for each iteration, instead of m trainings related to the estimation of the probability distribution for each set updated with a candidate. Therefore, the use of entropy for the k predictions of the candidates is computationally less expensive than using the methods presented above.

11.2 Data sets

The very high spatial resolution data sets used here are portions of the cities of Rome and Las Vegas, acquired by QuickBird in 2002 and 2004, respectively. Two different spatial resolutions were considered: 2.4 m multi-spectral for the Rome case and 0.6 m pansharpened multi-spectral for Las Vegas. Further, an 18 m spatial resolution hyper-spectral image of the Kennedy Space Center (KSC), acquired by AVIRIS in 1996, was used for comparison and validation purposes. Details of scenes and images are reported in Table 11.1. This variety in land-covers/land-uses made possible the evaluation of the flexibility of the active learning procedure when applied to different landscapes and spatial/spectral resolutions.

The pixels of the unlabeled Q set taken from the labeled training set ($Q = [\text{training set}] - X$) were used in order to avoid manual labeling between the iterations. Note that the labels of the candidates were never used in the selection process. A detailed description of the data sets follows.

Table 11.2: Classes, samples of the ground reference (GR), and legend color of the Rome data set.

Class	GR pixels	Color
Man made	22,318	Orange
Vegetation	2,673	Green
Soil	6,945	Brown

11.2.1 Rome

The Rome test site, shown in Figure 11.4a, represents part of the campus of the Tor Vergata University (Rome, Italy). This area is a typical suburban scene with residential, commercial, and industrial buildings. The different land-cover surfaces of interest were grouped in three main classes:

- *man-made*, including buildings, concrete, asphalt, gravel and sites under construction
- *green vegetation*
- *bare soil*, including low density/dry vegetation, and unproductive surfaces

The ground reference of 31,936 pixels (Figure 11.4b) was randomly split in a training set of 18,000 pixels used for both X and Q sets, a validation set of 7,000 pixels to estimate the optimal parameters, and a test set of 6,936 pixels to compute the test error at each step of the algorithm. The number of labeled pixels and reference map colors are given in Table 11.2.

The initial data set X was set to 300 pixels, which can be considered as a small training set for the dimensions of a very high spatial resolution image. Each algorithm ran for 70 epochs adding the 60 most relevant pixels to the actual training set at each iteration. After testing several hyper-parameters for the EQB sets and taking into account computational cost, the number of predictors k was set to 8. Each ensemble of bootstrap X'_l contained 75% of the pixels of X. To avoid the effects of different initializations on performance, the entire procedure ran 16 times with different starting sets X and Q.

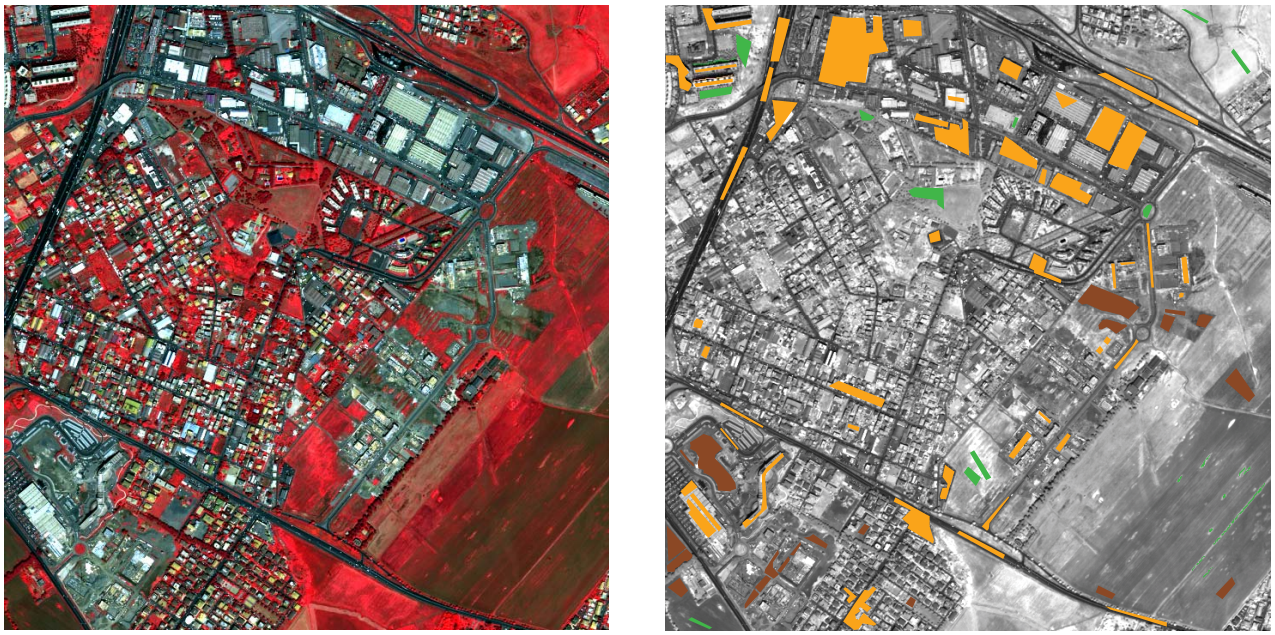
11.2.2 Las Vegas

The Las Vegas scene was previously described in Section 6.1.1. It contains regular criss-crossed roads and different examples of buildings characterized by similar heights (about one or two stories) with different dimensions, from small (residential houses) to large (commercial buildings). Details of the classes and on the number of labeled samples are reported in Table 11.3. The ground reference of 373023 pixels was split randomly into a training set of 30,000 pixels, a validation set of 25,000 pixels, and a test set of 318,023 pixels.

The initial data set X included 1,000 pixels, in order to take into account enough information for all the classes. The algorithm ran for 70 epochs, adding the 80 most relevant pixels to the current training set at each iteration. Following the search for the optimal EQB parameters, the number of EQB predictors k was set to 8. Each bootstrap sample X'_l contained 75% of the pixels of X. The entire procedure ran 11 times with different initial sets X and Q.

11.2.3 Kennedy Space Center

The third image was included in this analysis for comparison to the results achieved in [183]. The scene was acquired over the Kennedy Space Center, on March 23, 1996 by the hyper-spectral NASA AVIRIS instrument



(a) (b)

Figure 11.4: (a) Rome multi-spectral QuickBird image and (b) relative ground reference. Color codes in Table 11.2.

Table 11.3: Classes, samples of the ground reference (GR), and legend color of the Las Vegas data set.

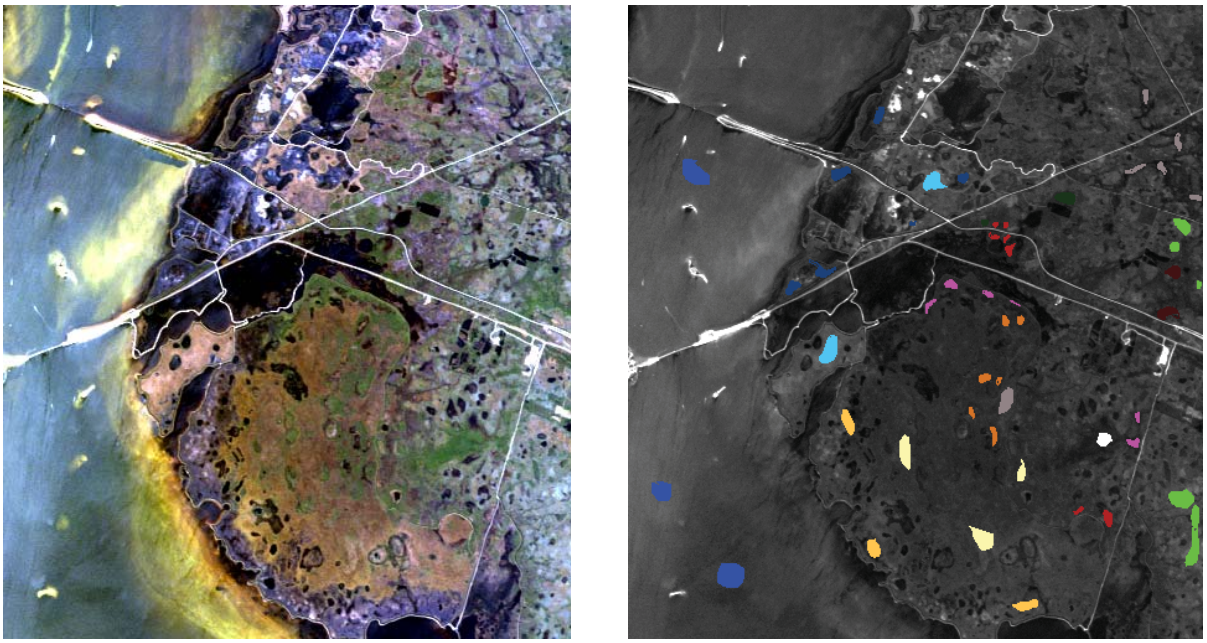
Class	GR pixels	Color
Residential buildings	87,590	Orange
Commercial buildings	22,769	Red
Asphalt	139,871	Black
Short vegetation	22,414	Light green
Trees	13,038	Dark Green
Soil	71,582	Brown
Water	1,472	Blue
Drainage channel	14,287	Cyan

Table 11.4: Classes, samples of the ground reference, and legend color of the KSC data set.

Class	GR pixels	Color
Scrub	761	Light green
Willow swamp	243	Pink
Cabbage palm hammock	256	Dark orange
Cabbage palm/oak hammock	252	Red
Slash pine	161	Dark green
Oak/broad-leaf hammock	229	Burgundy
Hardwood swamp	105	White
Graminoid marsh	431	Gray
Spartina marsh	520	Yellow
Cattail marsh	404	Orange
Salt marsh	419	Sky blue
Mud flats	503	Steel blue
Water	927	Blue

(224 bands of 10 nm width). Water absorption and low SNR bands were removed, resulting in a total of 176 bands. Thirteen classes representing the various land-cover types were defined (see Table 11.4) according to [183].

The 5,,211 pixels belonging to the ground reference were split randomly into a training set of 2500 pixels, a validation set of 1,300 pixels and a test set of 1,411 pixels. The initial data set X was set to 200 pixels. The algorithm ran for 70 epochs adding the 30 most relevant pixels to the actual training set at each iteration.



(a) (b)

Figure 11.5: (a) Kennedy Space Center hyper-spectral AVIRIS image and (b) relative ground reference. Color codes in Table 11.4.

11.3 Results and discussion

For each data set, the lower and the upper bound of the error were computed. The reference for the best achievable performance of the classifier was defined by a SVM moder trained on all the ground reference pixels, named here "Full SVM". A model that added randomly $Npts$ candidates from the Q set at every iteration was used as upper bound.

Since the candidates were chosen from the Q set only, the random selection performed stratified selection on the labeled areas. A second random sampling strategy, called Spatial Random Sampling (SpRS), was added to account for a more realistic random sampling, where the candidates were selected on a uniform spatial grid. To guarantee fair comparisons, no additional stratification with respect to the distribution of the labels was done. The error related to the SRS error can be interpreted as an upper bound because all the active methods have the objective of converging to the Full SVM performance faster than the SRS of examples from the list of candidates. It is important to recall that even SRS will converge to the Full SVM error rate, but slower than the active methods.

Optimal SVM parameters $\Theta = \{\sigma, C\}$ (RBF kernel was used) were found by grid search on the parameter space. The grid search procedure allowed the estimation of the best parameters for the initial SVM. Obviously, these parameters can become sub-optimal as the training set size increases. In this study, the re-estimation of parameters was necessary only for the Las Vegas case study (re-estimation of parameters is done when the solution seems to be trapped in a local minimum).

11.3.1 Rome

For the Rome data set, the Full SVM achieved an overall error of 8.70% with Kappa coefficient of 0.81. Figure 11.6a illustrates the evolution of the test error over the iterative process for the three algorithms considered and the

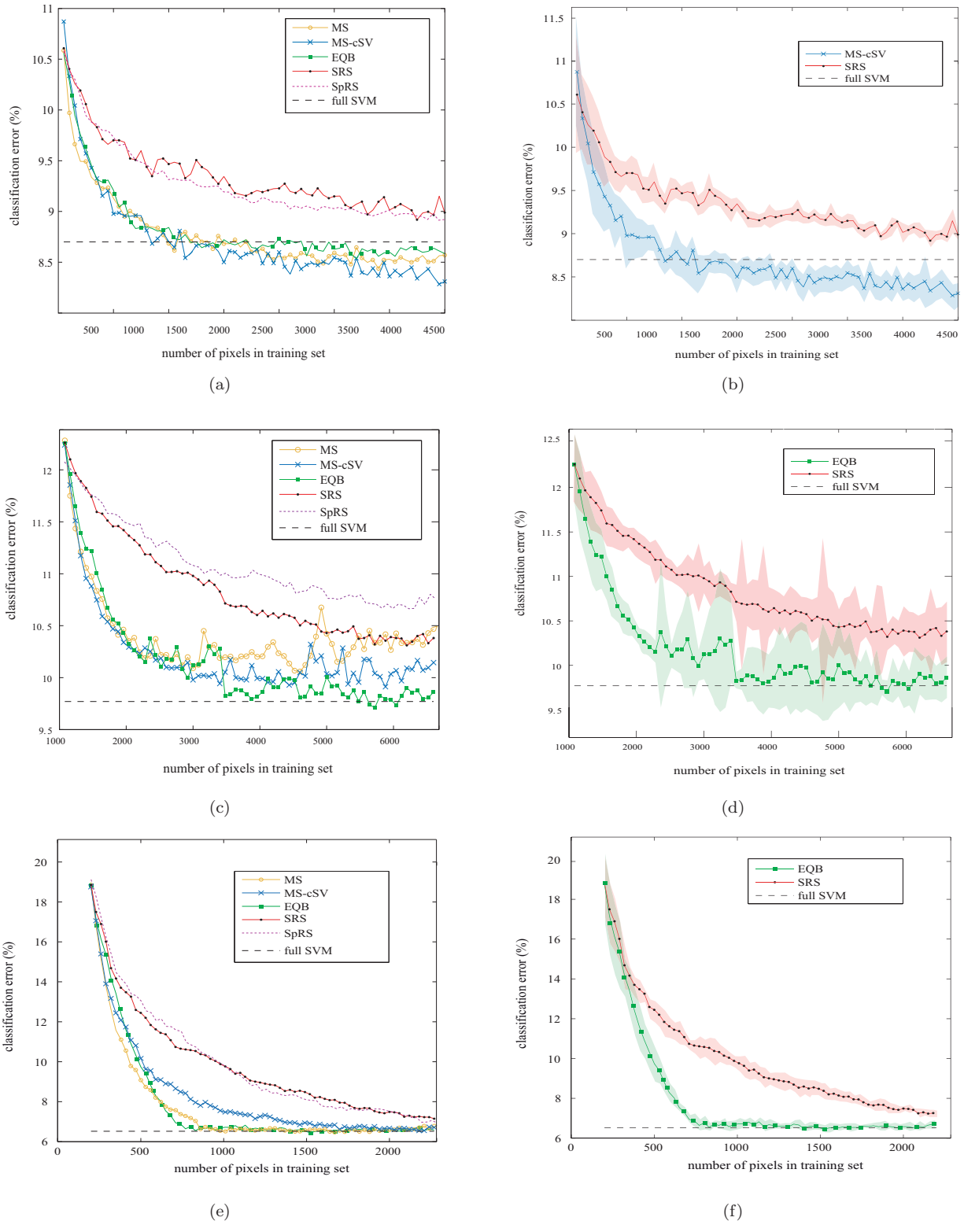


Figure 11.6: Classification error curves for (a) Rome, (b) Las Vegas, and (c) KSC. Each curve shows the mean error for growing size training sets over several runs of the algorithm starting with different initial sets X. In (d), (e) and (f) are shown the error bars of the best model against the SRS. The shaded areas show the standard deviation of over the results of the independent runs considered.

Table 11.5: Accuracies (%) and Kappa coefficient for the Rome data set.

	Full	MS	MS-cSV	EQB	SRS
iteration	-	26	26	26	26
training pixels	18,000	1,860	1,860	1,860	1,860
Man made	93.44	**94.21**	94.12	**94.21**	93.85
Vegetation	93.77	**91.44**	91.03	90.77	90.28
Soil	83.73	81.90	**82.93**	81.61	80.46
Accuracy	91.30	91.30	**91.43**	91.19	90.64
Kappa coeff.	0.810	0.809*	**0.813***	0.807*	0.794
Std. dev.		0.0033	0.0025	0.0042	0.0049
Conf. ($\alpha = 5\%$)		[0.807; 0.811]	[0.812; 0.814]	[0.804; 0.808]	[0.791; 0.797]

* = significantly different from SRS (Z test, [46])

SRS. The three active algorithms performed similarly and converged to the Full SVM error in about 25 iterations, i.e., using about 1,800 training pixels. This corresponds to 10% of the training set used by the Full SVM. The MS-cSV algorithm provided the best performance due to its higher accuracy for the class Soil. This class is the most difficult, because of its high overlap with the class Man made in the construction sites. No parameter re-estimation was performed since the quick convergence of the three active methods to the Full SVM accuracy. Regarding computational time, the cSV model performed the first 20 iterations in one hour (including the model selection) and ended with 35 minutes per iteration at iteration 70, when 4,500 pixels were considered in X. Therefore, the algorithm needed approximatively half an hour to converge to the optimal solution, remaining highly competitive with respect to the Full SVM.

Accuracies per class are given in Table 11.5 for iteration 26 (1,860 pixels). All the active methods outperformed SRS in terms of Kappa coefficient for the three classes. The methods based on margin sampling outperformed the EQB as shown in the curves of Figure 11.6a.

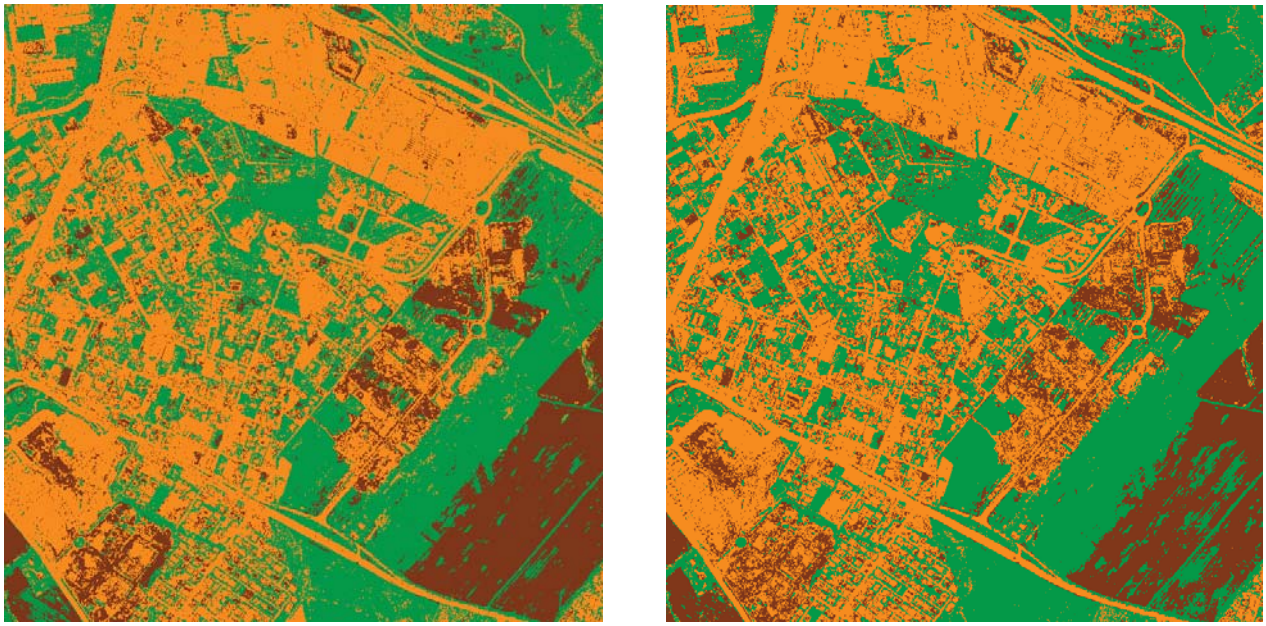
The classification maps of the Full SVM and MS-cSV are shown in Figure 11.7. The results of MS and EQB are similar and are omitted to avoid redundancy in the figures. The active learning maps were obtained with 10% of the training examples.

The biggest difference between Figure 11.7a and Figure 11.7b can be noted in the soil region at the bottom right corner of the scene. This region is characterized by mixed land-cover where both soil and vegetation are present. In this region, the active algorithm is still not optimal. Nonetheless, in some regions (for instance the bottom left corner for the class Soil and the bottom center for the class Vegetation), the active algorithm has a tendency to suppress noise that is generated by inconsistencies in the full training set. This is most likely related to the small size of the training set and to the active strategy. In fact, by including a few pixels carefully chosen near the boundary of the classes, the redundancy in the class definition is limited.

11.3.2 Las Vegas

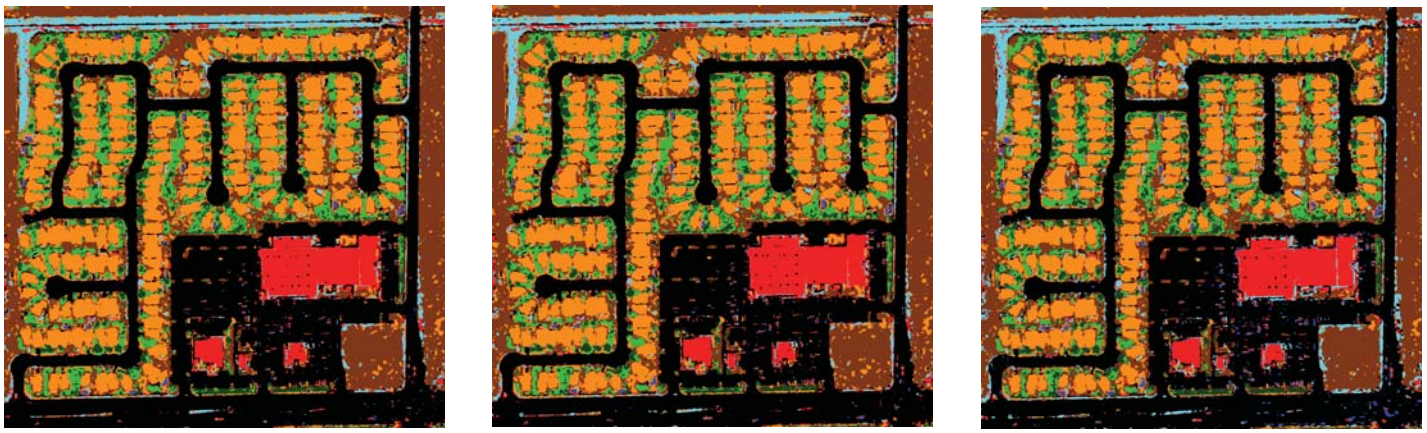
For the Las Vegas image, the composition of the training set was highly unbalanced (the class Water had only 1,472 labeled pixels out of a total of 318,023). Therefore, the active learning process was naturally much more difficult. To obtain convergence, a re-estimation of the parameters was done at iteration 30 (when the solution stabilized for the three active methods to a suboptimal result).

The classification maps for Full SVM, EQB, and SRS are shown in in Figure 11.8. The Full SVM achieved an error of 9.77% for the test set, with a Kappa coefficient of 0.870. The curves in Figure 11.6b show that only the EQB was able to converge to the Full SVM test error. This is due mainly to the parameter re-estimation at iteration 30 (about 3,400 pixels) that allowed the method to converge to the true minimum. The speed of the



(a) (b)

Figure 11.7: Classification map of the Rome image using (a) Full SVM and (b) MS-cSV. Color codes in Table 11.2.



(a) (b) (c)

Figure 11.8: Classification maps of the Las Vegas image using (a) Full SVM; (b) EQB; (c) SRS. Color codes in Table 11.3.

convergence of EQB was similar to the one observed for the other active methods, confirming its efficiency.

On the contrary, MS and MS-cSV did not improve with the re-estimation of the parameters and their results were similar to the ones obtained without re-estimation. This is due to the fact that these methods depend on the margin of the SVM, which is modified at every update of X. After every update, the current margin is only refined. However, when re-estimating the kernel parameters Θ, the margin changes radically, presenting to the algorithms a new active learning setting.

Despite the non-convergence, the MS-cSV performed better that the MS algorithm, taking advantage of the distribution of the pixels in Q after the first iterations, where both methods performed similarly. Moreover, the MS method did not converge, being equivalent to SRS after the 50^{th} iteration. An intuitive explanation is that

Table 11.6: Accuracies (%) and Kappa coefficient for the Las Vegas data set.

	Full	MS	MS-cSV	EQB	SRS
iteration	-	31	31	31	31
training pixels	30,000	3,480	3,480	3,480	3,480
Residential build.	85.24	84.75	84.86	**85.55**	82.25
Commercial build.	84.33	82.79	82.87	**84.77**	82.88
Asphalt	98.31	**98.23**	98.19	97.90	98.00
Short veg.	84.55	**82.59**	82.25	81.99	78.02
Trees	58.37	59.38	60.50	59.53	**63.61**
Soil	92.20	91.71	91.79	91.14	**91.88**
Water	67.35	65.86	65.10	65.07	**72.37**
Drainage ch.	80.47	77.21	78.27	79.56	**81.93**
Overall accuracy	90.23	89.64	89.73	**89.78**	89.09
Kappa coeff.	0.870	0.863	0.864*	**0.866***	0.855
Std. dev.		0.0054	0.0030	0.0020	0.0030
Conf. ($\alpha = 95\%$)		[0.859; 0.867]	[0.862; 0.866]	[0.865; 0.867]	[0.853; 0.857]

* = significantly different from SRS (Z test, [46])

MS-cSV avoids oversampling in dense regions close to the margin, and samples all the feature space equivalently.

Regarding global performance at iteration 31 (see Table 11.6), EQB showed the best results both in terms of accuracy (89.78%) and Kappa coefficient (0.866). These results corresponded to the good performance for the main classes of the image, where EQB even outperformed the Full SVM (residential and commercial buildings). MS-cSV also showed good performance (accuracy = 89.73%, Kappa coefficient = 0.864), higher than the MS results for five out of eight classes. Looking at the accuracies per class, SRS showed the best performance for the classes Trees, Drainage channel and (in particular) Water. These classes were the less extended in the ground reference. This result can be explained by the very high resolution of the image. For the active methods, the main sources of error are due to small objects such as cars, chimneys, road lines or dry bushes that contaminate the spectral signature of the main class. This can degrade the performance of an active learning process. For instance, cars do not have a specific class and are included in the ground reference in the class Asphalt. A SVM trained on spectral values will have the tendency to misinterpret these pixels and classify them as water or soil.

For an active learning process, this kind of a pixel has a high probability to be included in the training set because they are contradictory with respect to the class indicated by the ground reference. This causes a displacement of the decision boundary between Asphalt and Water/Soil into a zone otherwise clear. Such a displacement results in the improvement of the accuracy for the class Asphalt which becomes more robust to noise caused by small objects. However, accuracies for the classes Water or Soil are degraded, because spectral responses typical to these classes are classified as Asphalt. In contrast, SRS ignores these uncommon pixels. In fact, the random selection naturally pays little attention to pixels related to small objects. The analysis of the Kappa coefficient confirmed this hypothesis: small objects were labeled as Water and Soil by SRS much more than by the other two mappings. Even if Water or Soil pixels of the ground reference were better classified by SRS, commission errors remain important for the classes Asphalt and Residential buildings.

These considerations raise the question of the spatial resolution required for a classification task. In this case, the resolution is so high that objects introduce noise and degrade the solution.

11.3.3 Kennedy Space Center

For the KSC image, the Full SVM attained a test error of 6.52% (equaling the accuracy achieved in [183]) and a Kappa coefficient of 0.93. All the active learning algorithms converged to the lower bound at different speeds. The faster convergence was provided by the EQB algorithm, that reached the Full SVM error in about 20 iterations,

Table 11.7: Accuracies (%) and Kappa coefficient for the KSC data set.

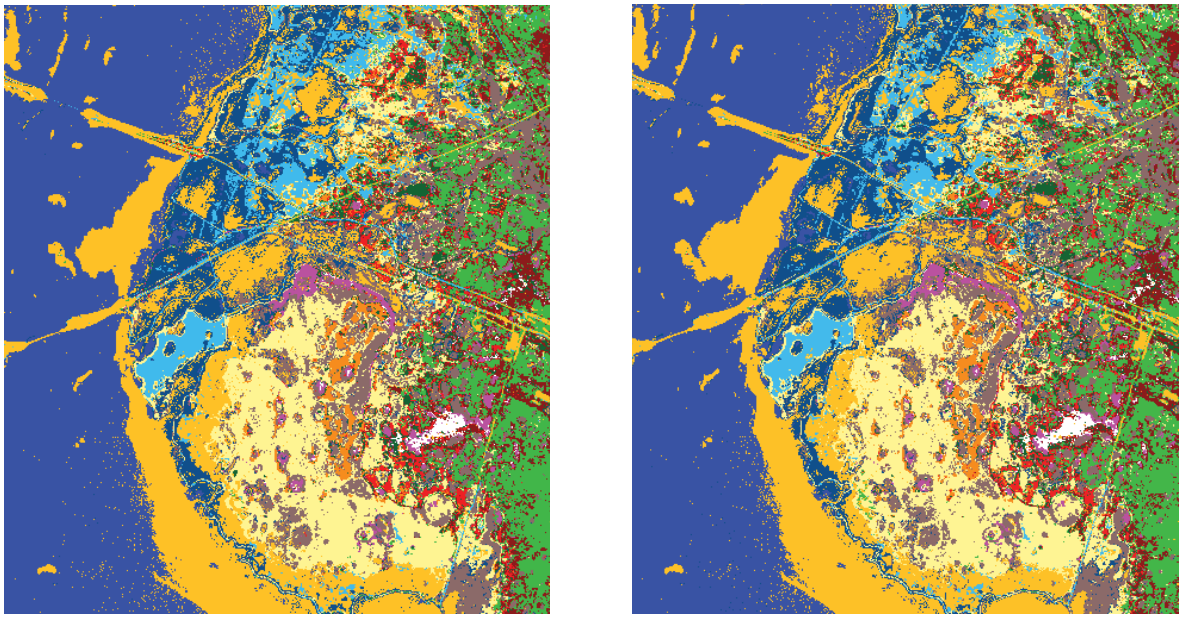
	Full	MS	MS-cSV	EQB	SRS
iteration	-	20	20	20	20
training pixels	2,500	800	800	800	800
Scrub	94.79	94.66	94.60	**94.79**	93.23
Willow swamp	88.14	**85.59**	85.38	84.53	81.36
Cabbage palm hammock	86.96	**86.59**	83.51	86.05	82.61
Cabbage palm/oak hammock	75.00	**76.73**	58.85	68.46	55.39
Slash pine	91.18	**91.07**	83.21	87.86	75.71
Oak/b. hammock	75.56	**76.73**	65.77	69.42	57.50
Hardwood swamp	83.78	**87.50**	84.87	81.25	82.57
Graminoid marsh	94.17	84.94	**94.31**	92.97	92.67
Spartina marsh	97.96	**97.70**	97.36	96.94	95.58
Cattail marsh	98.26	96.30	**98.04**	96.41	94.89
Salt marsh	99.11	98.66	**99.11**	97.43	96.87
Mud flats	93.06	88.80	**92.71**	91.41	90.28
Water	98.83	98.15	**98.74**	98.69	98.59
Accuracy	93.49	93.32	92.15	**93.36**	89.39
Kappa Coeff.	0.928	**0.919***	0.907*	0.910*	0.882
Std. dev.		0.0047	0.0057	0.0058	0.0054
Conf. ($\alpha = 95\%$)		[0.915; 0.923]	[0.902; 0.912]	[0.905; 0.915]	[0.877; 0.887]

* = significantly different from SRS (Z test, [46])

i.e. with a training set of 800 pixels. MS and MS-cSV showed a faster convergence in the first iterations. EQB showed a constant decrease and the best results in terms of overall accuracy (93.36%). This is related to the good results obtained for the classes Scrub and Water, the most represented in the test set. MS provided the best results in most of the classes and in terms of Kappa coefficient (0.919). Therefore, MS and EQB models seemed to be the most appropriate for this data set. MS-cSV converged to the Full SVM slower than the two other methods, as shown in Figure 11.9c. This can be explained by the results in Table 11.7. In fact, MS-cSV is unable to find the optimal solution for mixed classes, such as Cabbage palm/hammock and Oak/broadleaf hammock. These classes are very close in the hyper-spectral space and can impair the choice of the closest support vectors. Since MS-cSV does not select training points in the same dense area, the constraint on density of the candidates prevents their simultaneous selection, despite their importance. These pixels are sampled slower than with the MS algorithm, resulting in slower convergence to the Full SVM by the smaller accuracies for mixed classes. Therefore, MS-cSV seems to be less efficient in the presence of overlapping classes. Nonetheless, for non-mixed classes, MS-cSV often gives the best result in terms of overall accuracy.

11.3.4 Robustness to ill-posed scenarios

In an ill-posed scenario, where only a limited amount of labeled pixels per class is available in the initial training set X, the model built in the first iteration could fail to represent the true data distribution. Then, there is a risk that the candidates selected are not the most relevant to decrease the classification error. This could be particularly important for the EQB algorithm, where the entropy is computed over a committee of suboptimal classifiers. In this case, the selection is done in the wrong region of the feature space and there is no reason to believe that it would be worse than SRS. However, the benefits of EQB appear after a few iterations, as soon as a sufficient amount of pixels is selected to train the k models of the committee. Figure 11.10 shows this principle for the KSC image, starting with one labeled pixel per class (starting size of X is 13 pixels) and adding 30 pixels per iteration. After 4 iterations, the EQB algorithm starts to outperform SRS and converges to the Full SVM result when using about 800 pixels, equaling the results reported in Table 11.7.



(a) (b)

Figure 11.9: Classification of the KSC image using (a) Full SVM and (b) EQB. Color codes in Table 11.4.

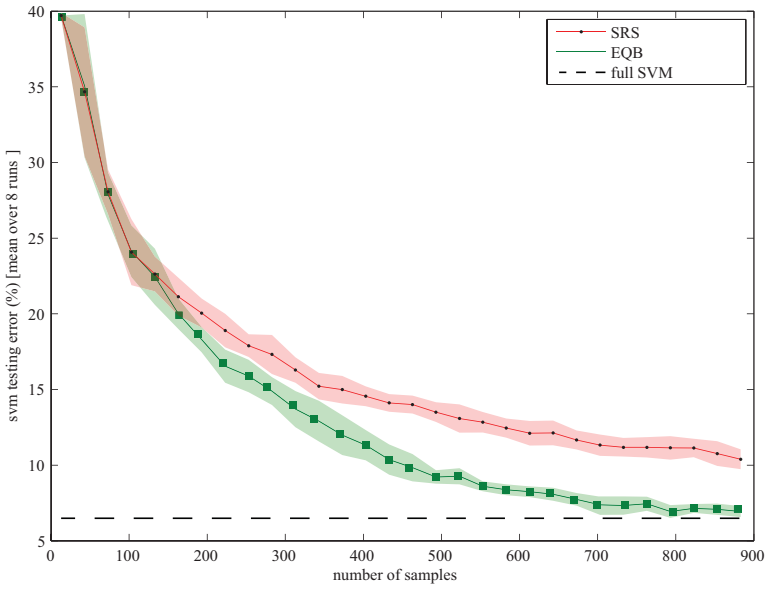


Figure 11.10: Results of EQB and SRS for the KSC image in an ill-posed scenario, where only 1 pixel per class is considered in X. Initial size of X is 13 pixels and 30 pixels are added at each iterations (markers on the curves). Shaded regions show the standard deviation of the predictions obtained over eight independent runs of the algorithms.

11.4 Conclusions

In this chapter, active learning models for remote sensing image classification were investigated. MS was discussed and two novel methods were proposed and applied for the classification of very high spatial resolution urban scenes. In these models, the predictor has control of the composition of the training set and chooses the most valuable

Table 11.8: Kappa coefficient for the data sets considered.

Method	Rome	Las Vegas	KSC
Full SVM	0.810	0.870	0.928
MS	0.809	0.863	**0.919**
MS-cSV	**0.813**	0.864	0.907
EQB	0.806	**0.866**	0.910
SRS	0.794	0.855	0.882

pixels for the improvement of its performance.

The first method, margin sampling by closest support vector, is a novel modification of the MS that takes the distribution of the unlabeled candidates in the feature space into account. In this way, the oversampling for dense regions is avoided, as well as the risk of not sampling important regions.

The second method, entropy query-by-bagging, is independent of the classifier and is based on committee learning. A committee of predictors labels the candidates and the entropy corresponding to the distribution of the predictions of the candidates is used as heuristic.

A few conclusions follow:

- training sets created actively can perform as well as a predictor trained on a complete ground reference (such as the Full SVM). Table 11.8 resumes the main results for the three images considered. For actively selected training sets, about 10% of the full SVM size is required for convergence to the same results in terms of classification accuracy

- for all applications, the convergence of the methods to the optimal result is quicker than the SRS, confirming the value of active learning methods

- the proposed EQB showed excellent performance for all data sets considered. The performance of this method is at least comparable to the MS, which is optimal for SVMs. The novelty of the EQB method lies in its independence of the classifier used that opens new possible applications for this method

- MS-cSV provides an interesting update of the classical MS method with its ability to handle the inclusion of several pixels at each iteration. The method showed better performance than MS on the QuickBird case studies, improving the MS efficiency along with convergence speed. Nonetheless, the method still needs improvement in order to handle situations with mixed classes, where the constraint on the closest support vector slows the speed of convergence

- issues related to small objects (such as cars) and the problems raised by their inclusion in the training set were addressed. Active methods, as well as the models that run on the whole ground reference (such as the Full SVM), suffer this problem. Nonetheless, both methods proposed showed enough robustness to result in higher accuracies than MS, especially for the main classes of the images considered

- all the active learning methods depend heavily on the quality of the initial data. If the initial training set is small, there is the risk that part of the feature space is not covered. If the uncovered part is in an area that the current predictor considers as correctly handled, it will be impossible to sample points from that area

Part III

Change detection of urban areas

Chapter 12

Introduction to the urban change detection problem

Detection of temporal changes is one of the most interesting aspects of the analysis of multi-temporal remote sensing images. In particular, change detection is very useful in applications such as land-use or land-cover change analysis, assessment of burned areas, studies of cultivation shifting or assessment of deforestation.

For many public and private institutions, knowledge of the dynamics of either natural resources or man-made structures is a valuable source of information in decision making [184]. Regional planners and decision makers require up-to-date information on the nature and impact of urban expansion or transitions to more intensive usage.

The availability of commercial very high spatial resolution satellite imagery has enlarged the number of applications in urban monitoring related to the detection of fine-scale objects such as single houses or small structures. At the same time, urban map updating represents a challenging area for the remote sensing community, due to the wide range of roof and road compositions characterized by different age, quality and materials.

Map updating is an intensive task that requires timely and accurate information [185]. The primary method of updating land-cover and land-use maps was, and in some case still is, through human interpretation. In this process, the full range of human interpretation capabilities can be employed, including the interpreters own knowledge of the area. However, it is time consuming, subject to errors of omission and the abilities of the interpreter can vary greatly. Also, there are limits to the ability of humans to absorb and process large volumes of information.

In general, different categories of change can be identified when comparing two or more very high spatial resolution images of the same scene. They may include, for example, newly built houses, roof variations and widened roads which may be important for public housing authorities. On the other hand, streets with temporary elements such as cars, trees with a full leaf canopy, or bare branches are examples of temporary variations of an existing object and not a conversion from one object to another. Therefore, all these changes may appear as irrelevant for public institutions. Moreover, technical aspects should be further taken into account. For example, shadowing effects and different off-nadir acquisitions may cause false signals which increase the difficulty of interpreting the changes detected.

Computer assisted methods offer approaches to detection and identification of land-cover and land-use changes. A lot of experience has already been accumulated in exploring change detection techniques for medium/high spatial resolutions, but just a few studies can be found for the new generation of very high spatial resolution satellite sensors.

In the remote sensing literature, two main approaches to the change detection problem are proposed: *unsupervised* and *supervised*. The former performs change detection by transforming the two separate multi-spectral

images into a single or multi-band image in which the areas of land-cover change can be successively detected. The latter is based on supervised classification methods, which require the availability of suitable training sets for the learning process. Although the supervised approaches exhibit some advantages over unsupervised methods, such as:

- the capability to explicitly recognize the kinds of land-cover or land-use transitions that have occurred
- the robustness to different atmospheric and light conditions at the two acquisition times
- the ability to process multi-sensor and/or multi-source images

the generation of an appropriate training set is usually a difficult and expensive task.

Many unsupervised techniques perform change detection using simple procedures to extract the final change map, e.g., by subtracting, on a pixel basis, the two images acquired at different times. More sophisticated techniques analyze the difference image using a Markov random field approach. In [186], the authors exploit the inter-pixel class dependency in the spatial domain by considering the spatial contextual information included in the neighborhood of each pixel. In [187], the proposed method combines the use of a MRF and a maximum a posteriori probability decision criterion in order to search for an optimal image of change. Bovolo and Bruzzone [188] introduced a set of formal definitions and presented a theoretical framework for the analysis of the distributions of changed and unchanged pixels in the context of the polar domain.

Among the supervised techniques, the most common is Post Classification Comparison (PCC) [189]. It performs change detection by comparing the classification maps obtained by independently classifying two remote-sensing images of the same area acquired at different times. In this way, the separate classification of multi-temporal images avoids the need to normalize for atmospheric conditions, sensor differences, etc. However, the performance of the PCC technique critically depends on the accuracies of the classification maps. In particular, the final change detection map exhibits an accuracy close to the product of the classification accuracies given at the two times [190]. This is principally due to the fact that PCC does not take into account the dependence existing between images of the same area acquired at two different times. The supervised Direct Multi-data Classification (DMC) is able to partially overcome this problem [191]. In this technique, pixels are characterized by a vector obtained by stacking the single features related to the images acquired at two times. Then, change detection is performed by considering each change as a single class and by training the classifier to recognize these transitions.

Appropriate training sets are required for the success of supervised methods. The training pixels at the two times should be related to the same points on the ground and should accurately represent the proportions of all the transitions in the entire images. Usually, in real applications, it is difficult to have training sets with such characteristics, but in general, this approach is more flexible than that based on unsupervised classification. In addition, they reduce the effects of different acquisition conditions and allow change detection using different sensors at different times.

This part is organized as follows. A novel parallel approach based on an neural architecture called Neural Architecture for very HIgh-Resolution Imagery (NAHIRI) is discussed in Chapter 13, while a change detection application of Pulse Coupled Neural Networks is addressed in Chapter 14.

Chapter 13

Neural-based parallel approach

Part of this Chapter's contents is extracted from:

1. F. Pacifici, F. Del Frate, C. Solimini and W. J. Emery, "An innovative neural-net method to detect temporal changes in high-resolution optical satellite imagery", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 9, pp. 2940-2952, September 2007

2. M. Chini, F. Pacifici, W. J. Emery, N. Pierdicca and F. Del Frate, "Comparing statistical and neural network methods applied to very high resolution satellite images showing changes in man-made structures at Rocky Flats", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 6, pp. 1812-1821, June 2008

The Neural Architecture for very HIgh Resolution Imagery is a framework for the change detection processing of very high resolution satellite imagery especially designed to deal with data sets acquired with different viewing angles. The distinctive feature of NAHIRI with respect to the methodologies already proposed in the literature is its ability to simultaneously exploit both the multi-spectral and the multi-temporal information contained in the input images. Moreover, it not only detects the different kinds of changes that have occurred, but also explicitly distinguishes the various typologies of class transitions, belonging to the family of supervised techniques.

As shown in Figure 13.1, the NAHIRI scheme uses a parallel approach that includes three different stages. *Classifier 1* and *Classifier 2* generate the so called *Change Map* using the multi-spectral information. In particular:

1. *Classifier 1* and *Classifier 1* independently produce the classification maps MAP_1 and MAP_2

2. the post classification comparison between MAP_1 and MAP_2 is computed, producing the *Change Map*

The information provided by the *Change Map* is composed of the $\{N_{cl}^2 - N_{cl} + 1\}$ possible transitions from one class to another, where N_{cl} is the number of chosen classes.

The innovative aspect of NAHIRI is the introduction of a third classifier, *Classifier 3*, which works in parallel with those described previously, and produces the so called *Change Mask*. This stage includes a multi-temporal operator which combines the multi-temporal information provided by the two images. The simplest way to exploit this information is to perform a DMC using a vector whose components contain the whole feature data set.

In order to discriminate between real changes and false alarms, the logic value of 1 is associated with changes and the logic value of 0 with no changes for each pixel in both *Change Mask* and *Change Map*. Obviously, the *Change Mask* and *Change Map* do not provide necessarily coherent information regarding scene changes. These 0 and 1 values are the inputs of the *AND* gate. The variation in the *Change Map* is considered as a valid change for the pixel only when both inputs of the *AND* gate have the value 1. Otherwise, the disagreement between the *Change Mask* and the *Change Map* is simply considered as a false alarm.

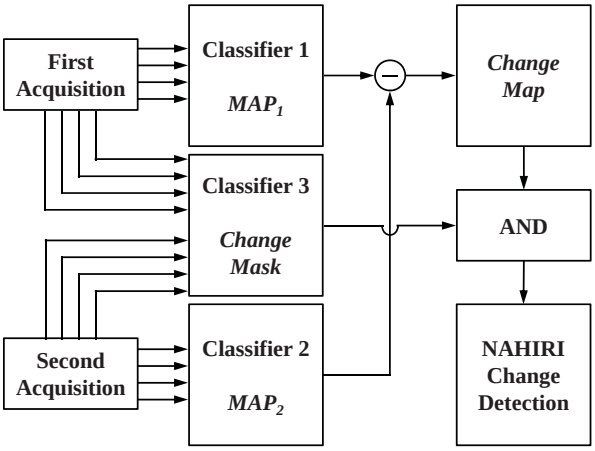


Figure 13.1: General schematic block diagram of the NAHIRI processing architecture.

The final output of the NAHIRI change detection technique is an image, which has different colors corresponding to different class transitions, whereas the original gray-scale satellite image background is associated with unchanged pixels. This type of presentation helps the end user to immediately individuate changes that have occurred in the study area, which is at the expense of missing the information on land-use class pertaining to any unchanged pixel.

The rest of this chapter is organized as follows. In Section 13.1, the parallel approach is applied to data sets with different spatial resolutions. In fact, although NAHIRI was designed to solve change detection problems at very high spatial resolution, its extension to moderate resolution images is straightforward and accurate. Successively, in Section 13.2 the robustness of the approach is investigated when different classifiers are exploited. In fact, the parallel architecture allows, in general, the user to exploit any kind of classification method, such as ML or SVMs. Conclusions are in Section 13.3.

13.1 The parallel approach at different spatial resolution

In this section, the NAHIRI architecture is applied to both QuickBird and Landsat images over two test areas, corresponding to different landscapes. One is located next to the campus of Tor Vergata University (Rome, Italy). The region is a typical sub-urban landscape that includes dense residential, commercial and industrial buildings. The other area is located in Rock Creek-Superior, Colorado, U. S. A., which includes single and multi family houses, shopping centers and low-density commercial buildings.

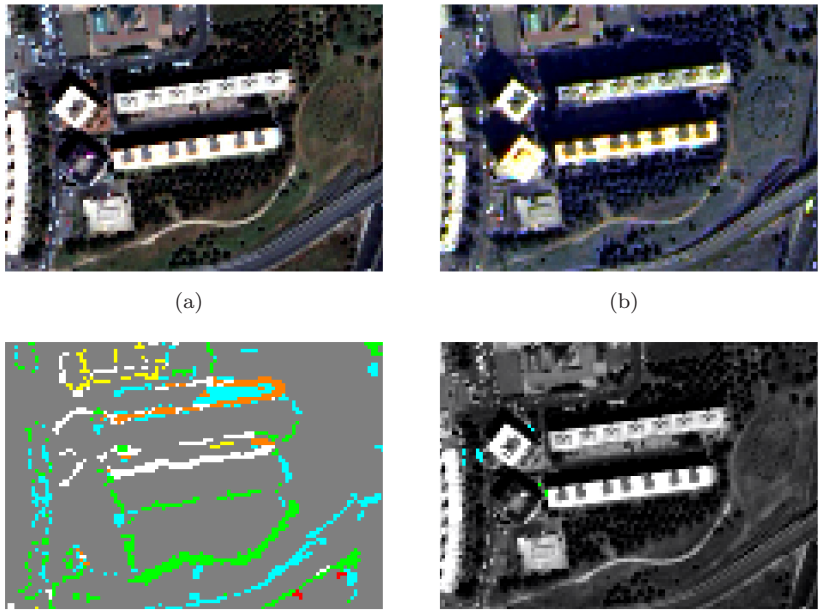
As already discussed, change detection at very high spatial resolution imagery over urban areas represents a difficult challenge. In fact, the effects of temporary objects, imperfect image co-registration, different image acquisition angles and solar conditions may occur frequently in the same scene, yielding false positives in change detection products.

These effects are illustrated for the case of QuickBird images in Figure 13.2, Figure 13.3 and Figure 13.4. Three different classes were investigated to distinguish between man-made and natural surfaces, . The color table for the change maps is reported in Table 13.1.

The scene in Figures 13.2a and 13.2b contained twelve story buildings and no relevant changes were occurred between the two acquisitions. Due to the different solar conditions and height of the buildings, the area covered by shadow presented a different characteristic. The displacement of the objects was caused by the different acquisition

Table 13.1: Color table of the PCC and NAHIRI change maps.

	AFTER		
BEFORE	Vegetation	Man-made	Soil
Vegetation	Gray	Cyan	Orange
Man-made	Green	Gray	White
Soil	Red	Yellow	Gray



(a) (b) (c) (d)

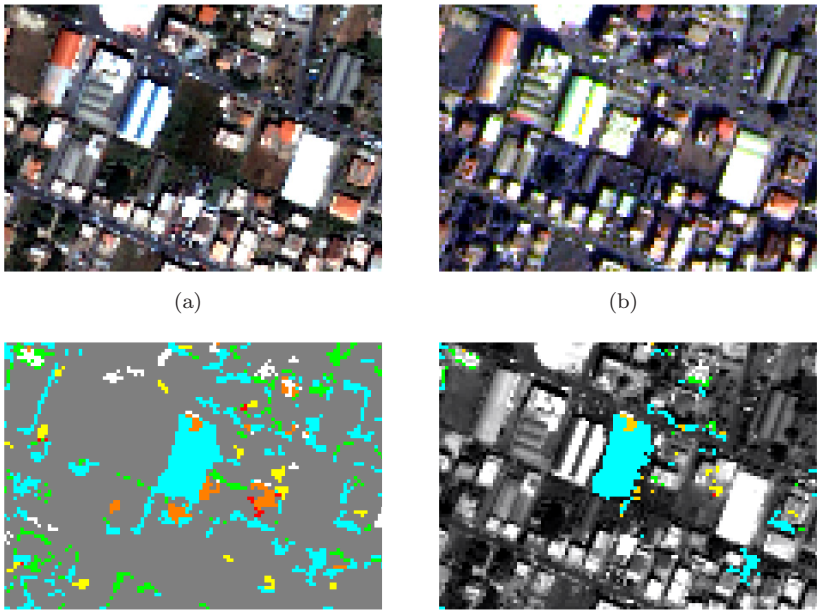
Figure 13.2: Effects due to different acquisition angles and solar conditions: (a) before and (b) after image of the area, while (c) and (d) are the PCC and the NAHIRI products, respectively. Color codes in Table 13.1.

angles and imperfect image registration. The PCC output of the scene is shown in Figure 13.2c. As expected, many false alarms were shown, especially along the edges of buildings. This is a very common problem in the change detection of urban areas. The NAHIRI output is shown in Figure 13.3d, in which the gray-scale satellite image background denotes areas with no changes and colors represent the various class transitions. In this case, the noise present in the PCC output was completely filtered out.

Figure 13.3 shows a real change which is the construction of a new building in the middle of the scene. The height of the buildings did not exceed four stories and the noise resulting from shadows was less evident when compared with Figure 13.2. The PCC output recognized the changed area shown in cyan in Figure 13.3c, but also several false alarms that appeared along the edges of the buildings. The construction of the new building was detected by the NAHIRI map as shown in Figure 13.3d, and at the same time the false alarms were largely filtered out. This exercise demonstrated the effectiveness of the NAHIRI algorithm to distinguish between real changes and false alarms.

No changes were detected between the two image acquisitions shown in Figure 13.4. This area included several buildings and a soccer field in the lower corner and represented well a typical dense urban scene mainly affected by imperfect image registration, commonly considered in the literature as an additional error. Also in this case, the NAHIRI output shown in Figure 13.4d greatly reduced the occurrence of false alarms in the PCC map of Figure 13.4c.

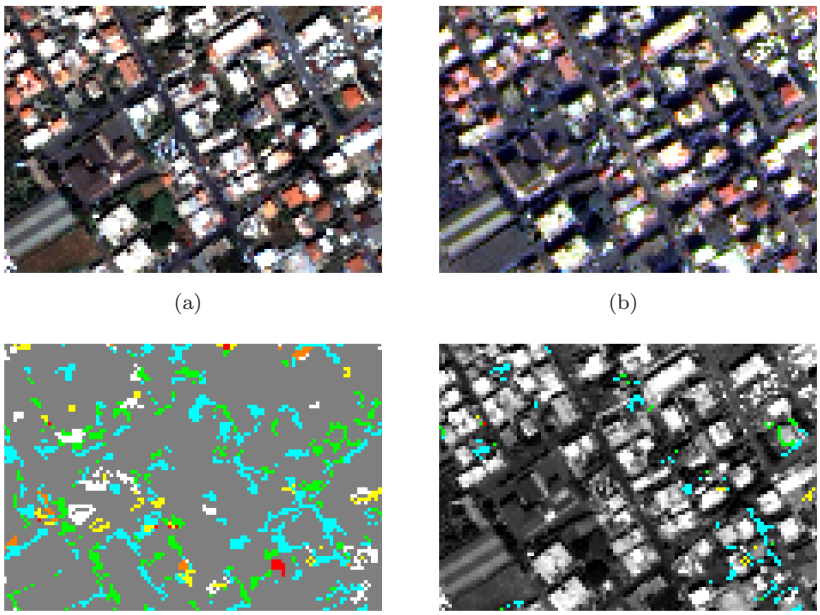
In images with lower resolutions like Landsat, misclassification errors are frequently related to mixed pixel



(a) (b)

(c) (d)

Figure 13.3: Effects due to real changes: (a) before and (b) after image of the area, while (c) and (d) are the PCC and the NAHIRI products, respectively. Color codes in Table 13.1.



(a) (b)

(c) (d)

Figure 13.4: Effects due to imperfect registration of the images: (a) before and (b) after image of the area, while (c) and (d) are the PCC and the NAHIRI products, respectively. Color codes in Table 13.1.

problems, particularly, in high density residential areas, where the mixed pixels can contain portions of buildings, vegetated areas, and roads. These errors give rise to a variety of problems when attempting to map change detection. First, they occur in widely spread single pixels, instead of groups of adjacent pixels, which are easier to detect and correct. Moreover, a pixel including different kinds of surfaces may be randomly attributed to any one of the aforementioned surface types, leading to a noisy change detection pattern that is difficult to interpret

Table 13.2: Comparison of the change detection accuracies between NAHIRI and PCC over the different study areas.

Site Information		Kappa coefficient		Overall Error (%)	
Location	*Spatial Res. (m)*	*NAHIRI*	*PCC*	*NAHIRI*	*PCC*
Tor Vergata Campus, Rome, Italy	2.8	0.783	0.444	5.5	22.2
Rock Creek-Superior, Colorado, U. S. A.	0.6	0.722	0.568	11.9	23.3
Rock Creek-Superior, Colorado, U. S. A.	30	0.811	0.619	5	18.3
	Mean	0.795	0.544	7.5	21.3

and correct.

To assess the performance of NAHIRI when processing lower resolution data, two different exercises were carried out. In one case, NAHIRI was applied to a pair of Landsat images taken over the Colorado area in 1992 and 1996. In the other case, one Landsat image (1996) and one QuickBird image, taken in 2002 over the same area of interest, were used. In this case, NAHIRI was applied after degrading the spatial resolution of the QuickBird image to approximately 30 m, corresponding to the Landsat resolution.

Figure 13.5 shows the area in (a) 1992, (b) 1996 and (c) 2002. The change detection products are shown in Figure 13.5d and Figure 13.5f for changes from 1992 to 1996 and in Figure 13.5g and Figure 13.5i for changes from 1996 to 2002. The maps in Figure 13.5d and Figure 13.5g represent the output of the PCC change detection, for 1992-1996 and 1996-2002, respectively. The maps in Figure 13.5e and Figure 13.5h represent the *Change Mask* stage of NAHIRI. The NAHIRI output is shown in Figure 13.5f and Figure 13.5i, for 1992-1996 and 1996-2002, respectively. The background of these figures has a satellite image gray-scale indicating areas where features are unchanged. Even at the lower resolution, the characteristic filtering effect of NAHIRI emerged and allowed the reduction of the classification error.

The NAHIRI accuracies were evaluated quantitatively against a PCC technique based on a neural network model for the one-step multi-spectral image classification. The relative output maps were subsequently compared in terms of classification accuracies, as shown in Table 13.2. Experimental results, obtained for both very high and high spatial resolution images, confirmed that NAHIRI, unlike other techniques in the literature, is a general approach that can be applied to a wide range of spatial resolutions and land-cover types. The mean of the Kappa coefficients for the PCC method is 0.544, while it is 0.795 for NAHIRI, which correspond to an accuracy increase of about 50%.

13.2 Comparison of neural networks and Bayesian classifier in the parallel approach

The goal of this section is to demonstrate the advantage and the robustness of the NAHIRI parallel architecture when different classifiers, such as neural networks and maximum likelihood, are used to map land-cover changes from a sequence of very high spatial resolution satellite images. The use of the ML approach with the parallel architecture also helps to justify the main factors that make NAHIRI a valuable architecture, due to the well-known statistical background underpinning the ML approach.

13.2.1 Data set

The test area is Rocky Flats, a site dedicated to the production of nuclear arms located immediately to the North West of the city of Denver, Colorado, U. S. A. From 1952 to 1989, the primary mission of Rocky Flats was to build plutonium triggers for nuclear bombs. In 1993, the U. S. Secretary of Energy announced that the site nuclear weapons production was officially over and the site started being cleaned up and dismantled in 1998.

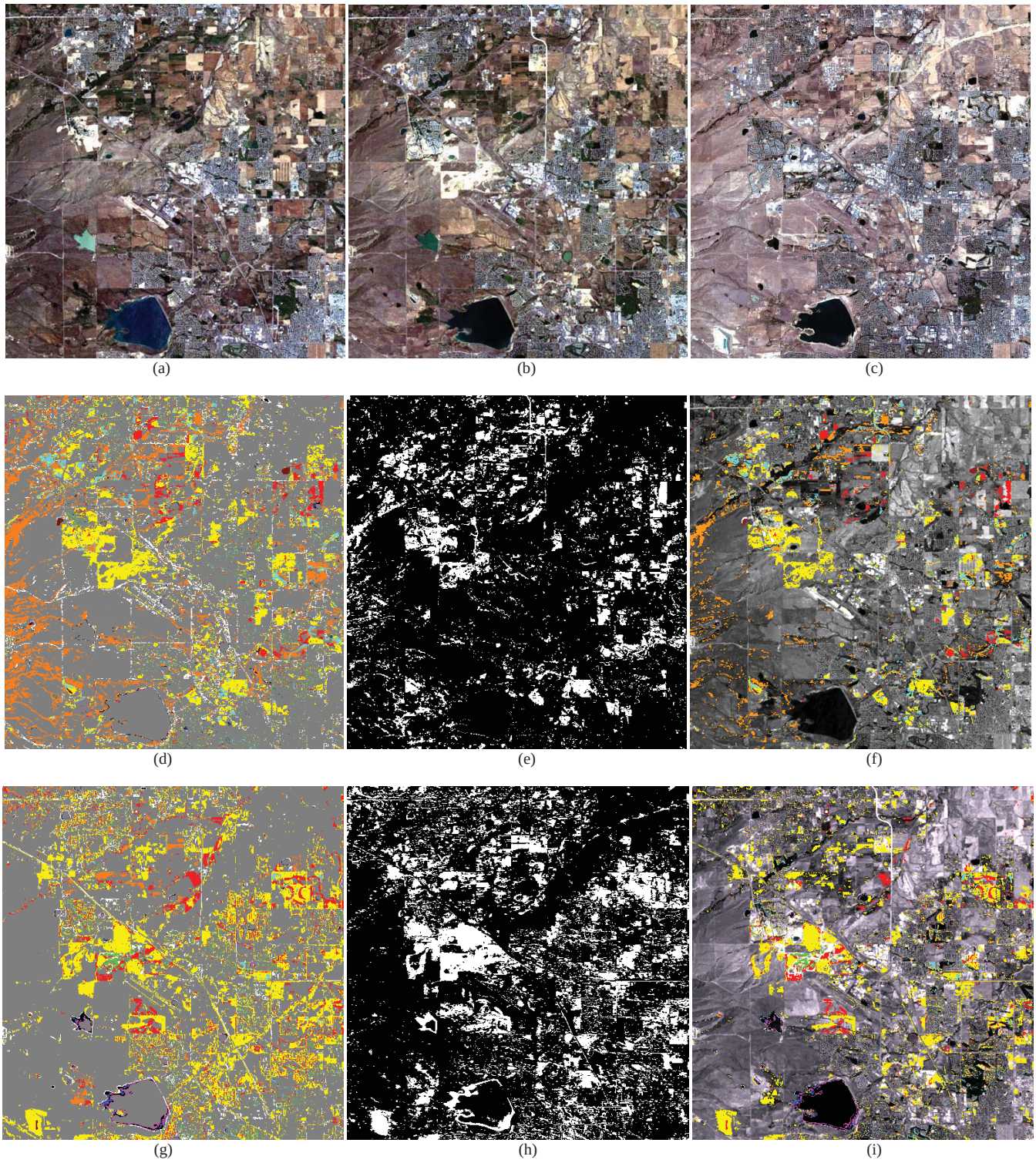


Figure 13.5: Colorado test area in (a) 1992, (b) 1996, and (c) 2002, and change detection between (d) and (f) 1992 and 1996, and (g) and (i) 1996 and 2002. The images in (d) and (g) represent the output of the PCC change detection for 1992-1996 and 1996-2002, respectively. The images in (e) and (h) represent the outputs of the *Change Mask*. The white and black indicate the presence and absence of changes, respectively. In (f) and (i) is shown the NAHIRI output for the 1992-1996 and the 1996-2002 case, respectively. In the background, the gray-level image indicates unchanged features. Color table in Table 13.1

(a) (b)

Figure 13.6: Multi-spectral QuickBird data of the Rocky Foats site imaged in (a) 2003 and (b) 2005. The two figures represent the portion of the images including the former nuclear weapons facilities of Rocky Flats.

Table 13.3: Training pixels.

Classes of Changes		Number of Pixels
1	Buildings to Soil	1,493
2	Steady Soil	3,352
3	Steady Water	380
4	Water to Soil	68
5	Soil to Asphalt	96
6	Asphalt to Soil	960
7	Soil to Water	17

About 800 buildings, some of them very large, were taken down to bare soil. This demolition was completed in mid-2005. The analysis of the Rocky Flats change was carried out using very high spatial resolution QuickBird images (about 2.4 m resolution) acquired on October 23, 2003 and October 15, 2005 (shown in Figure 13.6a and 13.6b, respectively).

13.2.2 Training and test set selection

Seven classes of change were defined, and corresponding training samples were selected by visual inspection of the two images used for classification. The selection of pixels, both for the training and testing phases, was particularly critical, given the poor sampling of some classes in the scene. The training set for each class should be representative of a sufficient number of independent samples, which is not always the case, as, for example, the transition of bare soil to water. The seven classes of change and the corresponding number of pixels selected for training the algorithms are shown in Table 13.3.

To assess the classification accuracy, a set of test pixels was chosen for each class. In order to select a test set that would have both statistical significance and avoids the correlated neighboring pixels, one needs in general to randomly select individual pixels across the image and subsequently label them by visual inspection of the images themselves (random sample). The sampling procedure used to determine the accuracy of the change detection product is dramatically different from sampling the image to assess the accuracy of thematic map products, considering that the changes often represent a very small portion of the whole scene. Therefore, change pixels (or polygons) may not be properly detected with a simple random/systematic sampling (unless the sampling intensity becomes very high).

Using a completely random selection may also lead to some problems. For example, large classes tend to be

Table 13.4: Validation pixels.

Classes of Changes		Number of Pixels
1	Buildings to Soil	74
2	Steady Soil	342
3	Steady Water	55
4	Water to Soil	28
5	Soil to Asphalt	34
6	Asphalt to Soil	126
7	Soil to Water	37

represented by a much larger number of samples than the smaller ones. Here, some very small classes (i.e. soil to water) were not represented at all. To ensure that small classes were also represented adequately, the SRS method was adopted. According to it, a preliminary classification was first applied to stratify the image into seven classes, and then the test pixels were randomly sampled for each of those classes. In this way, a total of 696 test pixels were selected as shown in Table 13.4.

13.2.3 Results

To assess the robustness of the architecture to different classification algorithm, a comparison between neural networks and maximum likelihood classifiers, both using NAHIRI, was carried out. The results were compared to a standard DMC approach based on both NN and ML, which represented a sort of benchmark.

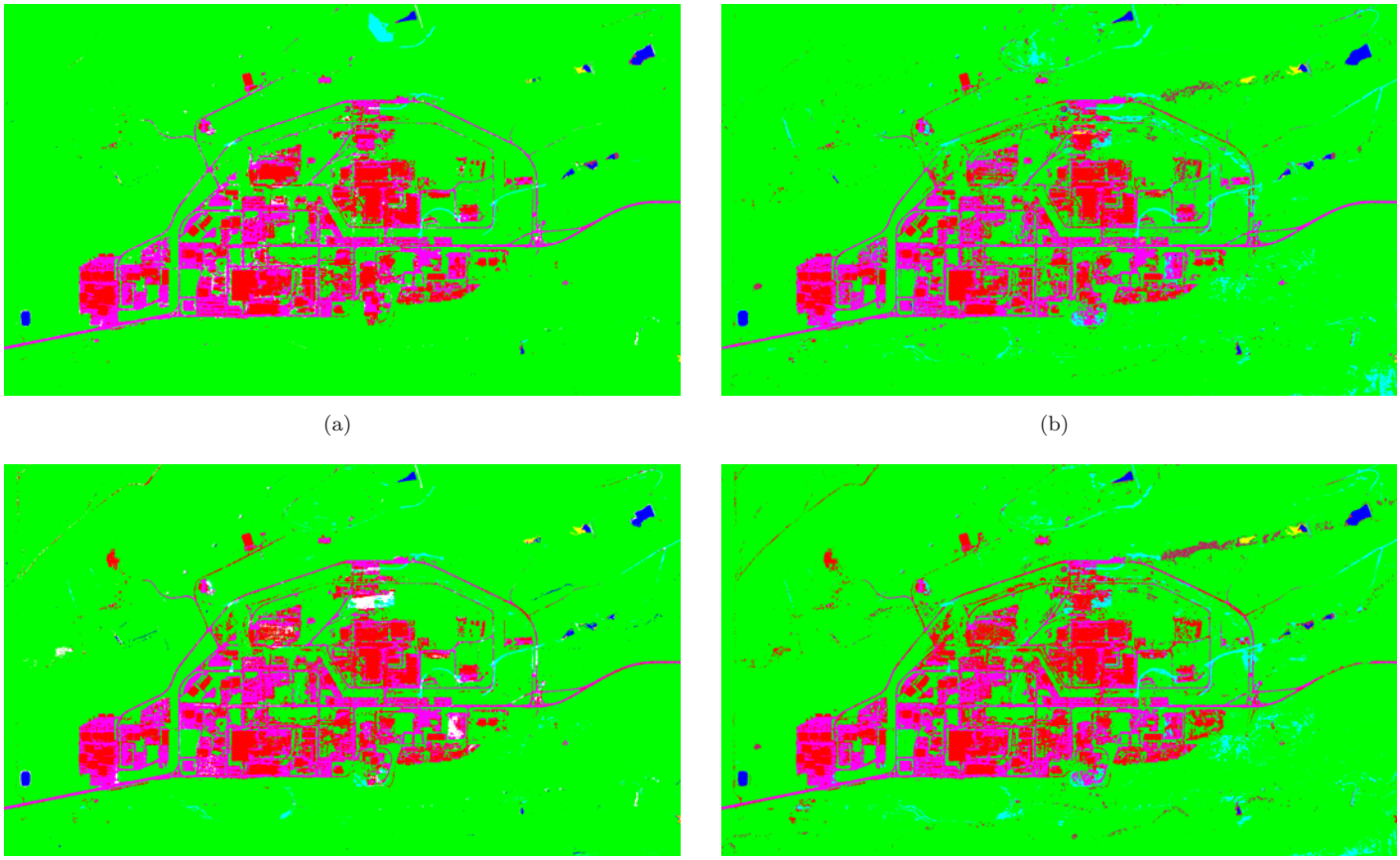
Note that as far as inputs to the classifiers are concerned, the multi-spectral and multi-temporal information from the satellite images can be exploited in different ways. For example, the radiances can be considered alone, but also band ratios or additional derived features (e.g., texture). In this study, the original channels (radiances) provided by the satellite and the normalized difference vegetation index (NDVI), which combines near infrared and red radiances, were used. Therefore, the input information consists globally of ten features, i.e., the eight channels of the QB satellite for the two acquisition dates and the two NDVI values.

The scene was initially classified using the NN-based NAHIRI as described in the previous section, using two network topologies (5-12-12-4 and 10-14-14-2) for the multi-spectral and the multi-temporal stage of Figure 13.1. The same training strategy was applied to all neural networks, where the cost function of the learning phase was minimized according to the scaled conjugate gradient algorithm.

As for the case of the NN classifiers, the two multi-spectral ML classifiers exploited a 5-element input feature vector each and N_{cl} classes, whilst the multi-temporal ML used a feature vector with 10 elements and $N_{cl} = 2$ classes. In this case, a sufficient number of training samples for each spectral class must be available to allow one to accurately estimate the elements of the mean vector and the covariance matrix of each class. For an F-dimensional multi-spectral space, at least $F + 1$ samples are required to avoid the covariance matrix being singular.

Concerning the training set, the same training pixels selected for the seven classes of change (see Table 13.3) were used for all the classifiers. They were rearranged differently to train the multi-spectral and multi-temporal stages of NAHIRI. As for the multi-spectral block, the initial or the final class was retained to classify the first or the second image, respectively. For instance, change classes 2, 5 and 7 were associated with Soil in the first image. As for the multi-temporal branch, all pixels belonging to the five classes of change and those corresponding to the unchanged conditions were merged together to obtain only two classes, i.e., Changed and Unchanged. While the original NAHIRI yields a unique class of No Change, in this section the class of unchanged pixels was estimated in a different way.

It should be noted that NAHIRI may produce more classes of change than those recognized by visual inspection during the early training phase. These additional classes were grouped into a unique No Relevant label, which



(a) (b) (c) (d)

Figure 13.7: Classification results using (a) NAHIRI-NN, (b) DMC-NN, (c) NAHIRI-ML and (d) DMC-ML. Green: bare soil unchanged; Red: house to bare soil; Magenta: asphalt to bare soil; Cyan: bare soil to asphalt; White: no relevant.

Table 13.5: NAHIRI-NN confusion matrix.

Classified as	True Class								
	1	2	3	4	5	6	7	NR	TOT
1	65	5	0	0	0	24	0	0	94
2	4	322	2	12	7	9	12	0	368
3	0	1	43	4	0	0	0	0	48
4	0	0	0	10	0	0	0	0	10
5	0	0	2	0	24	0	0	0	26
6	5	5	0	1	0	91	0	0	102
7	0	3	5	1	3	0	25	0	37
NR	0	6	3	0	0	2	0	0	11
TOT	74	342	55	28	34	126	37	0	696
Accuracy=83.3%					**Kappa coefficient=0.758**				

was representative of either a misclassification or an omission error of the visual interpreter.

The final classification product of the NN-based NAHIRI (here named *NAHIRI-NN*) is displayed in Figure 13.7a. The performance of the algorithm was evaluated using the test set selected randomly as described previously, resulting in an overall accuracy (number of correctly classified pixels) of 83.3% and a Kappa coefficient equal to 0.758, as shown in Table 13.5. Note that the confusion matrix, also reported in Table 13.5, is composed of eight (instead of seven) rows and columns, to take into account the presence of eleven pixels belonging to the Not Relevant class.

In order to asses if the set of test pixels was adequate to evaluate the classification accuracy, the minimum

Table 13.6: DMC-NN confusion matrix.

Classified as	True Class								
	1	2	3	4	5	6	7	NR	TOT
1	55	6	0	0	0	14	0	0	75
2	10	279	3	5	8	21	10	0	336
3	0	1	50	3	0	0	9	0	63
4	0	0	0	18	0	1	0	0	19
5	0	34	2	0	26	4	1	0	67
6	8	6	0	2	0	86	0	0	102
7	1	16	0	0	0	0	17	0	34
NR	0	0	0	0	0	0	0	0	0
TOT	774	342	55	28	34	126	37	0	696
Accuracy=76.3%							**Kappa coefficient=0.666**		

required sample size was computed. There are different approaches to establish it, as discussed in Richard [91]. For Van Gerdener *et al.* [192], the probability of sampling k pixels and finding them all correctly classified in a map having an accuracy of θ is given by a binomial distribution. Limiting such probability below a certain threshold (i.e., 0.05%) requires k to be above a value which is function of θ itself. Rosenfeld *et al.* [193] defined the number of samples required to guarantee that the percentage of correct classification approximates within 10% the true accuracy of each class at the 95% of confidence level. The final results in [192] indicate a sample size increasing from 5 to 60 pixels when the true classification accuracy goes from 0.5 to 0.95. Conversely, in [193] a number of test pixels ranged between 80 and 19 for the same interval of θ. Although these two trends are opposite, they are in agreement for a classification accuracy of 85%, both indicating a minimum sample size of 20 pixels. In this study, the mean classification accuracy was in the order of 80-85% and the minimum value required for the test set was generally fulfilled, as apparent in Table 13.4. The more critical condition concerned the class Water to Soil, whose correct classification was on the order of 50% and the number of test pixels was only 28. Otherwise, the test set can be considered reliable enough.

The previous results were compared to a standard DMC approach using a single NN which combined the multi-temporal and the multi-spectral signatures. To this aim, a NN was designed and fed by the ten input features previously described and composed by two hidden layers of twenty-four units each, resulting in a 10-24-24-7 topology. Also in this case, the cost function of the learning phase was minimized according to the scaled conjugate gradient algorithm using the same training pixels illustrated previously. The classification map obtained by this procedure (here called *DMC-NN*) is shown in Figure 13.7b, resulting in an overall accuracy of 76.3% (Kappa coefficient: 0.666) as shown in Table 13.6.

In this case, the improvement of NAHIRI with respect to the single NN classification was about 14% in terms of Kappa coefficient. Moreover, the resulting NAHIRI map clearly shows by visual inspection less isolated and miss classified pixels with respect to the other one.

Subsequently, the same two experiments illustrated previously were performed by simply replacing the NN classifier with the ML classifier, and preserving exactly the same conditions in terms of training and test pixels and input feature space. The outcomes of these two classification processes are shown in Figure 13.7c and Figure 13.7d, respectively. An overall accuracy of 76.3% (Kappa coefficient: 0.654) was achieved in the case of the NAHIRI architecture (here called *NAHIRI-ML*, shown in Figure 13.7(c)) to be compared to 67.4% (Kappa coefficient: 0.585) for the ML classification (here called *DMC-ML*, shown in Figure 13.7(d)). In this case, it was obtained an enhancement of about 12% in accuracy in terms of Kappa coefficient using NAHIRI-ML instead of DMC-ML.

The increase of the classification accuracy associated with NAHIRI seemed to be independent on the specific classification algorithm used, demonstrating the inherent value of the approach. The better results of the NN

Table 13.7: NAHIRI-ML confusion matrix.

Classified as	True Class								
	1	2	3	4	5	6	7	NR	TOT
1	61	10	0	0	0	18	0	0	89
2	3	301	5	8	16	8	34	0	375
3	0	1	47	3	0	0	0	0	51
4	0	0	0	14	0	1	0	0	15
5	0	16	0	0	18	4	0	0	38
6	7	4	0	3	0	88	0	0	102
7	0	0	0	0	0	0	2	0	2
NR	3	10	3	0	0	7	1	0	24
TOT	74	342	55	28	34	126	37	0	696
Accuracy=76.3%						**Kappa coefficient=0.654**			

Table 13.8: DMC-ML confusion matrix.

Classified as	True Class								
	1	2	3	4	5	6	7	NR	TOT
1	59	16	0	0	0	27	0	0	102
2	2	194	2	0	8	4	1	0	211
3	0	1	48	2	0	0	1	0	52
4	0	1	0	23	0	1	0	0	25
5	0	70	1	0	26	9	1	0	107
6	12	6	0	1	0	85	0	0	104
7	1	54	4	2	0	0	34	0	95
NR	0	0	0	0	0	0	0	0	0
TOT	74	342	55	28	34	126	37	0	696
Accuracy=67.4%						**Kappa coefficient=0.585**			

implementation, as compared to the ML one, was expected considering the data driven nature of the former as opposed to the parametric nature of the latter, which assumes a Gaussian distribution of the data (not always matching the real statistics). Most of the errors of ML were due to misclassification of steady soil surfaces.

Richard and Lippmann [194] demonstrated theoretically the relationship between Bayesian probabilities and the outputs of NNs. Their simulations demonstrated that for an M class problem, NNs estimate Bayesian probabilities with high accuracy when squared-error cost functions with sigmoidal non-linearities are considered and the training data set is adequate, reflecting the actual likelihood distribution and a priori class probabilities [195]. Therefore, the considerations regarding the role of the ML-based architecture can be also extended to the NN-based scheme, which corresponds to the NAHIRI scheme.

13.2.4 The fuzzy NAHIRI processing scheme

The disagreement between the multi-spectral and multi-temporal branches can be related to two different cases:

a the pixel transition is recognized by the *Change Mask* but not by the *Change Map*

b the *Change Mask* does not detect any change but the pixel is differently classified in MAP_1 and MAP_2

In the case of a disagreement, NAHIRI assumes no change in the pixel. For case (a) there is no ambiguity, considering that MAP_1 and MAP_2 provide the same results. Therefore, it is irrelevant which reference map is used for assessing the unchanged class. For case (b) NAHIRI labels the corresponding pixel on the basis of the classification map of the oldest image, without taking into account which classified map has less uncertainty. A different strategy can be adopted for case (b), as shown in the flowchart of Figure 13.8, where the reference map is not decided *a priori*, but it is dynamically evaluated on the basis of the output of the activation functions of MAP_1 and MAP_2.

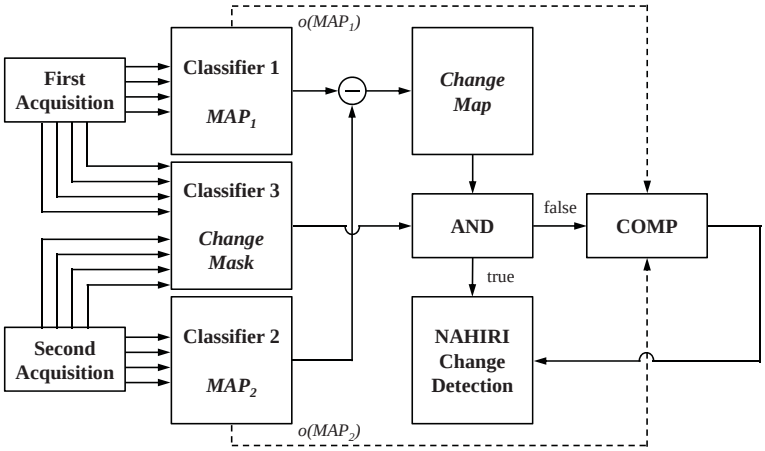


Figure 13.8: Schematic block of the fuzzy NAHIRI scheme.

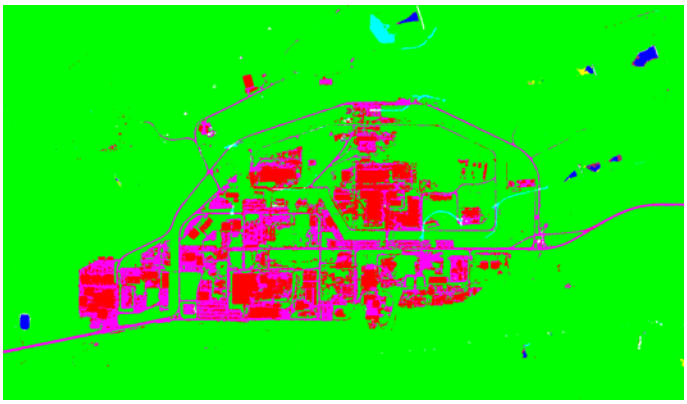
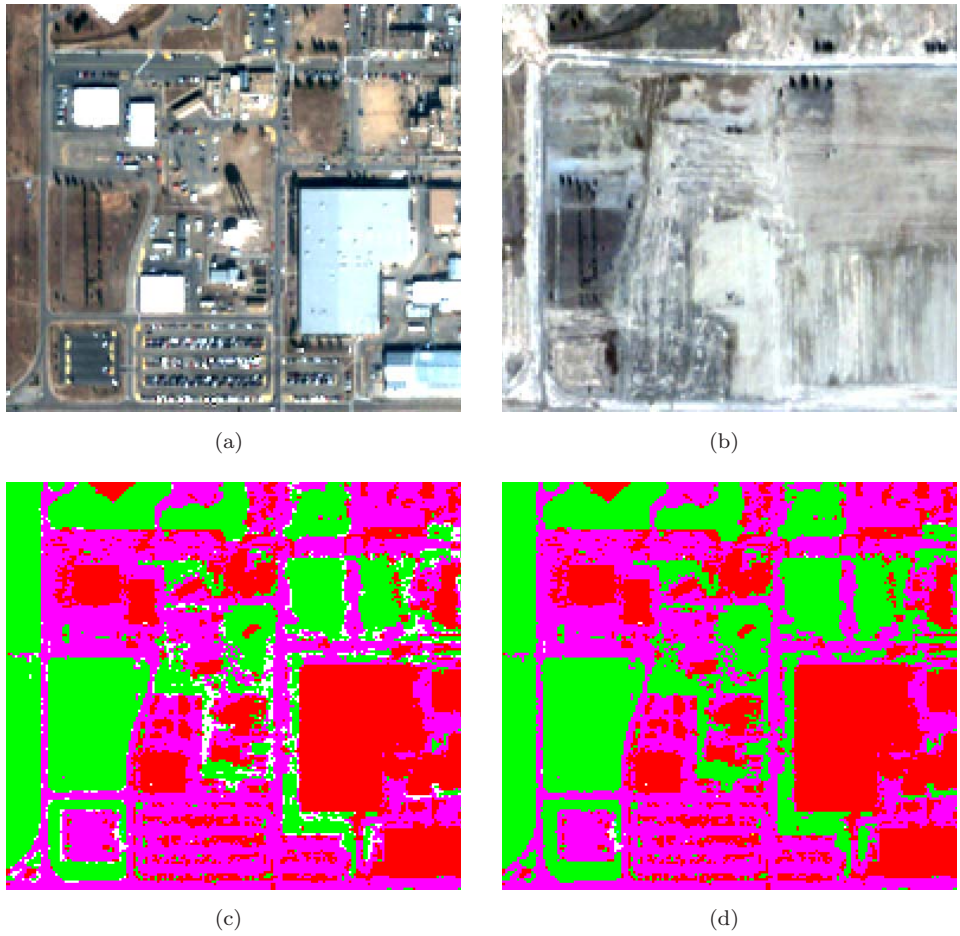


Figure 13.9: Classification results using the fuzzy NAHIRI scheme with NNs. Green: bare soil unchanged; Red: house to bare soil; Magenta: asphalt to bare soil; Cyan: bare soil to asphalt; White: no relevant.

Table 13.9: Fuzzy NAHIRI-NN confusion matrix.

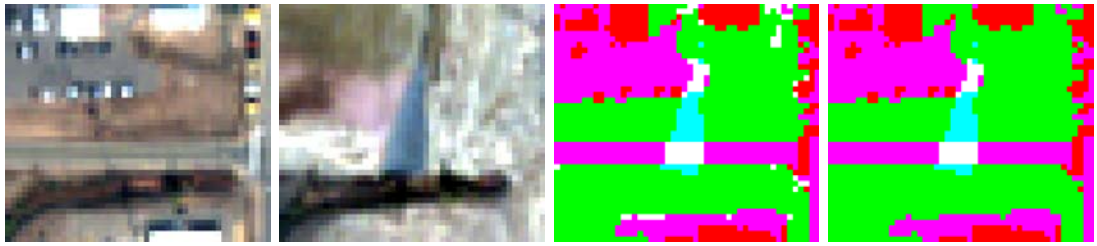
Classified as	True Class								
	1	2	3	4	5	6	7	NR	TOT
1	65	5	0	0	0	24	0	0	94
2	4	324	1	13	7	10	12	0	371
3	0	1	45	3	0	0	0	0	49
4	0	0	0	10	0	0	0	0	10
5	0	0	2	0	24	0	0	0	26
6	5	5	0	1	0	91	0	0	102
7	0	3	5	1	3	0	25	0	37
NR	0	4	2	0	0	1	0	0	7
TOT	74	342	55	28	34	126	37	0	696
Accuracy=83.9%						**Kappa coefficient=0.765**			

This new architecture takes into account the likeliest classification in the two dates by introducing a further decision block which compares the outputs of *Classifier 1* and *Classifier 2* (i.e., the activation functions for NAHIRI-NN, i.e. $o(Map1)$, or the posterior probability for NAHIRI-ML) in order to select the class with the highest value. In Figure 13.9 the final change detection map provided by the new NAHIRI scheme is shown. The related confusion matrix and overall accuracy are reported in Table 13.9. The Kappa coefficient increased from 0.758 to 0.766 with respect to the previous architecture.



(a) (b) (c) (d)

Figure 13.10: Detail of (a) 2003, (b) 2005 images and classification results using (c) the original NAHIRI and (d) the Fuzzy approach. Green: bare soil unchanged; Red: house to bare soil; Magenta: asphalt to bare soil; Cyan: bare soil to asphalt; White: no relevant.



(a) (b) (c) (d)

Figure 13.11: Detail of (a) 2003, (b) 2005 images and classification results using (c) the original NAHIRI and (d) the Fuzzy approach. Green: bare soil unchanged; Red: house to bare soil; Magenta: asphalt to bare soil; Cyan: bare soil to asphalt; White: no relevant.

Although the increase of Kappa coefficient is not too marked, the improvement of the new NAHIRI scheme is made evident by looking at the decreased number of pixels belonging to the Not Relevant class. They changed from 7,779 to 1,613 in the case of classical and Fuzzy NAHIRI approaches, respectively. Details are reported in Figure 13.10 and Figure 13.11, where white pixels, pertaining to the Not Relevant class and probably originating from mixed pixels along the edges of the objects, almost disappeared.

Table 13.10: Number of training pixels and input features in *Classifier 1* and *Classifier 2*.

Cover Classes	Training Pixels (2003)	Training Pixels (2005)	F
Buildings	1,493	0	5
Soil	3,465	5,873	
Water	448	397	
Asphalt	960	96	

Table 13.11: Number of training pixels and input features in *Classifier 3*.

Classes CD	Number of Pixels	F
No Change	3,732	10
Change	2,634	

It is worth noting in Figure 13.11 that NAHIRI was able to detect a change which was not identified by visual inspection and thus was not introduced in the training set. Namely, the image of 2005 displayed bare soil and a drainage channel (belonging to the Asphalt class). Part of the latter appeared in white in the classification. The white pattern represented the portion of the drainage channel built in regions where a road and parking lots were present in 2003. In this case, the white regions did not indicate an error of the classifier, which actually correctly recognized a steady Asphalt class not recognized by the visual inspection.

13.3 Conclusions

In this chapter, an image analysis technique based on NN architecture was discussed. NAHIRI, unlike other methods, is a general approach that can be applied to a wide range of spatial resolutions and land-cover types, as the method was applied to different test areas including different landscapes, such us urban, open space, and rural. The distinctive feature and the major innovation of this parallel approach is the presence of three classifiers that simultaneously exploit both the multi-spectral and the multi-temporal information.

The use of ML (due to the well-known statistical considerations) with the parallel architecture helped to explain the role of the NAHIRI scheme and the reasons why this architecture performed better than other methods proposed in the literature. In fact, the increase in dimensionality of the data worsens the estimation of the parameters and overcomes the increase in separability among classes associated with additional features. Therefore, one needs to use a robust way for estimating parameters or to reduce F. For instance, principal component analysis can be used to diminish the number of features [91]. Regularization methods can be also exploited. They mainly try to stabilize the estimated class covariance matrix by replacing it with a weighted sum of the class sample covariance matrix or the common (pooled) covariance matrix. Numerous methods of regularizing covariance evaluation, including regularized discriminant analysis [196] and leave-one-out covariance estimation [197] have also been proposed.

The NAHIRI approach is different from those mentioned above. In fact, the number of features is not reduced, but they are differently organized in the new architecture to reduce the difficulty to have poor training samples for some classes. The architecture introduces more classifiers, thus reducing their respective data dimensionality. Starting from a certain number of classes of change in the scene, the multi-spectral task uses a single image, thus reducing both the number of features and limiting itself to classify only land-cover classes, instead of class of changes. The multi-temporal task uses a bigger feature vector, but it reduces drastically the number of classes to only two, thus enabling to group the corresponding training pixels. In Table 13.10 and Table 13.11 the number of training pixels available for each class is noticeably increased for each of three classifiers in the NAHIRI scheme, in comparison with those available for each individual class of change (see Table 13.3).

Chapter 14

Automatic change detection with PCNNs

Part of this Chapter's contents is extracted from:

1. F. Pacifici and F. Del Frate, "Automatic change detection in very high resolution images with pulse-coupled neural networks", *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 1, pp. 58-62, January 2010

2. F. Pacifici, W. J. Emery, "Pulse Coupled Neural Networks for Automatic Urban Change Detection at Very High Spatial Resolution", in *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, E. Bayro-Corrochanpo, J. Eklundh, Eds. Lecture Notes in Computer Science, Springer, vol. 5856, pp. 929-942, Springer, ISBN: 978-3-642-10267-7, 2009

A novel change detection application based on Pulse Coupled Neural Networks is presented in this chapter. As illustrated in Chapter 3, PCNN are based on the implementation of the mechanisms underlying the visual cortex of small mammals and have interesting advantages with respect to more traditional neural network architectures, such as multi-layer perceptron. In particular, they are unsupervised and context sensitive.

The development of fully automatic change detection procedures for very high resolution images is not a trivial task as several issues have to be considered. As discussed in the previous chapter, the crucial difficulties include possible different viewing angles, mis-registrations, shadows and other seasonal and meteorological effects which combine to reduce the accuracy attainable with the change detection methods. However, this challenge has to be faced to fully exploit the big potential offered by the ever-increasing amount of information made available by ongoing and future satellite missions.

When dealing with such large amounts of data, supervised methods risk to become unsuitable and there is an urgent need to develop novel processing techniques for knowledge discovery in a fully automatic mode. This is even more compelling if the applications are dedicated to the monitoring of urban sprawl are considered. In these cases, the big potential provided by very high spatial resolution images has to be exploited for analyzing large areas, which would be unfeasible if completely automatic procedures are not taken into account.

PCNNs can be used to individuate, in a fully automatic manner, the areas of an image where a significant change occurred. In particular, the time signal $G[n]$, computed by:

$$G[n] = \frac{1}{N} \sum_{ij} Y_{ij}[n] \tag{14.0.1}$$

was shown to have properties of invariance to changes in rotation, scale, shift, or skew of an object within the scene (see Chapter 3). This last feature makes PCNNs a suitable approach for change detection in very high spatial resolution imagery, where the view angle of the sensor may play an important role.

In particular, the waves generated by the time signal in each iteration of the algorithm create specific signatures of the scene which are successively compared for the generation of the change map. This can be obtained by measuring the similarity between the time signals associated with the former image and the one associated with the latter. A rather simple and effective way to do this is to use a correlation function operating between the outputs of the PCNNs.

The proposed method is completely automated since it analyzes the correlation between the time signals associated with the original images. This means that no pre-processing, except for image registration, is required. Furthermore, PCNNs may be implemented to exploit simultaneously both contextual and spectral information which make them suitable for processing any kind of sub-meter resolution images.

This chapter is organized as follows. The performance of the algorithm is evaluated on different multi-spectral and panchromatic satellite sensors in Section 14.1, where qualitative and more quantitative results are reported. Conclusions are in Section 14.2

14.1 Experimental results

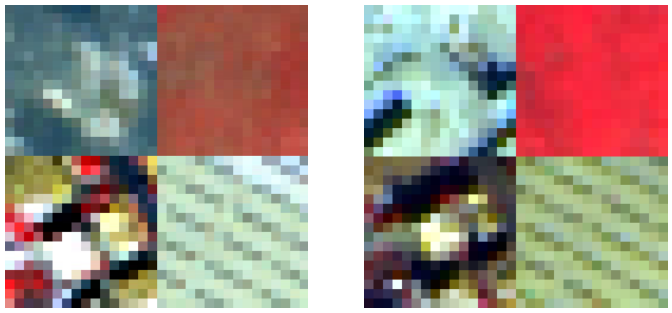
14.1.1 The time signal G[n] in the multi-spectral case

To investigate the time signal $G[n]$ of the satellite data, two images acquired by QuickBird on May 29, 2002 and March 13, 2003 over the Tor Vergata University campus (Rome, Italy) were exploited. Specifically, multi-spectral images (about 2.4 m resolution) were used to have a better understanding of the PCNN pulsing phase when applied to different bands. In this case, $N = 16 \times 16$ pixels.

Four different conditions, shown with false colors in Figure 14.1, were considered: (UL) large changes, (UR) change in vegetation cover, (DL) small changes and (DR) no-changes. The first area, large changes, represents the construction of a new commercial building. As shown in Figure 14.2a, from the very first epochs the pulsing activity of the two images is relatively different, especially if the waveform is concerned. The change in vegetation cover is illustrated in Figure 14.2b. During the first few epochs, waveform and time dependence of the two signals appear to be similar. For successive epochs, this correlation decreases, especially due to the well known behavior of the near infrared band. The time signal for small changes, i.e. when the changed pixels represent a fraction of a sub-image, is shown in Figure 14.2c. During the first epochs, waveforms show slight differences, while the time correlation seems to get lost faster than inthe previous example. Finally, for the no-changes case shown in Figure 14.2d, it is possible to note that during the initial epochs both the waveform and the time dependence of the two signals appear to be highly correlated.

Different values can be obtained considering different epoch intervals. This is concisely expressed in Figure 14.2, where some correlation values obtained for specific epochs are reported. In particular, it seems to not be useful to use a high number of epochs since it is not possible to completely distinguish different land changes. On the other hand, the information derived only from the first oscillation (epochs 5-11) appears to be valuable since it allows the discrimination of various land changes.

From this analysis, it seems that PCNNs, once processing an image pair, might be capable of automatically catching those portions of the image where changes occurred. In such a context, an approach based on *hot spot* detection rather than on changed-pixel detection may be more appropriate given the size of the targets (generally buildings) and the huge volume of data that it may be necessary to analyze in the near future. However, the implementation of a pixel based approach with PCNNs is straightforward, using a window sliding one pixel at the time.



(a) (b)

Figure 14.1: Multi-spectral QuickBird images (a) 2002 and (b) 2003 shown with false colors: (UL) large changes, (UR) change in vegetation cover, (DL) small changes and (DR) no-changes.

The accuracy of PCNNs in change detection was evaluated more quantitatively by applying PCNNs to the QuickBird imagery of Tor Vergata University. The panchromatic images were used in order to design a single PCNN working with higher resolution (0.6 m) rather than 4 different ones processing lower resolution images (2.4 m). The two panchromatic images are shown in Figure 14.3a and Figure 14.3b. Few changes occurred in the area during the time window analyzed. The main ones correspond to the construction of new commercial and residential areas. A complete ground reference of changes is reported in Figure 14.3c. Note that the ground reference included also houses that were already partially built in 2002.

The size of the PCNN was 100×100 neurons. For the reasons explained previously, it was decided to only look for *hot spots* where a change could be rather probable. To operate in this way, the PCNN output values were averaged over the $10,000$ neurons belonging to 100×100 boxes. An overlap between adjacent patches of 50 pixels (half patch) was considered. Increasing the overlap would mean having greater spatial resolution at the price of an increase in the computational burden. Considering this study was aimed at detecting objects of at least some decades of pixels (such as buildings), an overlap of 50 pixels was assumed to be a reasonable compromise. The computed mean correlation values were then used to discriminate between changed and not changed areas.

The PCNN results are shown in Figure 14.3d, while in Figure 14.3e, for the sake of comparison, the image difference result is reported. More in detail, in this latter case, an average value was computed for each box of the difference image and a threshold value was selected to discriminate between changed and not changed areas. In particular, the threshold value was chosen to maximize the number of true positives, keeping reasonably low the number of false positives.

What should be noted first is that, at least in this application, the PCNN algorithm did not provide any intermediate outputs, with the correlation values very close to 0 or 1. This eliminated the search for optimum thresholds of the final binary response. The accuracy is satisfactory, as 49 out of the 54 objects appearing in the ground reference were detected with no false-alarms. The missed objects were structures already present in the 2002 image (e.g., foundations or the first few stories of a building), but not yet completed. On the other side, the result given by the image difference technique, although a suitable threshold value was selected, is rather imprecise, presenting a remarkable number of false alarms.

The image shown in Figure 14.3f was obtained by a multi-scale procedure. This consisted in a new PCNN processing, this time on a pixel basis, of one of the hot spots generated with the first computation. In particular, it shows the change area corresponding to the box indicated by the "$*$" in Figure 14.3d. It can be noted that the output reported in Figure 14.3f is more uniformly distributed within the range between 0 and 1. Its was multiplied

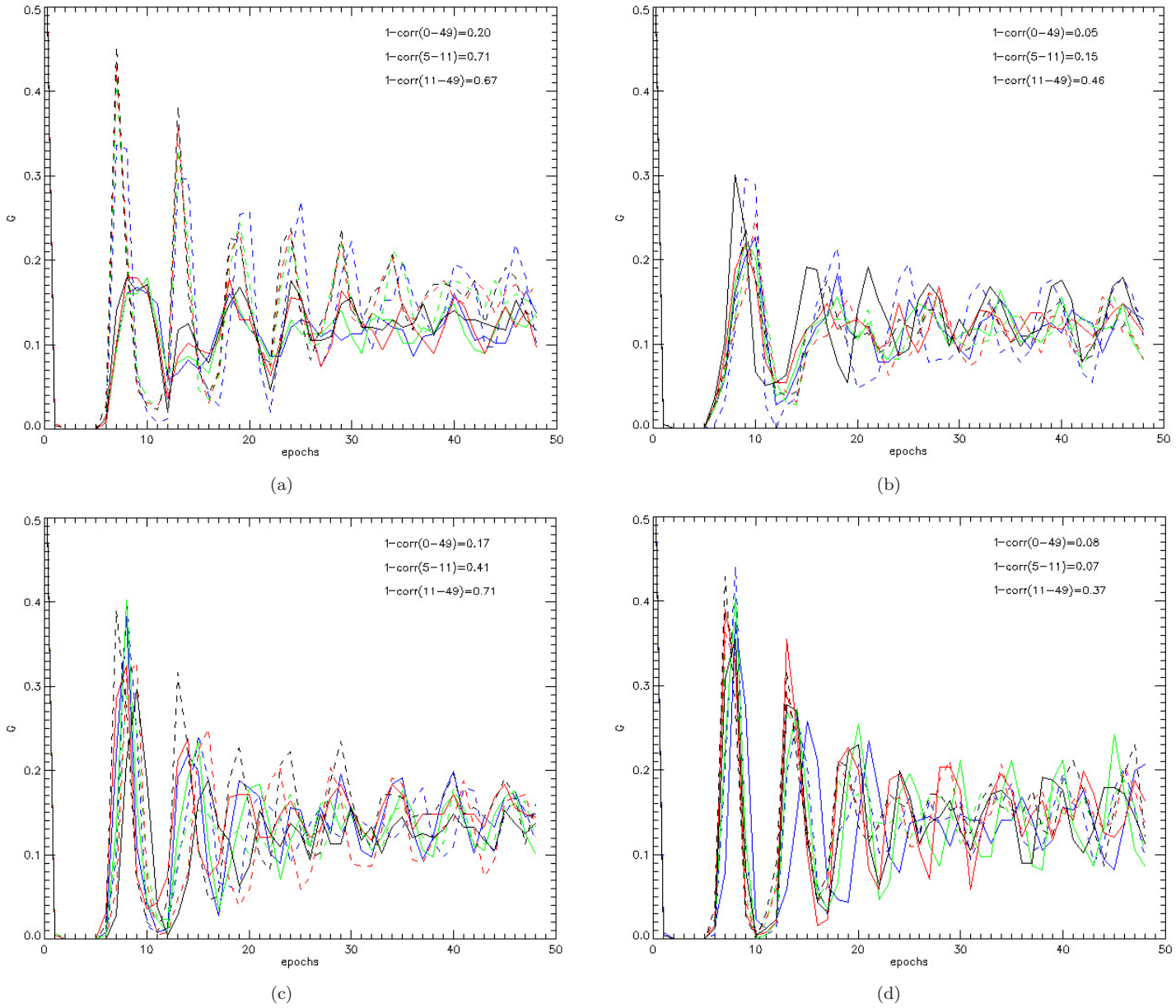
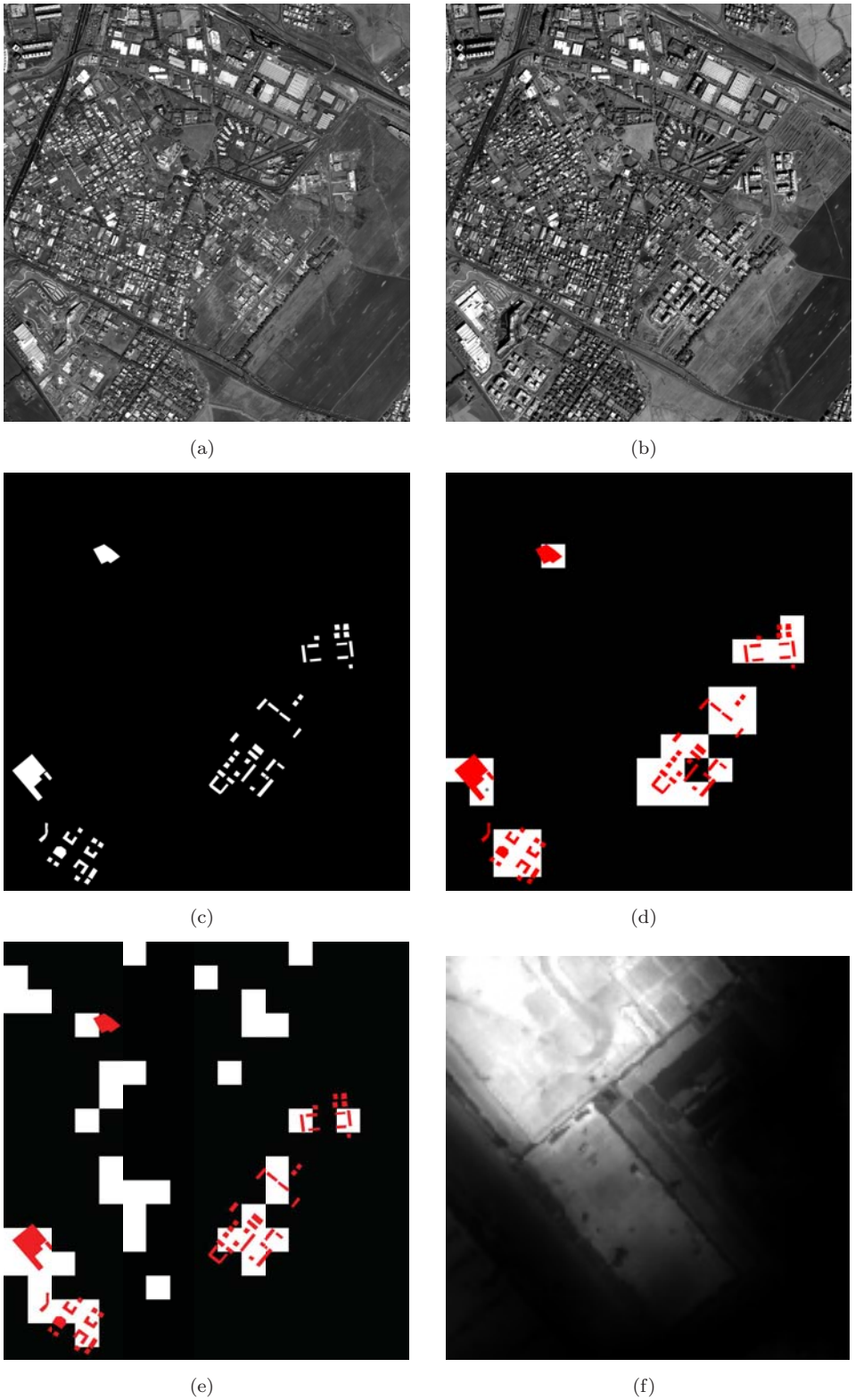


Figure 14.2: The pulsing activity of the two images for the four cases considered in Figure 14.1. UL case is reported in (a), UR in (b), DL in (c) and DR in (d). Continuous lines represent the 2002 image, while dotted lines correspond to the 2003 image. Red = red channel. Green = green channel, Blue = blue channel, Black = near infrared channel.

with the panchromatic image taken in 2003 to have a result which better exploits the very high resolution property of the original image.

14.1.2 Automatic change detection in data archives

This study area included the suburbs of Atlanta, Georgia (U. S. A.). The images were acquired by QuickBird in February 26, 2007 and by WorldView-1 in October 21, 2007 for an approximate area of 25 km^2 ($10,000 \times 10,000$ pixels). The size of this test case represented an operative scenario where PCNNs gave evidence of their potentialit in detecting hot spot areas in a large data archive. The two panchromatic images are shown in Figure 14.4a and Figure 14.4b, respectively. The ground reference of changes is highlighted in Figure 14.4a and Figure 14.4c



(a) (b) (c) (d) (e) (f)

Figure 14.3: Panchromatic image of the Tor Vergata University in (a) 2002 and (b) 2003 and (c) the relative ground reference. Change detection result obtained by: (d) the PCNN elaboration and (e) the standard image difference procedure. In (f) is shown the PCNN pixel-based analysis carried out over one of the previously detected changed areas indicated with "$*$" in (d).

in red. Note that the ground reference included also houses that were already partially built during the first acquisition. Many changes occurred although the small time window, mainly corresponding to the construction of new commercial and residential buildings.

As shown in Figure 14.4c, PCNN confirmed to have good capabilities in the automatic detection of the hot spots corresponding to areas which underwent changes due to the construction of new structures. Also in this test case, where the images had comparable viewing angles, PCNNs did not provide any intermediate outputs, with the correlation values alternatively very close to 0 or 1. This avoided a search for optimum thresholds to be applied to the final binary response.

The accuracy is satisfactory, as 30 out of the 34 objects appearing in the ground reference were detected with 6 false alarms, mainly due to the presence of leaves on the trees in the WorlView-1 image. The missed objects are basically structures that were already present in the first acquisition (e.g., foundations or the first few stories of a building) but not completed yet, or small isolated houses. Details of the detected hot spots (including a false alarm) are shown in Figure 14.4d.

14.1.3 Automatic change detection in severe viewing conditions

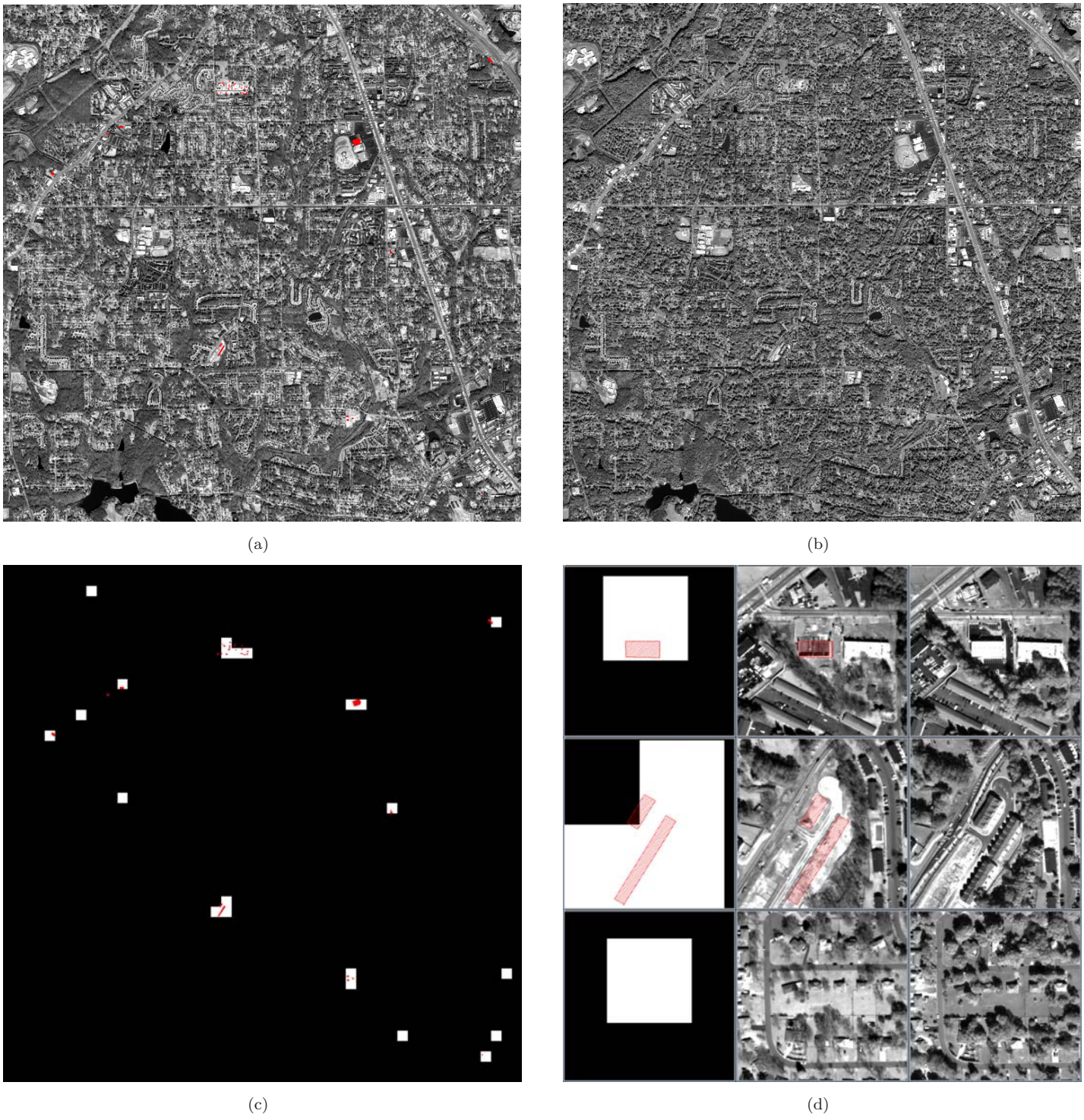
This study area is part of Washington D. C. (U. S. A.). The images were acquired by QuickBird in September 23, 2007 and by WorldView-1 in December 18, 2007 for an approximate area of 9 km^2 ($7,000 \times 5,000$ pixels). In this case, the images were acquired with very different view angles to investigate the performance of PCNNs under these particular conditions. The images are shown in Figure 14.5a and Figure 14.5b, respectively. Only one change occurred in the area due the small time window, corresponding to the demolition of a building (highlighted in red in Figure 14.5a and Figure 14.5c).

As shown in Figure 14.5c, PCNN detected correctly the only hot spot of change. Differently from the previous case, where values were close to 0 or 1, non-changed areas show correlation values slightly larger than 0. This may be expected due to the very different view angles of the imagery used. For example, the same building is viewed from different directions, occluding different portions of the scene, such as roads or other buildings. However, PCNNs appear to be robust enough against these problems as shown in the plot of Figure 14.6. Here, on the y-axis, the number of pixels associated to the same measured correlation value is reported. It can be noted that false alarms are characterized by correlation values in the range $\{0.00, 0.12\}$, while the correlation values of the detected hot spot are more than two times larger, i.e. 0.27. Therefore, in this extreme case, the search for an optimum threshold appeared to be straightforward. Details of the detected hot spot and an example false alarm are shown in Figure 14.5d, respectively.

14.2 Conclusions

In this chapter, the potential of a novel automatic change detection technique based on PCNNs was investigated. PCNNs are unsupervised, context sensitive, invariant to object scale, shift or rotation. Therefore, PCNNs have rather interesting properties for the automatic change detection of satellite images, especially at very high spatial resolution.

The effectiveness of these properties was addressed for the detection of changes in urban areas using very high resolution images. The two waves of the *time signal* $G[n]$, one for each image, generated by the PCNN during each iteration of the algorithm, create specific signatures of the scene which can be compared to detect changes. The method is completely automated and rather fast as it directly analyzes the correlation between two time signals

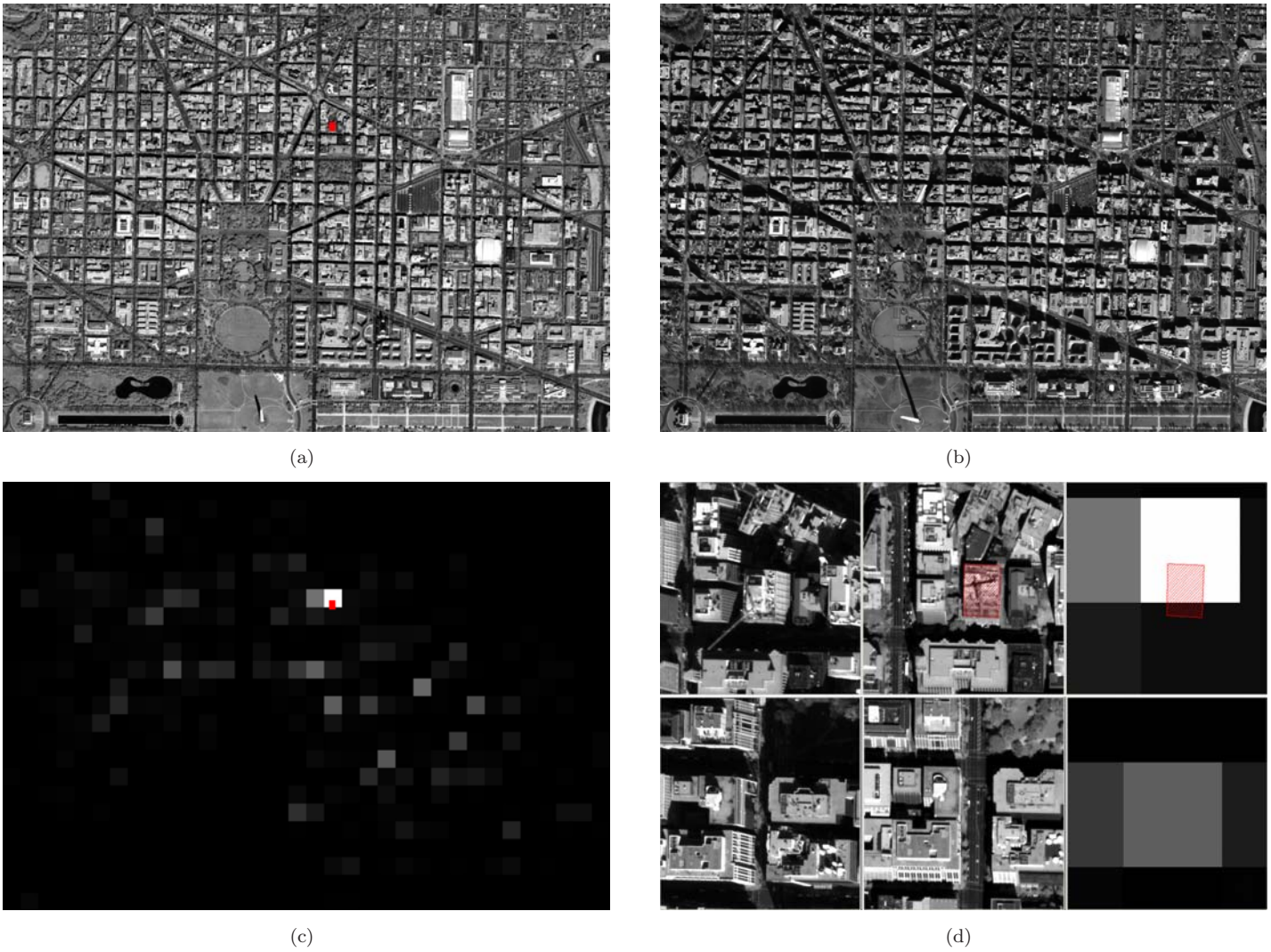


(a) (b) (c) (d)

Figure 14.4: QuickBird image with ground reference in red (a) and WorldView-1 image (b) of Atlanta. In (c), the change map provided by PCNN and (d) details of the detected hot spots, including a false alarm.

associated to the images. Moreover, no pre-processing, except for a raw image registration, is required.

The application of PCNNs to sub-meter resolution images of urban areas produced promising results. PCNNs were first applied to a couple of QuickBird images taken over the Tor Vergata test site (a time shift of less than one year), with an overall object accuracy of 90.7% (no false-alarms). In the other two study cases, the first



(a) (b)

(c) (d)

Figure 14.5: QuickBird image with ground reference in red (a) and WorldView-1 image (b) of Washington D. C. In (c), the change map provided by PCNN and (D) details of the detected hot spots, including a false alarm.

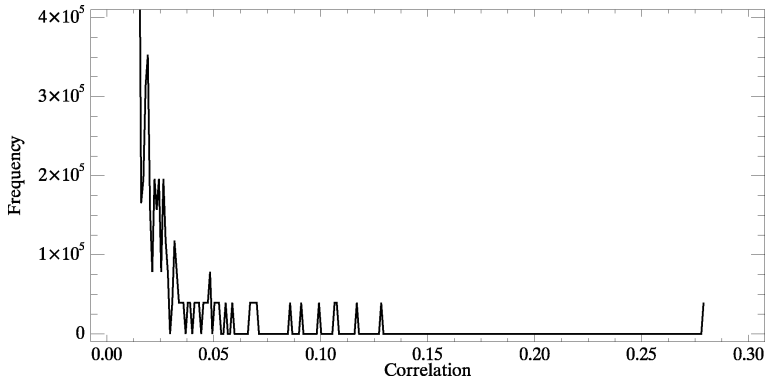


Figure 14.6: Frequency of correlation values for the Washington D. C. case. It can be noted that false alarms are characterized by correlation values in the range $\{0.00, 0.12\}$, while the correlation value of the detected hot spot is more than two times higher, i.e. 0.27.

acquisition was made by QuickBird and the second one by WorlView-1. For the Atalanta area, 30 out of the 34

objects appearing on the ground reference were detected with 6 false alarms, mainly due to presence of leaves on the trees in the WorlView-1 image. The goal of the Washington D. C. scene was to demonstrate the robustness of PCNNs when applied to images acquired with very different view angles. Also in this case, the results were satisfactory since false alarms showed less significant correlation values with respect to real changes.

As a final remark, it is important to observe that the approach is aimed at discovering changed sub-areas in the image (the *hot spots*) rather than analyzing the single pixel. This might be more convenient when large data sets have to be examined, as it should be the case in the very next years when new satellite missions will be providing additional data along with the ones already available.

Chapter 15

Conclusions

Reliable information in populated areas is essential for several applications, from urban planning to strategic decision making in cases of emergency. The aim of this thesis was to investigate the capability of the remote sensing data collected by the new generation of satellite sensors to provide useful information on human settlements.

The progress made with respect to the spatial/temporal/spectral resolutions of the sensor technology needs novel approaches to take advantage of the complex source of information, from the extraction of image features to methodologies to exploit the information extracted. For example, in Chapter 6 it was shown how the use of contextual information derived from textural and morphological analyses dramatically improved the classification accuracy compared to the use of spectral information alone. The multi-temporal component discussed in Chapter 7 provided essential information on urban dynamics as well. In fact, the seasonal variations can be exploited to detect and monitor vegetated areas as well as man made structures. Additionally, hyper-spectral sensors can be effectively used to detect narrow band absorption features to distinguish different material classes. However, the characteristics of this data can pose different processing problems, as discussed in Chapter 8. Finally, as a consequence of the increasing availability of multi-source remote sensing imagery, the integration of data from multiple sensors was investigated in Chapter 9. Data fusion proved to offer better performance over a single-sensor approach by efficiently exploiting the complementary information of the different source types.

The huge amount of data acquired everyday by the different generations of optical and SAR satellite sensors needs to be systematically collected, stored and (eventually) processed. However, the information extraction phase is generally too complex, too expensive and too dependent on user conjecture to be applied systematically over an adequate number of scenes, giving rise to the urgent need of automatic or semi-automatic techniques. In Chapter 10, the capabilities of a single neural network of performing automatic classification and feature extraction over a collection of archived images were explored, while in Chapter 11 two active learning models, MS-cSV and EQB, were proposed and discussed in detail for the semi-automatic definition of a suitable (in terms of information and computational costs) training set.

The last part of this thesis was aimed at developing novel techniques specifically designed for urban change detection purposes. A parallel approach based on a neural architecture was discussed in Chapter 13, while the application of PCNNs for change detection was illustrated in Chapter 14.

In the literature, several studies have been proposed for approaching the diverse urban-related issues using different methods. Here a few of them were reviewed and novel techniques were investigated and developed. However, the goal of an accurate and fast image analysis is still far from being achieved and many aspects still need to be investigated, especially with the next generation of finer resolution satellite sensors. For example, the

availability of WorldView-2 data will pose new exciting challenges to the remote sensing community due to its high spatial resolution 8 bands multi-spectral imagery and the frequent revisit time (about 1.1 days).

Acknowledgments

Many people have contributed during these years in countless ways and for a variety of reasons, and I would like to thank those who made this thesis possible. I have been lucky enough to have the support of many people and though not all played direct roles, each one of them contributed in helping me to get where I am today; things as simple as being a caring friend, hanging out, and having fun, made an enormous difference. Others were responsible for giving me a push in the right direction in life, and for everyone listed here I am eternally grateful for their help.

First of all, I would like to thank Prof. Domenico Solimini for his guidance throughout the past years of my studies; his experience improved my research skills and prepared me for future challenges. My special appreciation goes to Prof. Fabio Del Frate for the numerous fruitful discussions, his generosity with his time, and his advice and references, to name just a few of his contributions. I would also like to thank my friends and colleagues of the GeoInformation PhD program; Emanuele Angiuli, Alessandro Burini, Marco Del Greco, Andrea Della Vecchia, Riccardo Duca, Michele Iapaolo, Marco Lavalle, Michele Lazzarini, Giorgio Licciardi, Andrea Minchella, Alessandra Monerris, Chiara Pratola, Cosimo Putignano, Andrea Radius, Rachid Rahmoune and Pasquale Sellitto. I have learned so much from the GeoInformation program and I have enjoyed the personal interactions with each of these people. A special thanks to Chiara Solimini who frequently collaborated and interacted with me during all these years. In addition I would like to acknowledge Prof. Roberto Basili, Matteo Luciani, Francesco Mesiano, Stefano Robustelli and Riccardo Rossi for their amazing work.

I would like to thank Prof. William J. Emery for his support, encouragement and inspirations on several projects, for giving me the free space to explore different ideas, and for his instructions on both technical and non-technical aspects of my thesis. I would like to express my gratitude to him for the countless hours spent revising, explaining, revising and re-explaining the arguments of our discussions. His enthusiasm has been important for the completion of my research, and I cannot thank him enough for helping me to accomplish this goal. I benefited greatly from the academic community of the University of Colorado. I am especially grateful to Daniel Baldwin, Ian Crocker, Eldad Eshed, Chuck Fowler, Steve Hart, Nathan Longbothan, Dax Matthews, and Waqas Qazi.

I would like to show my gratitude to DigitalGlobe for encouraging and believing in me all these years, and for their generous financial support. I am particularly indebted to my new colleagues; Chuck Chaapel, Milan Karspeck, Keith Krause, Victor Leonard, Giovanni Marchisio, Gregory Miecznik, Kumar Navulur, Chris Padwick, Walter Scott, and Joseph Tankovich, who have made a major effort to smooth my way into my first year of professional research.

I would also like to thank people who have helped me in various stages of my PhD and from whose discussions I have benefited greatly, in particular I would like to thank Andrea Baraldi, Prof. Jocelyn Chanussot, Prof. Mihai Datcu, Roberto Fabrizi, Prof. Paolo Gamba, Fabrizio Pelliccia, Prof. Mauro Pierdicca, and Salvatore Stramondo. Special thanks to Marco Chini and Devis Tuia who have been more than just colleagues to me, they have also become good friends, and I have been lucky enough to have been able to share a great deal of time with them all

over the world. I really enjoyed working with both of them and have profited from it immensely.

I am grateful to all my friends for their support, particularly, Francesca and Sarah Bonanni, Lucia Ciaccia, Elena and Simona Manailescu, Davide Marazzita, Gianfranco Napoletano, Massimiliano Orazi, and Barbara Polsinelli, some of whom have spent countless late nights working with me on numerous projects.

To Guido, Kelsi, and Sandy Honeycutt; for the hospitality you have given me from my very first year in the United States, for making me feel like part of your family, and for countless fun nights full of laughter, thank you.

Most important, none of this would have been possible without the love and patience of my family, who has been a constant source of love, concern, support and strength all these years. I would like to express my heart-felt gratitude to them.

To conclude, I need to acknowledge the most important contributor to my personal well-being of these years in Colorado, my girlfriend Eidelheid Honeycutt, who has been my best friend and most enthusiastic supporter from the very beginning of my studies. Without her, I am certain I would never have made it. She made me a happy person and gave me the encouragement and love necessary to get things done. Finally, I would like to thank our dogs, Kourage and Rambo, who make our days more exciting.

Bibliography

[1] M. Pesaresi and D. Ehrlich, "A methodology to quantify built-up structures from optical vhr imagery", in *Global mapping of human settlement*, P. Gamba and M. Herold, Eds. CRC press, Boca Raton, FL, 2009.

[2] P. Gamba, F. Dell'Acqua, Mattia Stasolla, G. Trianni, and G. Lisini, "Limits and challenges of optical high-resolution satellite remote sensing for urban applications", in *Urban Remote Sensing: Monitoring, Synthesis and Modeling in the Urban Environment*, X. Yang, Ed. John Wiley & Sons, Chichester, UK, in press.

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation", in *Parallel Distributed Processing*, D. E. Rumelhart and J. L. McClelland, Eds. MIT Press, Cambridge, MA, 1987.

[4] R. P. Lippmann, "An introduction to computing with neural nets", *IEEE Acoustic Speech Signal Processing*, vol. 4, no. 2, pp. 4–22, April 1987.

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

[6] H. Bishof, W. Schneider, and A. J. Pinz, "Multispectral classification of landsat-images using neural networks", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30, no. 3, pp. 482–490, May 1992.

[7] M. S. Dawson, "Applications of electromagnetic scattering models to parameter retrieval and classification", in *Microwave Scattering and Emission Models and Their Applications*, A. K. Fung, Ed. Artech House, Norwood, MA, 1994.

[8] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators", *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.

[9] F. Del Frate and L.F. Wang, "Sunflower biomass estimation using a scattering model and a neural network algorithm", *International Journal of Remote Sensing*, vol. 22, pp. 1235–1244, 2001.

[10] L. Tsang, Z. Chen, S. Oh, R. J. Marks, and A. T. C. Chang, "Inversion of snow parameters from passive microwave remote sensing measurements by a neural network trained with a multiple scattering model", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30, no. 5, pp. 1015–1024, September 1992.

[11] P. Cipollini, G. Corsini, M. Diani, and R. Grasso, "Retrieval of sea water optically active parameters from hyperspectral data by means of generalized radial basis function neural networks", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 7, pp. 1508–1524, July 2001.

[12] T. Kohonen, *Self-Organizing Maps*, Springer, Germany, 2001.

[13] L. Bruzzone and F. Melgani, "Robust multiple estimator systems for the analysis of biophysical parameters from remotely sensed data", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 1, pp. 159–174, January 2005.

[14] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme learning machine: Theory and applications", *Neurocomputing*, vol. 70, pp. 489–501, May 2006.

[15] C. Bishop, *Neural Networks for pattern recognition*, Oxford University Press, New York, 1995.

[16] Y.H. Hu and J.N. Hwang, *Handbook of Neural Network Signal Processing*, CRC press, Boca Raton, FL, 2002.

[17] N.K. Kasabov, *Foundations of Neural Networks, Fuzzy Systems and Knowledge Engineering*, The MIT press, Cambridge, Massachusetts, 1996.

[18] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neural Networks and Fuzzy Logic Models*, The MIT press, Cambridge, Massachusetts, 2001.

[19] M. A. Arbib, *The handbook of brain theory and neural networks*, The MIT press, Cambridge, Massachusetts, 2003.

[20] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification", *IEEE Transaction on Neural Networks*, vol. 3, no. 5, pp. 683–697, September 1992.

[21] M. B. De Martino, "A partially recurrent architecture applied to classification problems", in *Proceedings of IEEE International Conference on Neural Networks*, December 1995, vol. 3, pp. 1244–1248, Perth, WA, Australia.

[22] A. Zell *et al.*, *SNNS, Stuttgart Neural Network Simulator*, University of Stuttgart, Stuttgart, Germany, 2008.

[23] M. F. Möller, "A scaled conjugate gradient algorithm for fast supervised learning", *Neural Networks*, vol. 6, no. 4, pp. 525–533, 1993.

[24] Y. C. Cheng, W.M. Qi, and W. Y. Cai, "Dynamic properties of elman and modified elman neural network", in *Proceedings of IEEE Machine Learning and Cybernetics*, November 2002, vol. 2, pp. 637–640.

[25] R. Eckhorn, H. J. Reitboeck, M. Arndt, and P. Dicke, "Feature linking via synchronization among distributed assemblies: simulations of results from cat visual cortex", *Neural Computation*, vol. 2, no. 3, pp. 293–307, 1990.

[26] G. Kuntimad and H. S. Ranganath, "Perfect image segmentation using pulse coupled neural networks", *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 591–598, May 1999.

[27] X. Gu, D. Yu, and L. Zhang, "Image thinning using pulse coupled neural network", *Pattern Recognition Letters*, vol. 25, no. 9, pp. 1075–1084, July 2004.

[28] J. A. Karvonen, "Baltic sea ice sar segmentation and classification using modified pulse-coupled neural networks", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 7, pp. 1566–1574, July 2004.

[29] K. Waldemark, T. Lindblad, V. Bečanović, J. L. L. Guillen, and P. Klingner, "Patterns from the sky satellite image analysis using pulse coupled neural networks for pre-processing, segmentation and edge detection", *Pattern Recognition Letters*, vol. 21, no. 3, pp. 227–237, March 2000.

[30] T. Lindblad and J. M. Kinser, *Image processing using pulse-coupled neural networks*, Springer-Verlag, Berlin Heidelberg, 2005.

[31] A. Verikas and M. Bacauskiene, "Feature selection with neural networks", *Pattern Recognition Letters*, vol. 23, no. 11, pp. 1323–1335, September 2002.

[32] D. A. Landgrebe, *Signal theory methods in multispectral remote sensing*, Wiley, Hoboken, NJ, 2003.

[33] J. A. Benediktsson, M. Pesaresi, and K. Arnason, "Classification and feature extraction for remote sensing images from urban areas based on morphological transformations", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 9, pp. 1940–1949, September 2003.

[34] J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 480–490, March 2005.

[35] A. Plaza, P. Martinez, J. Plaza, and R. Perez, "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 466–479, March 2005.

[36] C. Lee and D. A. Landgrebe, "Feature extraction based on decision boundaries", *IEEE Tranactions on Pattern analysis and Machine Intelligence*, vol. 15, no. 4, pp. 388–400, April 1993.

[37] B. C. Kuo and D. A. Landgrebe, "A robust classification procedure based on mixture classifiers andnonparametric weighted feature extraction", *IEEE Transaction on Geosciences and Remote Sensing*, vol. 40, no. 11, pp. 2486–2494, November 2002.

[38] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature extraction: foundations and applications*, Springer, Berlin, Germany, 2006.

[39] A. P. Carleer and E. Wolff, "Urban land cover multi-level region-based classification of vhr data by selecting relevant features", *International Journal of Remote Sensing*, vol. 27, no. 6, pp. 1035–1051, March 2006.

[40] T. A. Warner, K. Steinmaus, and H. Foote, "An evaluation of spatial autocorrelation feature selection", *International Journal of Remote Sensing*, vol. 20, no. 8, pp. 1601–1616, May 1999.

[41] J. S. Borak, "Feature selection and land cover classification of a modis-like dataset for a semiarid environment", *International Journal of Remote Sensing*, vol. 20, no. 5, pp. 919–938, March 1999.

[42] L. Bruzzone and S. B. Serpico, "A technique for features selection in multiclass problems", *International Journal of Remote Sensing*, vol. 21, no. 3, pp. 549–563, February 2000.

[43] T. Kavzoglu and P. M. Mather, "The role of feature selection in artificial neural network applications", *International Journal of Remote Sensing*, vol. 23, no. 15, pp. 2919–2937, August 2002.

[44] B. Demir and S. Ertürk, "Phase correlation based redundancy removal in feature weighting band selection for hyperspectral images", *International Journal of Remote Sensing*, vol. 29, no. 6, pp. 1801–1807, March 2008.

[45] G. Wilkinson, "Results and implications of a study of fifteen years of satellite image classification experiments", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 4, pp. 433–440, March 2005.

[46] G. M. Foody, "Thematic map comparison: Evaluating the statistical significance of differences in classification accuracy", *Photogrammetric Engineering and Remote Sensing*, vol. 50, no. 5, pp. 627–633, 2004.

[47] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines", *Machine Learning*, vol. 46, pp. 389–422, January 2002.

[48] J. Weston, A. Elisseeff, B. Schölkopf, and M. Tipping, "Use of the zero-norm with linear models and kernel methods", *Journal of Machine Learning Research*, vol. 3, pp. 1439–1461, March 2003.

[49] R. Archibald and G. Fann, "Feature selection and classification of hyperspectral images with support vector machines", *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 4, pp. 674–679, October 2007.

[50] A. Bazi and F. Melgani, "Toward an optimal SVM classification system for hyperspectral remote sensing images", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 11, pp. 3374–3376, November 2006.

[51] F. Del Frate, M. Iapaolo, S. Casadio, S. Godin-Beekmann, and M. Petitdidier, "Neural networks for the dimensionality reduction of gome measurement vector in the estimation of ozone profiles", *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 92, no. 3, pp. 275–291, May 2005.

[52] X. Zeng and D. S. Yeung, "Hidden neuron pruning of multilayer perceptrons using a quantified sensitivity measure", *Neurocomputing*, vol. 69, no. 7-9, pp. 825–837, March 2006.

[53] Y. Hirose, K. Yamashita, and S. Hijiya, "Back-propagation algorithm which varies the number of hidden units", *Neural Networks*, vol. 4, no. 1, pp. 61–66, 1991.

[54] T. Kavzoglu and P. M. Mather, "Pruning artificial neural networks: An example using land cover classification of multi-sensor images", *International Journal of Remote Sensing*, vol. 20, no. 14, pp. 2787–2803, September 1999.

[55] L. M. Belue and K. W. Bauer, "Determining input features for multilayer perceptrons", *Neurocomputing*, vol. 7, no. 2, pp. 111–121, March 1995.

[56] T. Cibas, F. Fogelman Soulié, P. Gallinari, and S. Raudys, "Variable selection with neural networks", *Neurocomputing*, vol. 12, no. 2-3, pp. 223–248, July 1996.

[57] H. Chandrasekaran, H. H. Chen, and M. T. Manry, "Pruning of basis functions in nonlinear approximators", *Neurocomputing*, vol. 34, no. 1, pp. 29–53, September 2000.

[58] G. Castellano, A. Fanelli, and M Pelillo, "An iterative pruning algorithm for feedforward neural networks", *IEEE Transaction on Neural Networks*, vol. 8, no. 3, pp. 519–531, May 1997.

[59] K. Suzuki, I. Horiba, and N. Sugie, "A simple neural network pruning algorithm with application to filter synthesis", *Neural Processing Letters*, vol. 13, no. 1, pp. 43–53, February 2001.

[60] R. Reed, "Pruning algorithms a survey", *IEEE Transactions on Neural Networks*, vol. 4, no. 5, pp. 740–747, September 1993.

[61] G. L. Tarr, *Multi-layered feedforward neural networks for image segmentation*, Ph.D. thesis, Air Force Institute of Technology, Wright-Patterson AFB, OH, 1991.

[62] M. Yacoub and Y. Bennani, *Intelligent engineering systems through artificial neural networks*, ASME, St Louis, MO, 1997.

[63] Global Land Project Secretariat, "Science plan and implementation strategy", *IGBP, Report 53, IHDP Report 19*, 2005.

[64] G. Trianni, *Techniques for fusion of remotely sensed data over urban environments*, Ph.D. thesis, Pavia University, Pavia, Italy, 2007.

[65] R. Bamler and M. Eineder, "The pyramids of gizeh seen by terrasar-x: a prime example for unexpected scattering mechanisms in sar", *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 3, pp. 468–470, July 2008.

[66] ESA, "Annual report", Tech. Rep., European Space Agency, 2006, [online]: http://www.esa.int/, last visited: February, 2010.

[67] DigitalGlobe, "Worldview-1 products quick reference guide", Tech. Rep., DigitalGlobe, Longmont, CO, U.S.A, 2008, [online]: http://digitalglobe.com, last visited: February, 2010.

[68] DigitalGlobe, "Quickbird imagery products: Product guide", Tech. Rep., DigitalGlobe, Longmont, CO, U.S.A, 2008, [online]: http://digitalglobe.com, last visited: February, 2010.

[69] P.K. Varshney, "Multisensor data fusion", *Electronics and Communication Engineering Journal*, vol. 6, no. 9, pp. 245–253, December 1997.

[70] C. Pohl and J. L. Van Genderen, "Multisensor image fusion in remote sensing: concepts, methods and applications", *International Journal of Remote Sensing*, vol. 19, no. 5, pp. 6–23, March 1998.

[71] G. Simone, A. Farina, F. C. Morabito, S. B. Serpico, and L. Bruzzone, "Image fusion techniques for remote sensing applications", *Information Fusion*, vol. 3, no. 1, pp. 315, March 2002.

[72] P. Gong, D. J. Marceau, and P. J. Howarth, "A comparison of spatial feature extraction algorithms for land-use classification with spot hrv data", *Remote Sensing of Environment*, vol. 40, pp. 137–151, 1992.

[73] J. Serra, *Image analysis and Mathematical Morphology*, Academic Press, Singapore, 1982.

[74] P. Soille, *Morphological image analysis*, Springer-Verlag, Berlin-Heidelberg, 2004.

[75] M. Tuceryan and A.K. Jain, *Handbook of Pattern Recognition and Computer Vision*, World Scientific, Singapore, 1993.

[76] R. M. Haralick, K. Shanmugan, and I. Dinstein, "Textural features for image classification", *IEEE Transactions on System, Man, Cybernetics*, vol. SMC3, pp. 610–621, November 1973.

[77] K. S. Shanmugan, V. Narayanan, V. S. Frost, J. A. Stiles, and J. C. Holtzman, "Textural features for dadar image analysis", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 19, pp. 153–156, 1981.

[78] P.M. Treitz, P.J. Howarth, P.J. Filho, and E.D. Soulis, "Agricultural crop classification using sar tone and texture statistics", *Canadian Journal of Remote Sensing*, vol. 26, pp. 18–29, 2000.

[79] D. A. Clausi and B. Yue, "Comparing cooccurrence probabilities and markov random fields for texture analysis of sar sea ice imagery", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 1, pp. 215–228, January 2004.

[80] R. Cossu, "Segmentation by means of textural analysis", *Pixel*, vol. 1, no. 2, pp. 21–24, 1988.

[81] A. Baraldi and F. Parmiggiani, "An investigation of the textural characteristics associated with gray level cooccurrence matrix statistical parameters", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 2, pp. 293–304, March 1999.

[82] V. Karathanassi, C. Iossifidis, and D. Rokos, "A texture-based classification method for classifying built areas according to their density", *International Journal of Remote Sensing*, vol. 21, no. 9, pp. 1807–1823, June 2000.

[83] Q. Zhang and I. Couloigner, "Benefit of the angular texture signature for the separation of parking lots and roads on high resolution multi-spectral imagery", *Pattern Recognition Letters*, vol. 27, no. 9, pp. 937–946, July 2006.

[84] A. Puissant, J. Hirsch, and C. Weber, "The utility of texture analysis to improve per-pixel classification for high to very high spatial resolution imagery", *International Journal of Remote Sensing*, vol. 26, no. 4, pp. 733–745, February 2005.

[85] D. Chen, D. A. Stow, and P. Gong, "Examining the effect of spatial resolution and texture window size on classification accuracy: an urban environment case", *International Journal of Remote Sensing*, vol. 25, no. 11, pp. 2177–2192, June 2004.

[86] T. Kurosu, S. Uratsuka, H. Maeno, and T. Kozu, "Texture statistics for classification of land use with multitemporal jers-1 sar single-look imagery", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 1, pp. 227–235, January 1999.

[87] L. Kurvonen and M. Hallikainen, "Textural information of multitemporal ers-1 and jers-1 sar images with application to land and forest type classification in boreal zone", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 1, pp. 680–689, January 1999.

[88] S. Arzandeh and J. Wang, "Texture evaluation of radarsat imagery for wetland mapping", *Canadian Journal of Remote Sensing*, vol. 28, pp. 653–666, 2002.

[89] F. Dell'Acqua and P. Gamba, "Discriminating urban environments using multi-scale texture and multiple sar images", *International Journal of Remote Sensing*, vol. 27, no. 18, pp. 3797–3812, September 2006.

[90] M. Pesaresi, "Texture analysis for urban pattern recognition using fine-resolution panchromatic satellite imagery", *Geographical and Environmental Modelling*, vol. 4, no. 1, pp. 43–63, May 2000.

[91] J. A. Richards and X. Jia, *Remote sensing digital image analysis*, Springer, Berlin, Germany, 2006.

[92] C. Small, "High spatial resolution spectral mixture analysis of urban reflectance", *Remote Sensing of Environment*, vol. 88, no. 1, pp. 170–186, November 2003.

[93] L. Bruzzone and L. Carlin, "A multilevel context-based system for classification of very high spatial resolution images", *IEEE Transaction on Geosciences and Remote Sensing*, vol. 44, no. 9, pp. 2587–2600, September 2006.

[94] A. Lorette, X. Descombes, and J. Zerubia, "Texture analysis through a markovian modelling and fuzzy classification: Application to urban area extraction from satellite images", *International Journal Computer Vision*, vol. 36, no. 3, pp. 221–236, February 2000.

[95] G. Rellier, X. Descombes, F. Falzon J., and Zerubia, "Texture feature analysis using a gauss-markov model in hyperspectral image classification", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 7, pp. 1543–1551, July 2004.

[96] Y. Zhao, L. Zhang, P. Li, and B. Huang, "Classification of high spatial resolution imagery using improved gaussian markov random-field-based texture features", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 5, pp. 1458–1468, May 2007.

[97] D. A. Clausi and H. Deng, "Design-based texture feature fusion using gabor filters and co-occurrence probabilities", *IEEE Transactions on Image Processing*, vol. 14, no. 7, pp. 925–936, July 2005.

[98] U. Kandaswamy, D. A. Adjeroh, and M. C. Lee, "Efficient texture analysis of sar imagery", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 9, pp. 2075–2083, September 2005.

[99] S. R. Sternberg, "Grayscale morphology", *Computer Vision Graphics and Image Processing*, vol. 35, no. 3, pp. 333–355, 1986.

[100] M. Pesaresi and J. A. Benediktsson, "A new approach for the morphological segmentation of high-resolution satellite images", *IEEE transactions on Geosciences and Remote sensing*, vol. 39, no. 2, pp. 309–320, February 2001.

[101] P. Soille and M. Pesaresi, "Advances in mathematical morphology applied to geoscience and remotesensing", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 9, pp. 2042–2055, September 2002.

[102] I. Epifanio and P. Soille, "Morphological texture features for unsupervised and supervised segmentationsof natural landscapes", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 4, pp. 1074–1083, April 2007.

[103] M. Pesaresi and I. Kannellopoulos, "Detection of urban features using morphological based segmentation and very high resolution remotely sensed data", in *Machine Vision and Advanced Image Processing in Remote Sensing*, I. Kanellopoulos, G. G. Wilkinson, and T. Moons, Eds. Springer Verlag, New York, NY, 1999.

[104] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using svmsand morphological profiles", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 11, pp. 3804 – 3814, November 2008.

[105] A. Plaza, P. Martinez, R. Perez, and J. Plaza, "A new method for target detection in hyperspectral imagery based on extended morphological profiles", in *Proceedings of IEEE Geoscience and Remote Sensing Symposium*, July 2003, vol. 6, pp. 3772– 3774, Toulouse, France.

[106] J. Crespo, J. Serra, and R. Schafer, "Theoretical aspects of morphological filters by reconstruction", *Signal Processing*, vol. 47, no. 2, pp. 201–225, November 1995.

[107] P. A. Devijver and J. Kittler, *Pattern recognition: A Statistical Approach*, Prentice Hall, Englewood Cliffs, NJ, 1982.

[108] P. Mitra, C. A. Murthy, and K. Pal, "Unsupervised feature selection using feature similarity", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 301–312, March 2002.

[109] F. Dell'Acqua and P. Gamba, "Texture-based characterization of urban environments on satellite sar images", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 1, pp. 153–159, January 2003.

[110] S. Dellepiane, D. D. Giusto, S. B. Serpico, and G. Vernazza, "Sar image recognition by integration of intensity and textural information", *International Journal of Remote Sensing*, vol. 12, no. 9, pp. 1915–1932, September 1991.

[111] S. Stramondo, C. Bignami, M. Chini, N. Pierdicca, and A. Tertulliani, "Satellite radar and optical remote sensing for earthquake damage detection: results from different case studies", *International Journal of Remote Sensing*, vol. 27, no. 20, pp. 4433–4447, October 2006.

[112] F. K. Li and R. M. Goldstein, "Studies of multi-baseline spaceborne interferometric synthetic aperture radar", *IEEE Transaction on Geoscience and Remote Sensing*, vol. 28, no. 1, pp. 88–97, January 1990.

[113] C. J. Oliver and S. Quegan, *Understanding synthetic aperture radar images*, Artech House, Norwood, MA, 1998.

[114] L. Bruzzone, M. Marconcini, U. Wegmüller, and A. Wiesmann, "An advanced system for the automatic classification of multitemporal sar images", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 6, pp. 1321–1334, June 2004.

[115] C. Chen and H. Mcnairn, "Identification of burnt areas in mediterranean forest environments from ers-2 sar time series", *International Journal of Remote Sensing*, vol. 25, no. 22, pp. 4873–4888, November 2004.

[116] M. Gimeno, J. San-Miguel-Ayanz, and G. Schmuck, "A neural network integrated approach for rice crop monitoring", *International Journal of Remote Sensing*, vol. 27, no. 7, pp. 1367–1393, April 2004.

[117] M. Bossard, J. Feranec, and J. Otahel, "Corine land cover technical guide", Tech. Rep., European Environment Agency, May 2000, [online]: http://reports.eea.europa.eu/COR0-landcover/en, last visited: February, 2010.

[118] W.M.F. Grey, A.J. Luckman, and D. Holland, "Mapping urban change in the uk using satellite radar interferometry", *Remote Sensing of Environment*, vol. 87, no. 1, pp. 16–22, September 2003.

[119] M. Born and E. Wolf, *Principles of Optics*, Pergamon Press, London, UK, 1959.

[120] E. Weber and H. A. Zebker, "Penetration depths inferred from interferometric volume decorrelation observed over greenland ice sheet", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 6, pp. 2571–2583, November 2000.

[121] U. Wegmuüller and C. L. Werner, "Sar interferometric signatures of forest", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 5, pp. 1153–1161, September 1995.

[122] T. Castel, J. M. Martinez, A. Beaudoin, and T. Strozzi U. Wegmüller, "Ers insar data for remote sensing hilly forested areas", *Remote Sensing of Environment*, vol. 73, pp. 73–86, 2000.

[123] M. E. Engdahl and J. M. Hyyppä, "Land-cover classification using multitemporal ers-1/2 insar data", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 7, pp. 1620–1628, July 2003.

[124] T. Strozzi, P. B. G. Dammert, U. Wegmüller, J. M. Martinez, J. I. H. Askne, A. Beaudoin, and M. T. Hallikainen, "Landuse mapping with ers sar interferometry", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 2, pp. 766–775, March 2000.

[125] M. Santoro, J. I. H. Askne, U. Wegmüller, and C.L. Werner, "Observations, modeling, and applications of ers-envisat coherence over land surfaces", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 8, pp. 2600–2611, August 2007.

[126] A. Al-Janobi, "Performance evaluation of cross-diagonal texture matrix method of texture analysis", *Pattern Recognition*, vol. 34, no. 1, pp. 171–180, January 2001.

[127] C. Sun and W.G.Lee, "Neighboring gray level dependence matrix for texture classification", *Computer Vision, Graphic, Image Processing*, vol. 3, no. 2, pp. 341–352, September 1983.

[128] P. Basili, P. Ciotti, G. D'Auria, N. Pierdicca F. S. Marzano, and P. Quarto, "Assessment of polarimetric features to discriminate land cover from the maestro 1 campaign", *International Journal of Remote Sensing*, vol. 15, no. 14, pp. 2887–2899, September 1994.

[129] F. Del Frate, A. Ortenzi, S. Casadio, and C. Zehner, "Application of neural algorithms for a real-time estimation of ozone profiles from gome measurements", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 10, pp. 2263–2270, October 2002.

[130] A. F. H. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, "Imaging spectrometry for earth remote sensing", *Science*, vol. 228, pp. 1147–1153, 1985.

[131] M. Gianinetto and G. Lechi, "The development of superspectral approaches for the improvement of land cover classification", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 11, pp. 2670–2679, 2004.

[132] J. Plaza, A. Plaza, R. Perez, and P. Martinez, "On the use of small training sets for neural network-based characterization of mixed pixels in remotely sensed hyper-spectral images", *Remote Sensing of Environment*, vol. 42, pp. 3032–3045, 2009.

[133] S. Kumar, J. Ghosh, and M. M. Crawford, "Best-bases feature extraction algorithms for classification of hyperspectral data", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 39, no. 7, pp. 1368–1379, July 2001.

[134] A. C. Jensen and A. S. Solberg, "Fast hyper-spectral feature reduction using piecewise constant function approximations", *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 4, pp. 547–551, October 2007.

[135] S. B. Serpico, M. D'Inca, F. Melgani, and G. Moser, "Comparison of feature reduction techniques for classification of hyperspectral remote sensing data", *Proc. SPIE-Image and Signal Processing for Remote Sensing VIII*, vol. 4885, pp. 347–358, June 2003.

[136] T. K. Ho, J. J. Hull, and S. N. Srihari, "Decision combination in multiple classifier systems", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66–75, January 1994.

[137] L. Lam and S. Y. Suen, "Application of majority voting to pattern recognition: an analysis of its behavior and performance", *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, vol. 27, no. 5, pp. 553–568, September 1997.

[138] D. Hall, *Mathematical Techniques in Multisensor Data Fusion*, Artech House, Boston, MA, 1992.

[139] A. H. Schistad Solberg, A. K. Jain, and T. Taxt, "Multisource classification of remotely sensed data: Fusion of landsat tm and sar images", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 4, pp. 768–778, July 1994.

[140] M. Fauvel, J. Chanussot, and J.A. Benediktsson, "Decision fusion for the classification of urban remote sensing images", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 10, pp. 2828–2838, October 2006.

[141] J. Chanussot, G. Mauris, and P. Lambert, "Fuzzy fusion techniques for linear features detection in multi-temporal sar images", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 3, pp. 1292–1305, May 1999.

[142] A. H. Schistad Solberg, T. Taxt, and A. K. Jain, "A markov random field model for classification of multisource satellite imagery", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 1, pp. 100–113, January 1996.

[143] A. H. Schistad Solberg, "Contextual data fusion applied to forest map revision", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 3, pp. 1234–1243, May 1999.

[144] 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 Transactions on Geoscience and Remote Sensing*, vol. 43, no. 7, pp. 1648–1664, July 2005.

[145] M. Q. Nguyen, P. M. Atkinson, and H. G. Lewis, "Superresolution mapping using a hopfield neural network with fused images", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 3, pp. 736–749, March 2006.

[146] S. Zhao, "Remote sensing data fusion using support vector machine", in *Proceedings of IEEE Geoscience and Remote Sensing Symposium*, 2004, pp. 2575–2578, Seoul, Korea.

[147] J. A. Benediktsson, J. R. Sveinsson, and P. H. Swain, "Hybrid consensus theoretic classification", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 4, pp. 833–843, July 1997.

[148] F. Dell'Acqua, P. Gamba, and G. Lisini, "Improvements to urban area characterization using multitemporal and multiangle sar images", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 9, pp. 1996–2004, September 2003.

[149] R. A. Schowengerdt, *Techniques for image processing and classification in remote sensing*, Academic Press, New York, NY, 1983.

[150] M. Datcu, K. Seidel, and G. Schwarz, "Information mining in remote sensing image archives", in *Machine Vision and Advanced Image Processing in Remote Sensing*, I. Kanellopoulos, G. G. Wilkinson, and T. Moons, Eds. Springer Verlag, New York, NY, 1999.

[151] M. Datcu, H. Daschiel, A. Pelizzari, M. Quartulli, A.Galoppo, A. Colapicchioni, M. Pastori, K. Seidel, P. G. Marchetti, and S. D'Elia, "Information mining in remote sensing image archives: System concepts", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 12, pp. 2923–2936, December 2003.

[152] W. Hsu, M. L. Lee, and J. Zhang, "Image mining: Trends and developments", *Journal of Intelligent Information Systems*, vol. 19, no. 1, pp. 7–23, 2002.

[153] M. Schröder, H. Rehrauer, K. Seidel, and M. Datcu, "Interactive learning and probabilistic retrieval in remote sensing image archives", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 5, pp. 2288–2298, September 2000.

[154] D. J. C. MacKay, "Information based objective functions for active data selection", *Neural Computation*, vol. 4, no. 4, pp. 590–604, July 1992.

[155] D. Cohn, L. Atlas, and R. Ladnery, "Improving generalization with active learning", *Neural Computation*, vol. 15, no. 2, pp. 201–221, May 1994.

[156] D. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models", *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[157] G. Schohn and D. Cohn, "Less is more: active learning with support vectors machines", in *Proceedings of 17th International Conference on Machine Learning*, 2000, pp. 839–846, Stanford, CA.

[158] C. Campbell, N. Cristianini, and A. Smola, "Query learning with large margin classifiers", in *Proceedings of 17th International Conference on Machine Learning*, 2000, pp. 111–118, Stanford, CA.

[159] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering", in *Proceedings of 21th International Conference on Machine Learning*, 2004, p. 79, Banff, Canada.

[160] A. Pozdnoukhov and M. Kanevski, "Monitoring network optimisation for spatial data classification using support vector machines", *International Journal of Environment and Pollution*, vol. 28, no. 3/4, pp. 465–484, 2006.

[161] T. Luo, K. Kramer, D. B. Goldgof, L. O. Hall, S. Samson, A. Remsen, and T. Hopkins, "Active learning to recognize multiple types of plankton", *Journal of Machine Learning Research*, vol. 6, pp. 589–613, December 2005.

[162] S. Cheng and F. Y. Shih, "An improved incremental training algorithm for support vector machines using active query", *Pattern Recognition*, vol. 40, no. 3, pp. 964–971, March 2007.

[163] M. Ferecatu and N. Boujemaa, "Interactive remote-sensing image retrieval using active relevance feedback", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 4, pp. 818–826, April 2007.

[164] S. Tong and D. Koller, "Support vector machines active learning with applications to text classification", *Journal of Machine Learning Research*, vol. 2, no. 1, pp. 45–66, March 2002.

[165] P. Mitra, C. A. Murphy, and S. K. Pal, "A probabilistic active support vector learning algorithm", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 3, pp. 413–418, March 2004.

[166] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers", in *Proceedings of 17th annual international ACM-SIGIR conference on Research and development in information retrieval*, 1994, pp. 3–12, London, UK.

[167] N. Roy and A. McCallum, "Toward optimal active learning through sampling estimation of error reduction", in *Proceedings of International Conference on Machine Learning*, 2001, pp. 441–448, Williamstown, MA.

[168] S. Kullback and R. A. Leibler, "On information and sufficiency", *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, March 1951.

[169] Y. Freund, H. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm", *Machine Learning*, vol. 28, no. 2/3, pp. 133–168, August 1971.

[170] P. Melville, *Creating diverse ensemble classifiers to reduce supervision*, Ph.D. thesis, University of Texas at Austin, 2005.

[171] L. I. Kunchev, *Combining Pattern Classifiers*, Wiley-Interscience, Hoboken, NJ, 2004.

[172] N. Abe and H. Mamitsuka, "Query learning strategies using boosting and bagging", in *Proceedings of International Conference on Machine Learning*, 1998, pp. 1–9, Madison, WI.

[173] Y. Freund and R. Schapire, "A decision-theoretic generalization of the on-line learning and application to boosting", *Journal of Computer and Systems Science*, vol. 55, no. 1, pp. 119–139, August 1999.

[174] L. Breiman, "Bagging predictors", Technical report 421, University of California at Berkley, 1994.

[175] P. Melville and R. J. Mooney, "Diverse ensembles for active learning", in *Proceedings of International Conference on Machine Learning*, 2004, p. 74, Banff, Canada.

[176] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em", *Machine Learning*, vol. 39, no. 3, pp. 103–134, June 2000.

[177] P. Mitra, B. U. Shankar, and S. K. Pal, "Segmentation of multispectral remote sensing images using active support vector machines", *Pattern Recognition Letters*, vol. 25, no. 9, pp. 1067–107, July 2004.

[178] S. Rajan, J. Ghosh, and M. M. Crawford, "An active learning approach to hyperspectral data classification", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 4, pp. 1231–1242, April 2008.

[179] G. Jun and J. Ghosh, "An efficient active learning algorithm with knowledge transfer for hyperspectral remote sensing data", in *Proceedings of IEEE Geoscience and Remote Sensing Symposium*, 2008, pp. 52–55, Boston, MA.

[180] Y. Zhang, X. Liao, and L. Carin, "Detection of buried targets via active selection of labeled data: Application to sensing subsurface uxo", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 11, pp. 2535–2543, November 2004.

[181] Q. Liu, X. Liao, and L. Carin, "Detection of unexploded ordnance via efficient semisupervised and active learning", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 9, pp. 2558–2567, September 2008.

[182] B. Efron, "Bootstrap methods: another look at the jackknife", *Annals of Statistics*, vol. 7, no. 1, pp. 1–26, March 1979.

[183] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 492–501, March 2008.

[184] J. R. Jensen and D. C. Cowen, "Remote sensing of urban/suburban infrastructure and socio-economic attributes", *Photogrammetric Engineering and Remote Sensing*, vol. 65, no. 5, pp. 611–622, May 1999.

[185] Y. Bazi, L. Bruzzone, and F. Melgani, "An unsupervised approach based on the generalized gaussian model to automatic change detection in multitemporal sar images", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 5, pp. 874–887, April 2005.

[186] L. Bruzzone and D. F. Prieto, "An adaptive semiparametric and contextbased approach to unsupervised change detection in multitemporal remote sensing images", *IEEE Transactions on Image Processing*, vol. 11, no. 4, pp. 452–466, April 2002.

[187] T. Kasetkasem and P. K. Varshney, "Image change detection algorithm based on markov random field models", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 8, pp. 1815–1823, August 2002.

[188] F. Bovolo and L. Bruzzone, "A theoretical framework for unsupervised change detection based on change vector analysis in the polar domain", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 1, pp. 218–236, January 2007.

[189] H. Alphan, H. Doygun, and Y. I. Unlukaplan, "Post-classification comparison of land cover using multitemporal landsat and aster imagery: the case of kahramanmaraş, turkey", *Environmental Monitoring and Assessment*, vol. 151, no. 1-4, pp. 327–336, April 2009.

[190] D. Yuan and C. Elvidge, "Nalc land cover change detection pilot study: Washington, d.c. area experiments", *Remote Sensing of Environment*, vol. 66, no. 2, pp. 166–178, 1998.

[191] A. Singh, "Digital change detection techniques using remotely-sensed data", *International Journal of Remote Sensing*, vol. 10, no. 6, pp. 989–1003, 1989.

[192] J. L. Van Genderen, B. F. Lock, and P. A. Vass, "Remote sensing: Statistical testing of thematic map accuracy", *Remote Sensing of Environment*, vol. 7, no. 1, pp. 3–14, 1978.

[193] G. H. Rosenfield, K. Fitzpatrick-Lins, and H. S. Ling, "Sempling for the thematic map accuracy testing", *Photogrammetric Engineering and remote Sensing*, vol. 48, pp. 131–137, 1982.

[194] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate bayesian a posteriori probabilities", *Neural Computing*, vol. 3, no. 4, pp. 461–483, July 1991.

[195] J. D. Paola and R. A. Schowengerdt, "A detailed comparison of backpropagation neural network and maximum likelihood classifiers for urban land use classification", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 33, no. 4, pp. 981–996, July 1995.

[196] J. Friedman, "Regularized discriminant analysis", *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165–175, March 1989.

[197] J. Hoffbeck and D. Landgrebe, "Covariance matrix estimation and classification with limited training data", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 7, pp. 763–767, July 1996.

List of acronyms

AVIRIS	Airborne Visible/Infrared Imaging Spectrometer
BPTT	Back-propagation Through Time
C	Closing
CGM	Conjugate Gradient Methods
CNES	Centre National d'Etudes Spatiales
CORINE	Coordination of Information on the Environment
CR	Closing by Reconstruction
CTH	Closing Top-Hat
CTH	Closing by Top-Hat
DG	Digital Number
DLR	Deutsches Zentrum für Luft und Raumfahrt
DMC	Direct Multi-data Classification
DMP	Differential Morphological Profile
ENVISAT	Environmental Satellite
EQB	Entropy Query-by-Bagging
ERS-1	European Remote Sensing satellite 1
ERS-2	European Remote Sensing satellite 2
ESA	European Space Agency
ESRIN	European Space Research Institute
ETM	Enhanced Thematic Mapper
FS	Feature Selection
GLCM	Gray-Level Co-occurence Matrix
IEEE	Institute of Electrical and Electronics Engineers
IRS	Indian Remote Sensing Satellite
KIM	Knowledge-driven Information Mining
KL	Kullback-Leibler
KSC	Kennedy Space Center
MB	Magnitude-Base
MLP	Multi-layer Perceptron
MMU	Minimum Mapping Unit
MRF	Markov Random Field
MS	Margin Sampling
MS-cSV	Margin Sampling by closest Support Vector
MV	Majority Voting
NAHIRI	Neural Architecture for very HIgh Resolution Imagery
NASA	National Aeronautics and Space Administration
NDVI	Normalized Difference Vegetation Index
NN	Neural Network
NON-SH	NON-Shadowed pixels
O	Opening
OR	Opening by Reconstruction

OTH	Opening Top-Hat
OTH	Opening by Top-Hat
PCA	Principal Component Analysis
PCC	Post Classification Comparison
PCNN	Pulse Coupled Neural Networks
RBF	Radial Basis Function
RFE	Recursive Feature Elimination
ROSIS	Reflective Optics System Imaging Spectrometer
SAR	Synthetic Aperture Radar
SCG	Scaling Gradient
SE	Structuring Element
SH	Shadowed pixels
SLC	Single Look Complex
SPOT	Satellite Pour l'Observation de la Terre
SpRS	Spatial Random Sampling
SRS	Stratified Random Sampling
SSE	Sum-of-Squares Error
SVM	Support Vector Machine
TR	Training Set
VA	Validation Set