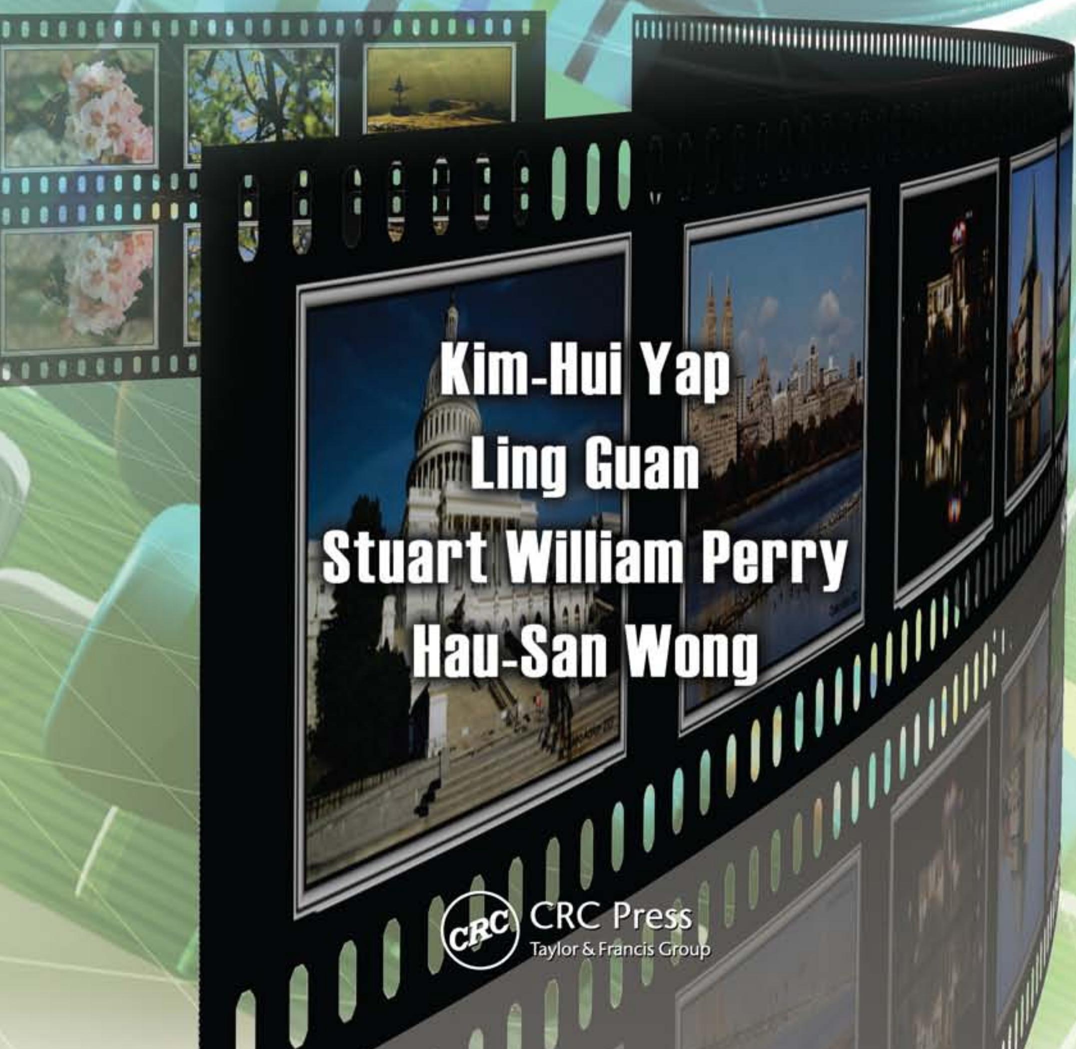
Adaptive Image Processing

A Computational Intelligence Perspective

SECOND EDITION



Kim-Hui Yap

Ling Guan

Stuart William Perry

Hau-San Wong

CRC Press
Taylor & Francis Group

Adaptive Image Processing

A Computational Intelligence Perspective

SECOND EDITION

IMAGE PROCESSING SERIES

Series Editor: Phillip A. Laplante, Pennsylvania State University

Published Titles

Adaptive Image Processing: A Computational Intelligence Perspective, Second Edition
Kim-Hui Yap, Ling Guan, Stuart William Perry, and Hau-San Wong

Color Image Processing: Methods and Applications
Rastislav Lukac and Konstantinos N. Plataniotis

Image Acquisition and Processing with LabVIEW™
Christopher G. Relf

Image and Video Compression for Multimedia Engineering, Second Edition
Yun Q. Shi and Huiyang Sun

Multimedia Image and Video Processing
Ling Guan, S.Y. Kung, and Jan Larsen

Shape Classification and Analysis: Theory and Practice, Second Edition
Luciano da Fontoura Costa and Roberto Marcondes Cesar, Jr.

Single-Sensor Imaging: Methods and Applications for Digital Cameras
Rastislav Lukac

Software Engineering for Image Processing Systems
Phillip A. Laplante

Adaptive Image Processing

A Computational Intelligence Perspective

SECOND EDITION

Kim-Hui Yap
Ling Guan
Stuart William Perry
Hau-San Wong

CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

Version Date: 20150511

International Standard Book Number-13: 978-1-4200-8436-8 (eBook - PDF)

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

Contents

Preface

In this book, we consider the application of computational intelligence techniques to the problem of adaptive image processing. In adaptive image processing, it is usually required to identify each image pixel with a particular feature type (e.g., smooth regions, edges, textures, etc.) for separate processing, which constitutes a segmentation problem. We will then establish image models to describe the desired appearance of the respective feature types or, in other words, to characterize each feature type. Finally, we modify the pixel values in such a way that the appearance of the processed features conforms more closely with that specified by the feature models, where the degree of discrepancy is usually measured in terms of cost function. In other words, we are searching for a set of parameters that minimize this function, that is, an optimization problem.

To satisfy the above requirements, we consider the application of computational intelligence (CI) techniques to this class of problems. Here we will adopt a specific definition of CI, which includes neural network techniques (NN), fuzzy set theory (FS), and evolutionary computation (EC). A distinguishing characteristic of these algorithms is that they are either biologically inspired, as in the cases of NN and EC, or are attempts to mimic how human beings perceive everyday concepts, as in FS.

The choice of these algorithms is due to the direct correspondence between some of the above requirements with the particular capabilities of specific CI approaches. For example, segmentation can be performed by using NN. In addition, for the purpose of optimization, we can embed the image model parameters as adjustable network weights to be optimized through the network's dynamic action. In contrast, the main role of fuzzy set theory is to address the requirement of characterization, that is, the specification of human visual preferences, which are usually expressed in fuzzy languages, in the form of multiple fuzzy sets over the domain of pixel value configurations, and the role of EC is mainly in addressing difficult optimization problems.

In this book, the essential aspects of the adaptive image processing problems are illustrated through a number of applications organized in two parts. The first part of the book focuses on adaptive image restoration. The problem is representative of the general adaptive image processing paradigm in that the three requirements of segmentation, characterization, and optimization are present. The second part of the book centers on image analysis and retrieval. It examines the problems of edge detection and characterization, self-organization for pattern discovery, and content-based image categorization and retrieval. This section will demonstrate how CI techniques can be used to address various challenges in adaptive image processing including low-level image processing, visual content analysis, and feature representation.

This book consists of 11 chapters. The first chapter provides material of an introductory nature to describe the basic concepts and current state of the art in the field of computational intelligence for image restoration, edge detection, image analysis, and retrieval. Chapter 2 gives a mathematical description of the restoration problem from the Hopfield neural network perspective and describes current algorithms based on this method. Chapter 3 extends the algorithm presented in Chapter 2 to implement adaptive constraint restoration methods for both spatially invariant and spatially variant degradations. Chapter 4 utilizes a perceptually motivated image error measure to introduce novel restoration algorithms. Chapter 5 examines how model-based neural networks can be used to solve image-restoration problems. Chapter 6 examines image-restoration algorithms making use of the principles of evolutionary computation. Chapter 7 examines the difficult concept of image restoration when insufficient knowledge of the degrading function is available. Chapter 8 examines the subject of edge detection and characterization using model-based neural networks. Chapter 9 provides an in-depth coverage of the self-organizing tree map, and demonstrates its application in image analysis and retrieval. Chapter 10 examines content representation in compressed domain image classification using evolutionary algorithm. Finally, Chapter 11 explores the fuzzy user perception and small sample problem in content-based image retrieval and develops CI techniques to address these challenges.

Acknowledgments

We are grateful to our colleagues, especially Dr. Kui Wu in the Media Technology Lab of Nanyang Technological University, Singapore for their contributions and helpful comments during the preparation of this book. Our special thanks to Professor Terry Caelli for the many stimulating exchanges that eventually led to the work in Chapter 8. We would also like to thank Nora Konopka and Amber Donley of CRC Press for their advice and assistance. Finally, we are grateful to our families for their patience and support while we worked on the book.

1

Introduction

1.1 Importance of Vision

All life-forms require methods for sensing the environment. Being able to sense one's surroundings is of such vital importance for survival that there has been a constant race for life-forms to develop more sophisticated sensory methods through the process of evolution. As a consequence of this process, advanced life-forms have at their disposal an array of highly accurate senses. Some unusual sensory abilities are present in the natural world, such as the ability to detect magnetic and electric fields, or the use of ultrasound waves to determine the structure of surrounding obstacles. Despite this, one of the most prized and universal senses utilized in the natural world is vision.

Advanced animals living aboveground rely heavily on vision. Birds and lizards maximize their fields of view with eyes on each side of their skulls, while other animals direct their eyes forward to observe the world in three dimensions. Nocturnal animals often have large eyes to maximize light intake, while predators such as eagles have very high resolution eyesight to identify prey while flying. The natural world is full of animals of almost every color imaginable. Some animals blend in with surroundings to escape visual detection, while others are brightly colored to attract mates or warn aggressors. Everywhere in the natural world, animals make use of vision for their daily survival. The reason for the heavy reliance on eyesight in the animal world is due to the rich amount of information provided by the visual sense. To survive in the wild, animals must be able to move rapidly. Hearing and smell provide warning regarding the presence of other animals, yet only a small number of animals such as bats have developed these senses sufficiently to effectively utilize the limited amount of information provided by these senses to perform useful actions, such as to escape from predators or chase down prey. For the majority of animals, only vision provides sufficient information in order for them to infer the correct responses under a variety of circumstances.

Humans rely on vision to a much greater extent than most other animals. Unlike the majority of creatures we see in three dimensions with high resolution and color. In humans the senses of smell and hearing have taken second place to vision. Humans have more facial muscles than any other animal, because in our society facial expression is used by each of us as the primary indicator of the emotional states of other humans, rather than the scent signals used by many mammals. In other words, the human world revolves around visual stimuli and the importance of effective visual information processing is paramount for the human visual system.

To interact effectively with the world, the human vision system must be able to extract, process, and recognize a large variety of visual structures from the captured images. Specifically, before the transformation of a set of visual stimuli into a meaningful scene, the vision system is required to identify different visual structures such as edges and regions from the captured visual stimuli. Rather than adopting a uniform approach of processing these extracted structures, the vision system should be able to adaptively tune to the specificities of these different structures in order to extract the maximum amount of information for the subsequent recognition stage. For example, the system should selectively enhance the associated attributes of different regions such as color and textures in an adaptive manner such that for some regions, more importance is placed on the extraction and processing of the color attribute, while for other regions the emphasis is placed on the associated textural patterns. Similarly, the vision system should also process the edges in an adaptive manner such that those associated with an object of interest should be distinguished from those associated with the less important ones.

To mimic this adaptive aspect of biological vision and to incorporate this capability into machine vision systems have been the main motivations of image processing and computer vision research for many years. Analogous to the eyes, modern machine vision systems are equipped with one or more cameras to capture light signals, which are then usually stored in the form of digital images or video sequences for subsequent processing. In other words, to fully incorporate the adaptive capabilities of biological vision systems into machines necessitates the design of an effective *adaptive image processing* system. The difficulties of this task can already be foreseen since we are attempting to model a system that is the product of billions of years of evolution and is naturally highly complex. To give machines some of the remarkable capabilities that we take for granted is the subject of intensive ongoing research and the theme of this book.

1.2 Adaptive Image Processing

The need for adaptive image processing arises due to the need to incorporate the above adaptive aspects of biological vision into machine vision systems. For such systems the visual stimuli are usually captured through cameras and presented in the form of digital images that are essentially arrays of pixels, each of which is associated with a gray level value indicating the magnitude of the light signal captured at the corresponding position. To effectively characterize a large variety of image types in image processing, this array of numbers is usually modeled as a 2D discrete nonstationary random process. As opposed to stationary random processes where the statistical properties of the signal remain unchanged with respect to the 2D spatial index, the nonstationary process models the inhomogeneities of visual structures that are inherent in a meaningful visual scene. It is this inhomogeneity that conveys useful information of a scene, usually composed of a number of different objects, to the viewer. On the other hand, a stationary 2D random signal, when viewed as a gray level image, does not usually correspond to the appearances of real-world objects.

For a particular image-processing application (we interpret the term "image processing" in a wide sense such that applications in image analysis are also included), we usually assume the existence of an underlying *image model* [1–3], which is a mathematical description of a hypothetical process through which the current image is generated. If we suppose that an image is adequately described by a stationary random process, which, though not accurate in general, is often invoked as a simplifying assumption, it is apparent that only a single image model corresponding to this random process is required for further image processing. On the other hand, more sophisticated image-processing algorithms will account for the nonstationarity of real images by adopting *multiple* image models for more accurate representation. Individual regions in the image can usually be associated with a different image model, and the complete image can be fully characterized by a finite number of these local image models.

1.3 Three Main Image Feature Classes

The inhomogeneity in images implies the existence of more than one image feature type that convey independent forms of information to the viewer. Although variations among different images can be great, a large number of images can be characterized by a small number of feature types. These are usually summarized under the labels of smooth regions, textures, and edges (Figure 1.1). In the following, we will describe the essential characteristics of these three kinds of features, and the image models usually employed for their characterization.

1.3.1 Smooth Regions

Smooth regions usually comprise the largest proportion of areas in images, because surfaces of artificial or natural objects, when imaged from a distance, can usually be regarded as smooth. A simple model for a smooth region is the assignment of a constant gray-level value to a restricted domain of the image lattice, together with the addition of Gaussian noise of appropriate variance to model the sensor noise [2,4].

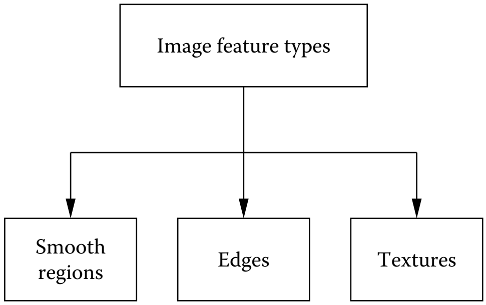


FIGURE 1.1
Three important classes of feature in images.

1.3.2 Edges

As opposed to smooth regions, edges comprise only a very small proportion of areas in images. Nevertheless, most of the information in an image is conveyed through these edges. This is easily seen when we look at the edge map of an image after edge detection: we can readily infer the original contents of the image through the edges alone. Since edges represent locations of abrupt transitions of gray-level values between adjacent regions, the simplest edge model is therefore a random variable of high variance, as opposed to the smooth region model that uses random variables with low variances. However, this simple model does not take into account the structural constraints in edges, which may then lead to their confusion with textured regions with equally high variances. More sophisticated edge models include the facet model [5], which approximates the different regions of constant gray level values around edges with separate piecewise continuous functions. There is also the edge-profile model, which describes the one-dimensional cross section of an edge in the direction of maximum gray level variation [6,7]. Attempts have been made to model this profile using a step function and various monotonically increasing functions. Whereas these models mainly characterize the *magnitude* of gray-level-value transition at the edge location, the edge diagram in terms of zero crossings of the second-order gray level derivatives, obtained through the process of Laplacian of Gaussian (LoG) filtering [8,9], characterizes the edge *positions* in an image. These three edge models are illustrated in Figure 1.2.

1.3.3 Textures

The appearance of textures is usually due to the presence of natural objects in an image. The textures usually have a noise-like appearance, although they are distinctly different from noise in that there usually exists certain discernible patterns within them. This is due to the correlations among the pixel values in specific directions. Due to this noise-like appearance, it is natural to model textures using a two-dimensional random field. The simplest approach

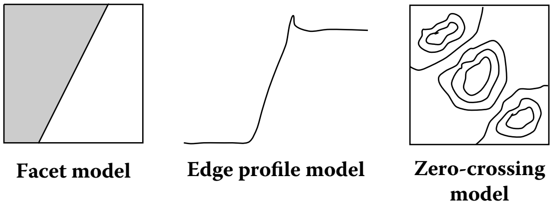


FIGURE 1.2
Examples of edge models.

is to use i.i.d. (independent and identically distributed) random variables with appropriate variances, but this does not take into account the correlations among the pixels. A generalization of this approach is the adoption of Gauss–Markov random field (GMRF) [10–14] and Gibbs random field [15,16] which model these local correlational properties. Another characteristic of textures is their self-similarities: the patterns usually look similar when observed under different magnifications. This leads to their representation as fractal processes [17,18] that possess this very self-similar property.

1.4 Difficulties in Adaptive Image-Processing System Design

Given the very different properties of these three feature types, it is usually necessary to incorporate spatial adaptivity into image-processing systems for optimal results. For an image-processing system, a set of system parameters is usually defined to control the quality of the processed image. Assuming the adoption of spatial domain-processing algorithms, the gray-level value x_{i_1,i_2} at spatial index (i_1, i_2) is determined according to the following relationship.

$$x_{i_1,i_2} = f(\mathbf{y}; \mathbf{p}_{SA}(i_1, i_2)) \tag{1.1}$$

In this equation, the mapping f summarizes the operations performed by the image-processing system. The vector $\mathbf{y}$ denotes the gray-level values of the original image before processing, and $\mathbf{p}_{SA}$ denotes a vector of *spatially adaptive* parameters as a function of the spatial index (i_1, i_2). It is reasonable to expect that different parameter vectors are to be adopted at different positions (i_1, i_2), which usually correspond to different feature types. As a result, an important consideration in the design of this adaptive image-processing system is the proper determination of the parameter vector $\mathbf{p}_{SA}(i_1, i_2)$ as a function of the spatial index (i_1, i_2).

On the other hand, for nonadaptive image-processing systems, we can simply adopt a constant assignment for $\mathbf{p}_{SA}(i_1, i_2)$

$$\mathbf{p}_{SA}(i_1, i_2) \equiv \mathbf{p}_{NA} \tag{1.2}$$

where $\mathbf{p}_{NA}$ is a constant parameter vector.

We consider examples of $\mathbf{p}_{SA}(i_1, i_2)$ in a number of specific image-processing applications below:

- In image filtering, we can define $\mathbf{p}_{SA}(i_1, i_2)$ to be the set of filter coefficients in the convolution mask [2]. Adaptive filtering [19,20] thus corresponds to using a different mask at different spatial locations, while nonadaptive filtering adopts the same mask for the whole image.
- In image restoration [21–23], a *regularization parameter* [24–26] is defined that controls the degree of ill-conditioning of the restoration process, or equivalently, the overall smoothness of the restored image. The vector $\mathbf{p}_{SA}(i_1, i_2)$ in this case corresponds to the scalar regularization parameter. Adaptive regularization [27–29] involves selecting different parameters at different locations, and nonadaptive regularization adopts a single parameter for the whole image.
- In edge detection, the usual practice is to select a single *threshold* parameter on the gradient magnitude to distinguish between the edge and nonedge points of the image [2,4], which corresponds to the case of nonadaptive thresholding. This can be considered as a special case of adaptive thresholding, where a threshold value is defined at each spatial location.

Given the above description of adaptive image processing, we can see that the corresponding problem of adaptive parameterization, that of determining the parameter vector $\mathbf{p}_{SA}(i_1, i_2)$ as a function of (i_1, i_2), is particularly acute compared with the nonadaptive case. In the nonadaptive case, and in particular for the case of a parameter vector of low dimensionality, it is usually possible to determine the optimal parameters by interactively choosing different parameter vectors and evaluating the final processed results.

On the other hand, for adaptive image processing, it is almost always the case that a parameter vector of high dimensionality, which consists of the concatenation of all the local parameter vectors, will be involved. If we relax the previous requirement to allow the subdivision of an image into regions and the assignment of the same local parameter vector to each region, the dimension of the resulting concatenated parameter vector can still be large. In addition, the requirement to identify each image pixel with a particular feature type itself constitutes a nontrivial *segmentation* problem. As a result, it is usually not possible to estimate the parameter vector by trial and error. Instead, we should look for a parameter assignment algorithm that would automate the whole process.

To achieve this purpose, we will first have to establish image models that describe the desired local gray-level value configurations for the respective image feature types or, in other words, to *characterize* each feature type. Since the local gray-level configurations of the processed image are in general a function of the system parameters as specified in Equation (1.1), we can associate a *cost function* with each gray-level configuration that measures its degree of conformance to the corresponding model, with the local system parameters as arguments of the cost function. We can then search for those system parameter values that minimize the cost function for each feature type, that is, an *optimization* process. Naturally, we should adopt *different* image models in order to obtain different system parameters for each type of feature.

In view of these requirements, we can summarize the requirements for a successful design of an adaptive image-processing system as follows:

1.4.1 Segmentation

Segmentation requires a proper understanding of the difference between the corresponding structural and statistical properties of the various feature types, including those of edges, textures, and smooth regions, to allow partition of an image into these basic feature types.

1.4.2 Characterization

Characterization requires an understanding of the most desirable gray-level value configurations in terms of the characteristics of the human vision system (HVS) for each of the basic feature types, and the subsequent formulation of these criteria into cost functions in terms of the image model parameters, such that the minimization of these cost functions will result in an approximation to the desired gray-level configurations for each feature type.

1.4.3 Optimization

In anticipation of the fact that the above criteria will not necessarily lead to well-behaved cost functions, and that some of the functions will be nonlinear or even nondifferentiable, we should adopt powerful optimization techniques for the searching of the optimal parameter vector.

These three main requirements are summarized in Figure 1.3.

In this book, our main emphasis is on two specific adaptive image-processing systems and their associated algorithms: the adaptive image-restoration algorithm and the adaptive edge-characterization

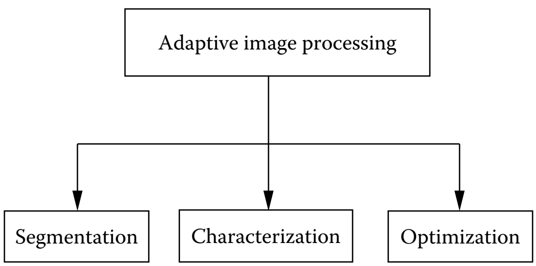


FIGURE 1.3
Three main requirements in adaptive image processing.

algorithm. For the former system, segmentation is first applied to partition the image into separate regions according to a local variance measure. Each region then undergoes characterization to establish whether it corresponds to a smooth, edge, or textured area. Optimization is then applied as a final step to determine the optimal regularization parameters for each of these regions. For the second system, a preliminary segmentation stage is applied to separate the edge pixels from nonedge pixels. These edge pixels then undergo the characterization process whereby the more salient ones among them (according to the users' preference) are identified. Optimization is finally applied to search for the optimal parameter values for a parametric model of this salient edge set.

1.5 Computational Intelligence Techniques

Considering the above stringent requirements for the satisfactory performance of an adaptive image-processing system, it will be natural to consider the class of algorithms commonly known as computational intelligence techniques. The term "computational intelligence" [30,31] has sometimes been used to refer to the general attempt to simulate human intelligence on computers, the so-called "artificial intelligence" (AI) approach [32]. However, in this book, we will adopt a more specific definition of computational intelligence techniques that are neural network techniques, fuzzy logic, and evolutionary computation (Figure 1.4). These are also referred to as the "numerical" AI approaches (or sometimes "soft computing" approach [33]) in contrast to the "symbolic" AI approaches as typified by the expression of human knowledge in terms of linguistic variables in expert systems [32].

A distinguishing characteristic of this class of algorithms is that they are usually biologically inspired: the design of neural networks [34,35], as the

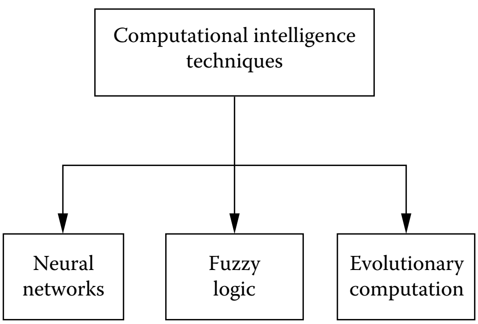


FIGURE 1.4
Three main classes of computational intelligence algorithms.

name implies, draws inspiration mainly from the structure of the human brain. Instead of adopting the serial processing architecture of the Von Neumann computer, a neural network consists of a large number of computational units or neurons (the use of this term again confirming the biological source of inspiration) that are massively interconnected with each other just as the real neurons in the human brain are interconnected with axons and dendrites. Each such connection between the artificial neurons is characterized by an adjustable *weight* that can be modified through a training process such that the overall behavior of the network is changed according to the nature of specific training examples provided, again reminding one of the human learning process.

On the other hand, fuzzy logic [36–38] is usually regarded as a formal way to describe how human beings perceive everyday concepts: whereas there is no exact height or speed corresponding to concepts like "tall" and "fast," respectively, there is usually a general consensus by humans as to approximately what levels of height and speed the terms are referring to. To mimic this aspect of human cognition on a machine, fuzzy logic avoids the arbitrary assignment of a particular numerical value to a single class. Instead, it defines each such class as a *fuzzy set* as opposed to a *crisp set*, and assigns a fuzzy set membership value within the interval [0,1] for each class that expresses the degree of membership of the particular numerical value in the class, thus generalizing the previous concept of crisp set membership values within the discrete set {0,1}.

For the third member of the class of computational intelligence algorithms, no concept is closer to biology than the concept of evolution, which is the incremental adaptation process by which living organisms increase their fitness to survive in a hostile environment through the processes of mutation and competition. Central to the process of evolution is the concept of a *population* in which the better adapted individuals gradually displace the not so well-adapted ones. Described within the context of an optimization algorithm, an *evolutionary computational algorithm* [39,40] mimics this aspect of evolution by generating a population of potential solutions to the optimization problem, instead of a sequence of single potential solutions, as in the case of gradient descent optimization or simulated annealing [16]. The potential solutions are allowed to compete against each other by comparing their respective cost function values associated with the optimization problem with each other. Solutions with high cost function values are displaced from the population while those with low cost values survive into the next generation. The displaced individuals in the population are replaced by generating new individuals from the survived solutions through the processes of mutation and recombination. In this way, many regions in the search space can be explored simultaneously, and the search process is not affected by local minima as no gradient evaluation is required for this algorithm.

We will now have a look at how the specific capabilities of these computational intelligence techniques can address the various problems encountered in the design and parameterization of an adaptive image-processing system.

1.5.1 Neural Networks

Artificial neural networks represent one of the first attempts to incorporate learning capabilities into computing machines. Corresponding to the biological neurons in human brain, we define artificial neurons that perform simple mathematical operations. These artificial neurons are connected with each other through *network weights* that specify the strength of the connection. Analogous to its biological counterpart, these network weights are adjustable through a learning process that enables the network to perform a variety of computational tasks. The neurons are usually arranged in *layers*, with the input layer accepting signals from the external environment, and the output layer emitting the result of the computations. Between these two layers are usually a number of *hidden* layers that perform the intermediate steps of computations. The architecture of a typical artificial neural network with one hidden layer is shown in Figure 1.5. In specific types of network, the hidden layers may be missing and only the input and output layers are present.

The adaptive capability of neural networks through the adjustment of the network weights will prove useful in addressing the requirements of segmentation, characterization, and optimization in adaptive image-processing system design. For segmentation, we can, for example, ask human users to specify which part of an image corresponds to edges, textures, and smooth regions, etc. We can then extract image features from the specified regions as training examples for a properly designed neural network such that the trained network will be capable of segmenting a previously unseen image into the primitive feature types. Previous works where a neural network is applied to the problem of image segmentation are detailed in References [41–43].

A neural network is also capable of performing characterization to a certain extent, especially in the process of *unsupervised competitive learning* [34,44], where both segmentation and characterization of training data are carried

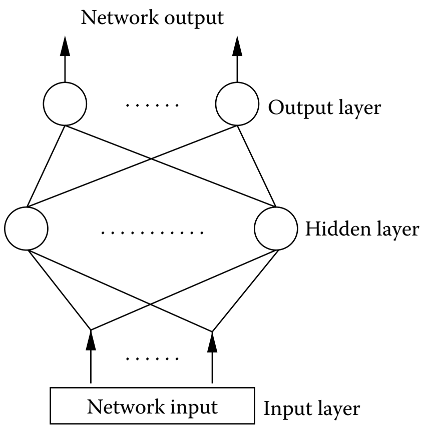


FIGURE 1.5
Architecture of a neural network with one hidden layer.

out: during the competitive learning process, individual neurons in the network, which represent distinct subclasses of training data, gradually build up *templates* of their associated subclasses in the form of *weight vectors*. These templates serve to characterize the individual subclasses.

In anticipation of the possible presence of nonlinearity in the cost functions for parameter estimation during the optimization process, a neural network is again an ideal candidate for accommodating such difficulties: the operation of a neural network is inherently nonlinear due to the presence of the sigmoid neuronal transfer function. We can also tailor the nonlinear neuronal transfer function specifically to a particular application. More generally, we can *map* a cost function onto a neural network by adopting an architecture such that the image model parameters will appear as adjustable weights in the network [45,46]. We can then search for the optimal image model parameters by minimizing the embedded cost function through the dynamic action of the neural network.

In addition, while the distributed nature of information storage in neural networks and the resulting fault-tolerance is usually regarded as an overriding factor in its adoption, we will, in this book, concentrate rather on the possibility of task localization in a neural network: we will subdivide the neurons into *neuron clusters*, with each cluster specialized for the performance of a certain task [47,48]. It is well known that similar localization of processing occurs in the human brain, as in the classification of the cerebral cortex into visual area, auditory area, speech area, and motor area, etc. [49,50]. In the context of adaptive image processing, we can, for example, subdivide the set of neurons in such a way that each cluster will process the three primitive feature types, namely, textures, edges, and smooth regions. The values of the connection weights in each subnetwork can be different, and we can even adopt different architectures and learning strategies for each subnetwork for optimal processing of its assigned feature type.

1.5.2 Fuzzy Logic

From the previous description of fuzzy techniques, it is obvious that its main application in adaptive image processing will be to address the requirement of characterization, that is, the specification of human visual preferences in terms of gray-level value configurations. Many concepts associated with image processing are inherently fuzzy, such as the description of a region as "dark" or "bright," and the incorporation of fuzzy set theory is usually required for satisfactory processing results [51–55]. The very use of the words "textures," "edges," and "smooth regions" to characterize the basic image feature types implies fuzziness: the difference between smooth regions and weak textures can be subtle, and the boundary between textures and edges is sometimes blurred if the textural patterns are strongly correlated in a certain direction so that we can regard the pattern as multiple edges. Since the image-processing system only recognizes gray-level configurations, it will be natural to define fuzzy sets with qualifying terms like "texture," "edge," and "smooth regions" over the set of corresponding gray-level configurations according to human preferences. However, one of the problems with this approach is that there is usually an extremely large number of possible gray-level configurations corresponding to each feature type, and human beings cannot usually relate what they perceive as a certain feature type to a particular configuration. In Chapter 5, a *scalar* measure has been established that characterizes the degree of resemblance of a gray-level configuration to either textures or edges. In addition, we can establish the exact interval of values of this measure where the configuration will more resemble textures than edges and vice versa. As a result, we can readily define fuzzy sets over this one-dimensional *universe of discourse* [37].

In addition, fuzzy set theory also plays an important role in the derivation of improved segmentation algorithms. A notable example is the *fuzzy c-means algorithm* [56–59], which is a generalization of the *k-means algorithm* [60] for data clustering. In the k-means algorithm, each data vector, which may contain feature values or gray-level values as individual components in image processing applications, is assumed to belong to one and only one class. This may result in inadequate characterization of certain data vectors that possess properties common to more than one class, but then get arbitrarily assigned to one of those classes. This is prevented in the fuzzy c-means algorithm, where each data vector is assumed to belong to every class to a different degree that is expressed by a numerical membership value in the interval [0,1]. This paradigm can now accommodate those data vectors that possess attributes common to more than one class, in the form of large membership values in several of these classes.

1.5.3 Evolutionary Computation

The often stated advantages of evolutionary computation include its implicit parallelism that allows simultaneous exploration of different regions of the search space [61], and its ability to avoid local minima [39,40]. However, in this book, we will emphasize its capability to search for the optimizer of a *nondifferentiable* cost function efficiently, that is, to satisfy the requirement of optimization. An example of a nondifferentiable cost function in image processing would be the metric that compares the probability density function (pdf) of a certain local attribute of the image (gray-level values, gradient magnitudes, etc.) with a desired pdf. We would, in general, like to adjust the parameters of the adaptive image-processing system in such a way that the distance between the pdf of the processed image is as close as possible to the desired pdf. In other words, we would like to minimize the distance as a function of the system parameters. In practice, we have to approximate the pdfs using histograms of the corresponding attributes, which involves the counting of discrete quantities. As a result, although the pdf of the processed image is a function of the system parameters, it is not differentiable with respect to these parameters. Although stochastic algorithms like simulated annealing can also be applied to minimize nondifferentiable cost functions,

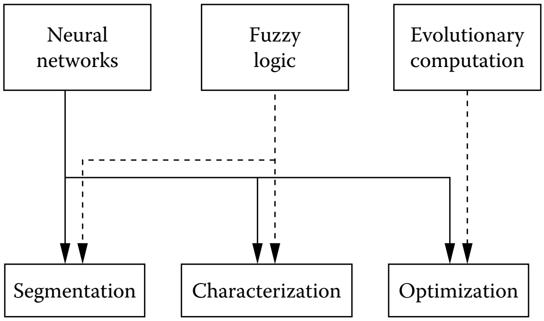


FIGURE 1.6
Relationships between the computational intelligence algorithms and the main requirements in adaptive image processing.

evolutionary computational algorithms represent a more efficient optimization approach due to the implicit parallelism of its population-based search strategy.

The relationship between the main classes of algorithms in computational intelligence and the major requirements in adaptive image processing is summarized in Figure 1.6.

1.6 Scope of the Book

In this book, as specific examples of adaptive image-processing systems, we consider the *adaptive regularization* problem in image restoration [27–29], the edge, characterization problem, the self-organization problem in image analysis, and the feature representation and fuzzy perception problem in image retrieval. We adopt computational intelligence techniques including neural networks, fuzzy methods, and evolutionary algorithms as the main approaches to address these problems due to their capabilities to satisfy all three requirements in adaptive image processing, as illustrated in Figure 1.6.

1.6.1 Image Restoration

The act of attempting to obtain the original image given the degraded image and some knowledge of the degrading factors is known as *image restoration*. The problem of restoring an original image, when given the degraded image, with or without knowledge of the degrading *point spread function* (PSF) or degree and type of noise present is an ill-posed problem [21,24,62,63] and can be approached in a number of ways such as those given in References [21,64,66]. For all useful cases a set of simultaneous equations is produced that is too large to be solved analytically. Common approaches to this problem can be divided into two categories: inverse filtering or transform-related techniques, and algebraic techniques. An excellent review of classical image-restoration techniques is given by Andrews and Hunt [21]. The following references also contain surveys of restoration techniques: Katsaggelos [23], Sondhi [67], Andrews [68], Hunt [69], and Frieden [70].

Image Degradations

There exist a large number of possible degradations that an image can suffer. Common degradations are blurring, motion, and noise. Blurring can be caused when an object in the image is outside the camera's depth of field some time during the exposure. Noise is generally a distortion due to the imaging system rather than the scene recorded. Noise results in random variations to pixel values in the image. This could be caused by the imaging system itself, or the recording or transmission medium. In this book, we consider only image degradations that may be described by a linear model. For these distortions, a suitable mathematical model is given in Chapter 2.

Adaptive Regularization

In regularized image restoration, the associated cost function consists of two terms: a data conformance term that is a function of the degraded image pixel values and the degradation mechanism, and the model conformance term that is usually specified as a continuity constraint on neighboring gray-level values to alleviate the problem of ill-conditioning characteristic of this kind of inverse problems. The *regularization parameter* [23,25] controls the relative contributions of the two terms toward the overall cost function. In general, if the regularization parameter is increased, the model conformance term is emphasized at the expense of the data conformance term, and the restored image becomes smoother while the edges and textured regions become blurred. On the contrary, if we decrease the parameter, the fidelity of the restored image is increased at the expense of decreased noise smoothing.

Perception-Based Error Measure for Image Restoration

The most common method to compare the similarity of two images is to compute their mean square error (MSE). However, the MSE relates to the power of the error signal and has little relationship to human visual perception. An important drawback to the MSE and any cost function that attempts to use the MSE to restore a degraded image is that the MSE treats the image as a stationary process. All pixels are given equal priority regardless of their relevance to human perception. This suggests that information is ignored. In view of the problems with classical error measures such as the MSE, Perry and Guan [71] and Perry [72] presented a different error measure, *local standard deviation mean square error* (LSMSE), which is based on the comparison of local standard deviations in the neighborhood of each pixel instead of their gray-level values. The LSMSE is calculated in the following way: Each pixel in the two images to be compared has its local standard deviation calculated over a small neighborhood centered on the pixel. The error between each pixel's local standard deviation in the first image and the corresponding pixel's local standard deviation in the second image is computed. The LSMSE is the mean squared error of these differences over all pixels in the image. The mean square error between the two standard deviations gives an indication of the degree of similarity between the two images. This error measure requires matching between the high- and low-variance regions of the image, which is more intuitive in terms of human visual perception. Generally throughout this book the size of the local neighborhoods used in the LSMSE calculation will be a 9-by-9 square centered on the pixel of interest. This alternative error measure will be heavily relied upon in Chapters 3 and 4. A mathematical description is given in Chapter 3.

Blind Deconvolution

In comparison with the determination of the regularization parameter for image restoration, the problem of *blind deconvolution* is considerably more difficult, since in this case the degradation mechanism, or equivalently the form of the point spread function, is unknown to the user. As a result, in addition to estimating the local regularization parameters, we have to estimate the coefficients of the point spread function itself. In Chapter 7, we describe an approach for blind deconvolution that is based on computational intelligence techniques. Specifically, the blind deconvolution problem is first formulated within the framework of evolutionary strategy where a pool of candidate PSFs is generated to form the population in evolutionary strategy (ES). A new cost function that incorporates the specific requirement of blind deconvolution in the form of a point spread function domain regularization term, which ensures the emergence of a valid PSF, in addition to the previous data fidelity measure and image regularization term is adopted as the fitness function in the evolutionary algorithm. This new blind deconvolution approach will be described in Chapter 7.

1.6.2 Edge Characterization and Detection

The characterization of important features in an image requires the detailed specification of those pixel configurations that human beings would regard as significant. In this work, we consider the problem of representing human preferences, especially with regard to image interpretation, again in the form of a model-based neural network with hierarchical architecture [48,73,74]. Since it is difficult to represent all aspects of human preferences in interpreting images using traditional mathematical models, we encode these preferences through a direct *learning* process, using image pixel configurations that humans usually regard as visually significant as training examples. As a first step, we consider the problem of edge characterization in such a network. This representation problem is important since its successful solution would allow computer vision systems to simulate to a certain extent the decision process of human beings when interpreting images.

Although the network can be considered as a particular implementation of the stages of segmentation and characterization in the overall adaptive image-processing scheme, it can also be regarded as a self-contained adaptive image-processing system on its own: the network is designed such that it automatically partitions the edges in an image into different classes depending on the gray-level values of the surrounding pixels of the edge, and applies different detection thresholds to each of the classes. This is in contrast to the usual approach where a single detection threshold is adopted across the whole image independent of the local context. More importantly, instead of providing *quantitative* values for the threshold as in the usual case, the users are asked to provide *qualitative* opinions on what they regard as edges by manually tracing their desired edges on an image. The gray-level configurations around the trace are then used as training examples for the model-based neural network to acquire an internal model of the edges, which is another example of the design of an adaptive image-processing system through the training process.

As seen above, we have proposed the use of a hierarchical model-based neural network for the solution of both these problems as a first attempt. It was observed later that, whereas the edge characterization problem can be satisfactorily represented by this framework, resulting in adequate characterization of those image edges that humans regard as significant, there are some inadequacies in using this framework exclusively for the solution of the adaptive regularization problem, especially in those cases where the images are more severely degraded. These inadequacies motivate our later adoption of fuzzy set theory and evolutionary computation techniques, in addition to the previous neural network techniques, for this problem.

1.6.3 Self-Organizing Tree Map for Knowledge Discovery

Computational technologies based on artificial neural networks have been the focus for much research into the problem of unsupervised learning and data clustering, where the goal is to formulate or discover significant patterns or features from within a given set of data without the guidance of a teacher. Input patterns are usually stored as a set of prototypes or clusters: representations or natural groupings of similar data. In forming a description of an unknown set of data, such techniques find application across a range of industrial tasks that warrant significant need for data mining, that is, bioinformatics research, high-dimensional image analysis and visualization, information retrieval, and computer vision. Inherently unsupervised in nature, neural-network architectures based on principles of self-organization appear to be a natural fit.

Such architectures are characterized by their adherence to four key properties [34]: synaptic self-amplification for mining correlated stimuli, competition over limited resources, cooperative encoding of information, and the implicit ability to encode pattern redundancy as knowledge. Such principles are, in many ways, a reflection of Turing's observations in 1952 [75]: "Global order can arise from Local interactions." Such mechanisms exhibit a basis in the process of associative memory, and, receiving much neurobiological support, are believed to be fundamental to the organization that takes place in the human brain.

Static architectures such as Kohonen's self-organizing feature map (SOFM) [76] represent one of the most fundamental realizations of such principles, and have been the foundation for much neural network-based research into knowledge discovery. Their popularity arises out of their ability to infer an ordered or topologically preserved mapping of the underlying data space. Thus, relationships between discovered prototypes are captured in an output layer that is connected by some predefined topology. Mappings onto this layer are such that order is maintained: thus patterns near to one another in the input space map to nearby locations in an output layer. Such mechanisms are useful for qualitative visualization [77,78] of high dimensional, multivariate data, where users are left to perceive possible clustered regions in a dimension that is more familiar to them (2D or 3D).

The hierarchical feature map (HFM) [79] extends such ideas to a pyramidal hierarchy of SOFMs, each progressively trained in a top-down manner, to achieve some semblance of hierarchical partitioning on the input space. At the other end of the self-organizing spectrum is neural gas (NG) [80], which completely abandons any imposed topology: instead relying on the consideration of k nearest neighbors for the refinement of prototypes.

One of the most challenging tasks in any unsupervised learning problem arises by virtue of the fact that an attempt is being made to quantitatively discover a set of dominant patterns (clusters or classes) in which to categorize underlying data without any knowledge of what an appropriate number of classes might be. There are essentially two approaches taken as a result: either attempt to perform a series of separate runs of a static clustering algorithm over a range of different class numbers and assess which yields a better result according to some independent index of quality, or maintain a purely dynamic architecture that attempts to progressively realize an appropriate number of classes throughout the course of parsing a data set. The latter of the two approaches is advantageous from a resource and time of execution point of view.

Many dynamic neural network-based architectures have been proposed, as they seem particularly suited to developing a model of an input space, one item of data at a time: they evolve internally, through progressive stimulation by individual samples. Such dynamic architectures are generally hierarchical or nonstationary in nature, and extend upon HFM/SOFM such as in the growing hierarchical SOM (GHSOM) [81,82], or extend upon NG as in the growing neural gas (GNG) algorithm [83] and its associated variants: growing grid (GG) [84] and growing cell structures (GCS) [85].

1.6.4 Content-Based Image Categorization and Retrieval

Content-based image categorization is the process of classifying a given image into one of the predefined categories based on content analysis. Content analysis of images refers to the extraction of features such as color, texture, shape, or spatial relationship from the images as the signatures, and from which the indexes of the images are built. Content analysis can be categorized into spatial domain analysis and compressed domain analysis. Spatial domain analysis performs feature extraction in the original image domain. On the other hand, compressed domain analysis extracts features in the compressed domain directly in order to reduce the computational time involved in the decompression of the images.

Content-based image retrieval (CBIR) has been developed as an alternative search technique that complements text-based image retrieval. It utilizes content analysis to retrieve images that are similar to the query from the database. Users can submit their query to the systems in the form of an example image, which is often known as query-by-example (QBE). The systems then perform image content analysis by extracting visual features from the query and compare them with the features of the images in the database. After similarity comparison, the systems display the retrieval results to the users.

Content Analysis

Previous approaches for content-based image classification mainly focus on spatial domain analysis. This, however, is often expensive in terms of computational and storage requirements as most digital images nowadays are stored in the compressed formats. Feature extraction performed in the spatial domain requires the decompression of the compressed images first, resulting in significant computational cost. To alleviate the computational load, some works have focused on performing content analysis in the compressed domain. Most of the approaches that adopt compressed domain features give more emphasis on computational efficiency than their effectiveness in content representation. It is worth noting that often the compressed domain features may not fully represent the actual image contents. To address this issue, evolutionary computation techniques have been applied to obtain proper transformation of the compressed domain features. In doing so, image classification accuracy can be improved using transformed features while retaining the efficiency of the compressed domain techniques. The detailed algorithms will be explained in Chapter 10.

Relevance Feedback in CBIR

Many CBIR systems have been developed over the years that include both commercial and research prototypes. However, a challenging issue that restricts the performance of the CBIR systems is the semantic gap between the low-level visual features and the high-level human perception. To bridge the semantic gap, *relevance feedback* has been introduced into the CBIR systems. The main idea is to incorporate human in the loop to enhance the retrieval accuracy. Users are allowed to provide their relevance judgement on the retrieved images. The feedbacks are then learned by the systems to discover user information needs. There have been a lot of studies on relevance feedback in CBIR in recent years with various algorithms developed. Although the incorporation of relevance feedback has been shown to boost the retrieval performance, there are still two important issues that need to be considered when developing an efficient and effective CBIR system: (1) imprecision of user perception on the relevance of the feedback images and (2) the small sample problem:

- Typically, in most interactive CBIR systems, a user is expected to provide a binary decision of either "fully relevant" or "totally irrelevant" on the feedback images. At times, this may not agree with the uncertainty embedded in user perception. For example, in a scenario application, a user intends to find pets, especially, dogs. If the retrieved results contain cats, the user would face a dilemma as to whether to classify the cats as either fully relevant or totally irrelevant. This is because these cat images only satisfy user information need up to a certain extent. Therefore, we need to take the potential imprecision of user perception into consideration when developing relevance feedback algorithm.
- In an interactive CBIR system with relevance feedback, it is tedious for users to label many images. This gives rise to the small sample problem where learning from a small number of training samples restricts the retrieval performance.

To address these two challenges, computational intelligence techniques, namely, neural networks, clustering, fuzzy reasoning, and SVM will be employed due to their effectiveness. We will describe the proposed approaches in more details in Chapter 11.

1.7 Contributions of the Current Work

With regard to the problems posed by the requirements of segmentation, characterization, and optimization in the design of an adaptive image-processing system, we have devised a system of interrelated solutions comprising the use of the main algorithm classes of computational intelligence techniques. The contributions of the work described in this book can be summarized as follows.

1.7.1 Application of Neural Networks for Image Restoration

Different neural network models, which will be described in Chapters 2, 3, 4, and 5, are adopted for the problem of image restoration. In particular, a model-based neural network with hierarchical architecture [48,73,74] is derived for the problem of adaptive regularization. The image is segmented into smooth regions and combined edge/textured regions, and we assign a single subnetwork to each of these regions for the estimation of the regional parameters. An important new concept arising from this work is our alternative viewpoint of the regularization parameters as model-based neuronal weights, which are then trainable through the supply of proper training examples. We derive the training examples through the application of *adaptive nonlinear filtering* [86] to individual pixel neighborhoods in the image for an *independent* estimate of the current pixel value.

1.7.2 Application of Neural Networks to Edge Characterization

A model-based neural network with hierarchical architecture is proposed for the problem of edge characterization and detection. Unlike previous edge-detection algorithms where various threshold parameters have to be specified [2,4], this parameterization task can be performed implicitly in a neural network by supplying training examples. The most important concept in this part of the work is to allow human users to communicate their preferences to the adaptive image-processing system through the provision of qualitative training examples in the form of edge tracings on an image, which is a more natural way of specifying preferences for humans, than the selection of *quantitative* values for a set of parameters. With the adoption of this network architecture and the associated training algorithm, it will be shown that the network can generalize from sparse examples of edges provided by human users to detect all significant edges in images not in the training set. More importantly, no retraining and alteration of architecture is required for applying the same network to noisy images, unlike conventional edge detectors that usually require threshold readjustment.

1.7.3 Application of Fuzzy Set Theory to Adaptive Regularization

For the adaptive regularization problem in image restoration, apart from the requirement of adopting different regularization parameters for smooth regions and regions with high gray-level variances, it is also desirable to further separate the latter regions into edge and textured regions. This is due to the different noise masking capabilities of these two feature types, which in turn requires different regularization parameter values. In our previous discussion of fuzzy set theory, we have described a possible solution to this problem, in the form of characterizing the gray-level configurations corresponding to the above two feature types, and then define fuzzy sets with qualifying terms like "texture" and "edge" over the respective sets of configurations. However, one of the problems with this approach is that there is usually an extremely large number of possible gray-level configurations corresponding to each feature type, and human beings cannot usually relate what they perceive as a certain feature type to a particular configuration. In Chapter 5, a *scalar* measure has been established that characterizes the degree of resemblance of a gray-level configuration to either textures or edges. In addition, we can establish the exact interval of values of this measure where the configuration will more resemble textures than edges and vice versa. As a result, we can readily define fuzzy sets over this one-dimensional *universe of discourse* [37].

1.7.4 Application of Evolutionary Programming to Adaptive Regularization and Blind Deconvolution

Apart from the neural network-based techniques, we have developed an alternative solution to the problem of adaptive regularization using evolutionary programming, which is a member of the class of evolutionary computational algorithms [39,40]. Returning again to the ETC measure, we have observed that the distribution of the values of this quantity assumes a typical form for a large class of images. In other words, the shape of the probability density function (pdf) of this measure is similar across a broad class of images and can be modeled using piecewise continuous functions. On the other hand, this pdf will be different for blurred images or incorrectly regularized images. As a result, the model pdf of the ETC measure serves as a kind of *signature* for correctly regularized images, and we should minimize the difference between the corresponding pdf of the image being restored and the model pdf using some kind of distance measure. The requirement to approximate this pdf using a histogram, which involves the counting of discrete quantities, and the resulting nondifferentiability of the distance measure with respect to the various regularization parameters, necessitates the use of evolutionary computational algorithms for optimization. We have adopted evolutionary programming that, unlike the genetic algorithm which is another widely applied member of this class of algorithms, operates directly on real-valued vectors instead of binary-coded strings and is therefore more suited to the adaptation of the regularization parameters. In this algorithm, we have derived a parametric representation that expresses the regularization parameter value as a function of the local image variance. Generating a population of these *regularization strategies* that are vectors of the above hyperparameters, we apply the processes of mutation, competition, and selection to the members of the population to obtain the optimal regularization strategy. This approach is then further extended to solve the problem of blind deconvolution by including the point spread function coefficients in the set of hyperparameters associated with each individual in the population.

1.7.5 Application of Self-Organization to Image Analysis and Retrieval

A recent approach known as the self-organizing tree map (SOTM) [87] inherently incorporates hierarchical properties by virtue of its growth, in a manner that is far more flexible in terms of revealing the underlying data space without being constrained by an imposed topological framework. As such, the SOTM exhibits many desirable properties over traditional SOFM-based strategies.

Chapter 9 of the book will provide an in-depth coverage of this architecture. Due to the adaptive nature, this family of unsupervised methods exhibits a number of desirable properties over the SOFM and its early derivatives such as (1) better topological preservation to ensure the ability to adapt to different datasets; (2) consistent topological descriptions of the underlying datasets; (3) robust and succinct allocation of cluster prototypes; (4) built-in awareness of topological information, local density, and variance indicators for optimal selection of cluster prototypes at runtime; and (5) a true automatic mode to deduce simultaneously, optimal number of clusters, their prototypes, and an appropriate topological mapping associating them.

Chapter 9 will then cover a series of pertinent real-world applications with regards to the processing of image and video data—from its role in more generic image-processing techniques such as the automated modeling and removal of impulse noise in digital images, to problems in digital asset management including the modeling of image and video content, indexing, and intelligent retrieval.

1.7.6 Application of Evolutionary Computation to Image Categorization

To address the issue of accuracy in content representation that is crucial for compressed domain image classification, we propose to perform transformation on the compressed-domain features. These feature values are modeled as realizations of random variables. The transformation on the random variable is then carried out by the merging and removal of histogram bin counts. To search for the optimal transformation on the random variable, genetic algorithm (GA) has been employed to perform the task. The approach has been further extended by adopting individually optimized transformations for different image classes, where a set of separate classification modules is associated with each of these transformations.

1.7.7 Application of Computational Intelligence to Content-Based Image Retrieval

In order to address the imprecision of user perception in relevance feedback of CBIR systems, a fuzzy labeling scheme that integrates the user's uncertain perception of image similarity is proposed. In addition to the "relevant" and "irrelevant" choices, the proposed scheme provides a third "fuzzy" option to the user. The user can provide a feedback as "fuzzy" if the image only satisfies his or her partial information needs. Under this scheme, the soft relevance of the fuzzy images has to be estimated. An a posteriori probability estimator is developed to achieve this. With the combined relevant, irrelevant, and fuzzy images, a recursive fuzzy radial basis function network (RFRBFN) has been developed to learn the user information needs. To address the small sample problem in relevance feedback, a predictive-label fuzzy support vector machine (PLFSVM) framework has been developed. Under this framework, a clustering algorithm together with consideration of the correlation between