CHAPTER 11

IMAGE REPRESENTATION, ANALYSIS, AND CLASSIFICATION

An image is considered a representation of objects with specific properties that are utilized in the imaging process. The characterization (or appearance) of objects in the image depends on their specific properties and interaction with the imaging process. For example, an image of an outdoor scene captured by a photographic camera provides a characterization of objects using the reflected light from their surfaces. Thus, the appearance of objects in the image will depend on the reflective properties of surfaces, illumination, viewing angle, and the optics of the photographic camera. As described in Chapters 4–7, different medical imaging modalities provide different characterizations of the physiological objects based on their properties and the type and parameters of the imaging modality. For example, a T_1-weighted magnetic resonance (MR) image of the brain would show a different contrast of soft tissue, ventricles, and cerebrospinal fluid than its T_2-weighted MR image.

In order to perform a computerized analysis of an image, it is important to establish a hierarchical framework of processing steps representing the image (data) and knowledge (model) domains. The hierarchical image representation with bottom-up and top-down analysis approaches is schematically presented in Figure 11.1. A scenario consists of a multiple image data set involving multidimensional, multimodality, or multisubject images and can be represented as a set of scenes consisting of specific objects in the corresponding images. The objects can be represented into surface regions (S-regions) consisting of one or more regions formed by contours and edges defining specific shapes of object structures.

The bottom-up analysis starts with the analysis at the pixel-level representation and moves up toward the understanding of the scene or the scenario (in the case of multiple images). The top-down analysis starts with the hypothesis of the presence of an object and then moves toward the pixel-level representation to verify or reject the hypothesis using the knowledge-based models. It is difficult to apply a complete bottom-up or top-down approach for efficient image analysis and understanding. Usually, an appropriate combination of both approaches provides the best results. Pixel-level analysis, also known as low-level analysis, can incorporate a bottom-up approach for the edge and region segmentation. The characteristic features of

Medical Image Analysis, *Second Edition*, by Atam P. Dhawan

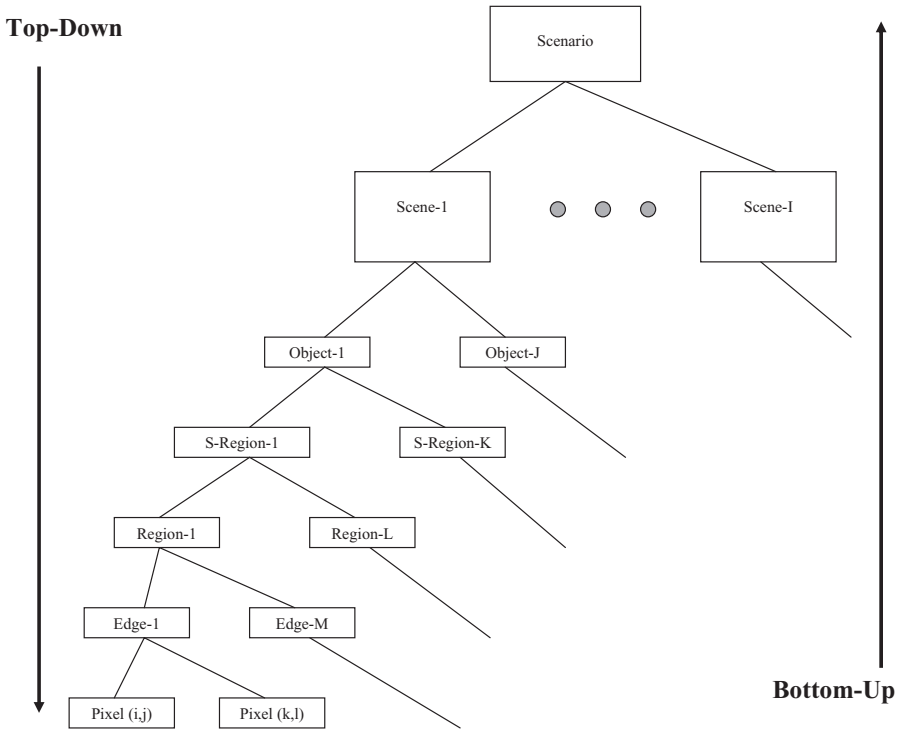


Figure 11.1 A hierarchical representation of image features.

segmented regions can be extracted for preliminary region and object analysis. Features can then be analyzed for object identification and classification using knowledge-based models with a top-down approach.

Figure 11.2 shows a paradigm of image analysis with associated structure of knowledge-based models that can be used at different stages of processing. The knowledge of physical constraints and tissue properties can be very useful in imaging and image reconstruction. For example, knowledge of the interaction of a radioactive pharmaceutical drug with body tissue is an important consideration in nuclear medicine imaging modalities. To improve contrast in proton density-based MR imaging, a paramagnetic contrast agent such as gadolinium (Gd) is used. The imaging sequences in MR imaging are specifically modified using knowledge of the physical properties of tissues and organs to be imaged. Anatomical locations of various organs in the body often impose a challenge in imaging the desired tissue or part of the organ. Using knowledge of physical regions of interest, data collection image reconstruction methods can be modified. The performance of image reconstruction algorithms can be improved using some physical models and properties of the imaging process. The pixel-level image analysis for edge and region segmentation can also be improved using probabilistic and knowledge-based models of expected regions and objects. For example, Hough transform can be used to extract specific regions of target shapes. A model-based tissue segmentation algorithm is described

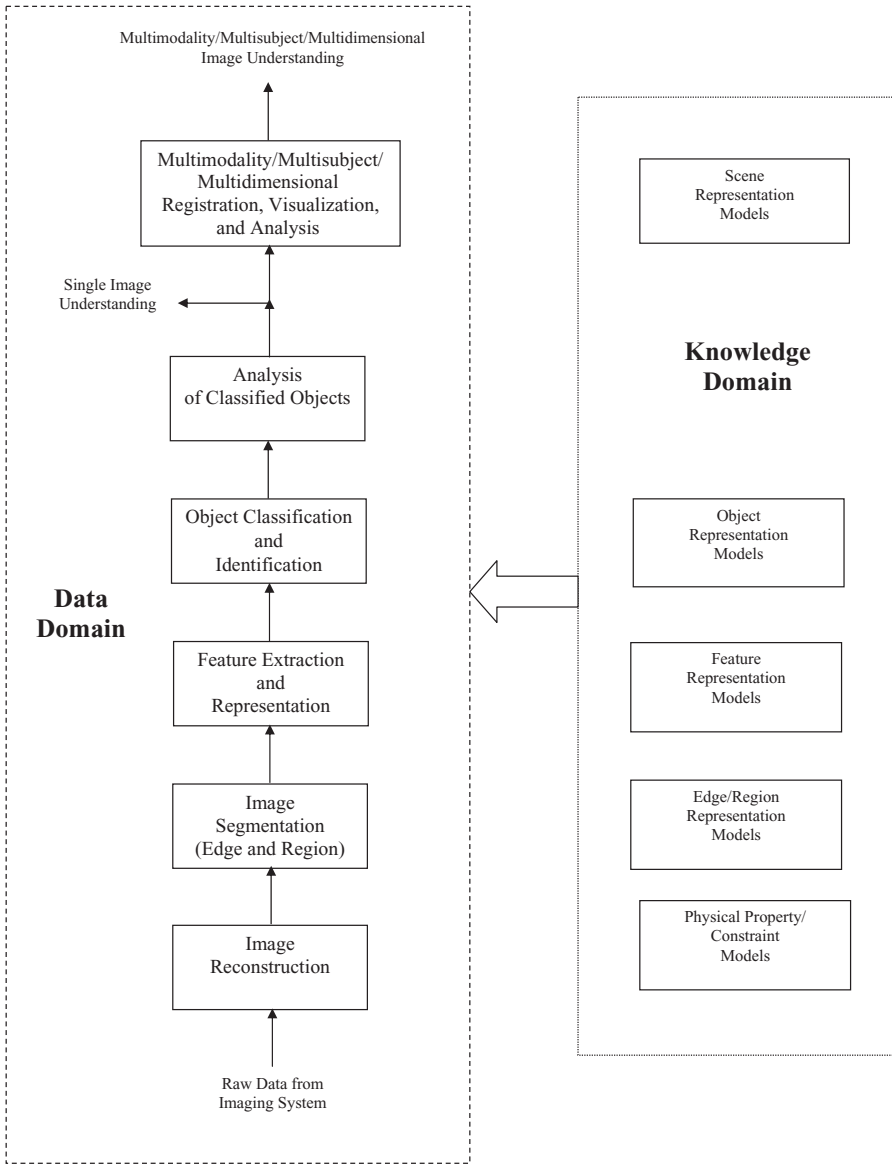


Figure 11.2 A hierarchical structure of medical image analysis.

in Chapter 10. Furthermore, the characteristic features of segmented regions can be efficiently analyzed and classified using probabilistic and knowledge-based models of expected objects. An object representation model usually provides the knowledge about the shape or other characteristic features of a single object for the classification analysis. In addition, the scene interpretation models can provide knowledge about the geometrical and relational features of other objects present in the image. For example, a three-dimensional (3-D) anatomical chest cavity model can provide

shape and relational information of the major cardiac chambers (left ventricle, right ventricle, left atrium, and right atrium) along with the cardiac cage, lungs, and liver. Such models are quite useful in improving the classification and understanding of occluded objects in the image.

This chapter describes feature extraction methods for region representation followed by classification methods for object identification and understanding.

11.1. FEATURE EXTRACTION AND REPRESENTATION

As described above, after segmentation, specific features representing the characteristics and properties of the segmented regions in the image need to be computed for object classification and understanding. Features are also important for measurements of parameters leading to direct image understanding. For example, a quantitative analysis of MR perfusion time-series images provides information about the parameters of cerebral perfusion. A complete feature representation of the segmented regions is critical for object classification and analysis. There are four major categories of features for region representation:

1. **Statistical Pixel-Level (SPL) Features:** These features provide quantitative information about the pixels within a segmented region. The SPL features may include mean, variance, and histogram of the gray values of pixels in the region. In addition, SPL features may include the area of the region and information about the contrast of pixels within the region and edge gradient of boundary pixels.
2. **Shape Feature:** These features provide information about the characteristic shape of the region boundary. The shape-based features may include circularity, compactness, moments, chain-codes, and Hough transform. Recently, morphological processing methods have also been used for shape description.
3. **Texture Features:** These features provide information about the local texture within the region or the corresponding part of the image. The texture features may be computed using the second-order histogram statistics or co-occurrence matrices. In addition, wavelet processing methods for spatio-frequency analysis have been used to represent local texture information.
4. **Relational Features:** These features provide information about the relational and hierarchical structure of the regions associated with a single object or a group of objects.

11.1.1 Statistical Pixel-Level Features

Once the regions are segmented in the image, gray values of pixels within the region can be used for computing the following SPL features (1, 2):

1. The histogram of the gray values of pixels in the image as

$$p(r_i) = \frac{n(r_i)}{n} \tag{11.1}$$

where $p(r_i)$ and $n(r_i)$ are, respectively, the probability and number of occurrences of a gray value r_i in the region and n is the total number of pixels in the region.

2. Mean m of the gray values of the pixels in the image can be computed as

$$m = \frac{1}{n}\sum_{i=0}^{L-1} r_i p(r_i) \tag{11.2}$$

where L is the total number gray values in the image with 0, 1, … , $L - 1$.

3. Variance and central moments in the region can be computed as

$$\mu_n = \sum_{i=0}^{L-1} p(r_i)(r_i - m)^n \tag{11.3}$$

where the second central moment μ_2 is the variance of the region. The third and fourth central moments can be computed, respectively, for $n = 3$ and $n = 4$. The third central moment is a measure of noncentrality, while the fourth central moment is a measure of flatness of the histogram.

4. Energy: Total energy E of the gray values of pixels in the region is given by

$$E = \sum_{i=0}^{L-1} [p(r_i)]^2. \tag{11.4}$$

5. Entropy: The entropy *Ent* as a measure of information represented by the distribution of gray values in the region is given by

$$Ent = -\sum_{i=0}^{L-1} p(r_i)\log_2(r_i). \tag{11.5}$$

6. Local contrast corresponding to each pixel can be computed by the difference between the gray value of the center pixel and the mean of the gray values of the neighborhood pixels. An adaptive method for computing contrast values is described in Chapter 7. The normalized local contrast $C(x, y)$ for the center pixel can also be computed as

$$C(x,y) = \frac{|P_c(x,y) - P_s(x,y)|}{\max\{P_c(x,y), P_s(x,y)\}} \tag{11.6}$$

where $P_c(x, y)$ and $P_s(x, y)$ are the average gray-level values of the pixels corresponding to the center and the surround regions respectively centered on the pixel (Chapter 9).

7. Additional features such as maximum and minimum gray values can also be used for representing regions.

8. The features based on the statistical distribution of local contrast values in the region also provide useful information about the characteristics of the regions representing objects. For example, features representing the mean, variance, energy, and entropy of contrast values in the segmented regions contribute significantly in classification analysis of X-ray mammograms (3).

9. Features based on the gradient information for the boundary pixels of the region are also an important consideration in defining the nature of edges. For example, the fading edges with low gradient form a characteristic feature of malignant melanoma and must be included in the classification analysis of images of skin lesions (4, 5).

11.1.2 Shape Features

There are several features that can be used to represent the geometric shape of a segmented region. The shape of a region is basically defined by the spatial distribution of boundary pixels. A simple approach for computing shape features for a two-dimensional (2-D) region is representing circularity, compactness, and elongatedness through the minimum bounded rectangle that covers the region. For example, Figure 11.3 shows a segmented region and its minimum bounded rectangle ABCD.

Several features using the boundary pixels of the segmented region can be computed as

1. Longest axis GE

2. Shortest axis HF

3. Perimeter and area of the minimum bounded rectangle ABCD

4. Elongation ratio: GE/HF

5. Perimeter p and area A of the segmented region

6. Hough transform of the region using the gradient information of the boundary pixels of the region (as described in Chapter 7)

7. Circularity ($C = 1$ for a circle) of the region computed as

$$C = \frac{4\pi A}{p^2}. \tag{11.7}$$

8. Compactness C_p of the region computed as

$$C_p = \frac{p^2}{A}. \tag{11.8}$$

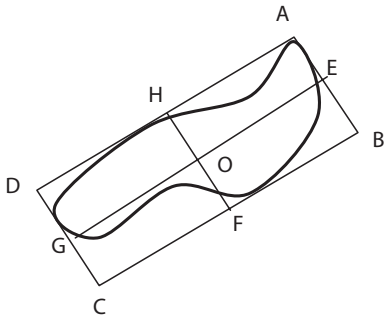


Figure 11.3 A segmented region with a minimum bounded region.

9. Chain code for boundary contour as obtained using a set of orientation primitives on the boundary segments derived from a piecewise linear approximation
10. Fourier descriptor of boundary contours as obtained using the Fourier transform of the sequence of boundary segments derived from a piecewise linear approximation
11. Central moments-based shape features for the segmented region
12. Morphological shape descriptors as obtained though morphological processing on the segmented region

Shape features using the chain code, Fourier descriptor, central moments, and morphological processing are described below.

11.1.2.1 Boundary Encoding: Chain Code Let us define a neighborhood matrix with orientation primitives with respect to the center pixel as shown in Figure 11.4. The codes of specific orientation are set for 8-connected neighborhood directions based on the location of the end of the boundary segment with respect to its origin at the center x_c. Thus, the orientation directions are coded with a numerical value ranging from 0 to 7. To apply these orientation codes to describe a boundary, the boundary contour needs to be approximated as a list of segments that have preselected length and directions. For example, the boundary of the region shown in Figure 11.5 is approximated in segments using the directions of 8-connected neighborhood and orientation codes shown in Figure 11.4. To obtain boundary segments representing a piecewise approximation of the original boundary contour, a discretization method, such as "divide and conquer," is applied. The divide and conquer method for curve approximation selects two points on a boundary contour as vertices. These vertices can be selected arbitrarily or on the basis of gradient information. Usually points with a significant change in the gradient direction are selected as potential vertices. A straight line joining the two selected vertices can be used to approximate the respective curve segment if it satisfies a maximum-deviation criterion for no further division of the curve segment. The maximum-deviation criterion is based on the perpendicular distance between any point on the original curve segment between the selected vertices and the corresponding approximated straight-line segment. If the perpendicular distance or deviation of any point on the curve

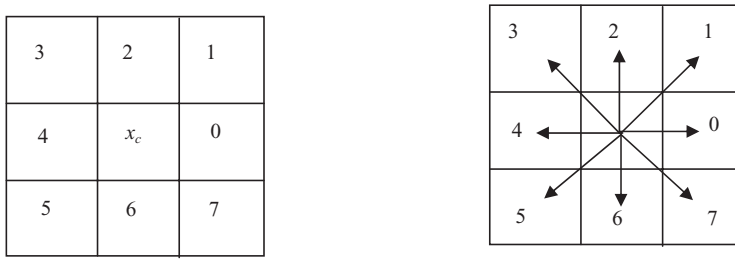


Figure 11.4 The 8-connected neighborhood codes (left) and the orientation directions (right) with respect to the center pixel x_c.

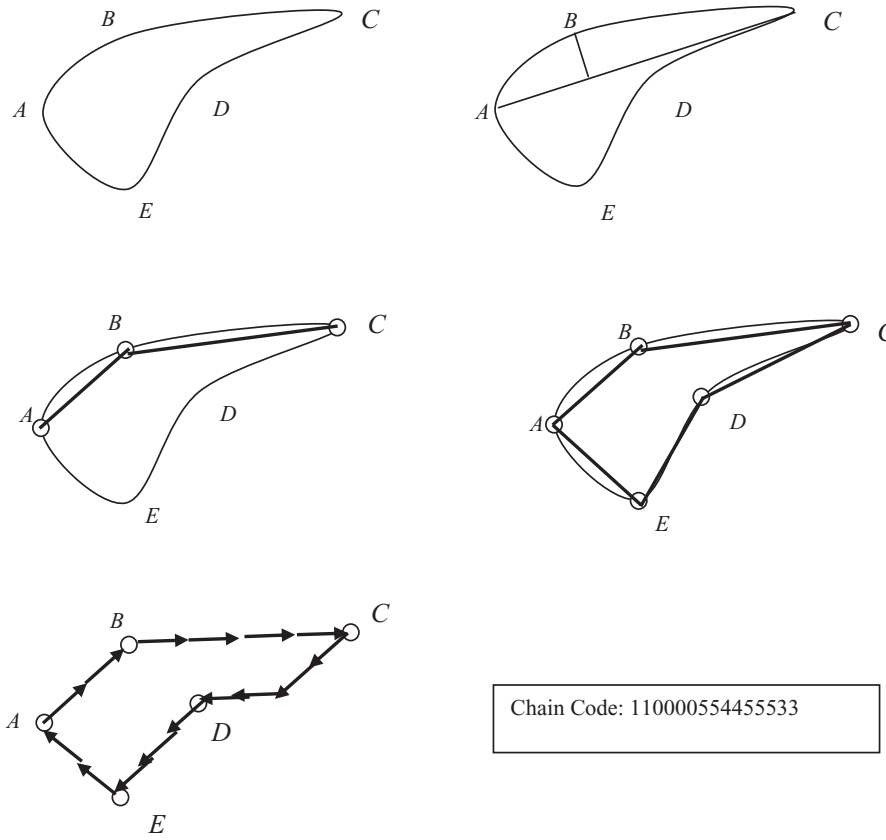


Figure 11.5 A schematic example of developing chain code for a region with boundary contour *ABCDE*. From top left to bottom right: the original boundary contour, two points *A* and *C* with maximum vertical distance parameter *BF*, two segments *AB* and *BC* approximating the contour *ABC*, five segments approximating the entire contour *ABCDE*, contour approximation represented in terms of orientation primitives, and the respective chain code of the boundary contour.

segment from the approximated straight-line segment exceeds a preselected deviation threshold, the curve segment is further divided at the point of maximum deviation. This process of dividing the segments with additional vertices continues until all approximated straight-line segments satisfy the maximum-deviation criterion. For example, two points *A* and *C* on the boundary contour *ABCDE* shown in Figure 11.5 have significant change in the direction of their respective gradients and are therefore taken as initial vertices for curve approximation. It can be seen in the example shown in Figure 11.5 that the approximated straight-line segment *AC* does not satisfy the maximum-deviation criterion, that is, the perpendicular distance BF is more than an acceptable threshold. The boundary arc segment *AC* is then further divided into two segments *AB* and *AC* with their respective approximated straight-line segments. As shown in Figure 11.5, this process is continued until a final approximation of five straight-line segments, *AB*, *BC*, *CD*, *DE*, and *EA* is obtained.

This representation is further approximated using the orientation primitives of the 8-connected neighborhood as defined in Figure 11.4. The chain code descriptor of the boundary can now be obtained using the orientation primitive based approximated representation. The boundary segment shown in Figure 11.5 can thus have a chain code *110000554455533*. It should be noted that though this code is starting at the vertex A, it is circular in nature. Two parameters can change the chain code: number of orientation primitives and the maximum-deviation threshold used in approximating the curve. However, other methods for approximating curves can also be used before orientation primitives are applied to obtain chain codes (1, 5, 6).

11.1.2.2 Boundary Encoding: Fourier Descriptor Fourier series may be used to approximate a closed boundary of a region. Let us assume that the boundary of an object is expressed as a sequence of N points (or pixels) with the coordinates $\mathbf{u}[n] = \{x(n), y(n)\}$ such that

$$u(n) = x(n) + iy(n); \quad n = 0,1,2,\ldots,N-1. \tag{11.9}$$

The discrete Fourier transform (DFT) of the sequence $\mathbf{u}[n]$ is the Fourier descriptor $\mathbf{F}_d[n]$ of the boundary and is defined as

$$\mathbf{F}_d[m] = \frac{1}{N}\sum_{n=0}^{N-1} u(n)e^{-2\pi im/N} \quad \text{for } 0 \le m \le N-1. \tag{11.10}$$

Rigid geometric transformation of a boundary such as translation, rotation, and scaling can be represented by simple operations on its Fourier transform. Thus, the Fourier descriptors can be used as shape descriptors for region matching dealing with translation, rotation, and scaling. Fourier descriptor-based boundary representation models have been used in medical imaging for segmentation and object identification (7).

11.1.2.3 Moments for Shape Description The shape of a boundary or contour can be represented quantitatively by the central moments for matching. The central moments represent specific geometrical properties of the shape and are invariant to translation, rotation, and scaling. The central moments μ_{pq} of a segmented region or binary image $f(x, y)$ are given by (8, 9)

$$\mu_{pq} = \sum_{i=1}^{L}\sum_{j=1}^{L}(x_i - \overline{x})^p (y_j - \overline{y})^q f(x,y)$$

where

$$\overline{x} = \sum_{i=1}^{L}\sum_{j=1}^{L} x_i f(x_i, y_j)$$

$$\overline{y} = \sum_{i=1}^{L}\sum_{j=1}^{L} y_j f(x_i, y_j). \tag{11.11}$$

For example, the central moment μ_{21} represents the vertical divergence of the shape of the region indicating the relative extent of the bottom of the region compared with the top. The normalized central moments are then defined as

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{\gamma}}$$

where

$$\gamma = \frac{p+q}{2} + 1. \tag{11.12}$$

The seven invariant moments $\phi_1 - \phi_7$ for shape matching are defined as (9)

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
\phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
&\quad + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].
\end{aligned} \tag{11.13}$$

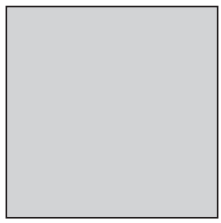
The invariant moments are used extensively in the literature for shape matching and pattern recognition (2, 6–10).

11.1.2.4 Morphological Processing for Shape Description Mathematical morphology is based on set theory. It provides basic tools to process images for filtering, thinning, and pruning operations. These tools are useful in the description of region shape, involving boundary and skeleton representations.

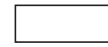
There are two fundamental operations in morphological processing: dilation and erosion. Most of the morphological processing algorithms are based on specific combinations of the dilation and erosion operations. Let us define two sets, A and B, belonging to an n-dimensional space Z^n. For gray-level images, the parameter n is equal to 3 considering 2-D space for x- and y-scoordinates and the third dimension for the gray values of pixels in the image. For a binary or segmented image with regions, the parameter n takes the value of 2 since the sets A and B can represent the boolean values of "1" (within the set) and "0" (outside the set). For simplicity, only binary or region-based ssegmented images are considered here with a 2-D representation of sets A and B. However, they can be extended to 3-D space for gray-level morphological image processing.

The set A can be assumed to be the binary image while the set B is considered to be the structuring element of the same dimensions. For example, Figure 11.6 shows a large region representing the set A and a smaller region representing the set B for the structuring element. The dilation of set A by the set B, $\boldsymbol{D}(A, B)$ is denoted by $A \oplus B$ and defined by the Minkowski set addition as

$$\mathrm{D}(A, B) = A \oplus B = \bigcup_{b \in B} A + b. \tag{11.14}$$

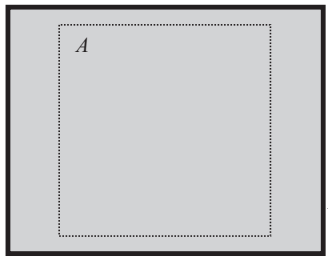


Set A

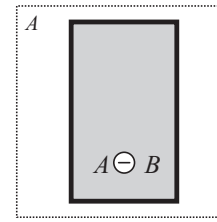


Set B

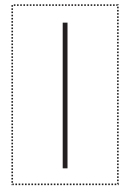
Figure 11.6 A large region with square shape representing the set A and a small region with rectangular shape representing the structuring element set B.



$A \oplus B$: Dilation of A by B



$A \ominus B$: Erosion of A by B



$(A \ominus B) \ominus B$

Figure 11.7 The dilation of set A by the structuring element set B (top left), the erosion of set A by the structuring element set B (top right), and the result of two successive erosions of set A by the structuring element set B (bottom).

It can be shown that

$$A \oplus B = \left\{ x | (\check{B})_x \cap A \neq 0 \right\} \tag{11.15}$$

where $\check{B} = \{-b | b \in B\}$ is the reflection set of B with respect to its origin.

Equation 11.15 states that the dilation of a set (shape) A by a structuring element B comprises all points x such that the reflected structuring element $\check{B}$ translated to x intersects A. Figure 11.7 shows the dilation of a shape (of a region) A by the structuring element B.

As all operations in set theory have their respective dual operations, the dual operation of dilation is erosion. The erosion of set (shape) A by the set B (structuring

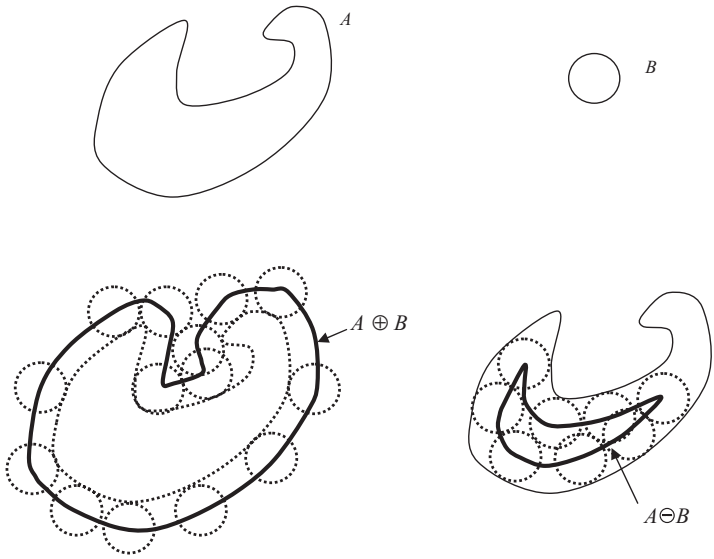


Figure 11.8 Dilation and erosion of an arbitrary shape region A (top left) by a circular structuring element B (top right): dilation of A by B (bottom left) and erosion of A by B (bottom right).

element), $\boldsymbol{E}(A, B)$, is denoted by $A \ominus B$ and is defined by the Minkowski set subtraction as

$$\mathrm{E}(A,B) = A \ominus B = \bigcap_{b \in B} A - b. \tag{11.16}$$

It can be shown that

$$A \ominus B = \{x \mid (B)_x \subseteq A\}. \tag{11.17}$$

Equation 11.17 states that the erosion of a set (shape) A by the structuring element B comprises all points x such that the structuring element B located at x is entirely inside A.

The dilation and erosion of an arbitrary shape region A by a circular structuring element set B is shown in Figure 11.8. It can be seen that the dilation operation provides the property of increasingness (dilated regions are larger) while the erosion provides the property of decreasingness (eroded regions are smaller).

As mentioned above, the dilation and erosion are dual operations to each other, that is,

$$A \oplus B = \left(A^c \ominus \breve{B}\right)^c$$

and

$$A \ominus B = \left(A^c \oplus \breve{B}\right)^c \tag{11.18}$$

where A^c represents the complement of set A.

Both of these operations also provide translation variance and distributive properties as

$$\begin{aligned}
&(A+x)\oplus B = A\oplus(B+x) = (A\oplus B)+x && \text{Translation invariance of dilation}\\
&(A_1\cup A_2)\oplus B = (A_1\oplus B)\cup(A_2\oplus B) && \text{Distributivity of union of dilation}\\
&(A+x)\ominus B = A\ominus(B-x) = (A\ominus B)+x && \text{Translation invariance of dilation}\\
&(A_1\cap A_2)\ominus B = (A_1\ominus B)\cap(A_2\ominus B) && \text{Distributivity of intersection of erosion}
\end{aligned} \tag{11.19}$$

Dilation and erosion operations successively change the shape of the region or binary image. The successive application of erosion as shown in Figure 11.7 can be used to describe the shape of the region. However, combinations of dilation and erosion such as opening and closing operations can also be used effectively for shape description (11–14).

The morphological opening of set A by the structuring element set B, $A\circ B$, is defined as the composition of erosion followed by dilation as

$$A\circ B = (A\ominus B)\oplus B. \tag{11.20}$$

The morphological closing of set A by the structuring element set B, $A\circ B$, is defined as the composition of erosion followed by dilation as

$$A\bullet B = (A\oplus B)\ominus B. \tag{11.21}$$

The advantage of using morphological opening and closing operations in shape description is that successive applications of these operations do not change the shape of the image or region. It can be shown that the union of openings is an opening, whereas the intersection of closings is a closing (11). Figure 11.9

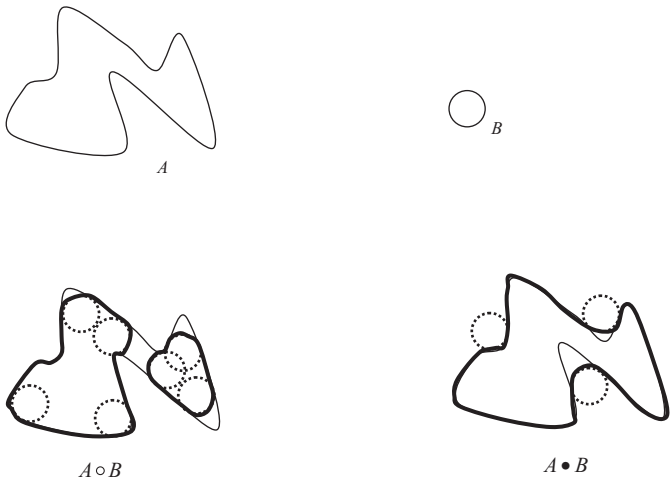


Figure 11.9 The morphological opening and closing of set A (top left) by the structuring element set B (top right): opening of A by B (bottom left) and closing of A by B (bottom right).

shows the opening and closing of an arbitrary shape by a circular structuring element.

Morphological decomposition methods have been used for shape representation and description (11–14) and applied in 3-D computed tomography (CT) image analysis of intracerebral brain hemorrhage (15). Let F be an input binary image or a segmented region and S be a structuring element for a shape decomposition process. Using a multiscale representation with a scale parameter r, a shape description of an image F can be given by (14, 15)

$$Z(r,0),\ldots,Z(r,N-1)$$
$$\text{with } Z(r,n)=\mathbf{M}(r,F,S^n) \tag{11.22}$$

where $\mathbf{M}$ is a morphological operation such as erosion, dilation, opening, or closing; r is the scale parameter; and $n = 0, 1, \ldots, N-1$ is a parameter that controls the size and shape of the structuring element S.

Morphological shape descriptors using a specific set of structuring elements on multiple scales can provide a translation, size, and rotation invariant shape representation (14, 16).

A popular method for shape description is using morphological operations for skeleton representation of regions of interest. A skeleton $K(A)$ of a set A using the structuring element set B can be computed using the erosion and opening operations as

$$K(A)=\bigcup_{n=0}^{N}K_n(A)$$
with
$$K_n(A)=(A\ominus nB)-(A\ominus nB)\circ B \tag{11.23}$$

where $(A\ominus nB)$ represents n successive erosions of the set A by the structuring element set B, and N denotes the last iterative step before the set A erodes to an empty set.

Figure 11.7 shows the skeleton of the set A that is obtained by two successive erosions by the structuring element set B. It should be noted that in this example, a customized shape of the structuring element is used which happens to provide the skeleton representation in just two erosions. In practice, a predefined set of structuring elements can be tried on a set of training shapes to determine the best structuring element (16) that can provide a good skeleton representation using the method described in Equation 11.23.

Morphological operations can also be used in image processing such as image smoothing, segmentation, boundary detection, and region filling. For example, morphological erosion can significantly reduce the background noise in the image. An opening operation can remove the speckle noise and provide smooth contours to the spatially distributed structures in the image. A closing operation preserves the peaks and reduces the sharp variations in the signal such as dark artifacts. Thus, a morphological opening followed by closing can reduce the bright and dark artifacts and noise in the image. The morphological gradient image can be obtained by subtracting

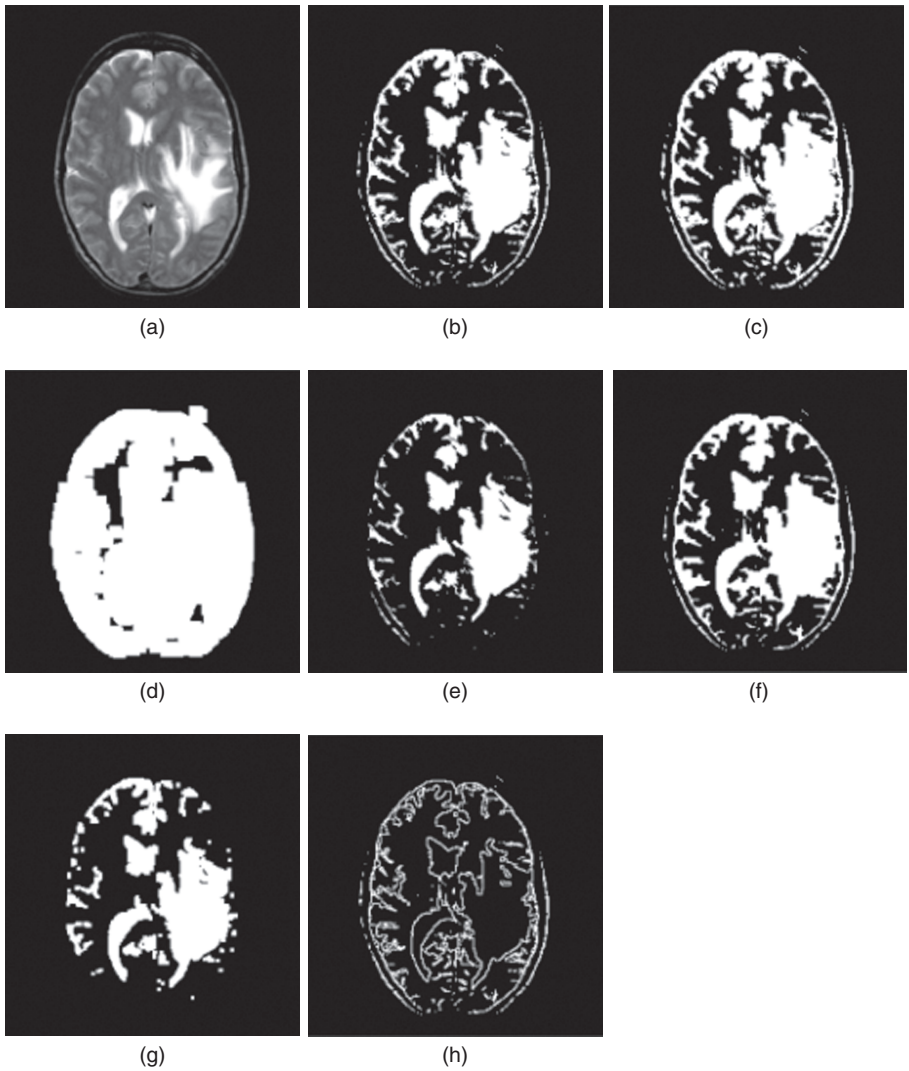


Figure 11.10 Example of morphological operations on MR brain image using a structuring element of $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ (a) the original MR brain image; (b) the thresholded MR brain image for morphological operations; (c) dilation of the thresholded MR brain image; (d) resultant image after five successive dilations of the thresholded brain image; (e) erosion of the thresholded MR brain image; (f) closing of the thresholded MR brain image; (g) opening of the thresholded MR brain image; and (h) morphological boundary detection on the thresholded MR brain image.

the eroded image from the dilated image. Edges can also be detected by subtracting the eroded image from the original image.

Figure 11.10 shows an MR brain image and the results of morphological operations applied to thresholded MR brain image.

11.1.3 Texture Features

Texture is an important spatial property that can be used in region segmentation as well as description. There are three major approaches to represent texture: statistical, structural, and spectral. Since texture is a property of the spatial arrangements of the gray values of pixels, the first-order histogram of gray values provides no information about the texture. Statistical methods representing the higher-order distribution of gray values in the image are used for texture representation. The second approach uses structural methods such as arrangements of prespecified primitives in texture representation. For example, a repetitive arrangement of square and triangular shapes can produce a specific texture. The third approach is based on spectral analysis methods such as Fourier and wavelet transforms. Using spectral analysis, texture is represented by a group of specific spatio-frequency components.

The gray-level co-occurrence matrix (GLCM) exploits the higher-order distribution of gray values of pixels that are defined with a specific distance or neighborhood criterion. In the simplest form, the GLCM $P(i, j)$ is the distribution of the number of occurrences of a pair of gray values i and j separated by a distance vector $\boldsymbol{d}=[dx, dy]$. For example, Figure 11.11 shows a matrix representation of an image with three gray values with its GLCM $P(i, j)$ for $\boldsymbol{d}=[1, 1]$.

The GLCM can be normalized by dividing each value in the matrix by the total number of occurrences, providing the probability of occurrence of a pair of gray values separated by a distance vector. Statistical texture features are computed from the normalized GLCM as the second-order histogram $H(y_q, y_r, \boldsymbol{d})$ representing the probability of occurrence of a pair of gray values yq and yr separated by a distance vector $\boldsymbol{d}$. Texture features can also be described by a difference histogram, $H_d(y_s, \boldsymbol{d})$, where $y_s = |y_q - y_r|$. $H_d(y_s, \boldsymbol{d})$ indicates the probability that a difference in gray levels exists between two distinct pixels. Commonly used texture features based on the second-order histogram statistics are as follows.

1. Entropy of $H(y_q, y_r, d)$, S_H:

$$S_H = -\sum_{y_q=y1}^{y_t}\sum_{y_r=y1}^{y_t} H(y_q, y_r, d)\log_{10}[H(y_q, y_r, d)]. \tag{11.24}$$

2	2	2	0	1
0	2	2	1	1
0	1	1	2	0
1	2	2	0	1
2	1	0	1	1

(a)

0	3	1	0
2	1	1	1
1	4	3	2 i
0	1	2	

j

(b)

Figure 11.11 (a) A matrix representation of a 5×5 pixel image with three gray values; (b) the GLCM $P(i, j)$ for $\boldsymbol{d} = [1, 1]$.

Entropy is a measure of texture nonuniformity. Lower entropy values indicate greater structural variation among the image regions.

2. Angular second moment of $H(y_q, y_r, d)$, ASM_H:

$$ASM_H = \sum_{y_q=y_1}^{y_t} \sum_{y_q=y_1}^{y_t} [H(y_q, y_r, d)]^2. \tag{11.25}$$

The ASM_H indicates the degree of homogeneity among textures, and is also representative of the energy in the image. A lower value of ASM_H is indicative of finer textures.

3. Contrast of $H(y_q, y_r, d)$:

$$Contrast = \sum_{y_q=y_1}^{y_t} \sum_{y_q=y_1}^{y_t} \partial(y_q, y_r) H(y_q, y_r, d) \tag{11.26}$$

where $\partial(y_q, y_r)$ is a measure of intensity similarity and is defined by $\partial(y_q, y_r) = (y_q - y_r)^2$. Thus, the contrast characterizes variation in pixel intensity.

4. Inverse difference moment of $H(y_q, y_r, d)$, IDM_H:

$$IDM_H = \sum_{y_q=y_1}^{y_t} \sum_{y_q=y_1}^{y_t} \frac{H(y_q, y_r, d)}{1 + \partial(y_q, y_r)}, \tag{11.27}$$

where $\partial(y_q, y_r)$ is defined as before. The *IDM*H provides a measure of the local homogeneity among textures.

5. Correlation of $H(y\text{q}, y\text{r}, d)$:

$$Cor_H = \frac{1}{\sigma_{y_q} \sigma_{y_r}} \sum_{y_q=y_1}^{y_t} \sum_{y_q=y_1}^{y_t} (y_q - \mu_{y_q})(y_r - \mu_{y_r}) H(y_q, y_r, d), \tag{11.28}$$

where μ_{y_q}, μ_{y_r}, σ_{y_q}, and σ_{y_r} are the respective means and standard deviations of yq and yr. The correlation can also be expanded and written in terms of the marginal distributions of the second-order histogram, which are defined as

$$H_m(y_q, d) = \sum_{y_r=y_1}^{y_t} H(y_q, y_r, d),$$

$$H_m(y_r, d) = \sum_{y_q=y_1}^{y_t} H(y_q, y_r, d). \tag{11.29}$$

The correlation attribute is greater for similar elements of the second-order histogram.

6. Mean of $H(y_q, y_r, d)$: μ_{H_m}:

$$\mu_{H_m} = \sum_{y_q=y_1}^{y_t} y_q H_m(y_q, d). \tag{11.30}$$

The mean characterizes the nature of the gray-level distribution. Its value is typically small if the distribution is localized around $y_q = y_1$.

7. Deviation of $H_{\mathrm{m}}(y_{\mathrm{q}},d)$: σ_{H_m}:

$$\sigma_{H_m} = \sqrt{\sum_{y_q=y_1}^{y_t}\left[y_q - \sum_{y_r=y_1}^{y_t} y_r H_m(y_r,d)\right]^2 H_m(y_q,d)}. \tag{11.31}$$

The deviation indicates the amount of spread around the mean of the marginal distribution. The deviation is small if the histogram is densely clustered about the mean.

8. Entropy of $H_{\mathrm{d}}(y_{\mathrm{q}},d)$: $S_{H_d(y_s,d)}$:

$$S_{H_d(y_s,d)} = -\sum_{y_s=y_1}^{y_t} H_d(y_s,d)\log_{10}[H_d(y_s,d)]. \tag{11.32}$$

9. Angular second moment of $H_{\mathrm{d}}(y_{\mathrm{s}},d)$: $ASM_{H_d(y_s,d)}$:

$$ASM_{H_d(y_s,d)} = \sum_{y_s=y_1}^{y_t} [H_d(y_s,d)]^2. \tag{11.33}$$

10. Mean of $H_{\mathrm{d}}(y_{\mathrm{s}},d)$: $\mu_{H_d(y_s,d)}$:

$$\mu_{H_d(y_s,d)} = \sum_{y_s=y_1}^{y_t} y_s[H_d(y_s,d)]. \tag{11.34}$$

The features computed using the difference histogram, $H\mathrm{d}(y\mathrm{s}, d)$, have the same significance as those attributes determined by the second-order statistics.

Figure 11.12a,b shows two images from digitized X-ray mammograms, with, respectively, regions of benign lesion and malignant cancer of the breast. Their respective second-order histograms computed from the gray-level co-occurrence matrices are shown in Figure 11.12c,d. It can be seen that the second-order gray-level histogram statistics have better correlation with the classification of breast cancer images than the first-order histogram (16).

11.1.4 Relational Features

Relational features provide information about adjacencies, repetitive patterns, and geometrical relationships among regions of an object. Such features can also be extended to describe the geometrical relationships between objects in an image or a scene. The relational features can be described in the form of graphs or rules using a specific syntax or language. The geometric properties of images using linear quad-trees (where one parent node is divided into four children nodes) are described by Samet and Tamminen (17). Figure 11.13 shows a block (pixel)-based image representation of the letter "A" with its quad-tree representation. The quad-tree-based region descriptor can directly provide quantitative features such as perimeter, area, and Euler number by tracking the list of nodes belonging to the region of interest (17). The adjacent relationships of the elements of the quad-tree are translation invariant and can be treated as rotational invariant under specific conditions. The quad-tree-based region descriptors can also be used for object recognition and classification using the tree matching algorithms (18).

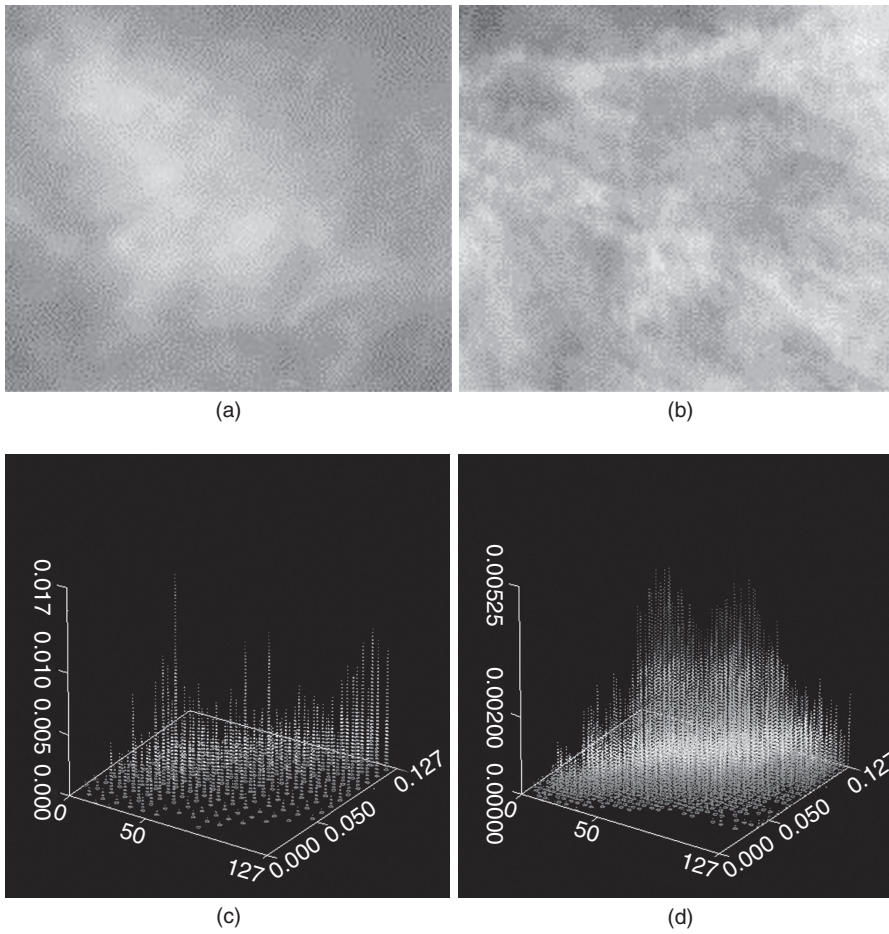


Figure 11.12 (a) A part of a digitized X-ray mammogram showing a region of benign lesion; (b) a part of a digitized X-ray mammogram showing a region of malignant cancer of the breast; (c) a second-order histograms of (a) computed from the gray-level co-occurrence matrices with a distance vector of [1, 1]; and (d) a second-order histogram of (b) computed from the gray-level co-occurrence matrices with a distance vector of [1, 1].

Tree and graph structures have been used effectively for knowledge representation and developing models for object recognition and classification. Figure 11.14 shows a tree structure representation of brain ventricles for applications in brain image segmentation and analysis (19, 20).

11.2. FEATURE SELECTION FOR CLASSIFICATION

A feature classification system can be viewed as a mapping from input features representing the given image to an output variable representing one of the categories or

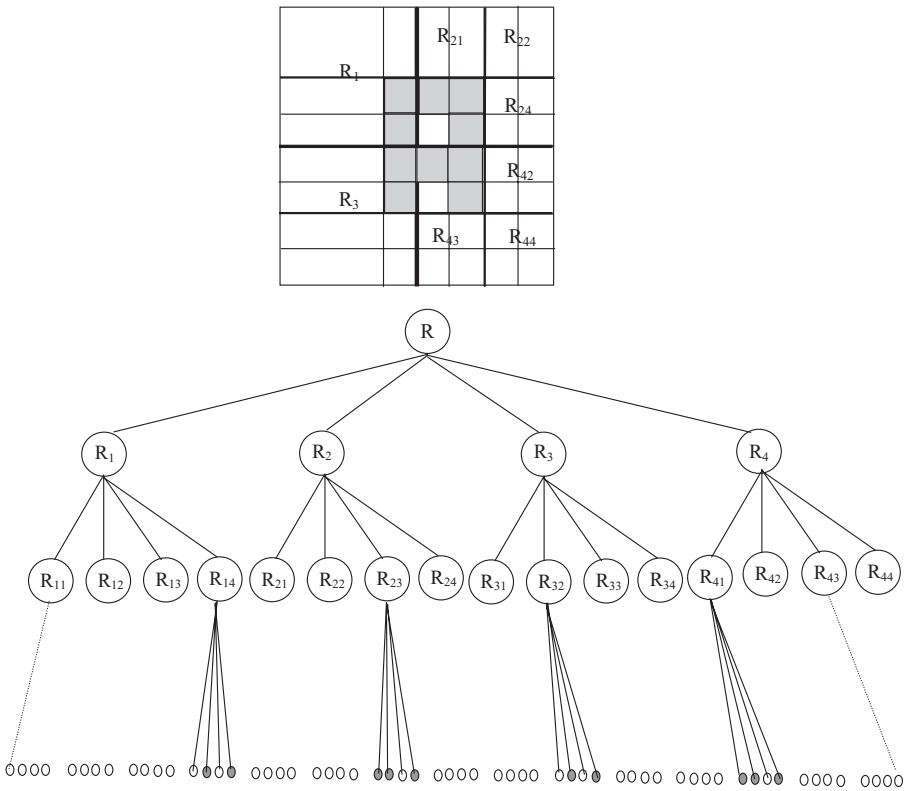


Figure 11.13 A block representation of an image with major quad partitions (top) and its quad-tree representation.

classes. For image analysis and classification task to interpret a medical image for diagnostic evaluation, a raw medical image may be first preprocessed with image enhancements and noise removal operations (described in Chapter 9). The preprocessed image is then analyzed for segmentation and extraction of features of interest. Usually, a large number of features with potential correlation to object classification can be computed. It is usually advantageous to select meaningful features that are well correlated to the task of object classification. Feature selection is the process of identifying the most effective subset of the correlated features for final feature representation and classification. Selection of correlated features leads to dimensionality reduction in the classification task, which improves the computational efficiency and classification performance, as only well-correlated features are used in the classifier.

The final set of features for classification can be determined through data correlation, clustering, and analysis algorithms to explore similarity patterns in the training data. Features that provide well separated clusters or clusters with minimum overlaps can be used for classification. The redundant and uncorrelated features are abandoned to reduce dimensionality for better classification. Data clustering methods such as agglomerative hierarchical clustering, *k*-means, fuzzy *c*-means, and adaptive fuzzy *c*-means clustering algorithms have been described in Chapter 10. Commonly

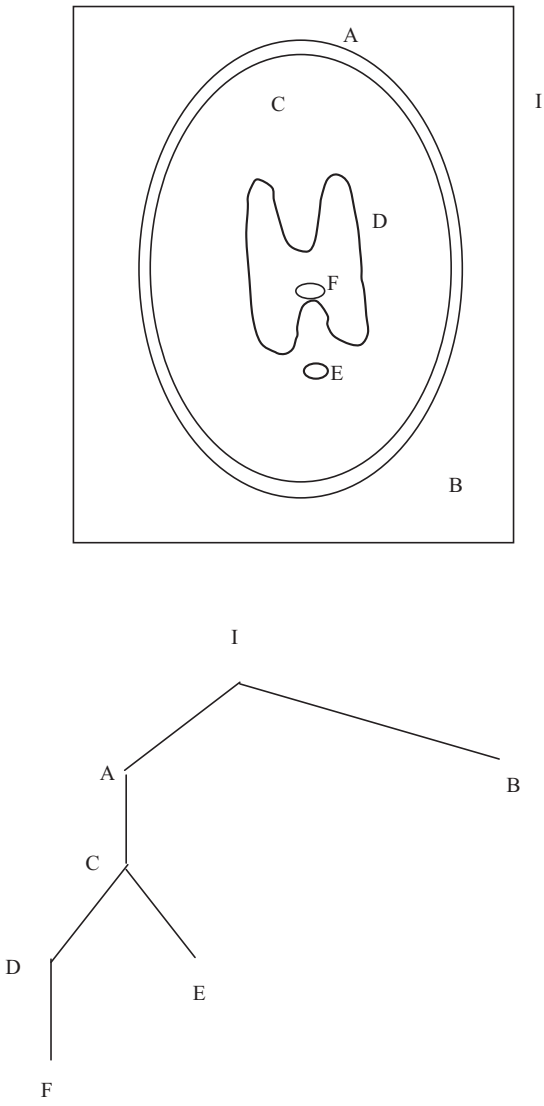


Figure 11.14 A 2-D brain ventricles and skull model (top) and region-based tree representation.

used approaches for feature selection and dimensionality reduction include linear discriminant analysis, principal component analysis (PCA), and genetic algorithm (GA)-based optimization methods. These methods are described below.

11.2.1 Linear Discriminant Analysis

Linear discriminant analysis methods are used to find a linear combination of features that can provide best possible separation among classes of data in the feature

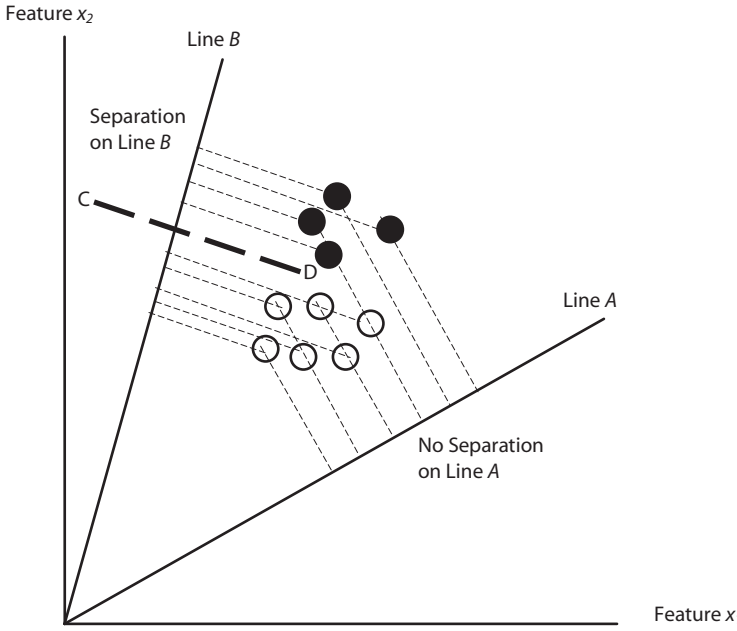


Figure 11.15 Projection of features or data points for two-class classification on two lines A and B in two-dimensional space. Open circles represent data points for class-1, while the black circles represent class-2 data points. Projections on line A are not separable, while projections on line B are well separated. The separating o decision boundary is shown by the dotted line CD.

space. A linear combination of features thus obtained reduces dimensionality for classification and provides better classification performance with a linear classifier. For example, let us assume that a d-dimensional set of features can be projected on an arbitrary line as shown in Figure 11.15. The projection of features onto the line can now be analyzed for separation among classes. Figure 11.15 shows 2-D feature space for two-class (e.g., benign and malignant) classification. It can be seen that the projections on line A are not separable while they become well separated if the line A is rotated to position B.

Let us assume that there is n number of d-dimensional features vectors, $\mathbf{x}$, in a d-dimensional space ($x_1, x_2, \ldots x_d$), in the training set that is projected onto a line in the direction $\mathbf{w}$ as an n-dimensional vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ as

$$y = \mathbf{w}^T \mathbf{x}. \tag{11.35}$$

It can be noted that each y_i is the projection of each $\mathbf{x_i}$ onto the line in the direction $\mathbf{w}$, if $\|\mathbf{w}\| = 1$.

Considering first the case of two classes (C_i; $i = 1, 2$), a line in the direction $\mathbf{w}$ has to be found such that the projected points from each class on the line must form two clusters that are well separated with respect to their mean and variances. Let μ_i be the mean of two d-dimensional samples (features or data points in two classes with n_i; $i = 1, 2$ number of samples in each class) as

$$\boldsymbol{\mu}_i = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathbf{C}_i} \mathbf{x} \tag{11.36}$$

The projected points on the line may be represented with respective mean values $\tilde{\mu}_i$ as

$$\tilde{\mu}_i = \frac{1}{n_i} \sum_{y \in \mathbf{C}_i} y = \frac{1}{n_i} \sum_{\mathbf{x} \in \mathbf{C}_i} \mathbf{w}^T \mathbf{x} = \mathbf{w}^T \boldsymbol{\mu}_\mathbf{i}. \tag{11.37}$$

Thus, the separation between the two classes with respect to their mean values can be expressed as

$$|\mu_1 - \mu_2| = |\mathbf{w}^T (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)|. \tag{11.38}$$

Let us define scatter $\tilde{s}_i^2$ of projected points within a class as

$$\tilde{s}_i^2 = \sum_{y \in C_i} (y - \tilde{\mu}_i^2) \tag{11.39}$$

The Fisher linear discriminant is then defined as the linear function $y = \mathbf{w}^T\mathbf{x}$ for which a criterion function $J(\mathbf{w})$ based on the separation of mean values and within-class scatter $(\tilde{s}_1^2 + \tilde{s}_2^2)$ is maximized. The criterion function $J(\mathbf{w})$ can be defined as (21)

$$J(\mathbf{w}) = \frac{|\tilde{\mu}_1 - \tilde{\mu}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2}. \tag{11.40}$$

Considering within-class scatter, a within-class matrix S_w can be defined as

$$S_w = \sum_{x \in C_1} (\mathbf{x} - \boldsymbol{\mu}_1)(\mathbf{x} - \boldsymbol{\mu}_1)^T + \sum_{x \in C_2} (\mathbf{x} - \boldsymbol{\mu}_2)(\mathbf{x} - \boldsymbol{\mu}_2)^T. \tag{11.41}$$

A between-class scatter matrix S_B based on the separation of mean values can be defined as

$$S_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T. \tag{11.42}$$

A generalized expression for two-class Fisher linear discriminant can be expressed in terms of S_w and S_B as

$$J(\mathbf{w}) = \frac{\mathbf{w}^T S_B \mathbf{w}}{\mathbf{w}^T S_W \mathbf{w}}. \tag{11.43}$$

It can be shown that a vector $\mathbf{w}$ that maximizes the criterion function $J(\mathbf{w})$ can be solved using the eigenvalues λ as (assuming S_w to be nonsingular):

$$S_W^{-1} S_B \mathbf{w} = \lambda \mathbf{w}. \tag{11.44}$$

The above representation leads to a general solution as

$$\mathbf{w} = S_W^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2). \tag{11.45}$$

It should be noted that the above solution leads to many-to-one mapping that may not always minimize the classification error. However, it can be shown that if the conditional probability density functions for both classes are considered to be

multivariate normal with equal covariance matrix Σ, the optimal linear discriminant function with a decision boundary for classification can be expressed as

$$\mathbf{w}^T\mathbf{x} + w_0 = 0 \tag{11.46}$$

where $\mathbf{w} = \Sigma^{-1}(\mu_1 - \mu_2)$ and w_0 is a contant involving prior probabilities.

The above described two-class fisher linear discriminant analysis can be extended and generalized for multiclass discriminant analysis to provide projection of a d-dimensional space to $(C - 1)$ dimensional space where C is the total number of classes with

$$y_i = \mathbf{w}_i^T\mathbf{x} \quad i = 1,2,\ldots,(C-1). \tag{11.47}$$

The between-class and within-class scatter matrices can now be defined as

$$S_B = \sum_{i=1}^{C} n_i(\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T$$

$$S_W = \sum_{i=1}^{C}\sum_{\mathbf{x}\in C_i} (\mathbf{x} - \boldsymbol{\mu}_i)(\mathbf{x} - \boldsymbol{\mu}_i)^T. \tag{11.48}$$

For multiple classes, the mapping from d-dimensional space to $(C - 1)$-dimensional space involves a transformation matrix $\mathbf{W}$ such that the criterion function $J(\mathbf{W})$ is expressed as

$$J(\mathbf{W}) = \frac{\left|\mathbf{W}^T S_B \mathbf{W}\right|}{\left|\mathbf{W}^T S_W \mathbf{W}\right|}. \tag{11.49}$$

11.2.2 PCA

Principal component analysis is an efficient method to reduce the dimensionality of a data set that consists of a large number of interrelated variables (22). The goal here is to map vectors $\mathbf{x}$ in a d-dimensional space (x_1, x_2, … x_d) onto vectors $\mathbf{z}$ in an M-dimensional space (z_1, z_2, … z_M) where $M < d$. Without loss of generality, we express vector $\mathbf{x}$ as a linear combination of a set of d orthonormal vectors $\mathbf{u}_i$

$$\mathbf{x} = \sum_{i=1}^{d} x_i \mathbf{u}_i \tag{11.50}$$

where the vectors $\mathbf{u}_i$ satisfy the orthonormality relation

$$\mathbf{u}_i^T \mathbf{u}_j = \delta_{ij} \tag{11.51}$$

Therefore, the coefficient in Equation 11.50 can be expressed as

$$x_i = \mathbf{u}_i^T \mathbf{x}. \tag{11.52}$$

Let us suppose that only a subset of $M < d$ of the basis vectors $\mathbf{u}_i$ are to be retained, so that only M coefficients of $\mathbf{x}_i$ are used. Let us assume a set of new basis vectors, $\mathbf{v}_i$, which meet the orthonormality requirement. As above, only M coefficients from $\mathbf{x}_i$ are used and the remain ing coefficients are replaced by b_i. The vector x can now be approximated as

$$\tilde{\mathbf{x}} = \sum_{i=1}^{M} x_i \mathbf{v}_i + \sum_{i=M+1}^{d} b_i \mathbf{v}_i \tag{11.53}$$

The next step is to minimize the sum of squares of errors over the entire data set as

$$E_M = \frac{1}{2} \sum_{i=M+1}^{d} \mathbf{v}_i^T A \mathbf{v}_i \tag{11.54}$$

where A is the covariance matrix for vector $\mathbf{x}$.

Now the error E_M should be minimized with respect to the choice of basis vectors $\mathbf{v}_i$. A minimum value is obtained when the basis vectors satisfy the condition:

$$A\mathbf{v}_i = \beta_i \mathbf{v}_i \tag{11.55}$$

where $\mathbf{v}_i$ with $i = M + 1 \ldots d$ represents the eigenvectors of the covariance matrix.

It can be shown that

$$E_M = \frac{1}{2} \sum_{i=M+1}^{d} \beta_i. \tag{11.56}$$

Therefore, the minimum error is achieved by rejecting the (d-M) smallest eigenvalues and their corresponding eigenvectors. As the first M largest eigenvalues are retained, each of the associated eigenvectors $\mathbf{v}_i$ is called a principal component.

For computer implementation, singular value decomposition (SVD) algorithm can be employed to calculate the eigenvalues and its corresponding eigenvectors (21, 22). In addition, several algorithms are available for implementation of the PCA method (22). However, the PCA method may not provide optimal selection of features for the data with sparse distribution and noise. GA-based optimization methods have been used to find more practical solutions to feature selection that can provide the best classification results.

11.2.3 GA-Based Optimization

A GA is a robust optimization and search method based on the natural selection principles. GA provide improved performance by exploiting past information and promoting competition for survival. A fundamental feature of GA is that they can adapt to specific problem parameters. These parameters are typically encoded as binary strings that are associated with a measure of goodness or fitness value. As in natural evolution, GA favors the survival of the fittest through selection and recombination. Through the process of reproduction, individual strings are copied according to their degree of fitness. In crossover operations, strings are probabilistically mated by swapping all variables located after a randomly chosen position. Mutation is a secondary genetic operator that randomly changes the value of a string position to introduce variation to the population and recover lost genetic information (23).

Genetic algorithms maintain a population of structures that are potential solutions to an objective function. Let us assume that features are encoded into binary

strings that can be represented as $A = a_1, a_2, \ldots, a_L$ where L is the specified string length or the number of representative bits. A simple GA operates on these strings according to the following iterative procedure:

1. Initialize a population of binary strings.
2. Evaluate the strings in the population.
3. Select candidate solutions for the next population and apply mutation and crossover operators to the parent strings.
4. Allocate space for new strings by removing members from the population.
5. Evaluate the new strings and add them to the population.
6. Repeat steps 3–5 until the stopping criterion is satisfied.

The structure of the GA is based on the encoding mechanism used to represent the variables in the optimization problem. The candidate solutions may encode any number of variable types, including continuous, discrete, and boolean variables. Although different string coding methods may be used (23), a simple binary encoding mechanism provides quite useful results. The allele of a gene in the chromosome indicates whether or not a feature is significant in the problem. The objective function evaluates each chromosome in a population to provide a measure of the fitness of a given string. Since the value of the objective function can vary widely between problems, a fitness function is used to normalize the objective function within the range of 0 to 1. The selection scheme uses this normalized value, or fitness, to evaluate a string.

One of the most basic reproduction techniques is proportionate selection, which is carried out by the roulette wheel selection scheme. In roulette wheel selection, each chromosome is given a segment of a roulette wheel whose size is proportionate to the chromosome's fitness. A chromosome is reproduced if a randomly generated number falls in the chromosome's corresponding roulette wheel slot. Since more fit chromosomes are allocated larger wheel portions, they are more likely to generate offspring after a spin of the wheel. The process is repeated until the population for the next generation is completely filled. However, due to sampling errors, the population must be very large in order for the actual number of offspring produced for an individual chromosome to approach the expected value for that chromosome.

In proportionate selection, a string is reproduced according to how its fitness compares with the population average, in other words, as $f_i / \bar{f}$, where f_i is the fitness of the string and $\bar{f}$ is the average fitness of the population. This proportionate expression is also known as the selective pressure on an individual. The mechanics of proportionate selection can be expressed as: A_i receives more than one offspring on average if $f_i > \bar{f}$; otherwise, A_i receives less than one offspring on average. Since the result of applying the proportionate fitness expression will always be a fraction, this value represents the expected number of offspring allocated to each string, not the actual number.

Once the parent population is selected through reproduction, the offspring population is created after application of genetic operators. The purpose of

recombination, also referred to as crossover, is to discover new regions of the search space, rather than relying on the same population of strings. In recombination, strings are randomly paired and selected for crossover. If the crossover probability condition is satisfied, then a crossover point along the length of the string pair is randomly chosen. The offspring are generated by exchanging the portion of the parent strings beyond the crossover position. For a string of length l, the $l - 1$ possible crossover positions may be chosen with equal probability.

Mutation is a secondary genetic operator that preserves the random nature of the search process and regenerates fit strings that may have been destroyed or lost during crossover or reproduction. The mutation rate controls the probability that a bit value will be changed. If the mutation probability condition is exceeded, then the selected bit is inverted.

An example of a complete cycle for the simple GA is illustrated in Table 11.1. The initial population contains four strings composed of 10 bits. The objective function determines the number of 1s in a chromosome, and the fitness function normalizes the value to lie in the range of 0 to 1.

The proportional selection method allocates 0, 1, 1, and 2 offspring to the initial offspring in their respective order. After selection, the offspring are randomly paired for crossover so that strings 1 and 3 and strings 2 and 4 are mated. However, since the crossover rate is 0.5, only strings 1 and 3 are selected for crossover, and the other strings are left intact. The pair of chromosomes then exchanges their genetic material after the fifth bit position, which is the randomly selected crossover point. The final step in the cycle is mutation. Since the mutation rate is selected to be 0.05, only two bits out of the 40 present in the population are mutated. The second

TABLE 11.1 A Sample Generational Cycle of the Simple Genetic Algorithm

	Chromosome	Fitness value	Average fitness
Population P_1 (initial population)	0001000010	0.2	
	0110011001	0.5	0.50
	1010100110	0.5	
	1110111011	0.8	
Population P_2 (after selection)	0110011001	0.5	
	1010100110	0.5	0.65
	1110111011	0.8	
	1110111011	0.8	
Population P_3 (after crossover)	01100|11011	0.6	
	1010100110	0.5	0.65
	11101|11001	0.7	
	1110111011	0.8	
Population P_4 (after mutation)	0110011011	0.6	
	1110100110	0.6	0.70
	1110111001	0.7	
	1111111011	0.9	

bit of string 2 and the fourth bit of string 4 are randomly selected for mutation. As can be seen from the Table 11.1, the average fitness of Population P_4 is significantly better than the initial fitness after only one generational cycle.

The average fitness value in the initial stages of a GA is typically low. Thus, during the first few generations the proportionate selection scheme may assign a large number of copies to a few strings with relatively superior fitness, known as super individuals. These strings will eventually dominate the population and cause the GA to converge prematurely. The proportionate selection procedure also suffers from decreasing selective pressure during the last generations when the average fitness value is high. Scaling techniques and ranking selection can help alleviate the problems of inconsistent selective pressure and domination by superior individuals.

Ranking selection techniques assign offspring to individuals by qualitatively comparing levels of fitness. The population is sorted according to their fitness values and allotted offspring based on their rank. In ranking selection, subsequent populations are not influenced by the balance of the current fitness distributions so that selective pressure is uniform. Each cycle of the simple GA produces a completely new population of offspring from the previous generation, known as generational replacement. Thus, the simple GA is naturally slower in manipulating useful areas of the search space for a large population. Steady-state replacement is an alternative method that typically replaces one or more of the worst members of the population each generation. Steady-state replacement can be combined with an elitist strategy, which retains the best strings in the population (23).

Genetic algorithms are global optimization techniques that are highly suited to searching in nonlinear, multidimensional problem spaces, and used for medical image analysis for computer-aided diagnosis such as analysis of mammographic microcalcification images for diagnosis of breast cancer (23–25). An example of such analysis is described in the last section of this chapter.

11.3. FEATURE AND IMAGE CLASSIFICATION

Features selected for image representation are classified for object recognition and characterization. For example, features representing mammographic microcalcifications are analyzed and classified for the detection of breast cancer. In the analysis of medical images, features and measurements can also be used for region segmentation to extract meaningful structures, which are then interpreted using knowledge-based models and classification methods.

11.3.1 Statistical Classification Methods

Statistical classification methods are broadly defined into two categories: unsupervised and supervised. The unsupervised methods cluster the data based on their separation in the feature space. Data clustering methods such as *k*-means and fuzzy clustering methods are commonly used for unsupervised classification. Probabilistic

methods such as nearest neighbor classifier and Bayesian classifier can be used for supervised classification.

11.3.1.1 Nearest Neighbor Classifier A popular statistical method for classification is the nearest neighbor classifier, which assigns a data point to the nearest class model in the feature space. It is apparent that the nearest neighbor classifier is a supervised method as it uses labeled clusters of training samples in the feature space as models of classes. Let us assume that there are C number of classes represented by c_j; $j = 1, 2, \ldots, C$. An unknown feature vector $\mathbf{f}$ is to be assigned to the class that is closest to the class model developed from clustering the labeled feature vectors during the training. A distance measure $D_j(\mathbf{f})$ is defined by the Euclidean distance in the feature space as

$$D_j(\mathbf{f}) = \|\mathbf{f} - \mathbf{u}_j\| \tag{11.57}$$

where $\mathbf{u}_j = \frac{1}{N_j} \sum_{f_j \in c_j} \mathbf{f}_j \;\; j = 1, 2, \ldots C$ is the mean of the feature vectors for the class c_j and N_j is the total number of feature vectors in the class c_j.

The unknown feature vector is assigned to the class c_i if

$$D_j(\mathbf{f}) = \min_{i=1}^{C}[D_i(\mathbf{f})]. \tag{11.58}$$

11.3.1.2 Bayesian Classifier A probabilistic approach can be applied to the task of classification to incorporate a priori knowledge to improve performance. Bayesian and maximum likelihood methods have been widely used in object recognition and classification for different applications. A maximum likelihood method for pixel classification for brain image segmentation is presented in Chapter 10.

Let us assume that the probability of a feature vector $\mathbf{f}$ belonging to the class c_i is denoted by $p(c_i/\mathbf{f})$. Let an average risk of wrong classification for assigning the feature vector to the class c_j be expressed by $r_j(\mathbf{f})$ as

$$r_j(\mathbf{f}) = \sum_{k=1}^{C} Z_{kj} p(c_k / \mathbf{f}) \tag{11.59}$$

where Z_{kj} is the penalty of classifying a feature vector to the class c_j when it belongs to the class c_k.

It can be shown that

$$r_j(\mathbf{f}) = \sum_{k=1}^{C} Z_{kj} p(\mathbf{f} / c_k) p(c_k) \tag{11.60}$$

where $p(c_k)$ is the probability of occurrence of class c_k.

A Bayes classifier assigns an unknown feature vector to the class c_i if

$$r_j(\mathbf{f}) < r_i(\mathbf{f})$$

or

$$\sum_{k=1}^{C} Z_{kj} p(\mathbf{f} / c_k) p(c_k) < \sum_{q=1}^{C} Z_{qi} p(\mathbf{f} / c_q) p(c_q) \quad \text{for } i = 1, 2, \ldots, C. \tag{11.61}$$

11.3.2 Rule-Based Systems

The decision making process of classification can be implemented using a rule-based system. A rule-based system analyzes the feature vector using multiple sets of rules that are designed to check specific conditions in the database of feature vectors to initiate an action. The rules are comprised of two parts: condition premises and actions. They are based on expert knowledge to infer the action if the conditions are satisfied. The action part of the rule may change the database or label a feature vector depending upon the state of the analysis. Usually, a rule-based system has three sets of rules: supervisory or strategy rules, focus of attention rules, and knowledge rules. The supervisory or strategy rules guide the analysis process and provide the control actions such as starting and stopping the analysis. The strategy rules also decide the rules to be scheduled and tested during the analysis process. The focus-of-attention rules provide specific features into analysis by accessing and extracting the required information or features from the database. These rules bring the information from the input database into the activity center where the execution of knowledge rules is scheduled. The knowledge rules analyze the information with respect to the required conditions and implement an action causing changes in the output database. These changes lead to the classification and labeling of the feature vectors for object recognition. Figure 11.16 presents a schematic diagram of a general-purpose rule-based system for image analysis.

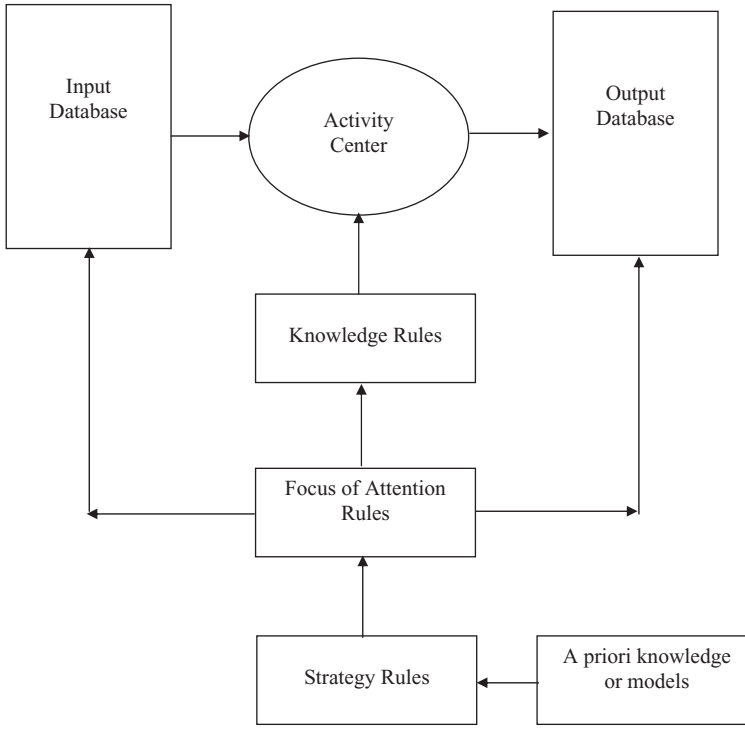


Figure 11.16 A schematic diagram of a rule-based system for image analysis.

Rule-based systems have been used for image segmentation (26) for biomedical applications (4, 27, 28). The examples of strategy, focus of attention, and knowledge rules for segmentation of CT images of the human chest are as follows (27). The symbols used in the analysis are shown in the capital case.

Strategy Rule SR1:
  If
    NONE REGION is ACTIVE
    NONE REGION is ANALYZED
  Then
    ACTIVATE FOCUS in SPINAL_CORD AREA

Strategy Rule SR2:
  If
    ANALYZED REGION is in SPINAL_CORD AREA
    ALL REGIONS in SPINAL_CORD AREA are NOT ANALYZED
  Then
    ACTIVATE FOCUS in SPINAL_CORD AREA

Strategy Rule SR3:
  If
    ALL REGIONS in SPINAL_CORD AREA are ANALYZED
    ALL REGION in LEFT_LUNG AREA are NOT ANALYZED
  Then
    ACTIVATE FOCUS in LEFT_LUNG AREA

Focus of Attention Rule FR1:
  If
    REGION-X is in FOCUS AREA
    REGION-X is LARGEST
    REGION-X is NOT ANALYZED
  Then
    ACTIVATE REGION-X

Focus of Attention Rule FR2:
  If
    REGION-X is in ACTIVE
    MODEL is NOT ACTIVE
  Then
    ACTIVATE KNOWLEDGE_MERGE rules

Knowledge Rule: Merge_Region_KR1
  If
    REGION-1 is SMALL
    REGION-1 has HIGH ADJACENCY with REGION-2
    DIFFERENCE between AVERAGE VALUE of REGION-1 and REGION-2 is LOW or VERY LOW
    REGION-2 is LARGE or VERY LARGE
  Then
    MERGE REGION-1 in REGION-2
    PUT_STATUS ANALYZED in REGION-1 and REGION-2

It should be noted that the sequence of execution of rules directly affects the sequential changes in the database. Thus, a different sequence of execution of rules may produce very different results. To avoid this problem, all rules in the system must be statistically independent. This condition is usually unrealistic because it is difficult to translate the expert-domain knowledge into statistically independent rules. Condition probability-based Bayes models and fuzzy inference may help the inference by execution of rules for better performance and robustness. Nevertheless, the formation of knowledge rules along with the probabilistic models is usually a very involved task.

11.3.3 Neural Network Classifiers

Several artificial neural network paradigms have been used for feature classification for object recognition and image interpretation. The paradigms include backpropagation, radial basis function, associative memories, and self-organizing feature maps. The backpropagation neural network (BPNN) and the radial basis function neural network (RBFNN) are described in Chapter 10 for pixel classification for image segmentation (28). Recently, fuzzy system-based approaches have been applied in artificial neural networks for better classification and generalization performance (29–32). A neuro-fuzzy pattern classifier is described here which has also been used for medical image analysis (33).

11.3.3.1 Neuro-Fuzzy Pattern Classification Any layer in a feedforward network such as BPNN performs partitioning of the multidimensional feature space into a specific number of subspaces. These subspaces are always convex and their number can be estimated (29). A computational neural element is shown in Figure 11.17 with an input vector $\boldsymbol{X}$, nonlinearity function $f(\varphi)$, and the final output vector $\boldsymbol{Y}$. The input synapses represented by $\boldsymbol{X}$ are linearly combined through connection weights w_i to provide the postsynaptic signal φ as:

$$\varphi = \sum_{i=1}^{d} x_i w_i + w_0 \tag{11.62}$$

where d is the total number of input synapses or features.

For $\varphi = 0$ (or any other constant), Equation 11.62 represents a $(d\text{-}1)$-dimensional hyperplane H in the d-dimensional input space separating two regions defined by the connection weights w_i (28, 29):

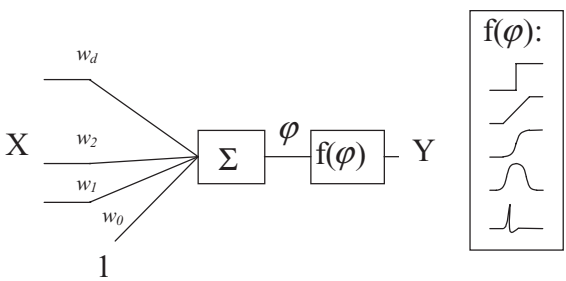


Figure 11.17 A computational neuron model with linear synapses.

$$(H: \quad \varphi = 0) \Rightarrow \left(H: \quad \sum_{i=1}^{d} x_i w_i + w_0 = 0 \right). \tag{11.63}$$

Each network layer comprises many hyperplanes that intersect to create a finite number of the aforementioned convex subspaces. Therefore, there is a direct relationship between the connection weight values and the obtained d-dimensional convex subspaces. The process of network training can be assumed as an attempt at finding an optimal dichotomy of the input space into these convex regions. Moreover, it can also be said that finding the optimal dichotomy of input space into convex subspaces is equivalent to network training. The classes are separated in the feature space by computing the homogeneous nonoverlapping closed convex subsets. The classification is obtained by placing separating hyperplanes between neighboring subsets representing classes. This completes the design of a network hyperplane layer. Since the hyperplane separation of the subsets results in the creation of the homogenous convex regions, the consecutive network layer is to determine in which region an unknown input pattern belongs.

In the approach presented by Grohman and Dhawan (33), a fuzzy membership function M_f is devised for each convex subset ($f = 1, 2, \ldots, K$). The classification decision is made by the output layer based on the "winner-take-all" principle. The resulting category C is the convex set category with the highest value of membership function for the input pattern. Thus, the neuro-fuzzy pattern classifier (NFPC) design method includes three stages: convex set creation, hyperplane placement (hyperplane layer creation), and construction of the fuzzy membership function for each convex set (generation of the fuzzy membership function layer). The architecture of the NFPC is shown in Figure 11.18.

There are two requirements for computing the convex sets: they have to be homogeneous and nonoverlapping. To satisfy the first condition, one needs to devise a method of finding one-category points within another category's hull. Thus, two

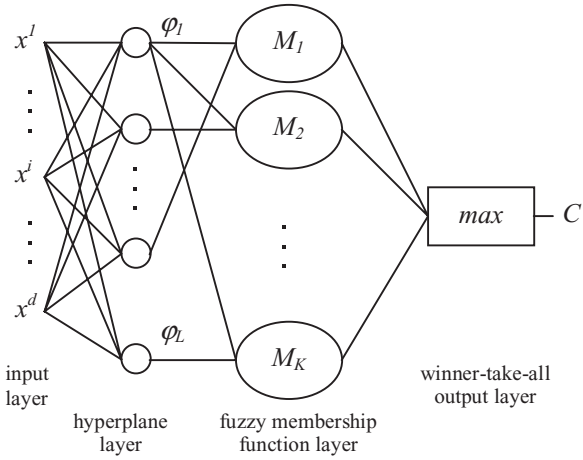


Figure 11.18 The architecture of the neuro-fuzzy pattern classifier.

problems can be defined: (1) how to find whether the point P lies inside of a convex hull (CH) of points; and (2) how to find out if two CH of points are overlapping. The second problem is more difficult to examine because hulls can be overlapping over a common (empty) space that contains no points from either category. When training samples are not completely noise-free, a compromise between computational efficiency and accuracy may be desirable. The following algorithm A1 addresses the first problem using the separation theorem, which states that for two closed non-overlapping convex sets S_1 and S_2 there always exists a hyperplane that separates the two sets.

Algorithm A1: Checking if the point P lies inside of a CH:

1. Consider P as origin.
2. Normalize points of CH (the vectors $V = (v_1, v_2, \ldots, v_n)$ from the origin).
3. Find min. and max. vector coordinates in each dimension.
4. Find set E of all vectors V that have at least one extreme coordinate.
5. Compute mean and use it as projection vector ϕ:

$$\phi = (\ \overline{v_i}\ \mid \forall\ \ v_i \in \mathrm{E}).$$

6. Set a maximum number of allowed iterations (usually = $2n$).
7. Find a set $U = (u_1, u_2, \ldots, u_m)$ (where $m \leq n$) of all points in CH that have negative projection on ϕ.
8. If U is empty (P is outside of CH), exit, or else proceed to step 9.
9. Compute coefficient ψ as

$$\psi = \phi^T \overline{U}$$
$$\overline{U} = \frac{1}{m}\sum_{i=1}^{m} u_i.$$

10. Calculate correction vector $\delta\phi$ by computing all of its k-dimensional components $\delta\phi^k$:

$$\left(\begin{array}{l} \overline{U}^k \neq 0 \Rightarrow \delta\phi^k = \dfrac{\psi}{\overline{U}^k d} \\ \overline{U}^k = 0 \Rightarrow \delta\phi^k = \dfrac{\psi}{d} \end{array}\right), \quad k = 1,2,\ldots,d$$

where d is the dimension of the feature space.

11. Update ϕ: $\phi = \phi - \eta \cdot \delta\phi$, where $\eta > 1$ is a training parameter.
12. If iteration limit exceeded exit (assume P inside of CH), otherwise go to 7.

The value of the training parameter η should be close to 1, so even the points lying outside but close to the hull can be found. Heuristically it has been found that the values of α should fall in the range $1.0001 < \eta < 1.01$. They are, however, dependent on the precision of the training data and should be adjusted accordingly. The main objective of the algorithm is to find the hyperplane (defined by its

orthogonal vector ϕ) separating P and CH. If such a hyperplane is found within a certain amount of iterations, the point is definitely outside of CH. If the hyperplane has not been found, it is assumed that P is inside. Now, having found the solution to problem 1, the convex subsets can be created using the algorithm A2 as follows.

Algorithm A2: Convex subset creation:

1. Select one category from the training set and consider all data points in the category. This is a positive set of samples. The training points from all the remaining categories constitute a negative set. Both sets are in d-dimensional linear space L. Mark all positive points as "not yet taken" and order them in a specific way. For example, choose an arbitrary starting point in the input space and order all positive points according to their Euclidean distance from that point. Use an index array Λ to store the order.
2. Construct the convex subsets:

   Initialize current subset S by assigning to it the first point in Λ. Loop over ordered positive category points (in Λ) until there are no more points remaining. Consider only points that have not yet been "taken":

   a. Add the current point P to the subset S.

   b. Loop over points from negative category. Consider only negative points that are closer than P to the middle of the current subset. Using A1, look for at least one negative point inside of conv S. If there is one, disregard the latest addition to S. Otherwise mark the current point P as "taken."

   c. Update Λ. Reorder the "not yet taken" positive category points according to their distance from the mean of points in the current subset.
3. If all points in the category have been assigned to a subset, proceed to step 4, otherwise go back to step 2 and create the next convex subset. The starting point is the first "not yet taken" point in the list.
4. Check if all categories have been divided into convex subsets. If not, go back to step 1 and create subsets of the next category.

In step 2b, it is not always necessary to use algorithm A1 to check the presence of every single negative point within the current convex subset. Once a separating hyperplane is found for one negative point it should be used to eliminate all other negative points that lie on the opposite side of the hyperplane than the convex subset, from the checklist. Thus both presented algorithms should be used together in order to save computations.

Once the nonoverlapping convex subsets are found, hyperplanes are paced to separate two neighboring subsets. The NPFC hyperplane layer comprises a set of all hyperplanes needed to fully separate all convex subsets from different categories. The hyperplanes define the convex regions constructed from the training samples. The fuzzy membership function M_f to reflect the true shape of the convex subset f ($f = 1, 2, \ldots, K$) can be computed as:

$$M_f(\mathbf{x}) = \sqrt[L_f]{\prod_{i=1}^{L_f} \theta_i}, \quad \theta_i = \frac{1}{\left(1 + e^{\lambda_{if} \varphi_i \mathbf{x}}\right)} \tag{11.64}$$

where L_f is the number of separating hyperplanes for the subset f; φ_i is the ith separating hyperplane function for the subset, in the vector form; $\mathbf{x}$ is the input vector in the augmented form; and λ_{if} is the steepness (scaling) coefficient for the ith hyperplane in the subset f.

The value of λ_{if} depends on the depth of convex subset f, as projected onto the separating hyperplane H_i (defined by ϕ_i):

$$\lambda_{if} = \frac{-\log\left(\frac{1-\chi}{\chi}\right)}{\mu_{if}}, \quad \mu_{if} = \frac{1}{n}\sum_{j=1}^{n} \varphi_i \mathbf{x}_j \tag{11.65}$$

where n is the number of training points in the convex subset f; ϕ_i is the ith hyperplane equation in the vector form; μ_{if} is the depth of the convex subset f, as projected onto the ith hyperplane; $\mathbf{x}_j$ is augmented coordinate vector of the jth point in the subset; and χ is the center value of the membership function.

The structure of the fuzzy membership function described above is shown in Figure 11.19. Scaling and multiplication stages are represented by Equations 11.65 and 11.64, respectively.

The output O of the classifier is the category C of the convex set fuzzy membership function M_i that attains the highest value for the specified input pattern x, that is:

$$\left(O = C \;\middle|\; \begin{array}{c} \forall \\ 1 \le f \le K \, M_f(x) < M_i(x) \;,\; M_i \in C \\ f \ne i \end{array} \right) \tag{11.66}$$

where K is the number of convex sets obtained during training (number of fuzzy function neurons in the fuzzy membership function layer), M_i is the highest fuzzy membership function value for the input x, and C is the category of convex subset used to construct membership function M_i.

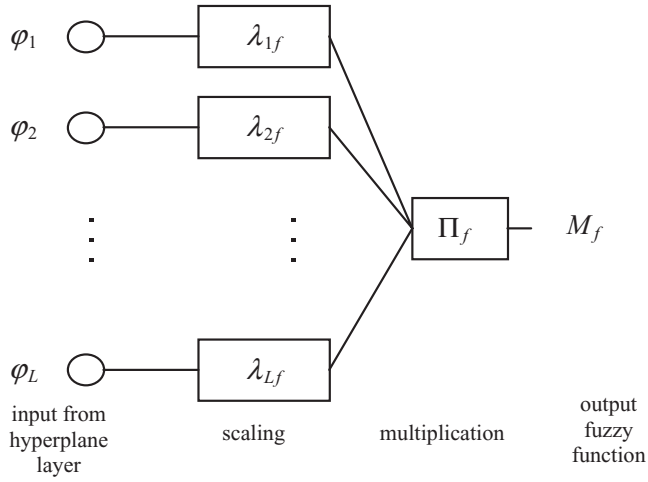


Figure 11.19 The structure of the fuzzy membership function.

In other words, the output is based on the winner-take-all principle, with the convex set category corresponding to M_i determining the output. A decision surface for each category can be determined by the fuzzy union of all of the fuzzy membership functions for the convex subsets belonging to the category. Thus the decision surface for a particular category can be defined as:

$$\left(M_{category}(x) = \max(M_i(x)) \;\middle|\; \begin{array}{c} \forall \\ i, M_i \in category \end{array} \right) \tag{11.67}$$

where $M_{category}(x)$ is the decision surface for the category and M_i is the fuzzy membership function for the convex cluster i.

Figure 11.20 shows an example of classification with two categories: dark and white dots. The hyperplanes are placed to separate two convex subsets of the black category from the convex subset of the white category. Figure 11.21a,b shows the fuzzy membership functions $M_1(x)$ and $M_2(x)$ for black category subsets. Figure 11.22 illustrates the membership function $M_3(x)$ for the white category. The resulting decision surface $M_{black}(x)$ for the black category is shown in Figure 11.23.

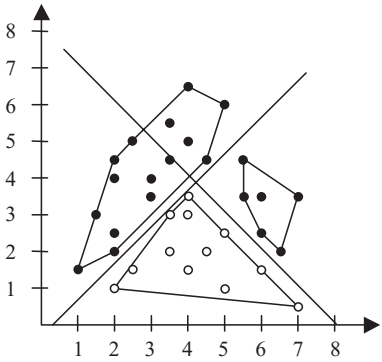


Figure 11.20 Convex set-based separation of two categories.

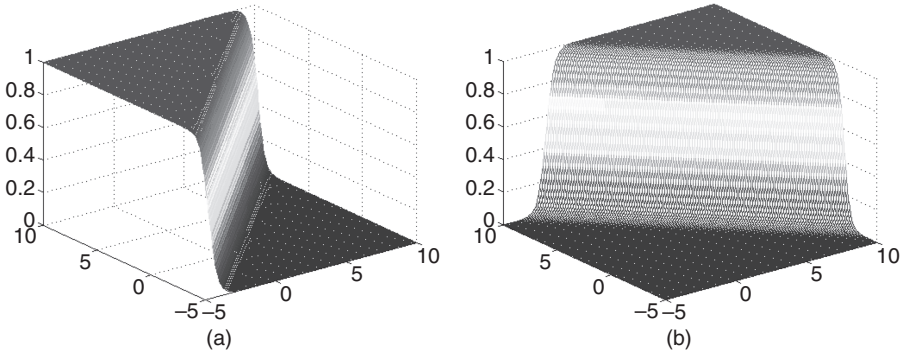


Figure 11.21 (a) Fuzzy membership function $M_1(x)$ for the subset #1 of the black category. (b) Fuzzy membership function $M_2(x)$ for the subset #2 of the black category.

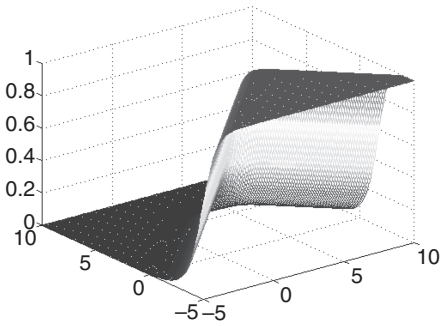


Figure 11.22 Fuzzy membership function $M_3(x)$ (decision surface) for the white category membership.

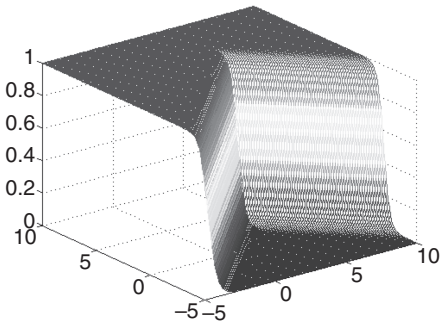


Figure 11.23 Resulting decision surface $M_{black}(x)$ for the black category membership function.

11.3.4 Support Vector Machine for Classification

As described above, the winner-take-all strategy can be effectively used for multi-classification. A set of prototypes such as fuzzy membership functions in the above described approach can be defined. A scoring function $\phi : \chi \times \mathrm{M} \rightarrow \Re$ is also defined measuring the similarity of an input feature vector, an element in χ with M_i prototypes defined in space M. Thus, a most similar prototype is selected to assign the respective class C, from a set $\{\gamma\}$, to the feature vector for classification. A multi-prototype approach for multiclass classification using the winner-take-all method can thus be expressed as (34–37)

$$H(\mathbf{x}) = C\left(\arg\max_{i\in\Omega} \phi(\mathbf{x}, M_i)\right) \tag{11.68}$$

where $\mathbf{x}$ is an input feature vector, Ω is the set of prototypes indexes, M_i ($i = 1$, 2, … k) are prototypes, and $C : \Omega \rightarrow \{\gamma\}$is the function to assign the class associated with a given prototype.

Use of large margin kernels for search of a linear discriminant model in high-dimensional feature space for pattern classification has been investigated by several investigators (34–37). For example, a radial basis function (RBF) can be used as a kernel function as

$$k(\mathbf{x},\mathbf{y}) = \exp\left(-\lambda\|\mathbf{x}-\mathbf{y}\|^2\right), \quad \lambda \geq 0. \tag{11.69}$$

A generalized kernel function can be expressed as

$$k(\mathbf{x},\mathbf{y}) = \left(\langle \mathbf{x},\mathbf{y}\rangle + u\right)^{\mathrm{d}}, \quad u \geq 0, d \in \mathrm{N} \tag{11.70}$$

where d is the dimensionality of classification space.

The relevance vector machine (RVM) (35) uses a model prototype for regression and classification exploiting a probabilistic Bayesian principle. There are several other models investigated for pattern classification using theoretic approaches from kernel-based classifier to linear programming perturbation-based methods (34–37).

A single prototype per class (as described above) can be used for multiclass classification using a Bayesian probabilistic model. For a correct classification using a multiclass classifier, the prototype of the correct class should have a larger score than the maximum scores related to all other incorrect classes. The multiclass margin for input vector $\mathbf{x}_i$ is then defined as (34)

$$p(\mathbf{x}_i, c_i \mid \mathbf{M}) = \langle M_{y_i}, \mathbf{x}_i \rangle - \max_{r \neq y_i} \langle M_r, \mathbf{x}_i \rangle \tag{11.71}$$

where y_i chosen such that $C(y_i) = c_i$, is the index of the prototype associated to the correct label for the training example $\mathbf{x}_i$. In the single prototype case, the associated class indices may be coincident, that is $y_i = c_i$.

It follows that for a correct classification of $\mathbf{x}_i$ with a margin greater than or equal to 1, the following condition has to be satisfied as described in Reference (34):

$$\langle M_{y_i}, \mathbf{x}_i \rangle \geq \theta_i + 1 \quad \text{where} \quad \theta_i = \max_{r \neq y_i} \langle M_r, \mathbf{x}_i \rangle. \tag{11.72}$$

Recently, the above single-prototype-based approach has been extended to multiprototype-based support vector machine (SVM) for multiclass classification by Aiolli and Sperduti (34).

11.4. IMAGE ANALYSIS AND CLASSIFICATION EXAMPLE: "DIFFICULT-TO-DIAGNOSE" MAMMOGRAPHIC MICROCALCIFICATIONS

Breast microcalcifications are often fuzzy and poorly defined structures. In such cases, it is difficult to distinguish between benign and malignant microcalcifications associated with breast cancer. There have been a number of approaches used in the computer-aided analysis of mammographic microcalcifications (2, 25, 38–41). Artificial neural networks and fuzzy logic-based feature analysis have also been used for detection (2, 25, 41).

In the analysis of difficult-to-diagnose mammographic microcalcifications, Dhawan et al. used the second-order histogram statistics and wavelet processing to represent texture feature for classification into benign and malignant cancer categories. Two sets of 10 wavelet features were computed for the discrete Daubechies filter prototypes, D_6 and D_{20}, where the subscripts indicate the size of the filter. In total, 40 features were extracted and used in a GA-based feature reduction and

correlation analysis. The initial set of 40 features included 10 binary segmented microcalcification cluster features (Feature #1–10), 10 global texture-based image structure features (Feature #11–20), and 20 wavelet analysis-based local texture features (Feature #21–40). These features are listed below (2).

1. Number of microcalcifications
2. Average number of pixels per microcalcification (area)
3. Standard deviation of number of gray levels per pixel
4. Average gray level per microcalcification
5. Standard deviation of gray levels
6. Average distance between microcalcifications
7. Standard deviation of distances between microcalcifications
8. Average distance between calcification and center of mass
9. Standard deviation of distances between calcification and center of mass
10. Potential energy of the system using the product of the average gray level and the area.
11. Entropy of $H(y_q, y_r, d)$
12. Angular second moment of $H(y_q, y_r, d)$
13. Contrast of $H(y_q, y_r, d)$
14. Inverse difference moment of $H(y_q, y_r, d)$
15. Correlation of $H(y_q, y_r, d)$
16. Mean of $H(y_q, y_r, d)$
17. Deviation of $H_m(y_q, d)$
18. Entropy of $H_d(y_s, d)$
19. Angular second moment of $H_d(y_s, d)$
20. Mean of $H_d(y_s, d)$
21. Energy for the D_6 wavelet packet at level 0
22. Energy for the D_6 low-low wavelet packet at level 1
23. Energy for the D_6 low-high wavelet packet at level 1
24. Energy for the D_6 high-low wavelet packet at level 1
25. Energy for the D_6 high-high wavelet packet at level 1
26. Entropy for the D_6 wavelet packet at level 0
27. Entropy for the D_6 low-low wavelet packet at level 1
28. Entropy for the D_6 low-high wavelet packet at level 1
29. Entropy for the D_6 high-low wavelet packet at level 1
30. Entropy for the D_6 high-high wavelet packet at level 1
31. Energy for the D_{20} wavelet packet at level 0
32. Energy for the D_{20} low-low wavelet packet at level 1
33. Energy for the D_{20} low-high wavelet packet at level 1

34. Energy for the D_{20} high-low wavelet packet at level 1

35. Energy for the D_{20} high-high wavelet packet at level 1

36. Entropy for the D_{20} wavelet packet at level 0

37. Entropy for the D_{20} low-low wavelet packet at level 1

38. Entropy for the D_{20} low-high wavelet packet at level 1

39. Entropy for the D_{20} high-low wavelet packet at level 1

40. Entropy for the D_{20} high-high wavelet packet at level 1

Genetic algorithms were used to select the best subset of features from the binary cluster, global, and local textural representations.

Using the GA, the initial set of 40 features was reduced to the two best correlated set of 20 features: VP1 (using the proportional selection in GA), and VR1 (using the ranking selection in GA). These feature sets are shown in Table 11.2, with the Feature number from the above list.

Selected features were used as inputs to the radial basis function for subsequent classification of the microcalcifications. Although the best chromosomes, and thus the feature inputs, were chosen on the basis of fitness, the ultimate measure of performance is the area under the receiver operating characteristic (ROC) curve. The optimal network architecture was determined heuristically for the combined feature set for different numbers of samples in the training and test sets. The maximum and average area over 40 partitions of the data are indicated in Table 11.3 for the

TABLE 11.2 Lists of Two Best Sets of Features Selected Through Genetic Algorithm Using Proportional Selection (VP1) and Ranking Selection (VR1)

Experiment	Feature list
VP1	1,3,5,8,11,13,16,20,21,22,23,25,26, 27,28,30,31,35,37,39
VR1	1,3,4,5,7,13,15,16,22,23,24,25.27,29,31,32,36,37,39,40

TABLE 11.3 ROC Performance of the Radial Basis Function Network Trained on Various Combined Input Sets and Number of Training Samples Using Proportional and Ranking Selection and *K*-Means Clustering to Place the Basis Units

Experiment	Network architecture	Maximum area	Average area	Standard deviation
VP-1	A	0.743	0.712	0.046
	B	0.824	0.773	0.048
	C	0.801	0.732	0.049
	D	0.829	0.795	0.047
	E	0.857	0.81	0.045
VR-l	A	0.77	0.738	0.047
	B	0.751	0.725	0.049
	C	0.794	0.737	0.047
	D	0.827	0.798	0.047
	E	0.874	0.83	0.044

combined feature set using both the proportional and ranking selection schemes for the input parameters selected by the GA. Only the performance of the combined feature set is presented in this chapter. It can be seen in Table 11.3 that the discrimination ability of each experiment typically increases with increasing numbers of training samples. Computer-aided analysis of mammographic microcalcifications seems to have the potential to help reduce the false positive rate without reducing the true positive rate for difficult-to-diagnose cases (2, 23–25).

11.5. EXERCISES

11.1. Describe a basic paradigm of image analysis and classification. Why is it important to compute feature to represent regions?

11.2. What do you understand by hierarchical representation of features? Explain with the help of an example.

11.3. How do knowledge models improve image analysis and classification tasks?

11.4. How can Hough transform be used in shape representation and analysis?

11.5. Explain the difference between the chain code and Fourier descriptor for boundary representation.

11.6. Show that Fourier boundary descriptors are size, rotation, and translation invariant.

11.7. What is the role of higher-order moments in shape representation and analysis?

11.8. Describe the seven invariant moments for shape description.

11.9. What are the fundamental operations in morphological image processing?

11.10. What is the difference between morphological closing and opening operations?

11.11. What is the role of a structuring element in morphological image processing?

11.12. Describe a rule-based system for region merging and splitting.

11.13. Define texture in an image. What are the major features to represent image texture?

11.14. Explain why you need to select the best correlated features for image classification.

11.15. Why is a dimensionality reduction approach such as linear discriminant analysis needed before the features are classified?

11.16. Explain the method associated with principal component analysis. How is this method different than linear discriminant analysis?

11.17. What are the different paradigms of classification?

11.18. Explain in detail the nearest-neighborhood classifier. What are the limitations and shortcomings of this method?

11.19. What are the advantages and disadvantages of a neural network classifier over statistical classifiers?

11.20. Can a two-layered backpropagation neural network provide a nonlinear partitioning of multidimensional feature space for classification?

11.21. Describe a structure of a multilayered backpropagation neural network classifier with training algorithm for multiclass classification.

11.22. In the MATLAB environment, segment serial brain images for ventricles using the region growing method. Obtain ventricle boundaries in segmented images.

11.23. Develop a relational model of the hierarchical structure of ventricle shapes and assign a set of features to each ventricle. Develop this model in the MATLAB environment.

11.24. Compute a set of features for representing shapes of all ventricles for the segmented images obtained in Exercise 11.16.

11.25. Develop a nearest-neighbor classifier using the computed features for classification of ventricles segmented in Exercise 11.16.

11.26. Select a new MR brain image and perform region segmentation for ventricular structure. Select correlated features for the segmented regions and use the classifier developed in Exercise 11.19 for classification of ventricles. Comment on the success of your classifier in ventricle classification.

11.27. Select a set of five features for the classification of mammographic microcalcifications. In the MATLAB environment, compute the selected features on the labeled images of benign and malignant microcalcifications. Divide all cases into two groups: training and test. Now develop a radial basis neural network (RBFNN) for microcalcification classification using the training set. Obtain the true positive and false positive rates of the classifier on the test cases. Comment on the success of the classifier.

11.28. Repeat Exercise 11.21 using a backpropagation neural network classifier and compare the results with those obtained using the RBFNN.

11.29. What is a support vector machine (SVM)? How is it different from backpropagation and RBF neural networks?

11.30. Repeat Exercise 11.21 using a SVM and compare the results with BPNN and RBFNN.

11.6. REFERENCES

1. M.H. Loew, "Feature extraction," in M. Sonka and J.M. Fitzpatrick (Eds), *Handbook of Medical Imaging,* Volume 2: *Medical Image Processing and Analysis*, SPIE Press, Bellingham, WA, 2000.
2. A.P. Dhawan, Y. Chitre, C. Kaiser-Bonasso, and M. Moskowitz, "Analysis of mammographic microcalcifications using gray levels image structure features," *IEEE Trans. Med. Imaging*, Vol. 15, pp. 246–259, 1996.
3. L. Xu, M. Jackowski, A. Goshtasby, C. Yu, D. Roseman, A.P. Dhawan, and S. Bines, "Segmentation of skin cancer images," *Image Vis. Comput.*, Vol. 17, pp. 65–74, 1999.

4. A.P. Dhawan and A. Sicsu, "Segmentation of images of skin lesions using color and texture information of surface pigmentation," *Comput. Med. Imaging Graph.*, Vol. 16, pp. 163–177, 1992.
5. D.H. Ballard and C.M. Brown, *Computer Vision*, Prentice Hall, Englewood Cliffs, NJ, 1982.
6. R.C. Gonzalaez and R.E. Wintz, *Digital Image Processing*, Prentice Hall, Englewood Cliffs, NJ, 2002.
7. M.K. Hu, "Visual pattern recognition by moments invariants," *IRE Trans. Inf. Theo.*, Vol. 8, pp. 179–187, 1962.
8. J. Flusser and T. Suk, "Pattern recognition by affine moments invariants," *Pattern Recog.*, Vol. 26, pp. 167–174, 1993.
9. J. Serra, *Image Analysis and Mathematical Morphology*, Academic Press, San Diego, CA, 1982.
10. L.H. Staib and J.S. Duncan, "Boundary finding with parametrically deformable models," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 14, pp. 1061–1075, 1992.
11. S. Sternberg, L. Shapiro, and R. MacDonald, "Ordered structural shape matching with primitive extraction by mathematical morphology," *Pattern Recog.*, Vol. 20, pp. 75–90, 1987.
12. P. Maragos, "Pattern spectrum and multiscale shape representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 11, pp. 701–716, 1989.
13. S. Loncaric and A.P. Dhawan, "A morphological signature transform for shape description," *Pattern Recog.*, Vol. 26, No. 7, pp. 1029–1037, 1993.
14. S. Loncaric, A.P. Dhawan, T. Brott, and J. Broderick, "3-D image analysis of intracerebral brain hemorrhage," *Comput. Methods Prog. Biomed.*, Vol. 46, pp. 207–216, 1995.
15. S. Loncaric and A.P. Dhawan, "Optimal MST-based shape description via genetic algorithms," *Pattern Recog.*, Vol. 28, pp. 571–579, 1995.
16. H. Samet and M. Tamminen, "Computing generic properties of images represented by quadtrees," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 7, pp. 229–240, 1985.
17. Y. Shirari and J. Tsujii, *Artificial Intelligence: Concepts, Techniques and Applications*, John Wiley & Sons, New York, 1984.
18. A.P. Dhawan and S. Juvvadi, "A knowledge-based approach to the analysis and interpretation of CT images," *Comput. Methods Prog. Biomed.*, Vol. 33, pp. 221–239, 1990.
19. J. Broderick, S. Narayan, A.P. Dhawan, M. Gaskil, and J. Khouri, "Ventricular measurement of multifocal brain lesions: Implications for treatment trials of vascular dementia and multiple sclerosis," *Neuroimaging*, Vol. 6, pp. 36–43, 1996.
20. A.M. Nazif and M.D. Levine, "Low-level image segmentation: An expert system," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 6, pp. 555–577, 1984.
21. R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, 2nd Edition, John Wiley & Sons, New York, 2001.
22. I.T. Jolliffe, *Principal Component Analysis, Springer Series in Statistics*, 2nd Edition, Springer, New York, 2002.
23. C. Peck and A.P. Dhawan, "A review and critique of genetic algorithm theories," *J. Evol. Comput.*, Vol. 3, pp. 39–80, 1995.
24. Y. Chitre, A.P. Dhawan, and M. Moskowitz, "Classification of mammographic microcalcifications," in K.W. Boyer and S. Astlay (Eds), *State of the Art in Digital Mammographic Image Analysis*, World Scientific Publishing Co., New York, 1994, pp. 167–197.
25. Z. Huo, M.L. Giger, and C.J. Vyborny, "Computerized analysis of multiple mammographic views: Potential usefulness of special view mammograms in computer aided diagnosis," *IEEE Trans. Med. Imaging*, Vol. 20, pp. 1285–1292, 2001.
26. S.A. Stansfield, "ANGY: A rule based expert system for automatic segmentation of coronary vessels from digital subtracted angiograms," *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 8, pp. 188–199, 1986.
27. L.K. Arata, A.P. Dhawan, A.V. Levy, J. Broderick, and M. Gaskil, "Three-dimensional anatomical model based segmentation of MR brain images through principal axes registration," *IEEE Trans. Biomed. Eng.*, Vol. 42, No. 11, pp. 1069–1078, 1995.
28. J.M. Zurada, *Introduction to Artificial Neural Systems*, West Publishing Co., Boston, 1992.
29. S. Mitra and S.K. Pal, "Fuzzy multi-layer perceptron, inferencing and rule generation," *IEEE Trans. Neural Netw.*, Vol. 6, No. 1, pp. 51–63, 1995.
30. L.A. Zadeh, "Fuzzy sets as a basis for a theory of possibility," *Fuzzy Sets Syst.*, Vol. 1, pp. 3–28, 1978.

31. Y-Q. Zhang and A. Kandel, "Compensatory neurofuzzy systems with fast learning algorithms," *IEEE Trans. Neural Netw.*, Vol. 9, No. 1, pp. 83–105, 1998.
32. V. Petridis and V.G. Kaburlasos, "Fuzzy lattice neural network (FLNN): A hybrid model for learning," *IEEE Trans. Neural Netw.*, Vol. 9, No. 5, pp. 877–890, 1998.
33. W. Grohman and A.P. Dhawan, "Fuzzy convex set based pattern classification of mammographic microcalcifications," *Pattern Recog.*, Vol. 34, No. 7, pp. 119–132, 2001.
34. F. Aiolli and A. Sperduti, "Multiclass classification with multi-prototype support vector machines," *J. Mach. Learn. Res.*, Vol. 6, pp. 817–850, 2005.
35. M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, Vol. 1, pp. 211–244, 2001.
36. T. Downs, K.E. Gates, and A. Masters, "Exact simplification of support vector solutions," *J. Mach. Learn. Res.*, Vol. 2, pp. 293–297, 2001.
37. F. Aiolli and A. Sperduti, "An efficient SMO-like algortihm for multiclass SVM," in *Proceedings of IEEE Workshop on Neural Networks for Signal Processing*, IEEE Press, Piscataway, NJ, pp. 297–306, 2002.
38. W.B. Johnson Jr, "Wavelet packets applied to mammograms," in R. Acharya and D.B. Goldof (Eds), Proc. SPIE 1905, *Biomedical Image Processing and Biomedical Visualization*, SPIE Press, Bellingham, WA, 1993, pp. 504–508.
39. R.N. Strickland and H.I. Hahn, "Detection of microcalcifications using wavelets," in A.G. Gale, A.M. Astley, D.R. Dance, and A.Y. Cairns (Eds), *Digital Mammography*, Proceedings, 2nd International Workshop on Digital Mammography, Elsevier Science B.V., Amsterdam, 1994, pp. 79–88.
40. W. Qian, L.P. Clarke, M. Kallergi, H.D. Li, R.P. Velthuizen, R.A. Clarke, and M.L. Silbigier, "Tree-structured nonlinear filter and wavelet transform for microcalcification segmentation in mammography," in R. Acharya and D.B. Goldof (Eds), Proc. SPIE 1905, *Biomedical Image Processing and Biomedical Visualization*, SPIE Press, Bellingham, WA, 1993, pp. 509–521.
41. N. Mudigonda, R.M. Rangayyan, and J.E. Leo Desautals, "Detection of breast masses in mammograms by density slicing and texture flow-field analysis," *IEEE Trans. Med. Imaging*, Vol. 120, pp. 1215–1227, 2001.