2

PATTERN CLASSIFICATION METHODS

The classical techniques of pattern recognition are concerned with assigning a given pattern to one of the known, finite classes. These techniques are surveyed briefly in this chapter and their possible applications and limitations examined. The problem of perception extends beyond that of classification; often it is necessary to generate descriptions of a new pattern and analyze its similarities and differences with other known patterns.

Before discussing different approaches, we need to study the digital representation of an image.

2.1 DIGITAL REPRESENTATION OF AN IMAGE

An image may be thought of as a function giving the light intensity at each point over a planar region. For operations by a digital computer, we need to sample this function at discrete intervals and quantize the intensity into discrete levels. The points at which the image is sampled are known as *picture elements*, commonly abbreviated as *pixels*. The intensity at each pixel is represented by an integer, say 0 for black and 255 for white, and is determined from the continuous image by averaging over a small neighborhood around the pixel location. It is common to use a square sampling grid with pixels equally spaced along the two sides of the grid.

The distance between grid points obviously affects the accuracy with which the original image is represented, and it determines the fine detail that can be resolved. (Of course, the resolution depends on the imaging system as well.) For most images of interest—those that have a bound on their spatial frequency—a certain sampling distance is sufficient to reproduce the image perfectly according to the well-known Shannon-Whittaker theorem in communication theory [1] (assuming no quantization of intensity levels). In the remainder of the book we will assume the input to be a digital image with a given resolution.

2.1.1 Connectivity in Digital Images

The geometry and topology of a digital plane differ from those of a continuous domain in many important aspects. While horizontal and vertical lines are easily represented on a square grid, straight lines at many other angles can only be approximated by a staircaselike pattern (see Fig. 2-1).

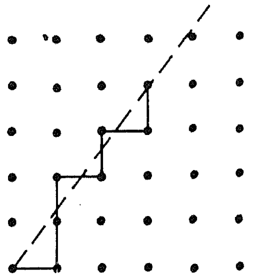


Figure 2-1: Digital approximation of a line

The connectivity of patterns in a discrete representation is more complex than for the continuous case. To determine connectivity, we need to define the notion of adjacency of two points in a digitized plane. In Fig. 2-2, either four pixels, *A* through *D*, or all eight pixels, *A* through *H*, may be considered adjacent to the center pixel. The two adjacencies are known as 4-adjacency and 8-adjacency, respectively. A set of points forms a 4 (or 8) connected figure if there is a path between any two points through 4 (or 8) adjacent points.

These definitions of connectedness can yield counterintuitive results in some cases. In Fig. 2-3, the set of points with value 1 is 8-connected, but so are their "interior" and "exterior" backgrounds. On the other hand, this set of points is not 4-connected, but neither are interior and exterior backgrounds, and we have an unconnected curve

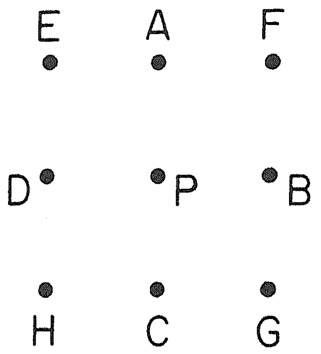


Figure 2-2: Neighbors of a pixel *P*

			1	1			
		1			1		
		1			1		
			1	1			

Figure 2-3: Connectivity in digital plane

separating two regions in a plane.

If a hexagonal grid is used, instead of a square one, this anomaly disappears. In a hexagonal grid, a point at the center of a hexagon is considered to be connected to the six pixels at the corners of the hexagon; see Fig. 2-4. However, hexagonal grids have not proved popular, perhaps because of the complexity of the digitization (details of the use of hexagonal grids may be found in [2]). It is interesting to note that triangular grids are the only other symmetric and isotropic grid that can be used to span a plane. A detailed treatment of digital topology may be found in [3].

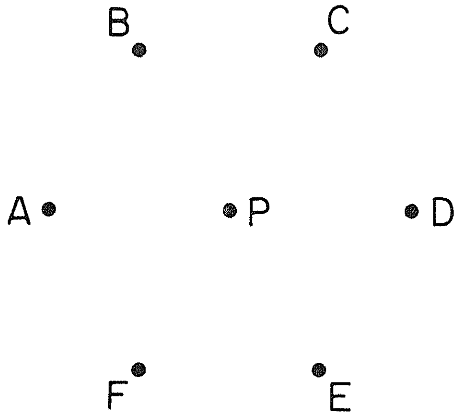


Figure 2-4: A point *P* and its hexagonal neighbors

2.2 TEMPLATE MATCHING

The most immediate method of classifying a pattern is to compare it with stored models of known patterns and choose the best match. In template matching, this comparison is performed directly on images. Templates consist of images of known patterns, one from each class. Each template is moved over a new image to find the best match. Let $F(i, j)$ represent the image and $A(i, j)$ represent the template. For a translation of the template by (x, y) a measure of the match between the image and the template is given by

$$E(x, y) = \sum_i \sum_j [F(i, j) - A(i - x, j - y)]^2 \tag{2-1}$$

where the summation is over the overlapping regions of the image and the translated template. Matches at different values of (x, y) are computed to search for a minimum value of E. The template giving the lowest error, E, is taken to be the best classification for the input pattern. Two simpler measures of match are the sum of absolute differences and the sum of the maxima of the absolute differences, instead of the sum of the squares of differences in Eq. (2-1) above.

Above measures require the patterns to be matched to have the same intensity values. A measure known as normalized cross-correlation is insensitive to the absolute intensity and also to the contrast and is defined by

$$\sigma(x, y) = \frac{\sum\sum\{F(i, j) \cdot A(i - x, j - y)\}}{\sqrt{\sum\sum F^2(i, j)} \cdot \sqrt{\sum\sum A^2(i - x, j - y)}} \tag{2-2}$$

with all summations over the same range as in Eq. (2-1) above. σ achieves a maximum value of 1 for an exact match.

As defined above, template matching achieves pattern classification invariant to translation, but not to rotation, scale (size), or perspective changes. Template matching with different scales and orientations is computationally expensive. To account for variations within a class, the template may be stretched and deformed to fit a pattern with a measure related to the amount of deformation [4]. Limited changes in perspective may be accommodated by using a number of templates with different perspective views.

It is sometimes useful to represent a template as consisting of smaller subtemplates with specified spatial relationships. Matching of subtemplates can then be independent of each other, and a further measure of the spatial relationships in the input pattern is used to evaluate the whole match. In one approach, the subtemplates were considered to be attached by springs, and extension or compression of these springs was used as a measure of a global match [5]. This technique allows different weights to be assigned to different spatial relationships and allows some flexibility within a pattern class. However, the basic limitations of sensitivity to changes in scale and perspective remain.

In summary, template-matching techniques are useful for applications where the number of classes and the variability within a class are small. A prime example of successful application is for recognition of printed, fixed-font alphabetic characters in commercially available optical character reader (OCR) devices.

2.3 PATTERN CLASSIFICATION IN FEATURE SPACE

An alternative to template matching, which proceeds at the image level, is to abstract some measurements or features from the pattern and classify it based on these measurements. This paradigm is illustrated schematically in Fig. 2-5 and allows the separation of the recognition problem into two more or less independent parts.

The measured features may be considered to span an n-dimensional feature space, and different regions of this space correspond to different pattern classes. A hypothetical example for two features is shown in Fig. 2-6. The power of this paradigm is strongly

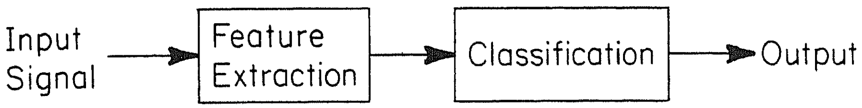


Figure 2-5: Block diagram for pattern-classification approach

dependent on the availability of features that are invariant to the expected changes in the input patterns. The choice of features is problem dependent. However, the classification methods can be independent of the problem domain and need not be restricted to pictorial inputs.

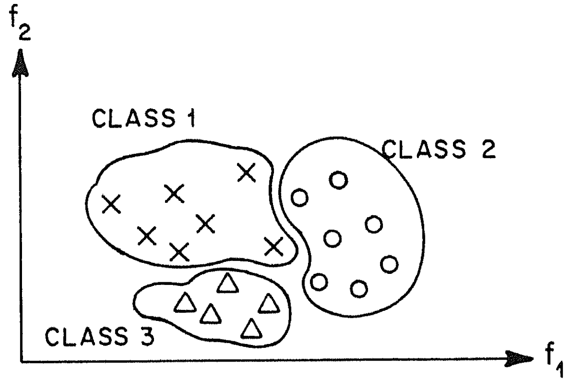


Figure 2-6: Three classes separated in feature space f_1-f_2

Elegant theories have been developed for classification that minimize the probability of false classification based on known (or assumed) a priori probabilities of occurrence of certain patterns and conditional probabilities of their occurrence given a feature vector. A simple technique is to assign a given point in feature space to the same class as that of its nearest (closest) neighbor in this space. This simple method has an error rate no worse than twice that of an optimal method. A major concern of the pattern-classification methods is with the simplicity of the classification rule, and a particular favorite is classification by dividing the feature space into linearly separable regions—that is, by hyperplanes in the feature space (straight lines in a plane). These classification methods are not described in detail here; several comprehensive textbooks on the subject exist [6-8]. A particular pattern classification machine, the perceptron, is discussed in some detail below.

2.4 PERCEPTRONS

A frequently suggested and attractive organization for pattern classification is that the analysis of an input pattern must proceed in stages. At each stage, computations are performed on local areas only, and global relations are derived from combinations of the local computations, in a presumed analogy with the human visual system. A specific scheme is as shown in Fig. 2-7. In the figure ϕ_i's are arbitrarily complex functions limited to operating on a local neighborhood and Ω is a decision function based on the results of computations of various ϕ_i's (the neighborhoods of ϕ_i's may overlap).

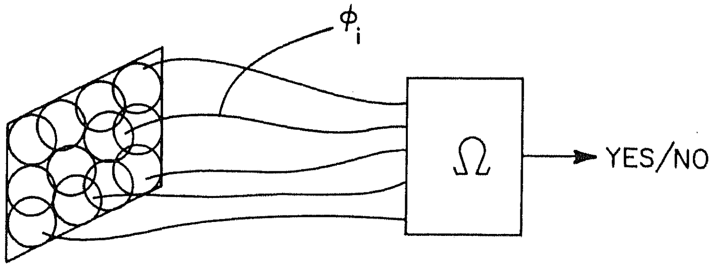


Figure 2-7: A perceptron

To be more specific, let each ϕ_i be a predicate function (that is, with a value of one or zero). Let the decision function, Ω, have value one (yes, true) if a weighted sum of the inputs, ϕ_i, exceeds a certain threshold and zero (no, false), otherwise. That is,

$$\Omega = 1, \text{ if } \sum_i \alpha_i\phi_i > \tau$$
$$= 0, \text{ otherwise}$$

A machine using such a decision function is called a *perceptron* and was proposed by Rosenblatt [9]. The thresholding element was believed to be similar to the simpler models of neurons in animal brains. It has been further suggested that such machines are capable of "learning," by adjustments of the weights in accordance with the output of the machine in response to known input patterns.

As a simple example, consider the problem of recognizing whether the input pattern is a rectangle of any size located at an arbitrary position, with one of the axes bein horizontal (see Fig. 2-8). Let us define four predicate functions $\phi_1(i, j)$, $\phi_2(i, j)$, $\phi_3(i, j)$, and $\phi_4(i, j)$, each using three pixels in the input pattern. For these functions to be true, pixel (i, j) must have a value of 1 and neighbors must have values as specified below:

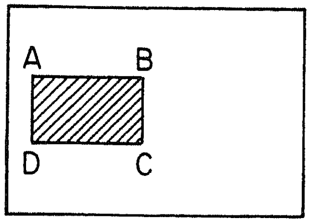


Figure 2-8: A rectangle in an image

$\phi_1(i, j)$ requires the south and east neighbors—that is, pixels $(i+1, j)$ and $(i, j+1)$—to have value 0,
$\phi_2(i, j)$ requires south and west neighbors to be 0,
$\phi_3(i, j)$ requires north and west neighbors to be 0, and
$\phi_4(i, j)$ requires north and east neighbors to be 0.

These functions essentially detect the presence of a certain type of corner in the input pattern. In Fig. 2-8, ϕ_1 has value 1 at corner *A*, ϕ_2 at corner *B*, ϕ_3 at corner *C*, and ϕ_4 at corner *D*. The presence of the desired rectangle is indicated if and only if

$$\sum\phi_1 + \sum\phi_2 + \sum\phi_3 + \sum\phi_4 \leqslant 4 \qquad (2\text{-}3)$$

(verification for the general case is left to the reader).

For many years it was believed that perceptrons were general classifiers capable of learning to discriminate between arbitrary sets of patterns by proper choice of α_i's (and perhaps ϕ_i's), and much effort was devoted to discovering efficient learning algorithms [10]. However, Minsky and Papert, in a classic book [11], demonstrated that the discrimination ability of these machines is extremely limited.

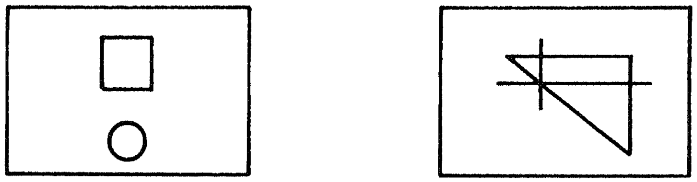
Before results on the power of perceptrons can be established, some restrictions must be placed on the feature detecting functions, the ϕ_i's. We allow the functions to be arbitrarily complex predicate functions, but restrict the range of inputs in the following two ways:

1. Each ϕ_i is allowed to operate in a neighborhood enclosed within a circle of a certain diameter. Machines with this restriction will be called *diameter limited* perceptrons.
2. Each ϕ_i may use at most only a limited number of points, say k,

selected from anywhere in the input pattern. Machines with this restriction will be called *order limited* (of order *k*) perceptrons.

Clearly, if the order or the diameter of a perceptron is allowed to be large enough to include all points in the input pattern, discrimination between any two patterns is possible (as each ϕ_i is arbitrarily complex). Such machines will be said to be *not* of finite diameter (or order). The interesting results are about the capabilities of finite order or finite diameter perceptrons. Some of the principal results of Minsky and Papert's work are stated below.

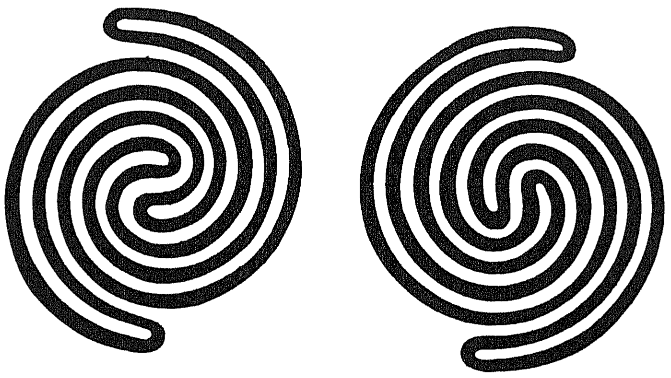
1. Finite-order and finite-diameter perceptrons can be devised to discriminate rectangles from other figures. Circles can be discriminated by order-limited perceptrons (of order four) but not by diameter-limited perceptrons. However, figures embedded in other figures cannot be so discriminated by *any* finite-order or finite-diameter perceptrons. For example, such machines cannot be devised to correctly answer that a rectangle is contained in Fig. 2-9(a) or (b).
2. If invariance to *any* group of transformations is desired, patterns must be discriminable by area alone.
3. If invariance to *any* topological transformation is desired, the patterns must be discriminable by their Euler number (number of connected components - number of holes).
4. A perceptron of order three can discriminate between convex and concave patterns.
5. A finite-order or finite-diameter perceptron cannot be constructed to discriminate between connected or disconnected patterns, such as in Fig. 2-10, where one of the patterns is connected, the other is not. (In this example, the connectedness property is difficult to perceive for humans as well, but the perceptrons also fail for simpler examples.)



(a) (b)

Figure 2-9: The rectangle as a component in two patterns

The above results are largely negative and somewhat surprising in that simple global properties such as connectedness cannot be inferred from local properties by a perceptron. Multilevel perceptrons, utilizing more than one level of composition of local features, have also been proposed. However, no important experimental results or theoretical analyses of such machines have been reported. (Results of preliminary analysis for machines slightly more general than those analyzed by Minsky and Papert also seem to be negative [12].)



(a) (b)

Figure 2-10: An unconnected and a connected figure

2.5 SYNTACTICAL APPROACHES

In the pattern-classification approaches of the previous two sections, the measured features were interpreted to span a feature space and *relations* between features were not considered. An alternative approach is to divide a pattern into primitive subpatterns and use the relations between subpatterns (or their features). For example, it is convenient to represent a triangle as three straight lines (primitives) connected at their end points. Such approaches are said to use *syntactical* or *structural* relations of a pattern.

Syntactical approaches were first applied to the *parsing* of sentences in programming and natural languages. These sentences may be viewed as one-dimensional patterns with words as subpatterns. For programming languages, the allowed relations of adjacency or concatenation of words of certain classes are defined by formal rules, known as *formal grammars*. However, attempts to find similar rules for natural language sentences have not been very successful, and it is widely recognized that the rules of grammar are dependent also on the meaning or the *semantics* of the words, and not just on their positions in the sentence.

Use of formal grammars to specify patterns in two and three dimensions is more complex, as the relationships between subpatterns are not completely specified by adjacency. In one early approach, the primitive subpatterns were restricted to be vectors with a given orientation, and the only allowed relations between them were by attachments at their beginnings or ends [13]. Two-dimensional patterns were represented by a predefined central "axis" vector. Rules for patterns formed by combining vectors in these ways can be specified by grammars similar to those used for string languages.

Unfortunately, the term syntactical approach is often used for techniques that restrict the range of syntactical relations to those specified by a formal grammar; a more appropriate term for the latter would be *grammatical* pattern recognition. If the class of patterns of interest can be specified by a formal grammar, many powerful techniques of formal language parsing become applicable. This has been achieved with some success for restricted applications; some examples are character recognition [14], bubble chamber particle trace analysis [13], and chromosome analysis [15]. However, the search for grammars describing large classes of natural patterns has not yet been very successful.

Grammatical pattern recognition is not discussed in the remainder of this book; a thorough treatment may be found in [16]. However, nongrammatical structural relationships between parts of patterns are a key method used in the techniques described in this book. We will use the term "structural" rather than "syntactical" in the hope of avoiding confusion with the grammatical methods.

2.6 SUMMARY

This chapter has provided a brief overview of mathematical pattern recognition (also called *classical* or *statistical* pattern recognition) techniques. Although the proposed paradigms are of great generality, their applications to pictorial pattern recognition have been limited, owing to difficulties of defining suitable features (or grammars). The view taken in this book is that such techniques may play a useful role in some parts of a complete system, but they do not provide all the capabilities required for machine perception, particularly for three-dimensional scenes.

The descriptive approach is developed in the remainder of the book. A good understanding of many issues can be obtained by studying first the simpler problems of perception of polyhedral scenes.

REFERENCES

[1] J. W. Goodman, *Introduction to Fourier Optics*, McGraw Hill, New York, 1968.

[2] M. J. E. Golay, "Hexagonal Parallel Pattern Transformations," *IEEE Transactions on Electronic Computers*, C-18, August 1969, pp. 733-740.

[3] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, New York, 1976.

[4] B. Widrow, "The Rubber-Mask Technique, I and II," *Pattern Recognition*, Vol. 5, 1973, pp. 175-211.

[5] M. A. Fischler and R. A. Elschlager, "The Representation and Matching of Pictorial Structures," *IEEE Transactions on Computers*, Vol. C-22, No. 1, January 1973, pp. 67-92.

[6] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.

[7] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, 1972.

[8] H. C. Andrews, *Introduction to Mathematical Techniques in Pattern Recognition*, John Wiley & Sons, New York, 1972.

[9] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan Books, Washington D.C., 1962.

[10] N. J. Nilsson, *Learning Machines: Foundations of Trainable Pattern Classifying Systems*, McGraw-Hill, New York, 1965.

[11] M. Minsky and S. Papert, *Perceptrons, An Introduction to Computational Geometry*, MIT Press, Cambridge, Mass., 1969.

[12] H. Abelson, "Computational Geometry of Linear Threshold Functions," MIT Artificial Intelligence Laboratory Memo 376, July 1976.

[13] A. C. Shaw, "A Formal Picture Description Scheme as a Basis for Picture Processing Systems," *Information and Control*, Vol. 14, 1969, pp. 9-52.

[14] R. Narasimhan, "Syntax-directed Interpretation of Classes of Pictures," *Communications of the ACM*, Vol. 9, 1966, pp. 166-173.

[15] R. S. Ledley, "High Speed Automatic Analysis of Bio-Medical Pictures," *Science*, Vol. 146, 1964, pp. 216-223.

[16] K. S. Fu, *Syntactical Methods in Pattern Recognition*, Academic Press, New York, 1974.