EE 604, Digital Image Processing

Image Pattern Classification

Dr. W. David Pan

Dept. of ECE

UAH

Pattern Classification

- A pattern is a special arrangement of **features**.
- A **pattern class** is a set of patterns that share some common properties
- The job of pattern recognition system is to assign a class label to each of its unknown input patterns.
- Four stages of pattern recognition
  - Sensing
  - Preprocessing
  - Feature Extraction
  - Classification

Image Classification Approaches

1. Prototype Matching
2. Optimal Statistical Formulation
   – Applications where the nature of the data is well understood, allowing for effective pairing of features and classifiers
   – Rely on a great deal of engineering to define the features and elements of a classifier
3. Neural Networks
   – Features are learned by systems, rather than being specified *a priori* by a human designer.

Unsupervised and Supervised Training

- Unlabeled Data
  - The class of each patter is unknown, e.g., seeking clusters in a data set.
  - **Unsupervised Training**
    - Design a classifier by using unlabeled data.
- Labeled Data
  - We know the class of each pattern, e.g., in character recognition problem.
  - **Supervised Training**

    Design a classifier with labeled data by dividing the datasets into three subsets in general:
    - *Training Set*
    - *Validation Set*
    - *Test Set*

Patterns & Pattern Classes

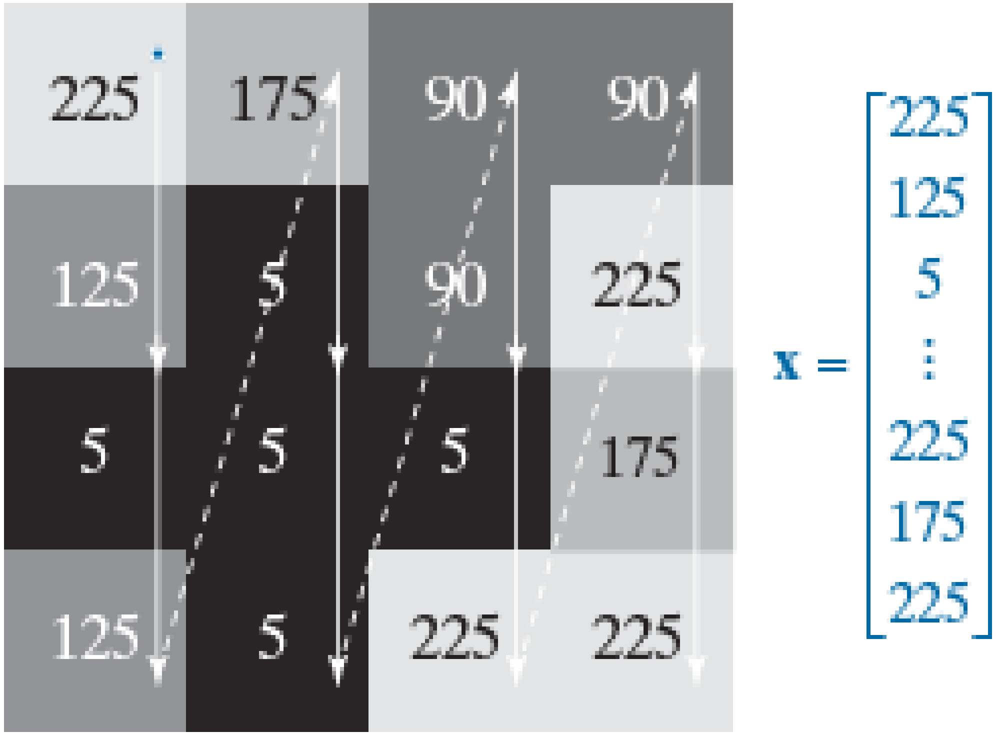
In image pattern classifications, the two principal pattern arrangements are

- Pattern Vectors
  - Quantitative patterns
- Structural Patterns
  - Composed of symbols arranged in the form of strings, trees, etc.

Pattern Vector Formed by Linear Indexing

a b

FIGURE 13.1 Using linear indexing to vectorize a grayscale image.



225	175	90	90
125	5	90	225
5	5	5	175
125	5	225	225

$$\mathbf{x} = \begin{bmatrix} 225 \\ 125 \\ 5 \\ \vdots \\ 225 \\ 175 \\ 225 \end{bmatrix}$$

Feature Vector

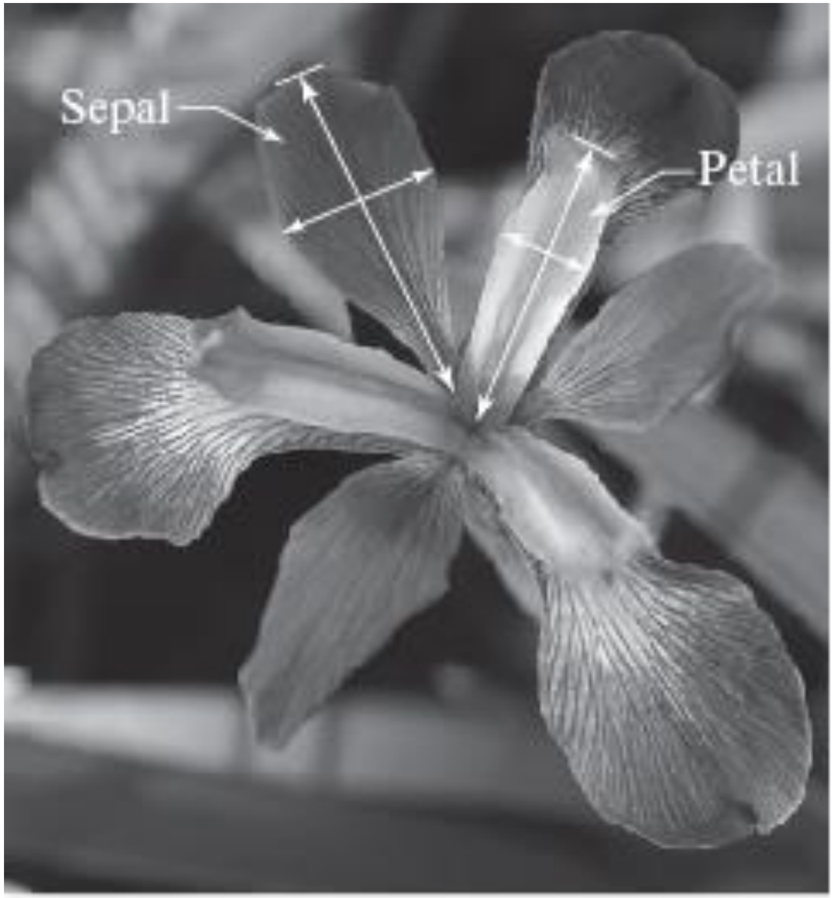


FIGURE 13.2 Petal and sepal width and length measurements (see arrows) performed on iris flowers for the purpose of data classification. The image shown is of the *Iris virginica* gender. (Image courtesy of USDA.)

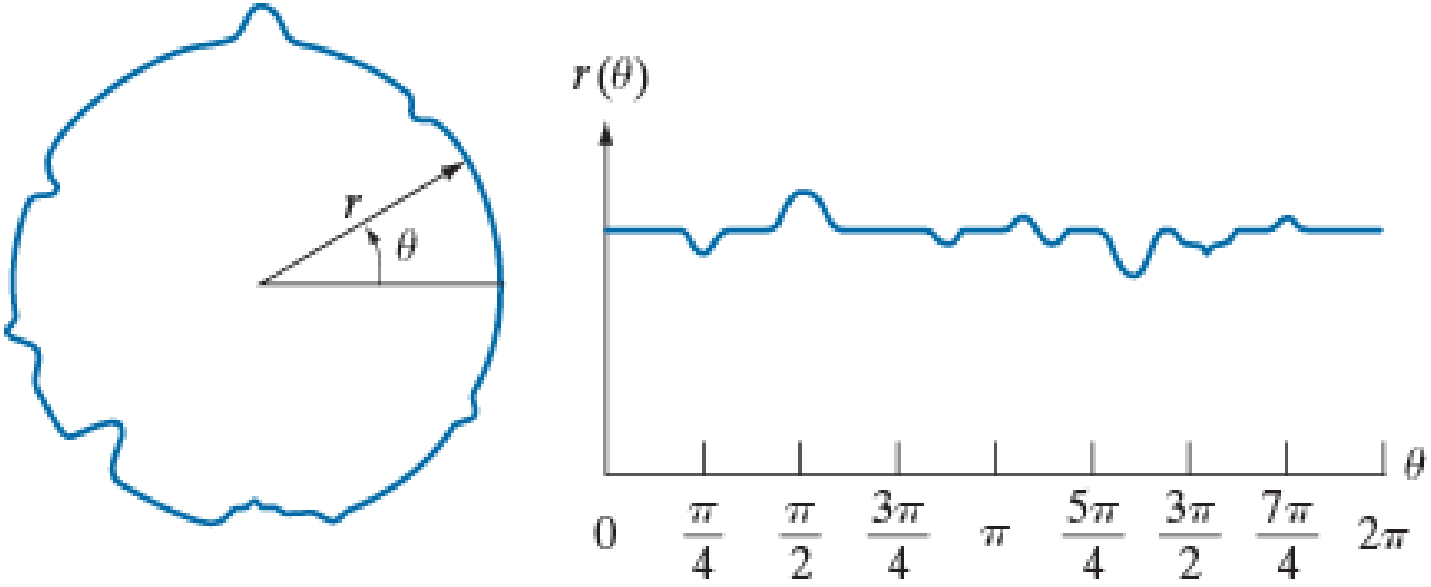
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

x_1 = Petal width
x_2 = Petal length
x_3 = Sepal width
x_4 = Sepal length

Other Examples

a b

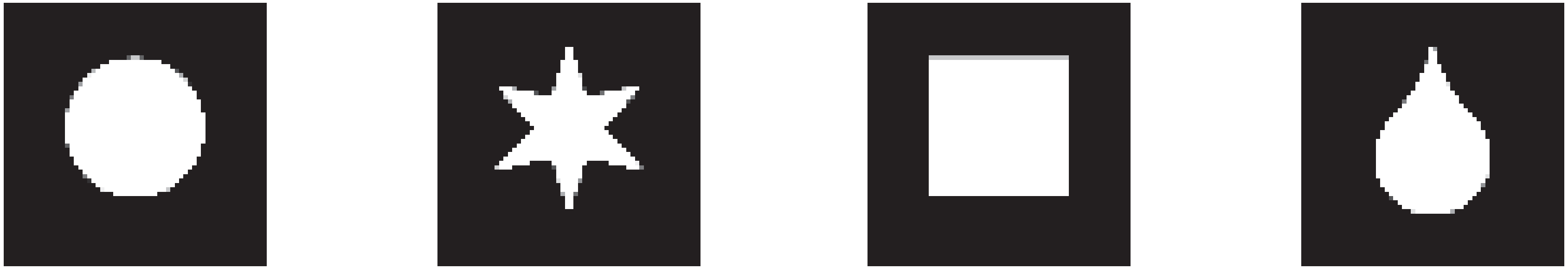
FIGURE 13.3
(a) A noisy object boundary, and (b) its corresponding signature.



$$\mathbf{x} = \begin{bmatrix} g\left(r(\theta_1)\right) \\ g\left(r(\theta_2)\right) \\ \vdots \\ g\left(r(\theta_n)\right) \end{bmatrix}$$

a b c d

FIGURE 13.4 Pattern vectors whose components capture both boundary and regional characteristics.



$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$x_1 =$ compactness
$x_2 =$ circularity
$x_3 =$ eccentricity

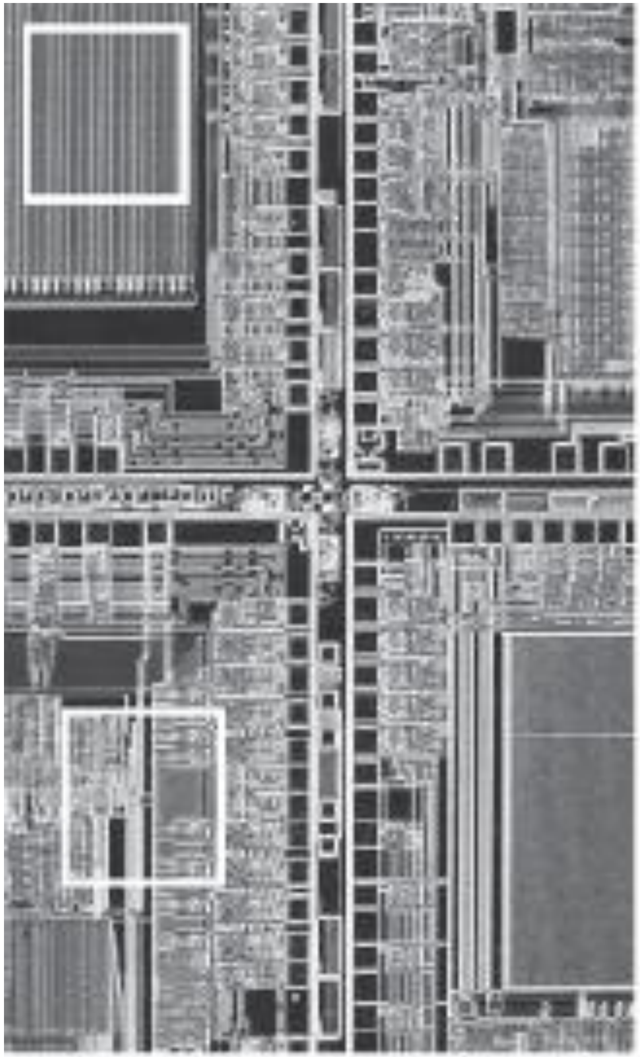


FIGURE 13.5 An example of pattern vectors based on properties of subimages. See Table 12.3 for an explanation of the components of **x**.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}$$

x_1 = max probability
x_2 = correlation
x_3 = contrast
x_4 = uniformity
x_5 = homogeneity
x_6 = entropy

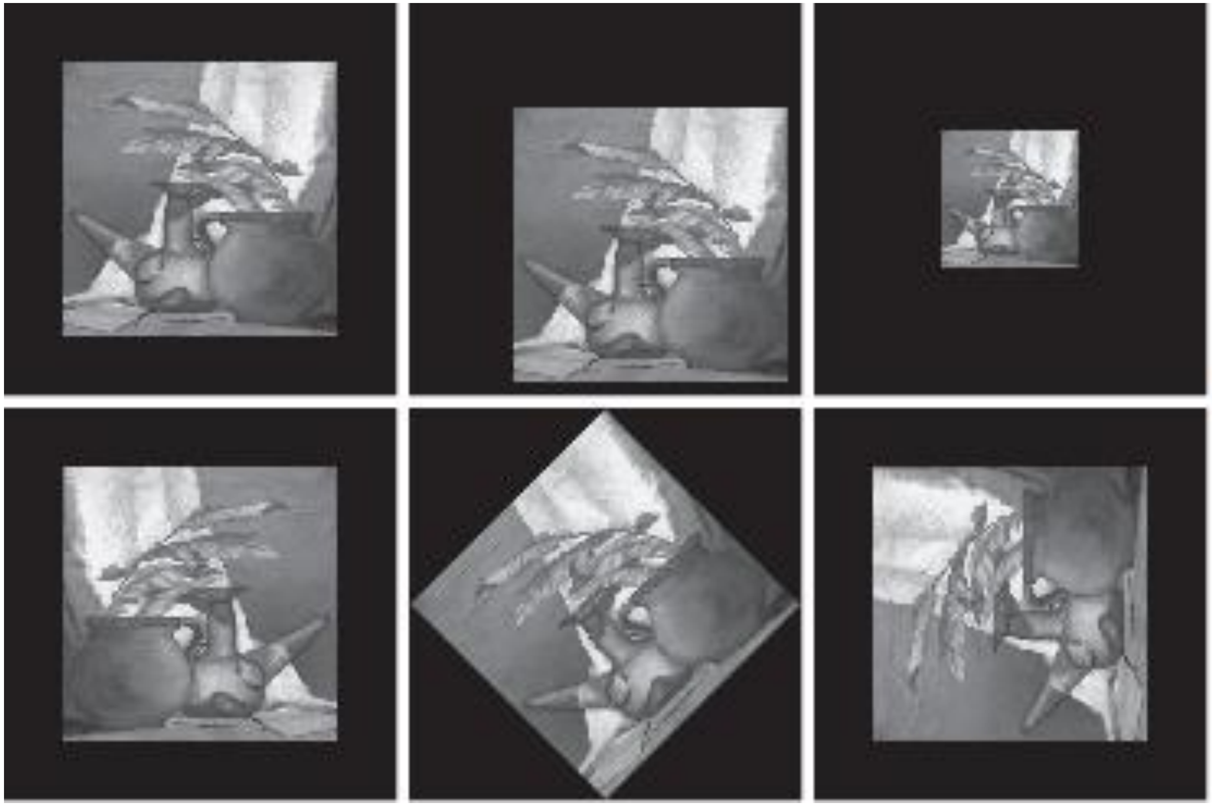


FIGURE 13.6 Feature vectors with components that are invariant to transformations such as rotation, scaling, and translation. The vector components are moment invariants.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \end{bmatrix}$$

The ϕ's are moment invariants

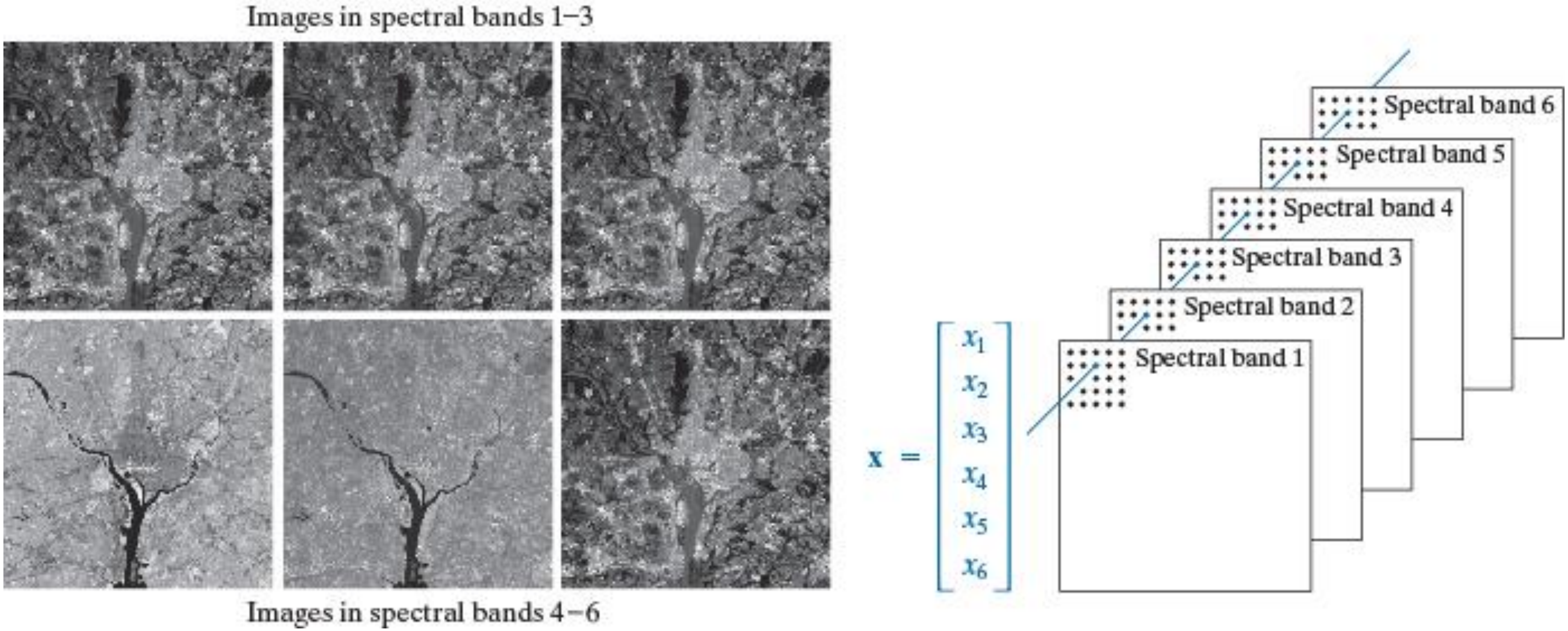


FIGURE 13.7 Pattern (feature) vectors formed by concatenating corresponding pixels from a set of registered images. (Original images courtesy of NASA.)

Structural Patterns for Shapes

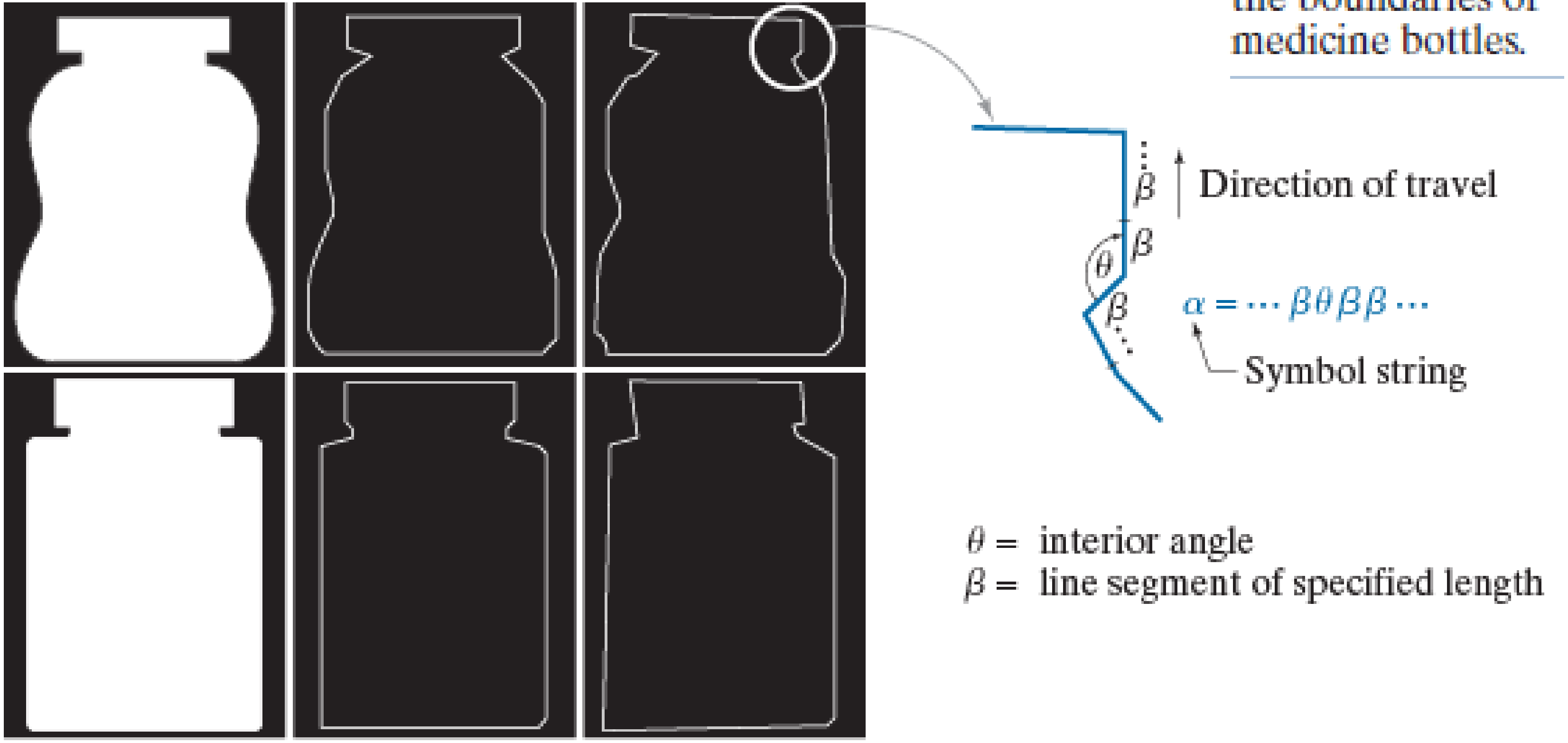

β
β
θ
β
Direction of travel
$\alpha = \cdots \beta\theta\beta\beta \cdots$
Symbol string
$\theta =$ interior angle
$\beta =$ line segment of specified length

FIGURE 13.8 Symbol string generated from a polygonal approximation of the boundaries of medicine bottles.

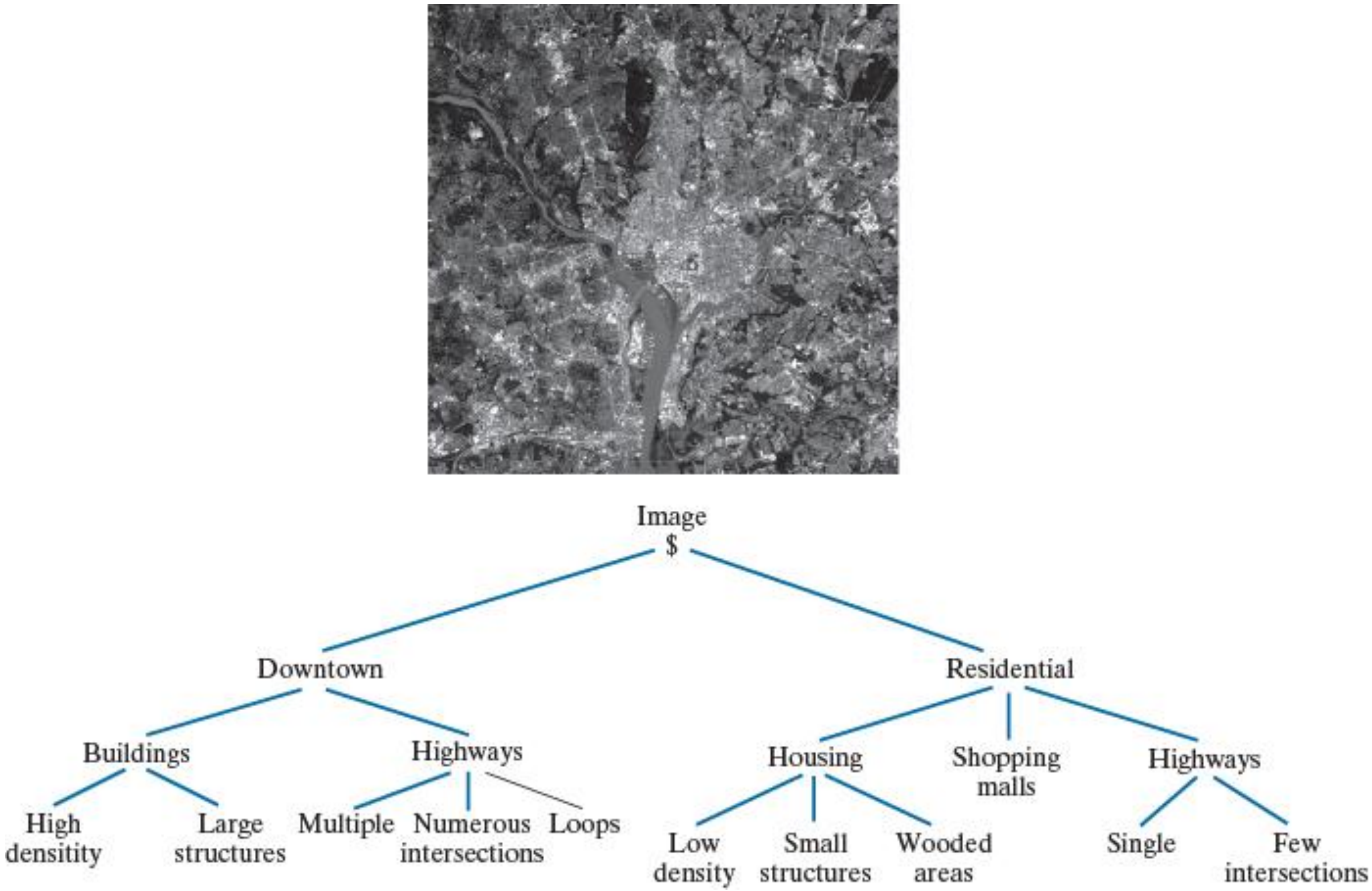


FIGURE 13.9 Tree representation of a satellite image showing a heavily built downtown area (Washington, D.C.) and surrounding residential areas. (Original image courtesy of NASA.)

Prototype Matching

- Minimum Distance Classifier
  - Compute a distance-based measure between an unknown pattern vector and each of the class prototypes.
  - The prototype vectors are the mean vectors of the various pattern classes

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x}_j \qquad j = 1, 2, \ldots, W$$

$$D_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{m}_j\| \qquad j = 1, 2, \ldots, W$$

$$\|\mathbf{a}\| = (\mathbf{a}^T \mathbf{a})^{1/2}$$ is the Euclidean Norm

  - Then assign the unknown pattern to the class of its closest prototype.

- It can be shown that it is equivalent to selecting a class that can maximize the following decision function:

$$d_j(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j \qquad j = 1, 2, \ldots, W$$

- The decision boundary between two classes:

$$\begin{aligned} d_{ij}(\mathbf{x}) &= d_i(\mathbf{x}) - d_j(\mathbf{x}) \\ &= \mathbf{x}^T(\mathbf{m}_i - \mathbf{m}_j) - \frac{1}{2}(\mathbf{m}_i - \mathbf{m}_j)^T(\mathbf{m}_i + \mathbf{m}_j) = 0 \end{aligned}$$

Illustration for Two Classes

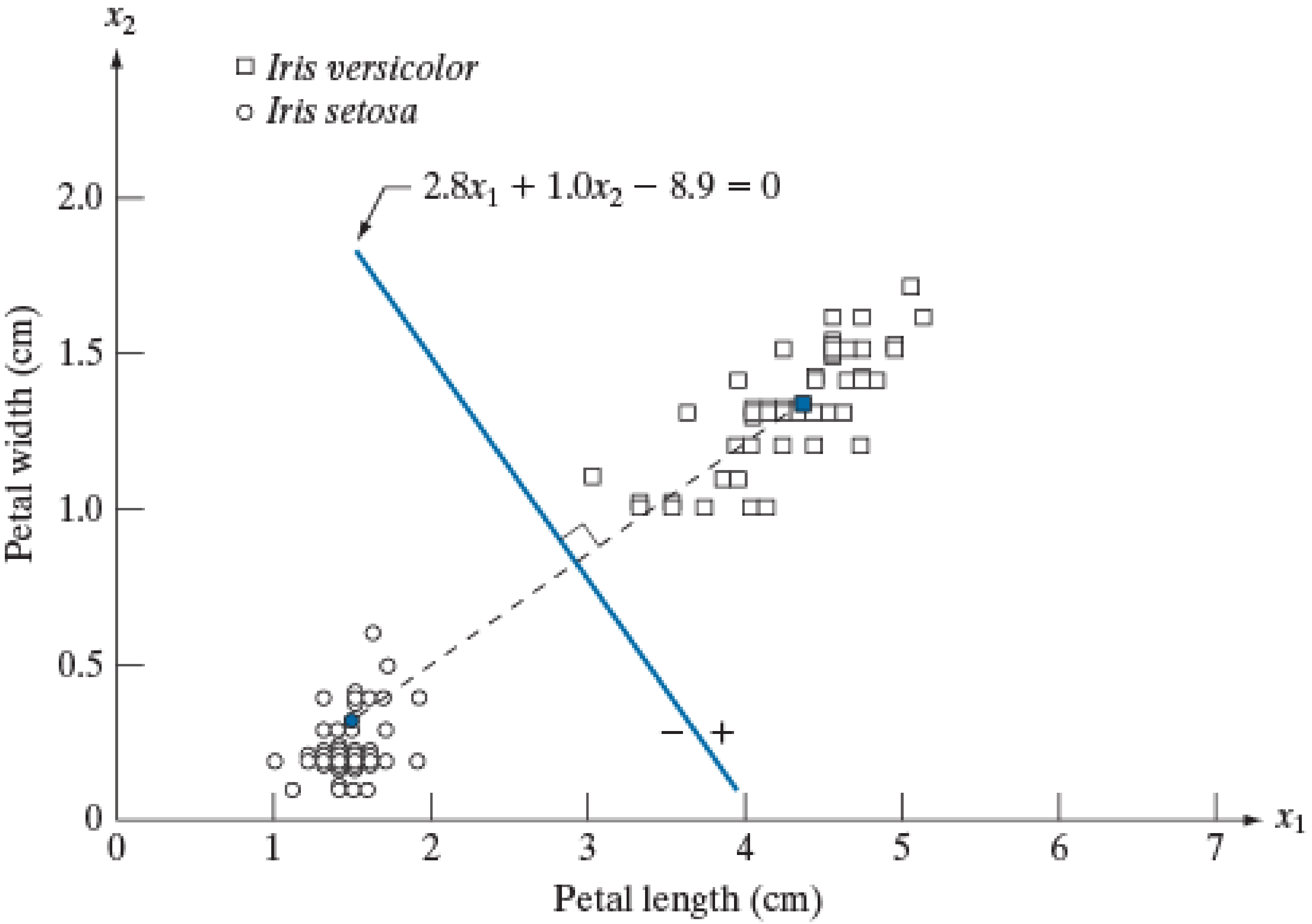


FIGURE 13.10 Decision boundary of a minimum distance classifier (based on two measurements) for the classes of Iris versicolor and Iris setosa. The dark dot and square are the means of the two classes.

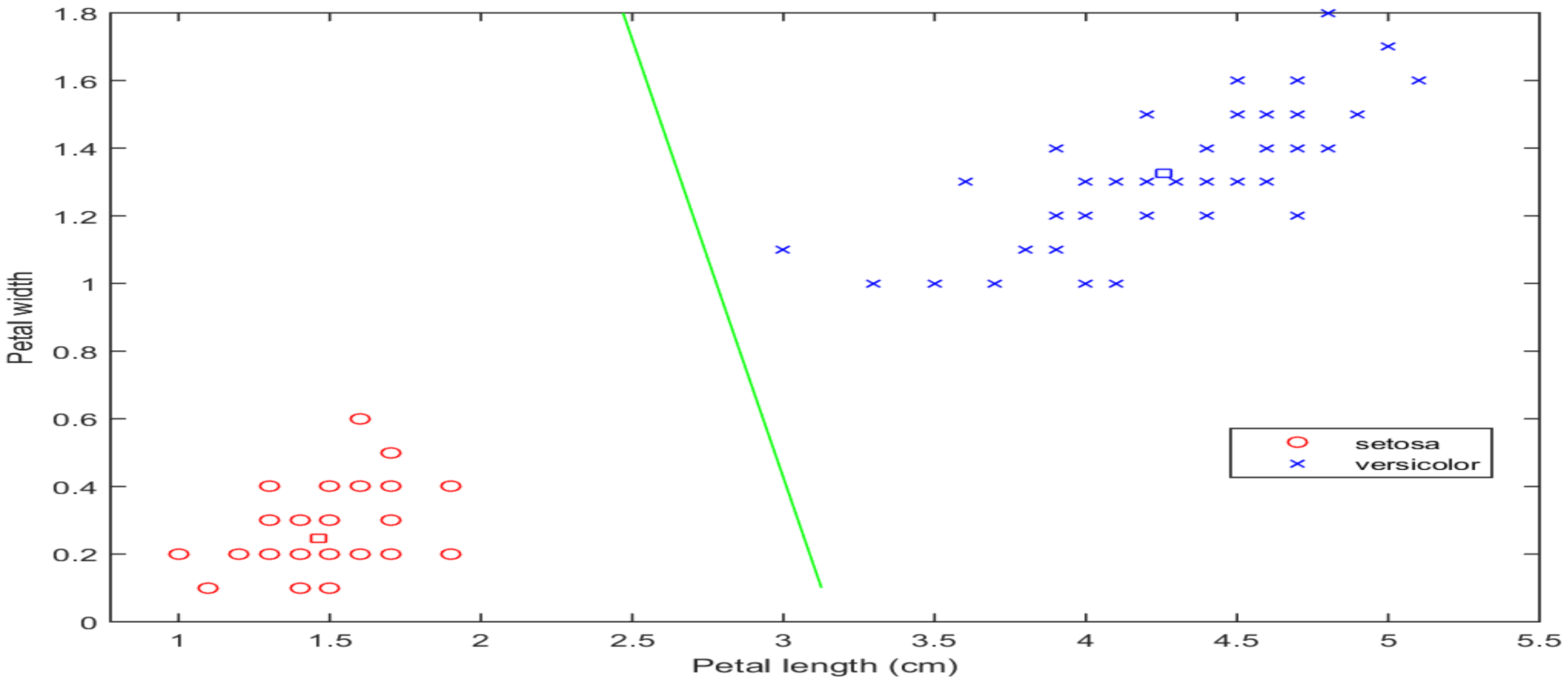
Detailed Derivations

$$\mathbf{m}_1 = (4.3, 1.3)^T \qquad \mathbf{m}_2 = (1.5, 0.3)^T.$$

$$\begin{aligned}
d_1(\mathbf{x}) &= \mathbf{x}^T\mathbf{m}_1 - \frac{1}{2}\mathbf{m}_1^T\mathbf{m}_1 \\
&= 4.3x_1 + 1.3x_2 - 10.1 \\
d_2(\mathbf{x}) &= \mathbf{x}^T\mathbf{m}_2 - \frac{1}{2}\mathbf{m}_2^T\mathbf{m}_2 \\
&= 1.5x_1 + 0.3x_2 - 1.17
\end{aligned}$$

$$\begin{aligned}
d_{12}(\mathbf{x}) &= d_1(\mathbf{x}) - d_2(\mathbf{x}) \\
&= 2.8x_1 + 1.0x_2 - 8.9 = 0
\end{aligned}$$

Matlab


1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Petal width
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5
Petal length (cm)
setosa
versicolor

Feature Design

- The minimum-distance classifier works well when the distance between means is large compared to the spread or randomness of each class with respect to its mean.
- We will show that the minimum-distance classifier yields optimum performance (in terms of minimizing the average loss of misclassification) when the distribution of each class about its mean is in the form of a spherical “hypercloud” in n-dimensional pattern space.
- One of the keys to accurate recognition performance is to specify features that are effective discriminators between classes.
- Systems based on the Banker’s Association E-13B font characters are example of how highly engineered features can be used in conjunction with a simple classifier to achieve superior results.

Magnetic Ink Character Recognition

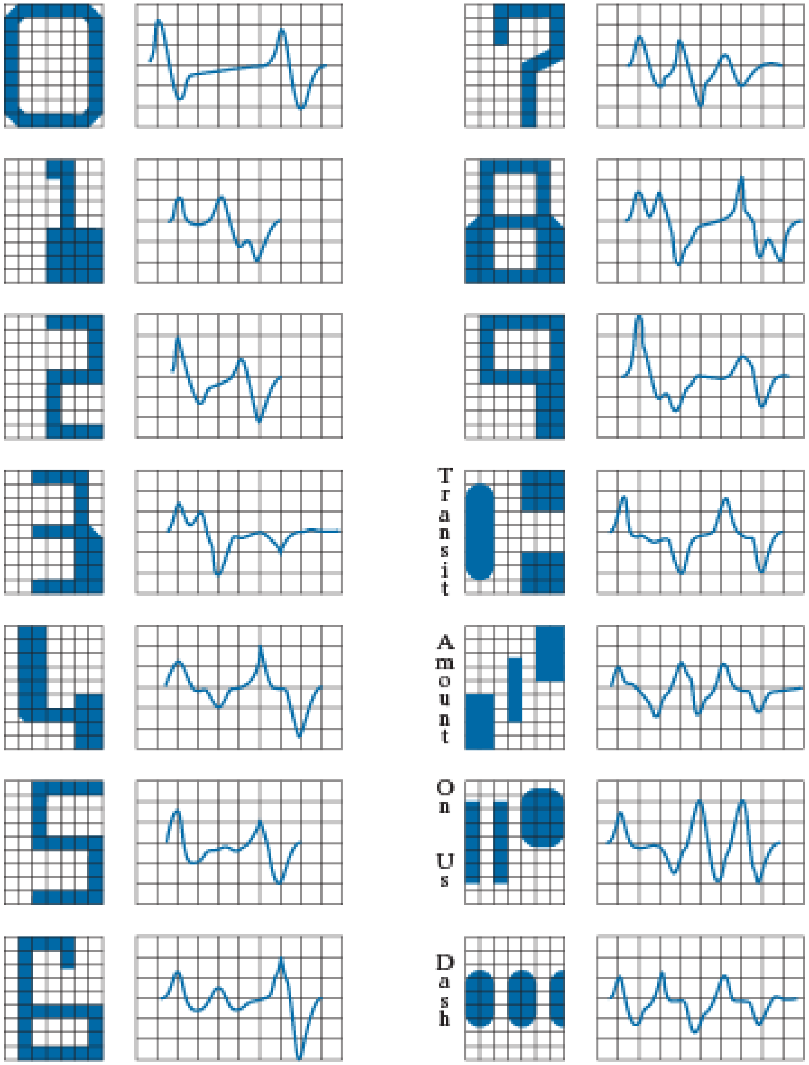


FIGURE 13.11 The American Bankers Association E-13B font character set and corresponding waveforms.

Matching by Correlation

Correlation Coefficient

Correlation of a kernel w with an image $f(x, y)$ is given by:

$$c(x, y) = \sum_{s} \sum_{t} w(s, t) f(x + s, y + t)$$

The kernel w is called the *template* (i.e., a prototype subimage)

We often perform matching using the *correlation coefficient* in order to avoid the sensitivity to changes in the amplitudes to either the kernel or the image pixels:

$$\gamma(x, y) = \frac{\sum_{s} \sum_{t} \left[w(s, t) - \overline{w} \right] \sum_{s} \sum_{t} \left[f(x + s, y + t) - \overline{f}(x + s, y + t) \right]}{\left\{ \sum_{s} \sum_{t} \left[w(s, t) - \overline{w} \right]^2 \sum_{s} \sum_{t} \left[f(x + s, y + t) - \overline{f}(x + s, y + t) \right]^2 \right\}^{\frac{1}{2}}}$$

Template Matching

- The correlation coefficient has values in the range of [-1, 1].

- Maximum correlation exists when the normalized template and the normalized sub-image have the best match

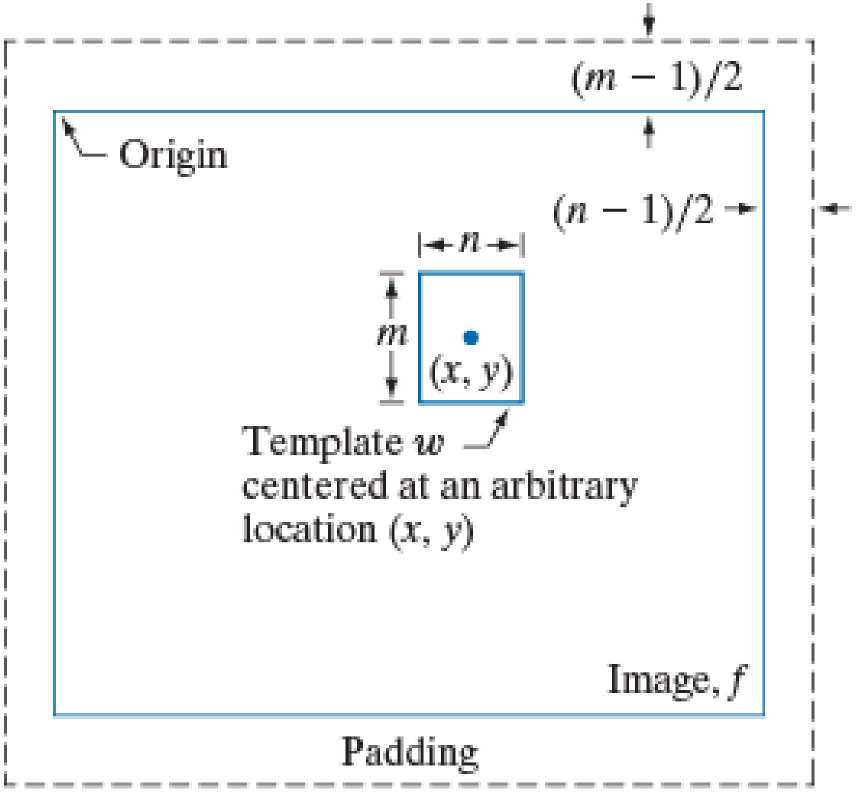
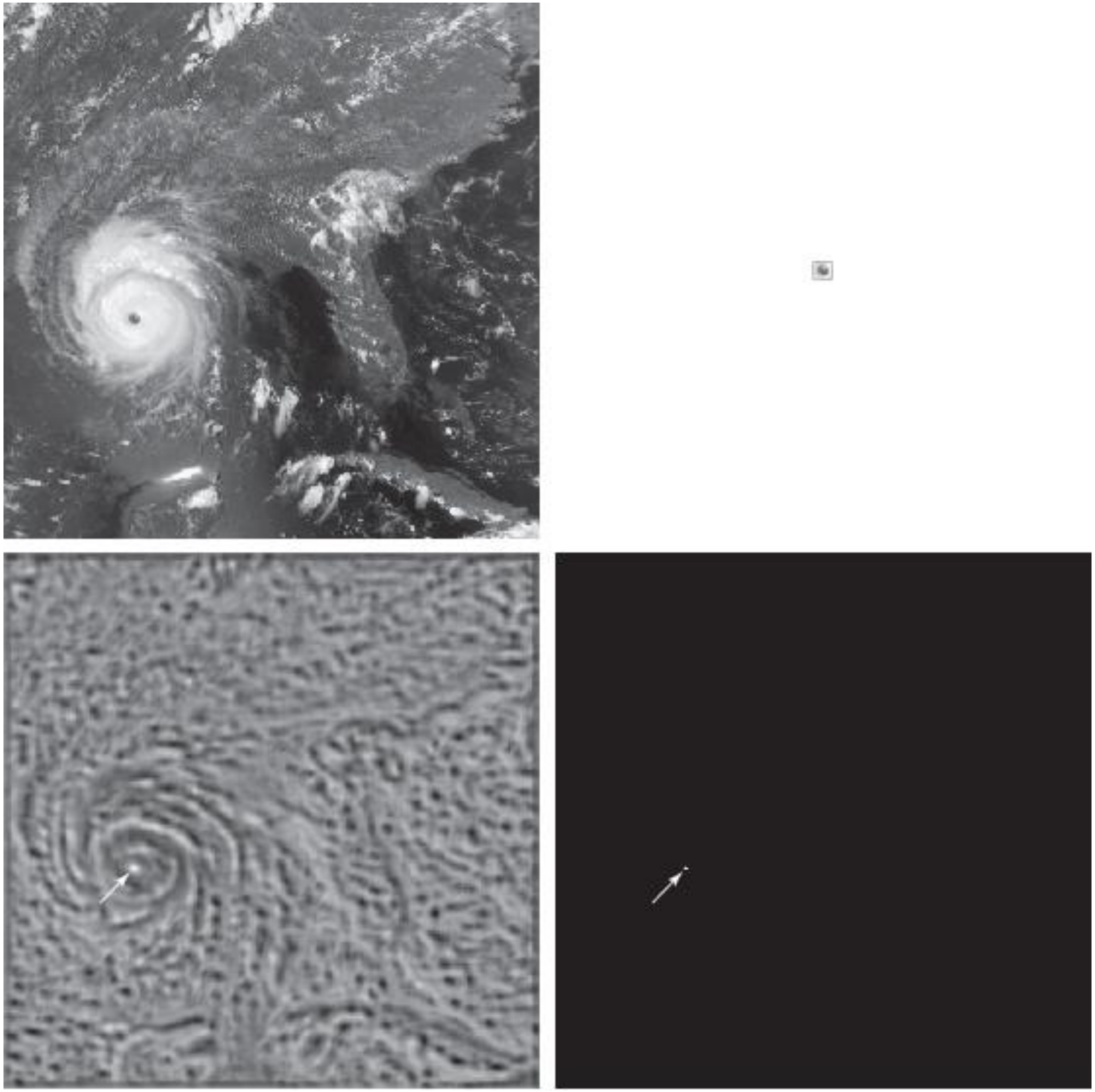


FIGURE 13.12 The mechanics of template matching.

a b
c d

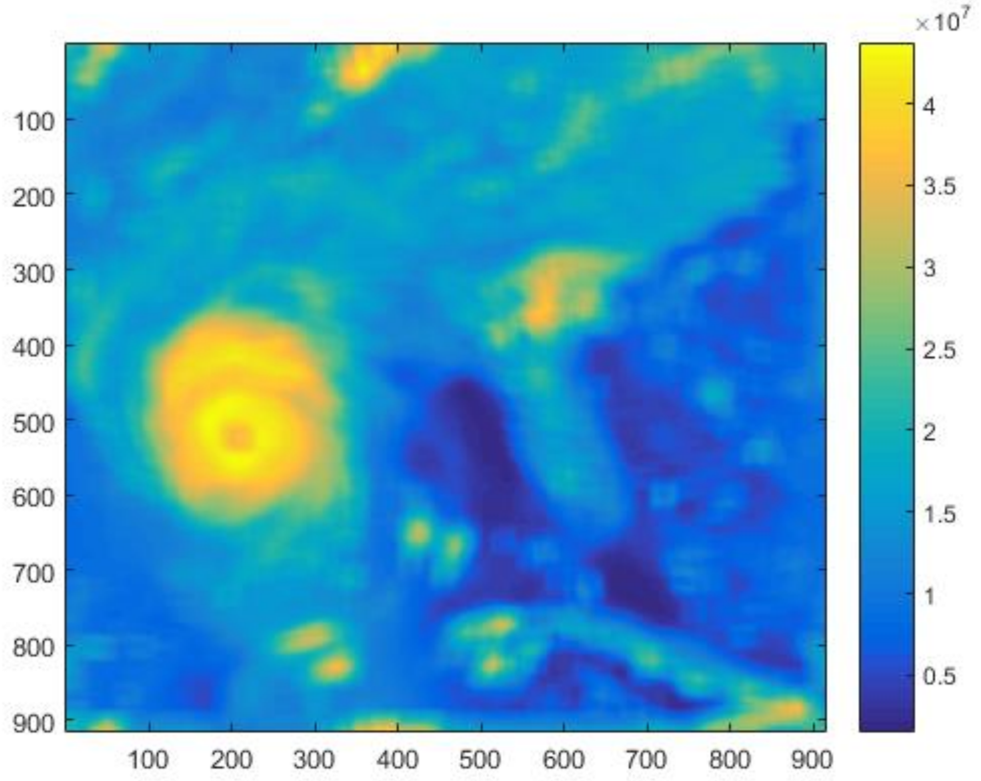
FIGURE 13.13
(a) 913 × 913 satellite image of Hurricane Andrew.
(b) 31 × 31 template of the eye of the storm.
(c) Correlation coefficient shown as an image (note the brightest point, indicated by an arrow).
(d) Location of the best match (identified by the arrow). This point is a single pixel, but its size was enlarged to make it easier to see.
(Original image courtesy of NOAA.)



Frequency Domain Processing

Spatial correlation can be obtained as the inverse Fourier transform of the product of the transform of one function times the conjugate of the transform of the other – More efficient computationally.

$$f(x, y) \circ w(x, y) \Leftrightarrow F(u, v)H^*(u, v)$$

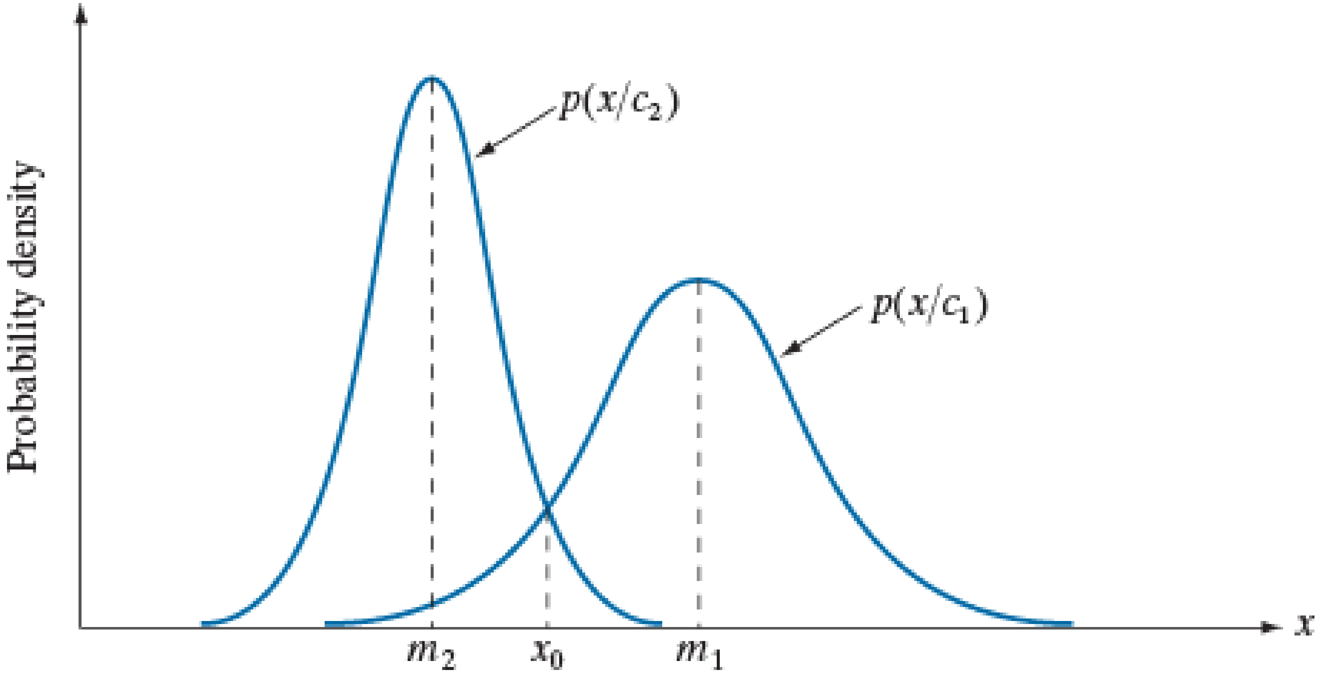


Optimal (Bayes) Statistical Classifier

Optimal Classification

- Probability considerations become important in pattern recognition because of the randomness under which pattern classes normally are generated.
- It is possible to derive a classification approach that is optimal in the sense that, on average, it yields the lowest probability of committing classification errors.

FIGURE 13.19 Probability density functions for two 1-D pattern classes. Point x_0 (at the intersection of the two curves) is the Bayes decision boundary if the two classes are equally likely to occur.



Conditional Probabilities and Bayes Theorem

- Joint Probability $P(A, B)$ for random events A and B.
- Conditional Probability $P(A|B) = \frac{P(A,B)}{P(B)}$. Similarly, $P(B|A) = \frac{P(A,B)}{P(A)}$
- If events A and B are independent, then $P(A, B) = P(A)P(B)$, implying that $P(B|A) = P(B)$ and $P(A|B) = P(A)$
- Example: Ice Cream
  70% of your friends like Chocolate, and 35% like Chocolate AND like Strawberry.
  Question: What percent of those who like Chocolate also like Strawberry?

  Answer:
  P(S|C) = P(C, S) / P(C) = 0.35/0.7 = 50%

Example

A noisy communication channel modeled by transition probabilities:

Given:

Binary source: $P(S0) + P(S1) = 1$

and the ***a priori*** probabilities: $P(R0|S0) + P(R1|S0) = 1$, $P(R0|S1) + P(R1|S1) = 1$

Question:

Determine $P(R0)$, $P(R1)$, and ***posterior*** probabilities $P(S0/R0)$, $P(S1/R1)$?

Answer:

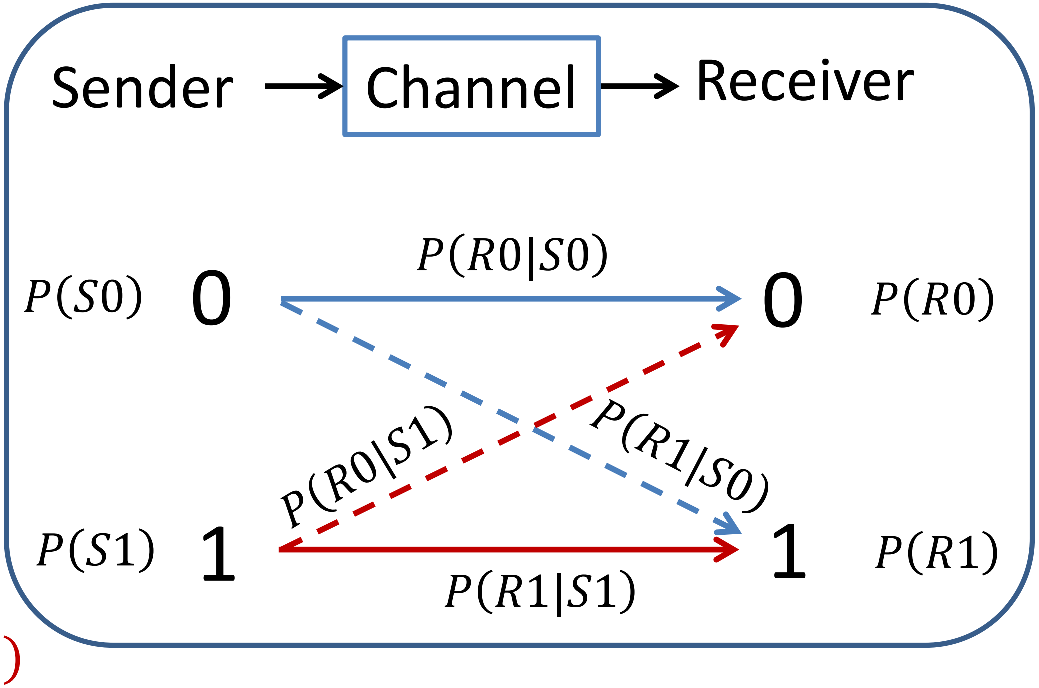
$$P(R0)$$
$$= P(R0, S0) + P(R0, S1)$$
$$= P(R0|S0)P(S0) + P(R0|S1)P(S1)$$

$$P(S0|R0)$$
$$= \frac{P(R0, S0)}{P(R0)} = \frac{P(R0|S0)P(S0)}{P(R0)}$$

Decision, given the same $P(R0)$:

Accept $R0$ if $P(S0|R0) > P(S1|R0)$,

or $P(R0|S0)P(S0) > P(R0|S1)P(S1)$



Bayes Classifier

- Given the prob. that a pattern vector $\boldsymbol{x}$ comes from class c_i is denoted by $p(c_i|x)$.
- If the pattern classifier decides that $\boldsymbol{x}$ came from class $c_j$when it actually came from c_i, it incurs a loss denoted by L_{ij}.
- Because the pattern vector $\boldsymbol{x}$ may belong to any one of N possible classes, the average loss incurred in assigning to class c_j is

$$r_j(\boldsymbol{x}) = \sum_{k=1}^{N} L_{kj} p(c_k|\boldsymbol{x})$$

which is called the *conditional average risk* in decision theory.

$$r_j(\boldsymbol{x}) = \sum_{k=1}^{N} L_{kj} p(c_k|\boldsymbol{x})$$

According to the Bayes Theorem

$$p(c_k|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|c_k)P(c_k)}{p(\boldsymbol{x})},$$

Therefore,

$$r_j(\boldsymbol{x}) = \frac{1}{p(\boldsymbol{x})} \sum_{k=1}^{N} L_{kj} p(\boldsymbol{x}|c_k) P(c_k)$$

where

$p(\boldsymbol{x}|c_k)$: PDF of the patterns from class c_k;
(*a priori* prob.)

$P(c_k)$: Prob. of occurrence of class c_k

Since $p(\boldsymbol{x})$ is a common term, we can rewrite $r_j(\boldsymbol{x})$ as

$$r_j(\boldsymbol{x}) = \sum_{k=1}^{N} L_{kj} p(\boldsymbol{x}|c_k) P(c_k)$$

The classifier that minimizes the total average loss Is called the **Bayes Classifier**,

where the classifier assigns an unknown pattern $\boldsymbol{x}$ to class c_i if $r_i(\boldsymbol{x}) < r_j(\boldsymbol{x})$ for $j = 1, 2, \dots, N; j \neq i$. That is

$$\sum_{k=1}^{N} L_{ki} p(\boldsymbol{x}|c_k) P(c_k) < \sum_{q=1}^{N} L_{qj} p\left(\boldsymbol{x}\middle|c_q\right) P\left(c_q\right)$$

If the loss for a correct decision is generally assigned a value of 0, and the loss for an incorrect decision is assigned a value of 1, then $L_{ij} = 1 - \delta_{ij}$.

Derivation of the Bayes Classifier

$$r_j(\boldsymbol{x}) = \sum_{k=1}^{N} L_{kj} p(\boldsymbol{x}|c_k) P(c_k) \quad \text{and} \quad L_{kj} = 1 - \delta_{kj}$$

$$r_j(\boldsymbol{x}) = \sum_{k=1}^{N} (1 - \delta_{ij}) p(\boldsymbol{x}|c_k) P(c_k)$$

$$= \sum_{k=1}^{N} p(\boldsymbol{x}|c_k) P(c_k) - \sum_{k=1}^{N} \delta_{ij} p(\boldsymbol{x}|c_k) P(c_k)$$

$$= p(\boldsymbol{x}) - p\left(\boldsymbol{x}|c_j\right) P\left(c_j\right)$$

Similarly,

$$r_i(\boldsymbol{x}) = p(\boldsymbol{x}) - p(\boldsymbol{x}|c_i) P(c_i)$$

Decision Rule

- classifier assigns an unknown pattern $\boldsymbol{x}$ to class c_i if
  $r_i(\boldsymbol{x}) < r_j(\boldsymbol{x})$ for $j = 1, 2, \dots, N; j \neq i$.

$$p(\boldsymbol{x}) - p(\boldsymbol{x}|c_i)P(c_i) < p(\boldsymbol{x}) - p(\boldsymbol{x}|c_j)P(c_j),$$

or equivalently

$$\boxed{p(\boldsymbol{x}|c_i)P(c_i) > p(\boldsymbol{x}|c_j)P(c_j)}$$

Decision Function

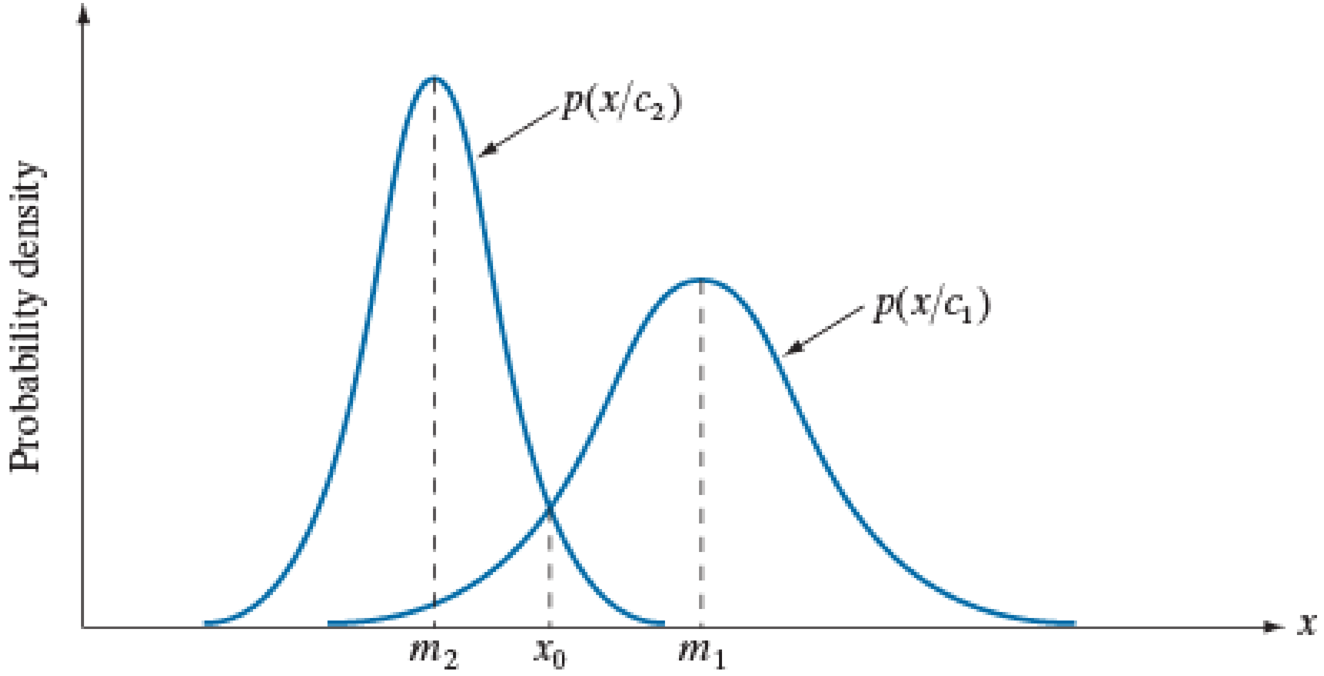
- The Bayes Classifier for a 0-1 loss function computes the decision function

  $$\boxed{d_j(\boldsymbol{x}) = p(\boldsymbol{x}|c_i)P(c_i)}$$

  for $j = 1, 2, \dots, N$ and assign a pattern $\boldsymbol{x}$ to class c_i if $d_i(\boldsymbol{x}) > d_j(\boldsymbol{x})$, for all $j \neq i$.

- For the optimality of Bayes decision function to hold, the *a priori* probability $p(\boldsymbol{x}|c_i)$ and the class probability $P(c_i)$ needs to be known or estimated from sample patterns during training.
- Usually assume Gaussian Distribution for $p(\boldsymbol{x}|c_i)$.

Gaussian Pattern Classes



$$d_j(x) = p(x|c_j)P(c_j) = \frac{1}{\sqrt{2\pi}\sigma_j} e^{-\frac{(x-m_j)^2}{2\sigma_j^2}} P(c_j)$$

where $\quad j = 1, 2$

n-Dimensional Gaussian PDF

$$p(\mathbf{x}/\omega_j) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}_j|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_j)^T \mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)}$$

where the mean vector is $\mathbf{m}_j = E_j\{\mathbf{x}\}$

and the covariance matrix is

$$\mathbf{C}_j = E_j\{(\mathbf{x} - \mathbf{m}_j)(\mathbf{x} - \mathbf{m}_j)^T\}$$

We can approximate with taking the averages of sample vectors:

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x}\in\omega_j} \mathbf{x} \qquad \mathbf{C}_j = \frac{1}{N_j} \sum_{\mathbf{x}\in\omega_j} \mathbf{x}\mathbf{x}^T - \mathbf{m}_j \mathbf{m}_j^T$$

Logarithm of the Decision Function

$$d_j(\mathbf{x}) = \ln\left[p(\mathbf{x}/\omega_j)P(\omega_j)\right] = \ln p(\mathbf{x}/\omega_j) + \ln P(\omega_j)$$

$$p(\mathbf{x}/\omega_j) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}_j|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x}-\mathbf{m}_j)}$$

$$d_j(\mathbf{x}) = \ln P(\omega_j) \boxed{- \frac{n}{2}\ln 2\pi} - \frac{1}{2}\ln|\mathbf{C}_j| - \frac{1}{2}\left[(\mathbf{x} - \mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x} - \mathbf{m}_j)\right]$$

$$d_j(\mathbf{x}) = \ln P(\omega_j) - \frac{1}{2}\ln|\mathbf{C}_j| - \frac{1}{2}\left[(\mathbf{x} - \mathbf{m}_j)^T\mathbf{C}_j^{-1}(\mathbf{x} - \mathbf{m}_j)\right]$$

- If the covariance matrix is identical. then

$$d_j(\mathbf{x}) = \ln P(\omega_j) + \mathbf{x}^T\mathbf{C}^{-1}\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{C}^{-1}\mathbf{m}_j$$

- If all classes are equally likely and the covariance matrix is an identity matrix, then

$$d_j(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_j - \frac{1}{2}\mathbf{m}_j^T\mathbf{m}_j \quad j = 1, 2, \ldots, W$$

- The same decision function for a minimum-distance classifier, which is optimal in the Bayes sense if
  - The pattern classes are Gaussian.
  - All covariance matrices are equal to identity matrix.
  - All classes are equally likely.

Example

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \quad m_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, m_2 = \begin{bmatrix} 9 \\ 9 \end{bmatrix},$$

$$C_1 = C_2 = C = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \qquad C^{-1} = \tfrac{1}{3}\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$$

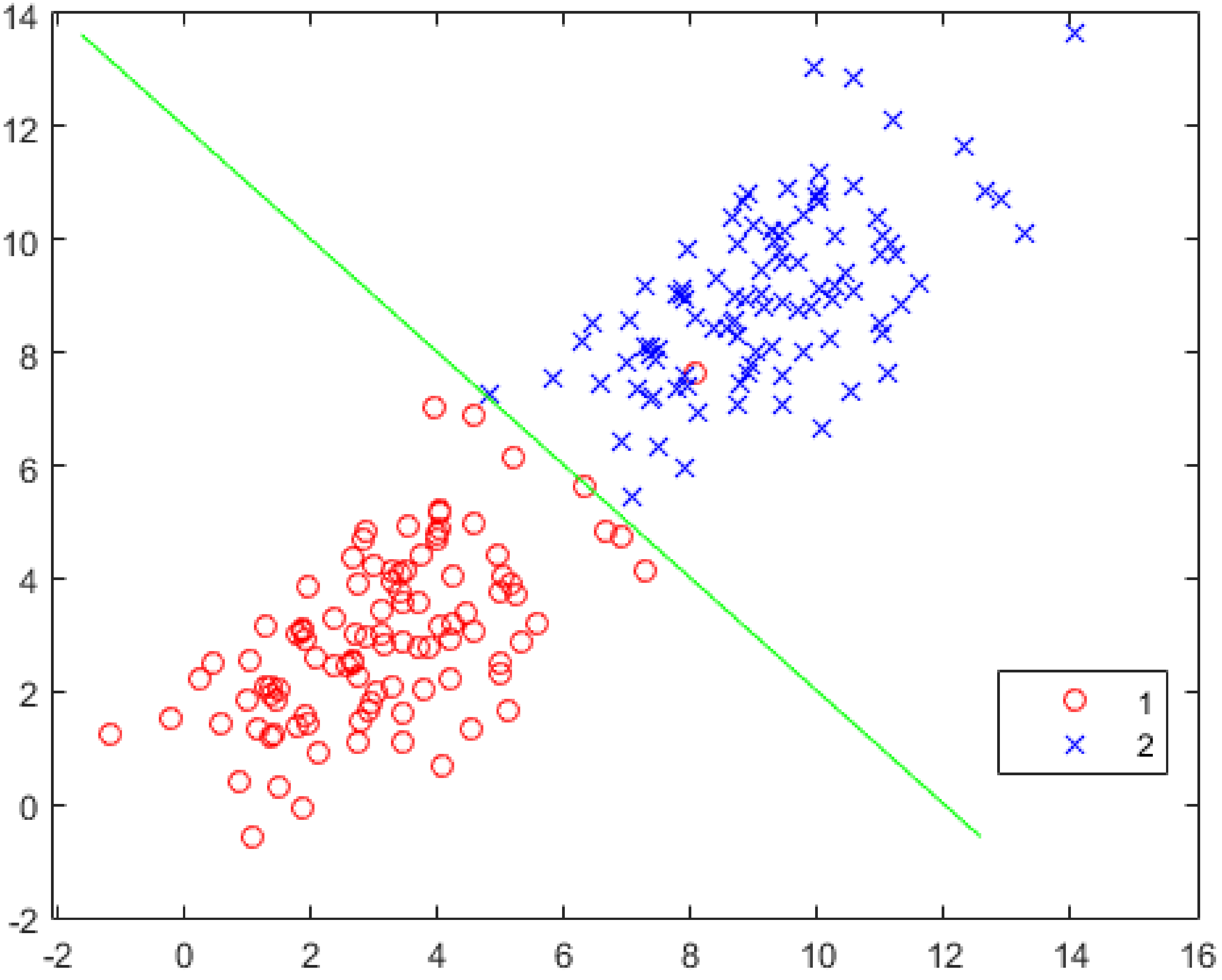
$$d_j(\boldsymbol{x}) = \boldsymbol{x}^T C^{-1} m_j - \tfrac{1}{2} m_j^T C^{-1} m_j$$

$d_1(\boldsymbol{x}) = x_1 + x_2 - 3$ and

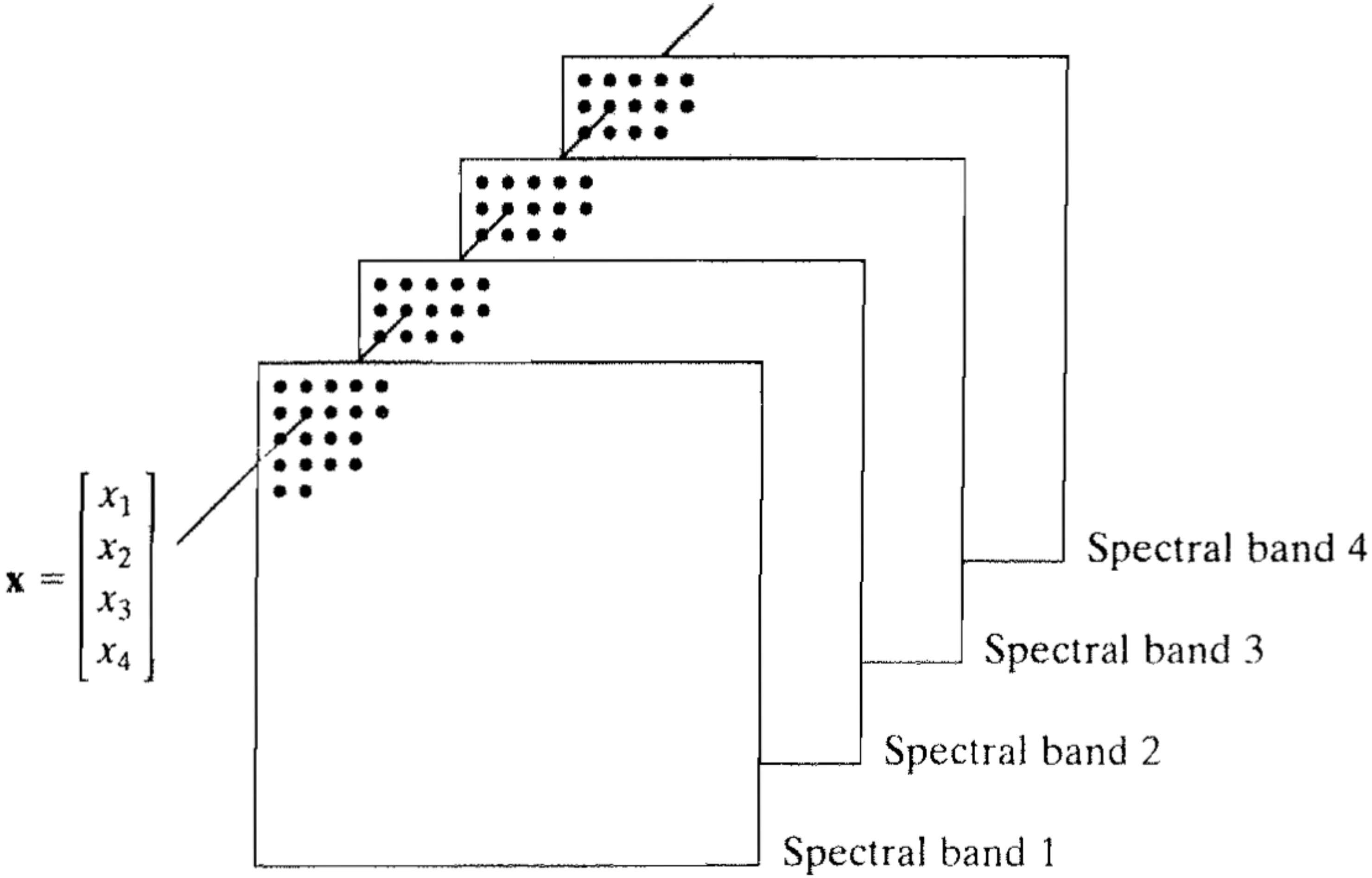
$d_2(\boldsymbol{x}) = 3x_1 + 3x_2 - 27$

The decision boundary is

$$d_2(\boldsymbol{x}) - d_1(x) = x_1 + x_2 - 12 = 0$$


14
12
10
8
6
4
2
0
-2
-2
0
2
4
6
8
10
12
14
16
○ 1
× 2

Multispectral Image Classification



$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$
Spectral band 4
Spectral band 3
Spectral band 2
Spectral band 1

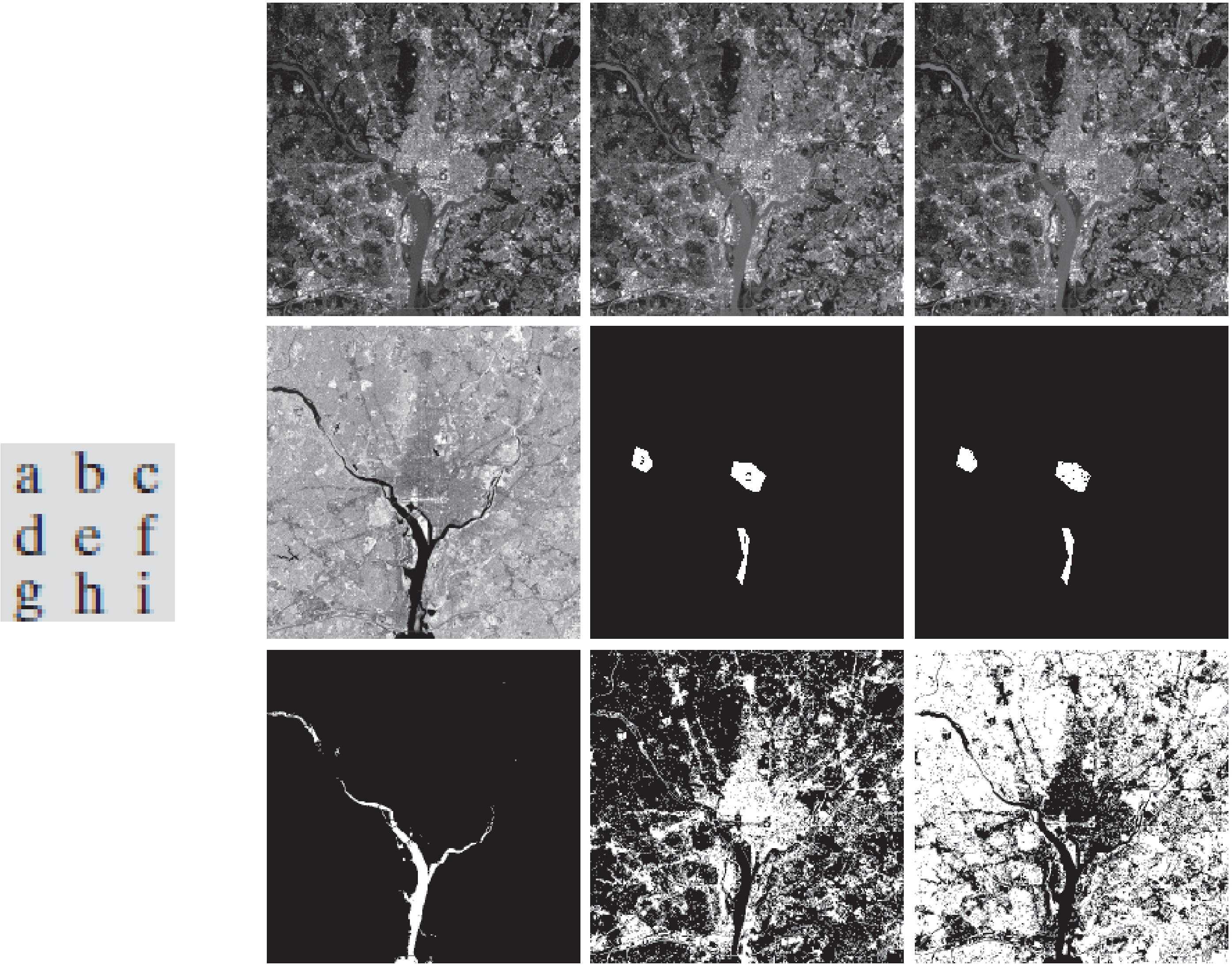


FIGURE 13.21 Bayes classification of multispectral data. (a)–(d) Images in the visible blue, visible green, visible red, and near infrared wavelength bands. (e) Masks for regions of water (labeled 1), urban development (labeled 2), and vegetation (labeled 3). (f) Results of classification; the black dots denote points classified incorrectly. The other (white) points were classified correctly. (g) All image pixels classified as water (in white). (h) All image pixels classified as urban development (in white). (i) All image pixels classified as vegetation (in white).

Accuracy of Classification

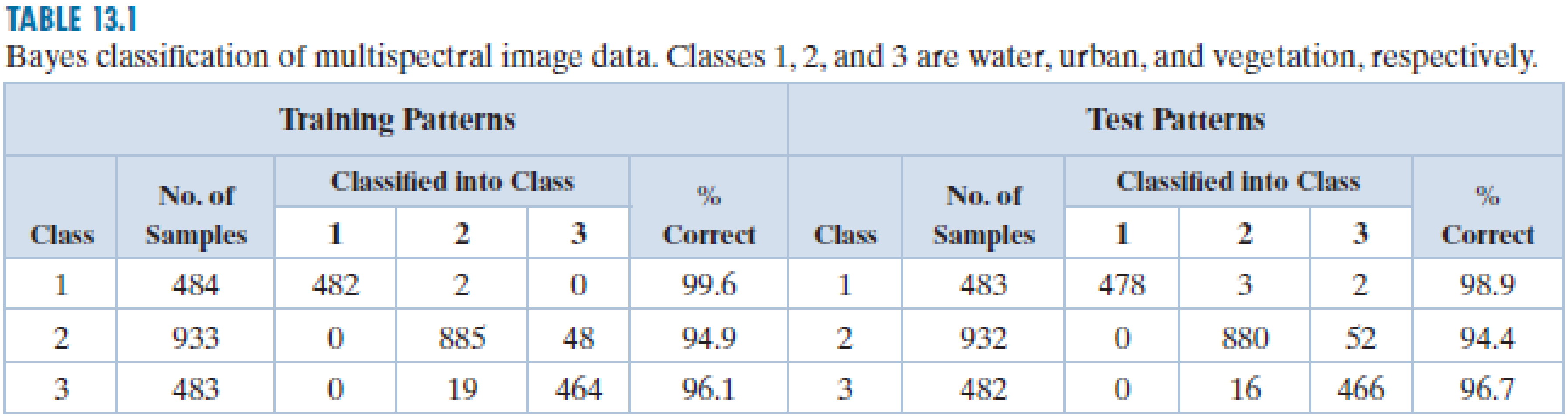


TABLE 13.1
Bayes classification of multispectral image data. Classes 1, 2, and 3 are water, urban, and vegetation, respectively.

Training Patterns						Test Patterns					
Class	No. of Samples	Classified into Class			% Correct	Class	No. of Samples	Classified into Class			% Correct
		1	2	3				1	2	3	
1	484	482	2	0	99.6	1	483	478	3	2	98.9
2	933	0	885	48	94.9	2	932	0	880	52	94.4
3	483	0	19	464	96.1	3	482	0	16	466	96.7

- Regions of interest with known labels are called *Ground Truth*.
- Half of the samples are used for training (i.e., for estimating the mean vectors and covariance matrices for 4-D pattern vector.
- The other half for independent testing to asses classifier performance.
- Assume equal class probabilities.

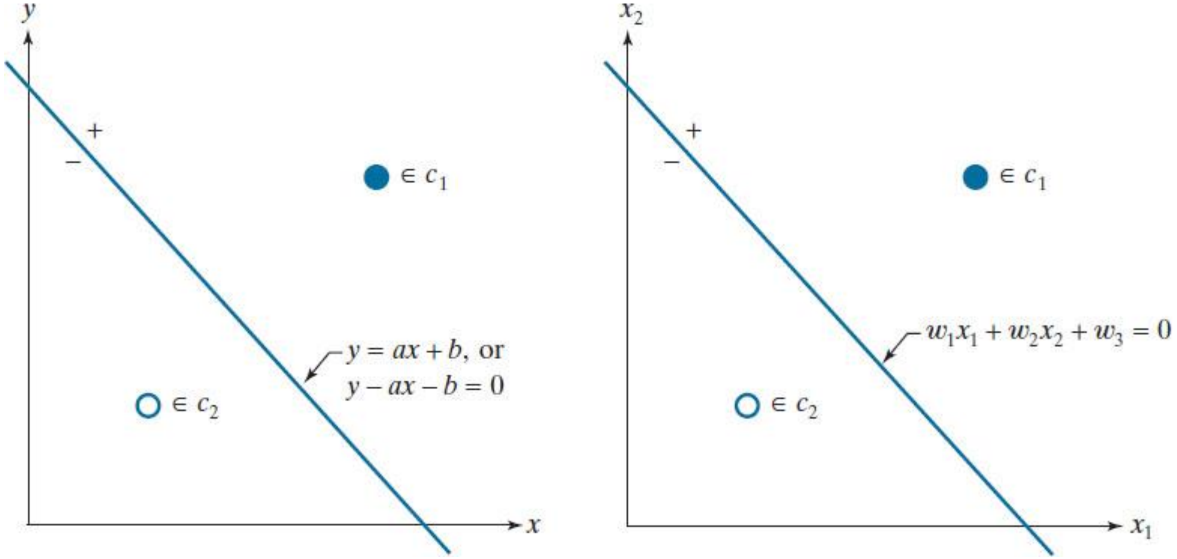
Neural Networks

Learning Machines

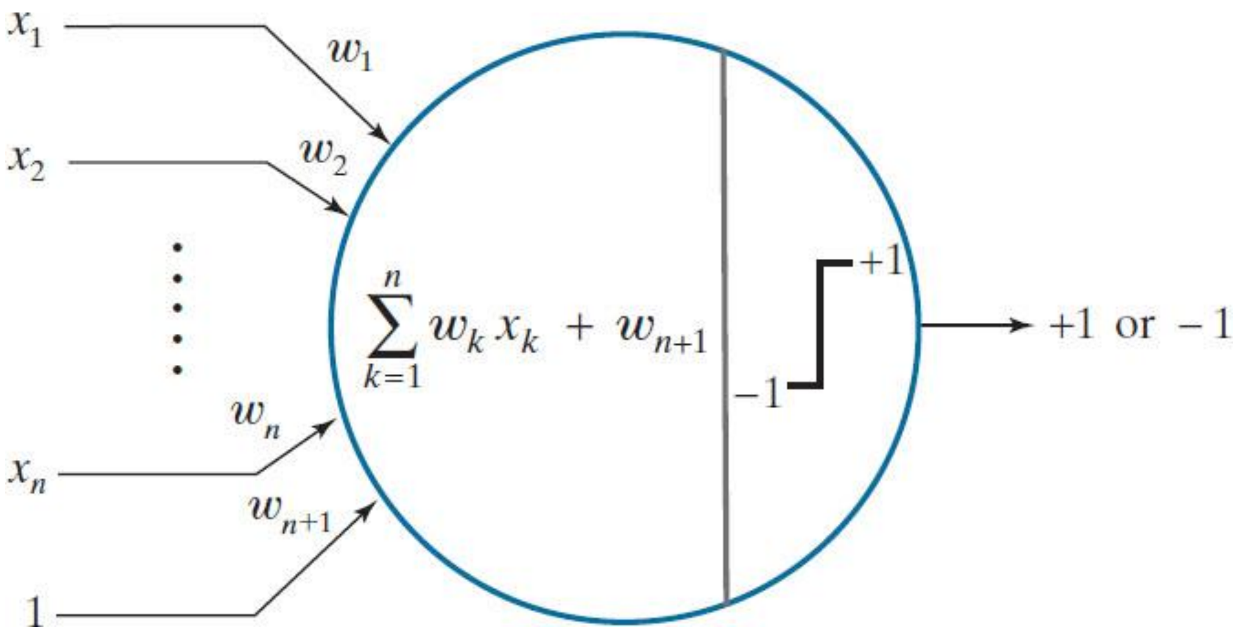
- Use of a multitude of elemental nonlinear computing elements (called artificial *neurons*), organized as networks, whose interconnections are similar to the way in which neurons are interconnected in the visual cortex of mammals.
- The resulting models are called *neural networks*.
- We use neural networks as a machine to adaptively learn the parameters of decision function via successive presentations of training patterns.
- **Perceptron** is such a simple learning machine.
- The perceptron convergence theorem states that the algorithm is guaranteed to converge to a solution in a finite number of steps if the two pattern classes are linearly separable.

Perceptron

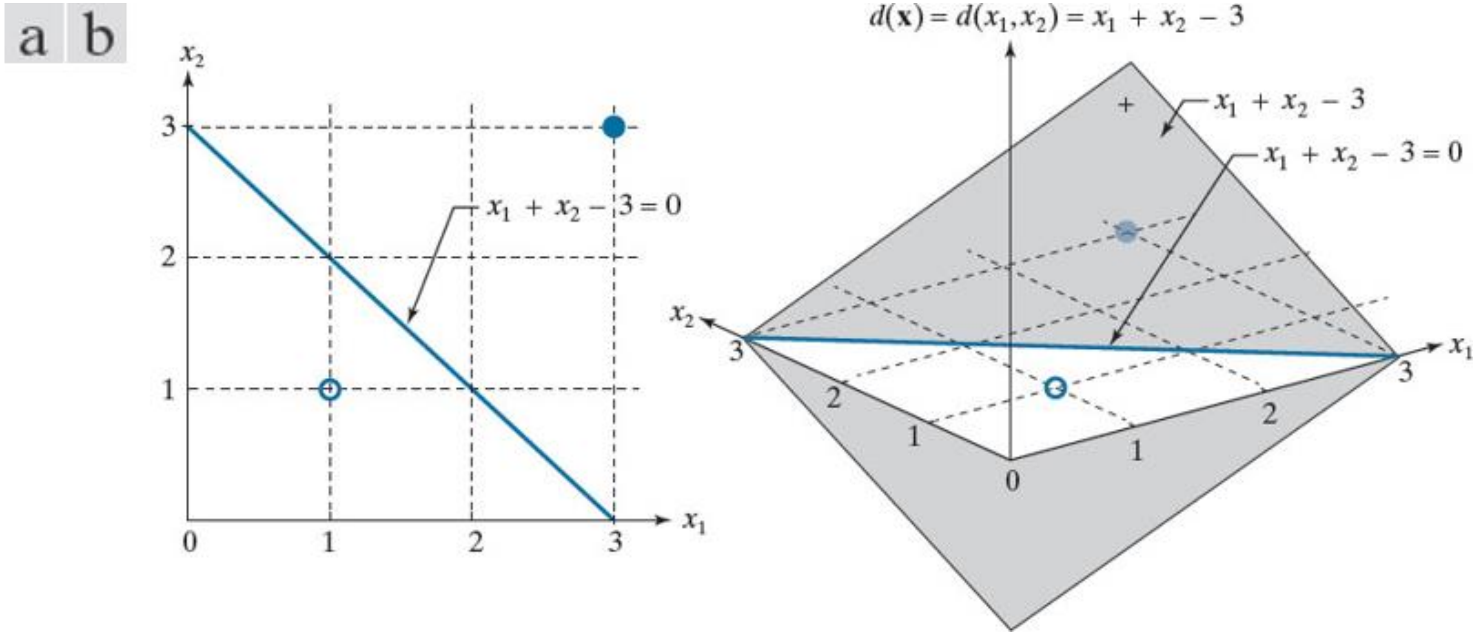
- A single perceptron learns a linear boundary between two linearly separable pattern classes.



(a) The simplest two-class example in 2-D, showing one possible decision boundary out of an infinite number of such boundaries. (b) Same as (a), but with the decision boundary expressed using more general notation.

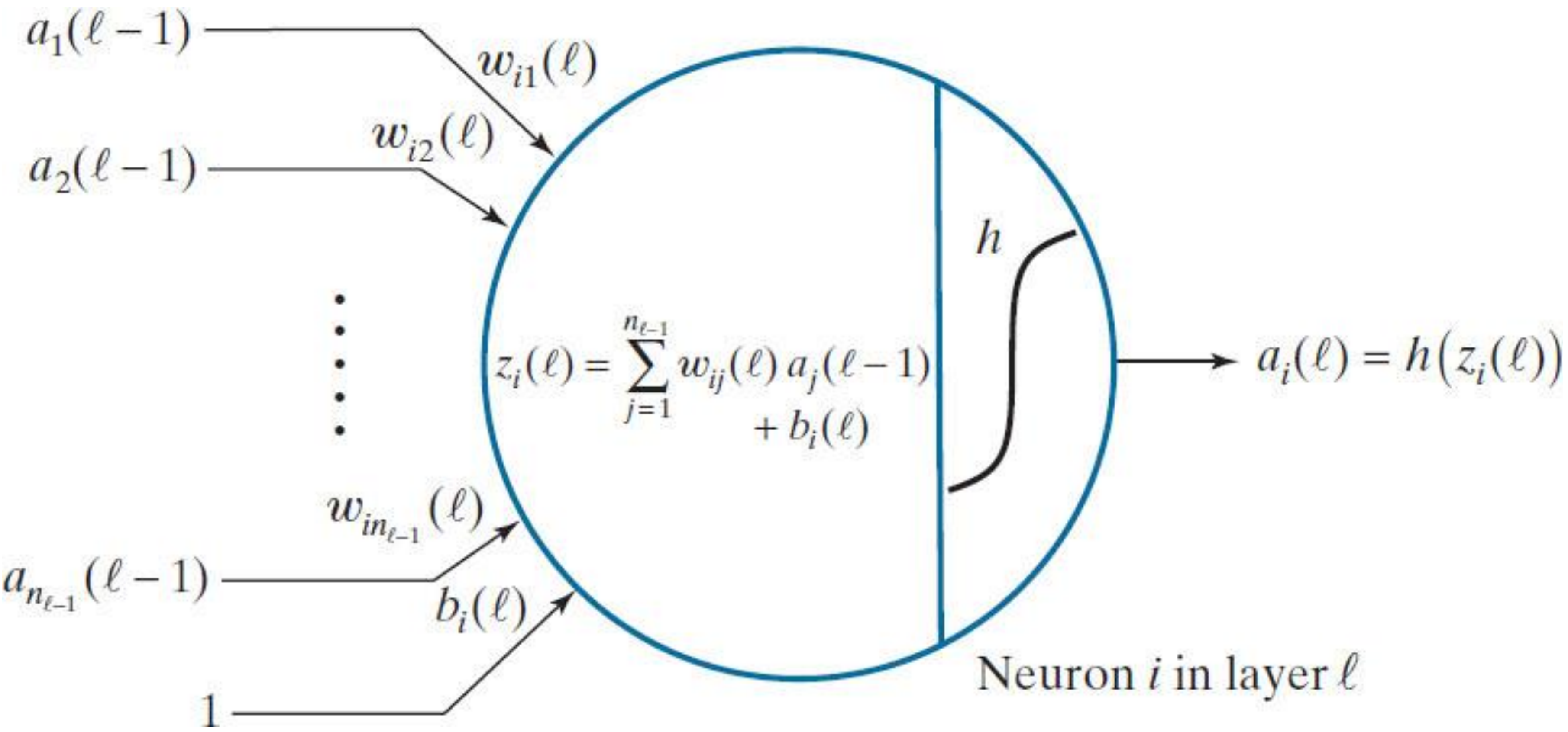


Schematic of a perceptron, showing the operations it performs.

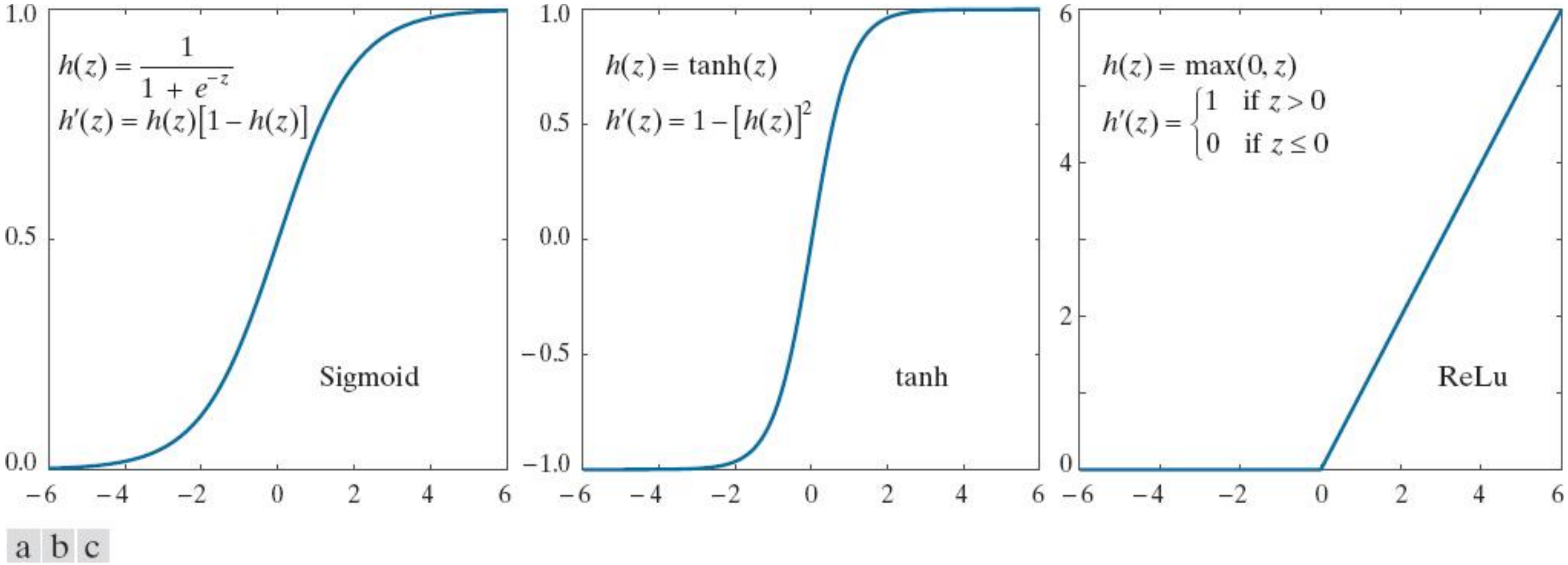


(a) Segment of the decision boundary learned by the perceptron algorithm. (b) Section of the decision surface. The decision boundary is the intersection of the decision surface with the

Model of An Artificial Neuron

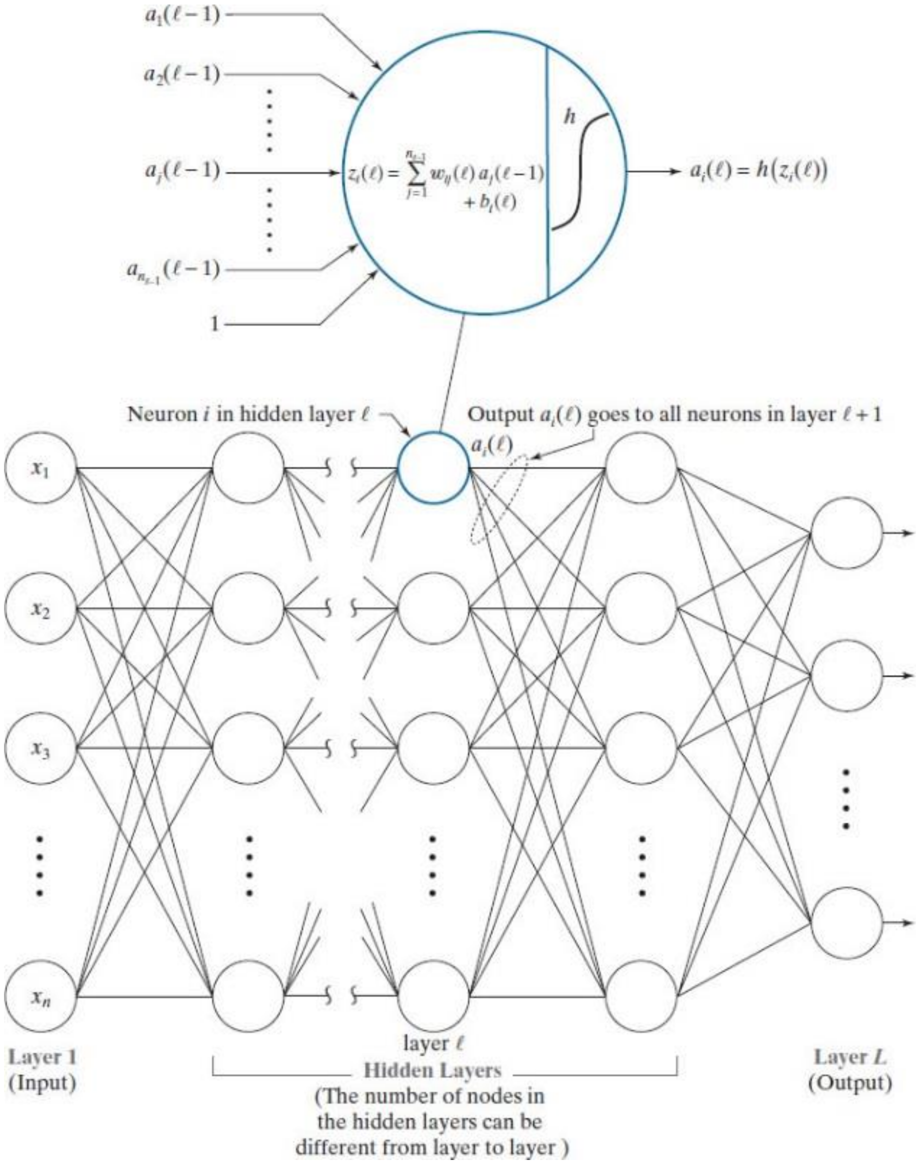

$a_1(\ell-1)$
$w_{i1}(\ell)$
$w_{i2}(\ell)$
$a_2(\ell-1)$
h
$z_i(\ell) = \sum_{j=1}^{n_{\ell-1}} w_{ij}(\ell)\, a_j(\ell-1) + b_i(\ell)$
$a_i(\ell) = h\big(z_i(\ell)\big)$
$w_{in_{\ell-1}}(\ell)$
$a_{n_{\ell-1}}(\ell-1)$
$b_i(\ell)$
1
Neuron i in layer ℓ

Various Activation Functions



(a) Sigmoid. (b) Hyperbolic tangent (also has a sigmoid shape, but it is centered about 0 in both dimensions). (c) Rectifier linear unit (ReLU).

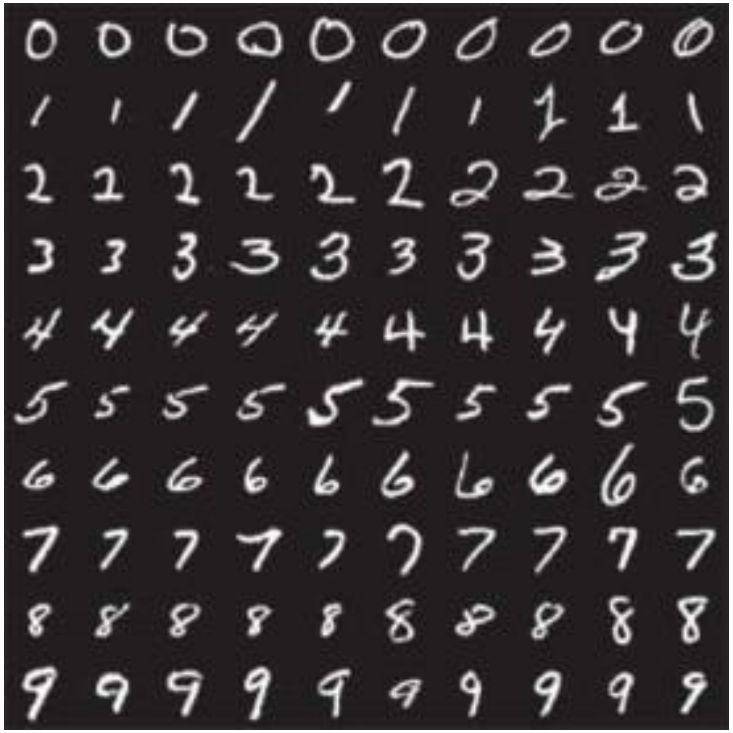
Model of a Feedforward, Fully Connected Neural Network



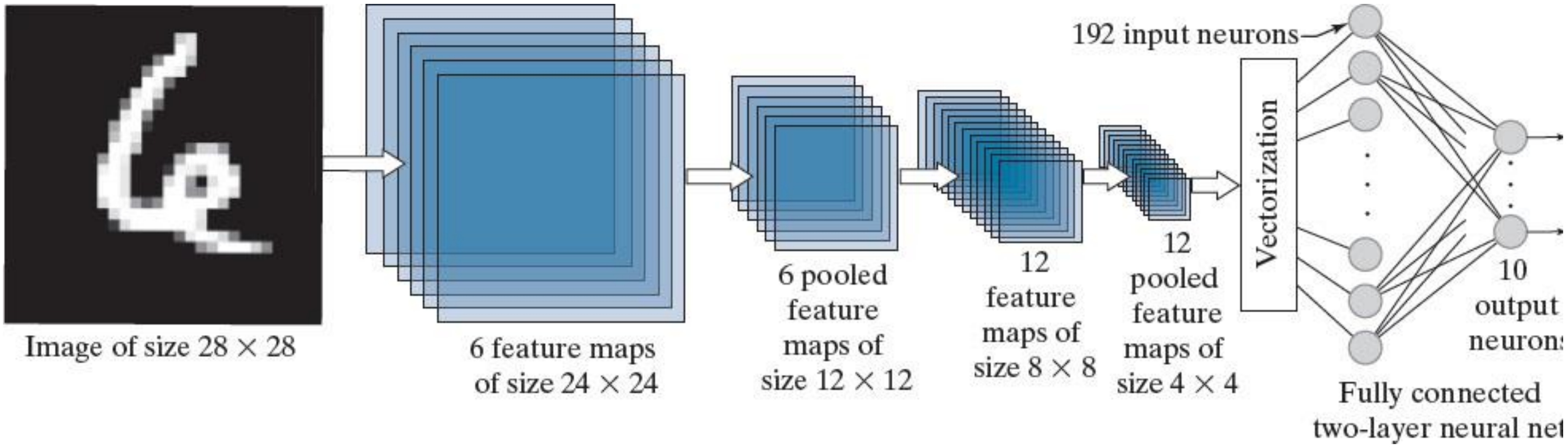
$a_1(\ell-1)$
$a_2(\ell-1)$
$a_j(\ell-1)$
$a_{n_{\ell-1}}(\ell-1)$
1
$z_i(\ell) = \sum_{j=1}^{n_{\ell-1}} w_{ij}(\ell)\, a_j(\ell-1) + b_i(\ell)$
h
$a_i(\ell) = h(z_i(\ell))$
Neuron i in hidden layer ℓ
Output $a_i(\ell)$ goes to all neurons in layer $\ell+1$
$a_i(\ell)$
x_1
x_2
x_3
x_n
Layer 1
(Input)
layer ℓ
Hidden Layers
(The number of nodes in the hidden layers can be different from layer to layer)
Layer L
(Output)

Note how the output of each neuron goes to the input of all neurons in the following layer, hence the name
fully connected for this type of architecture.

MNIST Image Dataset



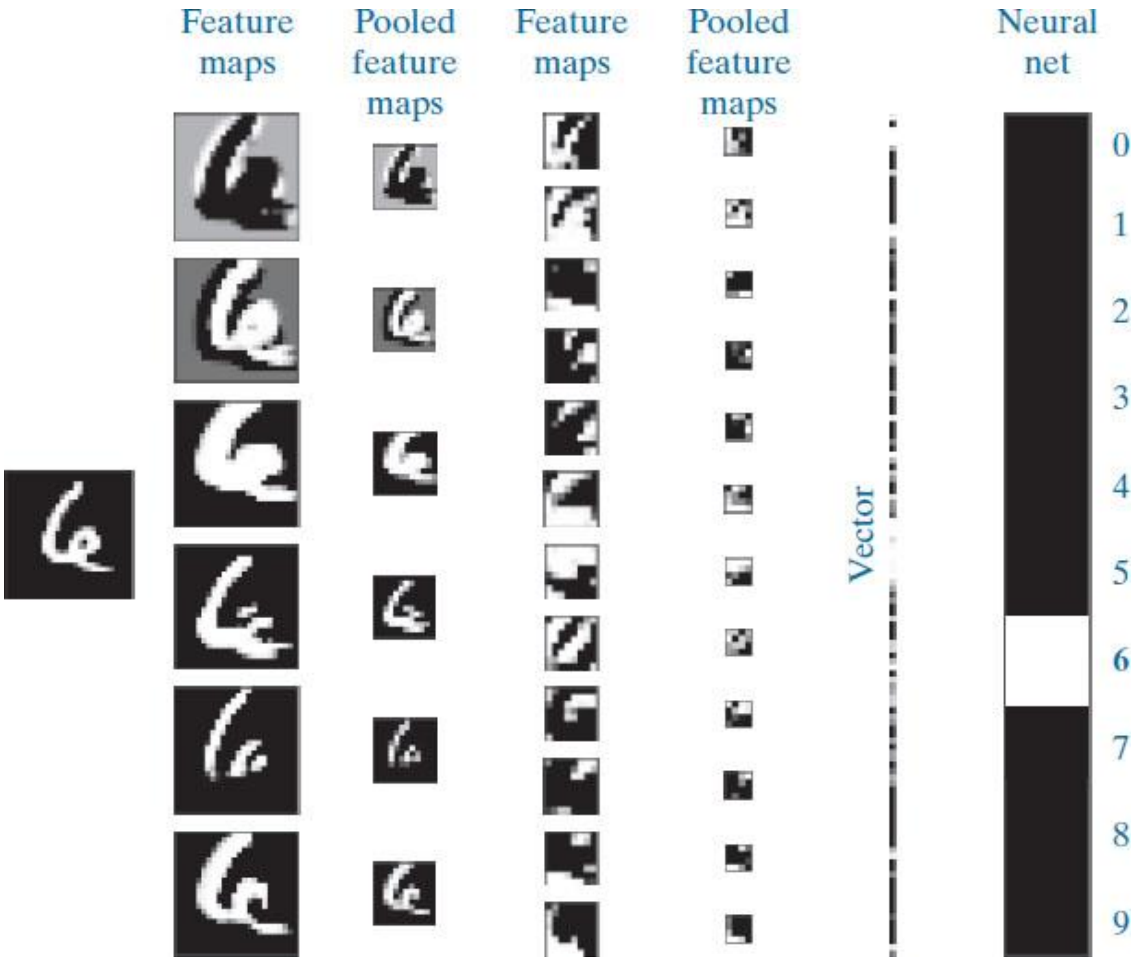
Convolutional Neural Network (CNN)



192 input neurons
Image of size 28 × 28
6 feature maps of size 24 × 24
6 pooled feature maps of size 12 × 12
12 feature maps of size 8 × 8
12 pooled feature maps of size 4 × 4
Vectorization
10 output neurons
Fully connected two-layer neural net

CNN used to recognize the ten digits in the MNIST database. The system was trained with 60,000 numerical character images of the same size as the image shown on the left.

Feature Maps



The output high value (in white) indicates that the CNN recognized the input properly.