COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Object Recognition

Lecture 11, April 21st, 2008

Lexing Xie

EE4830 Digital Image Processing
http://www.ee.columbia.edu/~xlx/ee4830/

1

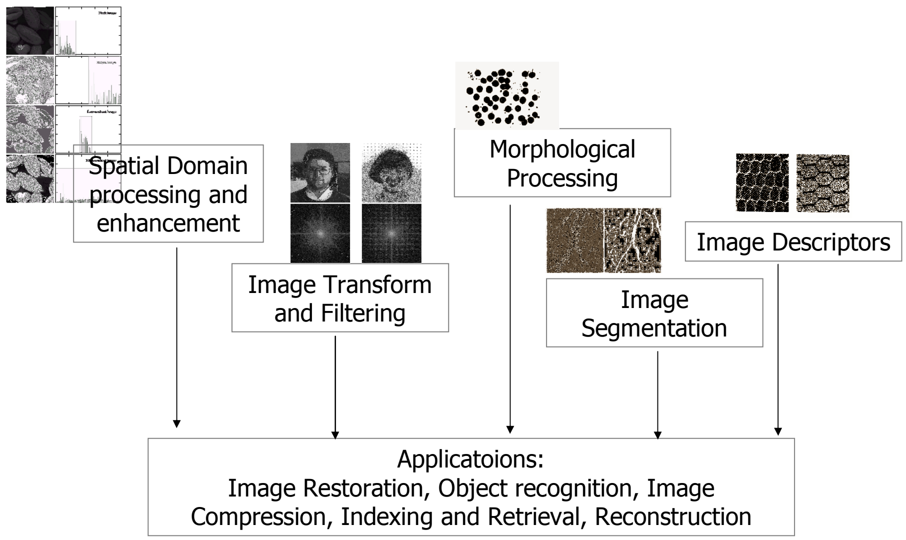
2

Announcements

- HW#5 due today
- HW#6
  - last HW of the semester
  - Due May 5th

3

Roadmap to Date



4

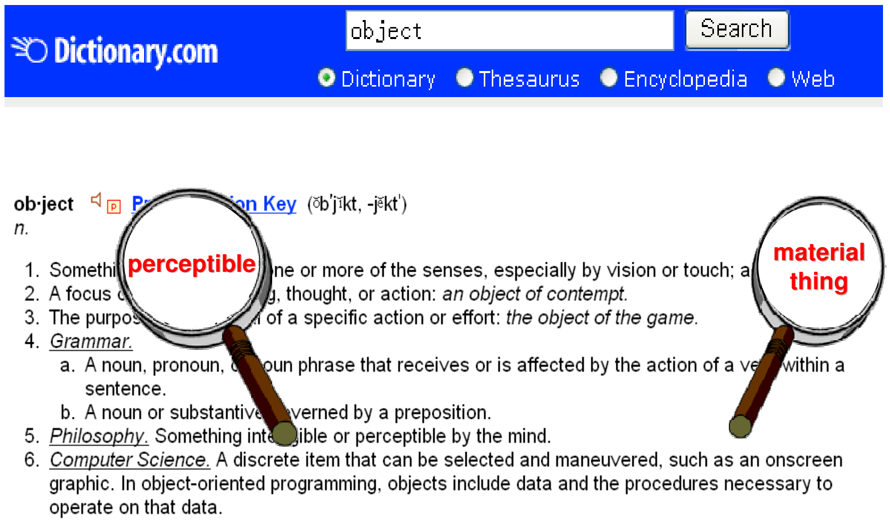
Lecture Outline

Problem: object recognition from images.

- What and why
- Pattern recognition primer
- Object recognition in controlled environments
- State of the art object recognition systems

What is Object Recognition?

5



Courtesy of http://people.csail.mit.edu/torralba/iccv2005/

What is Object Recognition?

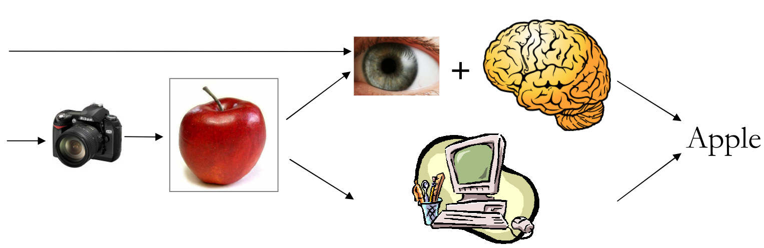
6

Sensory data → Descriptions

Color, texture, shape, motion, size, weight, smell, touch, sound, …

"toy", "stuffed Pooh", "a frontal, close-up shot of stuffed Pooh", "ToysRus #345812", …

One of the fundamental problems of computer vision:



Why?

7

- Science
  - How do we recognize objects?
- Practice
  - Robot navigation
  - Medical diagnosis
  - Security
  - Industrial inspection and automation
  - Human-computer interface
  - Information retrieval
  - …

Applications of Object Recognition

8

Some images from http://www.cs.utexas.edu/~grauman/research/research.html

9

Science & Technology

Computer vision

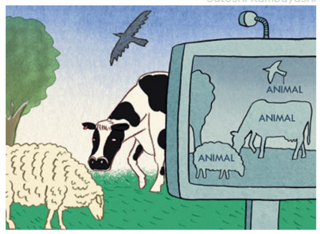
Easy on the eyes

Apr 4th 2007

From *The Economist* print edition

A computer can now recognise classes of things as accurately as a person can

Satoshi Kambayashi



NEVER underestimate a computer. Never overestimate one either. For many years Garry Kasparov, a world chess champion, said that a computer would never beat him (or, indeed, any other human in his position). In May 1997 he had to eat his words. Deep Blue, an invention of IBM, did just that.

This was impressive, but it demonstrated processing power rather than intelligence. Computers are generally good at solving specific problems, not specifically good at solving general ones. Deep Blue did not learn to play chess from experience. It was painstakingly programmed with thousands of "tactical weighting errors" devised by human experts. So whenever it selected a move, it used these to work through multitudes of possible options and their possible responses. No one is quite sure how Mr Kasparov's processor operates but it certainly does not do that. One theory goes that the human brain recognises strategic positions in a general way, and that this helps to reduce the problem to a manageable size.

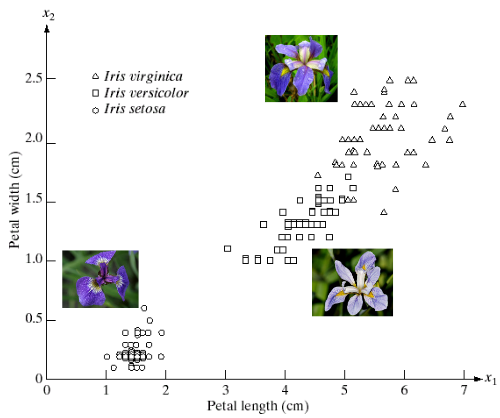
10

Lecture Outline

- Object recognition: what and why
- Object recognition in controlled environments
  - Distance-based classifiers
  - generalized linear classifiers
    - Neural networks
    - Bayes classifiers
  - Object recognition in practice
- General object recognition systems
- Summary

Objects as Vectors …

FIGURE 12.1
Three types of iris flowers described by two measurements.

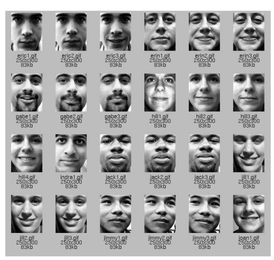


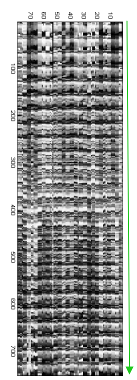
$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad y \in \{1, 2, 3\}$$

---

image

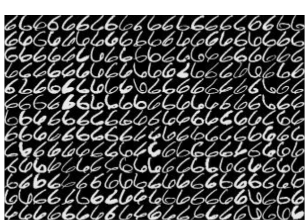
vector representation

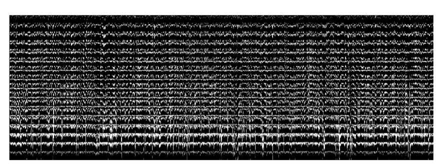




$$x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \cdots \\ x_{i,784} \end{bmatrix}$$

$y \in \{$female, male$\}$





$x_i, \; i = 1, \ldots, 1000 \qquad y \in \{0, 1, \ldots, 9\}$

pattern classifier from examples

- goal: given x, infer y
- learning from examples: supervised learning
  - given $(x_i, y_i=f(x_i))$, i=1,…,N for some unknown function f
  - find a "good approximation" to f
- rules versus data
  - encode human knowledge as rules
    - e.g. the petal length and width of iris
  - appropriate scenarios for supervised learning
    - no human expert (predict strength to cure AIDS given new molecule structure)
    - human can perform task but can't describe how they do it (e.g. handwriting recognition, object recognition)
    - the desired function is changing constantly w.r.t. time, or user (stock trading decisions, user-specific spam filtering)

minimum distance classifier

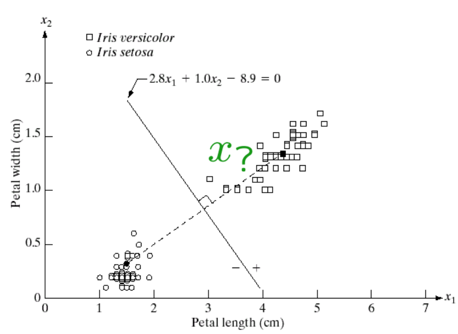


FIGURE 12.6 Decision boundary of minimum distance classifier for the classes of *Iris versicolor* and *Iris setosa*. The dark dot and square are the means.

$(x_i, y_i)\ i = 1, \ldots, N$

$x_i \in \mathcal{R}^2,\ \ y_i \in \{+1, -1\}$

step 1: calculate "class prototypes" as the means

$$m_j = \frac{1}{N_j} \sum_i x_i \delta(y_i = j)$$

step 2: use the prototypes to classify a new example

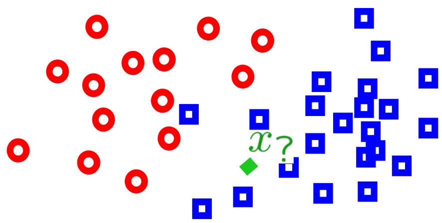
$$\hat{y}_? = \arg\min_j d(x_?, m_j),\ \ j = 1, 2$$

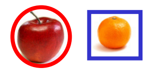
"discriminant" function f:

$$f(x) = sign(2.8x_1 + 1.0x_2 - 8.9)$$

15

nearest neighbor classifier





$(x_i, y_i)\ i = 1, \ldots, N$

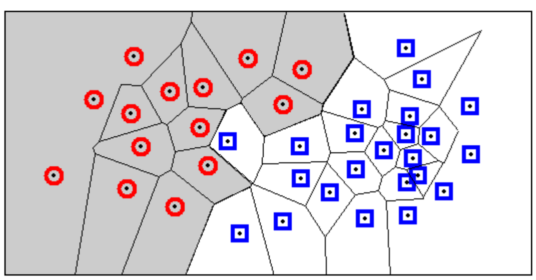
$x_i \in \mathcal{R}^2,\ \ y_i \in \{+1, -1\}$

- steps:
  - store all training examples
  - classify a new example $x_?$ by finding the training example (x_i, y_i) that's nearest to $x_?$ according to Euclidean distance, and copying the labels

$$\hat{y}_? = y_j,\ \ j = \arg\min_{i=1,\ldots,N} \|x_? - x_i\|$$

16

nearest neighbor classifier



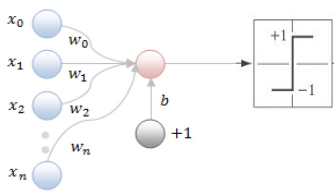
"discriminant" function f:
gray area -1; white area +1

- (implicit) decision boundaries form a subset of the Voronoi diagram of the training data – each line segment is equidistant between two points
- comments
  - prone to noisy, poorly scaled features
  - conditioned on the distance metric
  - "smooth" the decision by looking at K-neighbors and vote
  - good news: kNN is "universally consistent"

linear classifier

- two desirables
  - explicit (linear) decision boundary
  - use many training examples/prototypes but do no need to remember all

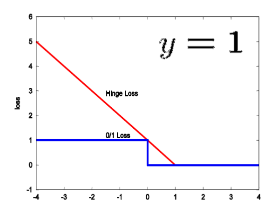
$$\hat{y} = f(x) = sign(w^T x + w_0) = sign(\sum_d w_d x_{id} + w_0)$$



$$o = \begin{cases} +1 & \text{if } w^T x + w_0 > 0 \\ -1 & \text{if } w^T x + w_0 \le 0 \end{cases}$$

the perceptron algorithm

$$\hat{y} = f(x) = sign(w^T x + b)$$



- learning a linear classifier
  - given training data (x_i, y_i) and loss function

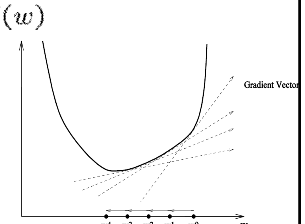
$$L(f(x), y)$$

  - find: weight vector [w; b] that minimizes expected loss on training data

$$\min \quad J(w) = \frac{1}{N}\sum_{i=1}^{N} L(f(x_i), y_i)$$

use hin

$$= \frac{1}{N}\sum_{i=1}^{N} max(0, 1 - y_i w^T x_i)$$



- start from initial weights w_0
- compute gradient $\nabla \tilde{J}(w) = \left[\frac{\partial \tilde{J}(w_0)}{\partial w_0}, \ldots, \frac{\partial \tilde{J}(w_D)}{\partial w_D}\right]$
- update $w_1 = w_0 - \eta \nabla \tilde{J}(w_0)$ η : learning rate
- repeat until convergence

computing the gradient

given $J(w) = \frac{1}{N} \sum_{i=1}^{N} max(0, 1 - y_i w^T x_i)$ compute gradient $\nabla \tilde{J}(w)$

let $\tilde{J}_i(w) = max(0, 1 - y_i w^T \cdot x_i)$ contribution from each training sample

$$\frac{\partial \tilde{J}(w_d)}{\partial w_d} = \frac{\partial}{\partial w_d}\left(\frac{1}{N}\sum_i \tilde{J}_i(w)\right) = \frac{1}{N}\sum_i \left(\frac{\partial}{\partial w_d}\tilde{J}_i(w)\right)$$

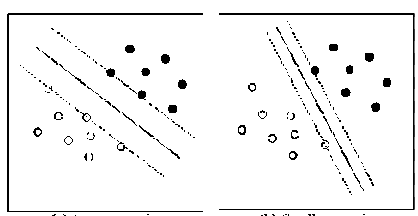
$$\frac{\partial \tilde{J}_i(w)}{\partial w_d} = \frac{\partial}{\partial w_d} \text{max}\left(0, 1 - y_i \textstyle\sum_{j=1}^{D} w_j x_{ij}\right) = \begin{cases} 0 & \text{if } y_i w^T x > 0 \\ -y_i x_{id} & \text{otherwise} \end{cases}$$

contribution from each dimension of each training sample

- η must decrease to zero in order to guarantee convergence.
- some algorithms (Newton's) can automatically select η.
- local minimum is the global minimum for hinge loss

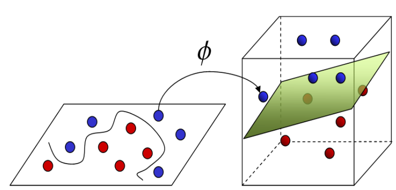
Support Vector Machines

- Two key ideas:
  - The "best" separating hyperplane has the largest margin.
  - Class boundary can be linear in a higher-dimensional space, e.g.,



(a) Larger margin. (b) Smaller margin.

$$\Phi\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{bmatrix} x_1^2 \\ \sqrt{2}x_1 x_2 \\ x_2^2 \end{bmatrix}$$



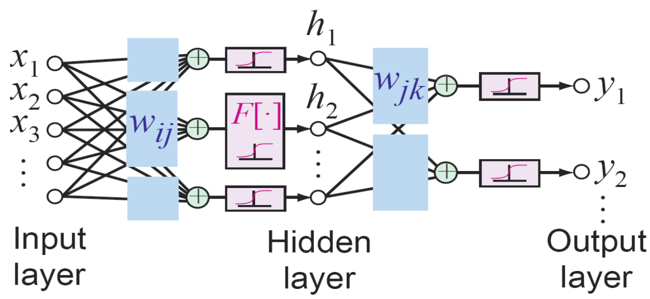
$$f(x) = sign(w^T \Phi(x)) = \sum_i \alpha_i K(x_i, x)$$

generalized linear discriminant

weighted (generalized) inner product with "support vectors"

Neural Networks

$$y_k = F[\sum_j w_{jk} \cdot F[\sum_j w_{ij} x_i]] \qquad F(u) = \frac{1}{1 + e^{-u}}$$



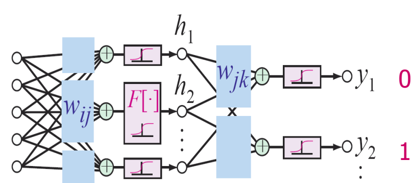
a single hidden layer, feed forward neural network is capable of approximating any continuous, multivariate function to any desired degree of accuracy and that failure to map a function arises from poor choice of network parameters, or an insufficient number of hidden neurons.

[Cybenko 1989]

Digit Recognition with Neural Net

- LeCun et al, 1992, 1998, …

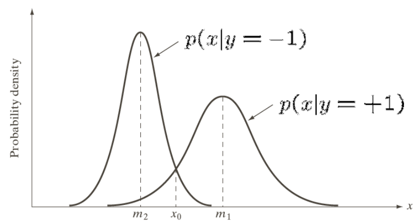
http://yann.lecun.com/exdb/mnist/

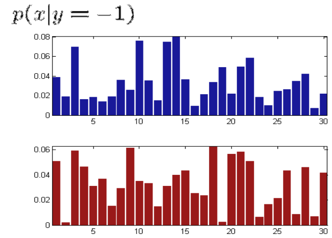


40 PCA + quadratic classifier	none	3.3	LeCun et al. 1998
1000 RBF + linear classifier	none	3.6	LeCun et al. 1998
K-NN, Tangent Distance	subsampling to 16x16 pixels	1.1	LeCun et al. 1998
SVM, Gaussian Kernel	none	1.4	
SVM deg 4 polynomial	deskewing	1.1	LeCun et al. 1998
Reduced Set SVM deg 5 polynomial	deskewing	1.0	LeCun et al. 1998
Virtual SVM deg-9 poly [distortions]	none	0.8	LeCun et al. 1998
Virtual SVM, deg-9 poly, 1-pixel jittered	none	0.68	DeCoste and Scholkopf, MLJ 2002
Virtual SVM, deg-9 poly, 1-pixel jittered	deskewing	0.68	DeCoste and Scholkopf, MLJ 2002
Virtual SVM, deg-9 poly, 2-pixel jittered	deskewing	0.56	DeCoste and Scholkopf, MLJ 2002
2-layer NN, 300 hidden units, mean square error	none	4.7	LeCun et al. 1998
2-layer NN, 300 HU, MSE, [distortions]	none	3.6	LeCun et al. 1998
2-layer NN, 300 HU	deskewing	1.6	LeCun et al. 1998
2-layer NN, 1000 hidden units	none	4.5	LeCun et al. 1998
2-layer NN, 1000 HU, [distortions]	none	3.8	LeCun et al. 1998
3-layer NN, 300+100 hidden units	none	3.05	LeCun et al. 1998
3-layer NN, 300+100 HU [distortions]	none	2.5	LeCun et al. 1998
3-layer NN, 500+150 hidden units	none	2.95	LeCun et al. 1998
3-layer NN, 500+150 HU [distortions]	none	2.45	LeCun et al. 1998
3-layer NN, 500+300 HU, softmax, cross entropy, weight decay	none	1.53	Hinton, unpublished, 2005
2-layer NN, 800 HU, Cross-Entropy Loss	none	1.6	Simard et al., ICDAR 2003
2-layer NN, 800 HU, cross-entropy [affine distortions]	none		
2-layer NN, 800 HU, MSE [elastic distortions]	none	0.9	Simard et al., ICDAR 2003
2-layer NN, 800 HU, cross-entropy [elastic distortions]	none	0.7	Simard et al., ICDAR 2003

probabilistic classifiers

- what about probabilities
  - p(x|y) is usually easy to obtain from training data
  - can we estimate p(y|x) ?





Bayes classifier

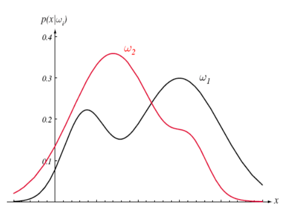


FIGURE 2.1. Hypothetical class-conditional probability density functions show t probability density of measuring a particular feature value x given the pattern is category ω_i. If x represents the lightness of a fish, the two curves might describe t difference in lightness of populations of two types of fish. Density functions are norm ized, and thus the area under each curve is 1.0. From: Richard O. Duda, Peter E. Ha and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & So Inc.

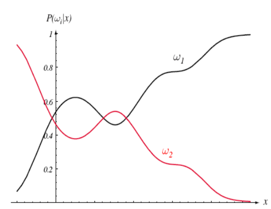


FIGURE 2.2. Posterior probabilities for the particular priors $P(\omega_1) = 2/3$ and $P(\omega_2) = 1/3$ for the class-conditional probability densities shown in Fig. 2.1. Thus in this case, given that a pattern is measured to have feature value $x = 14$, the probability it is in category ω_2 is roughly 0.08, and that it is in ω_1 is 0.92. At every x, the posteriors sum to 1.0. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

$$
\begin{aligned}
p(y=+1|x) &= p(y=+1)\frac{p(x|y=+1)}{p(x)} \\
&= p(y=+1)\frac{p(x|y=+1)}{p(y=+1)p(x|y=+1)+p(y=-1)p(x|y=-1)}
\end{aligned}
$$

$$
f(x) = \frac{p(y=+1|x)}{p(y=-1|x)} = \frac{p(y=+1)p(x|y=+1)}{p(y=-1)p(x|y=-1)}
$$

25

Bayes classifier for Gaussian classes

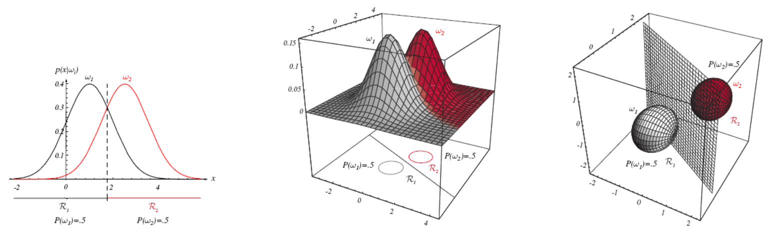
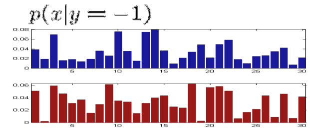


FIGURE 2.10. If the covariance matrices for two distributions are equal and proportional to the identity matrix, then the distributions are spherical in d dimensions, and the boundary is a generalized hyperplane of $d-1$ dimensions, perpendicular to the line separating the means. In these one-, two-, and three-dimensional examples, we indicate $p(\mathbf{x}|\omega_i)$ and the boundaries for the case $P(\omega_1) = P(\omega_2)$. In the three-dimensional case, the grid plane separates $\mathcal{R}_1$ from $\mathcal{R}_2$. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

26

estimating the conditionals

- how do we estimate p(x|y)
  - $x_1, x_2, \ldots, x_N$ discrete:
    count over observed samples
    to get the conditional histograms
  - $x_1, x_2, \ldots, x_N$ continuous and conditionally Gaussian



x scalar $\quad p(x|y=j) = \frac{1}{\sqrt{2\pi}\sigma_j}\exp\{-(x-\mu_j)^2/\sigma_j^2\}$

$$\mu_j = \frac{1}{N_j}\sum_{\{i|y_i=j\}} x_i$$

$$\sigma_j = \frac{1}{N_j}\sum_{\{i|y_i=j\}} x_i^2 - \mu_j^2$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_d \end{bmatrix} \quad p(x|y=j) = \frac{1}{\sqrt{2\pi|C_j|}}\exp\{-(x-\mu_j)C_j^{-1}(x-\mu_j)^T\}$$

$$\mu_j = \frac{1}{N_j}\sum_{\{i|y_i=j\}} x_i$$

$$\begin{aligned} C_j &= \frac{1}{N_j}\sum_{\{i|y_i=j\}} (x_i-\mu_j)(x_i-\mu_j)^T \\ &= \frac{1}{N_j}\sum_{\{i|y_i=j\}} x_i x_i^T - \mu_j\mu_j^T \end{aligned}$$

27

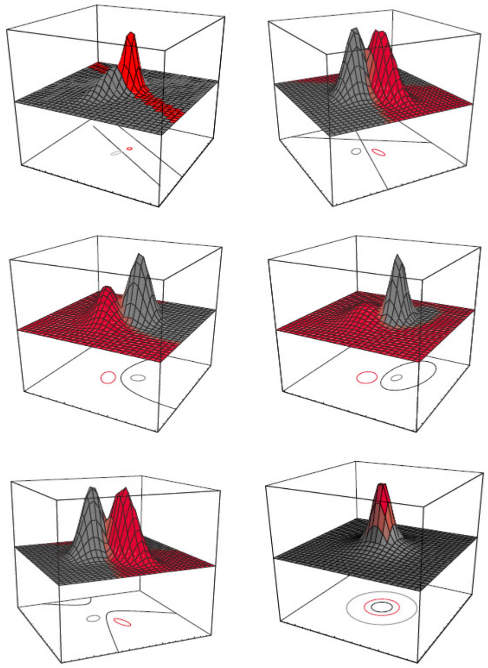
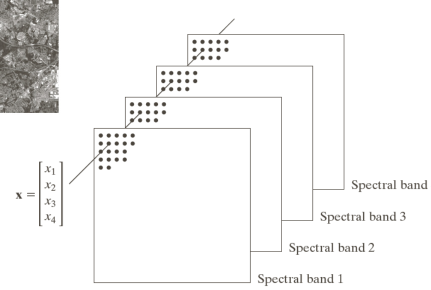


FIGURE 2.14. Arbitrary Gaussian distributions lead to Bayes decision boundaries that are general hyperquadrics. Conversely, given any hyperquadric, one can find two Gaus-

28

Bayes classifier example

FIGURE 12.4 Satellite image of a heavily built downtown area (Washington, D.C.) and surrounding residential areas. (Courtesy of NASA.)



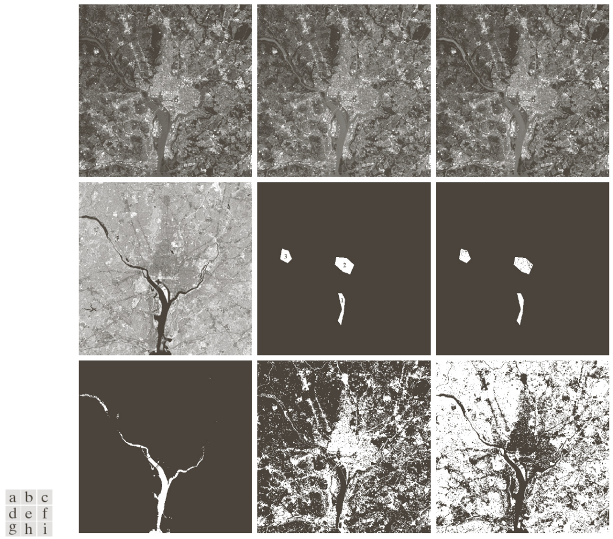


FIGURE 12.13 Bayes classification of multispectral data. (a)–(d) Images in the visible blue, visible green, visible red, and near infrared wavelengths. (e) Mask showing sample regions of water (1), urban development (2), and vegetation (3). (f) Results of classification; the black dots denote points 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 vegetations (in white).

classification results

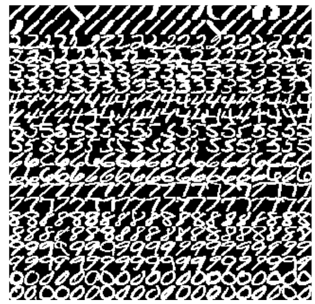
TABLE 12.1
Bayes classification of multispectral image data.

Training Patterns						Independent Patterns					
		Classified into Class						Classified into Class			
Class	No. of Samples	1	2	3	% Correct	Class	No. of Samples	1	2	3	% Correct
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

homework problem 1: classifying digits

31

- instruction/code available
  - load digits from the MNIST dataset
  - baseline 1-NN classifier
- experiment/observe/improve
  - k-NN, with k=3, 5
  - SVM / linear classifier
  - compute error rate
  - examples that are correctly/incorrectly classified



$$\text{err rate} = \frac{\#\ \text{miss-classified digits}}{\text{total}\ \#\text{of digits}} \times 100\%$$

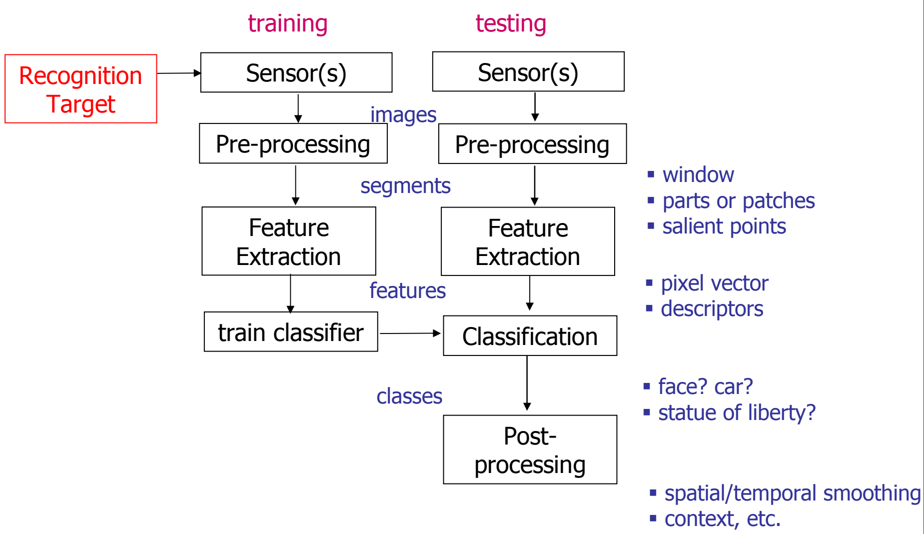
Lecture Outline

32

- object recognition: what and why
- object recognition as pattern classification
- general object recognition systems
  - object recognition: a systems view
  - current commercial systems
  - real-world challenges
  - survey of state-of the art
- demo websites

Object Recognition End-to-End

33



Object Recognition in Practice

34

- Commercial object recognition
  - Currently a $4 billion/year industry for inspection and assembly
  - Almost entirely based on template matching
- Upcoming applications
  - Mobile robots, toys, user interfaces
  - Location recognition
  - Digital camera panoramas, 3D scene modeling

courtesy of David Lowe,
website and CVPR 2003 Tutorial

Industrial Applications

35

The Computer Vision Industry

David Lowe

This web page provides links to companies that develop products using computer vision. Computer vision (also often referred to as "machine vision" or "automated imaging") is the automated extraction of information from images. This differs from image processing, in which an image is processed to produce another image. This page covers only products based on computer or machine vision, and it does not cover image processing or any of the many suppliers of sensors or other equipment to the industry.

Companies are categorized under their principal application area, and then listed alphabetically. Companies are listed only if they have web pages giving information about their products. Please let me know of any links that are missing.

Automobile driver assistance

Iteris (Anaheim, California). Lane departure warning systems for trucks and cars that monitor position on the road. Used in over 10,000 trucks (2005). Also creates traffic monitoring systems.

MobilEye (Jerusalem, Israel). Vision systems that warn automobile drivers of danger, provide adaptive cruise control, and give driver assistance.

Smart Eye (Göteborg, Sweden). Systems to track eye and gaze position of a driver to detect drowsiness or inattention.

Automobile traffic management

Appian Technology (Bourne End, Buckinghamshire, UK). Systems for reading automobile license plates.

AutoVu (Montreal, Canada). Systems for reading automobile license plates.

Image Sensing Systems (St. Paul, Minnesota). Created the Autoscope system that uses roadside video cameras for real-time traffic management. Over 40,000 cameras are in use.

Film and Television

2D3 (Oxford, UK). Systems for tracking objects in video or film and solving for 3D motion to allow for precise augmentation with 3D computer graphics.

Hawkeye (Winchester, UK). Uses multiple cameras to precisely track tennis and cricket balls for sports refereeing and commentary.

Image Metrics (Manchester, England). A markerless tracking system for the human face that can be used to map detailed motion and facial expressions to synthetic characters.

Imagineer Systems (Guildford, UK). Computer vision software for the film and video industries.

http://www.cs.ubc.ca/spider/lowe/vision.html

36

http://www.appian-tech.com/

37

http://www.sportvision.com/

38

http://www.dipix.com/

What to Recognize

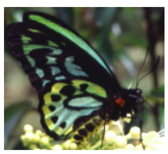
39

Wild card

Tower Bridge

The Stata Center

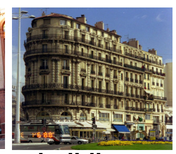
Specific



butterfly

butterfly

building



building

Categories

Kristen Grauman, http://www.cs.utexas.edu/~grauman/research/

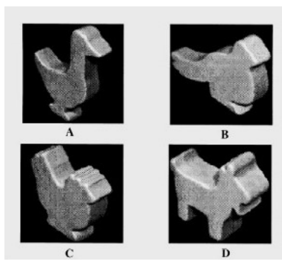
40

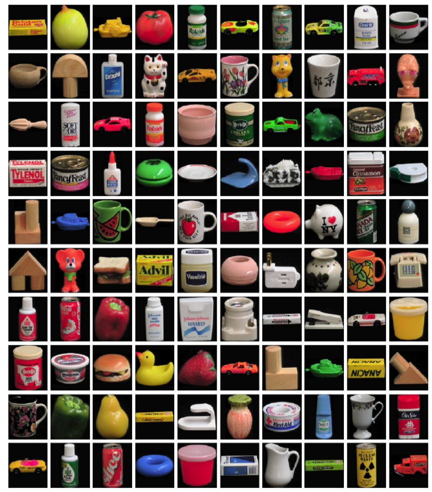
Challenges of object recognition …

Recognize Specific Objects (1)

41

Appearance Matching

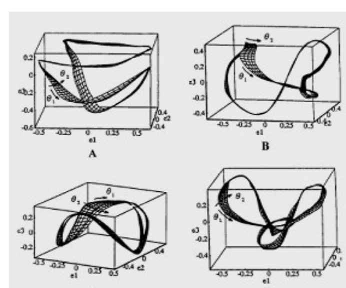


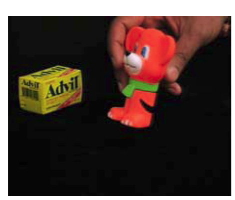


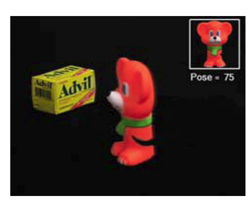
[Nayar, Murase et. al.]

42

- PCA on the training set.
- Estimate parameters of a low-dimensional pose manifold with splines.
- Match new image to the closest point on the manifold.



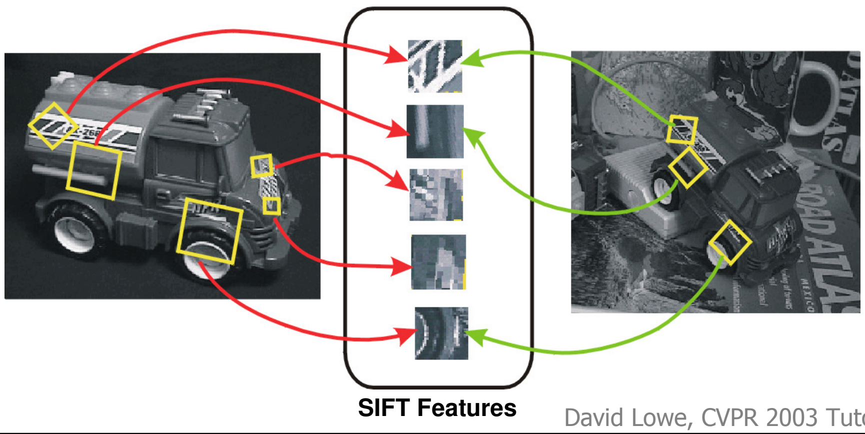




Recognize Specific Objects (2)

43

- Part-based approach
  - Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters
  - select "interest points" that are stable extrema points across different scales.

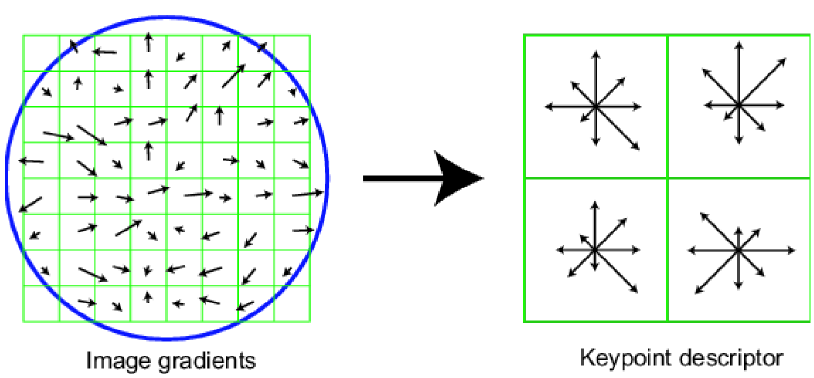


David Lowe, CVPR 2003 Tutorial

SIFT Descriptor

44

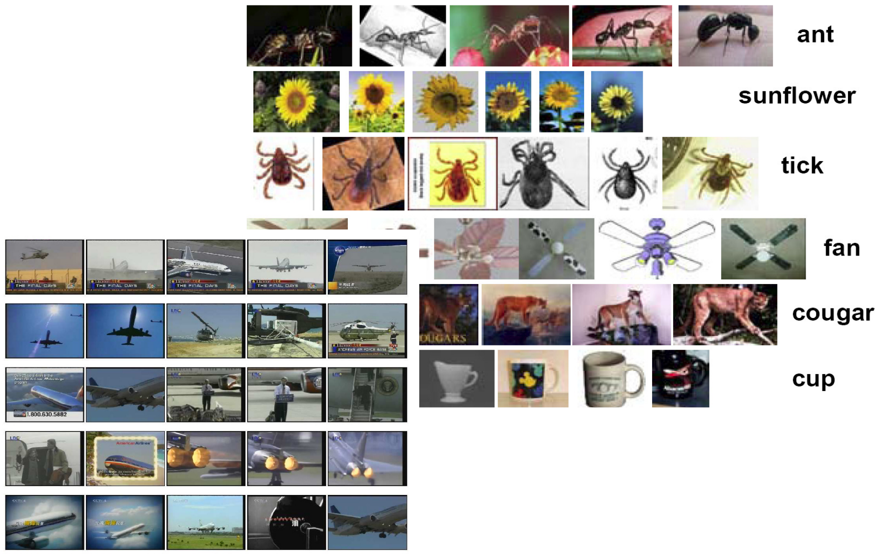
- Thresholded image gradients are sampled over 16x16 array of locations in scale space (Gaussian-weighted).
- Create array of orientation histograms
- 8 orientations x 4x4 histogram array = 128 dimensions



David Lowe, CVPR 2003 Tutorial

Object Category Recognition

45



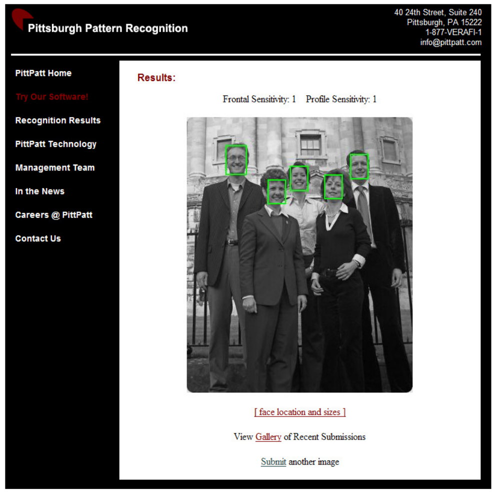
46

Overview of object category recognition …
see iccv tutorial

Demos

47

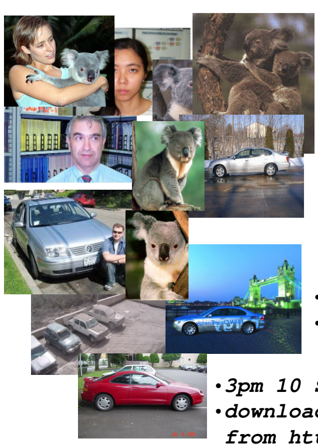
- Pittpatt http://demo.pittpatt.com/



It's not just vision…

48

Integrate with mobile sensor information (GPS, time, nearby object or people), calendar, schedule…
Infer semantically rich meta-data labels from joint sources.



- *10am 7 Sep 05*
- *Australian park*
- *Jim, Jill nearby*

- *4pm 8 Sep 05*
- *Sydney*

- *8pm 10 Oct 05*
- *London*

- *3pm 10 Sep 05*
- *downloaded from http://...*

"two koalas seen on nat. park trip with Jim and Jill"

"Jill and koala on nat. park trip"

"John and his new car"

"office parking lot"

"car to consider purchasing"

http://www.cs.utexas.edu/~grauman/research/research.html

Summary

- The object recognition problem
- Pattern classification primer
- Object recognition grown up

- Readings: G&W 12.1-12.2
- Reference: Duda, Hart, Stork, "Pattern Classification", 2nd Ed.

- Next time: Image Compression

Additional acknowledgements: Dan Ellis, EE6820 Slides; Duda, Hart& Stork, Pattern classificaion 2nd Ed., David Claus and Christoph F. Eick: Nearest Neighbor Editing and Condensing Techniques