Recognition, Detection, and Classification (Part 1)

Computer Vision I

CSE 252A

Lecture 13

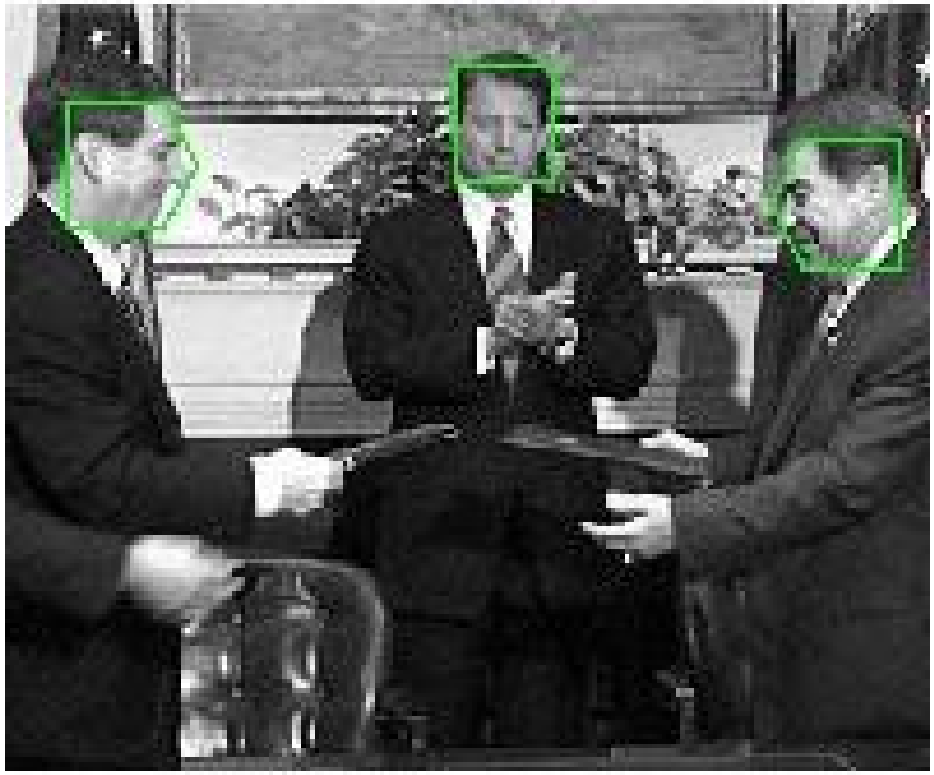
Announcements

- Assignment 3 is due Nov 22, 11:59 PM
- Assignment 4 will be released Nov 22
  - Due Dec 6, 11:59 PM

Recognition

Given a database of objects and an image determine what, if any of the objects are present in the image.

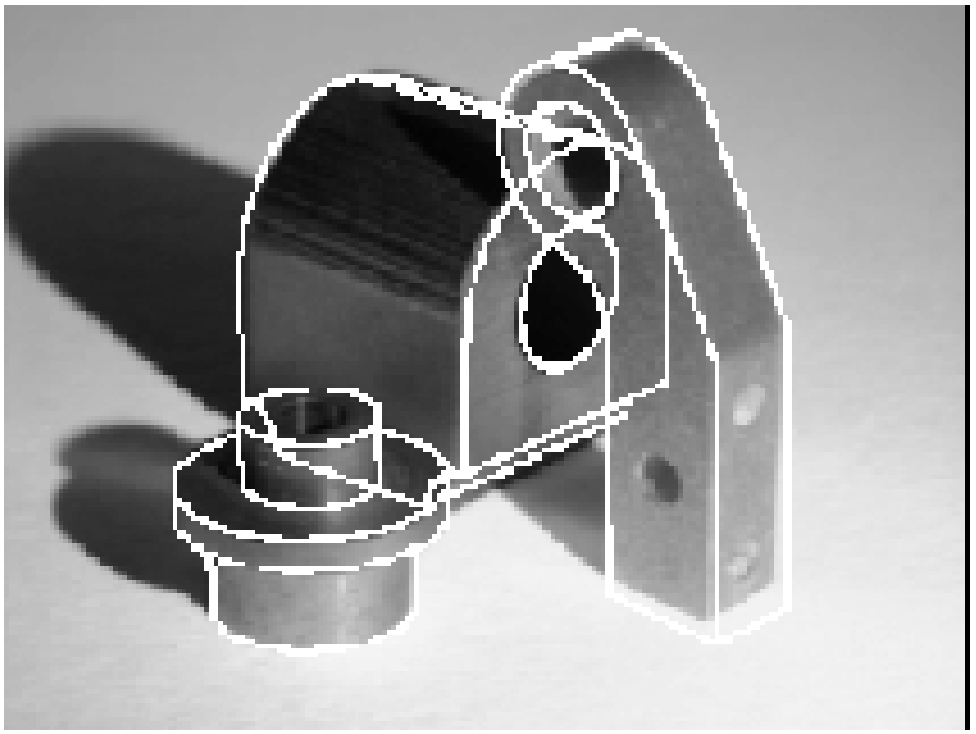
Recognition



Given a database of objects and an image determine what, if any of the objects are present in the image.

… and where are they and how many

Recognition

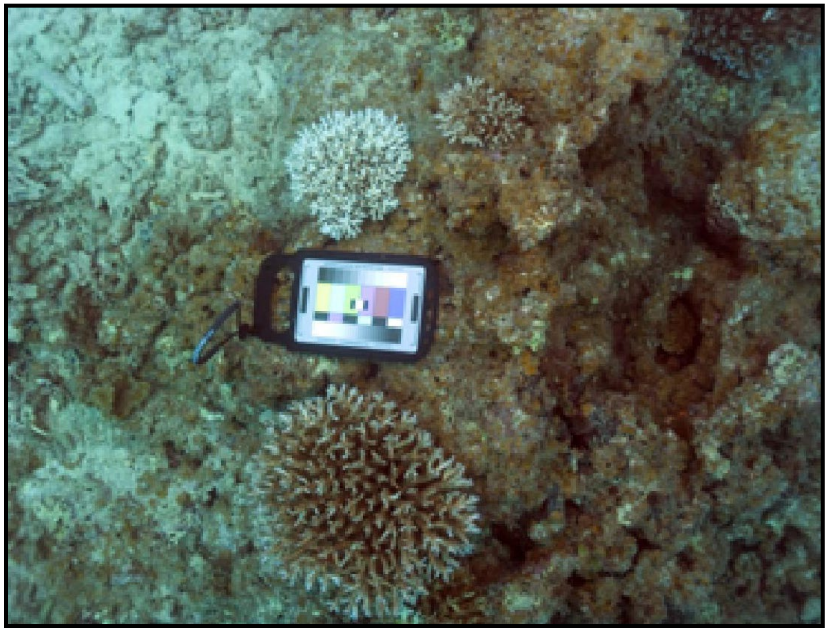


Given a database of objects and an image determine what, if any of the objects are present in the image.

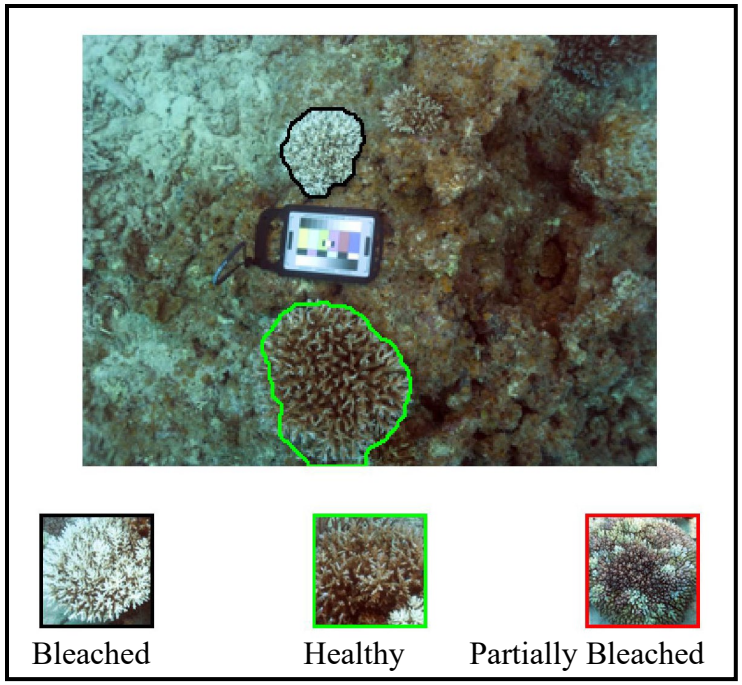
… and where are they and how many

… and what is 3D pose

Example: where are the coral heads and which ones are healthy and which are bleached?

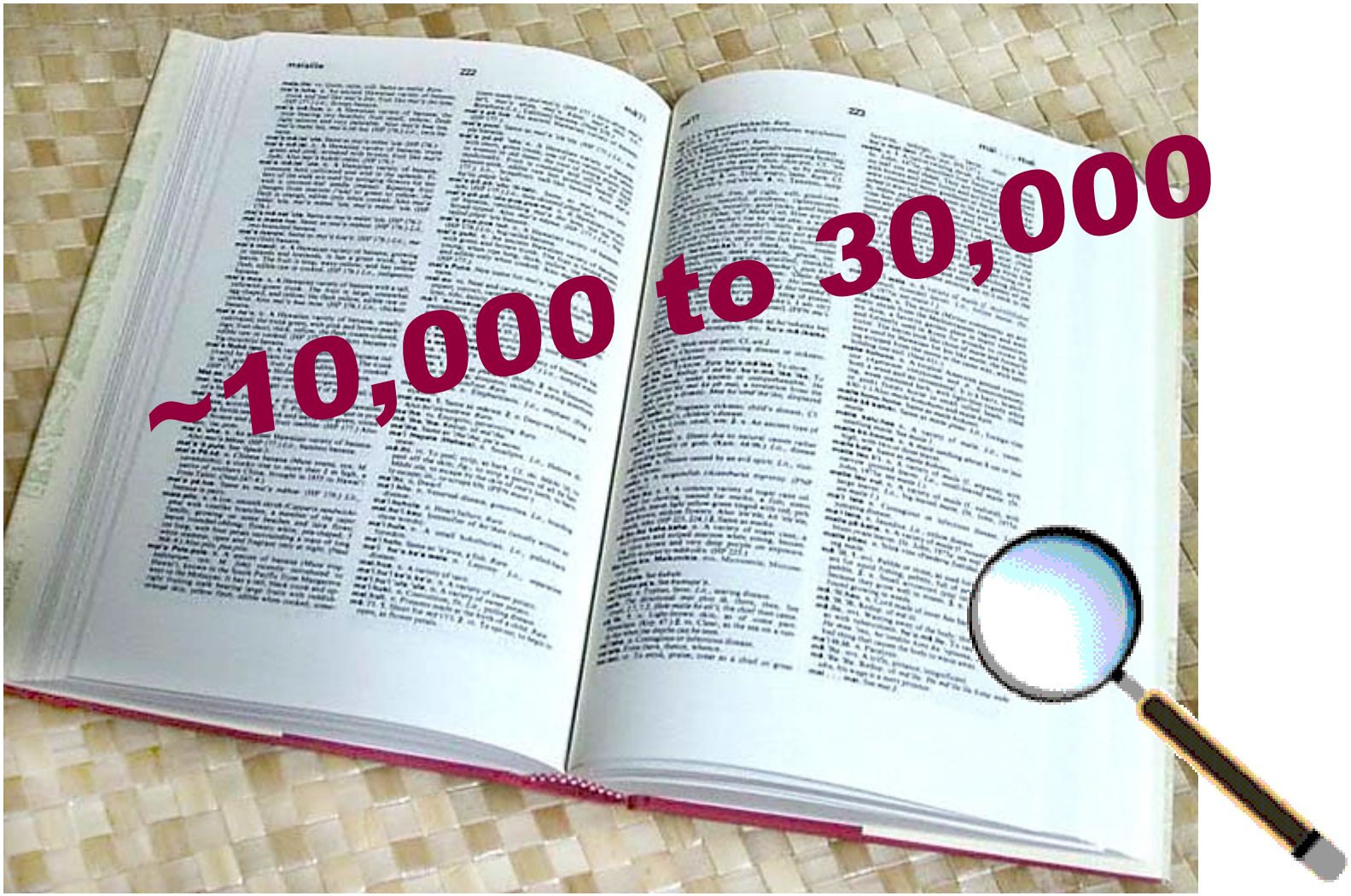


Input Image



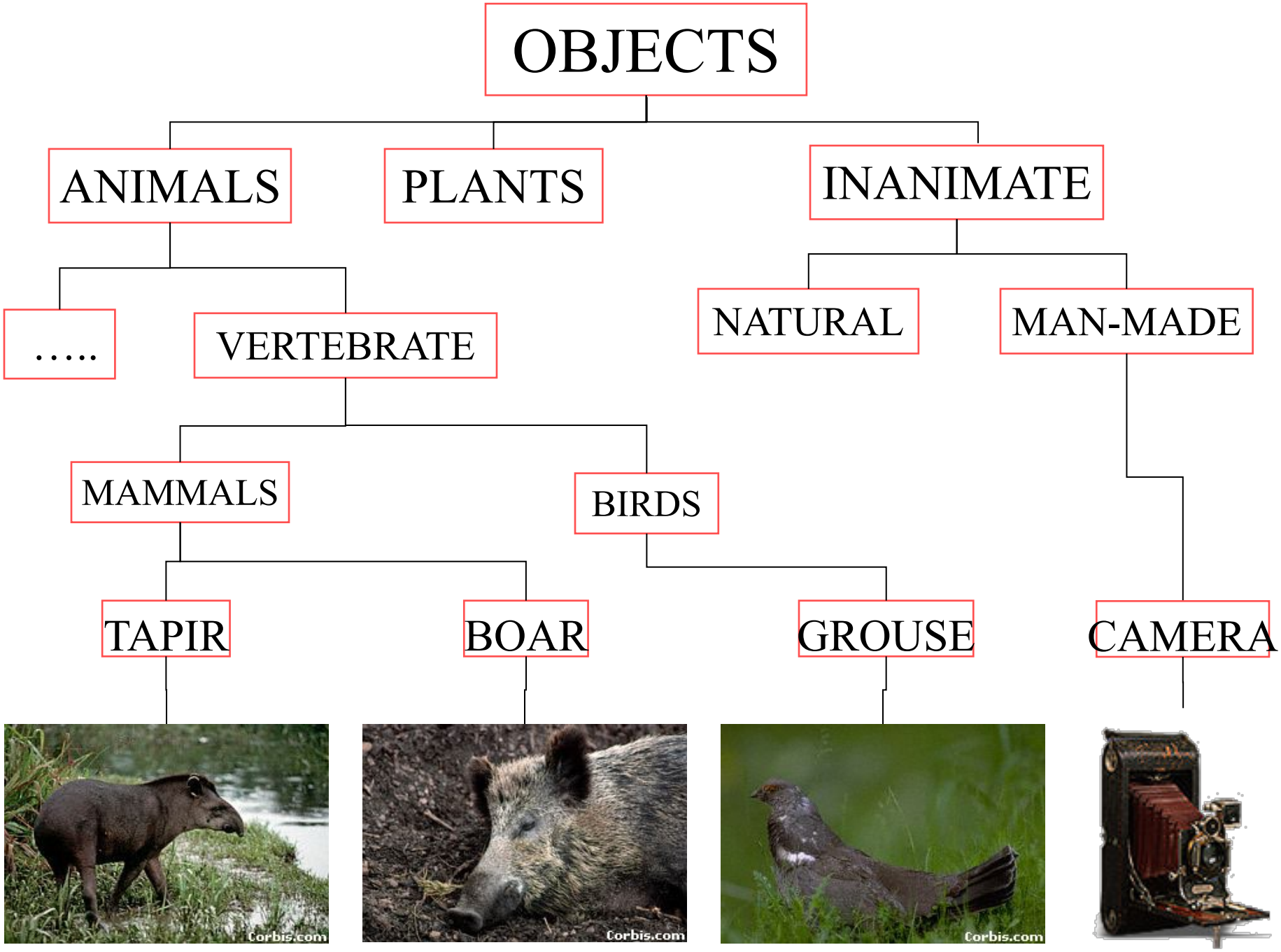
Segmented/labeled Image

How many visual object categories are there?



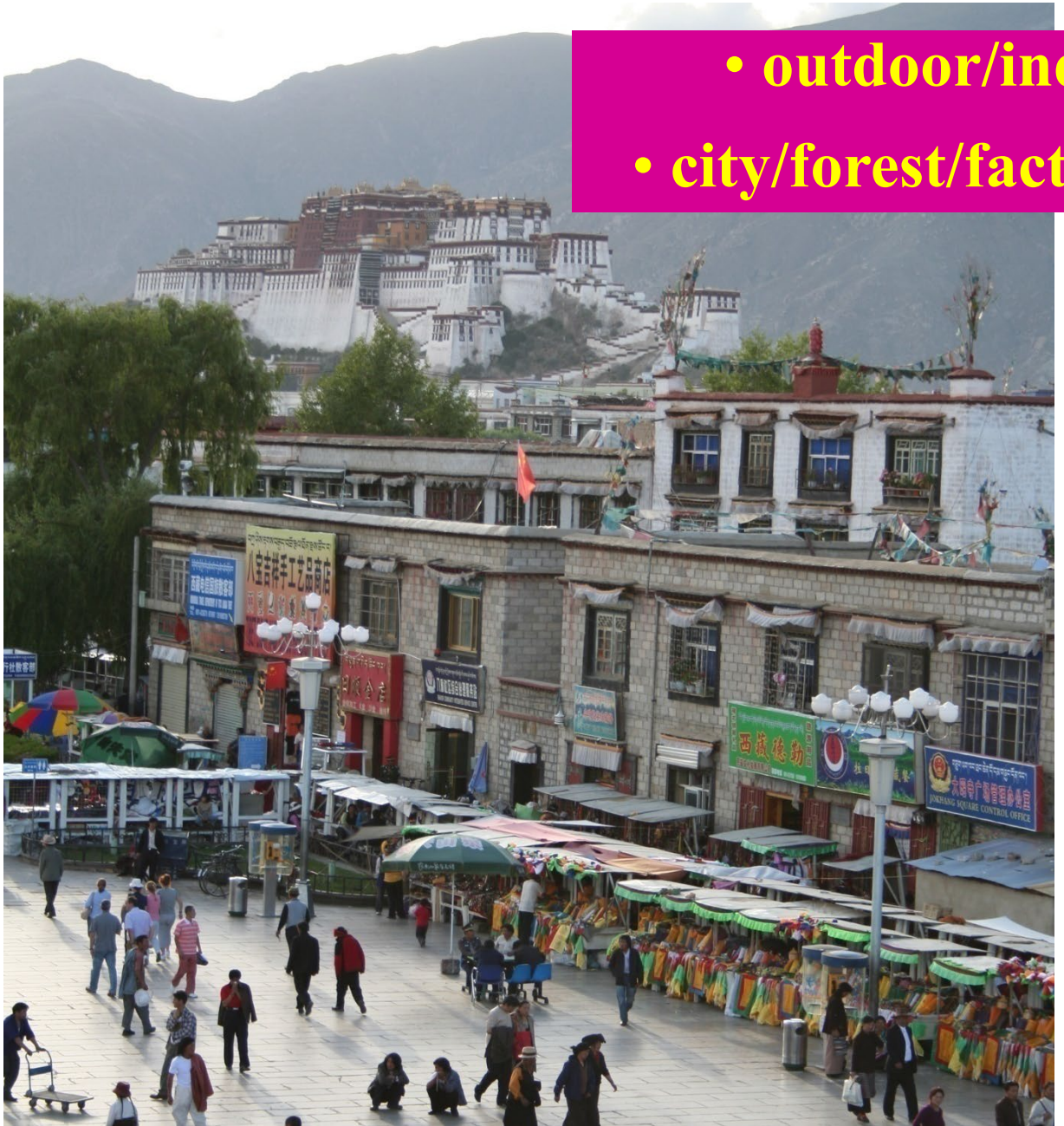
Biederman 1987


~10,000 to 30,000


OBJECTS
ANIMALS
PLANTS
INANIMATE
.....
VERTEBRATE
NATURAL
MAN-MADE
MAMMALS
BIRDS
TAPIR
BOAR
GROUSE
CAMERA
Corbis.com
Corbis.com
Corbis.com

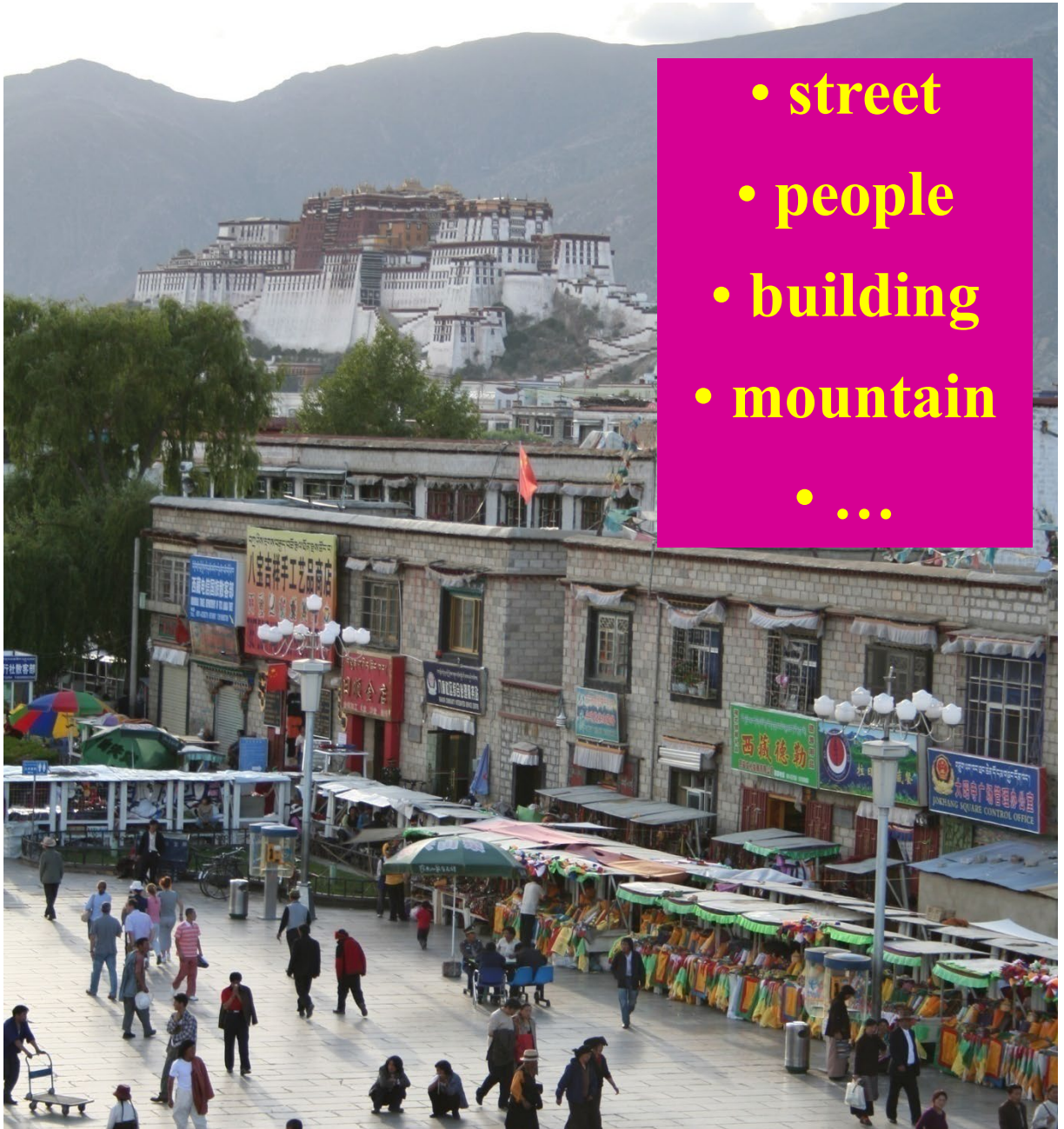
Specific recognition tasks

Scene categorization



- outdoor/indoor
- city/forest/factory/etc.

Image annotation/tagging



Object detection

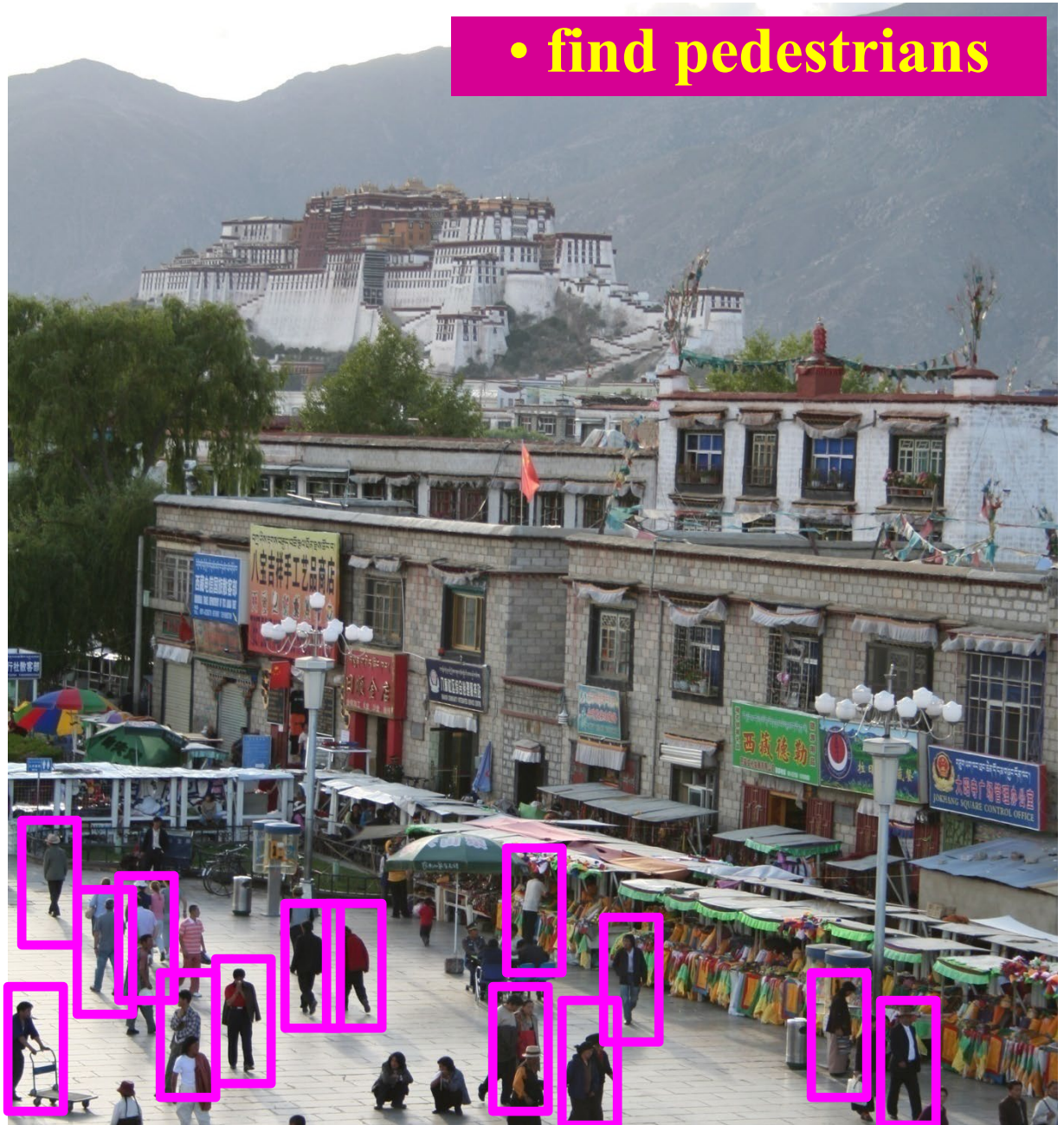
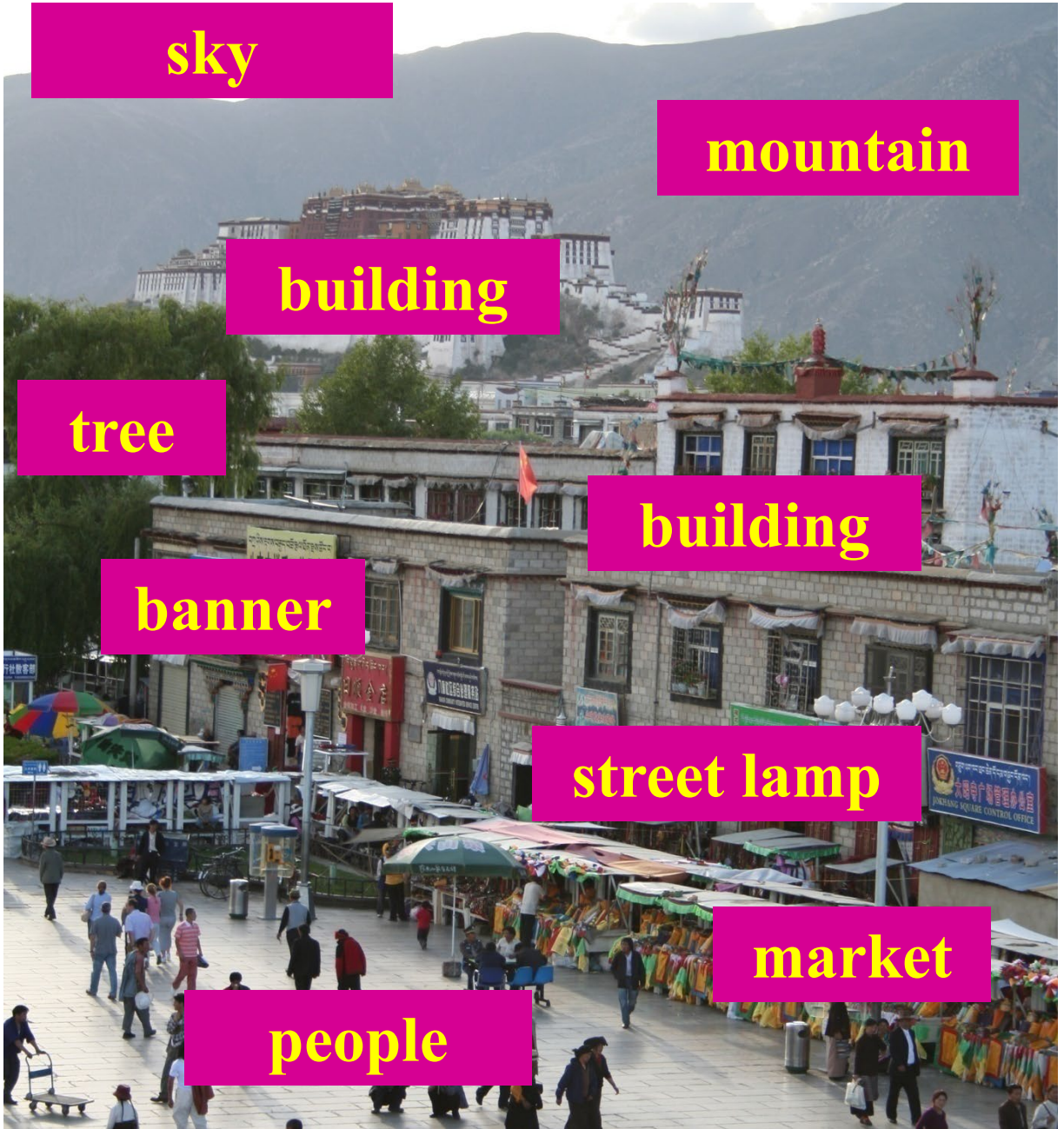


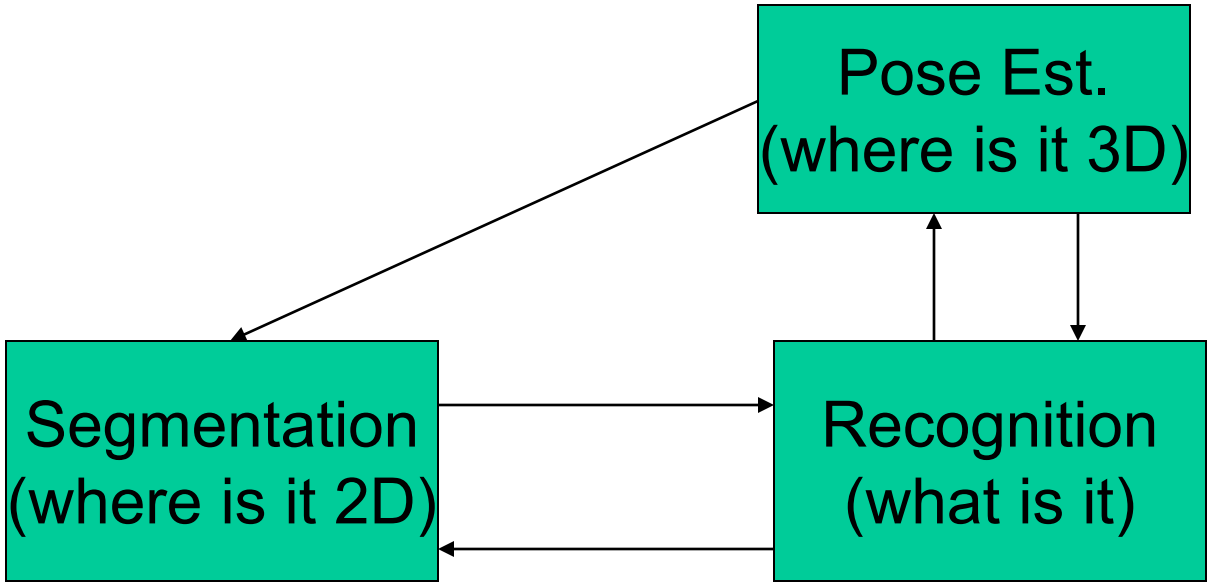
Image parsing/Semantic Segmentation



Object Recognition: The Problem

Given: A database D of "known" objects and an image I:

1. Determine which (if any) objects in D appear in I
2. Determine the pose (rotation and translation) of the object



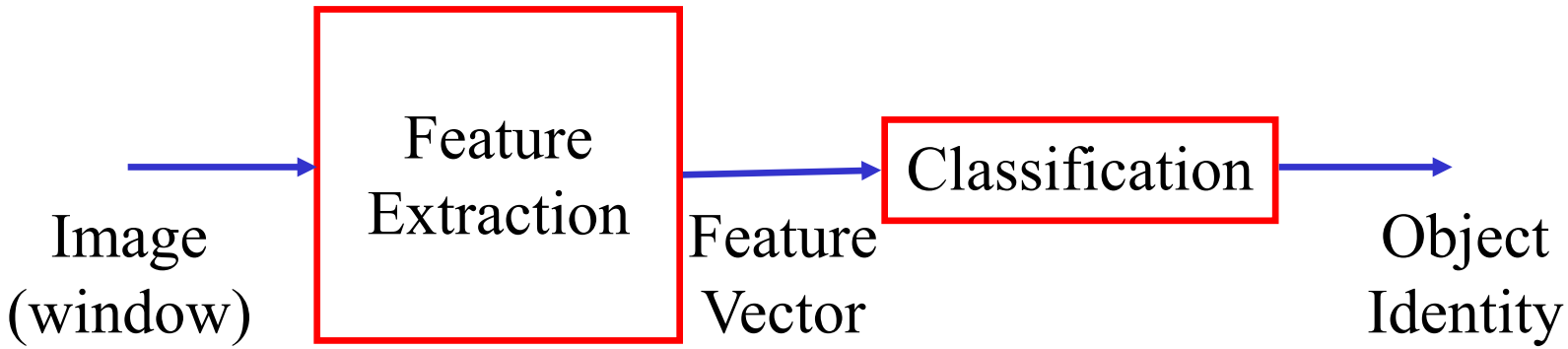
WHAT AND WHERE!!!

Recognition Challenges

- Within-class variability
  - Different objects within the class have different shapes or different material characteristics
  - Deformable
  - Articulated
  - Compositional

- Pose variability:
  - 2-D Image transformation (translation, rotation, scale)
  - 3-D Pose Variability and perspective projection
- Lighting
  - Direction (multiple sources & type)
  - Color
  - Shadows
- Occlusion – partial occlusion, self occlusion
- Clutter in background -> false positives

Sketch of a Pattern Recognition Architecture



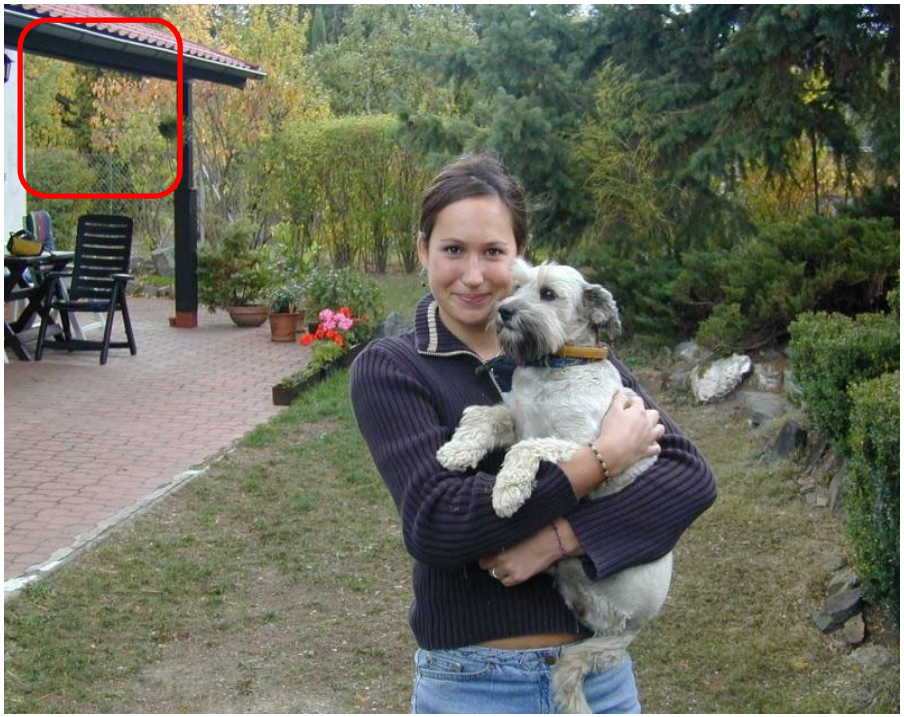
Features might be:

- Histograms of colors
- SIFT descriptors
- Outputs of Convolutional Neural Network
- …

Classifiers might be:

- Nearest Neighbor
- Bayesian classifier
- Logistic regression
- Support vector machine
- Softmax
- …

Sliding window approach to face detection

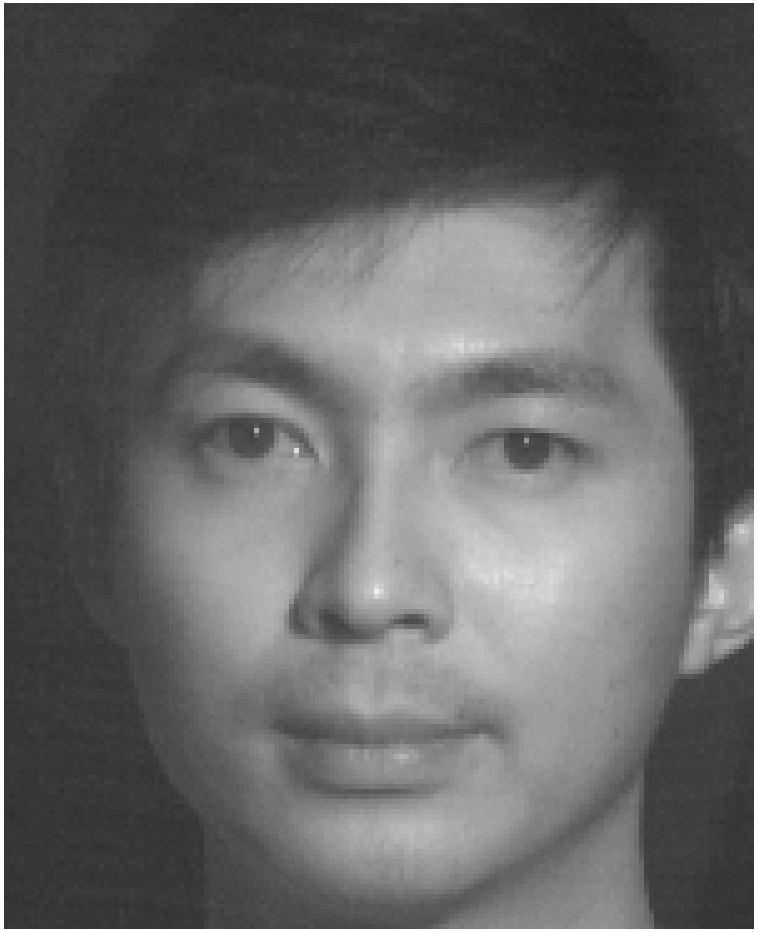


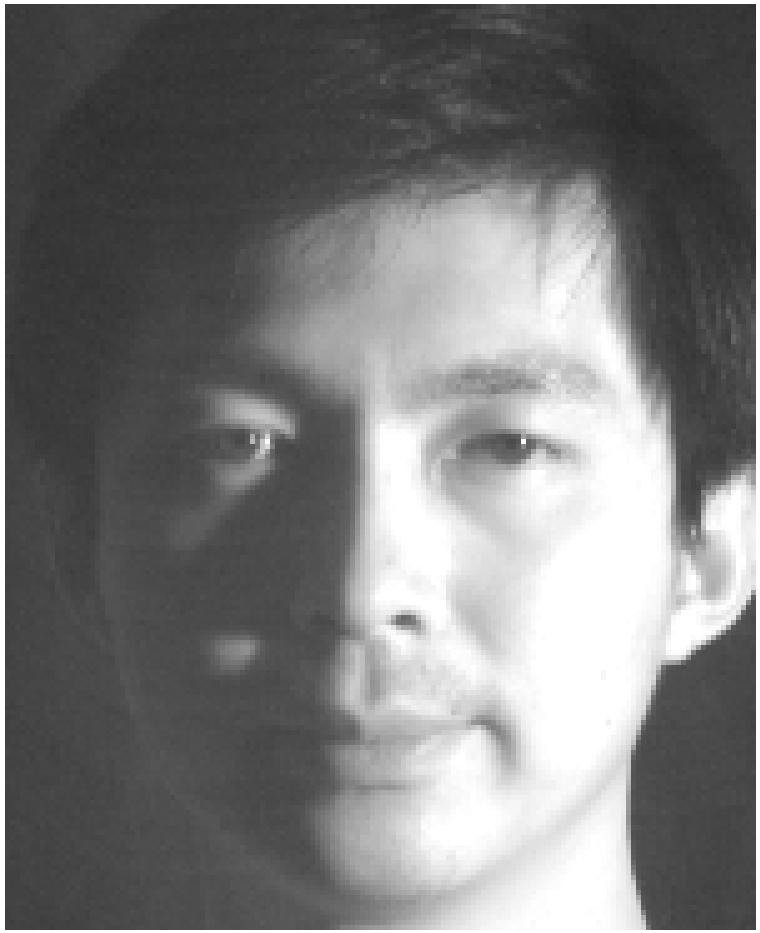
Binary classification example: Face Detection

- Scan window over image.
- Classify window as either:
  - Face
  - Non-face

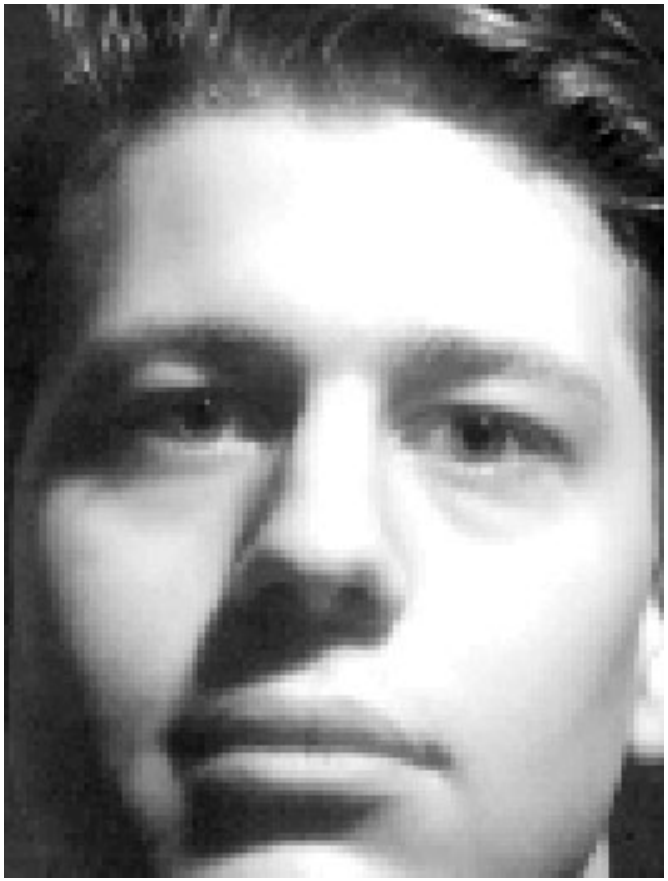
See for example, Viola-Jones face detector in OpenCV

Binary classification example: are these two faces from the same person?



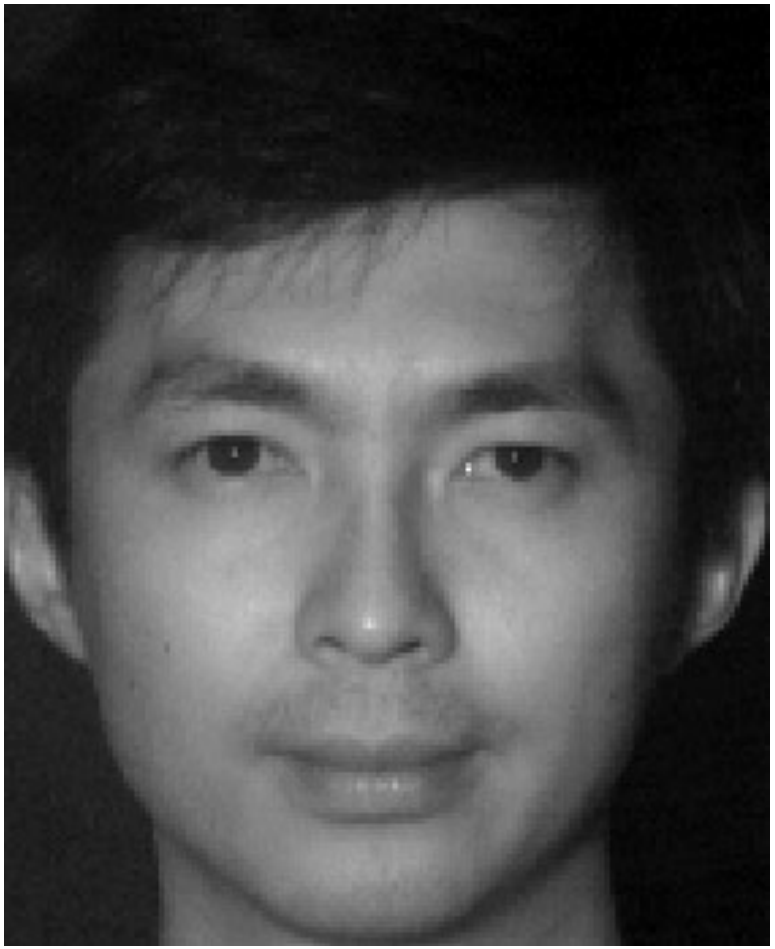


Binary classification example: are these two faces from the same person?



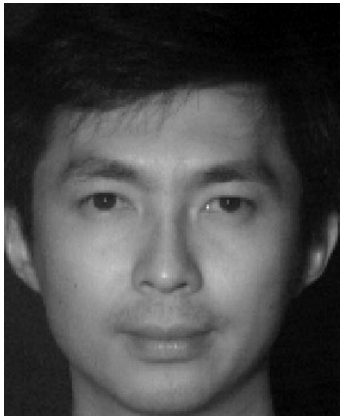
Binary classification example: are these two faces from the same person?

Binary classification example: are these two faces from the same person?



Evaluating a Binary Classifier

Same face or different face?



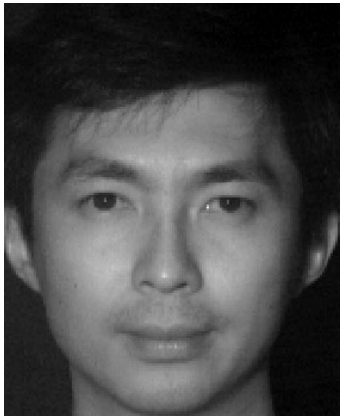
		Classifier Output	
		Same	**Different**
Truth	**Same**	True Positive Rate	False Negative Rate
	Different	False Positive Rate	True Negative Rate

True Positive Rate (TPR) + False Negative Rate (FNR) = 1
False Positive Rate (FPR) + True Negative Rate (TNR) = 1

i.e., sum of rows is 1

Evaluating a Binary Classifier

Same face or different face?



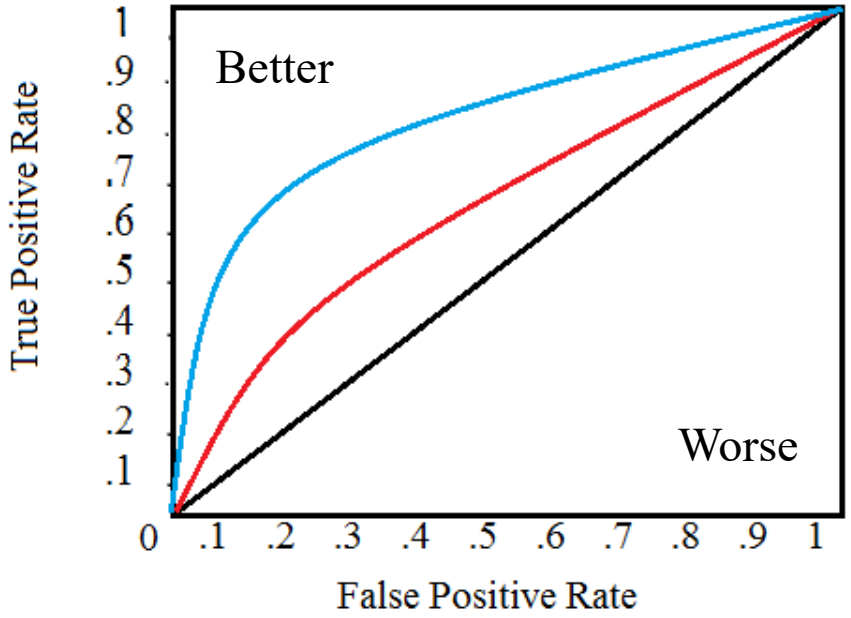
		Classifier Output	
		Same	**Different**
Truth	**Same**	0.98	0.02
	Different	0.05	0.95

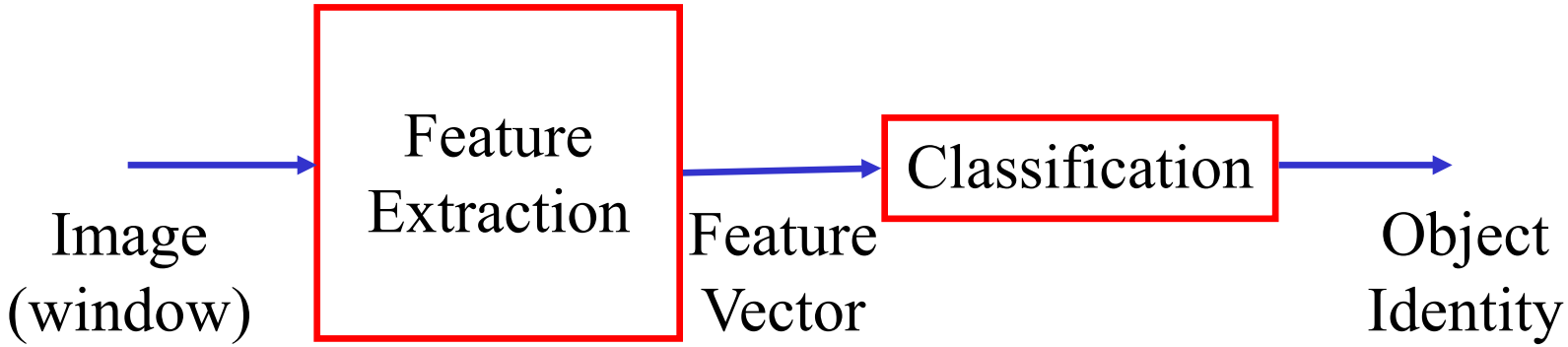
True Positive Rate (TPR) + False Negative Rate (FNR) = 1
False Positive Rate (FPR) + True Negative Rate (TNR) = 1

i.e., sum of rows is 1

ROC Curve: Evaluating a binary classifier

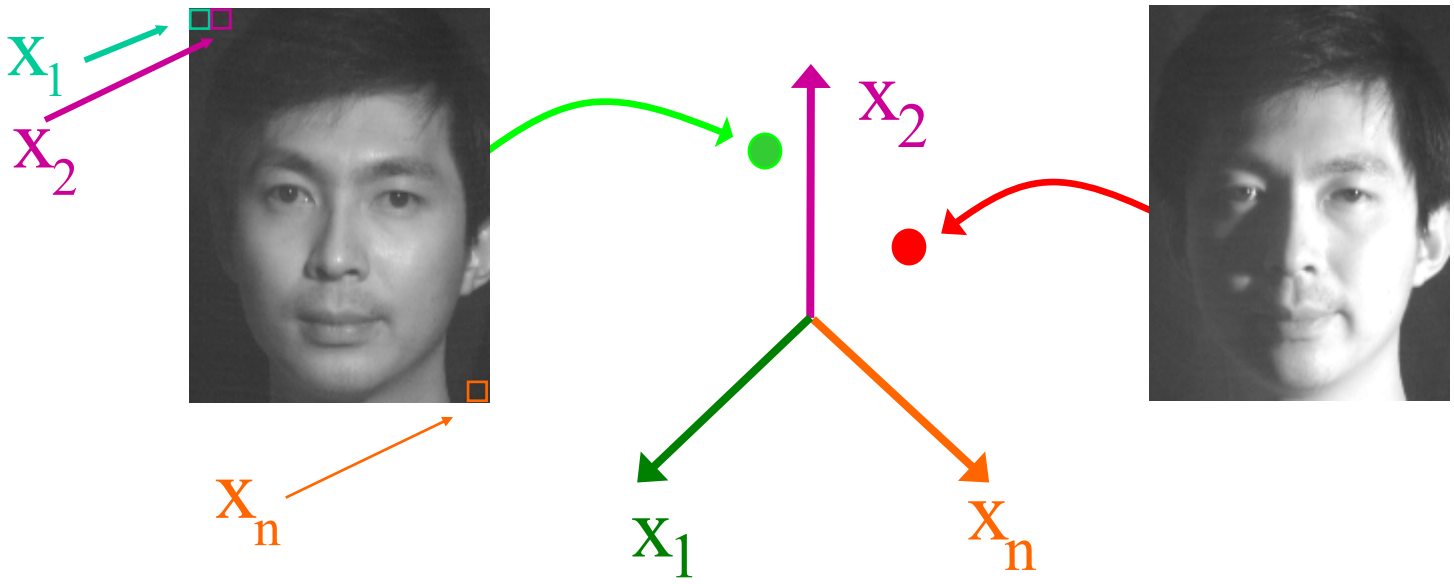
- For a face detector, there are two types of errors:
  - False Positives (e.g., non-face is detected as a face)
  - False Negatives (e.g., face is missed)
- Most classifiers have a **threshold** that can be varied, which leads to
- ROC Curve (Receiver Operator Characteristic) - Plot of tradeoff between False Positive Rate and True Positive Rate as a result of varying the threshold.





- So, what are the features?
- So, what is the classifier?

Simplest feature: Image as a Feature Vector



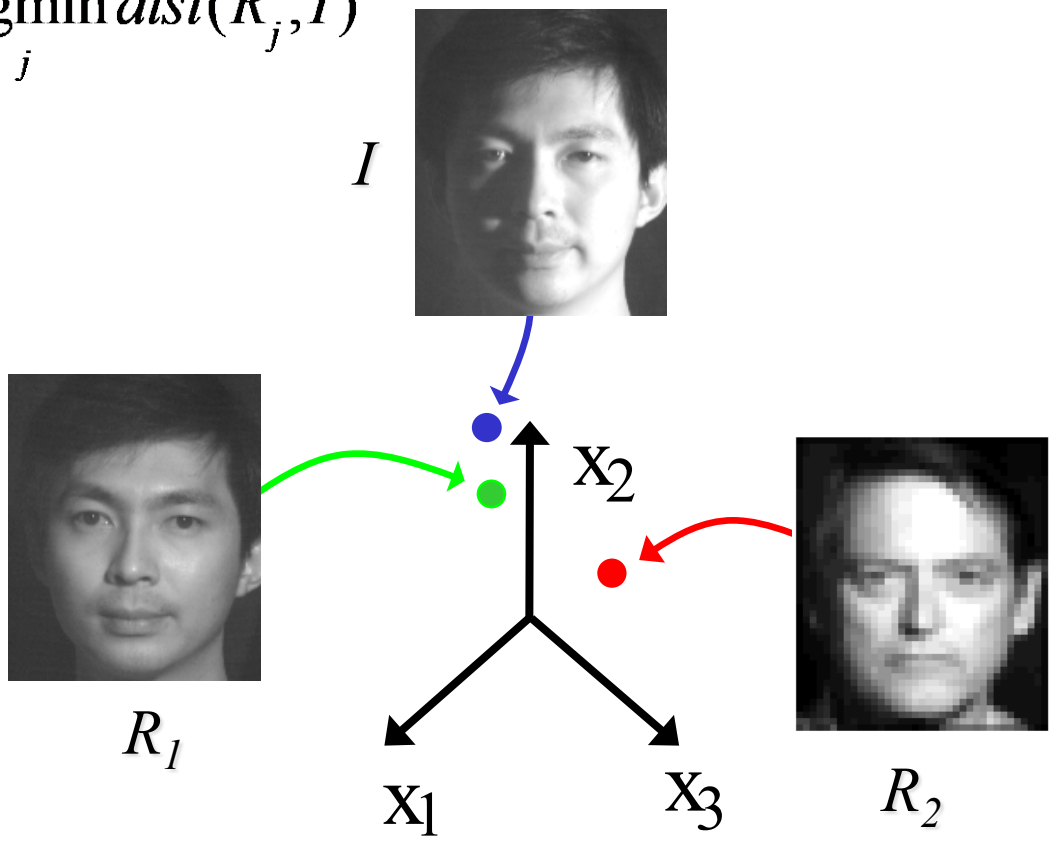
- Consider an n-pixel image to be a point in an n-dimensional space, $\mathbf{x} \in \mathbf{R}^n$
- Each pixel value $\mathbf{x_i}$ is a coordinate of $\mathbf{x}$

Nearest Neighbor Classifier

n classes (different people)

$\{R_j : j=1,\ldots, n\}$: set of training images, one per person

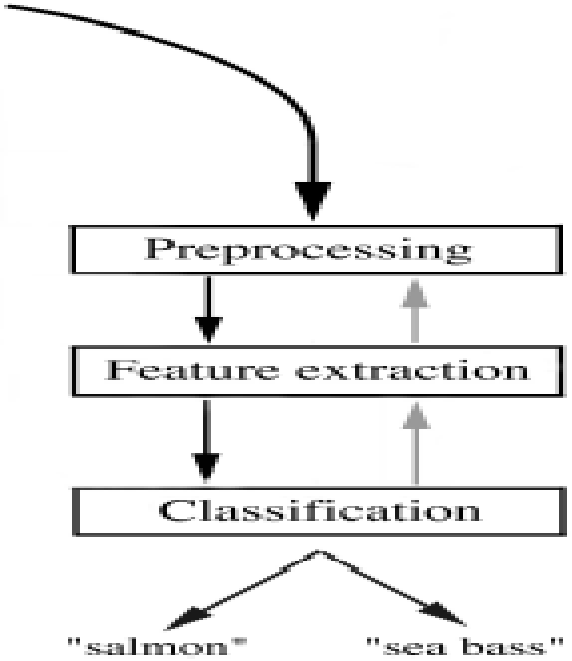
$$label = \underset{j}{\text{argmin}}\, dist(R_j, I)$$



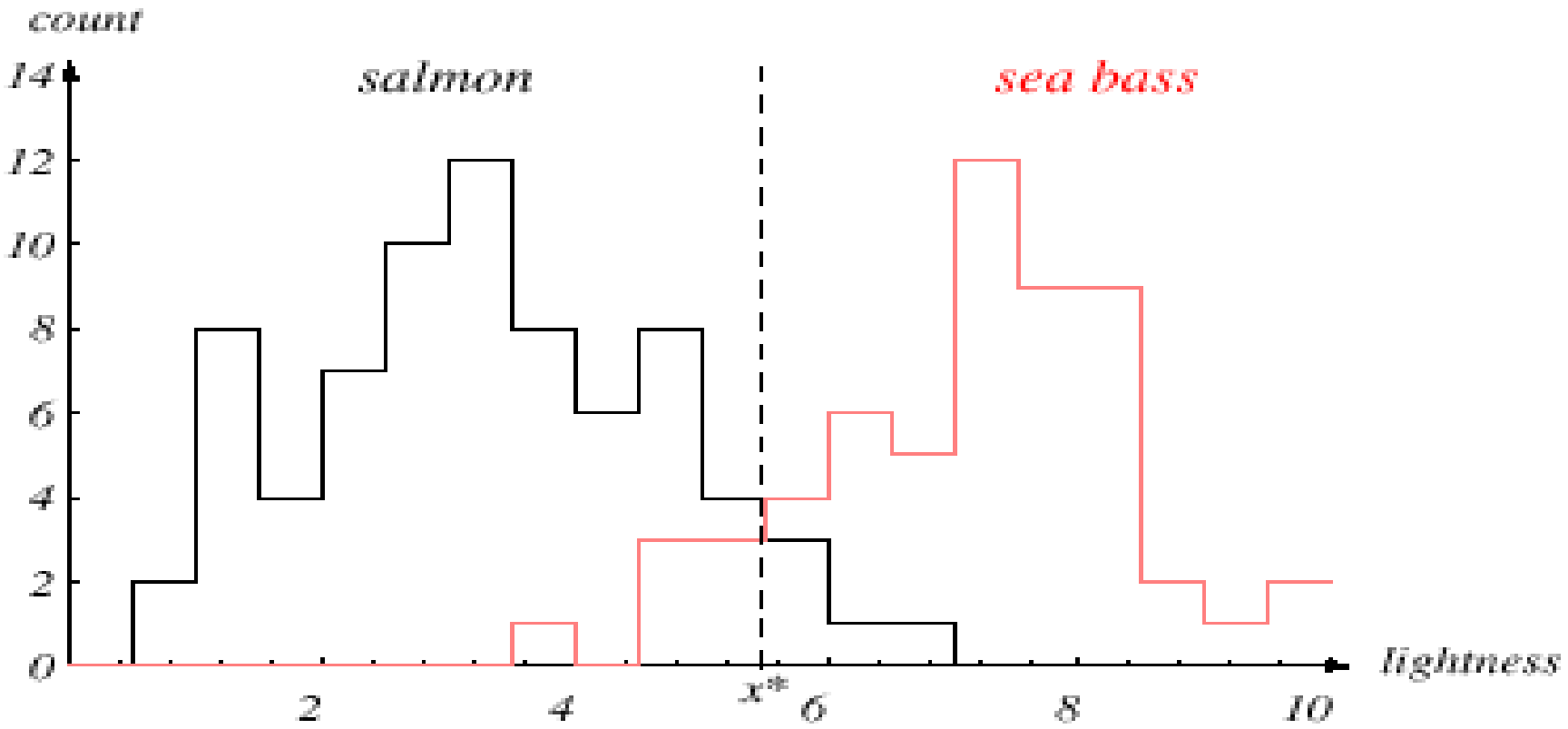
Your summer internship

Bayesian Classification Example

- Sorting incoming Fish on a conveyor according to species using optical sensing

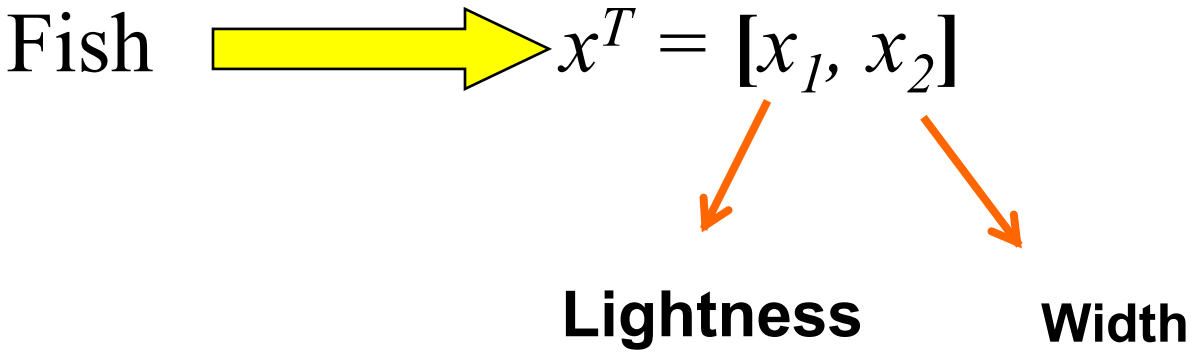


Bayesian Classification Example: salmon or sea bass

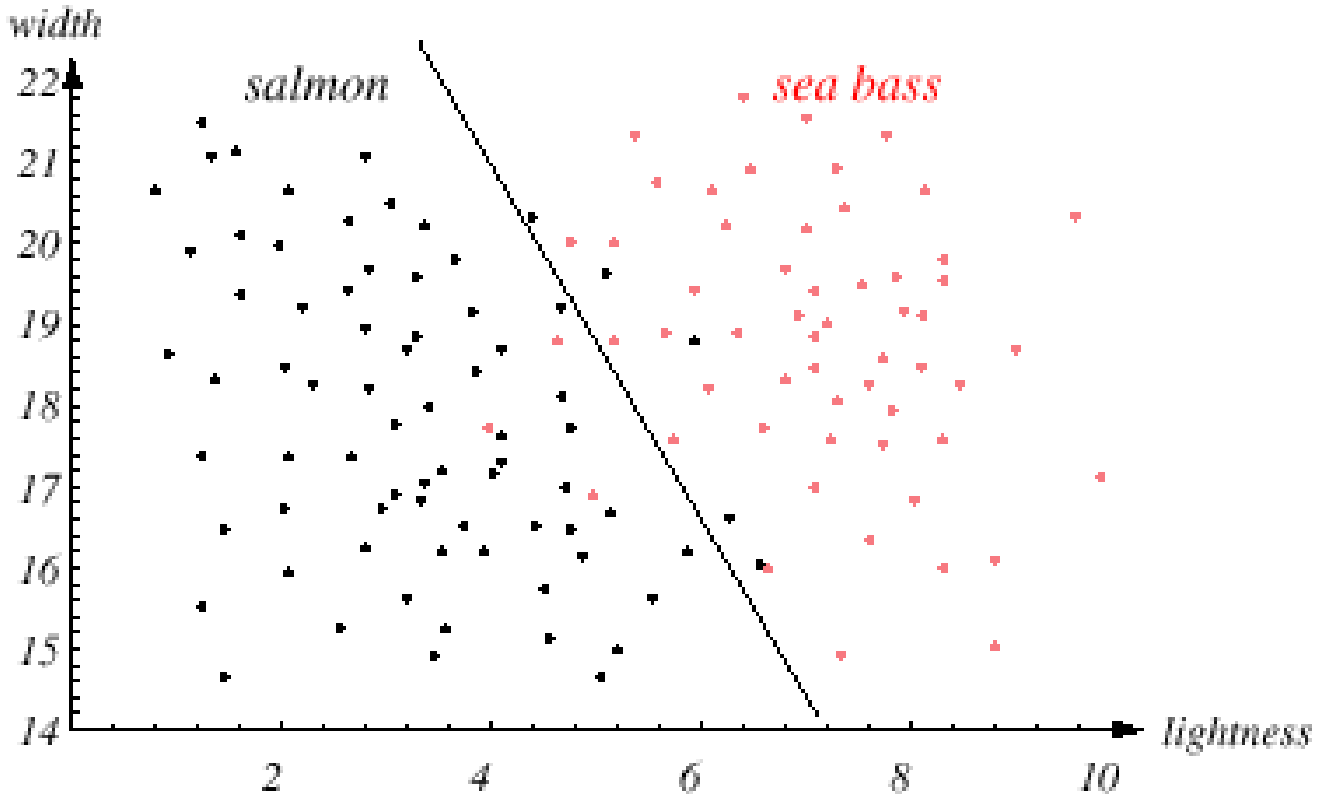


Bayesian Classification Example: salmon or sea bass

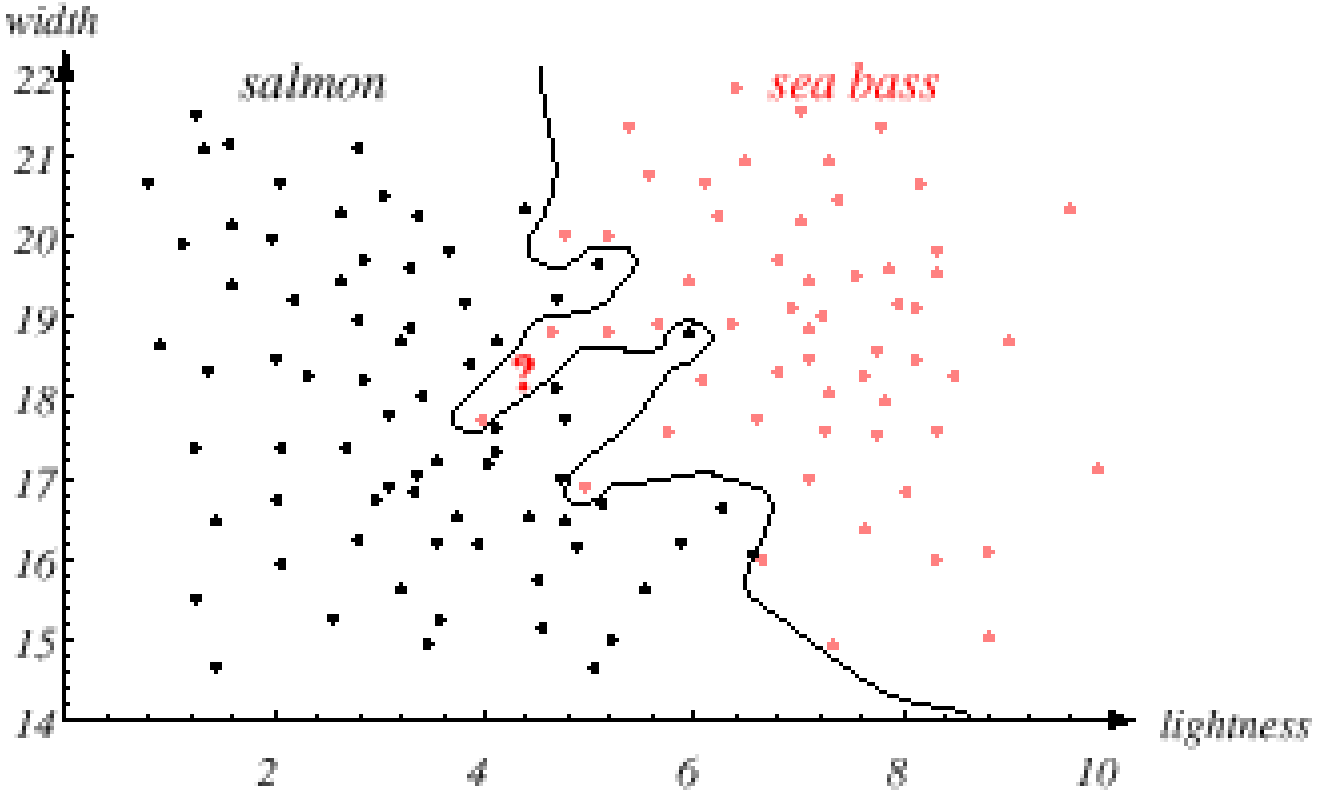
- Adopt the lightness and add the width of the fish



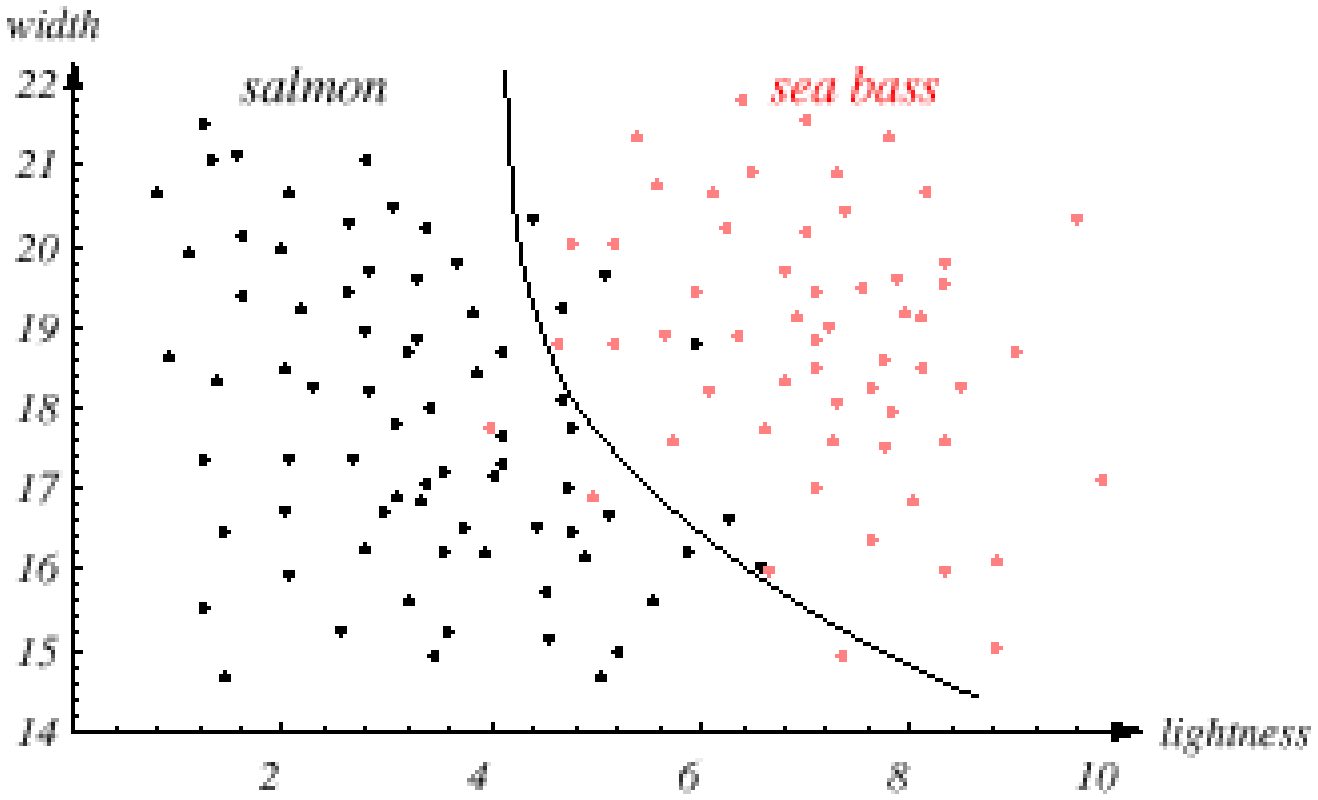
Bayesian Classification Example: salmon or sea bass



Bayesian Classification Example: salmon or sea bass



Bayesian Classification Example: salmon or sea bass



Bayesian Classification

- The sea bass/salmon example

  - State of nature is a random variable, ω_I
    - ω_1 – *the fish is a salmon*
    - ω_2 – *the fish is a sea bass*

  - *Prior Probabilities*
    - $P(\omega_1)$, $P(\omega_2)$

    - $P(\omega_i) > 0$
    - $P(\omega_1) + P(\omega_2) = 1$ (exclusivity and exhaustivity)

  - *Example prior: bass & salmon are equally likely*
    - $P(\omega_1) = P(\omega_2) = ½$ (uniform priors)

Bayesian Classification

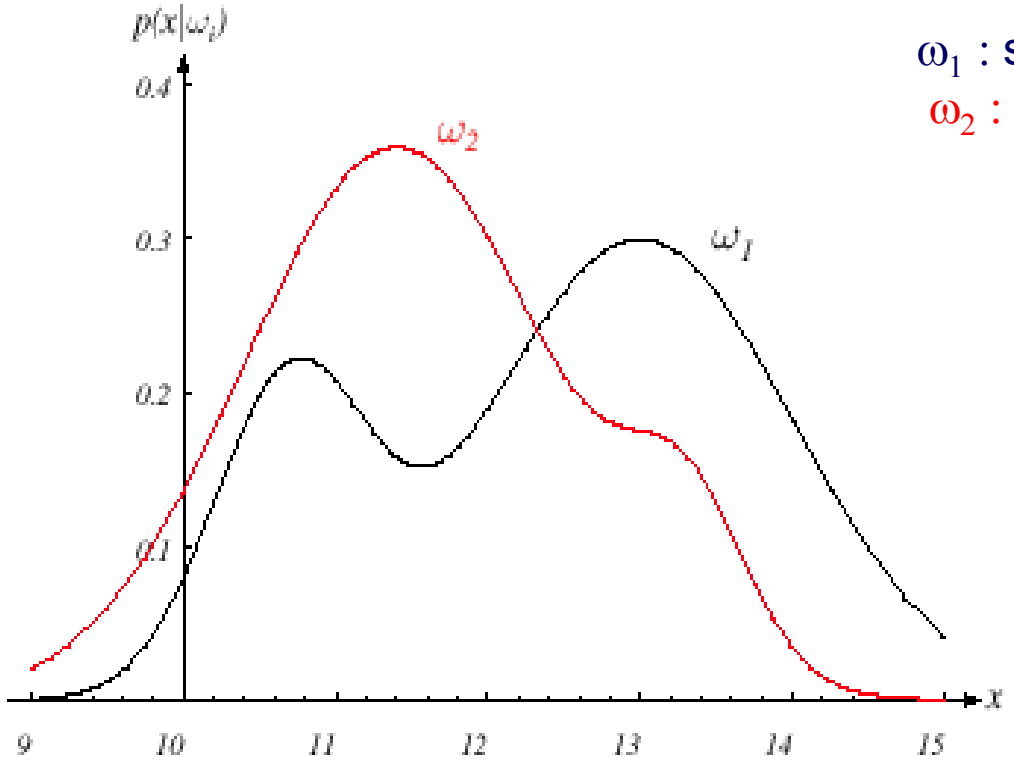
- Decision rule with only the prior information
  - Decide ω_1 if $P(\omega_1) > P(\omega_2)$ otherwise decide ω_2

- Use of the class–conditional information

- *Let* $\boldsymbol{x}$ *be a vector of features*

- $P(\boldsymbol{x} \mid \omega_1)$ and $P(\boldsymbol{x} \mid \omega_2)$ describe the distribution in features (say lightness) between populations of sea-bass and salmon.

Likelihood: $P(x \mid \omega_i)$



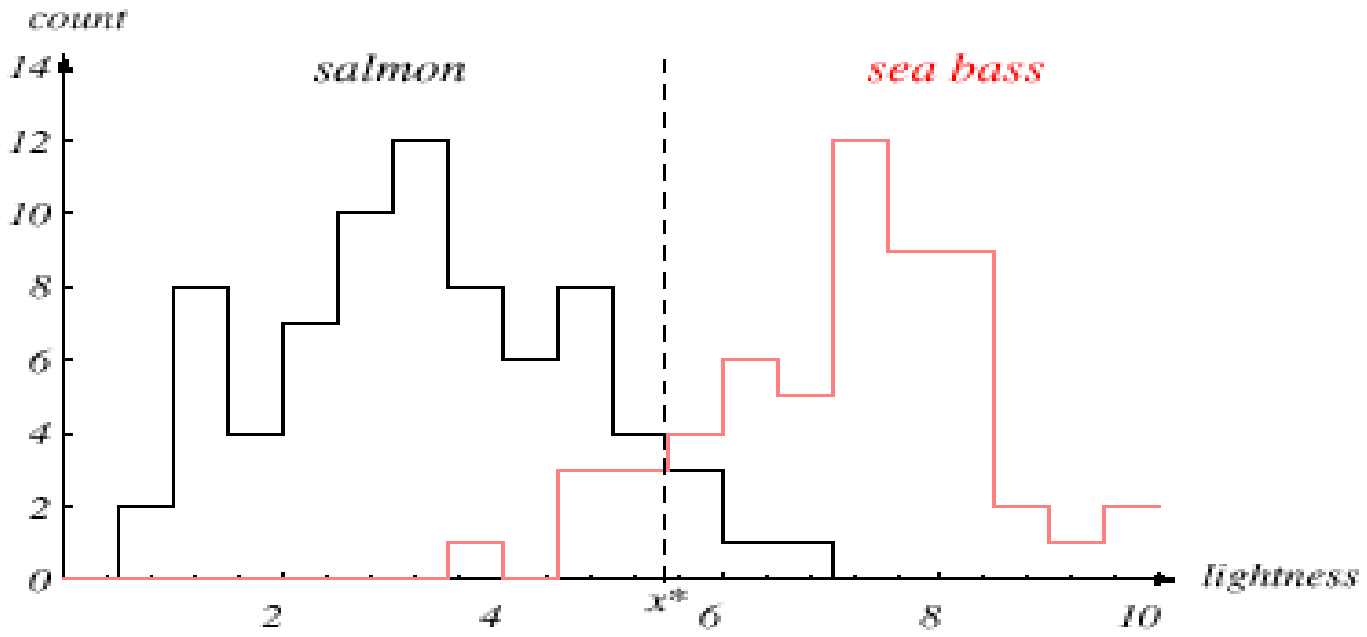
ω_1 : salmon

ω_2 : sea bass

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

Remember this histogram?

If the histogram is normalized by dividing by the number of samples, the result is sampled $P(x \mid \omega_i)$



Computing the Posterior

- Posterior $P(\omega_j \mid x)$, likelihood $P(x \mid \omega_j)$, prior $P(x)$

$$P(\omega_j \mid x) = \frac{P(x \mid \omega_j)P(\omega_j)}{P(x)} \qquad \textbf{(BAYES RULE)}$$

where in case of two categories the probability density P(x) is

$$P(x) = \sum_{j=1}^{j=2} P(x \mid \omega_j)P(\omega_j)$$

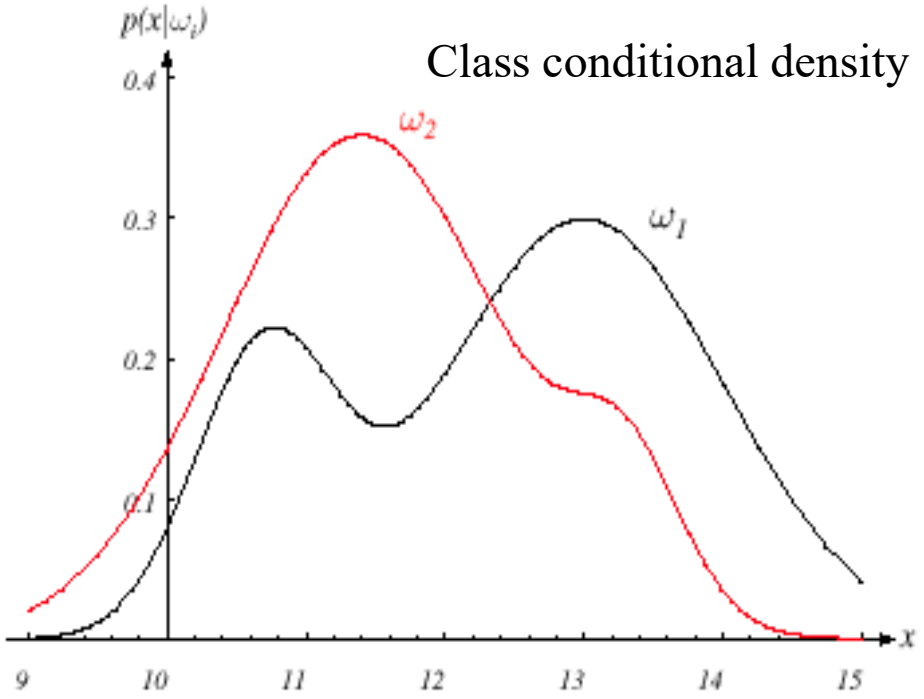
  - Given a measurement x, one can evaluate $P(\omega_j \mid x)$ for all classes

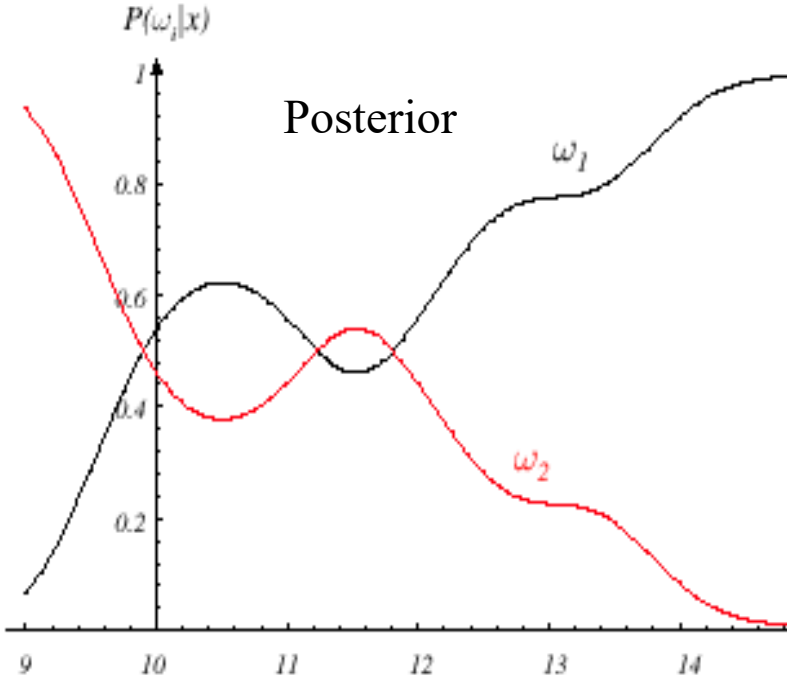
Maximum a posteriori classifier (MAP)

$$g_j(x) = P(\omega_j \mid x) = \frac{P(x \mid \omega_j) P(\omega_j)}{P(x)}$$

$P(x|\omega_i)$: Class conditional density
$P(\omega_i)$: Prior of class j

Classification: $\hat{j} = \arg\max_j g_j(x)$





Bayesian Classification

- Think of a classifier as set of scalar functions $g_i(\mathbf{x})$ – one for each class *i* – that assigns a score to the feature vector **x,** and then choses the class *i* with the highest score

- The Bayesian classifier assigns a score based on the *a posteriori* probabilities: $g_i(\mathbf{x}) = P(\omega_i \mid \mathbf{x})$

Decision Regions and Decision Boundary

Decision region R_i of class ω_i

R_i is the region of the feature space that will be classified as class ω_i

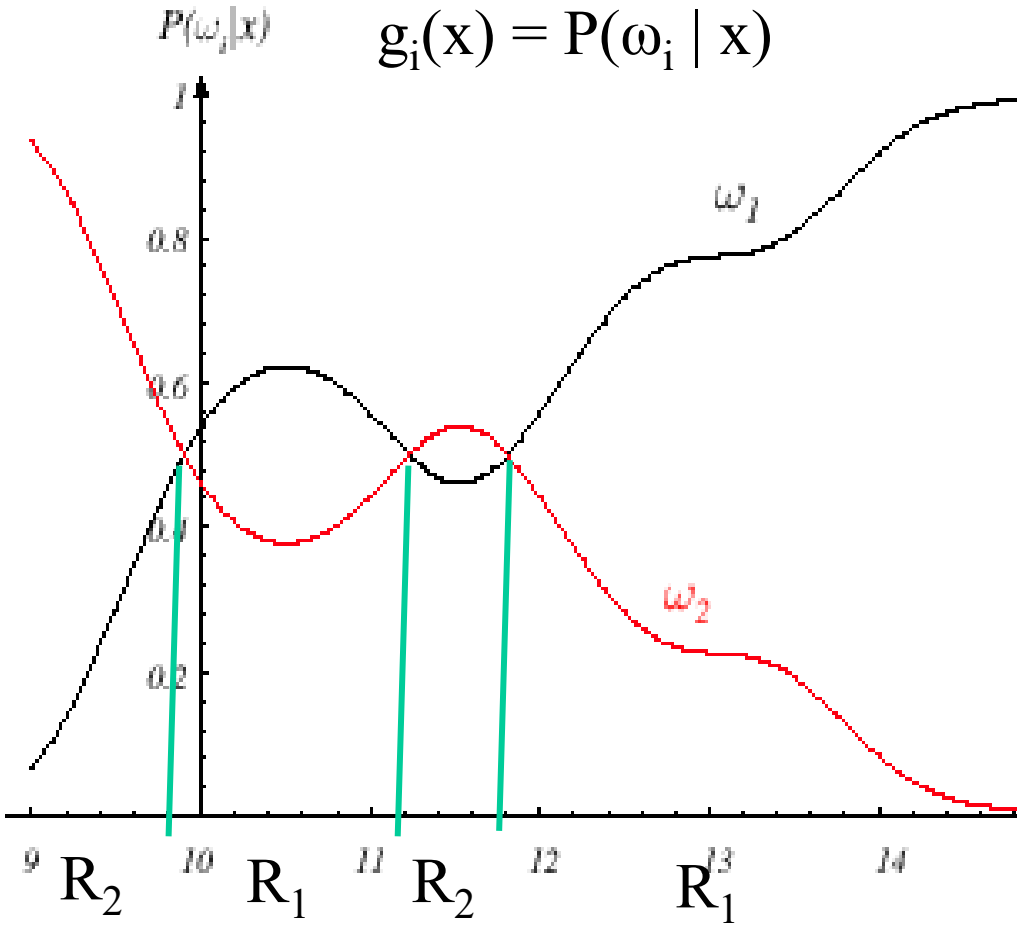
$$R_i = \{\mathbf{x} : g_i(\mathbf{x}) > g_j(\mathbf{x}), \forall j, i \neq j\}$$

Decision boundary between decision regions between R_i and R_j for classes ω_i and ω_j

$$H_{ij} = \{\mathbf{x} : g_i(\mathbf{x}) = g_j(\mathbf{x}), g_i(\mathbf{x}) > g_k(\mathbf{x}), \ \forall k, i \neq k\}$$

Decision Regions and Boundaries

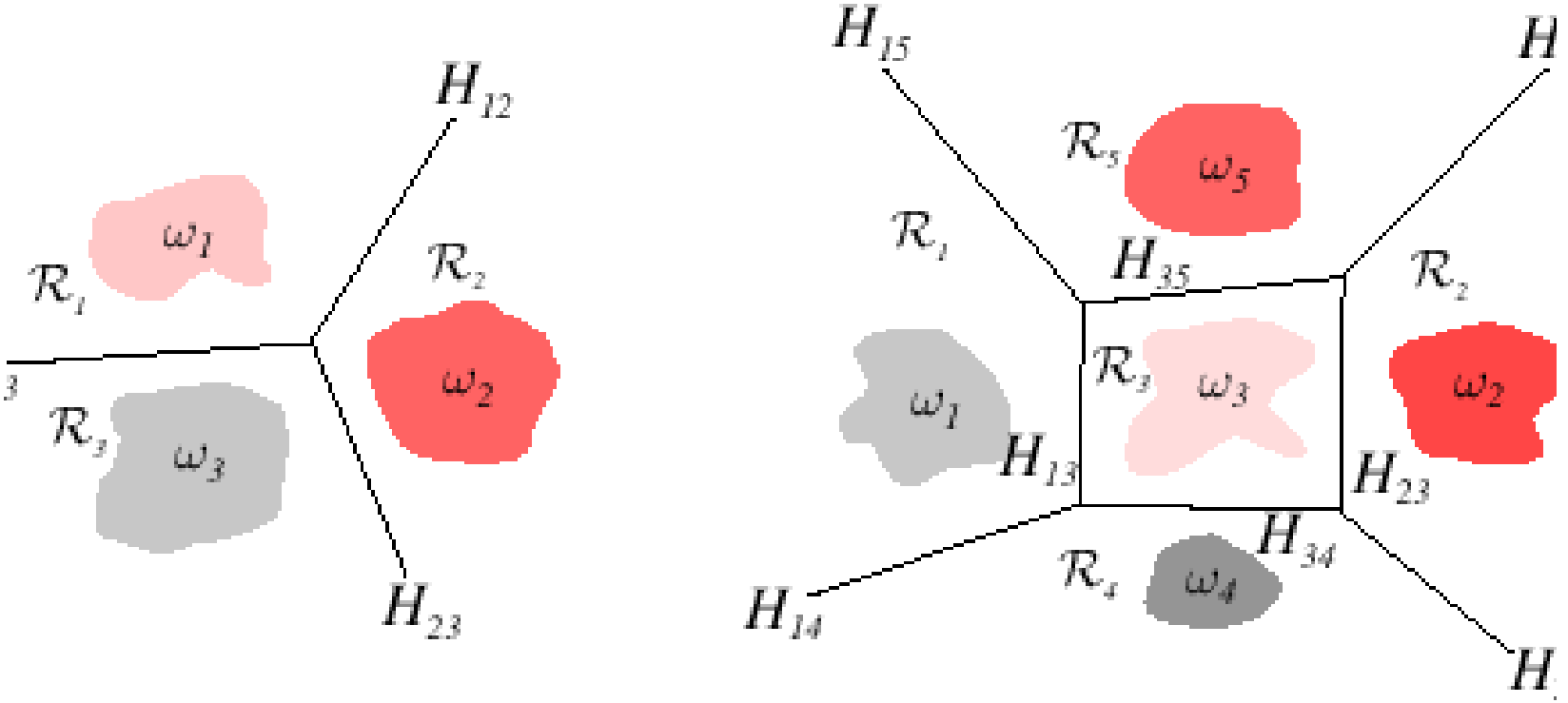
$g_i(x) = P(\omega_i \mid x)$



$$R_i = \{\mathbf{x} : g_i(\mathbf{x}) > g_j(\mathbf{x}), \forall j, i \neq j\}$$

$$H_{ij} = \{\mathbf{x} : g_i(\mathbf{x}) = g_j(\mathbf{x}), g_i(\mathbf{x}) > g_k(\mathbf{x}), \ \forall k, i \neq k\}$$

Multiclass – 2D Feature Space



E 5.4. Decision boundaries produced by a linear machine for a three-class
d a five-class problem. From: Richard O. Duda, Peter E. Hart, and David G.
Classification. Copyright © 2001 by John Wiley & Sons, Inc.

Evaluating Multi-class classifiers

- Confusion matrix
  - True positive (TP): diagonal
  - False negative (FN): along row, excluding diagonal
  - False positive (FP): along column, excluding diagonal

	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck	Actual	Recall
airplane	**880**	6	40	3	3	2	3	9	8	6	960	91.67%
automobile	0	**982**	0	2	1	11	2	0	1	20	1019	96.37%
bird	4	1	**954**	6	9	3	19	1	3	3	1003	95.11%
cat	2	1	9	**935**	6	29	18	4	1	9	1014	92.21%
deer	2	0	7	17	**920**	5	9	28	2	1	991	92.84%
dog	0	0	8	86	8	**895**	15	10	1	3	1026	87.23%
frog	0	2	3	7	3	0	**993**	0	0	1	1009	98.41%
horse	0	1	4	3	6	5	3	**956**	0	3	981	97.45%
ship	17	7	0	2	1	0	4	0	**940**	16	987	95.24%
truck	9	12	1	1	0	37	1	2	5	**942**	1010	93.27%
Predicted	914	1012	1026	1062	957	987	1067	1010	961	1004	**10000**	93.98%
Precision	96.28%	97.04%	92.98%	88.04%	96.13%	90.68%	93.06%	94.65%	97.81%	93.82%	94.05%	**93.9...**

$$\text{Recall} = \frac{tp}{tp + fn}$$

Total population

Average recall

$$\text{Precision} = \frac{tp}{tp + fp}$$

Average precision

$$\frac{\text{Total true positives}}{\text{Total population}}$$

- The Bayes decision rule is optimal in that it minimizes the probability of error.

- The challenge – To implement it you need:
  - $P(\omega_j)$ – this is pretty easy to get
  - $P(\mathbf{x} \mid \omega_j)$ – this is really hard to measure and approximate, especially when x is higher than a few dimensions.

- What can we do?
  - Parametric form for $P(\mathbf{x} \mid \omega_j)$
  - K-th nearest neighbor classifier
  - Learn decision rules $g_i(x)$ directly, rather than trying to model and estimate $P(x \mid \omega_j)$ and using Bayes.
  - Reduce dimension of **x**

Plug-in classifiers

- Assume that class conditional distributions $P(x|\omega_i)$ have some parametric form with parameters W
- Given training samples x for class ω_i, estimate parameters W

- Common parametric form:
  - Uniform over region
  - Normal distribution with shared covariance matrix, different means
  - Normal distribution but with different covariance matrices

Support Vector Machines

- Bayes classifiers and generative approaches in general try to model of the posterior, $p(\omega_i|\mathbf{x})$
- Instead, try to obtain the decision boundary directly
  - Potentially easier, because we need to encode only the geometry of the boundary, not any irrelevant wiggles in the posterior.
  - Not all points affect the decision boundary

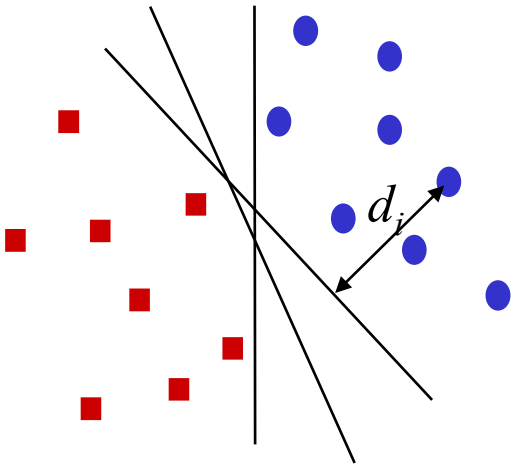
 [From Marc Polleyfes]

Support Vector Machines

- Set S of points $x_i \in R^n$, each x_i belongs to one of two classes $y_i \in \{-1, 1\}$
- The goals is to find a hyperplane that divides S in these two classes

S is separable if $\exists w \in R^n, b \in R$

$$y_i\left(w^T x_i + b\right) \geq 1$$



Separating hyperplanes

$$w^T x + b = 0$$

$$d_i = \frac{w^T x_i + b}{w} \qquad w = \|w\|$$

Closest point

$$y_i d_i = \frac{1}{w}$$

[From Marc Polleyfes]

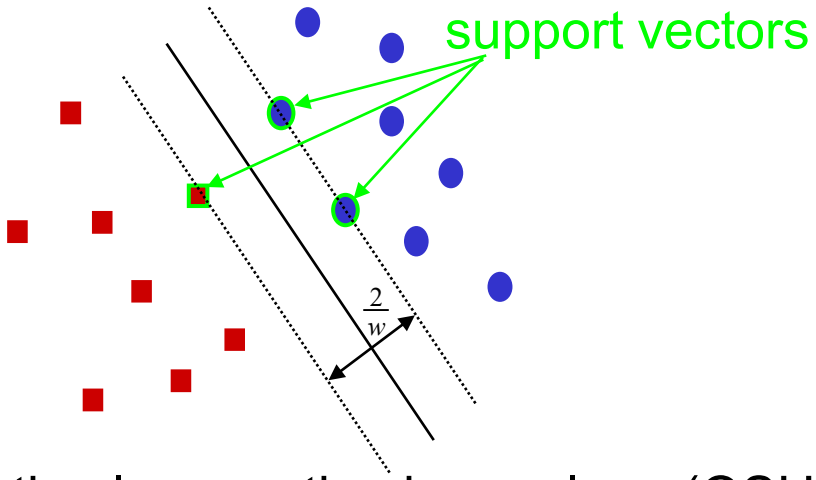
Support Vector Machines

- Optimal separating hyperplane maximizes $\frac{1}{w}$

Problem 1:

Minimize $\frac{1}{2}\mathrm{w}^{\mathrm{T}}\mathrm{w}$

Subject to $y_i\left(\mathrm{w}^{\mathrm{T}}\mathrm{x}_i + b\right) \geq 1, \;\; i = 1,2,\ldots,N$



Optimal separating hyperplane (OSH)

Solve using Lagrange multipliers

- Lagrangian

$$L(\mathrm{w}, b, \alpha) = \tfrac{1}{2}\mathrm{w}^{\mathrm{T}}\mathrm{w} - \sum_{i=1}^{N} \alpha_i \left\{ y_i \left(\mathrm{w}^{\mathrm{T}} \mathrm{x}_i + b \right) - 1 \right\}$$

  - at solution

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{N} y_i \alpha_i = 0$$

$$\frac{\partial L}{\partial \mathrm{w}} = \mathrm{w} - \sum_{i=1}^{N} \alpha_i y_i \, \mathrm{x}_i = 0$$

$$L(\boldsymbol{\alpha}) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \, \mathrm{x}_i . \mathrm{x}_j$$

  - therefore

$$\boldsymbol{\alpha} \geq 0$$

 [From Marc Polleyfes]

Decision function

- Once $\mathbf{w}$ and b have been computed the classification decision for input $\mathbf{x}$ is given by

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}^T \mathbf{x} + b)$$

- Note the globally optimal solution can always be obtained (convex problem)

[From Marc Polleyfes]

Bag-of-features models

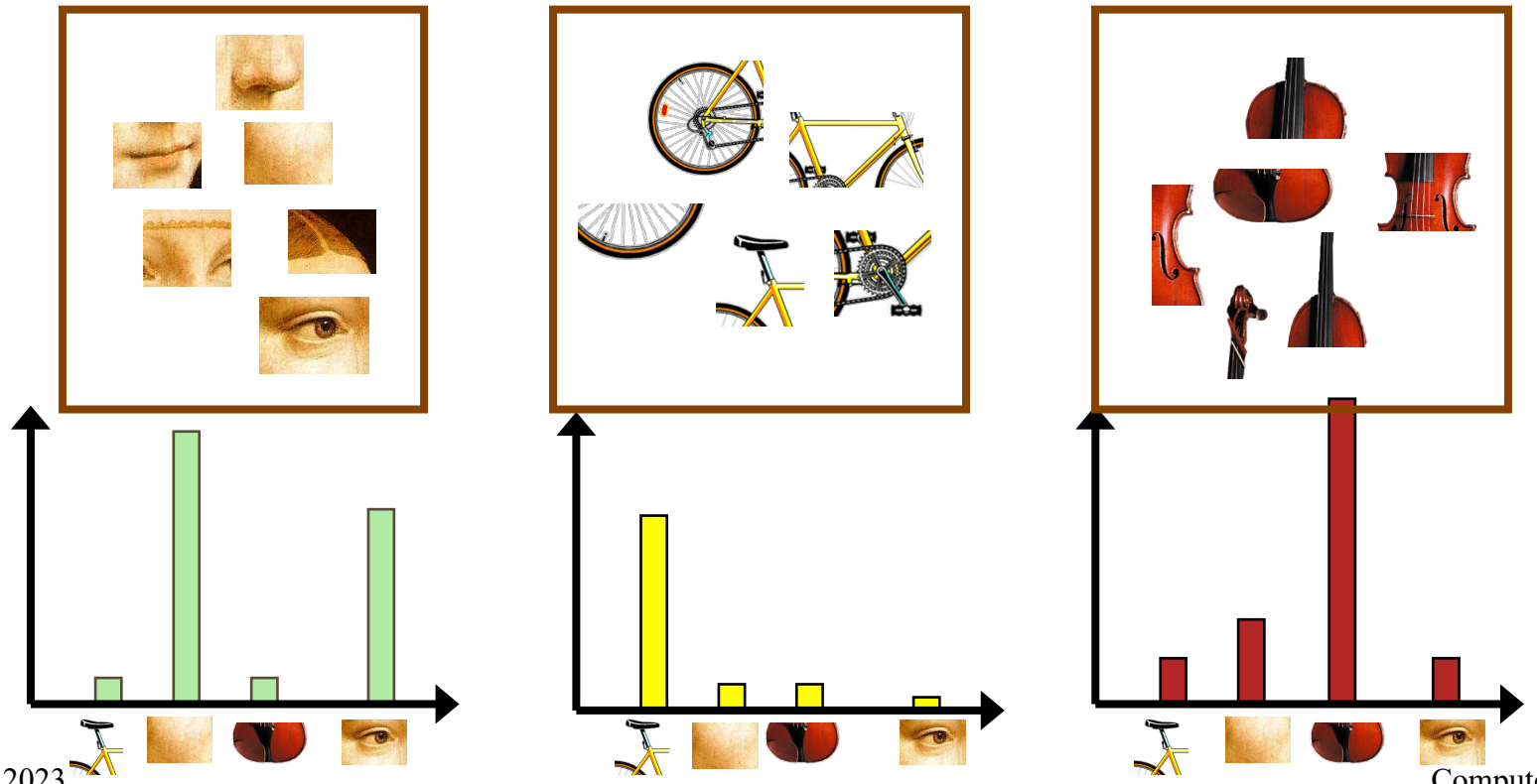
Object → **Bag of 'words'**

Slides from Svetlana Lazebnik who borrowed from others

Bag-of-features models

Bag-of-features steps

1. Extract features
2. Learn “visual vocabulary”
3. Quantize features using visual vocabulary
4. Represent images by frequencies (histogram) of “visual words”
5. Recognition using histograms as input to classifier



Feature extraction

- Regular grid or interest regions

Pattern Classification Summary

- Supervised vs. Unsupervised: Do we have labels?
- Supervised
  - Nearest Neighbor
  - Bayesian
    - Plug in classifier
    - Distribution-based
    - Projection Methods (Fisher's, LDA)
  - Neural Network
  - Support Vector Machine
  - Kernel methods
- Unsupervised
  - Clustering
  - Reinforcement learning

Next Lectures

- Recognition, detection, and classification