High-Level Computer Vision

- Detection of classes of objects (faces, motorbikes, trees, cheetahs) in images

- Recognition of specific objects such as George Bush or machine part #45732

- Classification of images or parts of images for medical or scientific applications

- Recognition of events in surveillance videos

- Measurement of distances for robotics

High-level vision uses techniques from AI

- Graph-Matching: A*, Constraint Satisfaction, Branch and Bound Search, Simulated Annealing

- Learning Methodologies: Decision Trees, Neural Nets, SVMs, EM Classifier

- Probabilistic Reasoning, Belief Propagation, Graphical Models

Graph Matching for Object Recognition

- For each specific object, we have a geometric model.
- The geometric model leads to a symbolic model in terms of image features and their spatial relationships.
- An image is represented by all of its features and their spatial relationships.
- This leads to a graph matching problem.

Model-based Recognition as Graph Matching

- Let U = the set of model features.
- Let R be a relation expressing their spatial relationships.
- Let L = the set of image features.
- Let S be a relation expressing their spatial relationships.
- The ideal solution would be a subgraph isomorphism f: U-> L satisfying
- if $(u_1, u_2, \ldots, u_n) \varepsilon R$, then $(f(u_1),f(u_2),\ldots,f(u_n)) \varepsilon S$

House Example

2D model

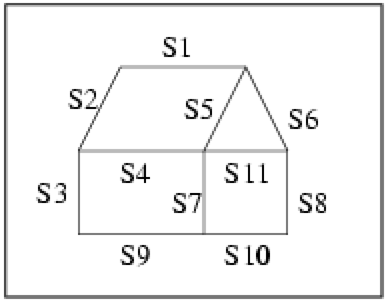


Image 1

P

2D image

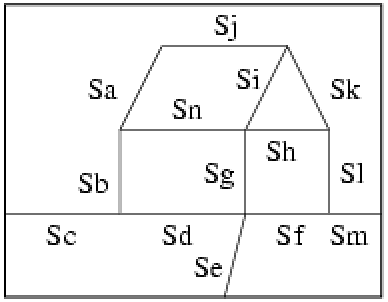


Image 2

L

RP and RL are connection relations.

P = {S1,S2,S3,S4,S5,S6,S7,S8,S9,S10,S11}.

L = {Sa,Sb,Sc,Sd,Se,Sf,Sg,Sh,Si,Sj,Sk,Sl,Sm}.

R_P = { (S1,S2), (S1,S5), (S1,S6), (S2,S3), (S2,S4), (S3,S4), (S3,S9), (S4,S5), (S4,S7), (S4,S11), (S5,S6), (S5,S7), (S5,S11), (S6,S8), (S6,S11), (S7,S9), (S7,S10), (S7,S11), (S8,S10), (S8,S11), (S9,S10) }.

R_L = { (Sa,Sb), (Sa,Sj), (Sa,Sn), (Sb,Sc), (Sb,Sd), (Sb,Sn), (Sc,Sd), (Sd,Se), (Sd,Sf), (Sd,Sg), (Se,Sf), (Se,Sg), (Sf,Sg), (Sf,Sl), (Sf,Sm), (Sg,Sh), (Sg,Si), (Sg,Sn), (Sh,Si), (Sh,Sk), (Sh,Sl), (Sh,Sn), (Si,Sj), (Si,Sk), (Si,Sn), (Sj,Sk), (Sk,Sl), (Sl,Sm) }.

f(S1)=Sj

f(S2)=Sa

f(S3)=Sb

f(S4)=Sn

f(S5)=Si

f(S6)=Sk

f(S7)=Sg

f(S8) = Sl

f(S9)=Sd

f(S10)=Sf

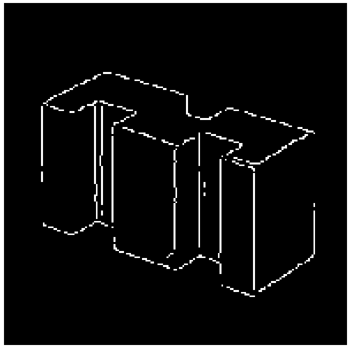
f(S11)=Sh

But this is too simplistic

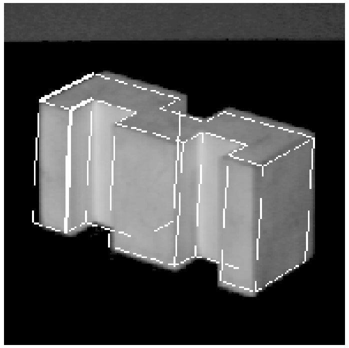
- The model specifies all the features of the object that may appear in the image.
- Some of them don’t appear at all, due to occlusion or failures at low or mid level.
- Some of them are broken and not recognized.
- Some of them are distorted.
- Relationships don’t all hold.

TRIBORS: view class matching of polyhedral objects

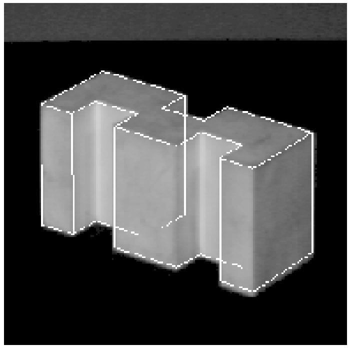
edges from image



model overlayed



improved location



- A view-class is a typical 2D view of a 3D object.

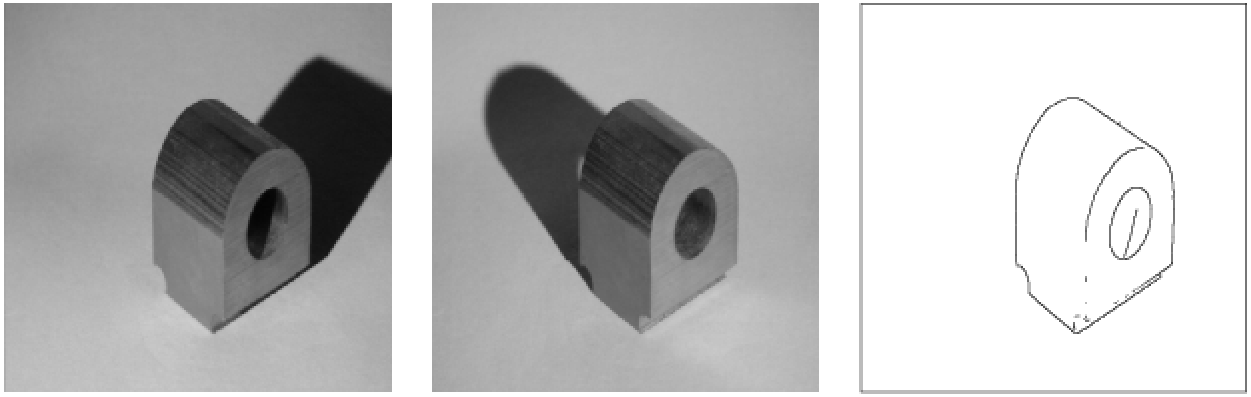
- Each object had 4-5 view classes (hand selected).

- The representation of a view class for matching included:
  - triplets of line segments visible in that class
  - the probability of detectability of each triplet

The first version of this program used iterative-deepening A* search.

RIO: Relational Indexing for Object Recognition

- RIO worked with more complex parts that could have
  - planar surfaces
  - cylindrical surfaces
  - threads

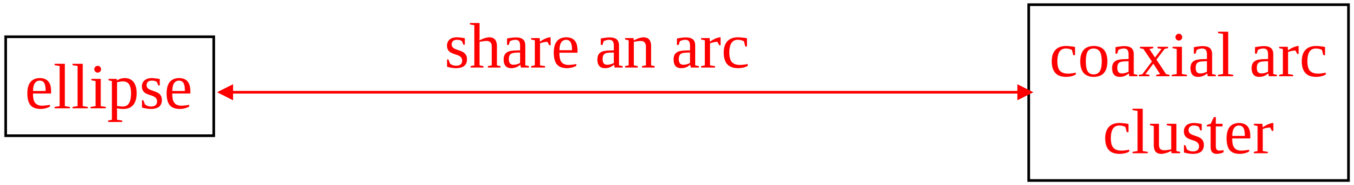


Object Representation in RIO

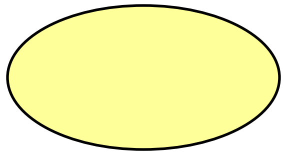
- 3D objects are represented by a 3D mesh and set of 2D view classes.

- Each view class is represented by an attributed graph whose nodes are features and whose attributed edges are relationships.

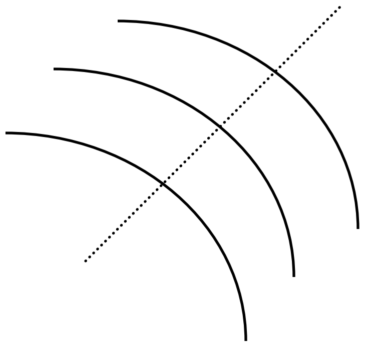
- For purposes of indexing, attributed graphs are stored as sets of 2-graphs, graphs with 2 nodes and 2 relationships.



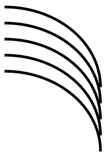
RIO Features



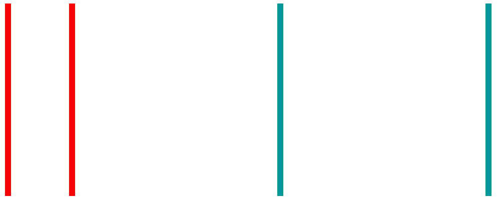
ellipses



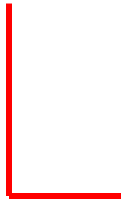
coaxials

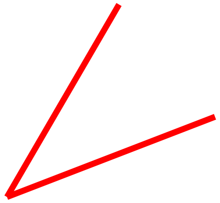


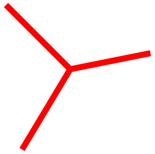
coaxials-multi



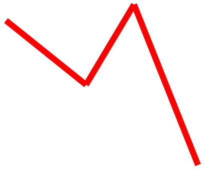
parallel lines
close and far

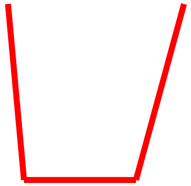






junctions

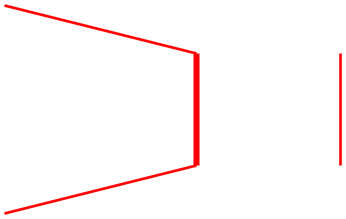




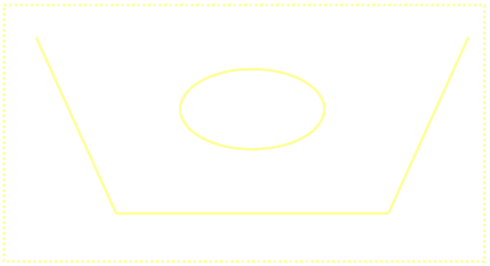
triples

L V Y Z U

RIO Relationships

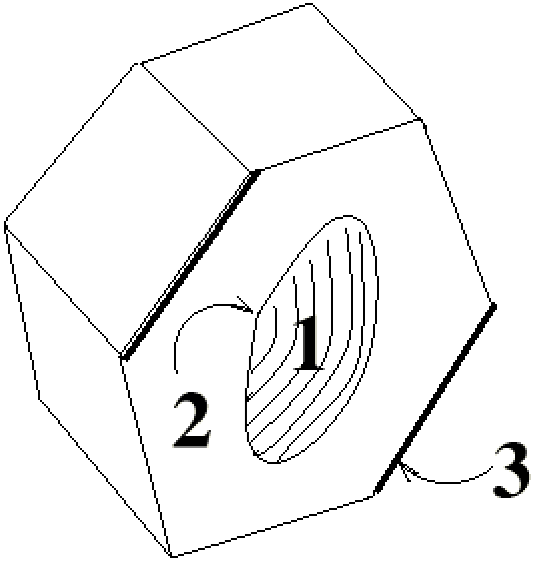


- share one arc
- share one line
- share two lines
- coaxial
- close at extremal points
- bounding box encloses / enclosed by



Hexnut Object

MODEL-VIEW



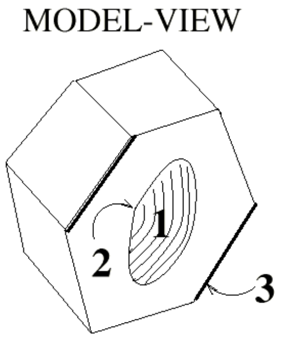
RELATIONS:
a: encloses
b: coaxial

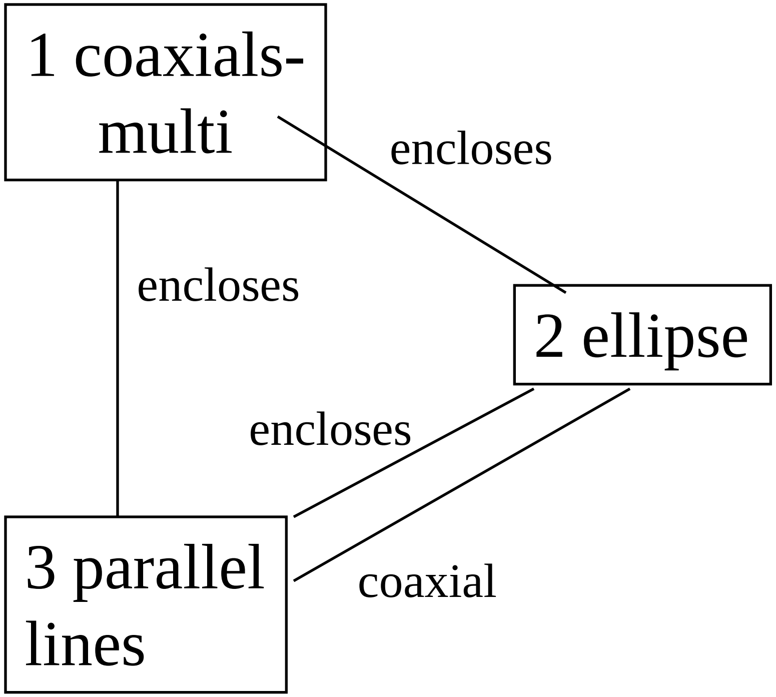
FEATURES:
1: coaxials-multi
2: ellipse
3: parallel lines

How are 1, 2, and 3 related?

What other features and relationships can you find?

Graph and 2-Graph Representations





1 coaxials-multi

encloses

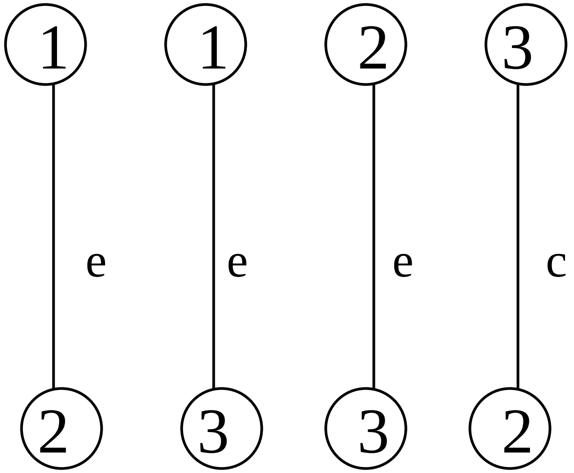
encloses

2 ellipse

encloses

3 parallel lines

coaxial



RDF!

Relational Indexing for Recognition

Preprocessing (off-line) Phase

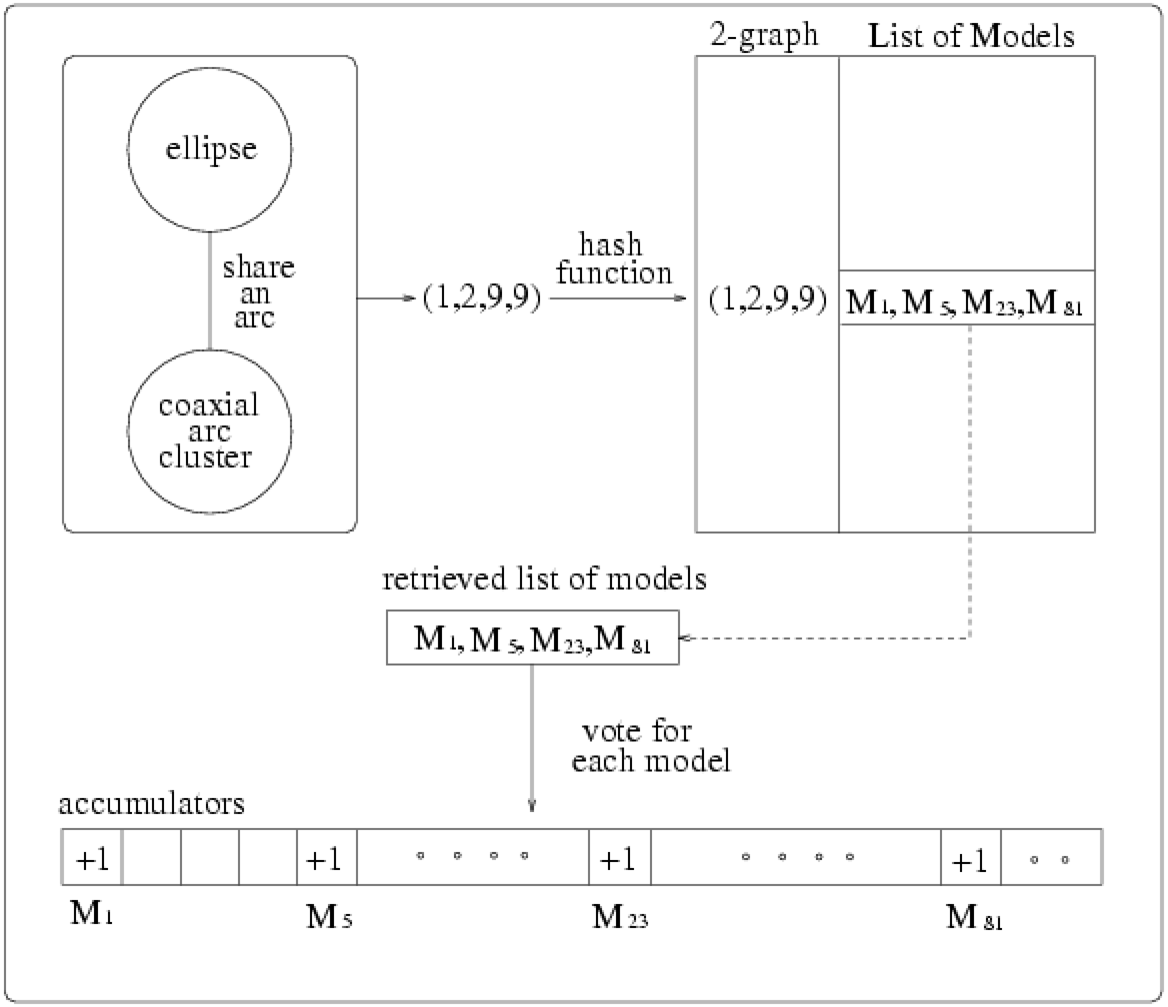
for each model view Mi in the database

- encode each 2-graph of Mi to produce an index
- store Mi and associated information in the indexed bin of a hash table H

Matching (on-line) phase

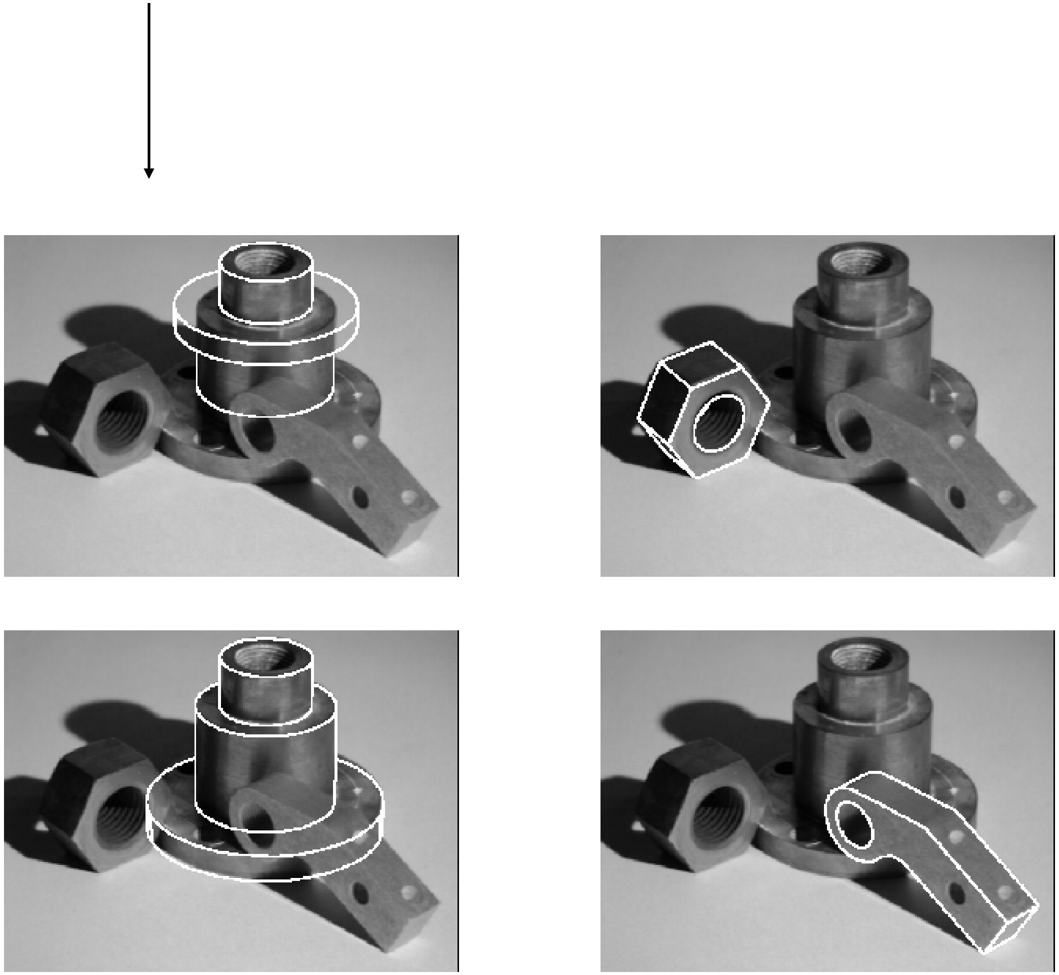
1. Construct a relational (2-graph) description D for the scene
2. For each 2-graph G of D
    - encode it, producing an index to access the hash table H
    - cast a vote for each Mi in the associated bin
3. Select the Mi’s with high votes as possible hypotheses
4. Verify or disprove via alignment, using the 3D meshes

The Voting Process



RIO Verifications

incorrect
hypothesis



1. The matched features of the hypothesized object are used to determine its **pose**.

2. The **3D mesh** of the object is used to project all its features onto the image.

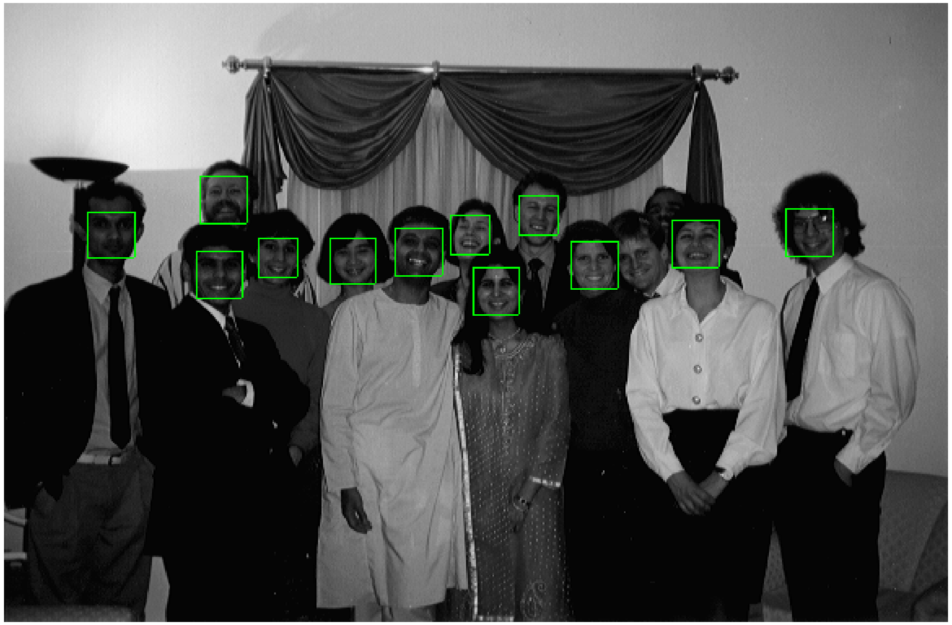
3. A **verification procedure** checks how well the object features line up with edges on the image.

Use of classifiers is big in computer vision today.

- 2 Examples:
  - Rowley’s Face Detection using neural nets
  - Yi’s image classification using EM

Object Detection:
Rowley's Face Finder

1. convert to gray scale
2. normalize for lighting
3. histogram equalization
4. apply neural net(s)
   trained on 16K images



What data is fed to
the classifier?

32 x 32 windows in
a pyramid structure

Object Class Recognition using Images of Abstract Regions

Yi Li, Jeff A. Bilmes, and Linda G. Shapiro

Department of Computer Science and Engineering

Department of Electrical Engineering

University of Washington

Problem Statement

Given: Some images and their corresponding descriptions

{trees, grass, cherry trees}

{cheetah, trunk}

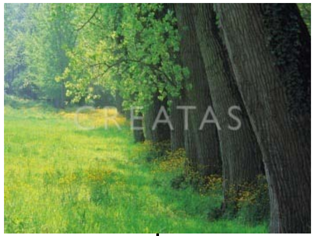
{mountains, sky}

{beach, sky, trees, water}

●●●

To solve: What object classes are present in new images

?


?

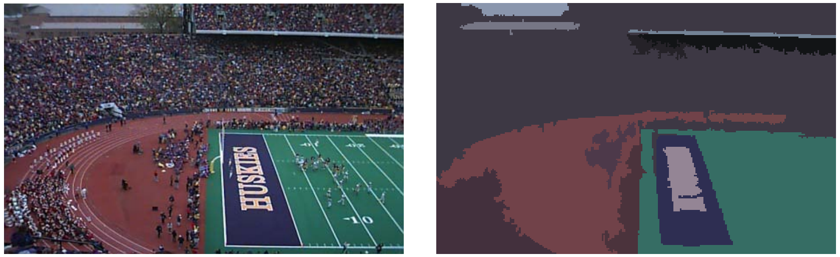
?

?

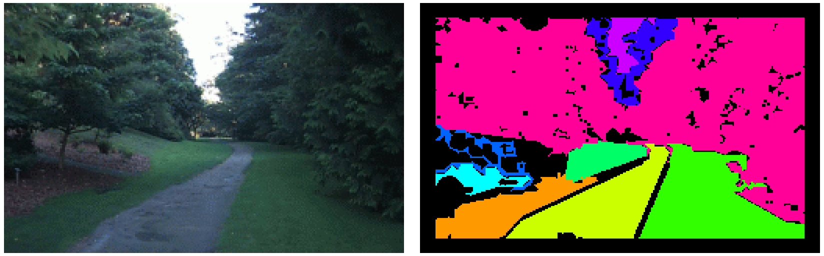
●●●

Image Features for Object Recognition

- Color



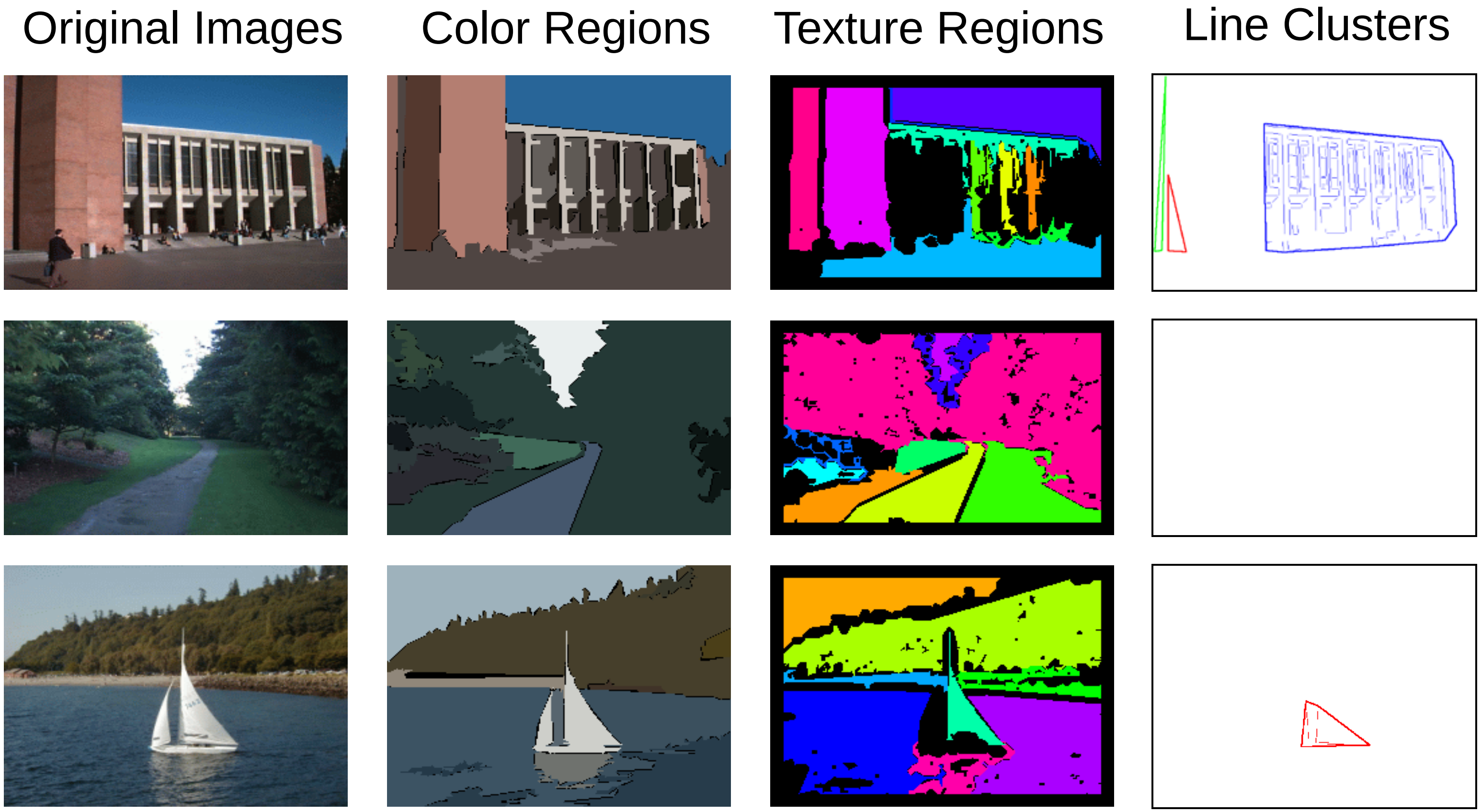
- Texture



- Structure

- Context

Abstract Regions



Original Images

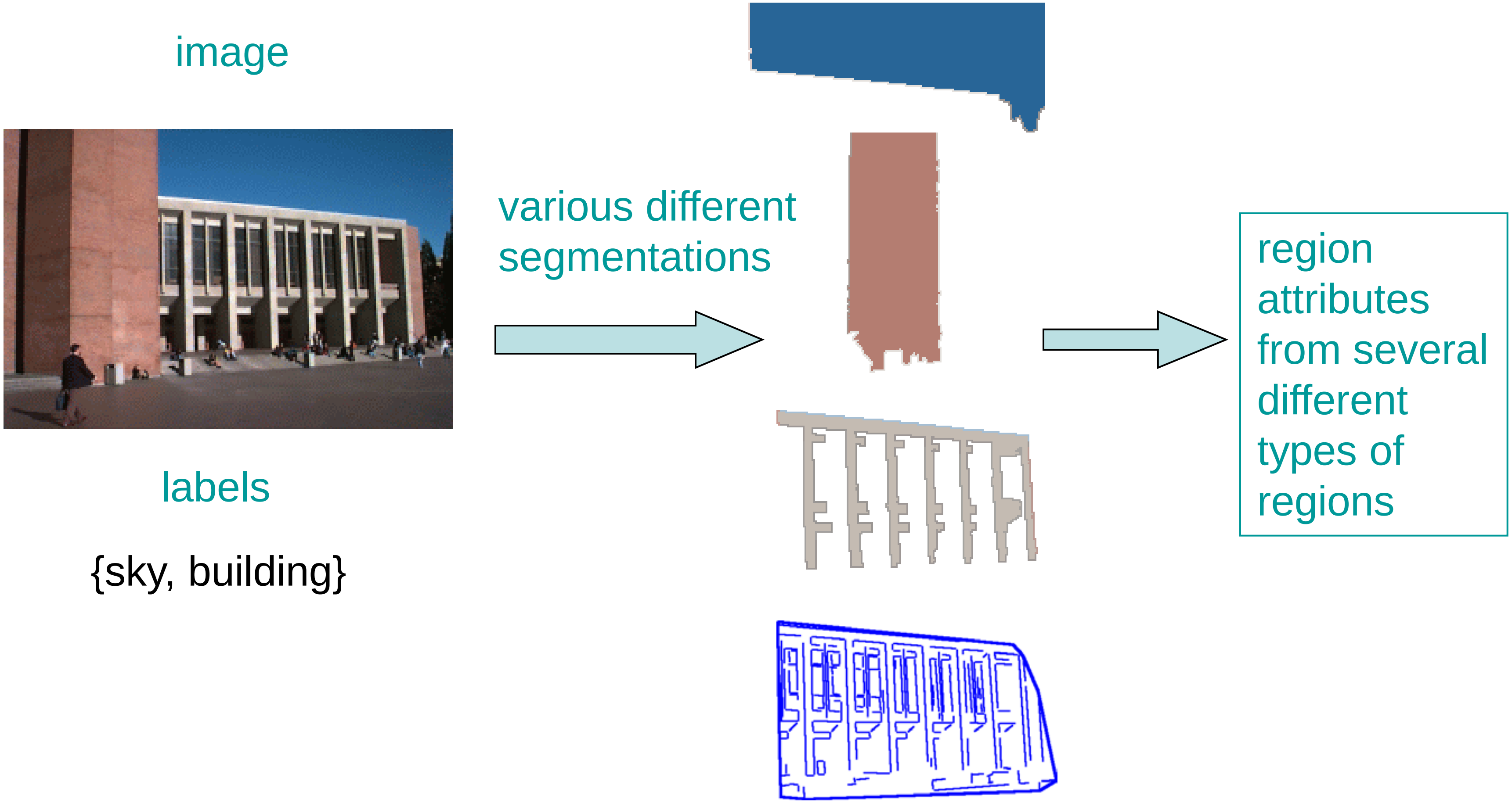
Color Regions

Texture Regions

Line Clusters

Abstract Regions

Multiple segmentations whose regions are not labeled; a list of labels is provided for each training image.



image

labels

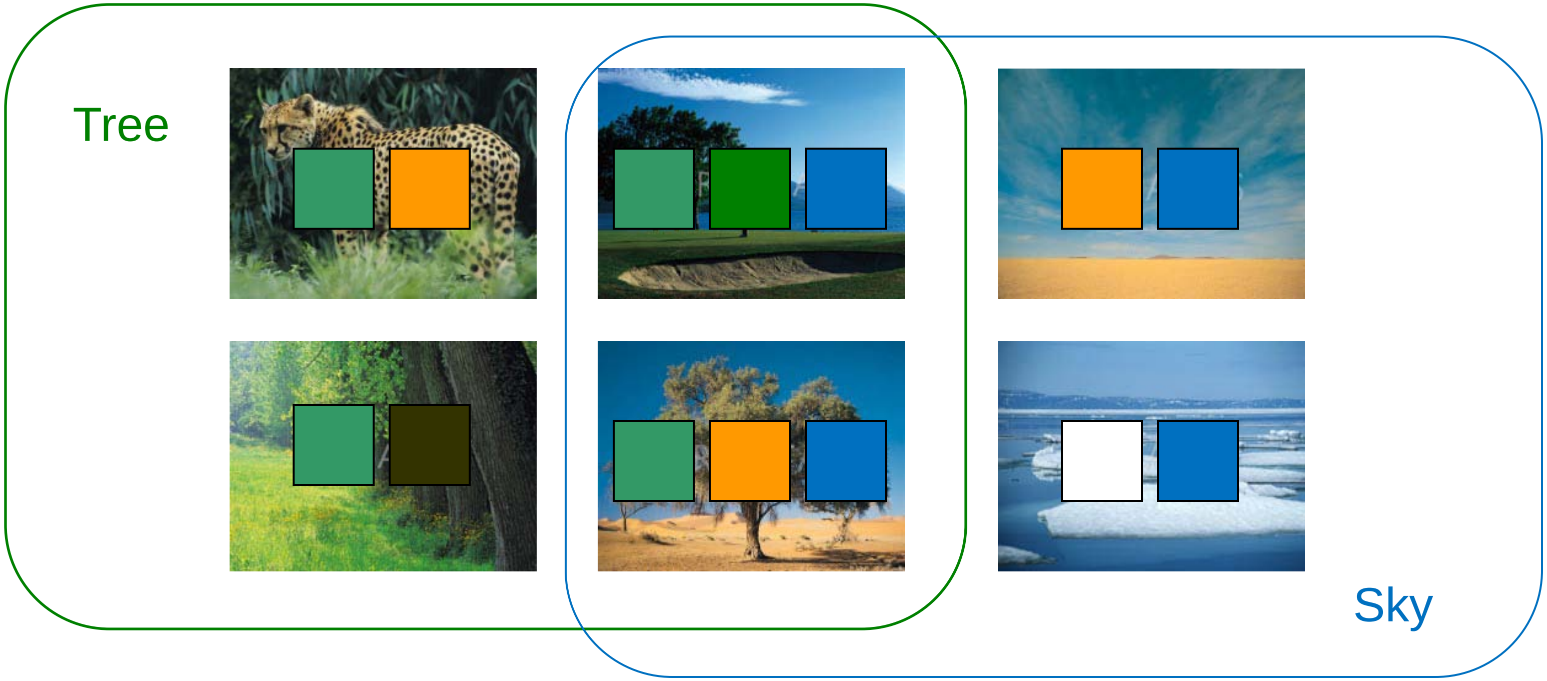
{sky, building}

various different segmentations

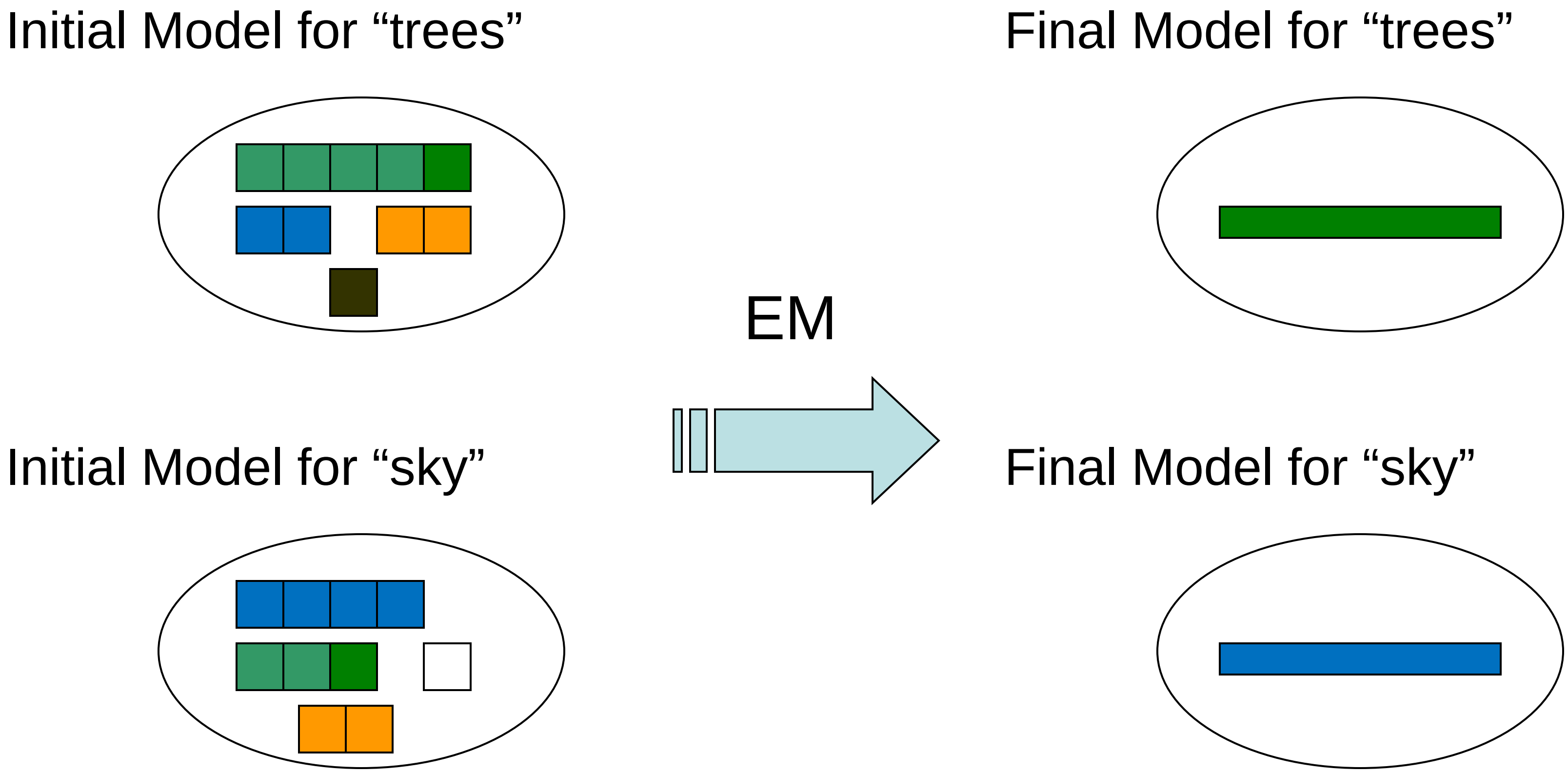
region attributes from several different types of regions

Model Initial Estimation

- Estimate the initial model of an object using all the region features from all images that contain the object



EM Classifier: the Idea



Initial Model for “trees”

Final Model for “trees”

EM

Initial Model for “sky”

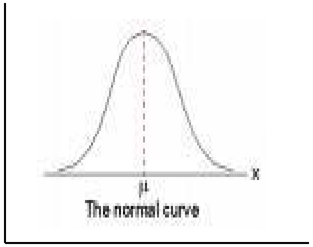
Final Model for “sky”

EM Algorithm

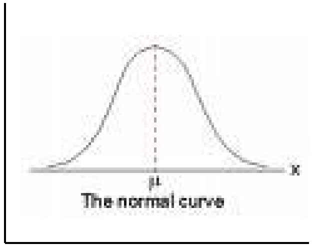
- Start with K clusters, each represented by a probability distribution
- Assuming a Gaussian or Normal distribution, each cluster is represented by its mean and variance (or covariance matrix) and has a weight.
- Go through the training data and soft-assign it to each cluster. Do this by computing the probability that each training vector belongs to each cluster.
- Using the results of the soft assignment, recompute the parameters of each cluster.
- Perform the last 2 steps iteratively.

1-D EM with Gaussian Distributions

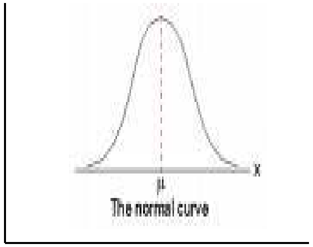
- Each cluster C_j is represented by a Gaussian distribution $N(\mu_j , \sigma_j)$.
- Initialization: For each cluster C_j initialize its mean μ_j , variance σ_j, and weight α_j.



$N(\mu_1 , \sigma_1)$
$\alpha_1 = P(C_1)$



$N(\mu_2 , \sigma_2)$
$\alpha_2 = P(C_2)$



$N(\mu_3 , \sigma_3)$
$\alpha_3 = P(C_3)$

- With no other knowledge, use random means and variances and equal weights.

Expectation

- For each point x_i and each cluster C_j compute $P(C_j \mid x_i)$, the probability of cluster j given point i.

- $P(C_j \mid x_i) = P(x_i \mid C_j)\, P(C_j\,)\, /\, P(x_i)$

- $P(x_i) = \sum_j P(x_i \mid C_j)\, P(C_j)$

- Question: Where do we get $P(x_i \mid C_j)$ and $P(C_j)$?

Answer:

1. Use the pdf for a normal distribution:

$$P(x_i \mid C_j) = \frac{1}{\sqrt{2\pi}\ \sigma_j}\ e^{-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}}$$

2. Use $\alpha_j = P(C_j)$ from the current parameters of cluster C_j.

Maximization

- Having computed $P(C_j \mid x_i)$ for each point x_i and each cluster C_j, use them to compute new mean, variance, and weight for each cluster.

$$\mu_j = \frac{\sum_i p(C_j \mid x_i) \cdot x_i}{\sum_i p(C_j \mid x_i)}$$

$$\sigma_j = \frac{\sum_i p(C_j \mid x_i) \cdot (x_i - \mu_j) \cdot (x_i - \mu_j)^T}{\sum_i p(C_j \mid x_i)}$$

$$\alpha_j = p(C_j) = \frac{\sum_i p(C_j \mid x_i)}{N}$$

Standard EM to EM Classifier

- That’s the standard EM algorithm.
- For n-dimensional data, the variance becomes a co-variance matrix, which changes the formulas slightly.
- But we used an EM variant to produce a classifier.
- The next slide indicates the differences between what we used and the standard.

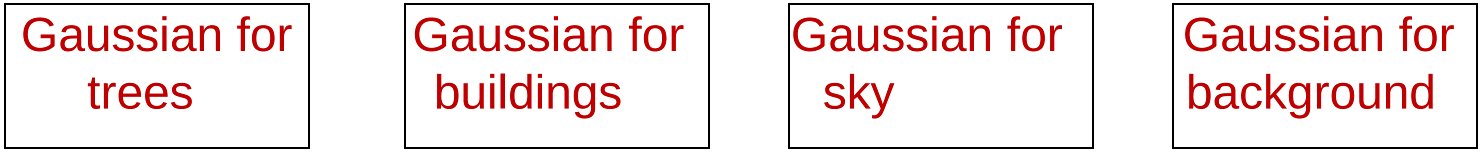
EM Classifier

1. Fixed Gaussian components (one Gaussian per object class) and fixed weights corresponding to the frequencies of the corresponding objects in the training data.

2. Customized initialization uses only the training images that contain a particular object class to initialize its Gaussian.

3. Controlled expectation step ensures that a feature vector only contributes to the Gaussian components representing objects present in its training image.

4. Extra background component absorbs noise.



Gaussian for trees

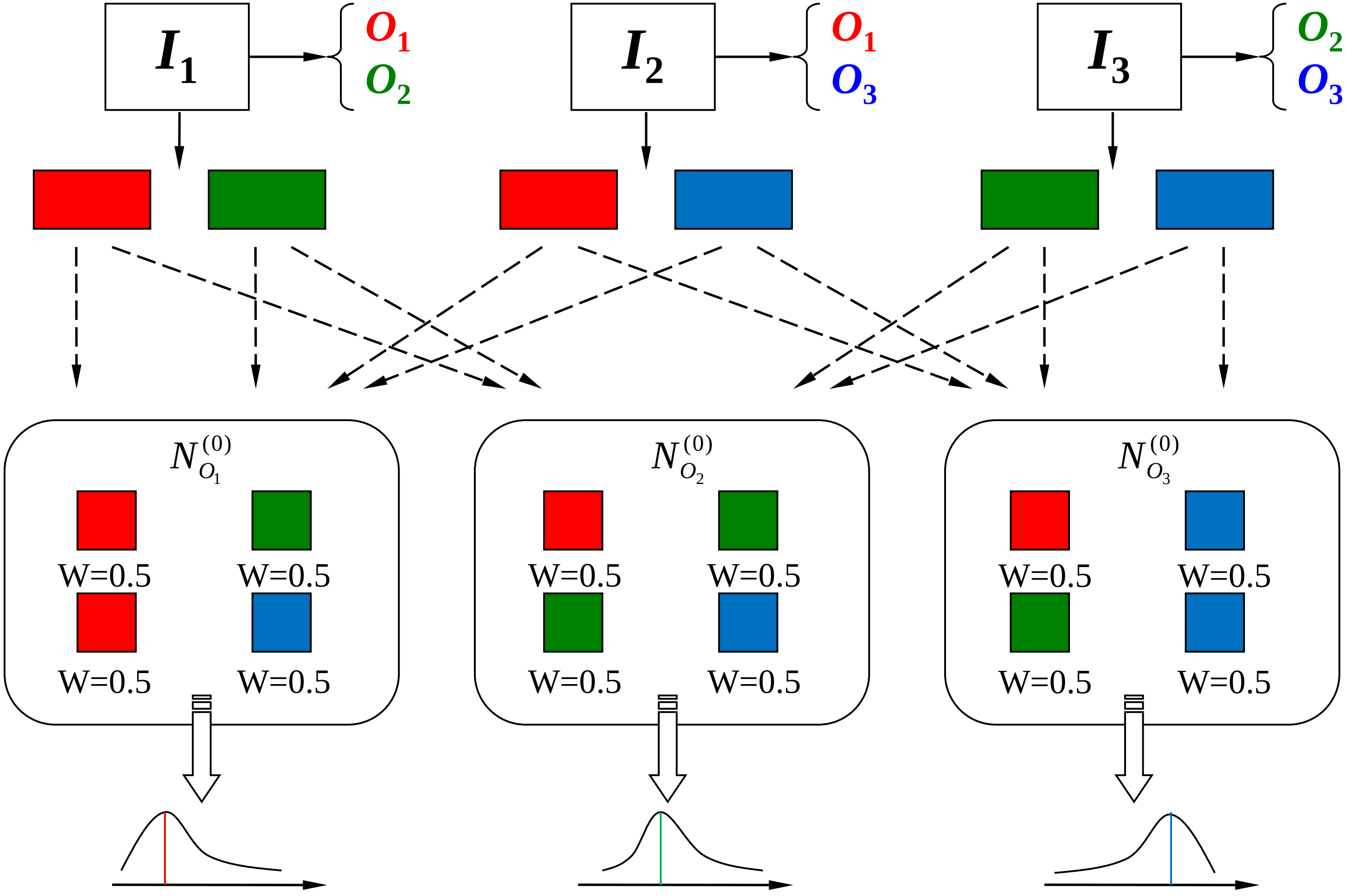
Gaussian for buildings

Gaussian for sky

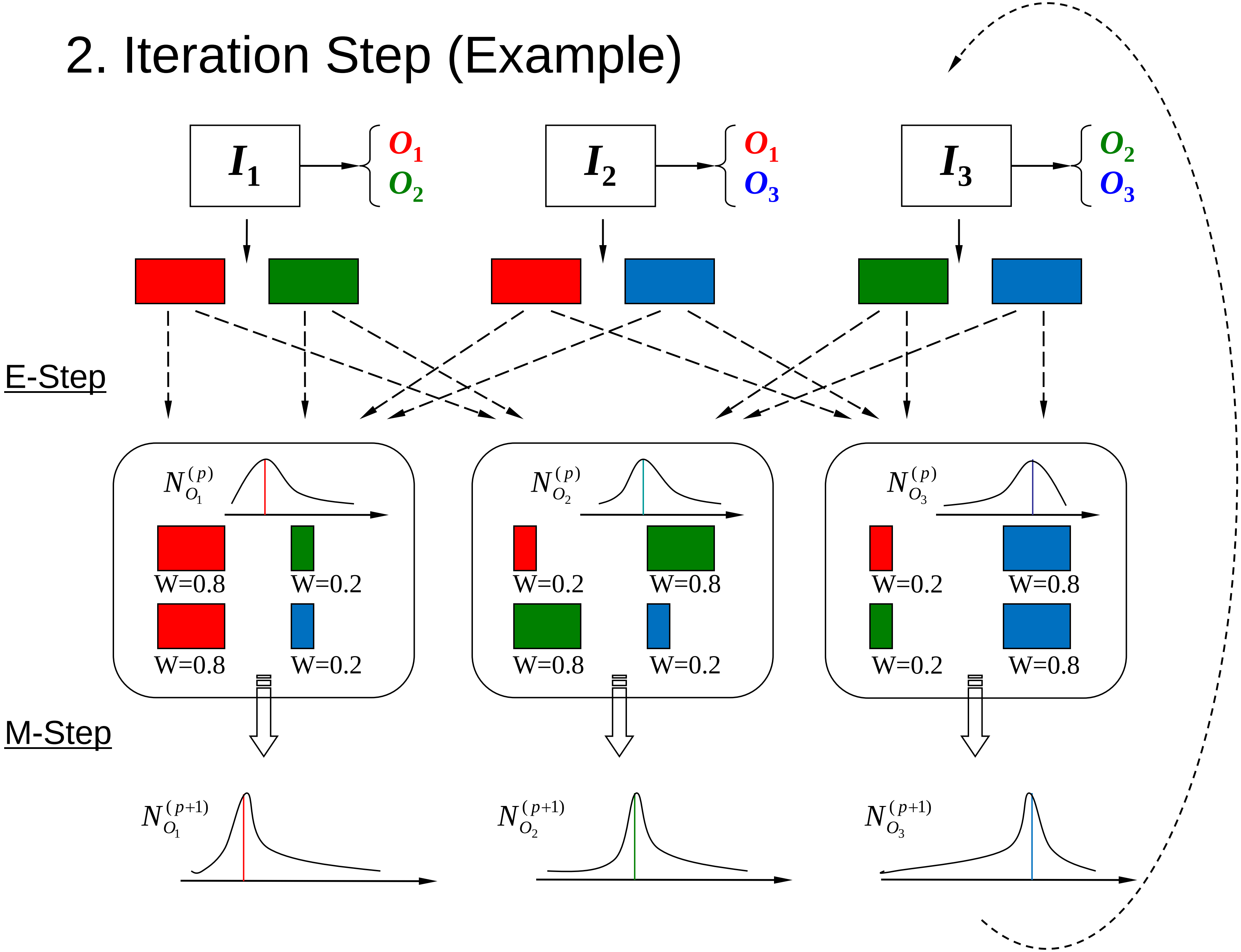
Gaussian for background

1. Initialization Step (Example)

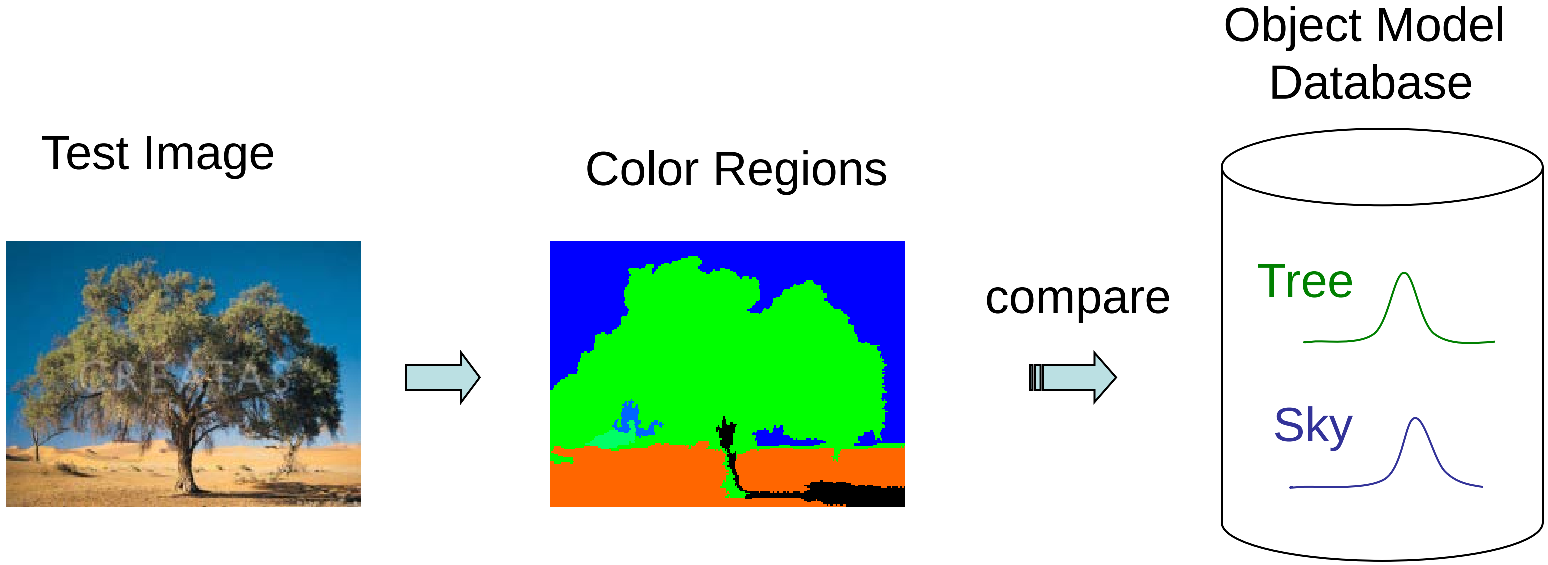
Image & description



2. Iteration Step (Example)



Recognition



How do you decide if a particular object is in an image?

To calculate $p(tree / image)$

$p(tree / image) = f$

$p($ tree/ $)$

$p($ tree/ $)$

$p($ tree/ $)$

$p($ tree/ $)$

$$p(o \mid F_I^a) = \underset{r^a \in F_I^a}{f} (p(o \mid r^a))$$

f is a function that combines probabilities from all the color regions in the image.

e.g. max or mean

Combining different types of abstract regions: First Try

- Treat the different types of regions independently and combine at the time of classification.

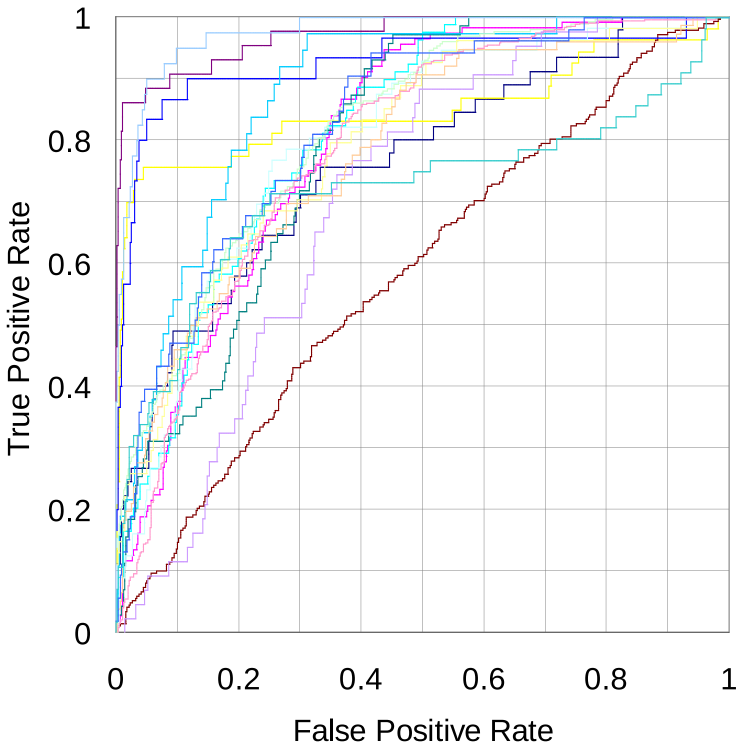
$$p(o \mid \{F_I^a\}) = \prod_a p(o \mid F_I^a)$$

- Form intersections of the different types of regions, creating smaller regions that have both color and texture properties for classification.

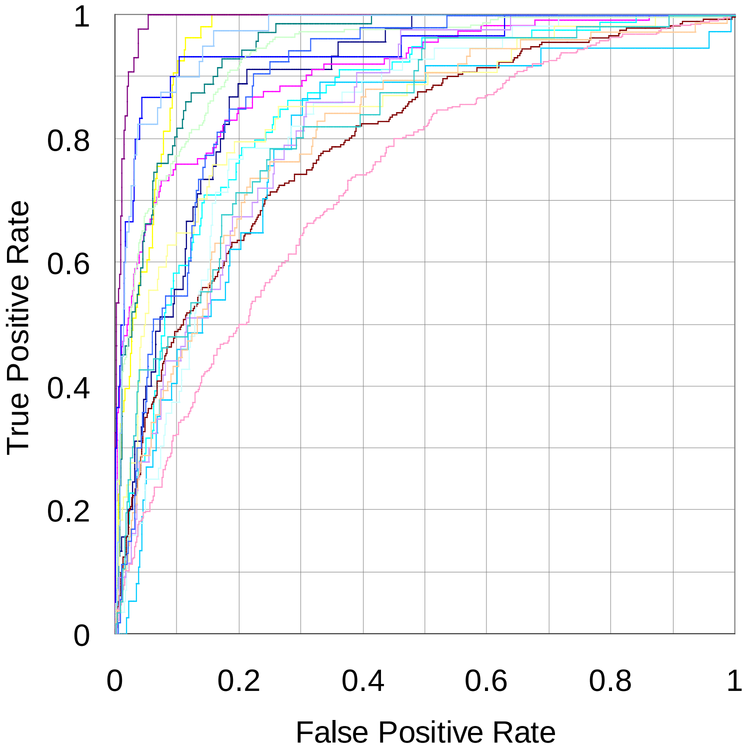
Experiments (on 860 images)

- 18 keywords: mountains (30), orangutan (37), track (40), tree trunk (43), football field (43), beach (45), prairie grass (53), cherry tree (53), snow (54), zebra (56), polar bear (56), lion (71), water (76), chimpanzee (79), cheetah (112), sky (259), grass (272), tree (361).
- A set of cross-validation experiments (80% as training set and the other 20% as test set)
- The poorest results are on object classes “tree,” “grass,” and “water,” each of which has a high variance; a single Gaussian model is insufficient.

ROC Charts:
True Positive vs. False Positive



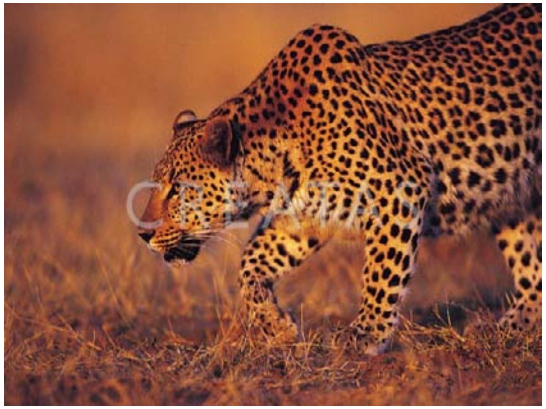
Independent Treatment of
Color and Texture

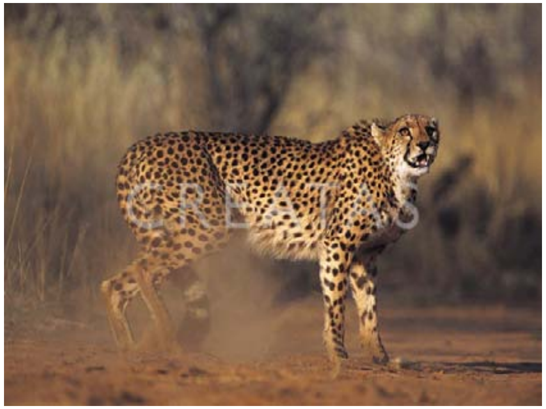


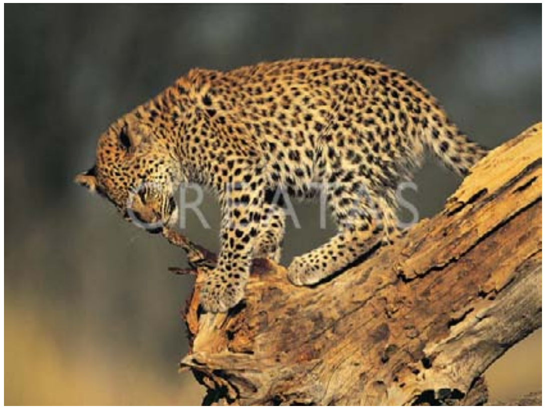
Using Intersections of
Color and Texture Regions

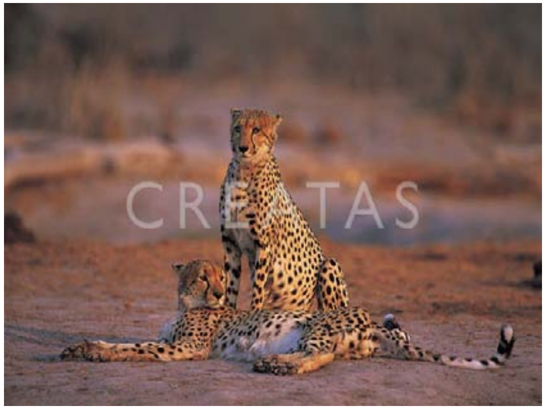
Sample Retrieval Results

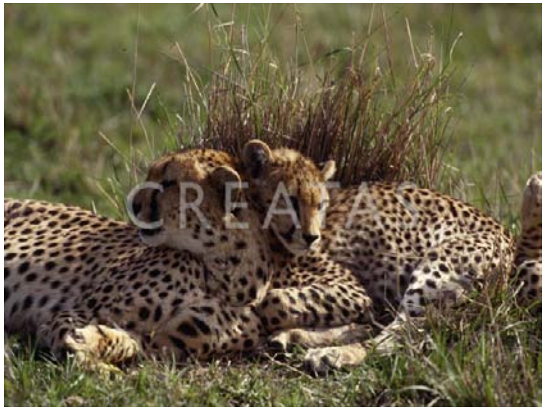
cheetah









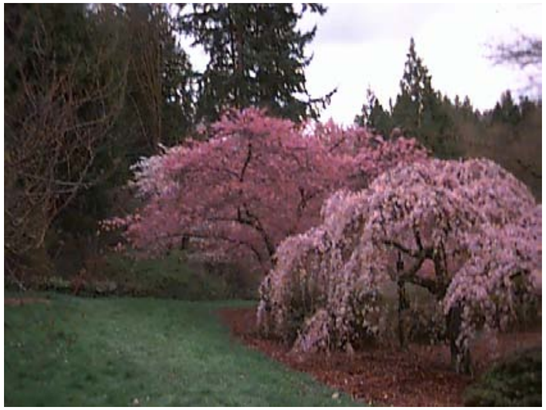


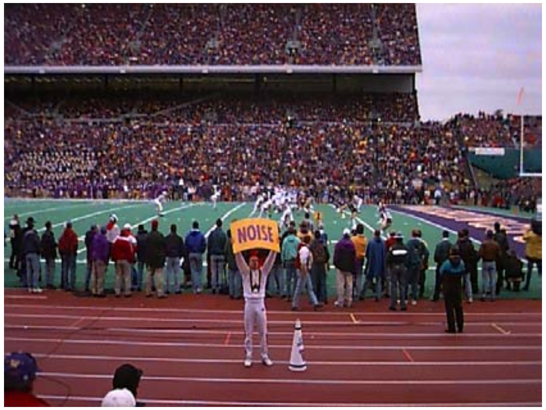
Sample Results (Cont.)

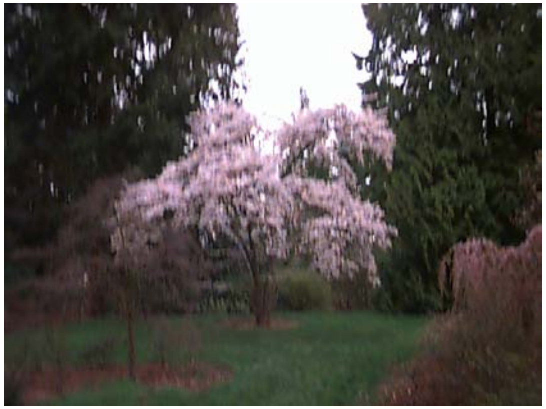
grass

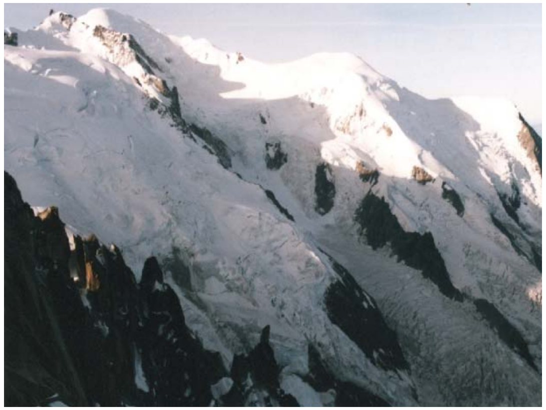
Sample Results (Cont.)

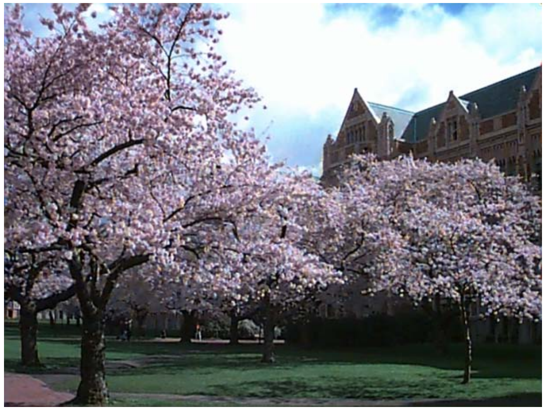
cherry tree





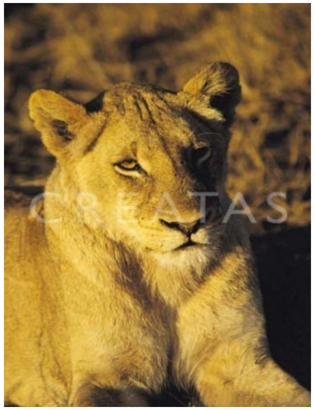


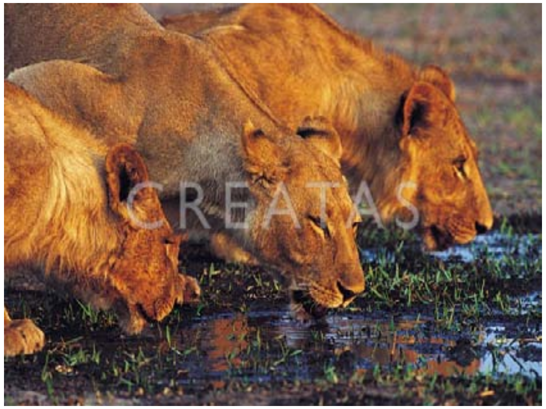


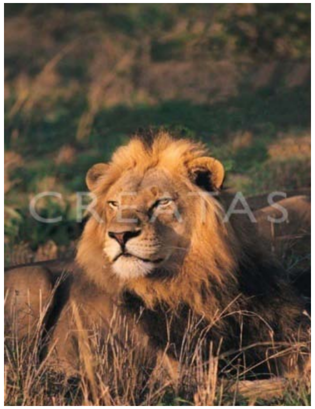


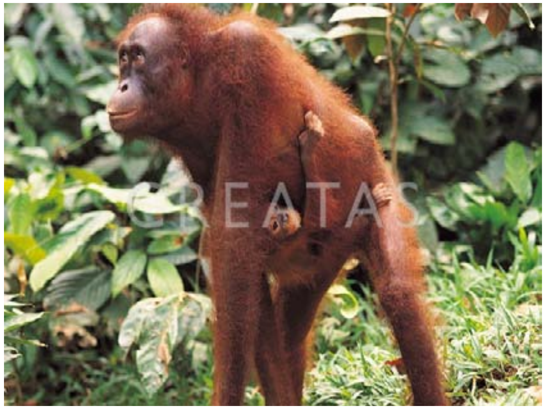
Sample Results (Cont.)

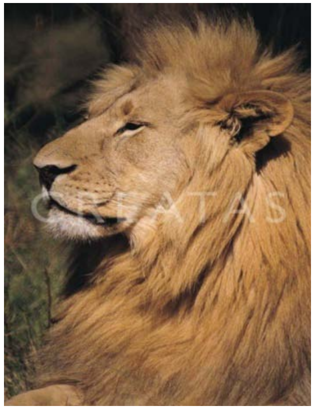
lion











Summary

- Designed a set of abstract region features: color, texture, structure, . . .
- Developed a new semi-supervised EM-like algorithm to recognize object classes in color photographic images of outdoor scenes; tested on 860 images.
- Compared two different methods of combining different types of abstract regions. The intersection method had a higher performance

A Better Approach to Combining Different Feature Types

Phase 1:

- Treat each type of abstract region separately
- For abstract region type ***a*** and for object class ***o***, use the EM algorithm to construct clusters that are multivariate Gaussians over the features for type ***a*** regions.

Consider only abstract region type color (c) and object class object (o)

- At the end of Phase 1, we can compute the distribution of color feature vectors in an image containing object *o*.

$$P(X^c|o) = \sum_{m=1}^{M^c} w_m^c \cdot N(X^c; \mu_m^c, \Sigma_m^c)$$

- M^c is the number of components (clusters).
- The *w's* are the weights (α's) of the components.
- The *μ's* and *$\sum$'s* are the parameters of the components.
- $N(X^c, \mu^c{}_m, \Sigma^c{}_m)$ specifies the probabilty that X^c belongs to a particular normal distribution.

Now we can determine which components are likely to be present in an image.

- The probability that the feature vector X from color region *r* of image I_i comes from component *m* is given by

$$P(X^c_{i,r}, m^c) = w^c_m \cdot N(X^c_{i,r}, \mu^c_m, \Sigma^c_m)$$

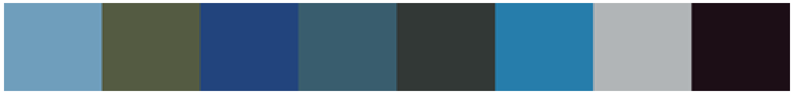
- Then the probability that image I_i has a region that comes from component *m* is

$$P(I_i, m^c) = f(\{P(X^c_{i,r}, m^c) | r = 1, 2, \ldots\})$$

- where f is an aggregate function such as mean or max

Aggregate Scores for Color

Components



		1	2	3	4	5	6	7	8
beach		.93	.16	.94	.24	.10	.99	.32	.00
beach		.66	.80	.00	.72	.19	.01	.22	.02
not beach		.43	.03	.00	.00	.00	.00	.15	.00

We now use positive and negative training images, calculate for each the probabilities of regions of each component, and form a training matrix.

$$\begin{array}{c} I_1^+ \\ I_2^+ \\ \vdots \\ I_1^- \\ I_2^- \\ \vdots \end{array} \begin{bmatrix} P(I_1^+, 1^c) & P(I_1^+, 2^c) & \cdots & P(I_1^+, M^c) \\ P(I_2^+, 1^c) & P(I_2^+, 2^c) & \cdots & P(I_2^+, M^c) \\ \vdots & & & \\ P(I_1^-, 1^c) & P(I_1^-, 2^c) & \cdots & P(I_1^-, M^c) \\ P(I_2^-, 1^c) & P(I_2^-, 2^c) & \cdots & P(I_2^-, M^c) \\ \vdots & & & \end{bmatrix}$$

Phase 2 Learning

- Let C_i be row i of the training matrix.
- Each such row is a feature vector for the color features of regions of image I_i that relates them to the Phase 1 components.
- Now we can use a second-stage classifier to learn $P(o|I_i)$ for each object class o and image I_i .

Multiple Feature Case

- We calculate separate Gaussian mixture models for each different features type:

- Color: C_i
- Texture: T_i
- Structure: S_i

- and any more features we have (motion).

Now we concatenate the matrix rows from the different region types to obtain a multi-feature-type training matrix and train a neural net classifier to classify images.

$$\begin{array}{ccccc} \textit{color} & \textit{texture} & \textit{structure} & & \textit{everything} \\ \begin{pmatrix} C_1^+ \\ C_2^+ \\ . \\ . \\ C_1^- \\ C_2^- \\ . \\ . \end{pmatrix} & \begin{pmatrix} T_1^+ \\ T_2^+ \\ . \\ . \\ T_1^- \\ T_2^- \\ . \\ . \end{pmatrix} & \begin{pmatrix} S_1^+ \\ S_2^+ \\ . \\ . \\ S_1^- \\ S_2^- \\ . \\ . \end{pmatrix} & \longrightarrow & \begin{pmatrix} C_1^+ & T_1^+ & S_1^+ \\ C_2^+ & T_2^+ & S_2^+ \\ . & . & . \\ . & . & . \\ C_1^- & T_1^- & S_1^- \\ C_2^- & T_2^- & S_2^- \\ . & . & . \\ . & . & . \end{pmatrix} \end{array}$$

ICPR04 Data Set with General Labels

	EM-variant with single Gaussian per object	EM-variant extension to mixture models	Gen/Dis with Classical EM clustering	Gen/Dis with EM-variant extension
African animal	71.8%	85.7%	89.2%	90.5%
arctic	80.0%	79.8%	90.0%	85.1%
beach	88.0%	90.8%	89.6%	91.1%
grass	76.9%	69.6%	75.4%	77.8%
mountain	94.0%	96.6%	97.5%	93.5%
primate	74.7%	86.9%	91.1%	90.9%
sky	91.9%	84.9%	93.0%	93.1%
stadium	95.2%	98.9%	99.9%	100.0%
tree	70.7%	79.0%	87.4%	88.2%
water	82.9%	82.3%	83.1%	82.4%
MEAN	**82.6%**	**85.4%**	**89.6%**	**89.3%**

Comparison to ALIP: the Benchmark Image Set

- Test database used in SIMPLIcity paper and ALIP paper.

- 10 classes (*African people*, *beach*, *buildings*, *buses*, *dinosaurs*, *elephants*, *flowers*, *food*, *horses*, *mountains*). 100 images each.

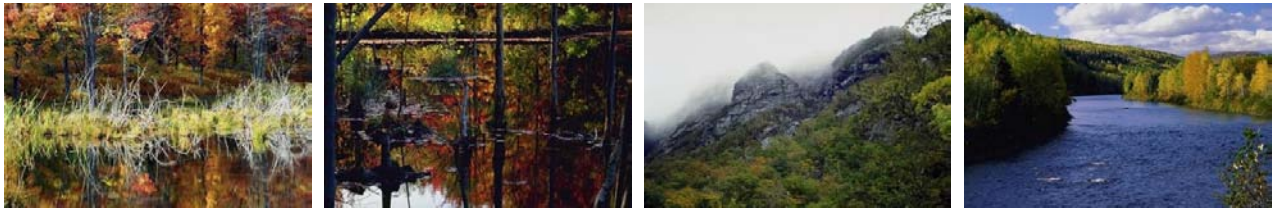
Comparison to ALIP: the Benchmark Image Set

	ALIP	cs	ts	st	ts+st	cs+st	cs+ts	cs+ts+st
African	52	69	23	26	35	79	72	74
beach	32	44	38	39	51	48	59	64
buildings	64	43	40	41	67	70	70	78
buses	46	60	72	92	86	85	84	95
dinosaurs	100	88	70	37	86	89	94	93
elephants	40	53	8	27	38	64	64	69
flowers	90	85	52	33	78	87	86	91
food	68	63	49	41	66	77	84	85
horses	60	94	41	50	64	92	93	89
mountains	84	43	33	26	43	63	55	65
MEAN	**63.6**	**64.2**	**42.6**	**41.2**	**61.4**	**75.4**	**76.1**	**80.3**

Comparison to ALIP: the 60K Image Set

0. Africa, people, landscape, animal

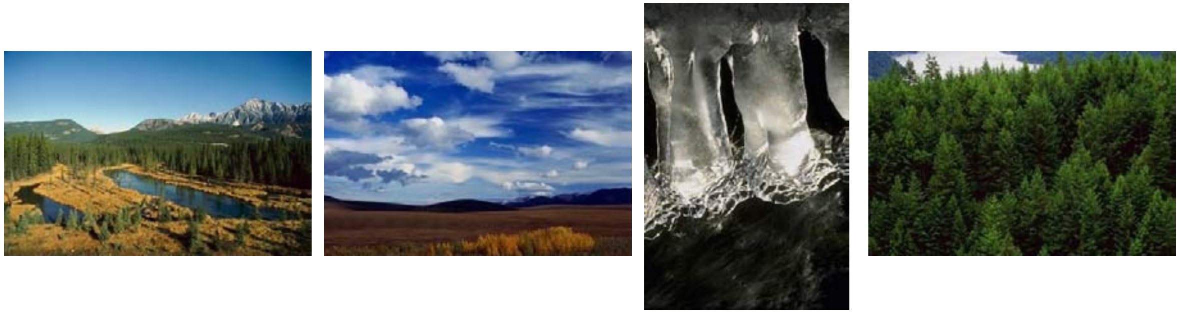
1. autumn, tree, landscape, lake



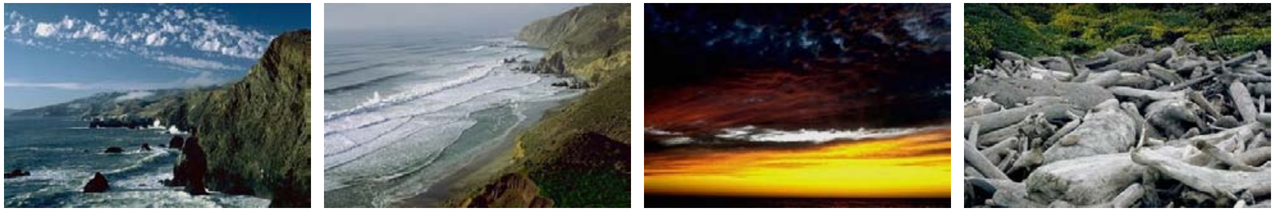
2. Bhutan, Asia, people, landscape, church

Comparison to ALIP: the 60K Image Set

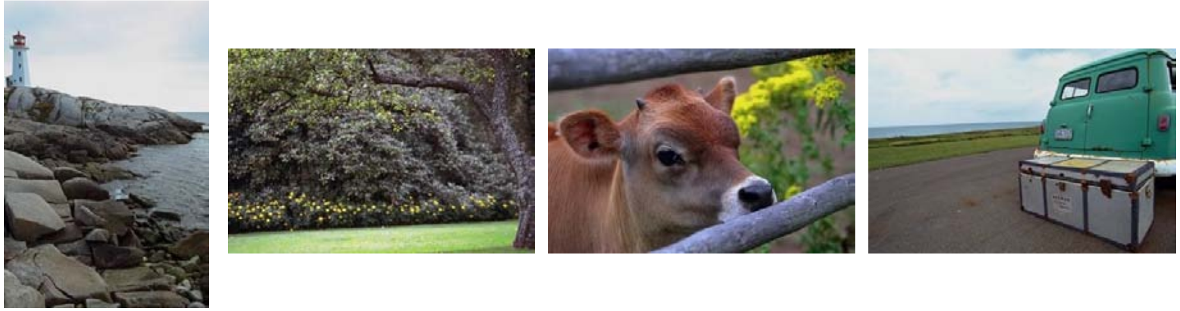
3. California, sea, beach, ocean, flower



4. Canada, sea, boat, house, flower, ocean



5. Canada, west, mountain, landscape, cloud, snow, lake



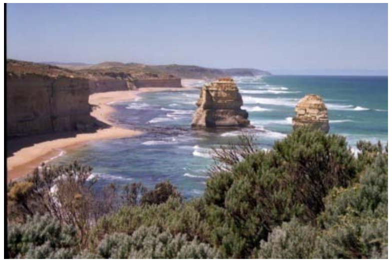
Comparison to ALIP: the 60K Image Set

Number of top-ranked categories required	1	2	3	4	5
ALIP	11.88	17.06	20.76	23.24	26.05
Gen/Dis	11.56	17.65	21.99	25.06	27.75

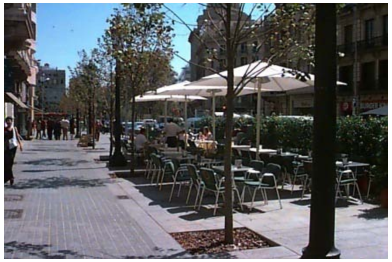
The table shows the percentage of test images whose true categories were included in the top-ranked categories.

Groundtruth Data Set

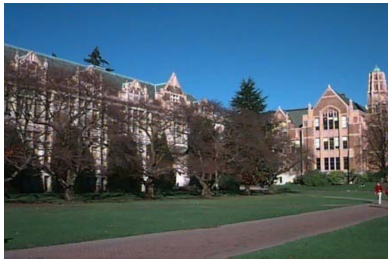
- UW Ground truth database (1224 images)
- 31 elementary object categories: *river* (30), *beach* (31), *bridge* (33), *track* (35), *pole* (38), *football field* (41), *frozen lake* (42), *lantern* (42), *husky stadium* (44), *hill* (49), *cherry tree* (54), *car* (60), *boat* (67), *stone* (70), *ground* (81), *flower* (85), *lake* (86), *sidewalk* (88), *street* (96), *snow* (98), *cloud* (119), *rock* (122), *house* (175), *bush* (178), *mountain* (231), *water* (290), *building* (316), *grass* (322), *people* (344), *tree* (589), *sky* (659)
- 20 high-level concepts: *Asian city , Australia, Barcelona, campus, Cannon Beach, Columbia Gorge, European city, Geneva, Green Lake, Greenland, Indonesia, indoor, Iran, Italy, Japan, park, San Juans, spring flowers, Swiss mountains, and Yellowstone*.



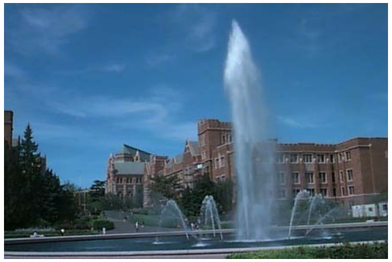
beach, sky, tree, water



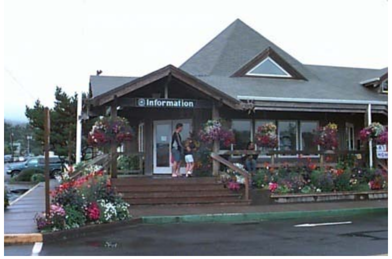
people, street, tree



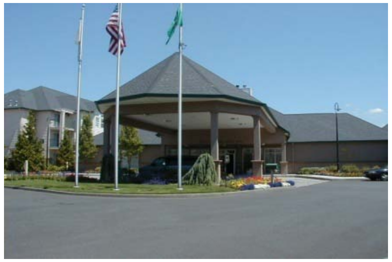
building, grass, people,
sidewalk, sky, tree



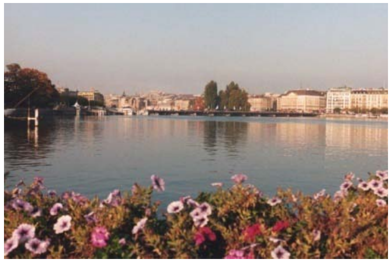
building, bush, sky,
tree, water



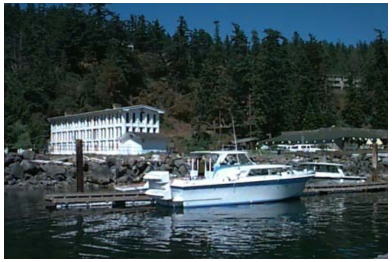
flower, house, people,
pole, sidewalk, sky



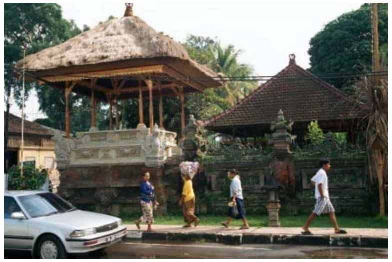
flower, grass, house,
pole, sky, street, tree



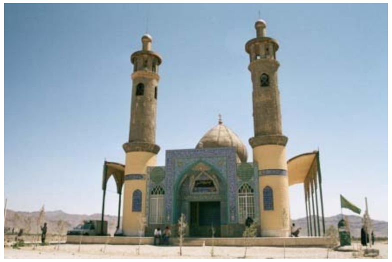
building, flower, sky,
tree, water



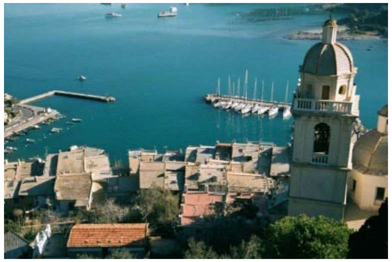
boat, rock, sky,
tree, water



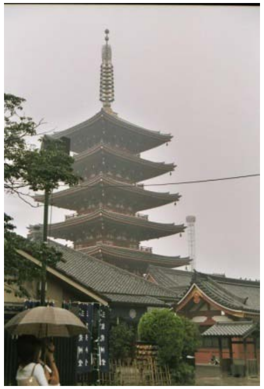
building, car, people, tree



car, people, sky



boat, house, water



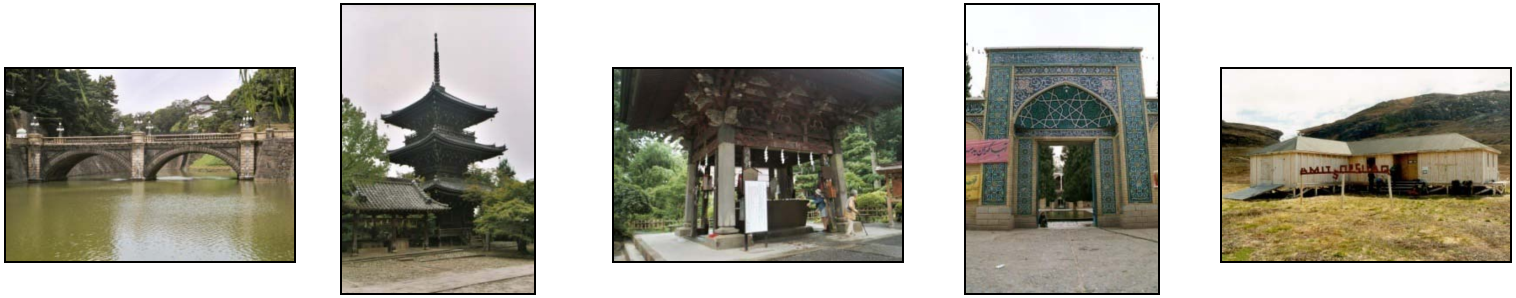
building

Groundtruth Data Set: ROC Scores

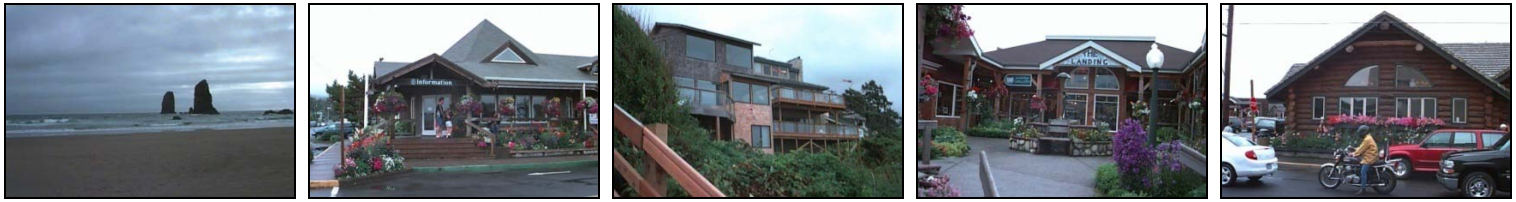
street	60.4	*tree*	80.8	*stone*	87.1	*columbia gorge*	94.5
people	68.0	*bush*	81.0	*hill*	87.4	*green lake*	94.9
rock	73.5	*flower*	81.1	*mountain*	88.3	*italy*	95.1
sky	74.1	*iran*	82.2	*beach*	89.0	*swiss moutains*	95.7
ground	74.3	*bridge*	82.7	*snow*	92.0	*sanjuans*	96.5
river	74.7	*car*	82.9	*lake*	92.8	*cherry tree*	96.9
grass	74.9	*pole*	83.3	*frozen lake*	92.8	*indoor*	97.0
building	75.4	*yellowstone*	83.7	*japan*	92.9	*greenland*	98.7
cloud	75.4	*water*	83.9	*campus*	92.9	*cannon beach*	99.2
boat	76.8	*indonesia*	84.3	*barcelona*	92.9	*track*	99.6
lantern	78.1	*sidewalk*	85.7	*geneva*	93.3	*football field*	99.8
australia	79.7	*asian city*	86.7	*park*	94.0	*husky stadium*	100.0
house	80.1	*european city*	87.0	*spring flowers*	94.4		

Groundtruth Data Set: Top Results

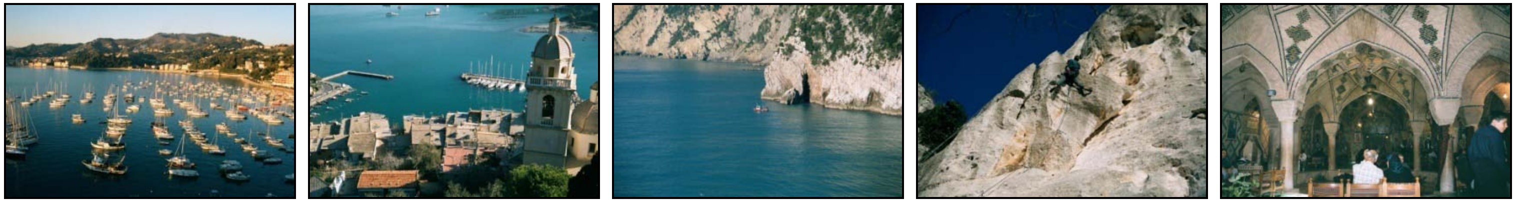
Asian city



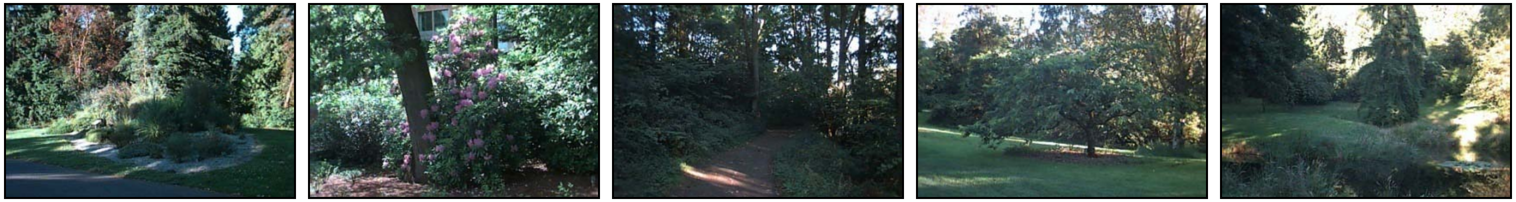
Cannon beach



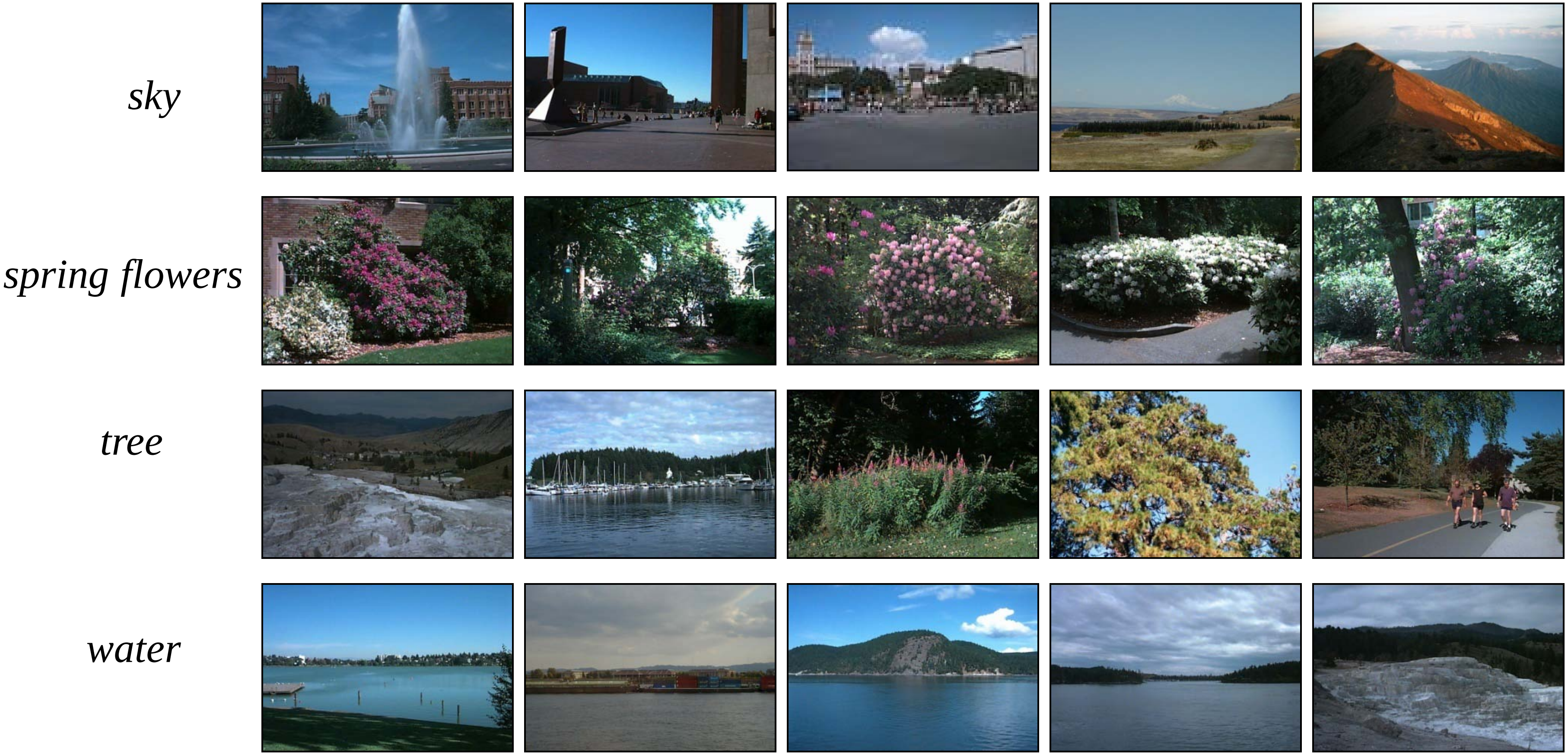
Italy



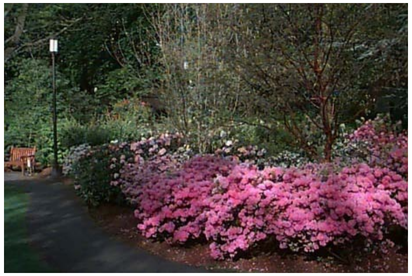
park



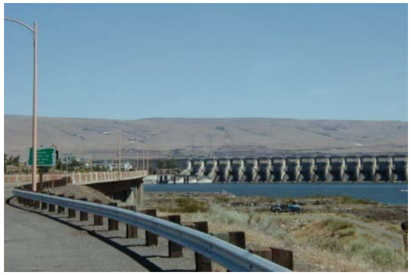
Groundtruth Data Set: Top Results



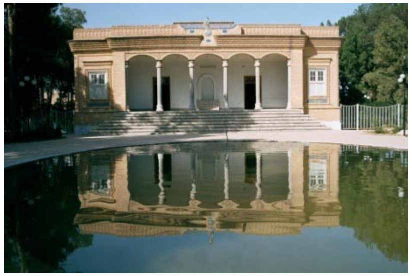
Groundtruth Data Set: Annotation Samples



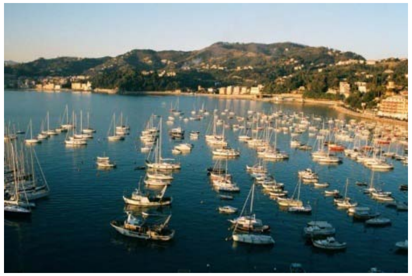
tree(97.3), **bush**(91.6), **spring flowers**(90.3), **flower**(84.4), park(84.3), **sidewalk**(67.5), **grass**(52.5), **pole**(34.1)



sky(99.8), **Columbia gorge**(98.8), lantern(94.2), **street**(89.2), house(85.8), bridge(80.8), car(80.5), hill(78.3), boat(73.1), pole(72.3), **water**(64.3), mountain(63.8), **building**(9.5)



sky(95.1), **Iran**(89.3), house(88.6), **building**(80.1), boat(71.7), bridge(67.0), **water**(13.5), **tree**(7.7)



Italy(99.9), grass(98.5), **sky**(93.8), rock(88.8), **boat**(80.1), **water**(77.1), Iran(64.2), stone(63.9), bridge(59.6), **European**(56.3), sidewalk(51.1), **house**(5.3)