# 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



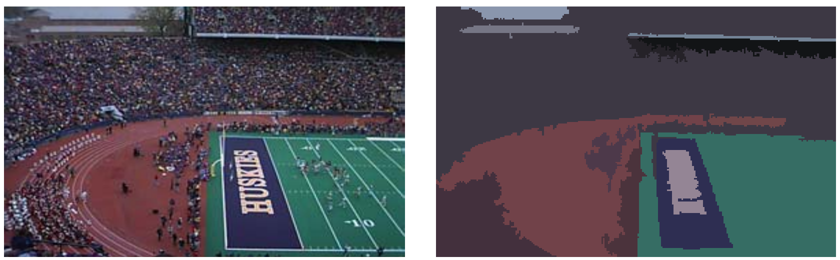


...

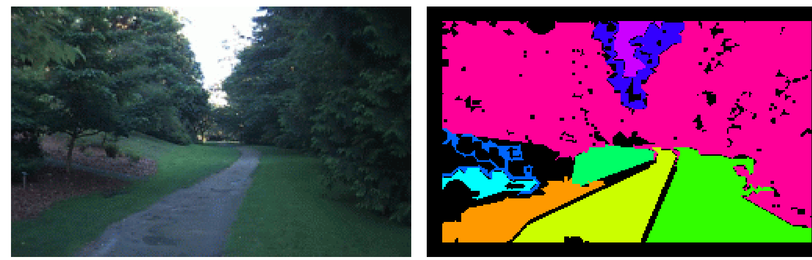
? ? ? ?

# Image Features for Object Recognition

- Color

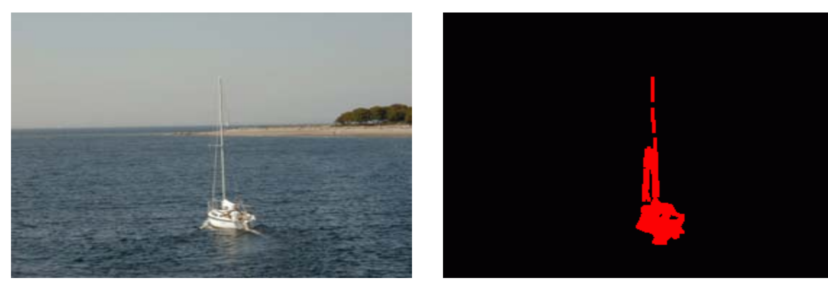



- Texture



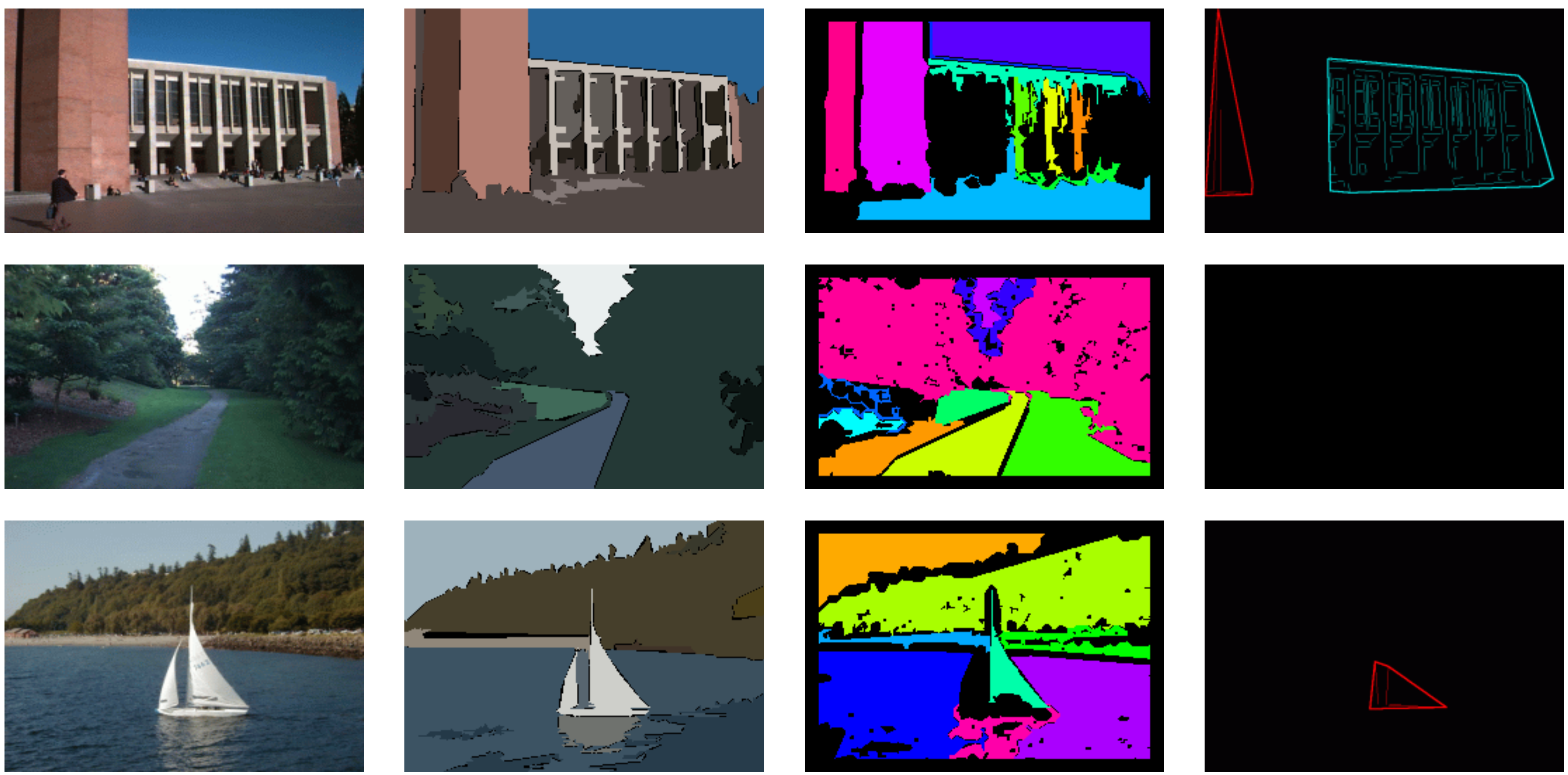
- Structure

- Context

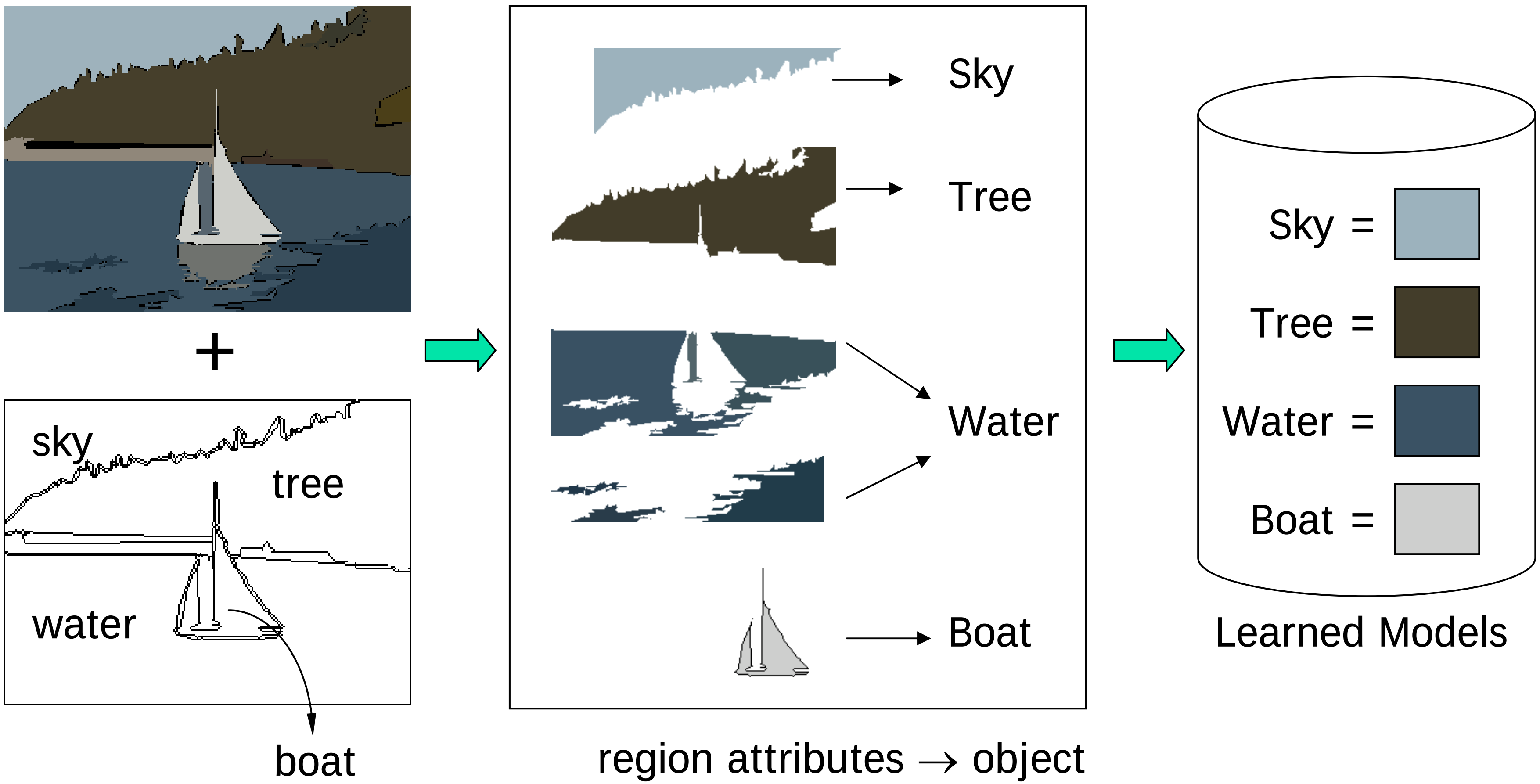


# Abstract Regions

Original Images | Color Regions | Texture Regions | Line Clusters



# Object Model Learning (Ideal)



# Our Scenario: Abstract Regions

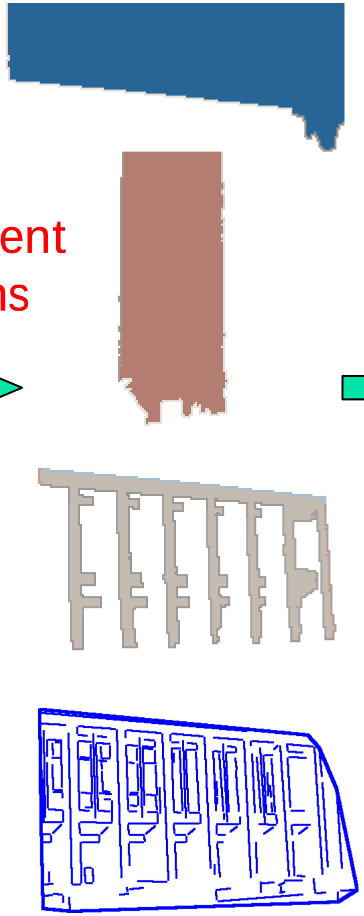
Multiple segmentations whose regions are not labeled; a list of labels is provided for each training image.

image

labels

{sky, building}

various different segmentations



region attributes from several different types of regions

# Object Model Learning

<u>Assumptions</u>

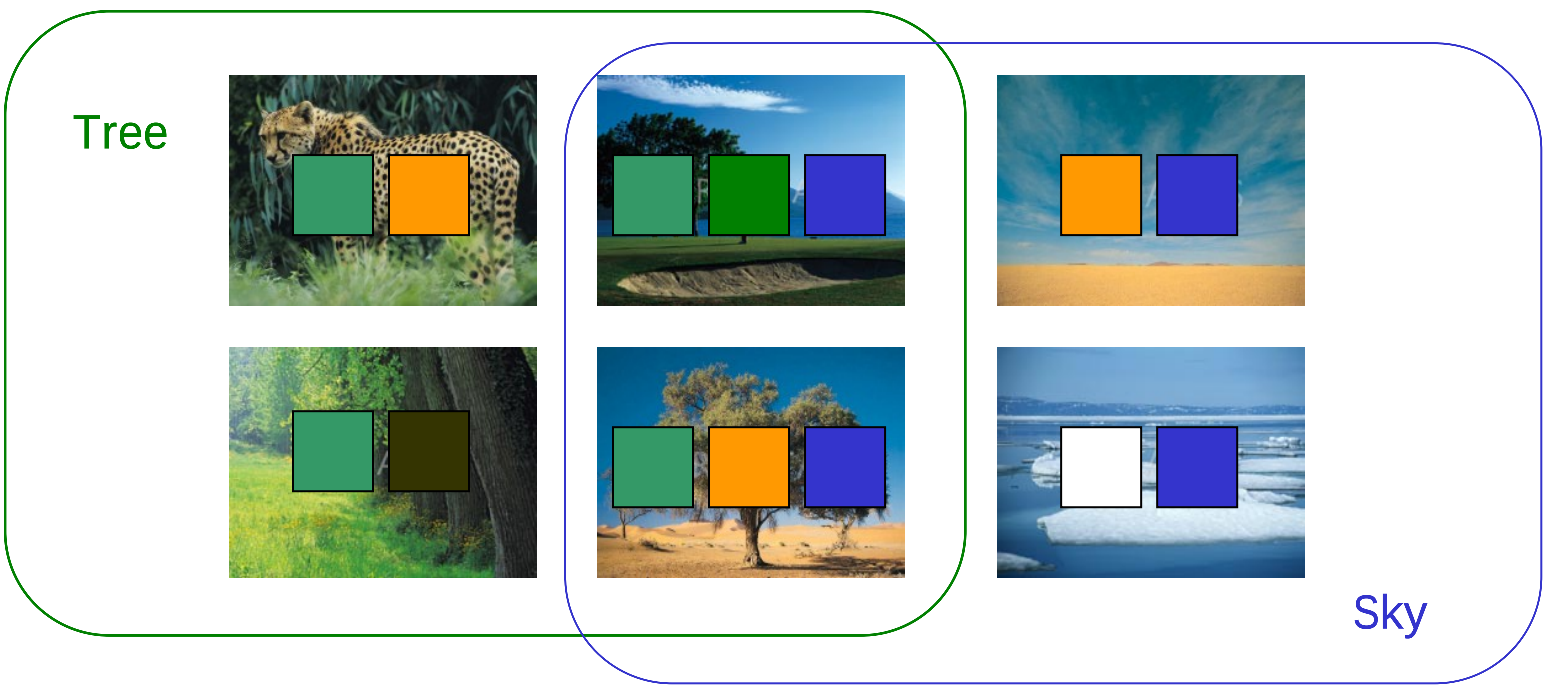
- The feature distribution of each object within a region is a Gaussian;

- Each image is a set of regions;

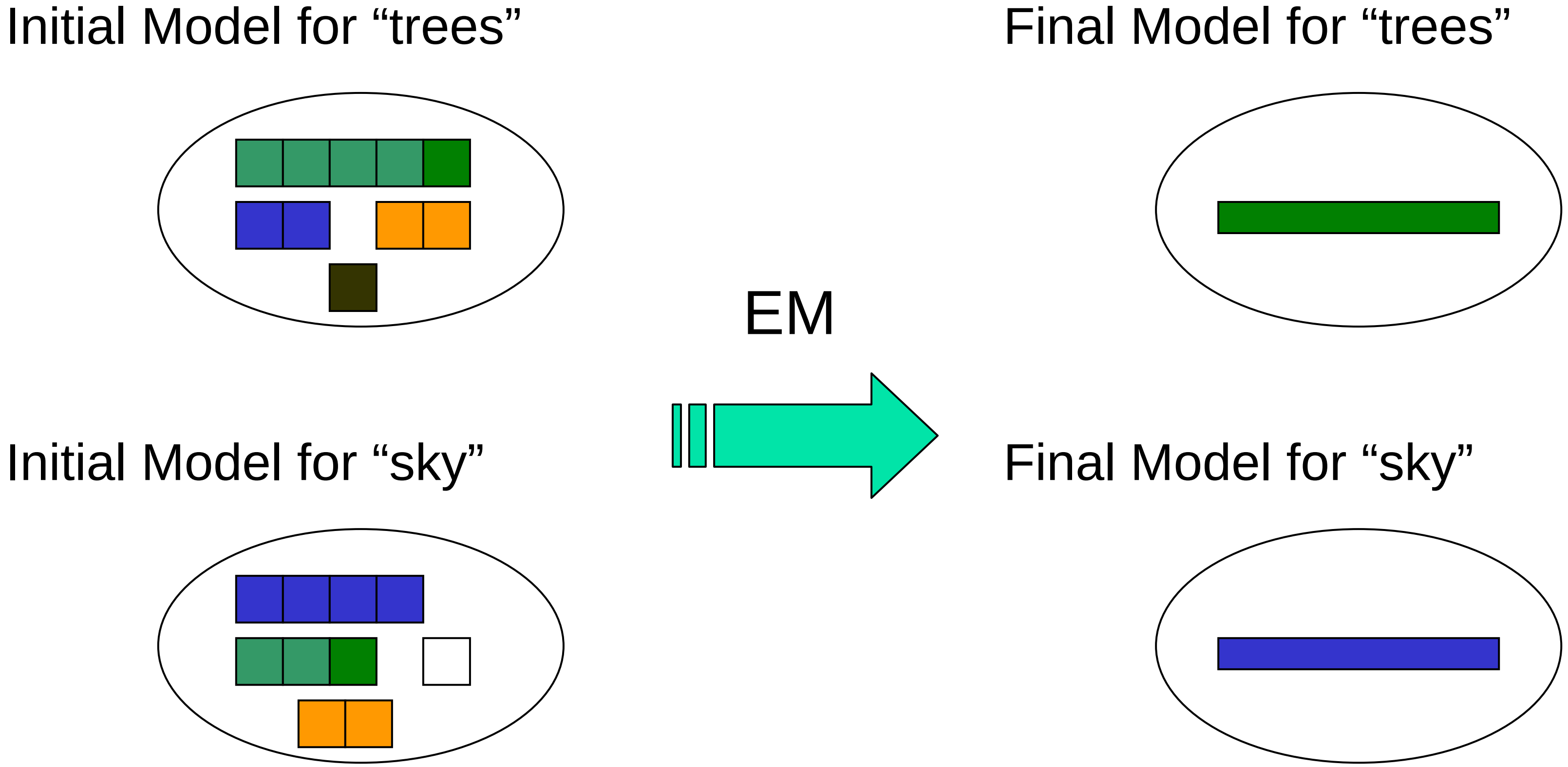
  each region can be modeled as a mixture of multivariate Gaussian distributions.

# Model Initial Estimation

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



# EM Variant

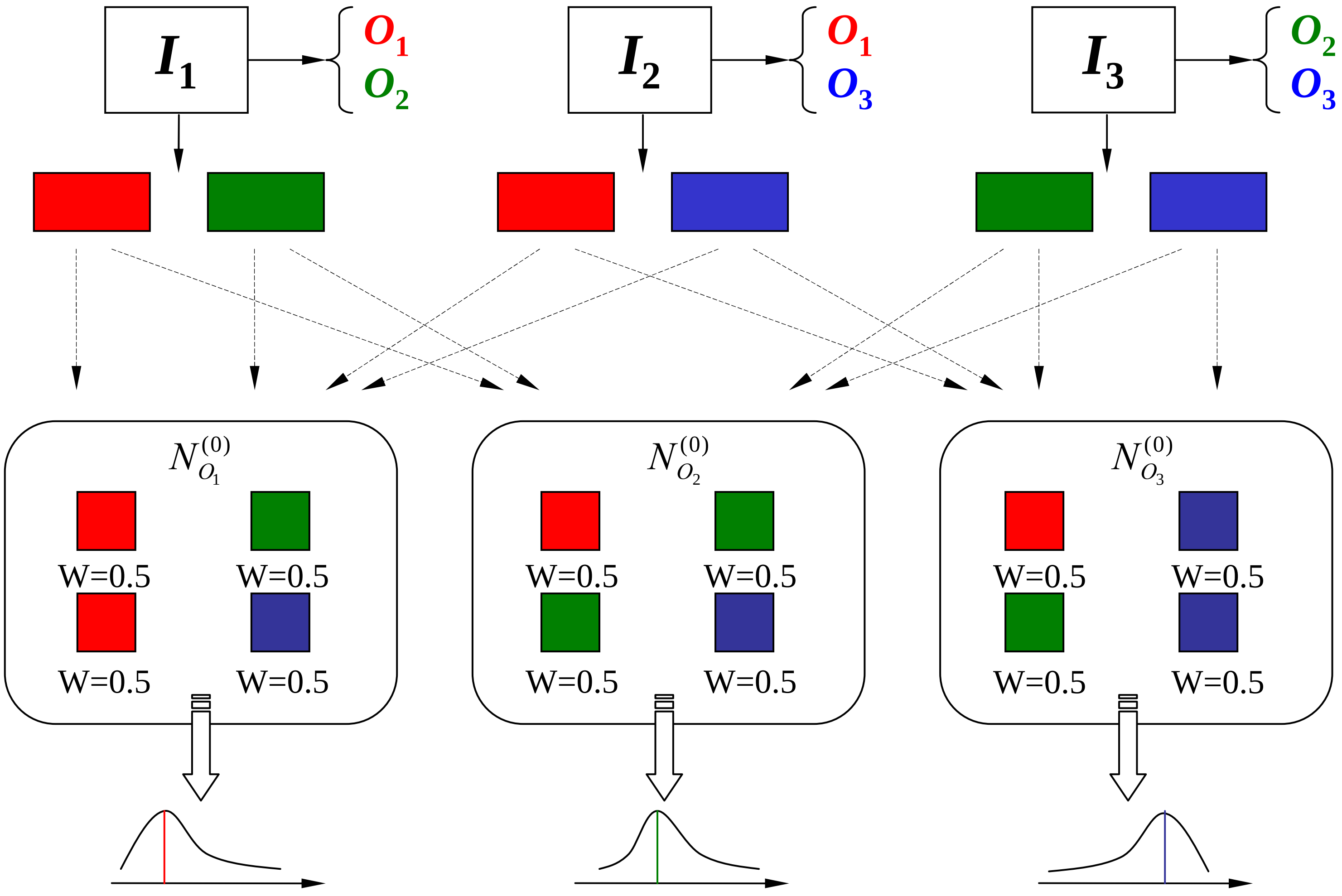

Initial Model for “trees”
Final Model for “trees”
EM
Initial Model for “sky”
Final Model for “sky”

# EM Variant

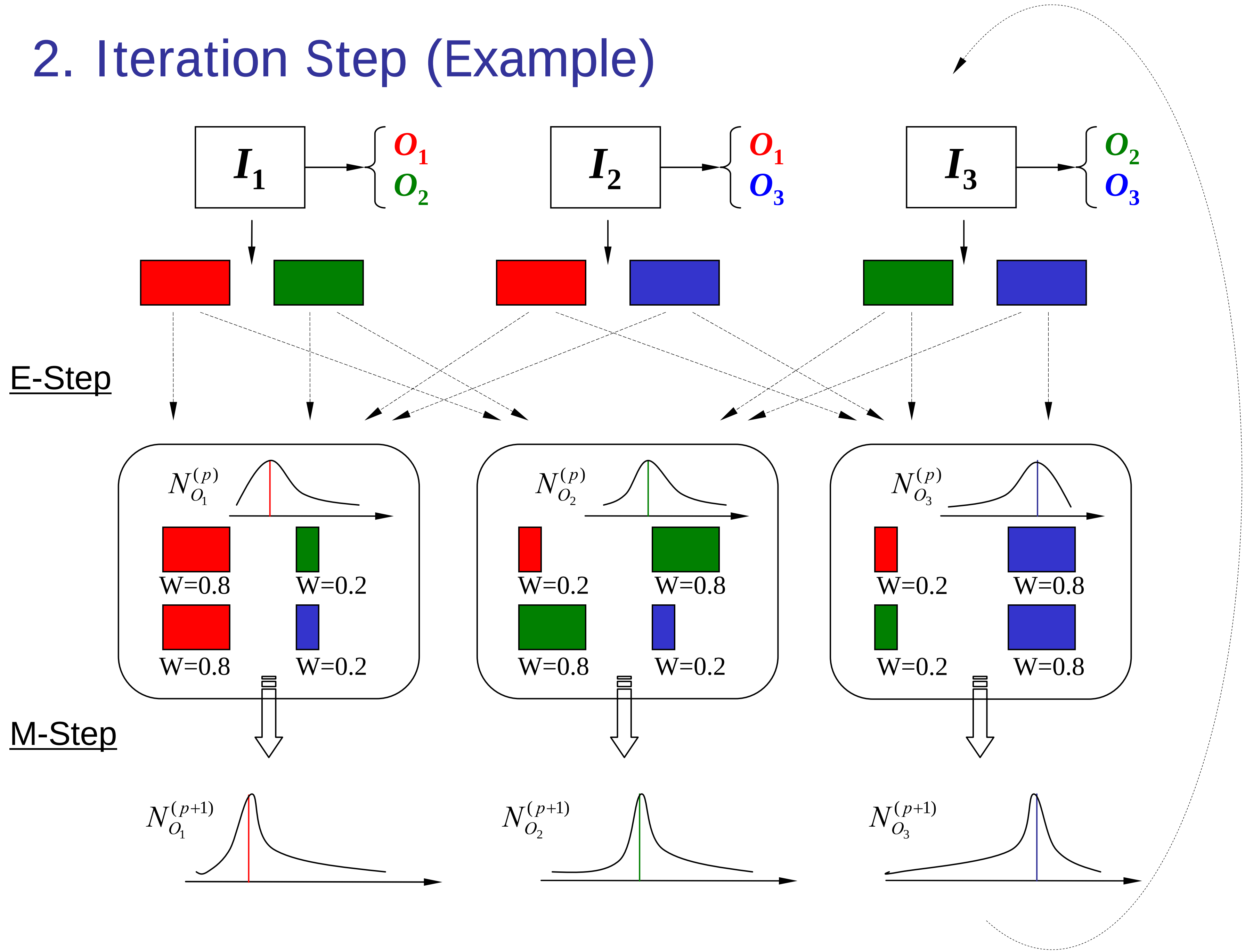
- Fixed components corresponding to the given object labels and fixed component responsibilities corresponding to the frequencies of the corresponding objects in the training data.
- Customized initialization takes advantage of known labels to generates more accurate estimates in the first step.
- Controlled posterior calculation ensures that a feature vector only contributes to the Gaussian components representing objects present in its training image.
- Extra background component absorbs noise.

# 1. Initialization Step (Example)

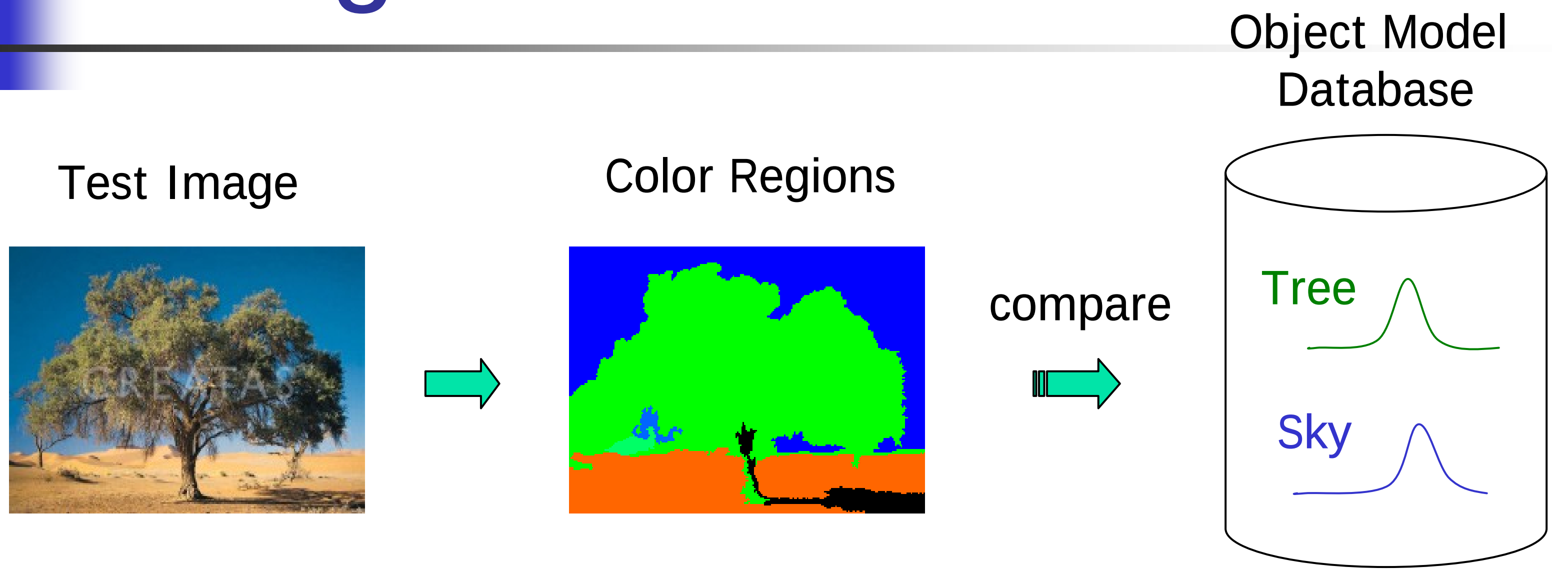
Image & description



# 2. Iteration Step (Example)



# Recognition




To calculate $p(tree \mid image)$

$$p(tree \mid image) = f\begin{pmatrix} p(\ tree \mid \quad ) \\ p(\ tree \mid \quad ) \\ p(\ tree \mid \quad ) \\ p(\ tree \mid \quad ) \end{pmatrix}$$

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

# Combining different abstract regions

- 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.

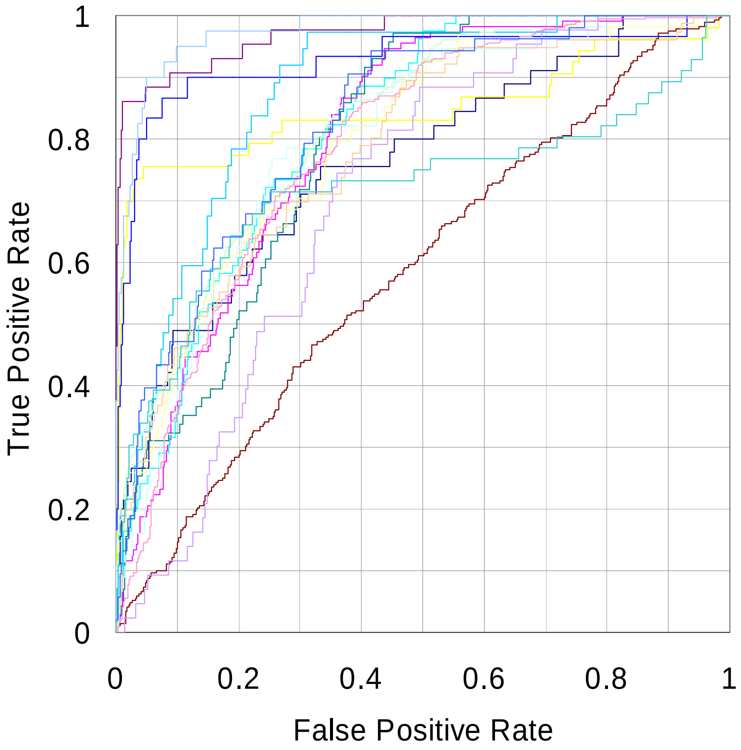
# Experiments (on 860 images)

- 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).

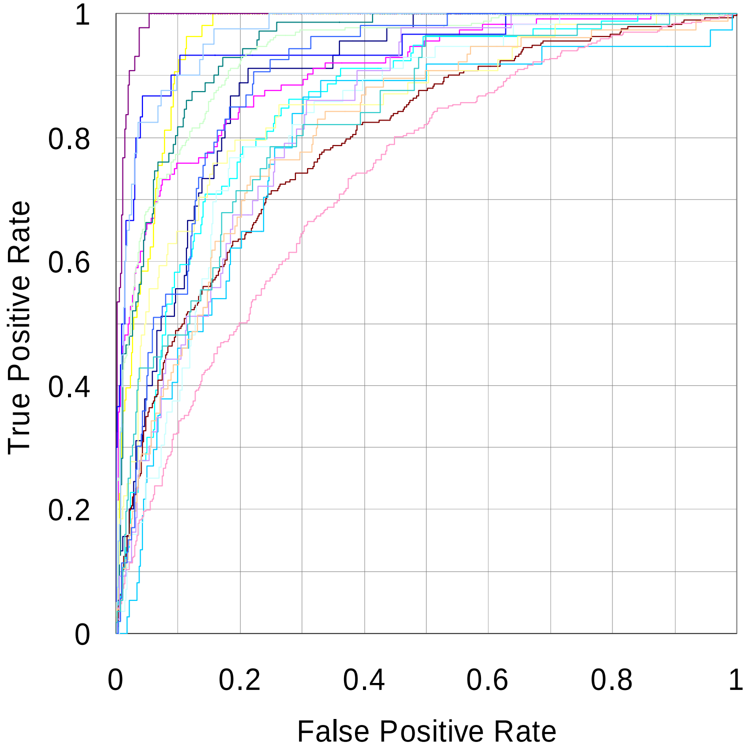
- A set of cross-validation experiments (80% as training set and the other 20% as test set)

- 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



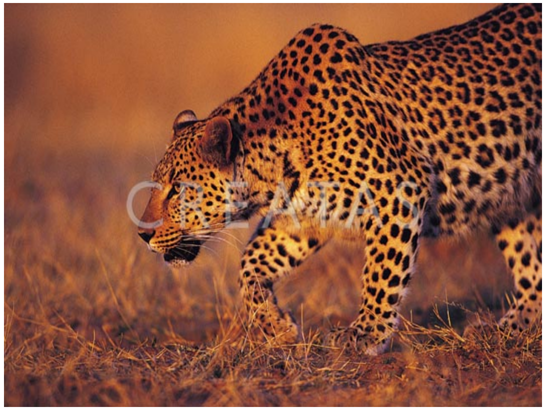

Independent Treatment of
Color and Texture




Using Intersections of
Color and Texture Regions

# Sample Results

cheetah



# Sample Results (Cont.)

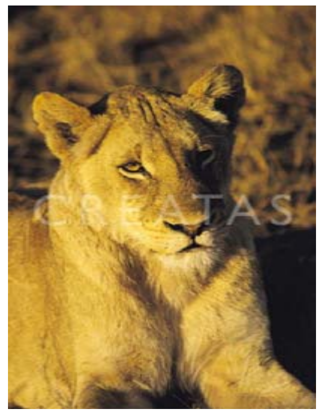
grass

# Sample Results (Cont.)

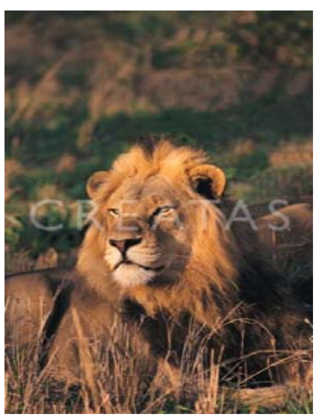
cherry tree

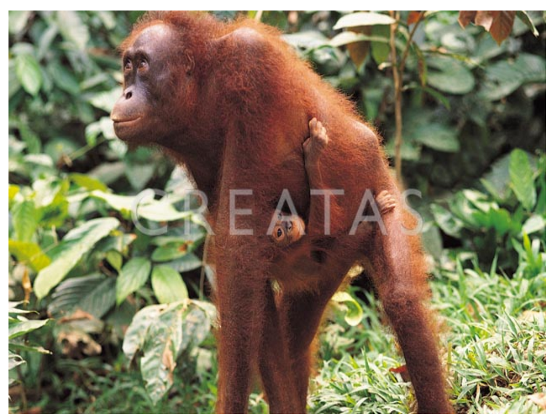

# Sample Results (Cont.)

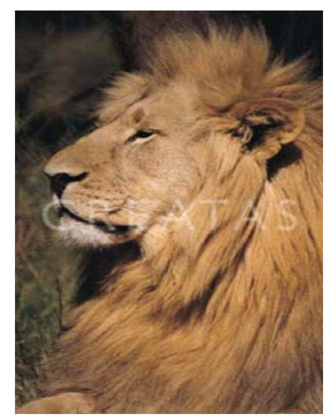
lion







# 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

# Our New 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 a model that is a mixture of 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 vector 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.
- The $w$'s are the weights of the components.
- The $\mu$'s and $\sum$'s are the parameters of the components

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

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

$$P(X_{i,r}^c, m^c) = w_m^c \cdot N(X_{i,r}^c, \mu_m^c, \Sigma_m^c)$$

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

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

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

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

$$
\begin{array}{c c}
\begin{array}{c} I_1^+ \\ I_2^+ \\ \vdots \\ I_1^- \\ I_2^- \\ \vdots \end{array} &
\left[ \begin{array}{cccc}
P(I_1^+, 1^c) & P(I_1^+, 2^c) & \cdots & P(I_1^+, M^c) \\
P(I_2^+, 1^c) & P(I_2^+, 2^c) & \cdots & P(I_2^+, M^c) \\
\vdots & & & \\
P(I_1^-, 1^c) & P(I_1^-, 2^c) & \cdots & P(I_1^-, M^c) \\
P(I_2^-, 1^c) & P(I_2^-, 2^c) & \cdots & P(I_2^-, M^c) \\
\vdots & & &
\end{array} \right]
\end{array}
$$

# Phase 2 Learning

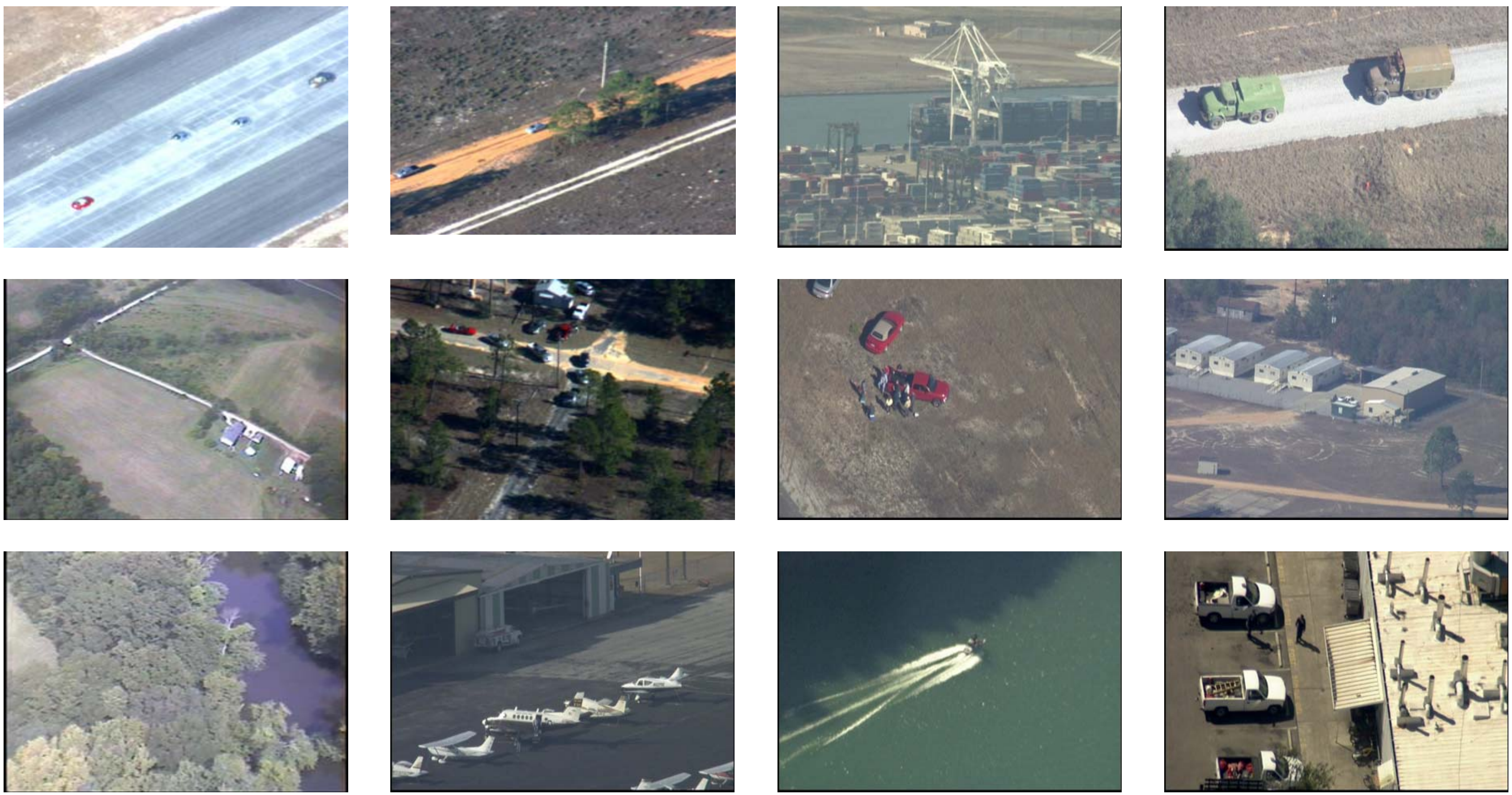
- Let $Y_{I_i}^{1^c:M^c}$ be row $i$ of the matrix.
- Each such row is a feature vector for the color features of regions of image $I_i$ that relates them to the Phase 1 components.
- Now we can use a second-stage classifier to learn $P(o|I_i)$ for each object class $o$ and image $I_i$ .

# Multiple Feature Case

- We calculate separate Gaussian mixture models for each different features type:
- Color: $Y_{I_i}^{1^c:M^c}$
- Texture: $Y_{I_i}^{1^t:M^t}$
- Structure: $Y_{I_i}^{1^s:M^s}$
- 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.

# VACE Test Image Set (828 images and 10 object classes): from Boeing, VIVID, and NGA videos

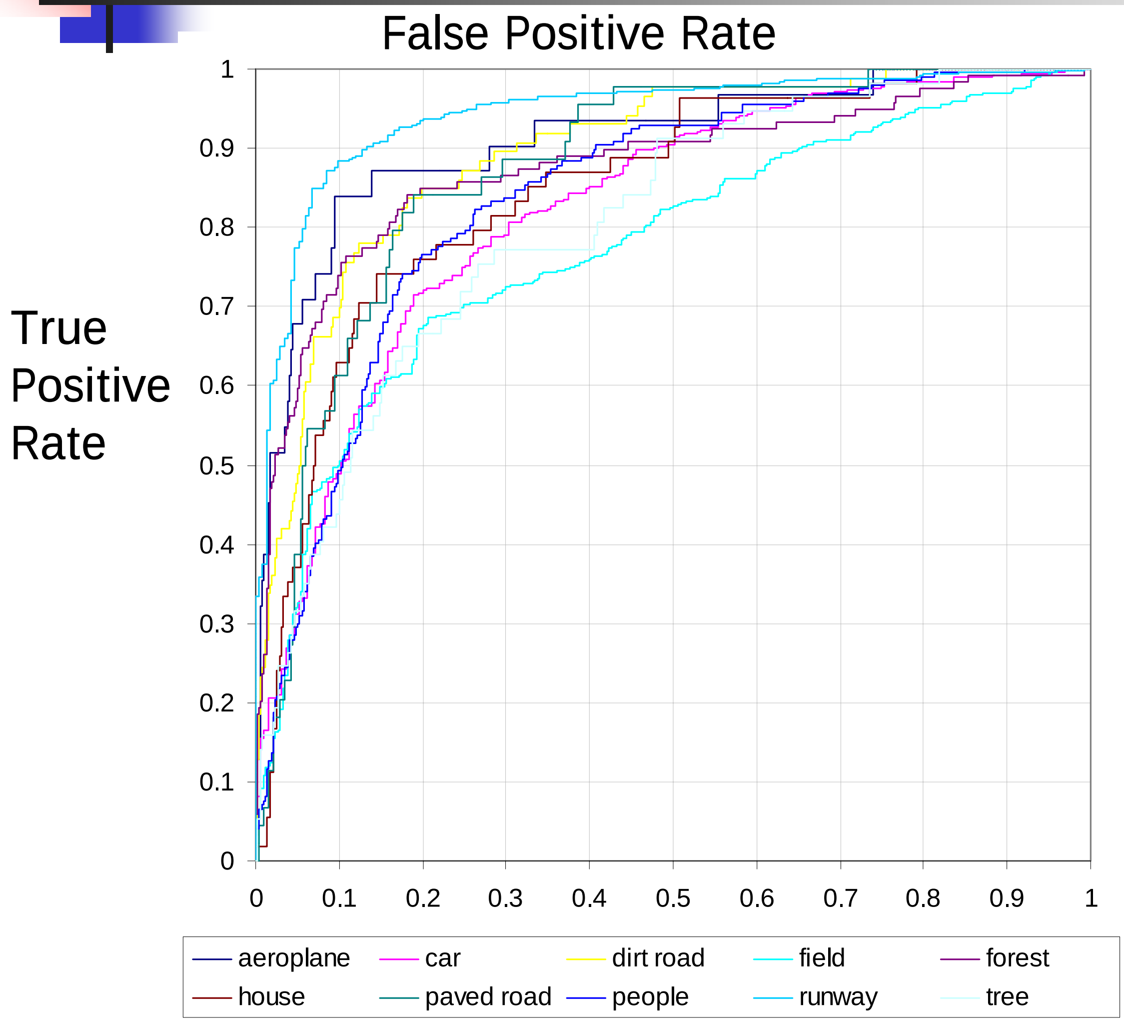


# Experiments

| | airplane | car | dirt road | field | forest | house | paved road | people | runway | tree |
|---|---|---|---|---|---|---|---|---|---|---|
| cs | 81.2 | 81.6 | 86.8 | 77.2 | 83.3 | 82.4 | 79.9 | **83.9** | 92.9 | 77.5 |
| st | 83.5 | 68.8 | 70.1 | 68.2 | 71.3 | 78.2 | 66.9 | 49.7 | 80.3 | 61.0 |
| cs+st | 90.1 | 78.9 | 86.4 | **77.5** | 86.4 | 83.7 | 81.5 | 83.9 | 93.9 | 77.5 |
| cs+ts | 78.4 | 81.1 | **89.5** | 74.2 | 86.7 | 80.8 | 79.8 | 83.8 | **94.4** | **80.6** |
| cs+ts+st | **91.1** | **82.3** | 88.1 | 74.1 | **87.6** | **84.9** | **87.5** | 79.7 | 93.6 | 77.1 |

*cs: color seg. ts: texture seg. st: structure

# Experiments: ROC Curves



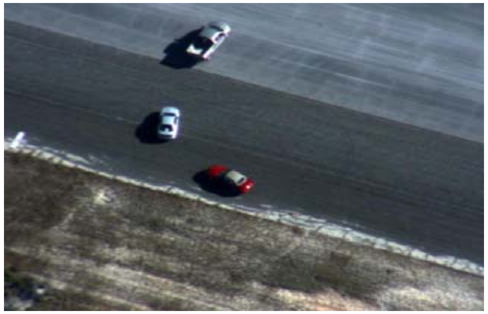

| | |
|---|---|
| field | 77.5 |
| tree | 80.6 |
| car | 82.3 |
| people | 83.9 |
| house | 84.9 |
| paved road | 87.5 |
| forest | 87.6 |
| dirt road | 89.5 |
| airplane | 91.1 |
| runway | 94.4 |

# Objects detected in frames

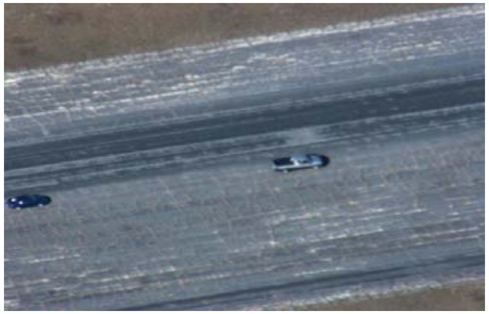
**forest**(94.37) **house**(64.09) car(46.5) dirt road(23.44) paved road(4.77) tree(2.29) airplane(1.47) runway(0.03) field(0.02) people(0)

**runway**(99.98) **field**(98.66) **car**(96.24) people(10.04) airplane(2.74) paved road(2.39) forest(0.82) house(0.48) dirt road(0.41) tree(0)

**car**(94.3) **dirt road**(91.7) **field**(16.17) tree(14.23) paved road(5.34) airplane(5.17) people(3.91) forest(0.53) house(0.47) runway(0.41)



**runway**(100) **car**(99.23) **field**(98.07) dirt road(92.1) house(85.24) tree(19.43) paved road(5.77) airplane(3.56) forest(2.85) people(0.07)

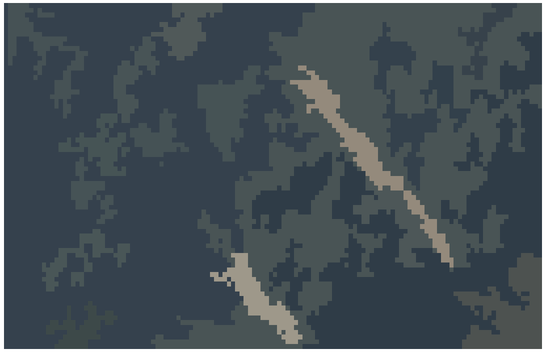


**runway**(99.98) **car**(99.84) **field**(99.27) paved road(18.28) people(13.13) tree(8.71) airplane(7.94) forest(1.67) house(0.14) dirt road(0.08)

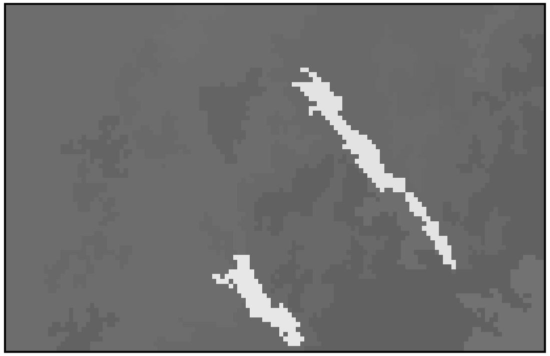
**car**(97.92) **forest**(94.2) **paved road**(85) **dirt road**(72.94) tree(68.84) airplane(39.13) house(33.17) people(12.97) field(2.38) runway(0.04)

# Locating Objects

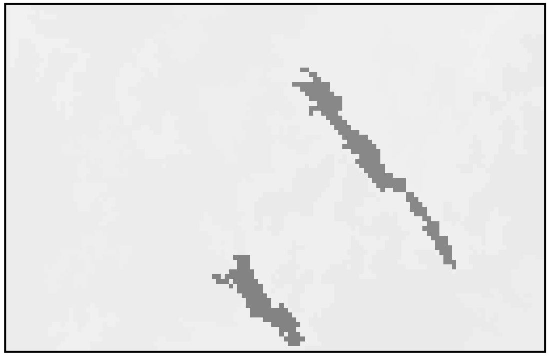
(Currently by color alone, developing an algorithm that will use multiple feature types.)



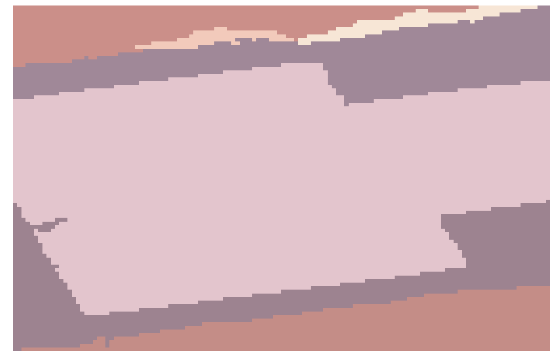
**dirt road** (90.6)



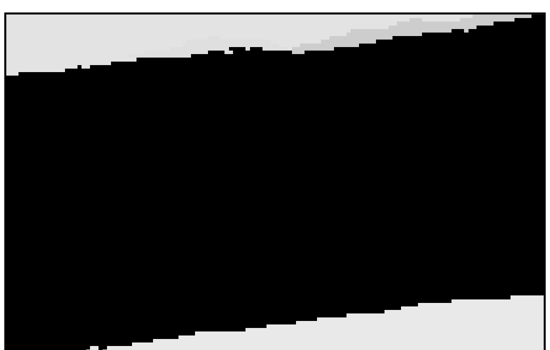
**forest** (94.8)



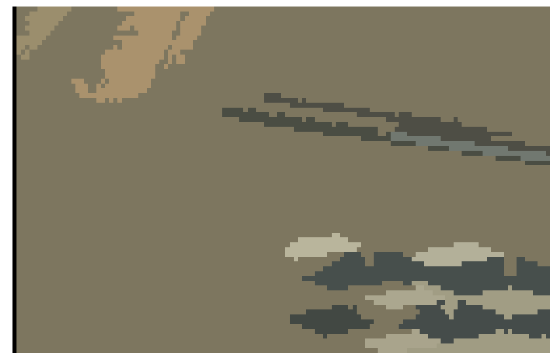
# Locating Objects



**field** (91.7) **runway** (100)



# Locating Objects



**field** (97)

**house** (81.1)

# Structure Feature Experiments

## (from other data sets with more manmade structures)

- 1,951 total from freefoto.com
- bus (1,013) house/building (609) skyscraper (329)

# Structure Feature Experiments: Area Under the ROC Curves

1. Structure (with color pairs)
   - Attributes (10)
     - Color pair
     - Number of lines
     - Orientation of lines
     - Line overlap
     - Line intersection

2. Structure (with color pairs) + Color Segmentation

3. Structure (without color pairs) + Color Segmentation

| | bus | house/ building | skyscraper |
|---|---|---|---|
| Structure only | 0.900 | 0.787 | 0.887 |
| Structure + Color Seg | 0.924 | 0.853 | 0.926 |
| $\text{Structure}^2$ + Color Seg | 0.940 | 0.860 | 0.919 |