COMPUTER VISION

Introduction

Computer vision is the analysis of digital images by a computer for such applications as:

- **Industrial**: part localization and inspection, robotics
- **Medical**: disease classification, screening, planning
- **Military**: autonomous vehicles, tank recognition
- **Intelligence Gathering**: face recognition, video analysis
- **Security**: video analysis
- **Science**: classification, measurement
- **Document Processing**: text recognition, diagram conversion
Medical Applications

CT image of a patient’s abdomen

Find the organs to avoid during radiation.

liver  kidney  kidney  spleen
Medical Applications

- child with cleft
- nose region

- depth area difference
- before surgery
- after surgery
- control
Medical Applications

Breast Cancer Biopsy Analysis
Robotics

Robot Navigation

Object Recognition

Kinect-Enabled Autonomous Mini-Car Navigation

Universität München

tas

technik autonemer systeme
3D Object Reconstruction

Building Rome in a Day

(a)

(b)

(c)
Image Databases:

Images from my Ground-Truth collection.

- Retrieve all images that have trees.
- Retrieve all images that have buildings.
- Retrieve all images that have antelope.
Surveillance: Object and Event Recognition in Aerial Videos

Original Video Frame

Color Regions

Structure Regions
The Three Stages of Computer Vision

• low-level (image processing)
  
  image → image

• mid-level (feature extraction)
  
  image → features

• high-level (the intelligent part)
  
  features → analysis
Low-Level

Canny edge operator

original image

edge image

Mid-Level (Lines and Curves)

ORT line & circle extraction

data structure

circular arcs and line segments
Mid-level (Regions)

K-means clustering (followed by connected component analysis)

original color image

regions of homogeneous color

data structure
Low- to High-Level

Building Recognition
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.
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 $f: U \rightarrow L$ satisfying

if $(u_1, u_2, ..., u_n) \in R$, then $(f(u_1), f(u_2), ..., f(u_n)) \in S$
House Example

2D model

2D image

RP and RL are connection relations.

\[ P = \{ S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9, S_{10}, S_{11} \}. \]

\[ L = \{ S_a, S_b, S_c, S_d, S_e, S_f, S_g, S_h, S_i, S_j, S_k, S_l, S_m \}. \]

\[ R_P = \{ (S_1, S_2), (S_1, S_5), (S_1, S_6), (S_2, S_3), (S_2, S_4), (S_3, S_4), (S_3, S_9), (S_4, S_5), (S_4, S_7), (S_4, S_{11}), (S_5, S_6), (S_5, S_7), (S_5, S_{11}), (S_6, S_8), (S_6, S_{11}), (S_7, S_8), (S_7, S_{10}), (S_7, S_{11}), (S_8, S_{10}), (S_8, S_{11}), (S_9, S_{10}) \}. \]

\[ R_L = \{ (S_a, S_b), (S_a, S_j), (S_a, S_n), (S_b, S_c), (S_b, S_d), (S_b, S_n), (S_c, S_d), (S_d, S_e), (S_d, S_f), (S_d, S_g), (S_e, S_f), (S_e, S_g), (S_f, S_l), (S_f, S_m), (S_g, S_h), (S_g, S_i), (S_g, S_n), (S_h, S_i), (S_h, S_k), (S_h, S_l), (S_h, S_n), (S_i, S_j), (S_i, S_k), (S_i, S_n), (S_j, S_k), (S_k, S_l), (S_l, S_m) \}. \]

\[ f(S_1) = S_j \quad f(S_4) = S_n \quad f(S_7) = S_g \quad f(S_{10}) = S_f \]

\[ f(S_2) = S_a \quad f(S_5) = S_i \quad f(S_8) = S_l \quad f(S_{11}) = S_h \]

\[ f(S_3) = S_b \quad f(S_6) = S_k \quad f(S_9) = S_d \]
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. STILL TOO MUCH OF A TOY PROBLEM.
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: L, V, Y, Z, U
- Triples
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 coaxial multi

encloses

1

encloses

2 ellipse

encloses

2

c

c

3 parallel lines

coaxial

3

RDF!
Relational Indexing for Recognition

Preprocessing (off-line) Phase

for each model view $M_i$ in the database

- **encode** each 2-graph of $M_i$ to produce an index
- store $M_i$ 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

The diagram illustrates the voting process with the following components:

- **Ellipse** connected to **coaxial arc cluster** sharing an arc.
- **Hash function** leading to the index (1,29,9).
- CLI 2-graph with **List of Models** containing models M1, M5, M23, M81.
- Retrived list of models: M1, M5, M23, M81.
- Accumulators with votes: M1, M5, M23, M81.
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”

Initial Model for “sky”

Final Model for “sky”

EM
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 $\mu_j$, variance $\sigma_j$, and weight $\alpha_j$.

\[
\begin{align*}
N(\mu_1, \sigma_1) & \quad \alpha_1 = P(C_1) \\
N(\mu_2, \sigma_2) & \quad \alpha_2 = P(C_2) \\
N(\mu_3, \sigma_3) & \quad \alpha_3 = P(C_3)
\end{align*}
\]

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

\[ I_1 \rightarrow O_1, O_2 \]
\[ I_2 \rightarrow O_1, O_3 \]
\[ I_3 \rightarrow O_2, O_3 \]
2. Iteration Step (Example)

E-Step

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

To calculate $p(\text{tree} \mid \text{image})$

$$p(\text{tree} \mid \text{image}) = f \left( \begin{array}{c} p(\text{tree} \mid \text{image}) = \frac{1}{3} \left( p(\text{tree} \mid \text{image}) + 2 \cdot p(\text{tree} \mid \text{image}) \right) \end{array} \right)$$

$$p(o \mid F^a_i) = f \left( \sum_{r^a \in F^a_i} p(o \mid r^a) \right)$$

$f$ 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 \( o \).

\[
P(X^c|o) = \sum_{m=1}^{M^c} w^c_m \cdot N(X^c; \mu^c_m, \Sigma^c_m)
\]

- \( M^c \) is the number of components (clusters).
- The \( w^c \)'s are the weights (\( \alpha \)'s) of the components.
- The \( \mu^c \)'s and \( \Sigma^c \)'s are the parameters of the components.
- \( N(X^c, \mu^c_m, \Sigma^c_m) \) specifies the probability that \( X^c \) belongs to a particular normal distribution.
Color Components for Class $o$

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

component 1
$\mu_1, \Sigma_1, w_1$

component 2
$\mu_2, \Sigma_2, w_2$

component $M^c$
$\mu_M, \Sigma_M, w_M$

color feature vector $X^c$ for region $r$
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

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

$$f_{x}(x_1, \ldots, x_k) = \frac{1}{(2\pi)^{k/2}|\Sigma|^{1/2}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)$$
And determine the probability that the whole image is related to component \( m \) as a function of the feature vectors of all its regions.

- 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
Aggregate Scores for Color

Components

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
</table>

beach

|   | .93 | .16 | .94 | .24 | .10 | .99 | .32 | .00 |

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.

\[
\begin{bmatrix}
  P(I_1^+, 1^c) & P(I_1^+, 2^c) & \cdots & P(I_1^+, M^c) \\
  P(I_2^+, 1^c) & P(I_2^+, 2^c) & \cdots & P(I_2^+, M^c) \\
  \vdots & \vdots & \ddots & \vdots \\
  P(I_1^-, 1^c) & P(I_1^-, 2^c) & \cdots & P(I_1^-, M^c) \\
  P(I_2^-, 1^c) & P(I_2^-, 2^c) & \cdots & P(I_2^-, M^c) \\
  \vdots & \vdots & \ddots & \vdots 
\end{bmatrix}
\]
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.
ICPR04 Data Set with General Labels

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

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

<table>
<thead>
<tr>
<th>Number of top-ranked categories required</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALIP</td>
<td>11.88</td>
<td>17.06</td>
<td>20.76</td>
<td>23.24</td>
<td>26.05</td>
</tr>
<tr>
<td>Gen/Dis</td>
<td>11.56</td>
<td>17.65</td>
<td>21.99</td>
<td>25.06</td>
<td>27.75</td>
</tr>
</tbody>
</table>

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

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

Asian city

Cannon beach

Italy

park
Groundtruth Data Set: Top Results

- **sky**
  - Images of various sky conditions and landscapes.

- **spring flowers**
  - Images of spring flowers in different settings.

- **tree**
  - Images of trees in various environments.

- **water**
  - Images of water bodies in different scenarios.
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)