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

Sample Retrieval Results

cheetah

Sample Results (Cont.)

grass



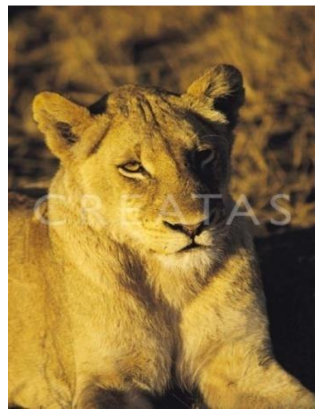


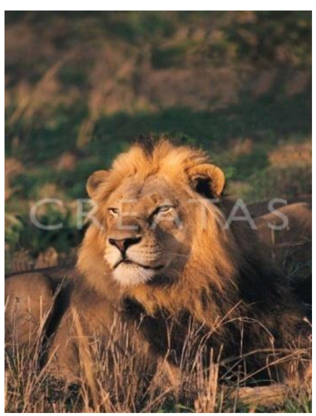
Sample Results (Cont.)

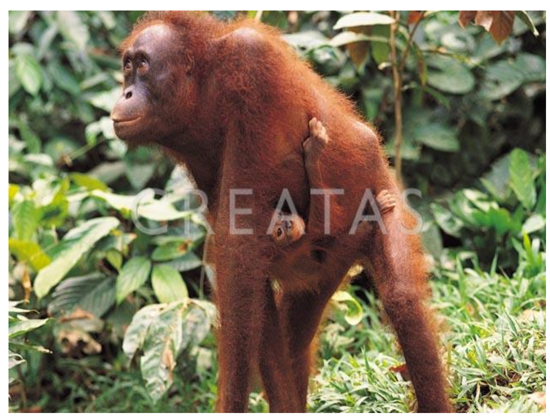
cherry tree

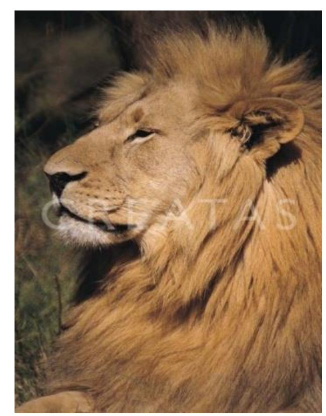
Sample Results (Cont.)

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)

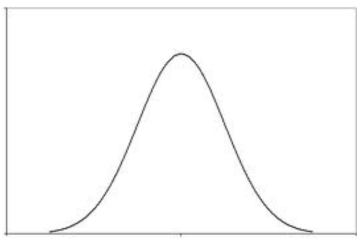
- At the end of Phase 1, we can compute a probability distribution of color feature vectors in an image containing object *o*.

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

- M^c is the number of components (clusters).
- The *w's* are the weights (α's) of the components.
- The *μ's* and *∑'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.

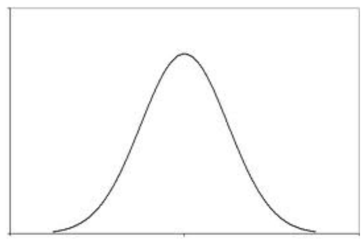
Color Components for Class o

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



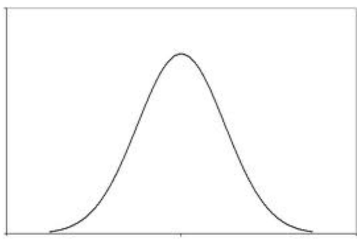
component 1

μ_1, Σ_1, w_1



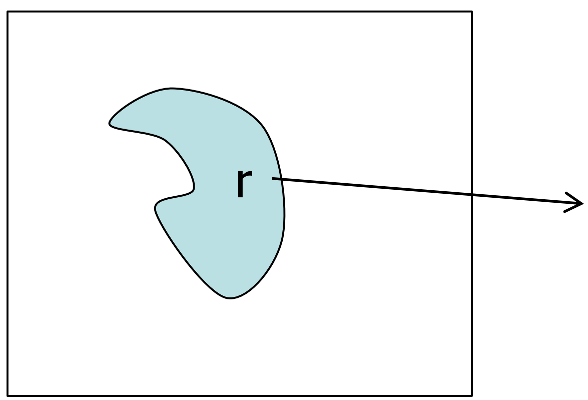
component 2

μ_2, Σ_2, w_2



component M^c

μ_M, Σ_M, w_M



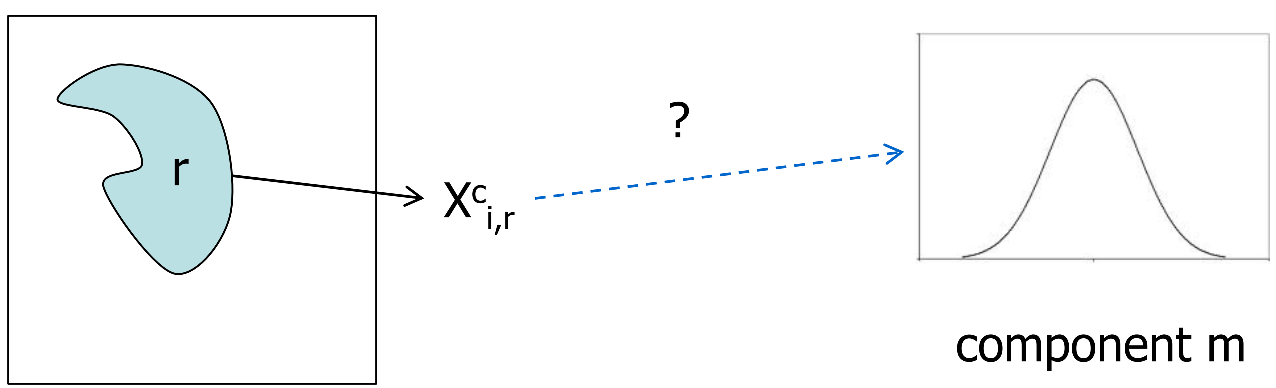
color feature vector
X^c for region r

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

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

$$\boxed{f_{\mathbf{x}}(x_1, \ldots, x_k) = \frac{1}{(2\pi)^{k/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)}$$



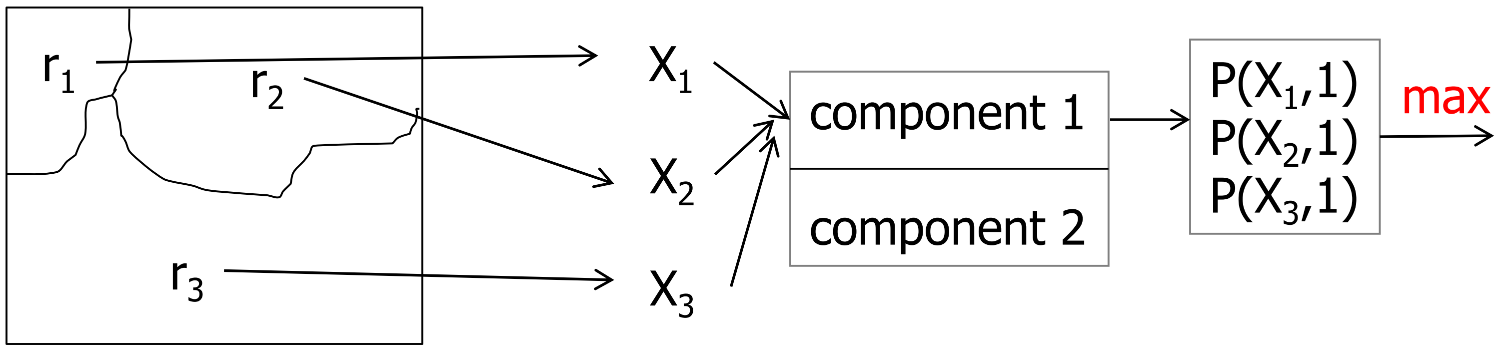
component m

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

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

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



Aggregate Scores for Color

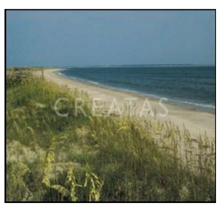
Components

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.

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

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.

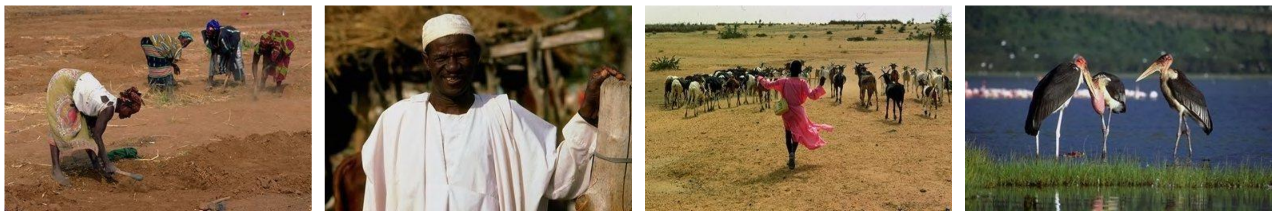
- 10 classes (*African people*, *beach*, *buildings*, *buses*, *dinosaurs*, *elephants*, *flowers*, *food*, *horses*, *mountains*). 100 images each.

Comparison to ALIP: the Benchmark Image Set

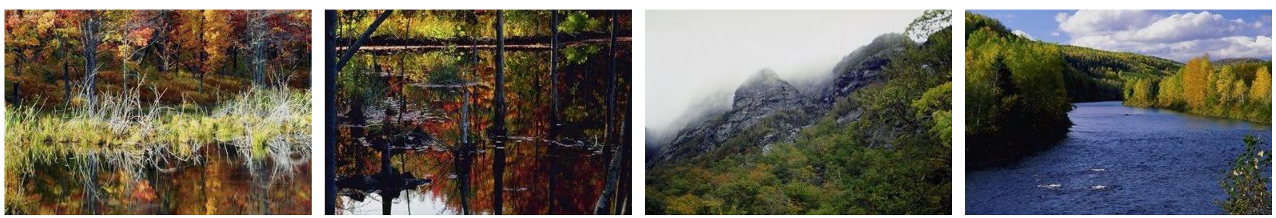
	ALIP	cs	ts	st	ts+st	cs+st	cs+ts	cs+ts+st
African	52	69	23	26	35	79	72	74
beach	32	44	38	39	51	48	59	64
buildings	64	43	40	41	67	70	70	78
buses	46	60	72	92	86	85	84	95
dinosaurs	100	88	70	37	86	89	94	93
elephants	40	53	8	27	38	64	64	69
flowers	90	85	52	33	78	87	86	91
food	68	63	49	41	66	77	84	85
horses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



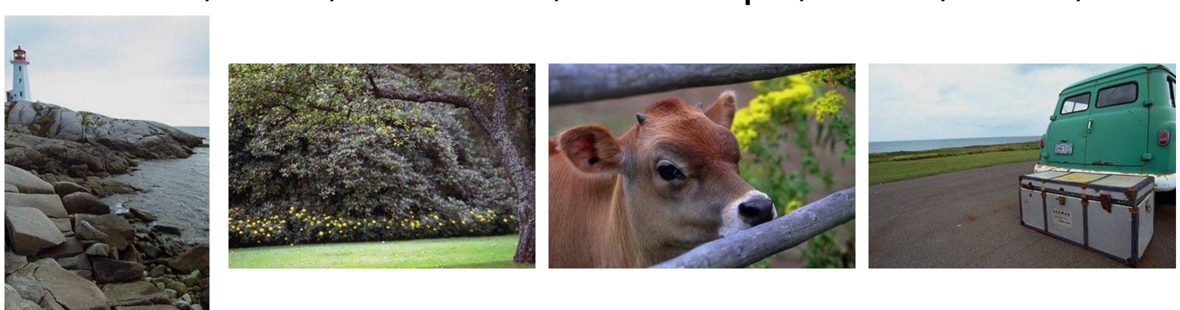
2. Bhutan, Asia, people, landscape, church

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

sky

spring flowers

tree

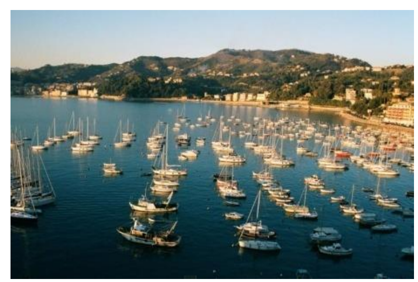
water

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)

Object detection, deep learning, and R-CNNs

Partly from Ross Girshick

Microsoft Research

Now at Facebook

Outline

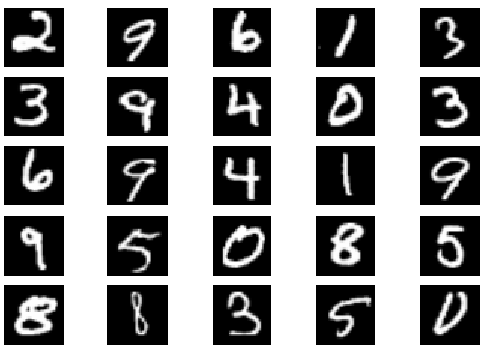
- Object detection
  - the task, evaluation, datasets

- Convolutional Neural Networks (CNNs)
  - overview and history

- Region-based Convolutional Networks (R-CNNs)

Image classification

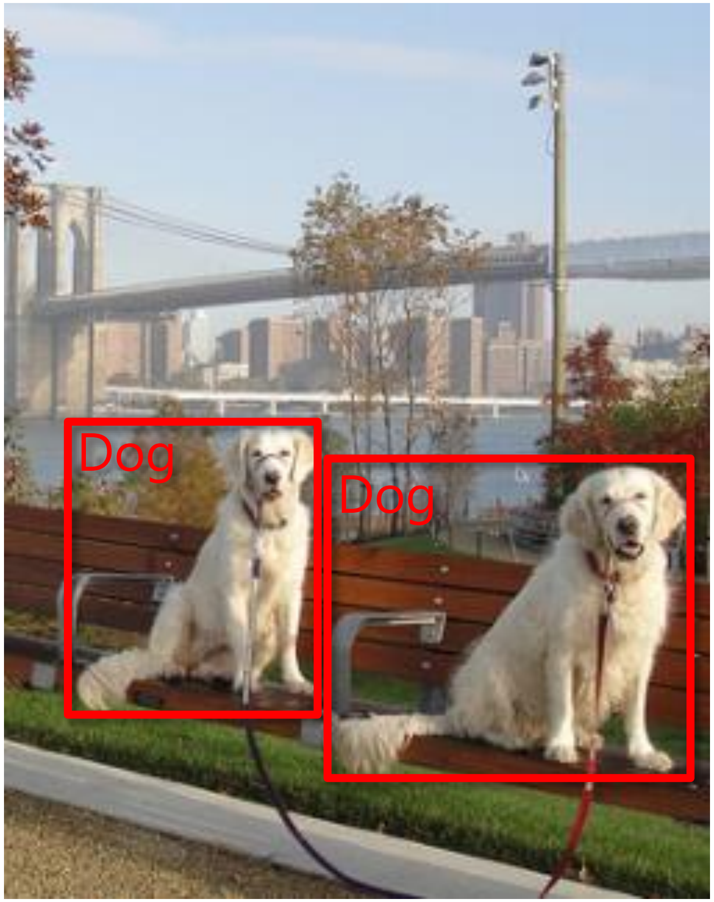
- K classes
- Task: assign correct class label to the whole image



Digit classification (MNIST)

Object recognition (Caltech-101)

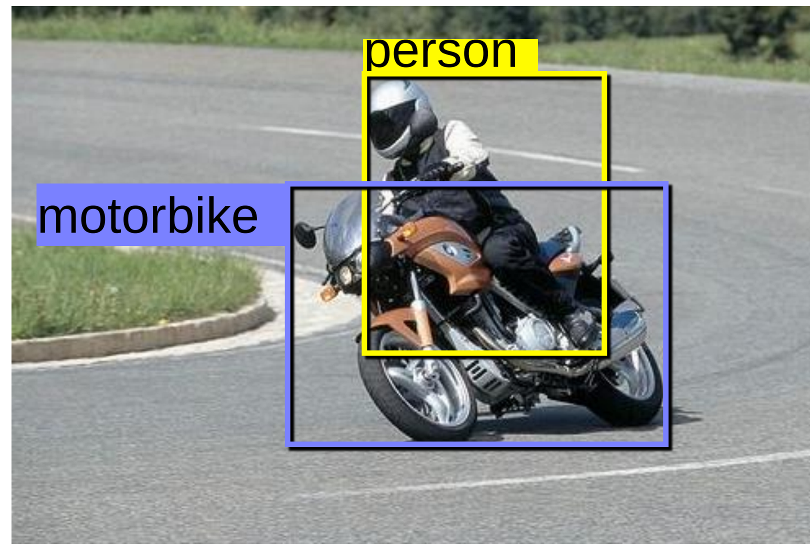
Classification vs. Detection



Problem formulation

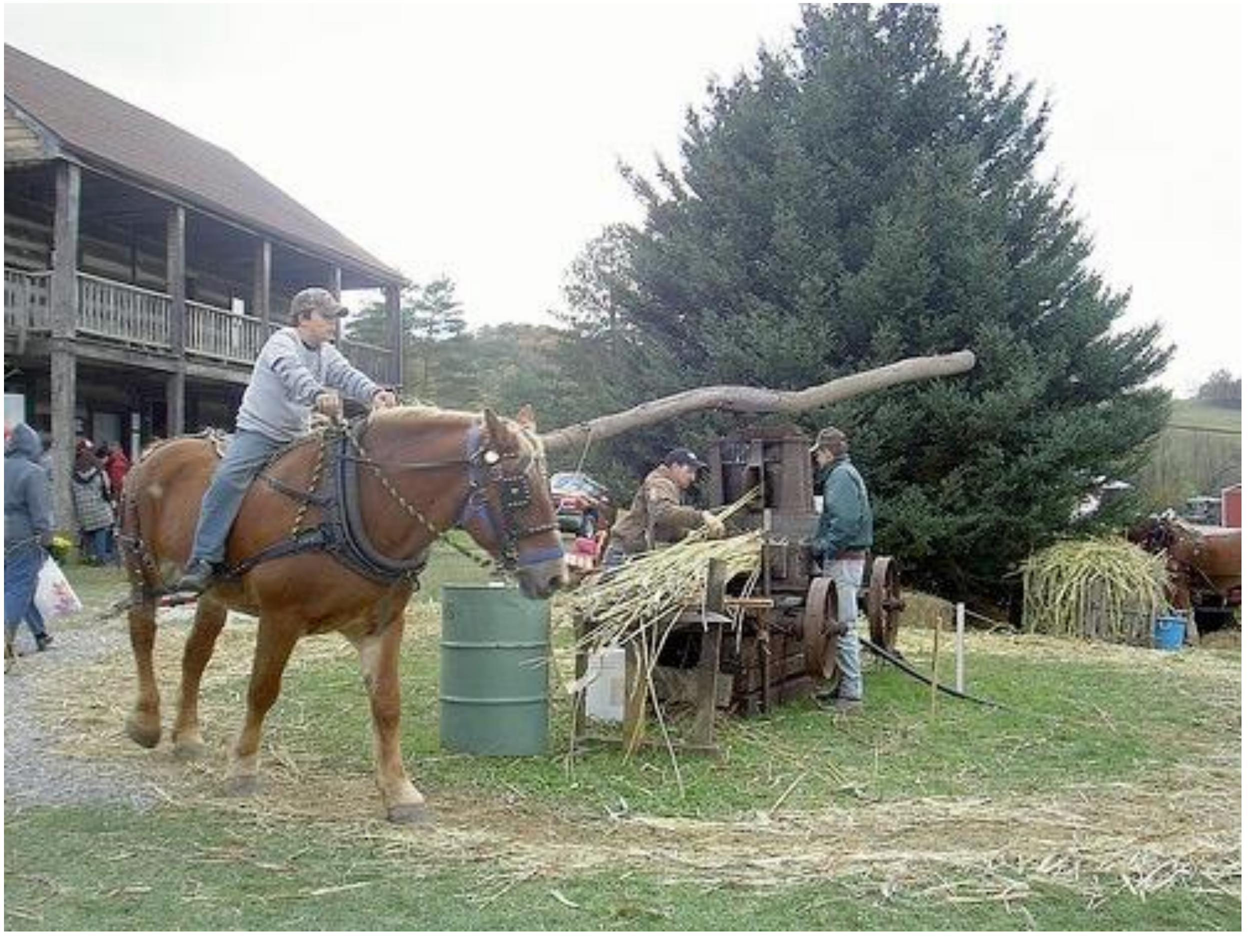
{ airplane, bird, motorbike, person, sofa }

Input



Desired output

Evaluating a detector

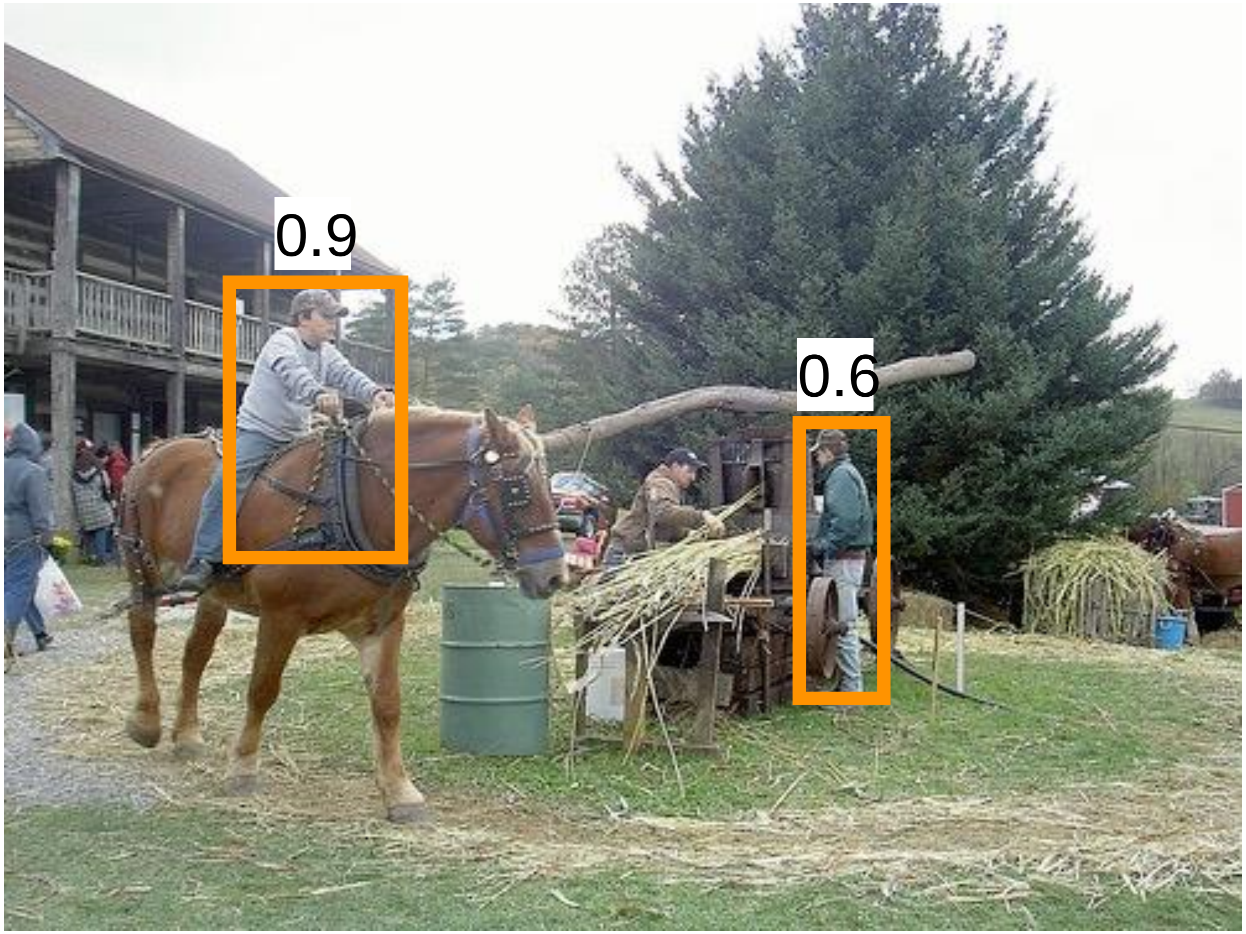


Test image (previously unseen)

First detection ...

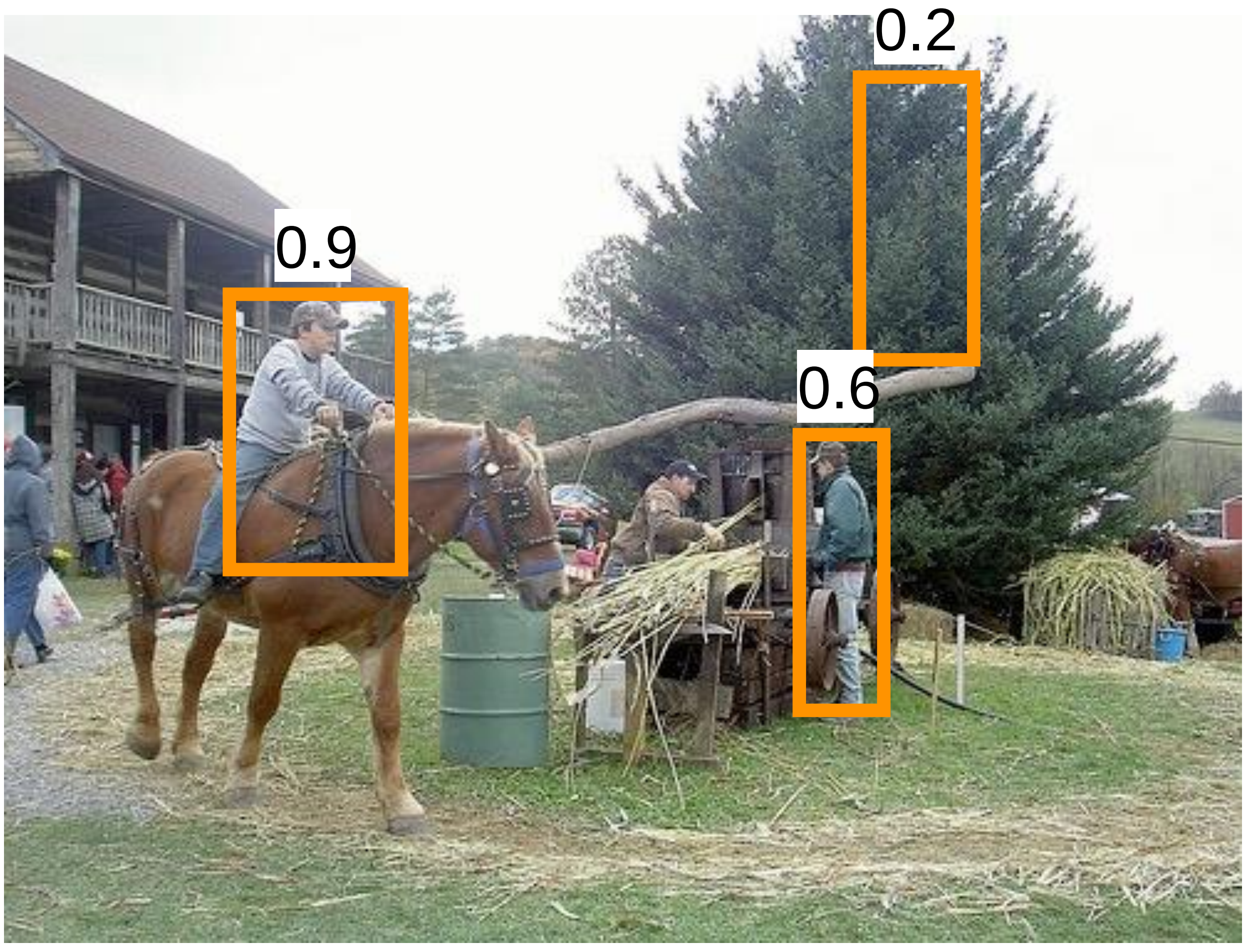
‘person’ detector predictions

Second detection ...



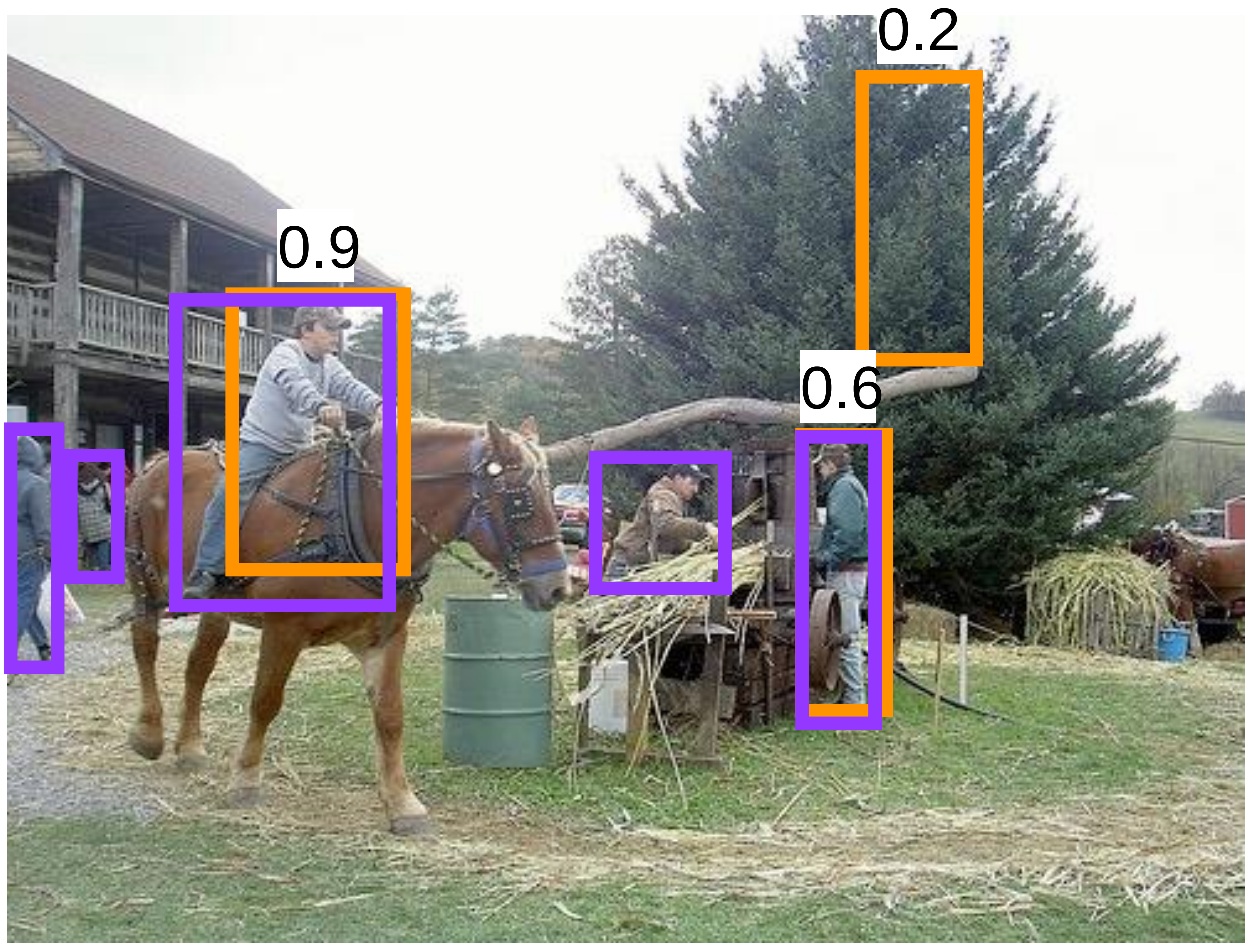
'person' detector predictions

Third detection ...



‘person’ detector predictions

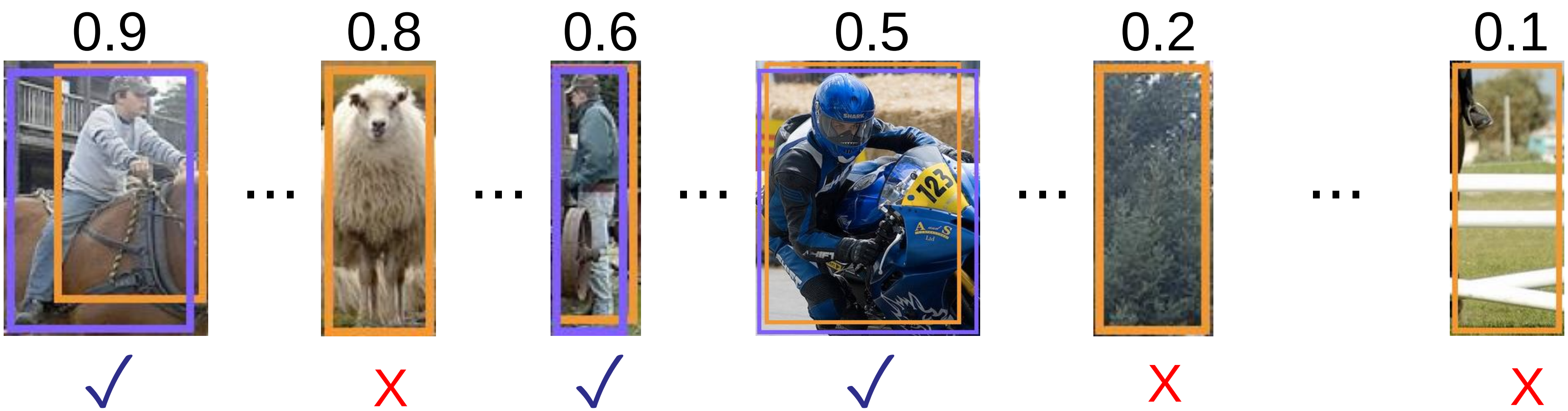
Compare to ground truth



‘person’ detector predictions

ground truth ‘person’ boxes

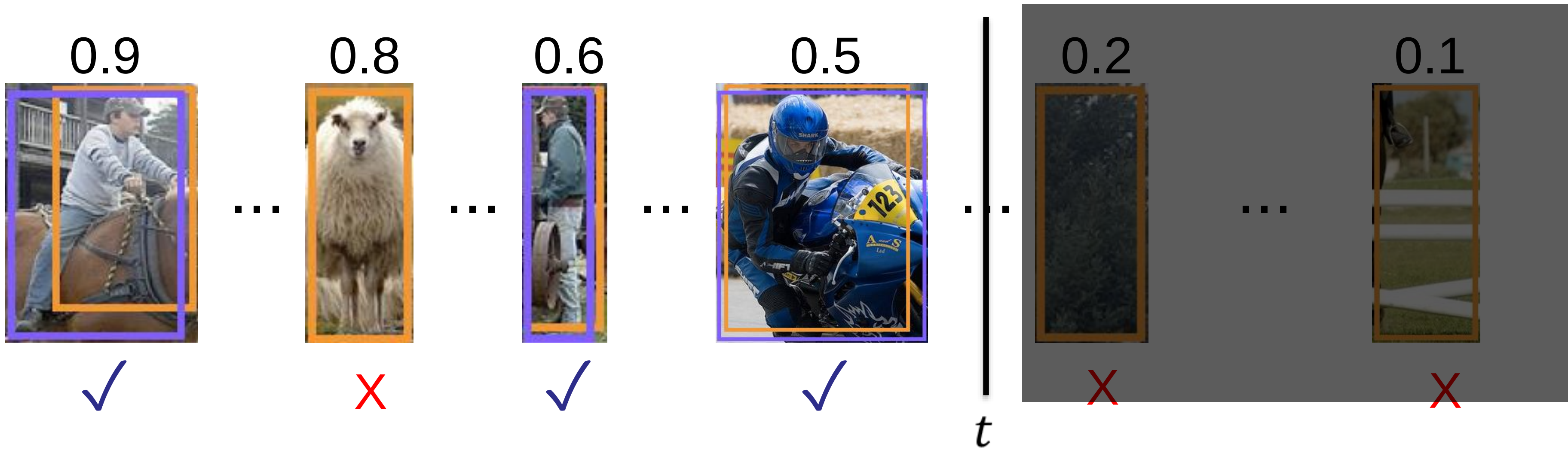
Sort by confidence



true positive
(high overlap)

false positive
(no overlap, low overlap, or duplicate)

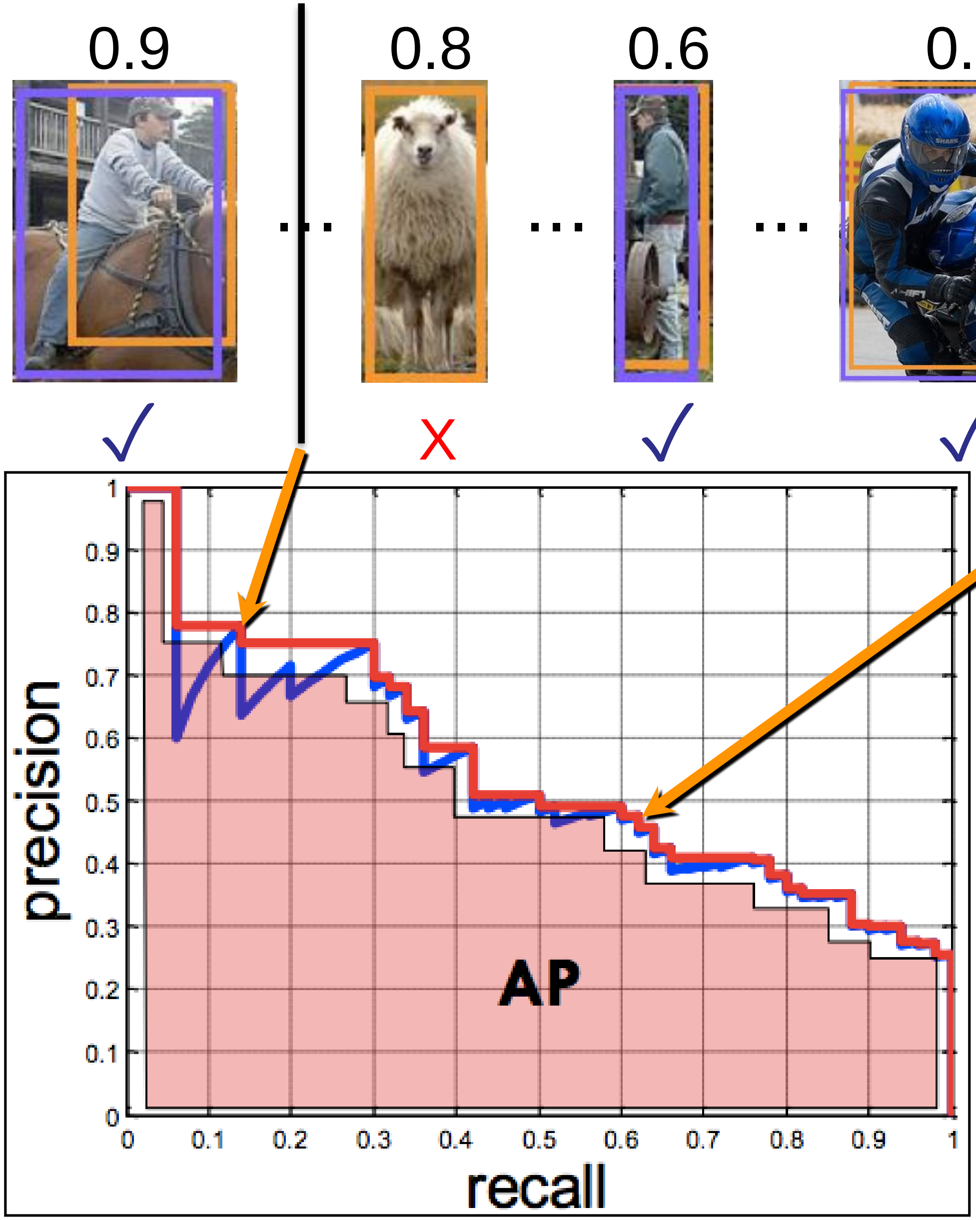
Evaluation metric



$$precision@t = \frac{\#true\ positives@t}{\#true\ positives@t + \#false\ positives@t} \qquad \frac{\checkmark}{\checkmark + X}$$

$$recall@t = \frac{\#true\ positives@t}{\#ground\ truth\ objects}$$

Evaluation metric



Average Precision (AP)

0% is worst

100% is best

mean AP over classes (mAP)