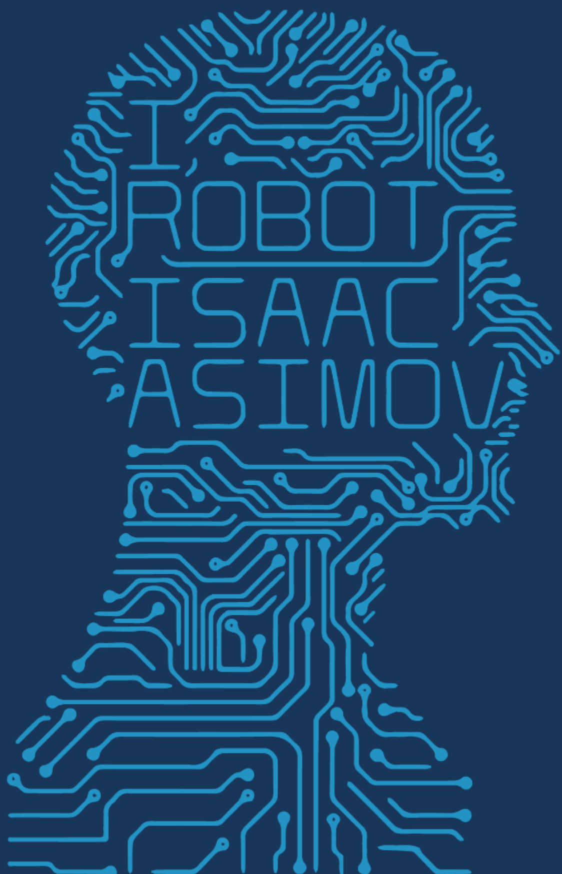
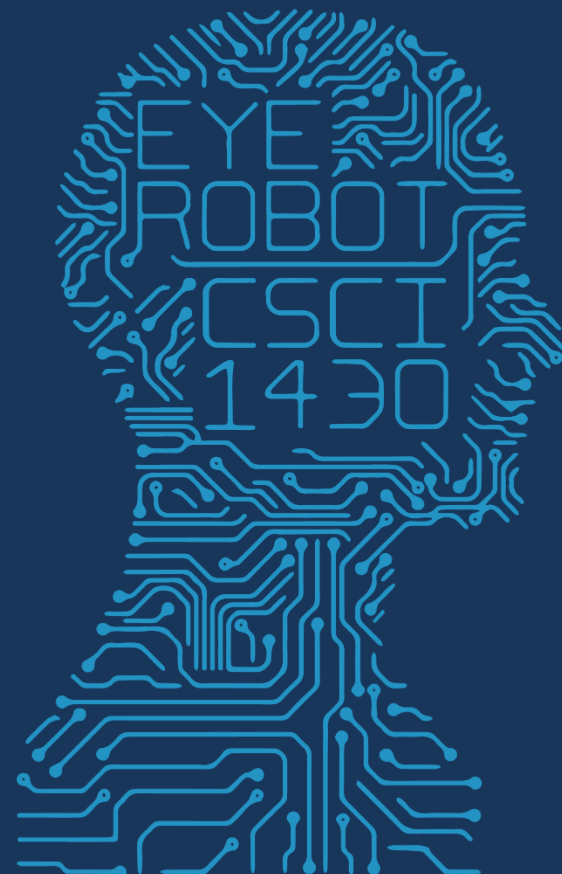


1950

Future Vision



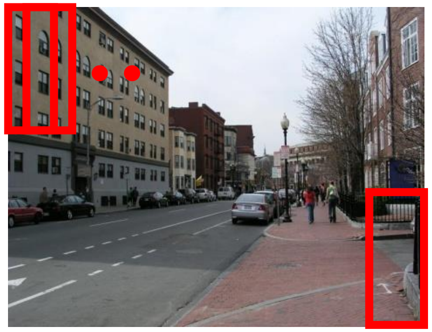
2020

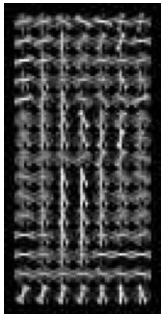
Computer Vision

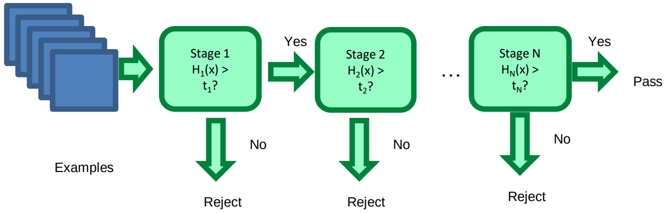

note:
black & white

Object detection

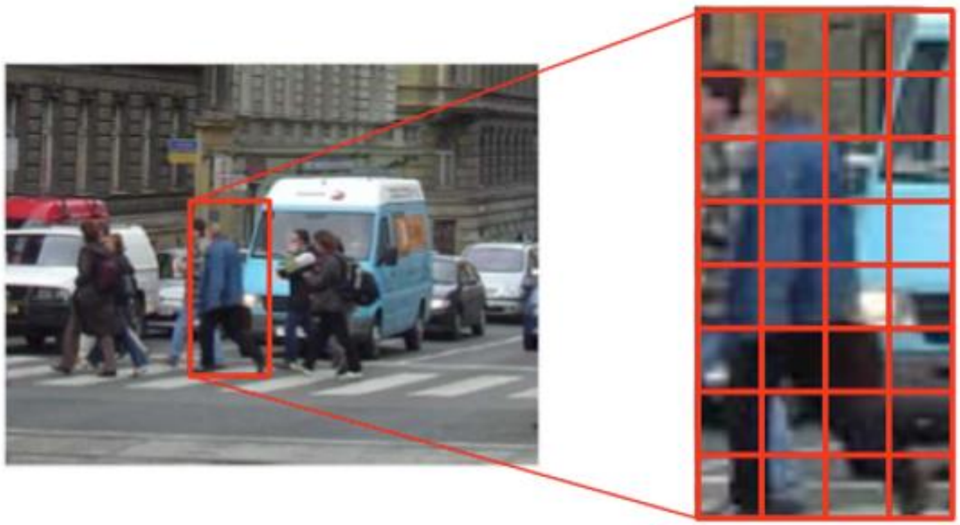
- Sliding window for search
- Features based on differences of intensity (gradient, wavelet, etc.)
- Boosting for feature selection
- Integral images, cascade for speed
- Bootstrapping to deal with many, many negative examples







Starting point: sliding window classifiers

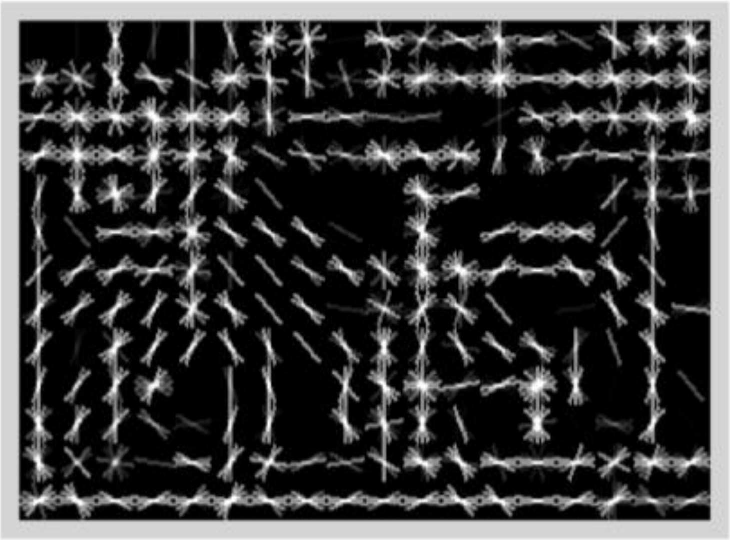


Feature vector
$x = [\ldots, \ldots, \ldots, \ldots]$

- Detect objects by testing each subwindow
  - Reduces object detection to binary classification
  - Dalal & Triggs: HOG features + linear SVM classifier
  - Previous state of the art for detecting people

Felzenszwalb

Histogram of Gradient (HOG) features

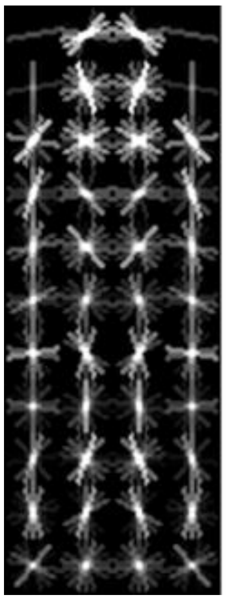


- Image is partitioned into 8x8 pixel blocks
- In each block we compute a histogram of gradient orientations
  - Invariant to changes in lighting, small deformations, etc.
- Compute features at different resolutions (pyramid)

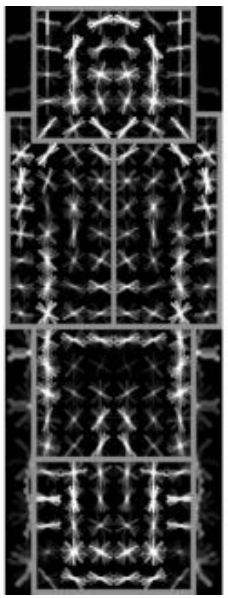
Felzenszwalb

Discriminative part-based models

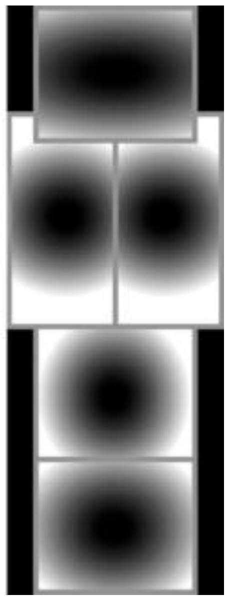
Root filter

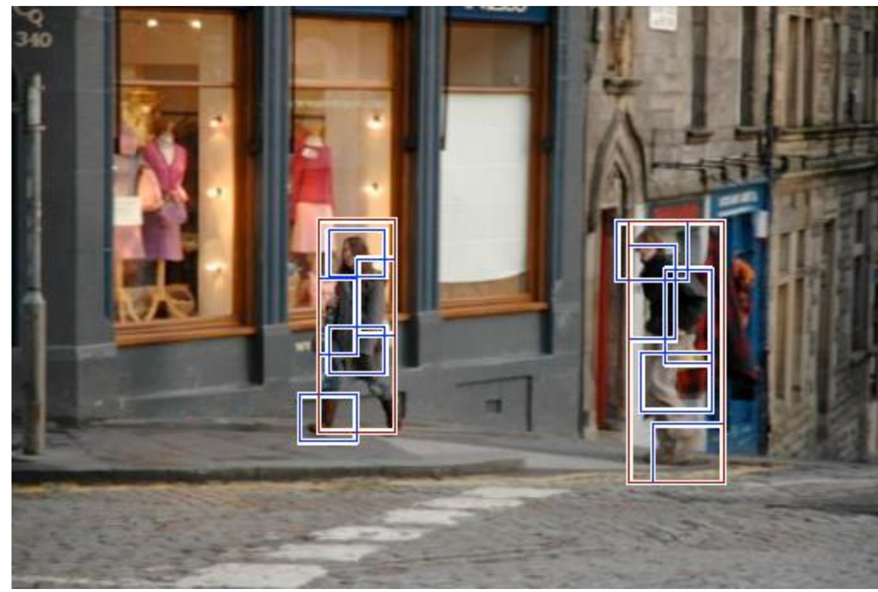


Part filters



Deformation weights



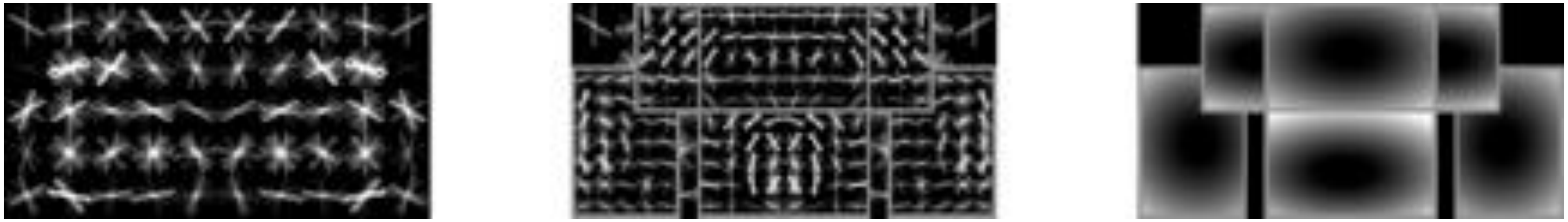


P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, [Object Detection with Discriminatively Trained Part Based Models](), PAMI 32(9), 2010

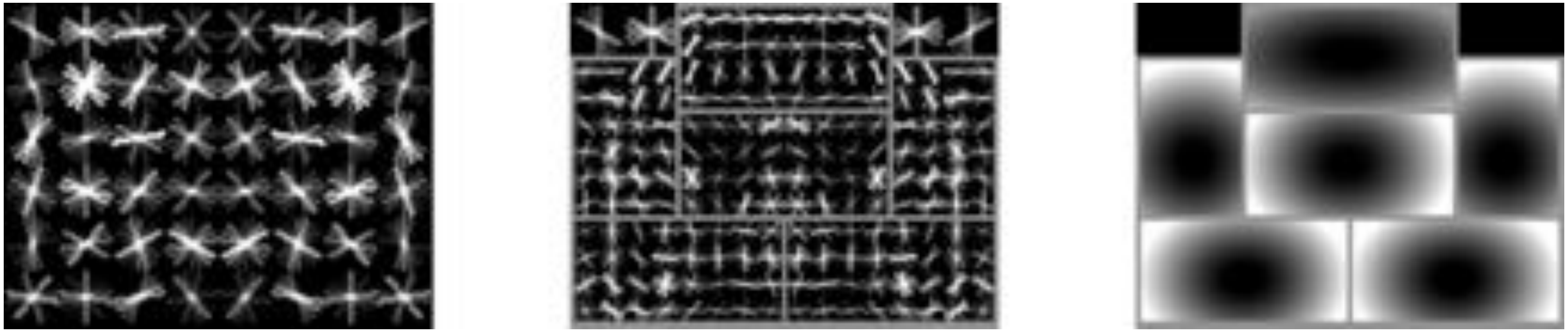
Felzenszwalb

Car model

Component 1

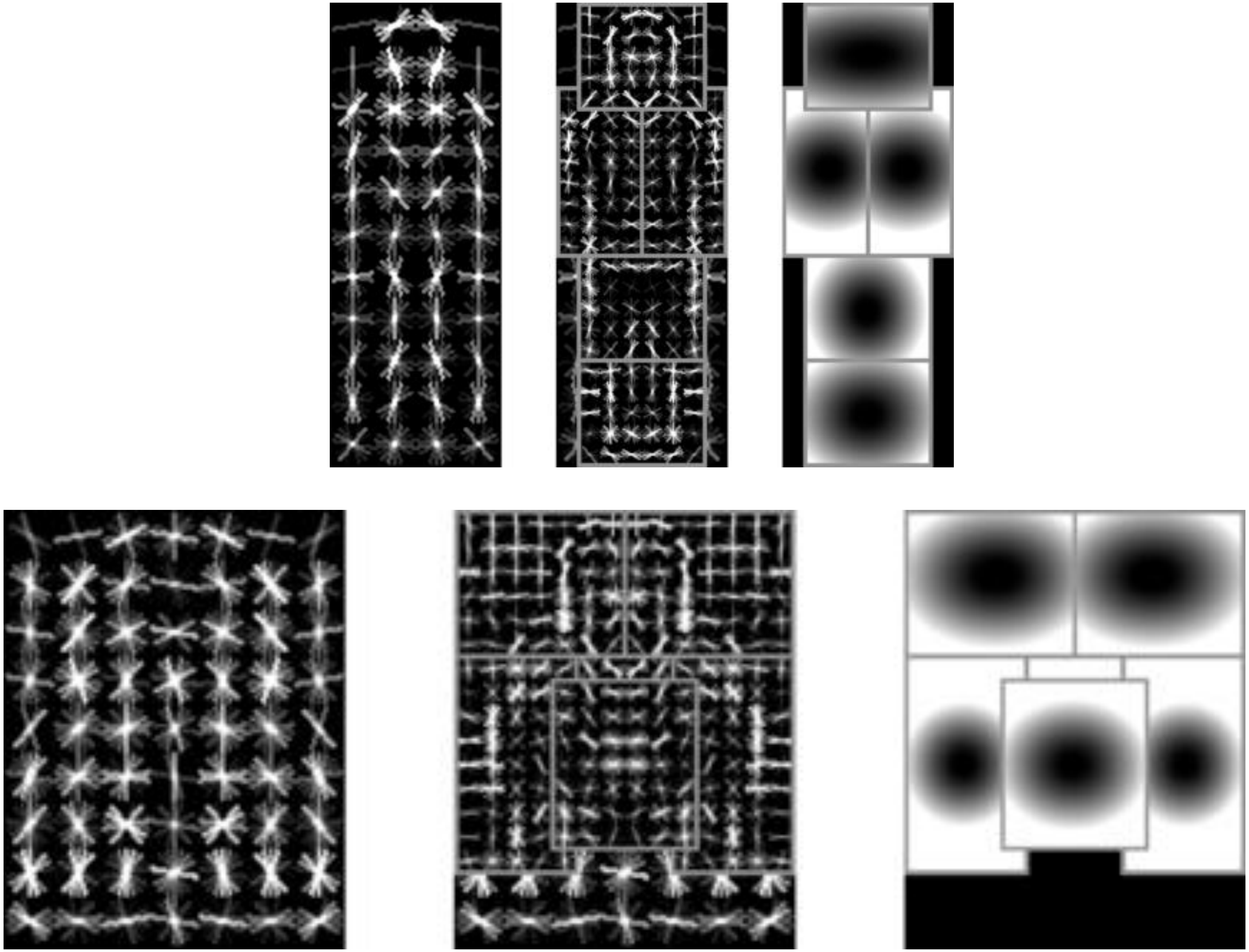


Component 2



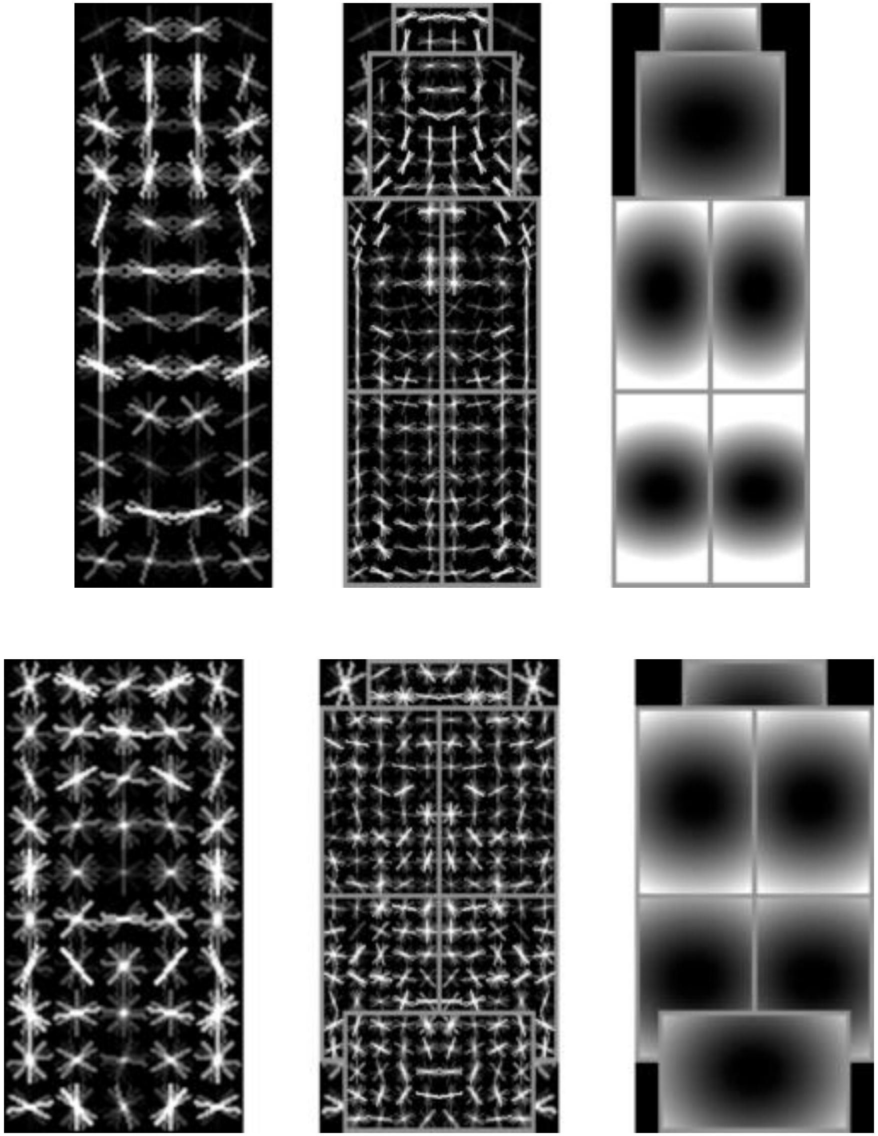
Felzenszwalb

Person model



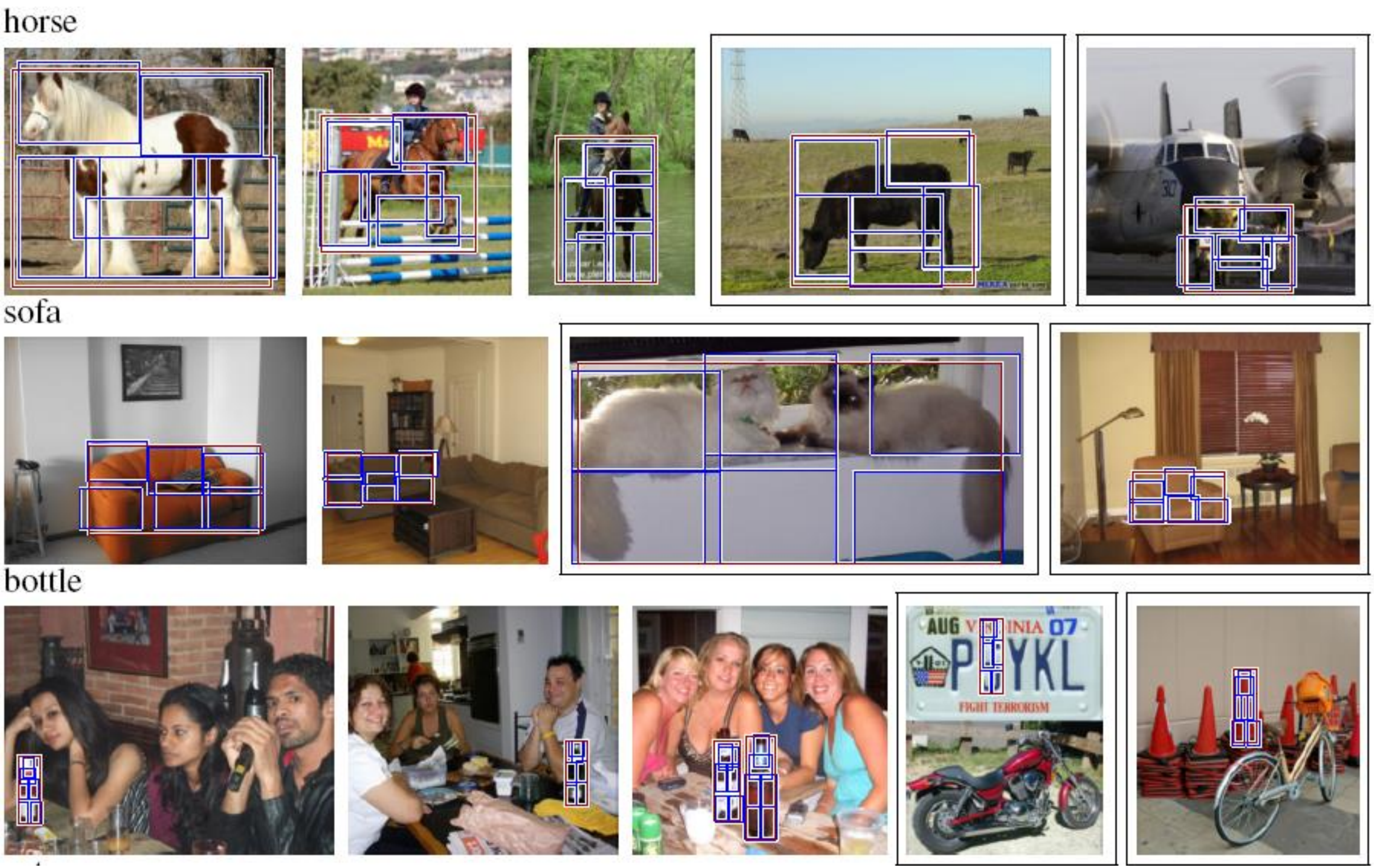
Felzenszwalb

Bottle model



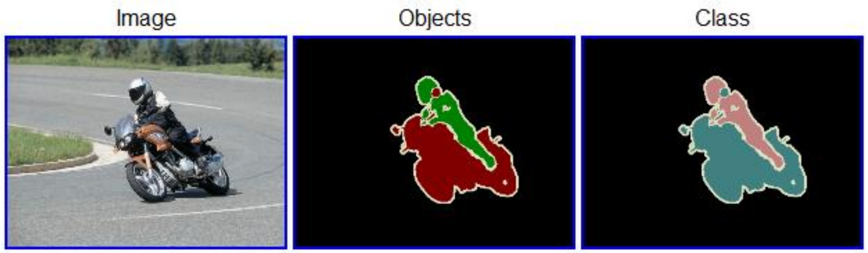
Felzenszwalb

Good detections?



The PASCAL Visual Object Classes Challenge 2009 (VOC2009)

- Twenty object categories (aeroplane to TV/monitor)
- Three challenges:
  - Classification challenge (is there an X in this image?)
  - Detection challenge (draw a box around every X)
  - Segmentation challenge



Snavely

Dataset: Collection

- Images downloaded from **flickr**
  - 500,000 images downloaded and random subset selected for annotation

Dataset: Annotation

- "Complete" annotation of all objects
- Annotated over web with <u>written guidelines</u>
  - High quality (?)

Dataset: Annotation

- "Complete" annotation of all objects
- Annotated over web with written guidelines
  - High quality (?)

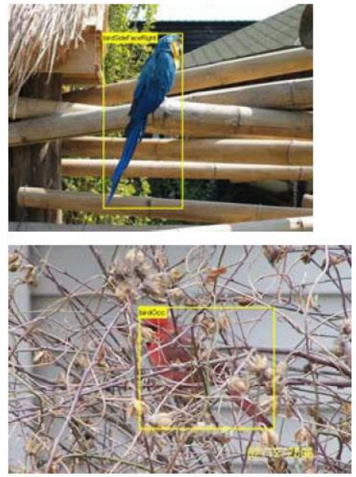
20 classes.
- Train / validation data has 11,530 images containing 27,450 ROI annotated objects and 6,929 segmentations.

Examples

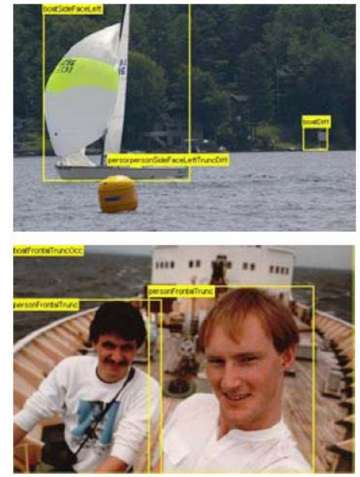
Aeroplane

Bicycle

Bird



Boat



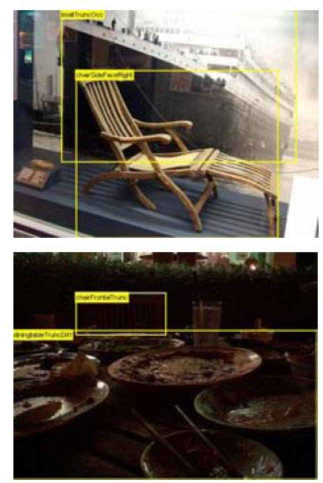
Bottle

Bus

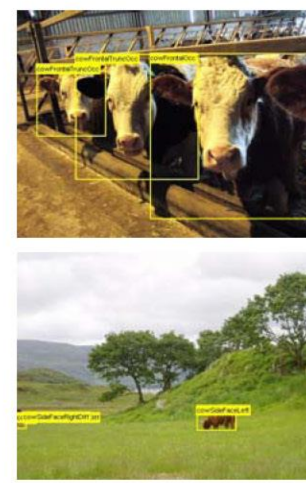
Car

Cat

Chair

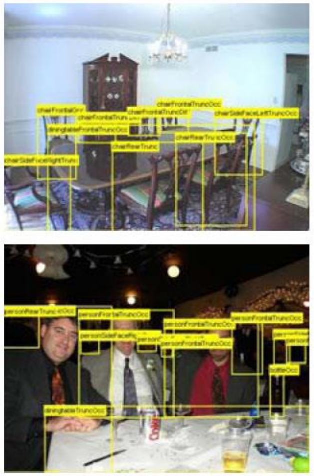


Cow



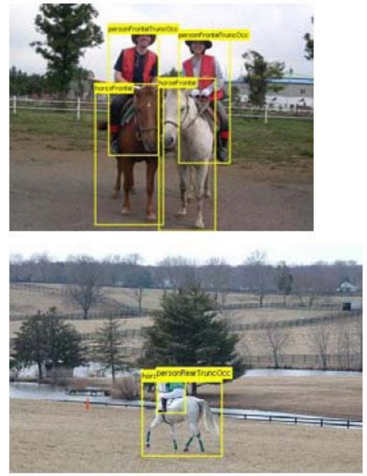
Examples

Dining Table



Dog

Horse

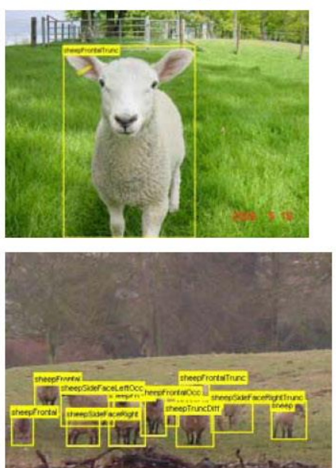


Motorbike

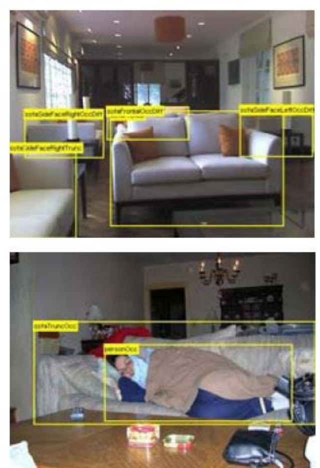
Person

Potted Plant

Sheep

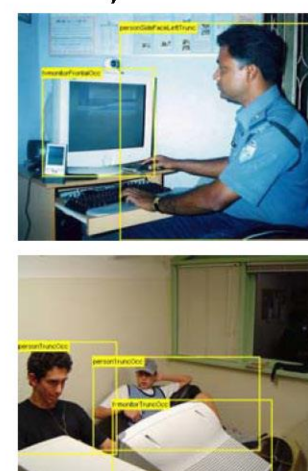


Sofa



Train

TV/Monitor



Classification Challenge

- Predict whether at least one object of a given class is present in an image

is there a cat?

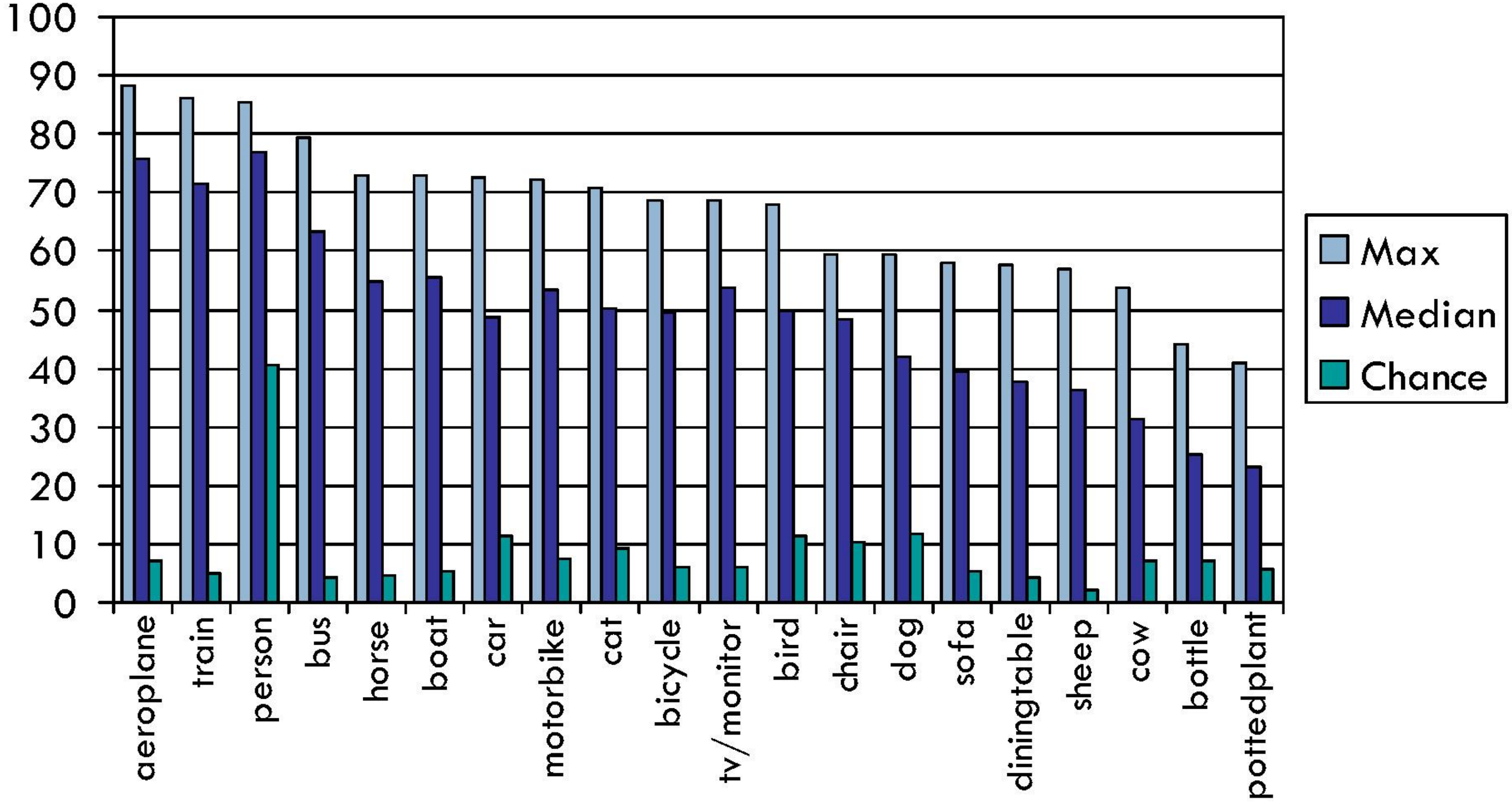
Results: AP by Method and Class

	aero plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	dining table	dog	horse	motor bike	person	potted plant	sheep	sofa	train	tv/ monitor
CVC_FLAT	85.3	57.8	66.0	66.1	36.2	70.6	60.6	63.5	55.1	44.6	53.4	49.1	64.4	66.8	84.8	37.4	44.1	47.9	81.9	67.5
CVC_FLAT-HOG-ESS	86.3	60.7	66.4	65.3	41.0	71.7	64.7	63.9	55.5	40.1	51.3	45.9	65.2	68.9	85.0	40.8	49.0	49.1	81.8	68.6
CVC_PLUS	86.6	58.4	66.7	67.3	34.8	70.4	60.0	64.2	52.5	43.0	50.8	46.5	64.1	66.8	84.4	37.5	45.1	45.4	82.1	67.0
FIRSTNIKON_AVGSRKDA	83.3	59.3	62.7	65.3	30.2	71.6	58.2	62.2	54.3	40.7	49.2	50.0	66.6	62.9	83.3	34.2	48.2	46.1	83.4	65.5
FIRSTNIKON_AVGSVM	83.8	58.2	62.6	65.2	32.0	69.8	57.7	61.1	54.5	44.0	50.3	49.6	64.6	61.7	83.2	33.4	46.5	48.0	81.6	65.3
FIRSTNIKON_BOOSTSRKDA	83.0	59.2	61.4	64.6	33.2	71.1	57.5	61.0	54.8	40.7	48.3	50.0	65.5	63.4	82.8	32.8	47.0	47.1	83.3	64.6
FIRSTNIKON_BOOSTSVMS	83.5	56.8	61.8	65.5	33.2	69.7	57.3	60.5	54.6	43.1	48.3	50.3	64.3	62.4	82.3	32.9	46.9	48.4	82.0	64.2
LEAR_CHI-SVM-MULT-LOC	79.5	55.5	54.5	63.9	43.7	70.3	66.4	56.5	54.4	38.8	44.1	46.2	58.5	64.2	82.2	39.1	41.3	39.8	73.6	66.2
NECUIUC_CDCV	88.1	68.0	68.0	72.5	41.0	78.9	70.4	70.4	58.1	53.4	55.7	59.3	73.1	71.3	84.5	32.3	53.3	56.7	86.0	66.8
NECUIUC_CLS-DTCT	88.0	68.6	67.9	72.9	44.2	79.5	72.5	70.8	59.5	53.6	57.5	59.0	72.6	72.3	85.3	36.6	56.9	57.9	85.9	68.0
NECUIUC_LL-CDCV	87.1	67.4	65.8	72.3	40.9	78.3	69.7	69.7	58.5	50.1	55.1	56.3	71.8	70.8	84.1	31.4	51.5	55.1	84.7	65.2
NECUIUC_LN-CDCV	87.7	67.8	68.1	71.1	39.1	78.5	70.6	70.7	57.4	51.7	53.3	59.2	71.6	70.6	84.0	30.9	51.7	55.9	85.9	66.7
UVASURREY_BASELINE	84.1	59.2	62.7	65.4	35.7	70.6	59.8	61.3	56.7	45.3	52.4	50.6	66.1	66.6	83.7	34.8	47.2	47.7	80.8	65.9
UVASURREY_MKFDA+BOW	84.7	63.9	66.1	67.3	37.9	74.1	63.2	64.0	57.1	46.2	54.7	53.5	68.1	70.6	85.2	38.5	47.2	49.3	83.2	68.1
UVASURREY_TUNECOLORKERNELSEL	85.0	62.8	65.1	66.5	37.6	73.5	62.1	62.0	57.4	45.1	54.5	52.5	67.7	69.8	84.8	39.1	46.8	49.9	82.9	68.1
UVASURREY_TUNECOLORSPECKDA	84.6	62.4	65.6	67.2	39.4	74.0	63.4	62.8	56.7	43.8	54.7	52.7	67.3	70.6	85.0	38.8	46.9	50.0	82.2	66.2

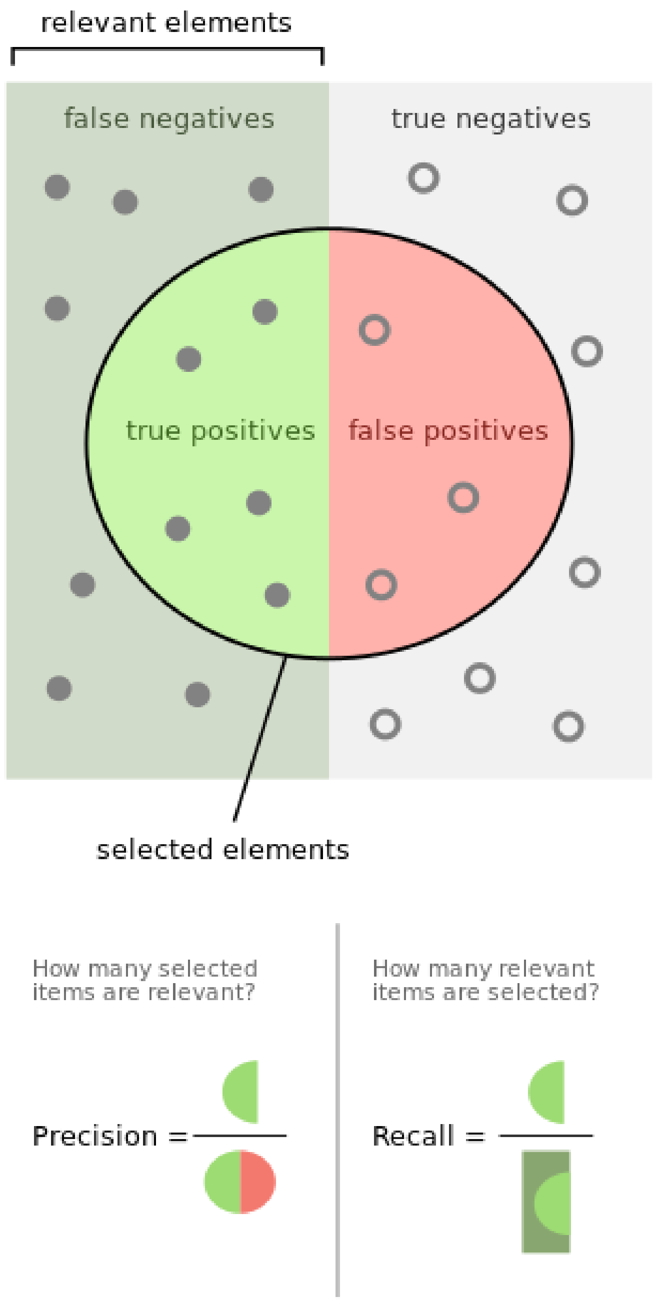
- Only methods in 1st, 2nd or 3rd place by group shown
- Groups: CVC, FIRST/Nikon, NEC/UIUC, UVA/Surrey

AP by Class

AP = average precision

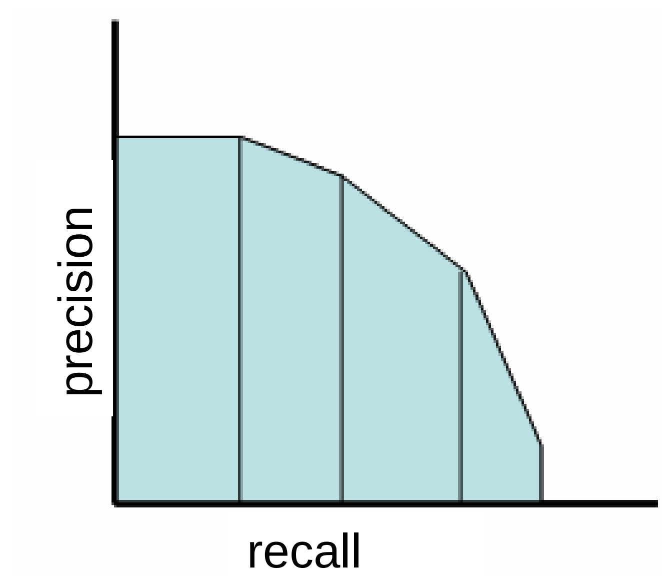


- Max AP: 88.1% (aeroplane) ... 40.8% (potted plant)



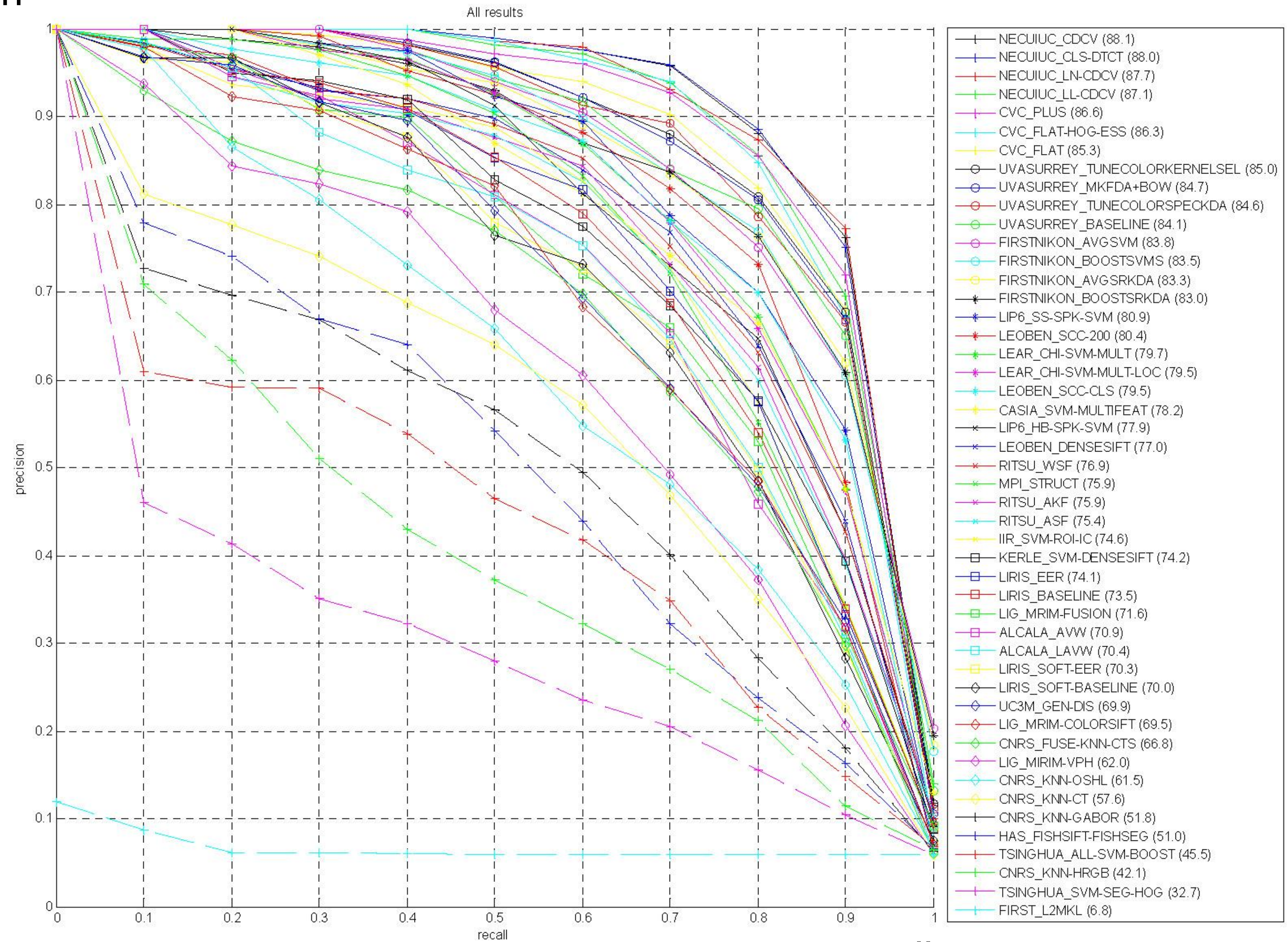
Set threshold on ‘detection’ to create one pair of precision / recall values.

Vary threshold across all values to generate precision / recall curves:



Precision/Recall: Aeroplane (All)

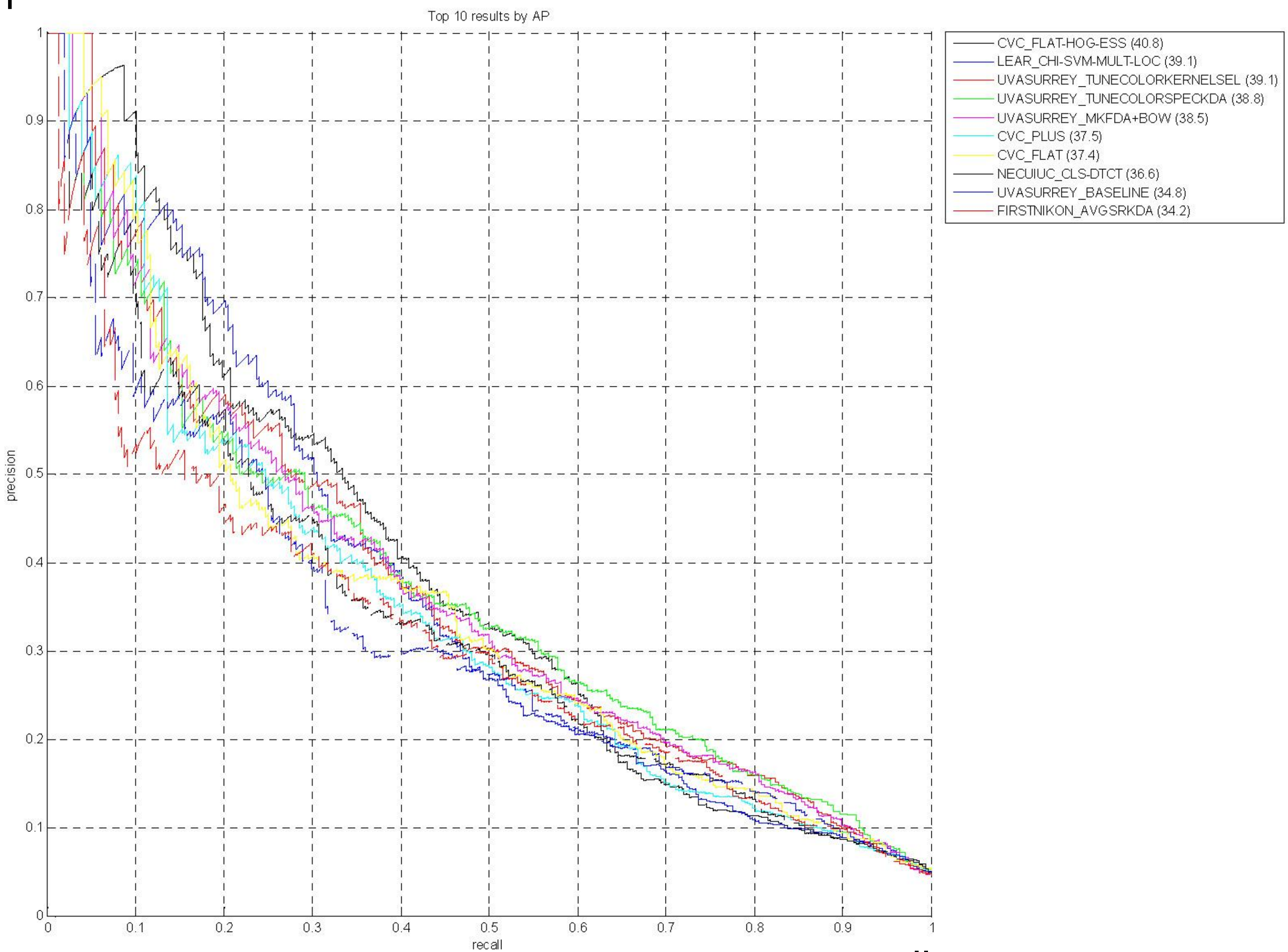
precision



recall

Precision/Recall: Potted plant (Top 10 by AP)

precision



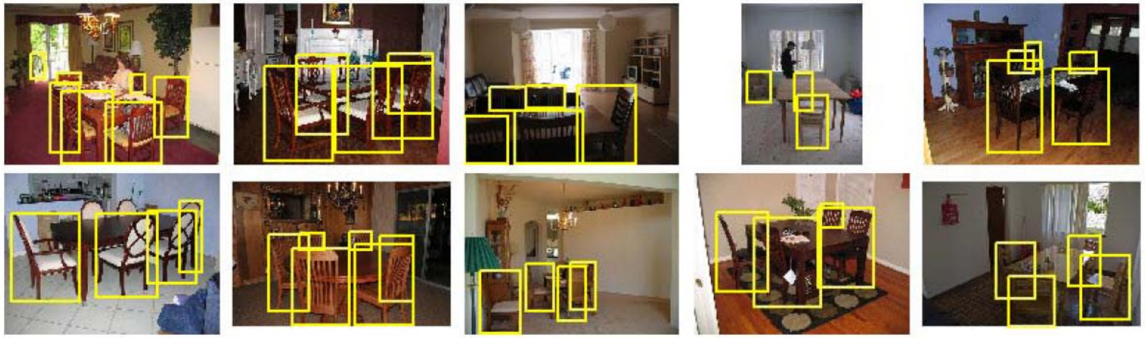
recall

Ranked Images: Aeroplane

- Class images: Highest ranked

Ranked Images: Chair

- Class images: Highest ranked

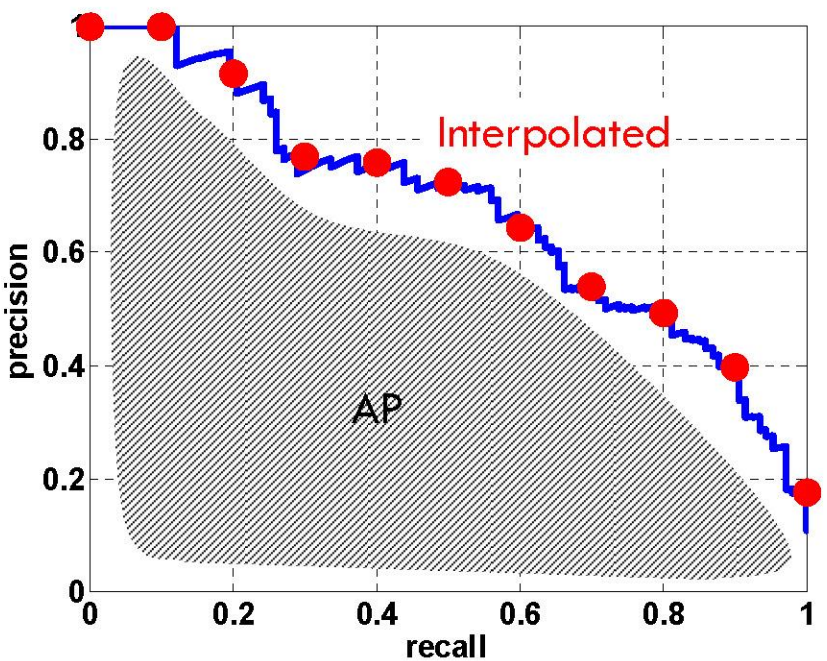


Detection Challenge

- Predict the bounding boxes of all objects of a given class in an image (if any)

Evaluation

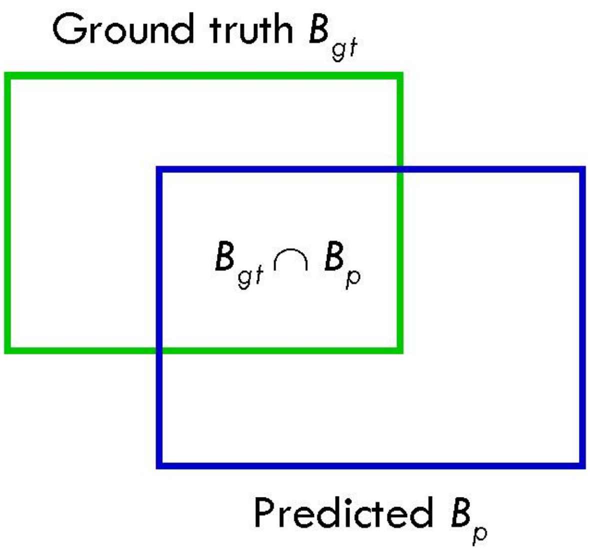
- Average Precision [TREC] averages precision over the entire range of recall
  - Curve interpolated to reduce influence of "outliers"



- A good score requires both high recall and high precision
- Application-independent
- Penalizes methods giving high precision but low recall

Evaluating Bounding Boxes

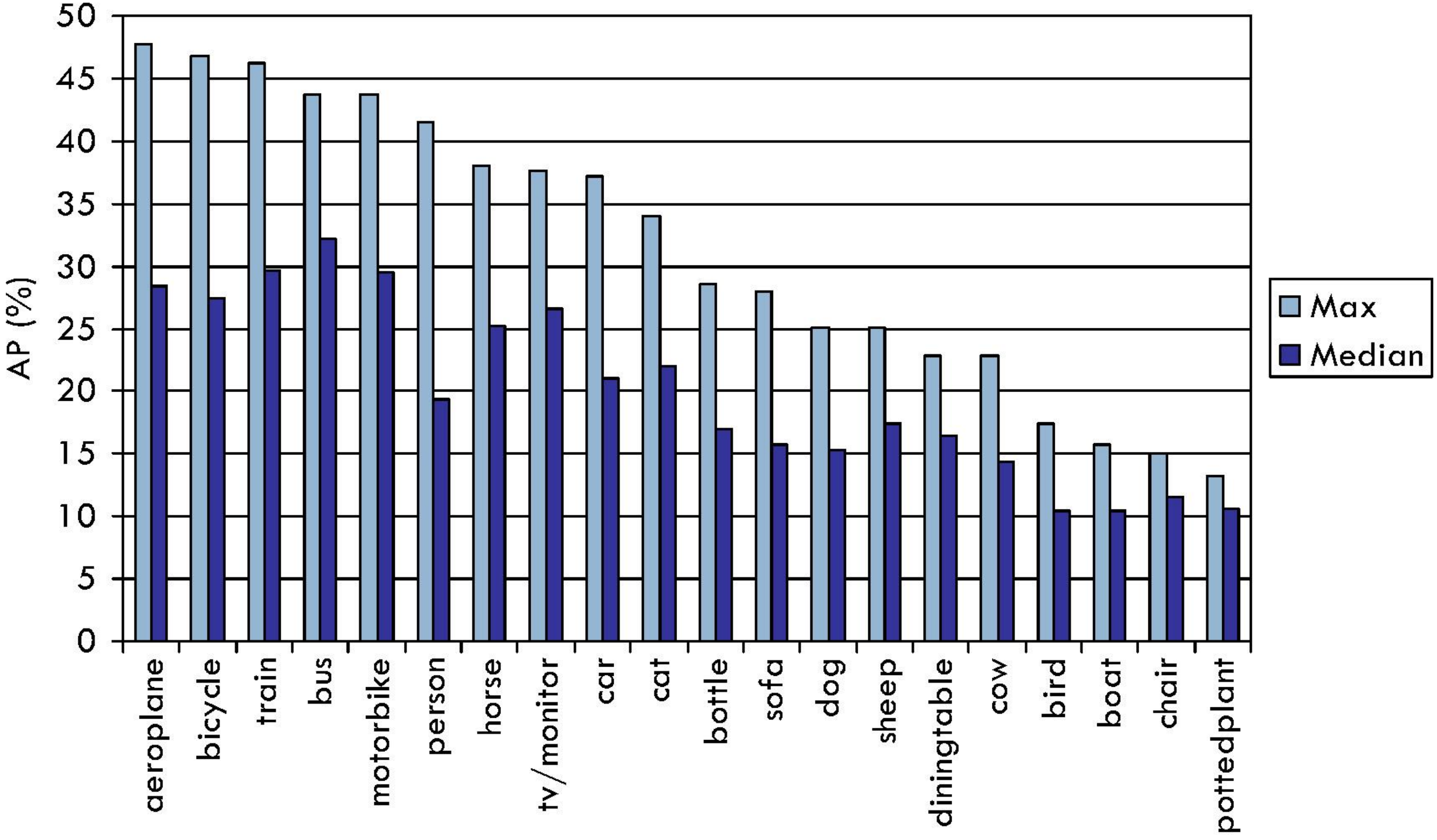
- Area of Overlap (AO) Measure



$$AO(B_{gt}, B_p) = \frac{|B_{gt} \bigcap B_p|}{|B_{gt} \bigcup B_p|}$$

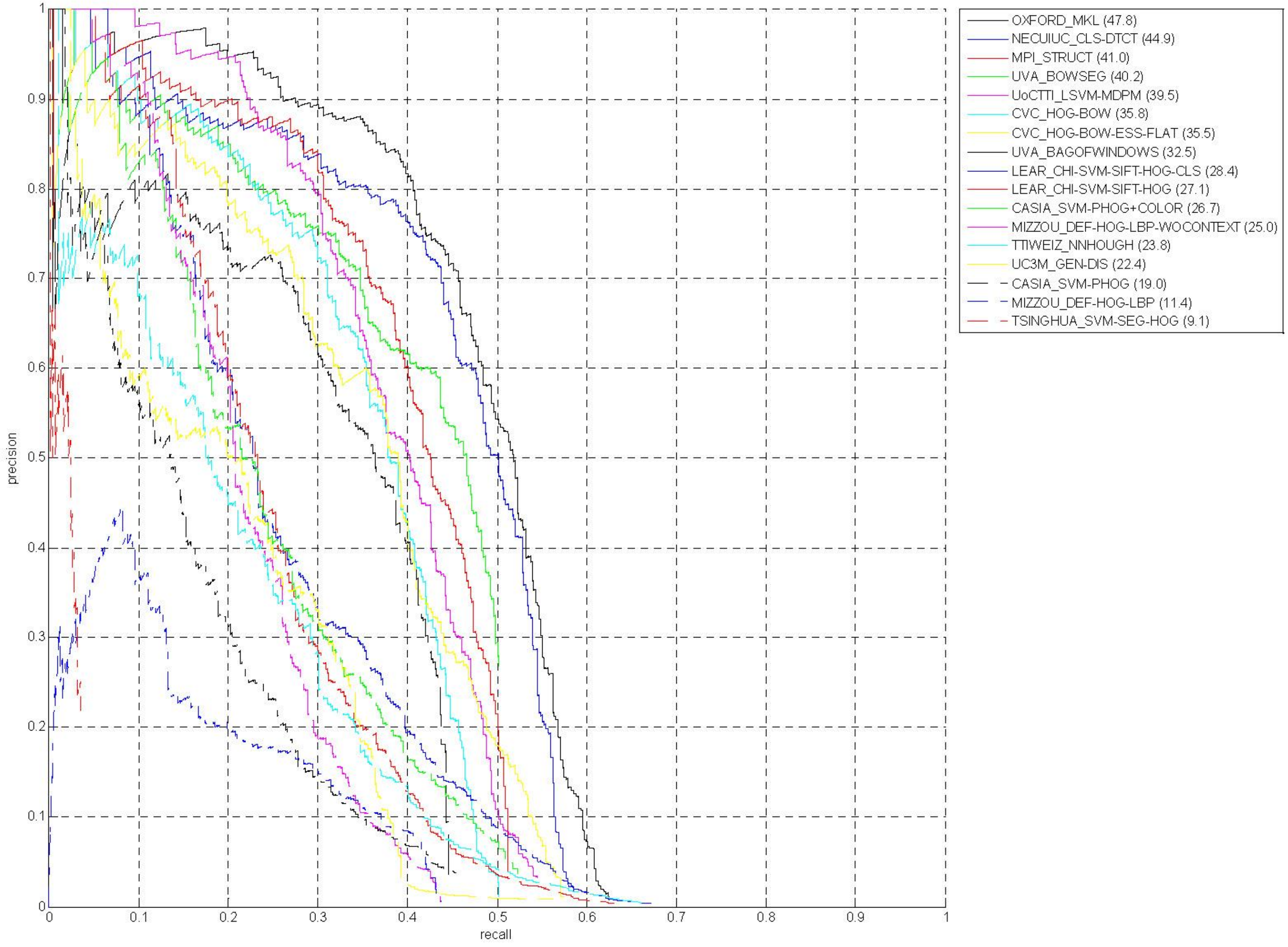
- Need to define a threshold t such that $AO(B_{gt}, B_p)$ implies a correct detection: 50%

AP by Class

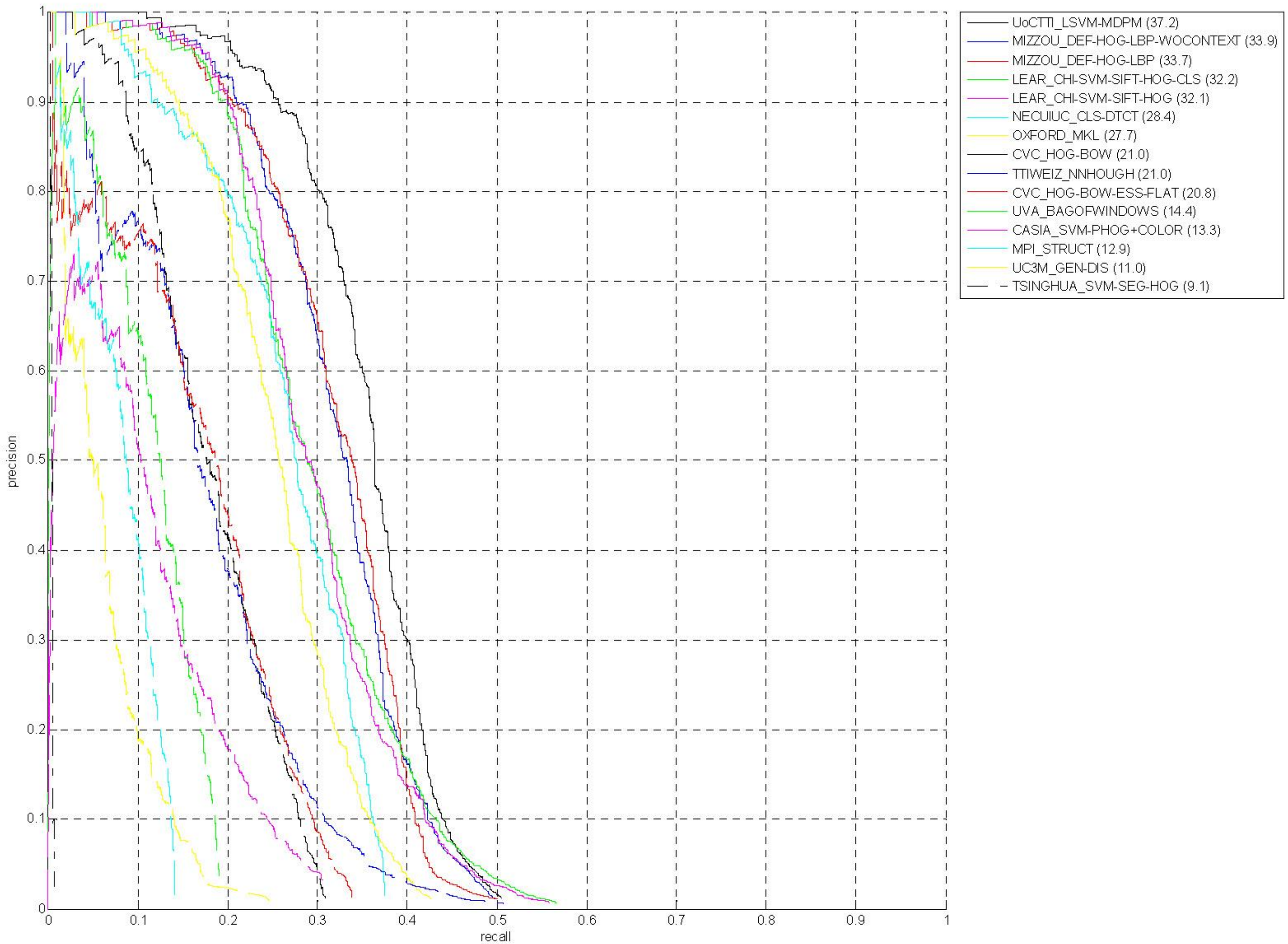


Chance essentially 0

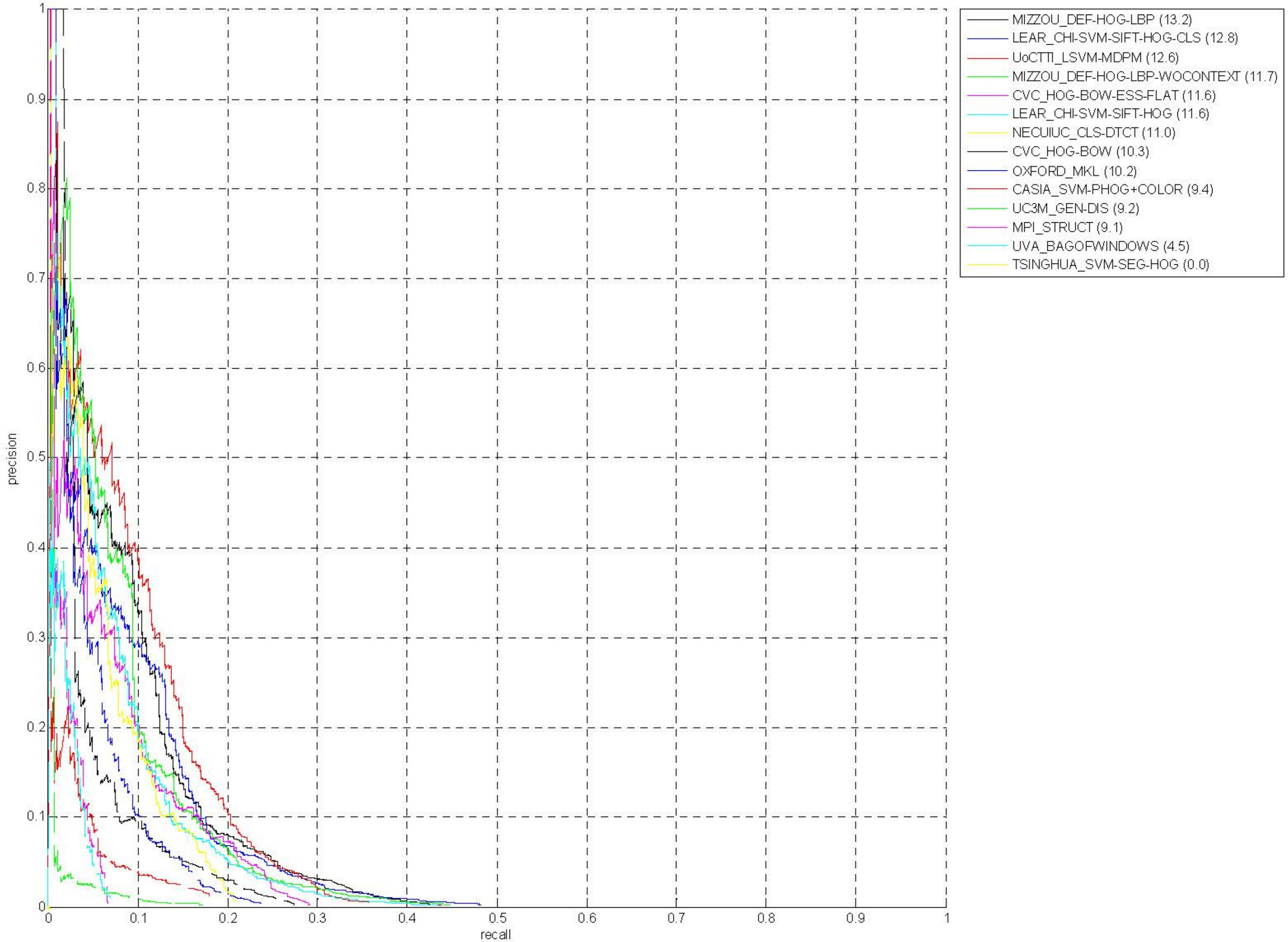
Precision/Recall - Aeroplane



Precision/Recall - Car



Precision/Recall – Potted plant



True Positives - Person

UoCTTI_LSVM-MDPM

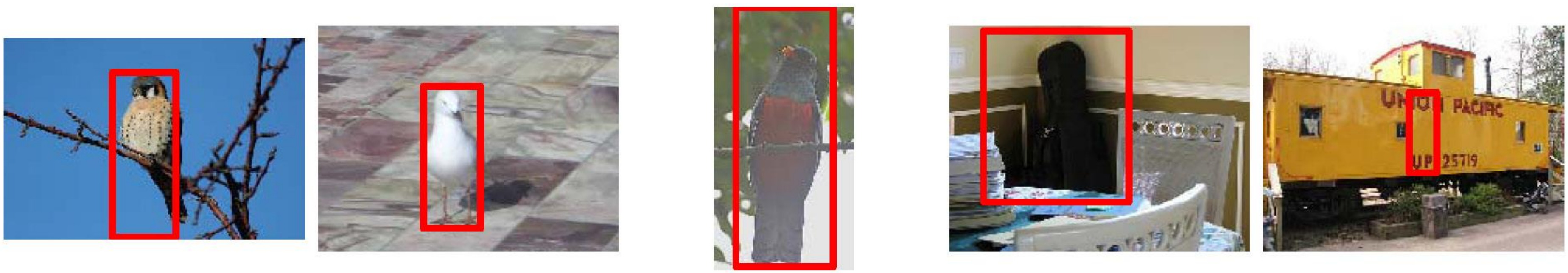
MIZZOU_DEF-HOG-LBP

NECUIUC_CLS-DTCT

False Positives - Person

UoCTTI_LSVM-MDPM

MIZZOU_DEF-HOG-LBP



NECUIUC_CLS-DTCT

“Near Misses” - Person

UoCTTI_LSVM-MDPM

MIZZOU_DEF-HOG-LBP

NECUIUC_CLS-DTCT

True Positives - Bicycle

UoCTTI_LSVM-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT

False Positives - Bicycle

UoCTTI_LSVM-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT

HOGgles (Vondrick et al. ICCV 2013)

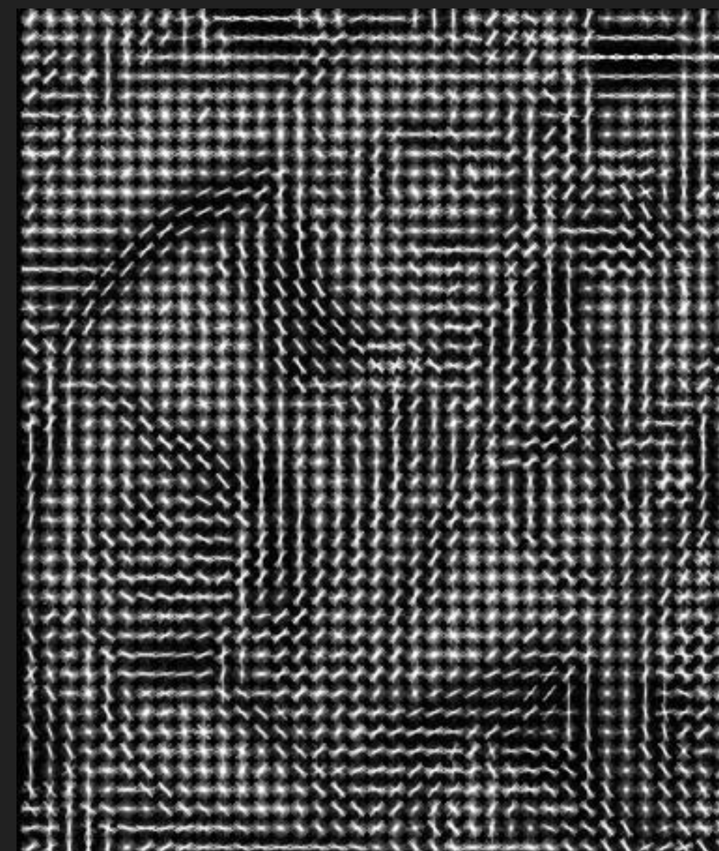
Vondrick et al.

Vondrick et al.

Vondrick et al.

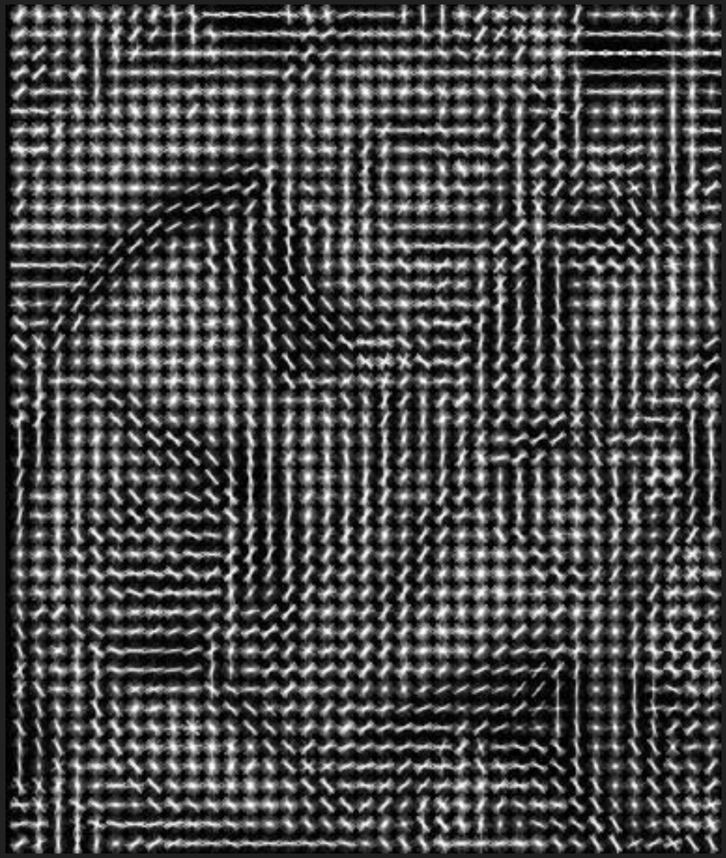
Vondrick et al.

What information is lost?



Vondrick et al.

What information is lost?



Vondrick et al.

How can we ‘invert’ lossy HOG?

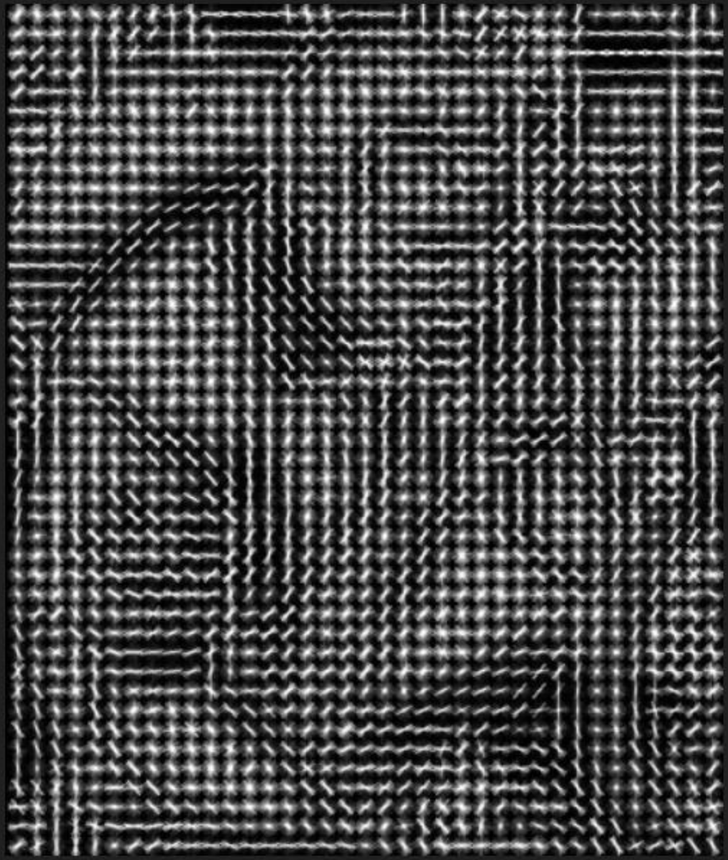
- Gradient computation
  - Without width or ‘edge blur’, i.e., not edges from Eldar 1999
- Oriented magnitude sum (via bins)
  - Loss of precision
  - Loss of specificity – any number of values can sum to the same total
- Normalization
  - No way to unnormalize without knowing normalization factors

Many different image patches translate to the same HOG feature : (

What information is lost?

$x = input\ patch$
$y = HOG\ descriptor$
$\phi(x) = HOG\ transform$

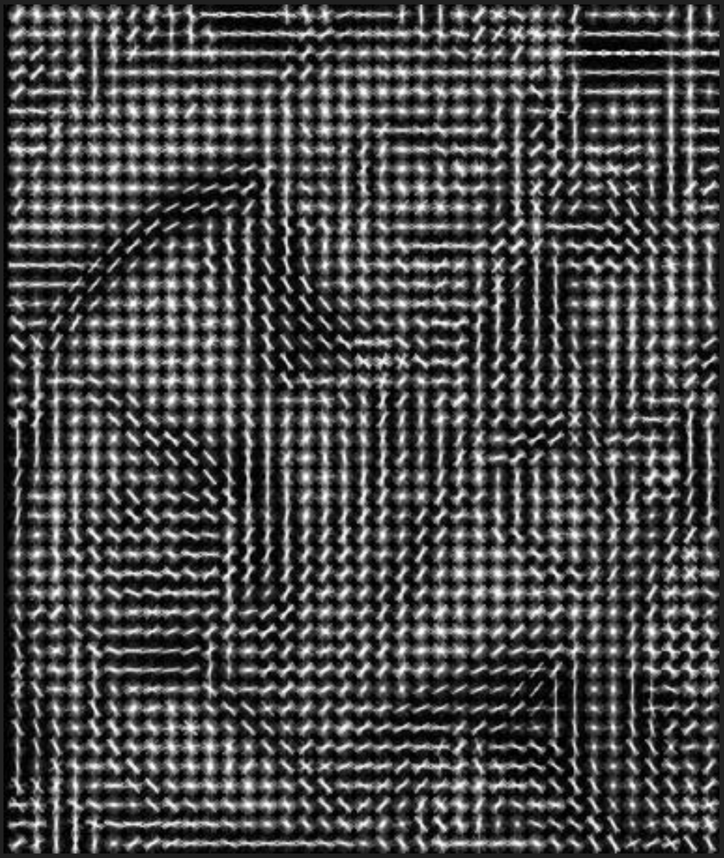
$$\min_{x \in \mathbb{R}^d} ||\phi(x) - y||_2^2$$



Hard to optimize!
Many-to-one = unconstrained!

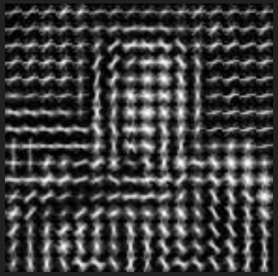
Vondrick et al.

What information is lost?



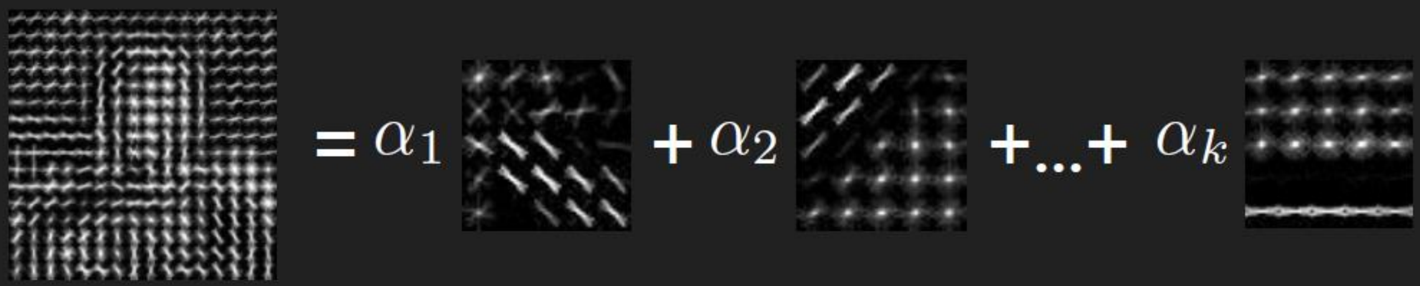
Vondrick et al.

Method: Paired Dictionary



Vondrick et al.

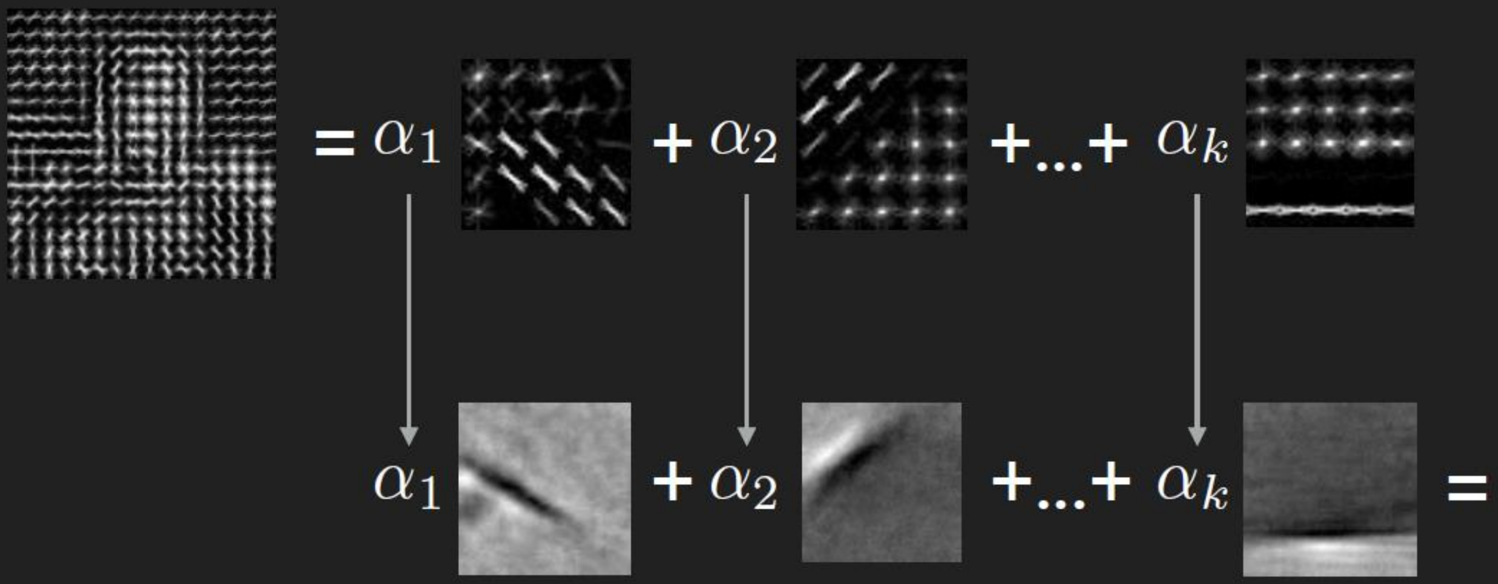
Method: Paired Dictionary



How to constrain (two parts):

1. Learn a basis over HOG windows

Method: Paired Dictionary

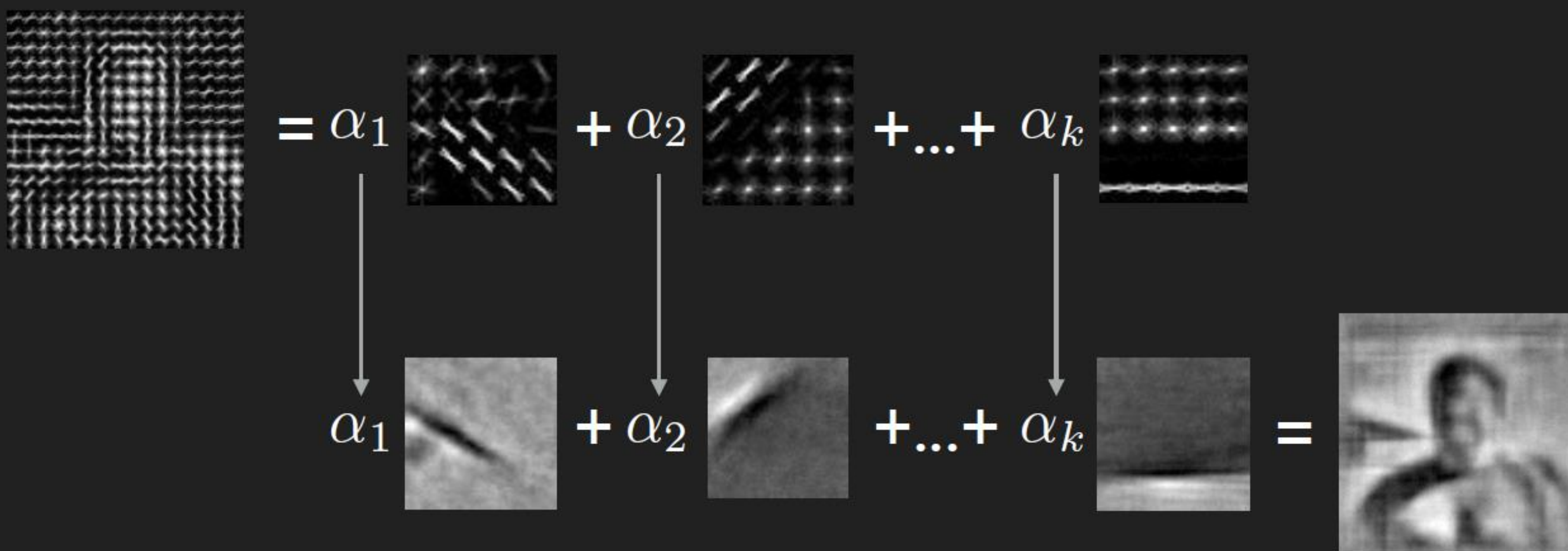


How to constrain (two parts):

1. Learn a basis over HOG windows
2. Simultaneously learn a basis over input windows, and *share the weights* $\alpha_1 \ldots \alpha_k$ *over the training data*

Vondrick et al.

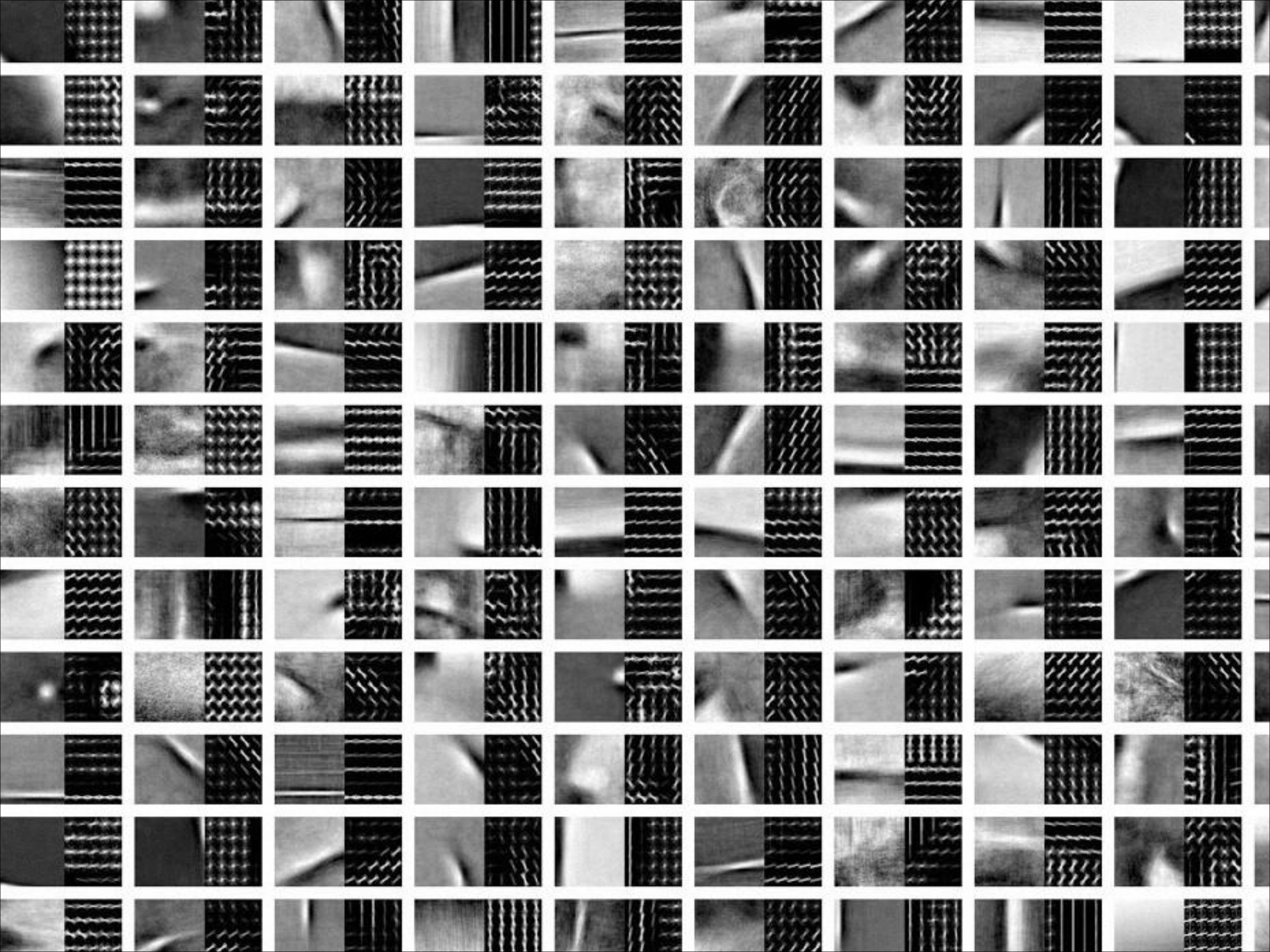
Method: Paired Dictionary

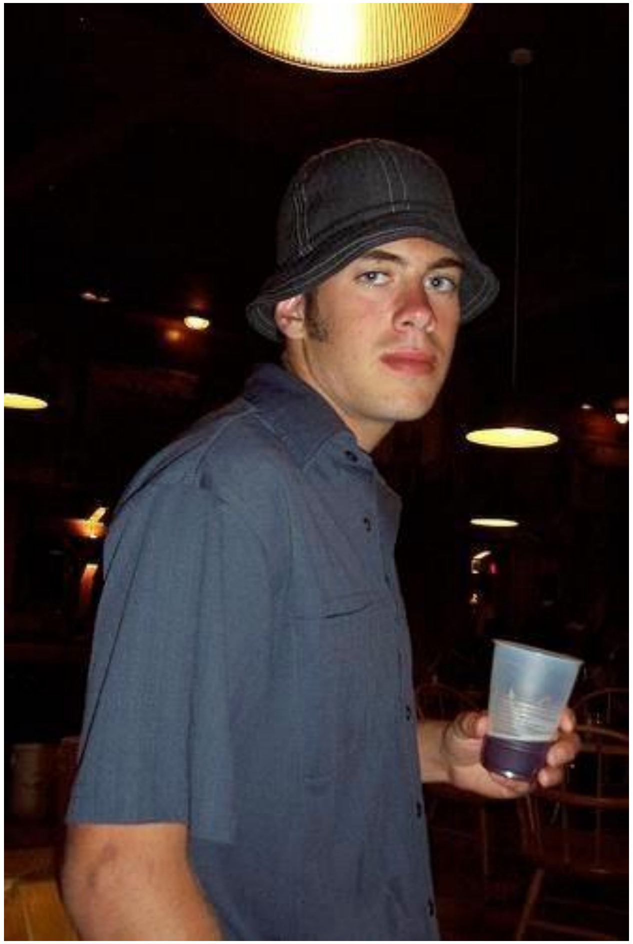


Inference to invert HOG:

1. Transform HOG patch into basis vectors
2. Take weights and apply to input basis

Vondrick et al.



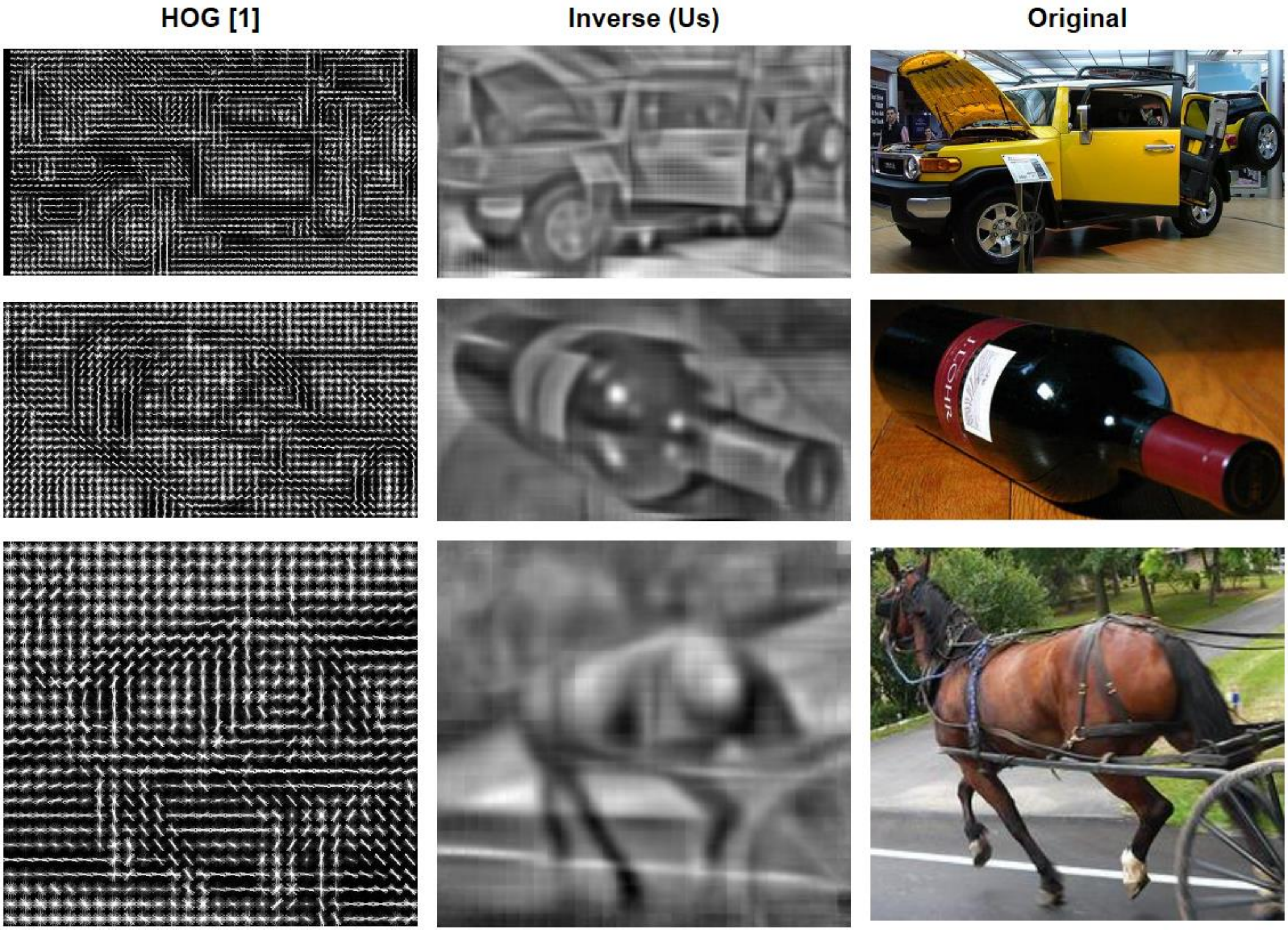


HumanVision

HOGVision

Vondrick et al.

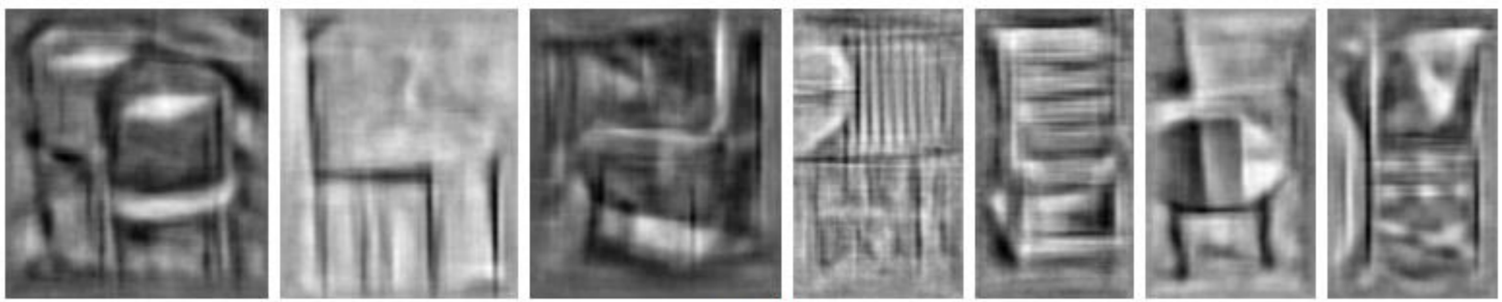
HOGgles (Vondrick et al. ICCV 2013)



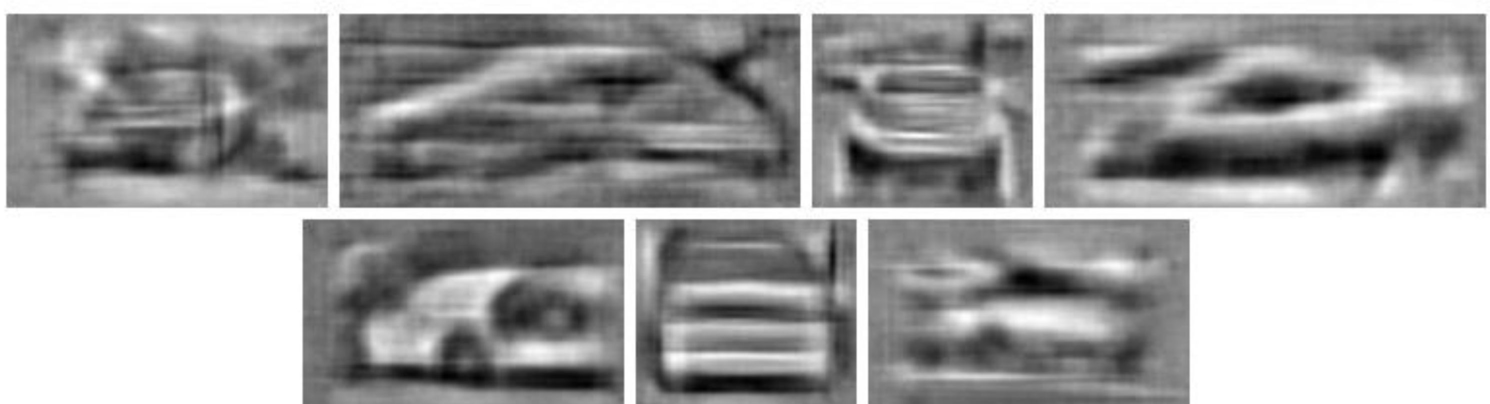
Visualizing Top Detections

We have visualized some high scoring detections from the deformable parts model. Can you guess which are false alarms? Click on the images below to reveal the corresponding RGB patch. You might be surprised!

Person



Chair



Car

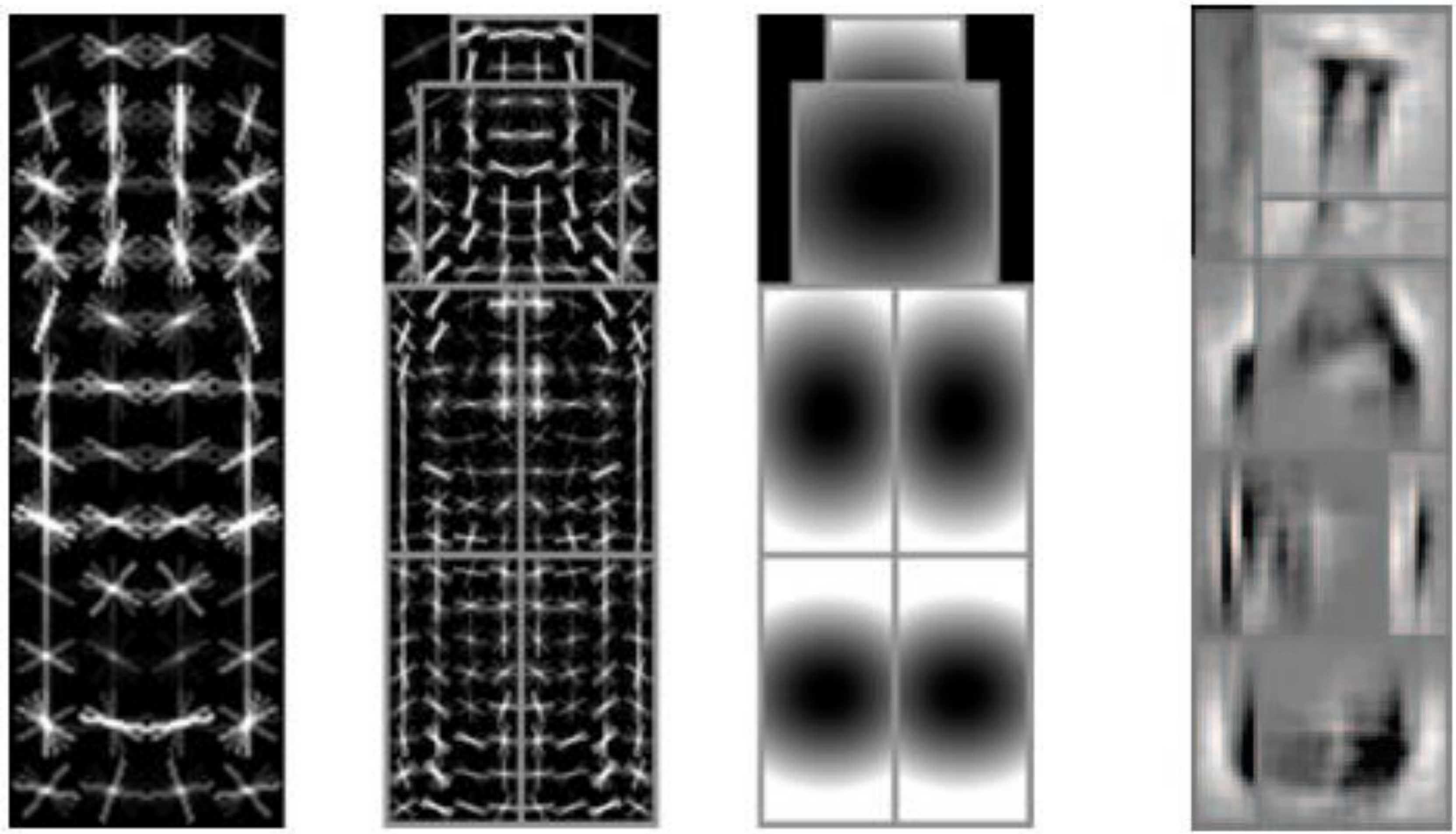
Vondrick et al.

Recursive HOG!

Original x $\quad$ $x' = \phi^{-1}\left(\phi(x)\right)$ $\quad$ $x'' = \phi^{-1}\left(\phi(x')\right)$

Figure 11: We recursively compute HOG and invert it with a paired dictionary. While there is some information loss, our visualizations still do a good job at accurately representing HOG features. $\phi(\cdot)$ is HOG, and $\phi^{-1}(\cdot)$ is the inverse.

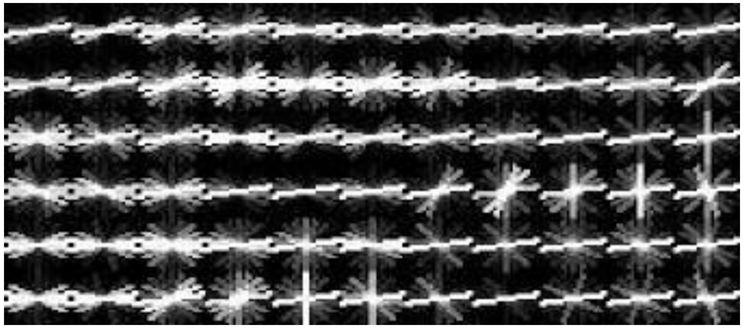
Bottle Deformable Parts Models + HOGgles



Why did the detector fail?

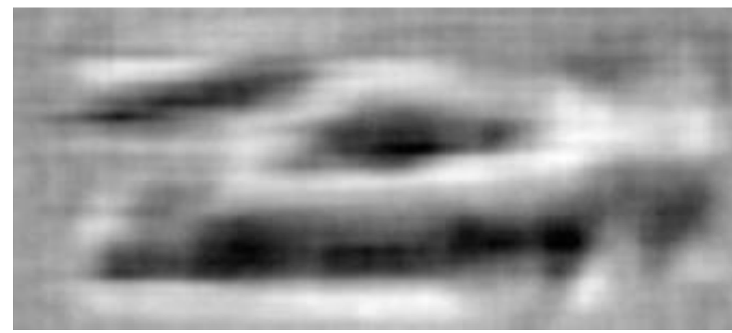
Vondrick et al.

Why did the detector fail?



Vondrick et al.

Why did the detector fail?



Vondrick et al.

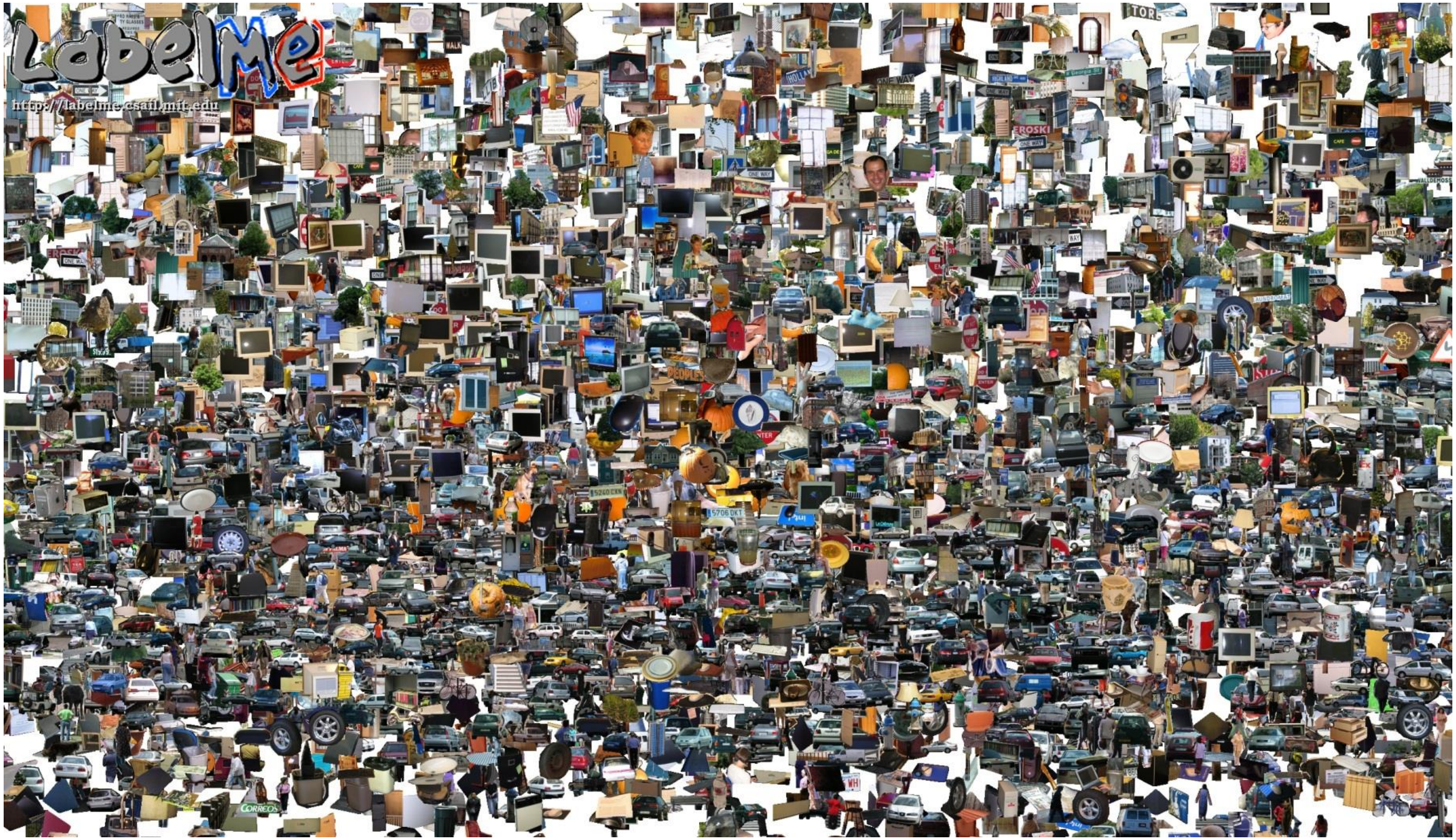
Code Available

Try it on your projects!

http://web.mit.edu/vondrick/ihog/

```
ihog = invertHOG(feat);
```

Opportunities of Scale



Many slides from James Hays, Alyosha Efros, and Derek Hoiem

Graphic from Antonio Torralba

PASCAL VOC: 2010-2014

LLC
2009: 66.5%

Context-SVM
2010: 73.8%

GHM
2011: 78.7%

25%

Sub-category
2012: 82.2%

Deep feature
2013: 79.0%

UNIVERSITY OF OXFORD

Deep feature
2014: 83.2%

HCP
2014: 91.4%

Shuicheng Yan

Computer Vision so far

- The geometry of image formation
  - Ancient / Renaissance
- Signal processing / Convolution
  - 1800, but really the 50’s and 60’s
- Hand-designed Features for recognition, either instance-level or categorical
  - 1999 (SIFT), 2003 (Video Google), 2005 (Dalal-Triggs), 2006 (spatial pyramid)
- Learning from Data
  - 1991 (EigenFaces) but late 90’s to now especially

What has changed in the last decade?

- The Internet
- Crowdsourcing
- Learning representations from the data these sources provide (deep learning)

Google and massive data-driven algorithms

A.I. for the postmodern world:

- all questions have already been answered…many times, in many ways
- Google is dumb, the "intelligence" is <u>in the data</u>

Big Idea

- Do we need computer vision systems to have strong AI-like reasoning about our world?
- What if invariance / generalization isn't actually the core difficulty of computer vision?
- What if we can perform high level reasoning with brute-force, data-driven algorithms?

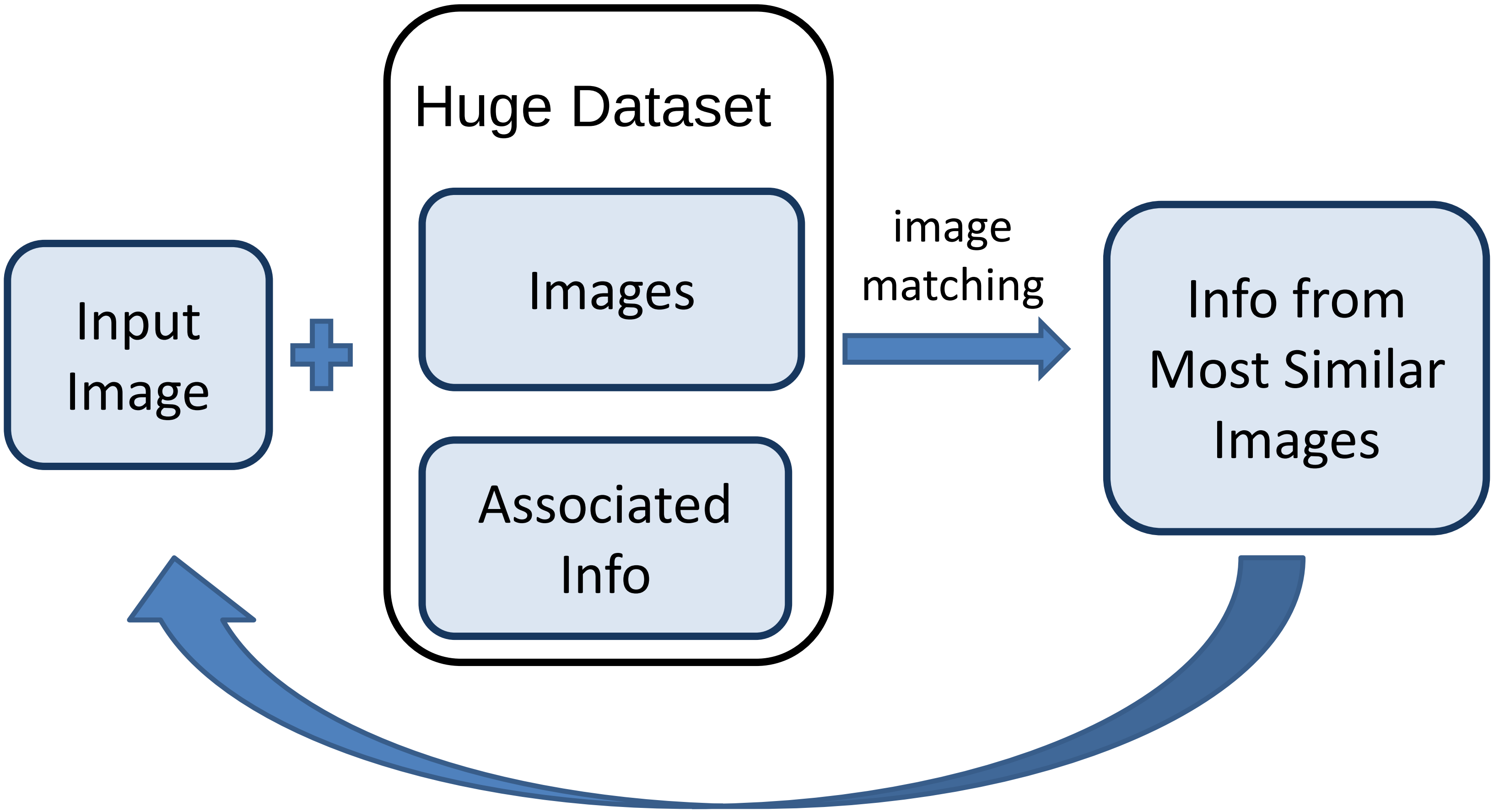
The Unreasonable Effectiveness of Data

Peter Norvig
Google

Peter Norvig

The Unreasonable
Effectiveness of Data

General Principal



Hopefully, If you have enough images, the dataset will contain very similar images that you can find with simple matching methods.

Powers of 10

Number of images on my hard drive: 10^{6}

Number of images seen during my first 10 years: 10^{8}

(3 images/second * 60 * 60 * 16 * 365 * 10 = 630,720,000)

Number of images seen by all humanity: 10^{20}

106,456,367,669 humans[1] * 60 years * 3 images/second * 60 * 60 * 16 * 365 =

1 from http://www.prb.org/Articles/2002/HowManyPeopleHaveEverLivedonEarth.aspx

Number of photons in the universe: 10^{88}

Number of all 32x32 images: 10^{7373}

$256^{32*32*3} \sim 10^{7373}$

Understanding scenes encompasses all kinds of knowledge

But not all scenes are so original



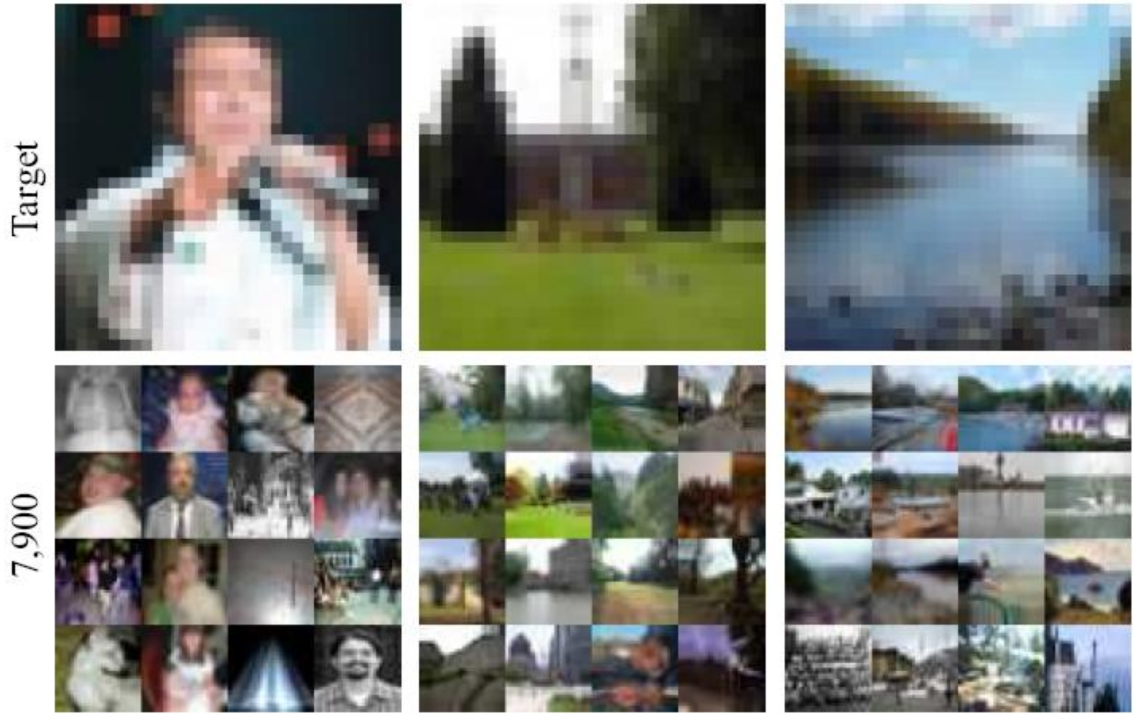





Lots

Of

Images

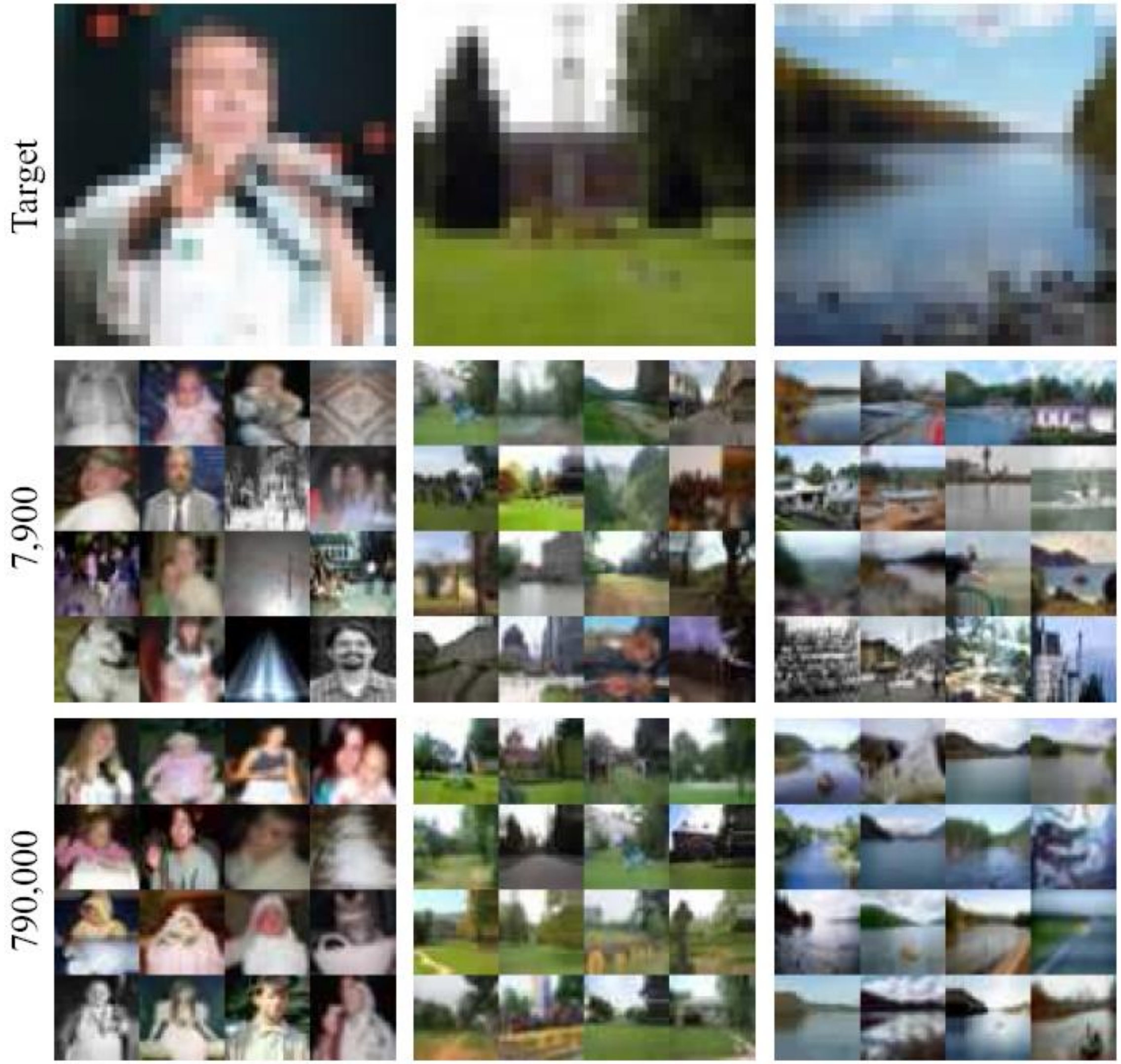


A. Torralba, R. Fergus, W.T.Freeman. PAMI 2008

Lots

Of

Images

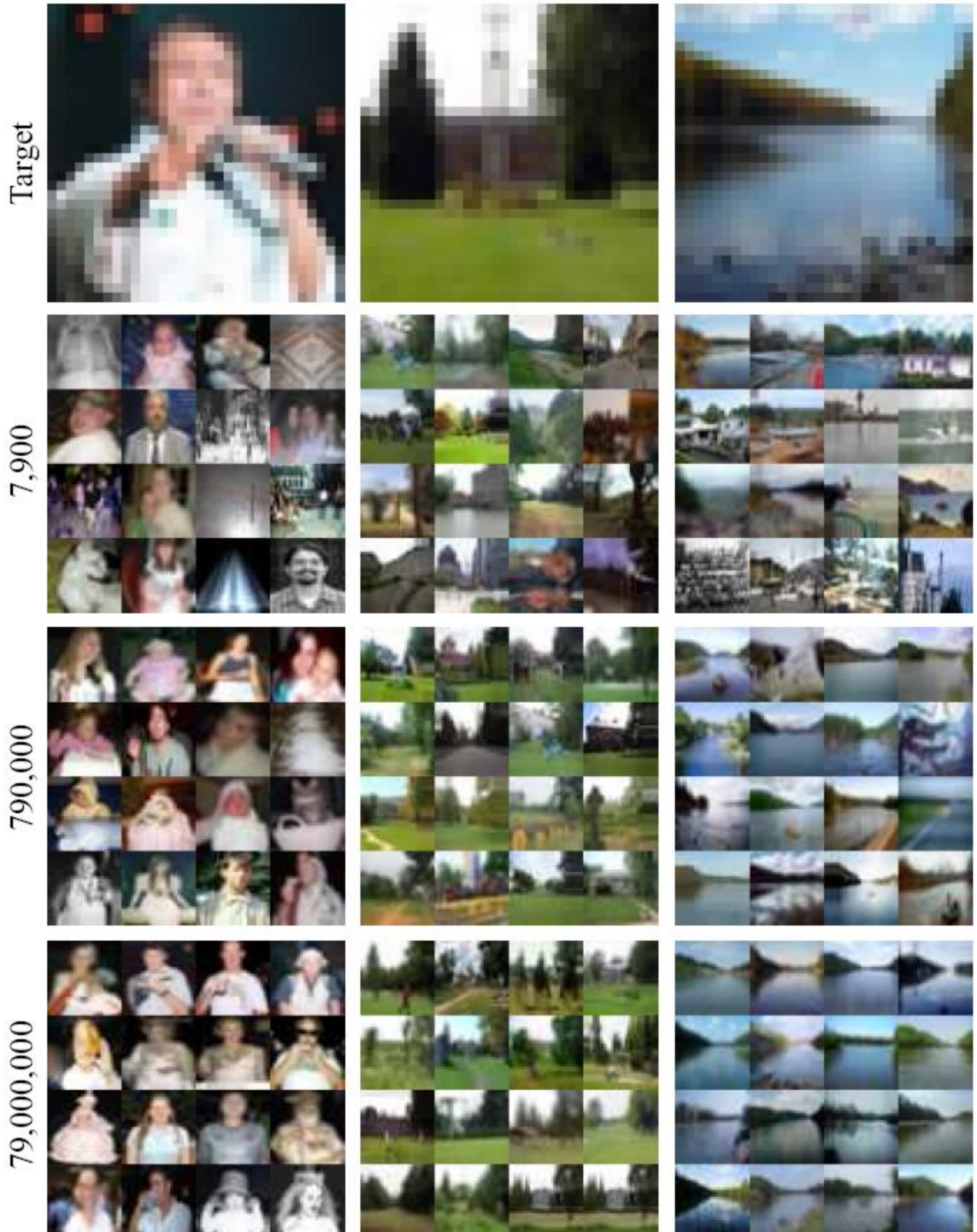


A. Torralba, R. Fergus, W.T.Freeman. PAMI 2008

Lots

Of

Images



Application: Automatic Colorization


Input


Color Transfer


Color Transfer


Matches (gray)


Matches (w/ color)


Avg Color of Match

Application: Automatic Colorization

Input

Color Transfer

Color Transfer

Matches (gray)

Matches (w/ color)

Avg Color of Match

How much can an image tell about its geographic location?

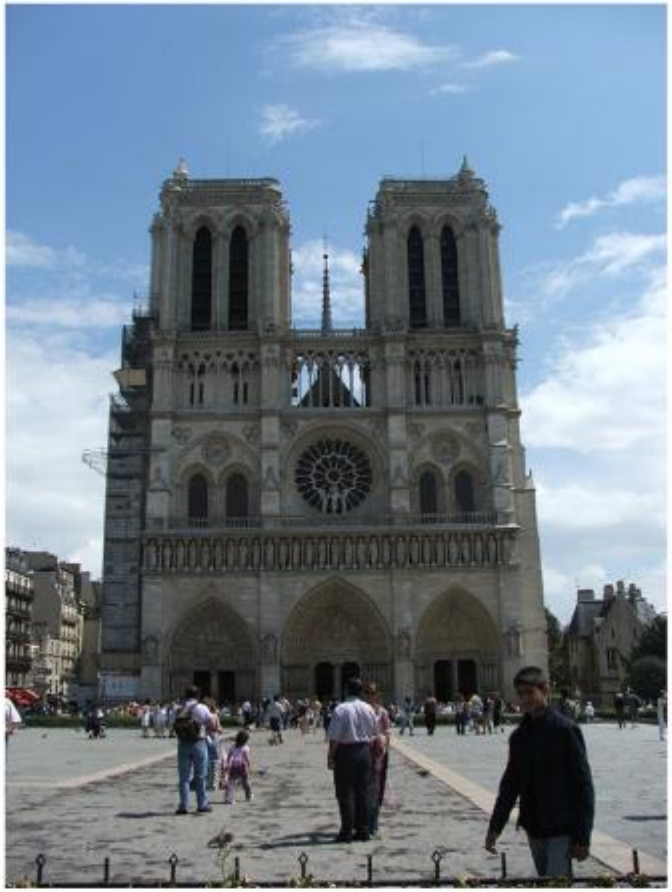
How much can an image tell about its geographic location?

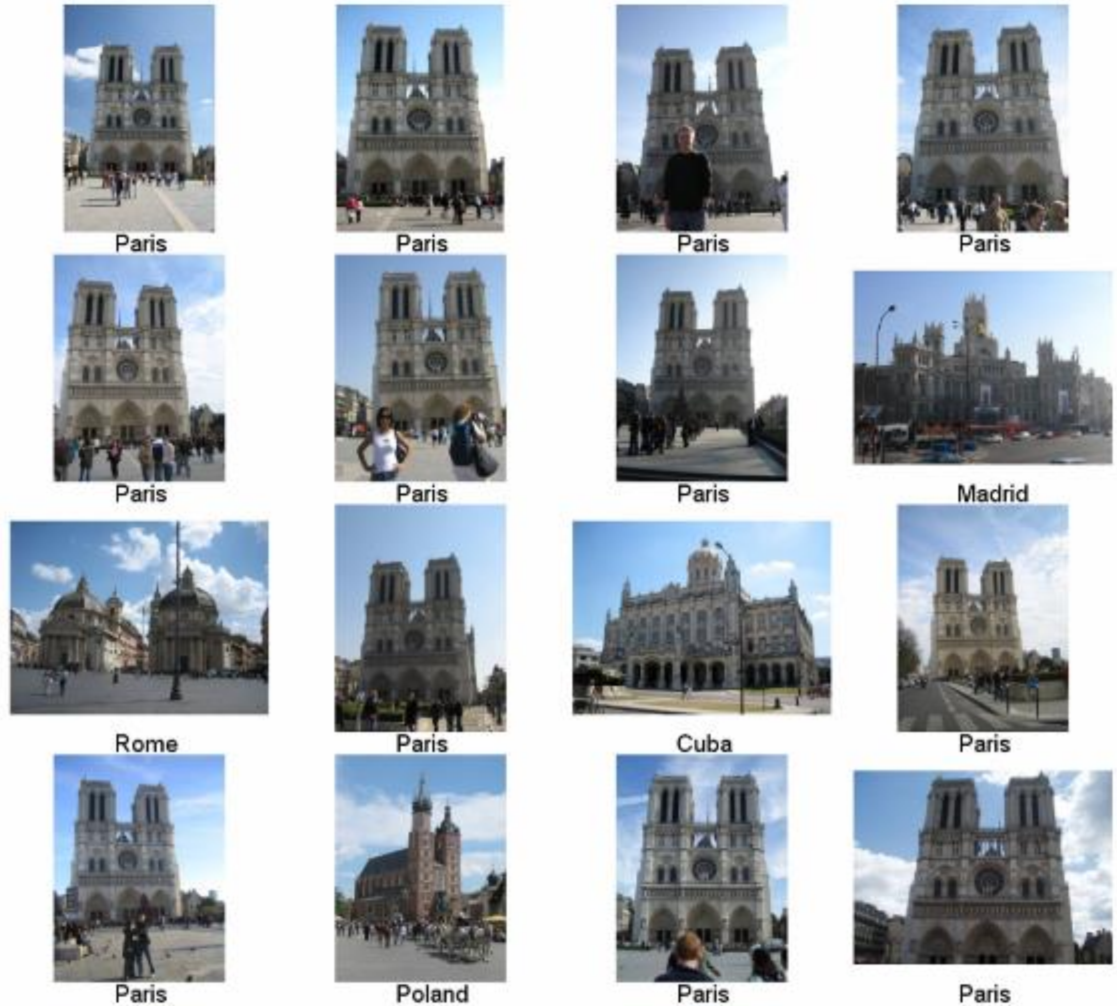
6 million geo-tagged Flickr images

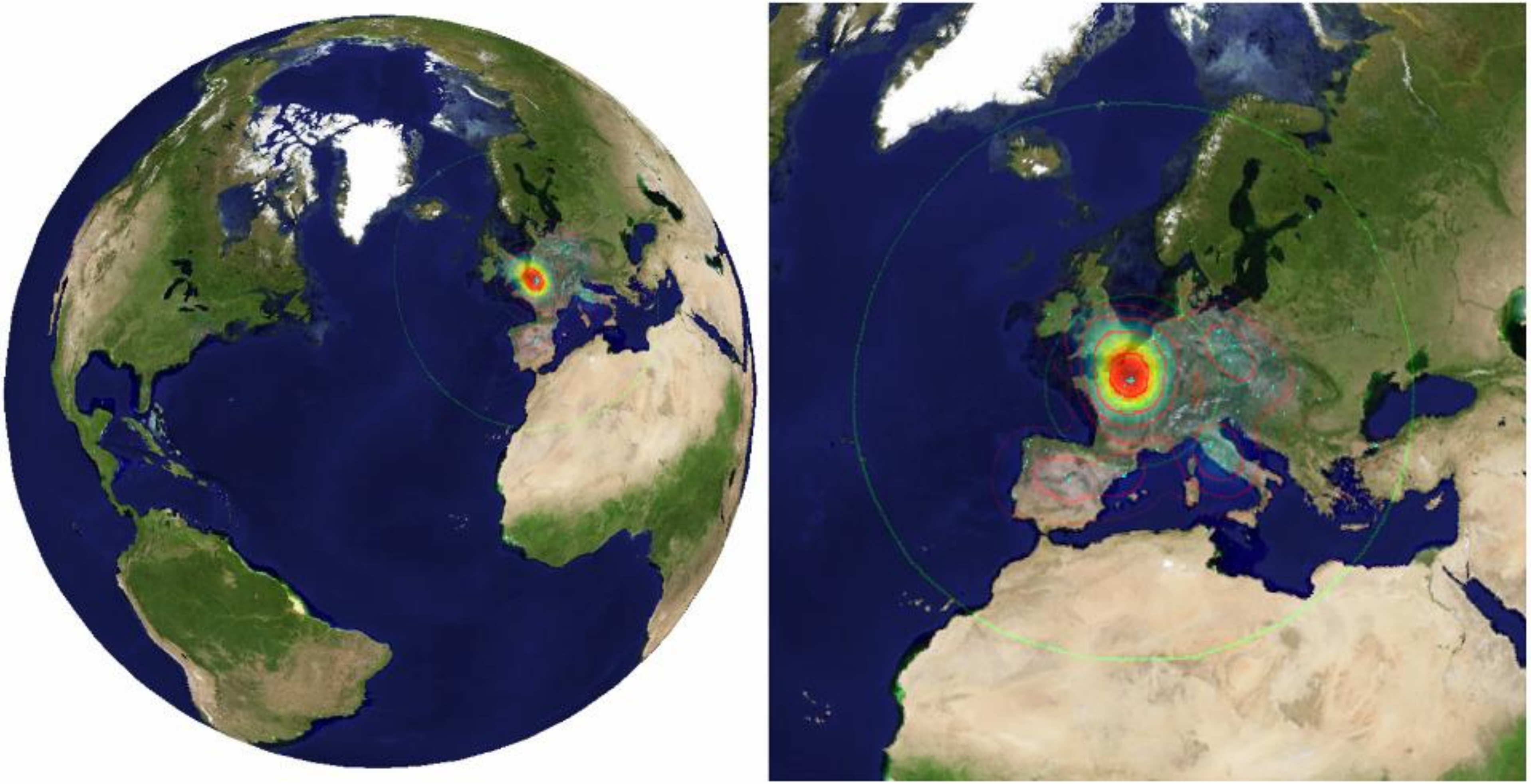
http://graphics.cs.cmu.edu/projects/im2gps/

im2gps (Hays & Efros, CVPR 2008)

Nearest Neighbors according to gist + bag of SIFT + color histogram + a few others



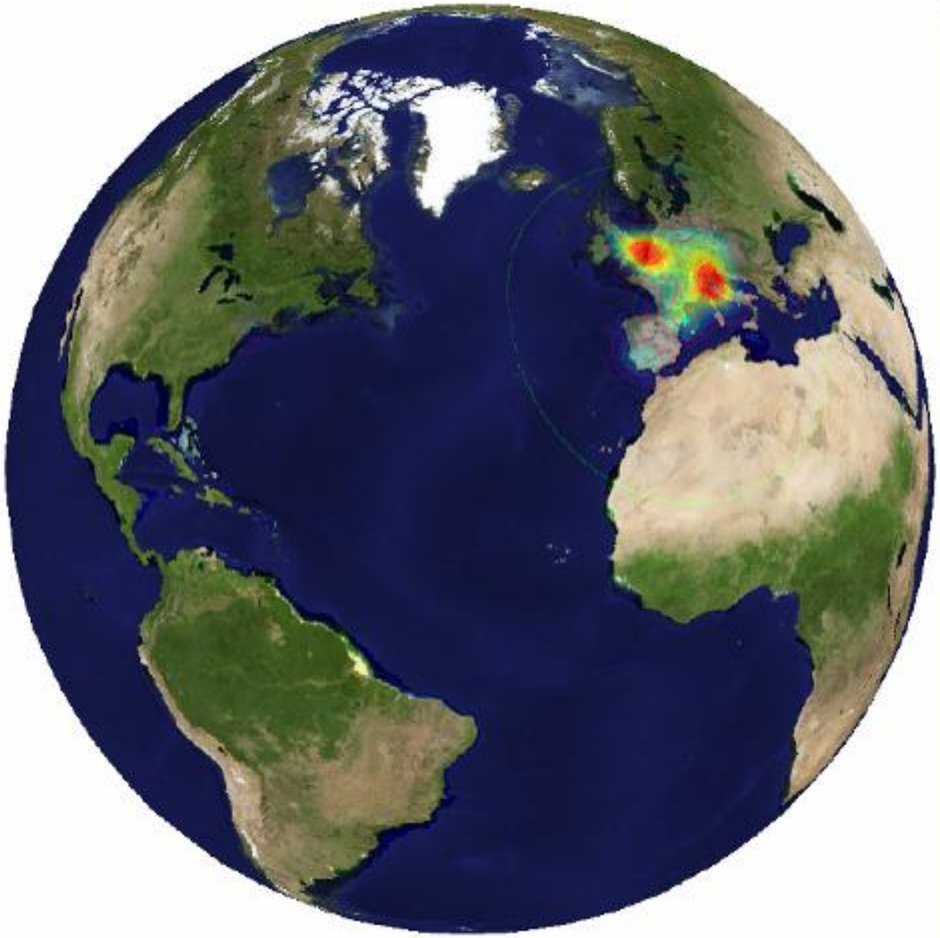


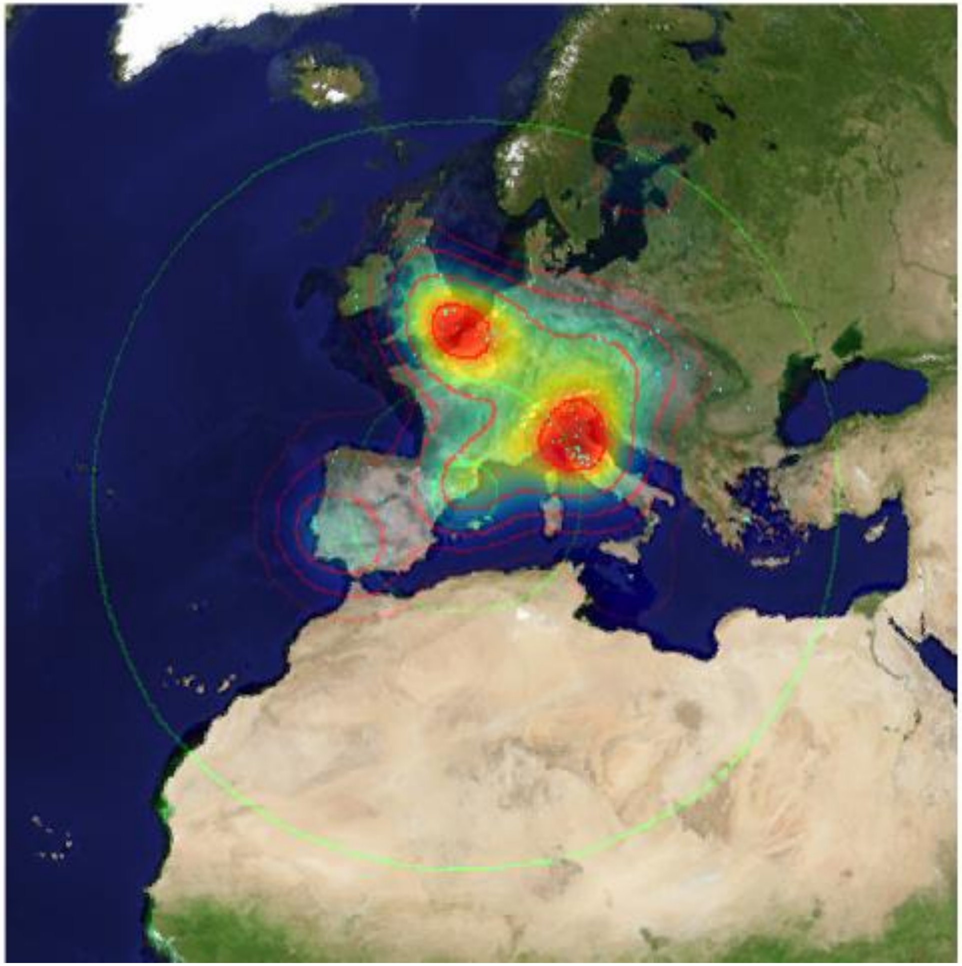


Im2gps

Example Scene Matches

Voting Scheme





im2gps

Philippines

Houston

Thailand

Houston

Maldives

Philippines

NewZealand

Bermuda

Palau

Mexico2

Brazil

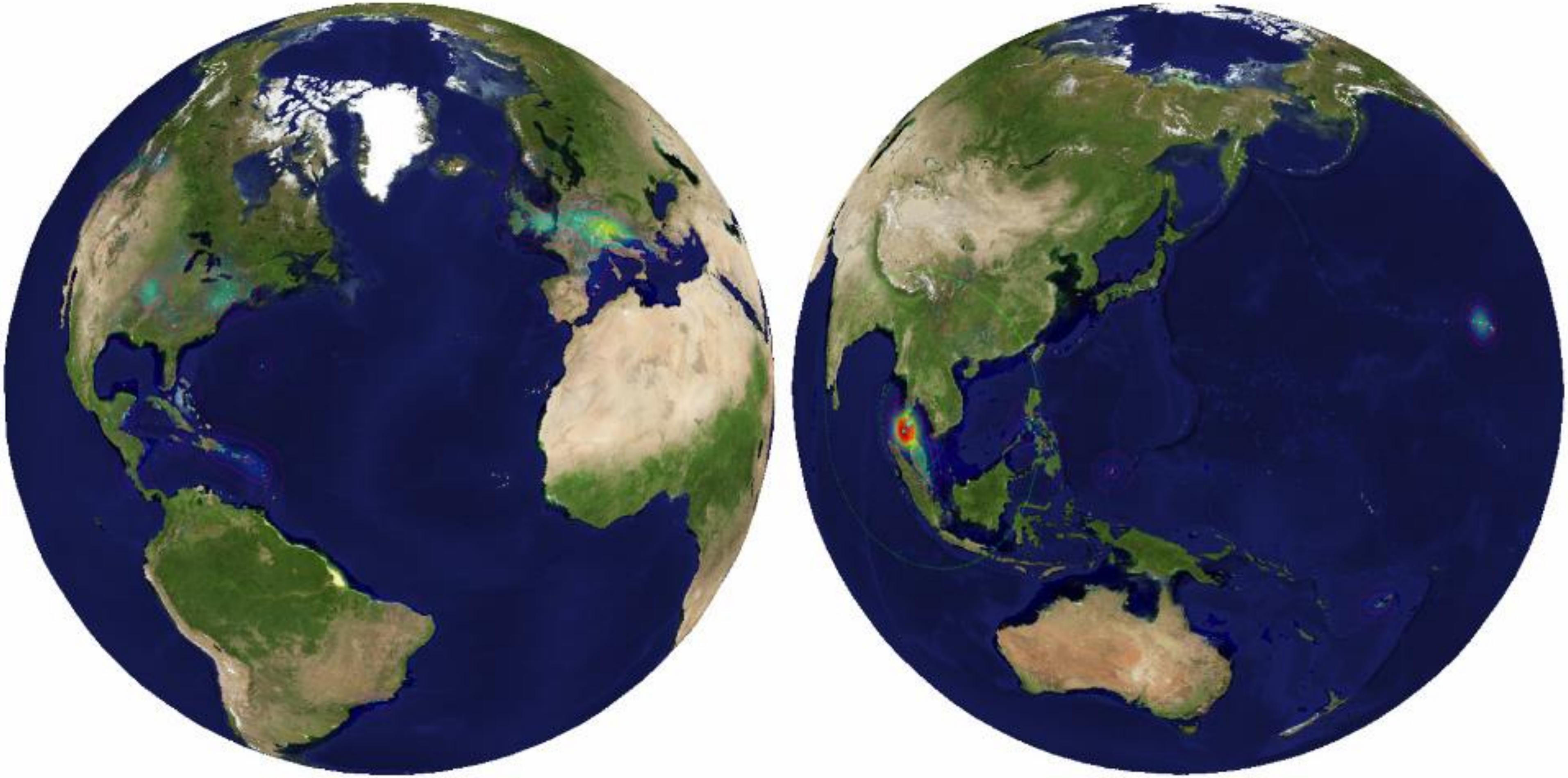
Mendoza

Brazil

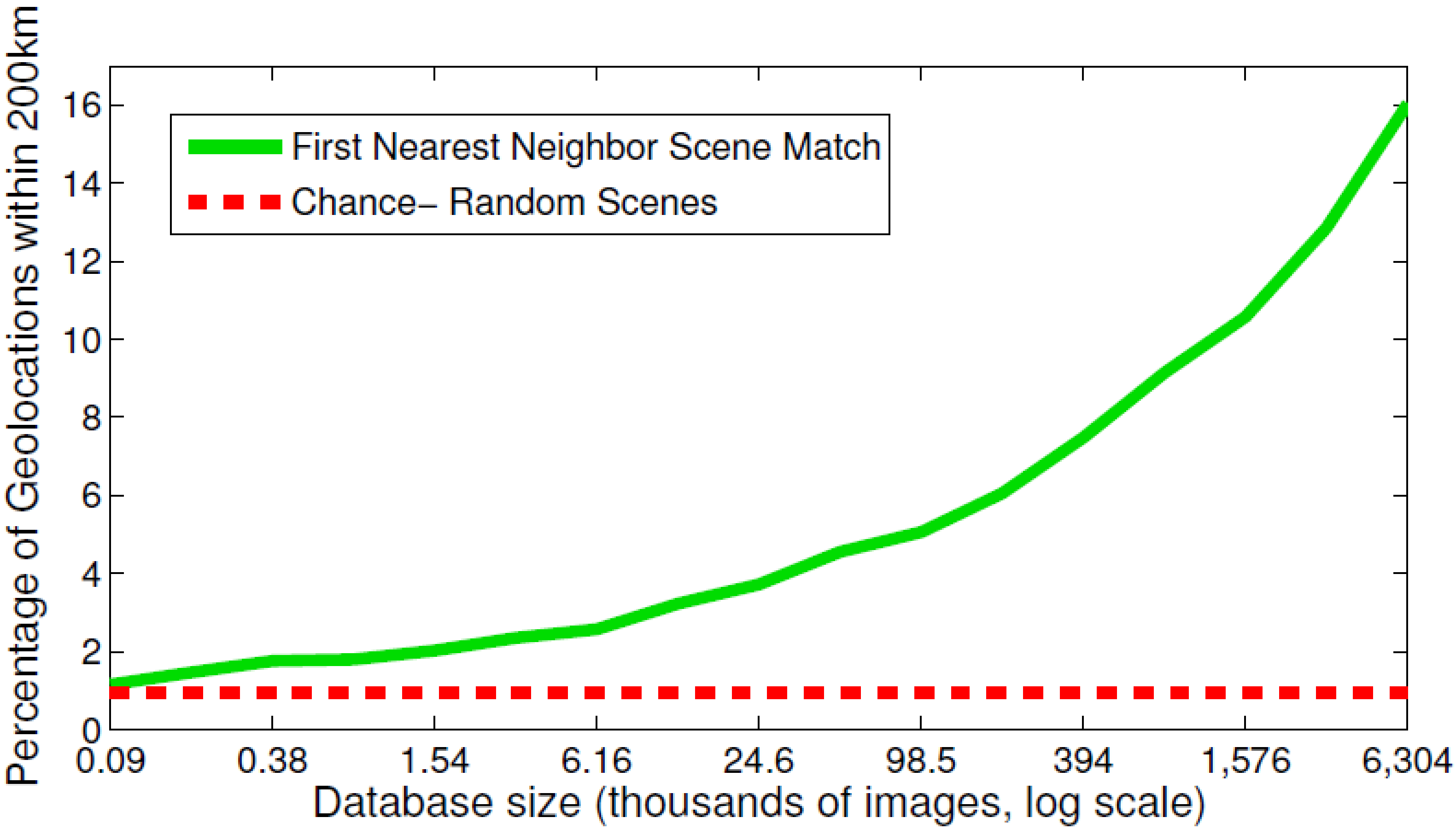
Thailand

Arkansas

Hawaii



Effect of Dataset Size

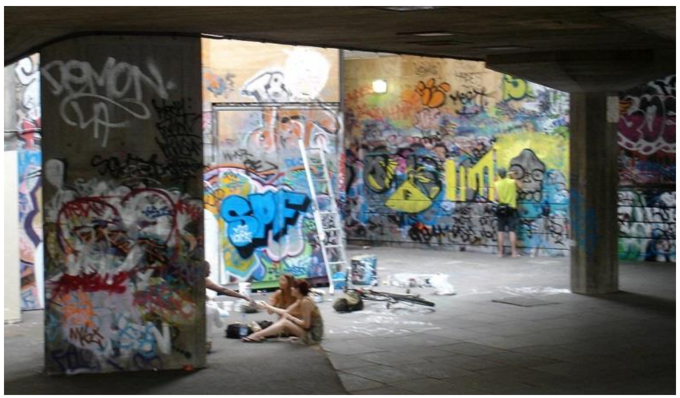


Where is This?

[Vesselova, Kalogerakis, Hertzmann, Hays, Efros. Image Sequence Geolocation. ICCV'09]

Where is This?

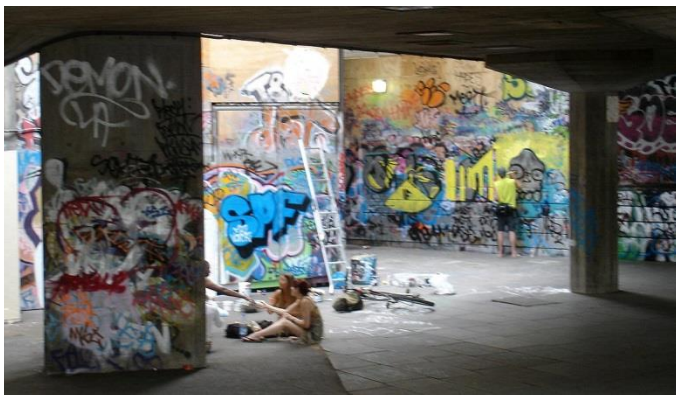
Where are These?



15:14,
June 18th, 2006

16:31,
June 18th, 2006

Where are These?



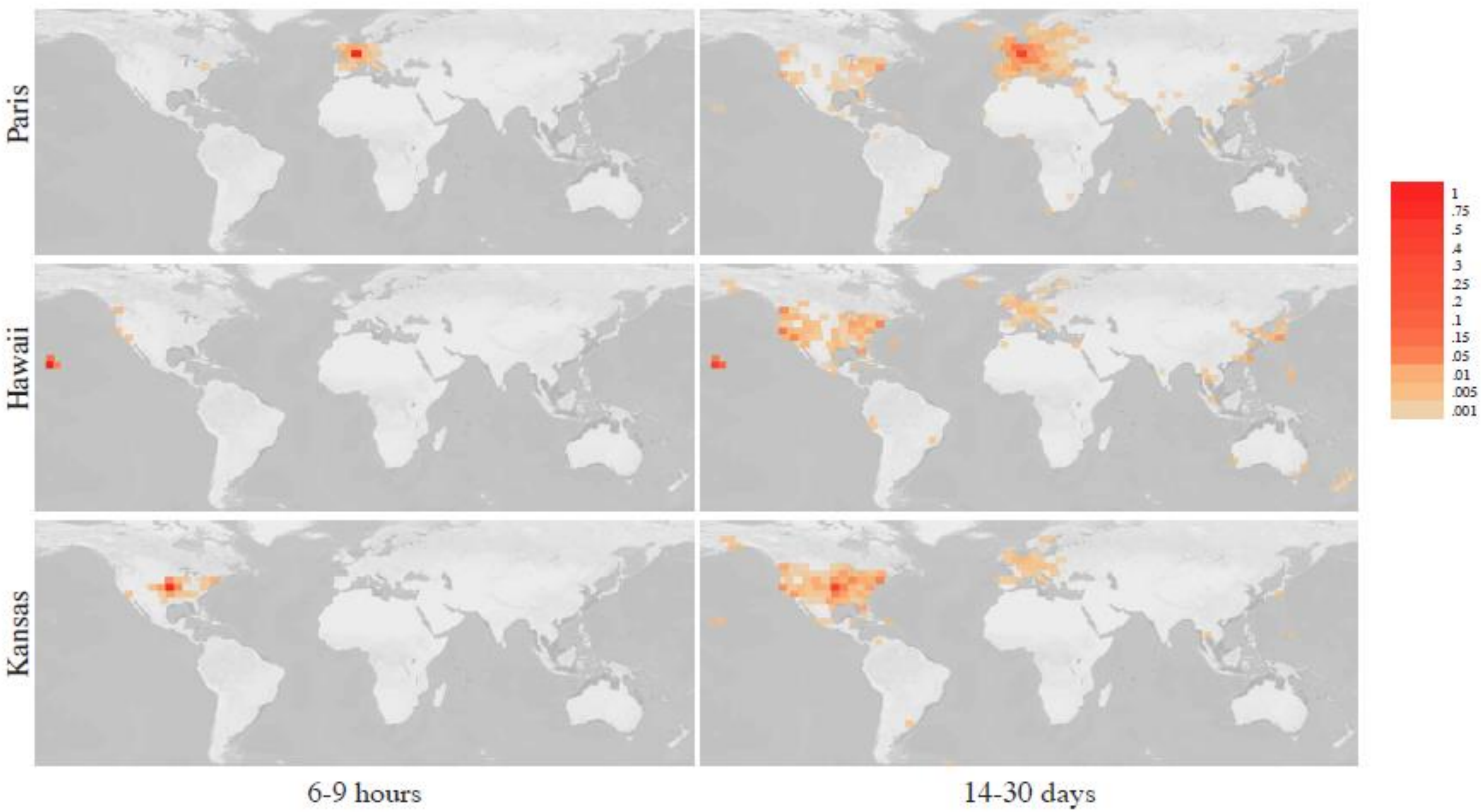
15:14,
June 18th, 2006

16:31,
June 18th, 2006

17:24,
June 19th, 2006

Results

- im2gps – 10% (geo-loc within 400 km)
- temporal im2gps – 56%



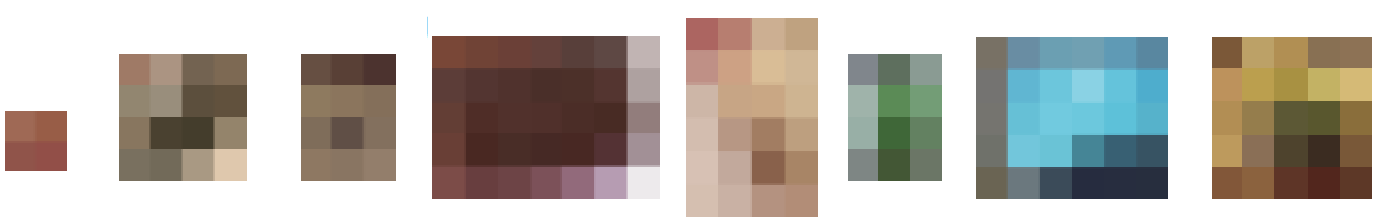
Tiny Images

80 million tiny images: a large dataset for non-parametric object and scene recognition

Antonio Torralba, Rob Fergus and William T. Freeman. PAMI 2008.
http://groups.csail.mit.edu/vision/TinyImages/

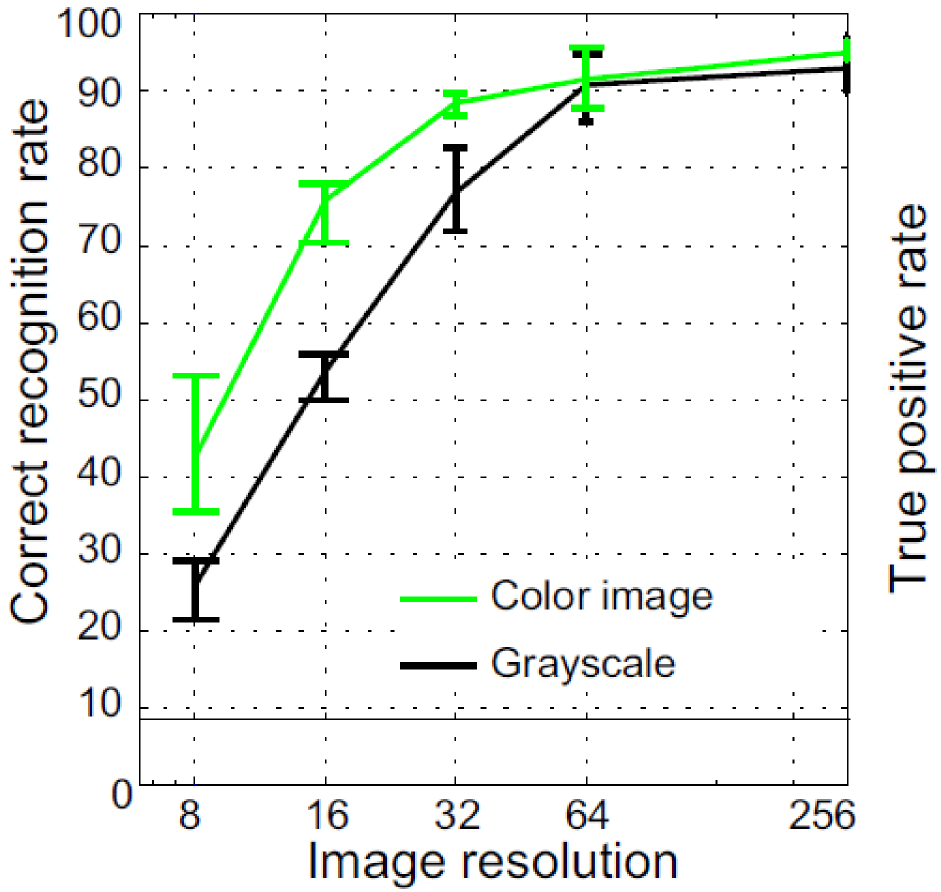
32x32


256x256
32x32



c) Segmentation of 32x32 images

Given a benchmark, resolution and human scene recognition accuracy increase to a limit

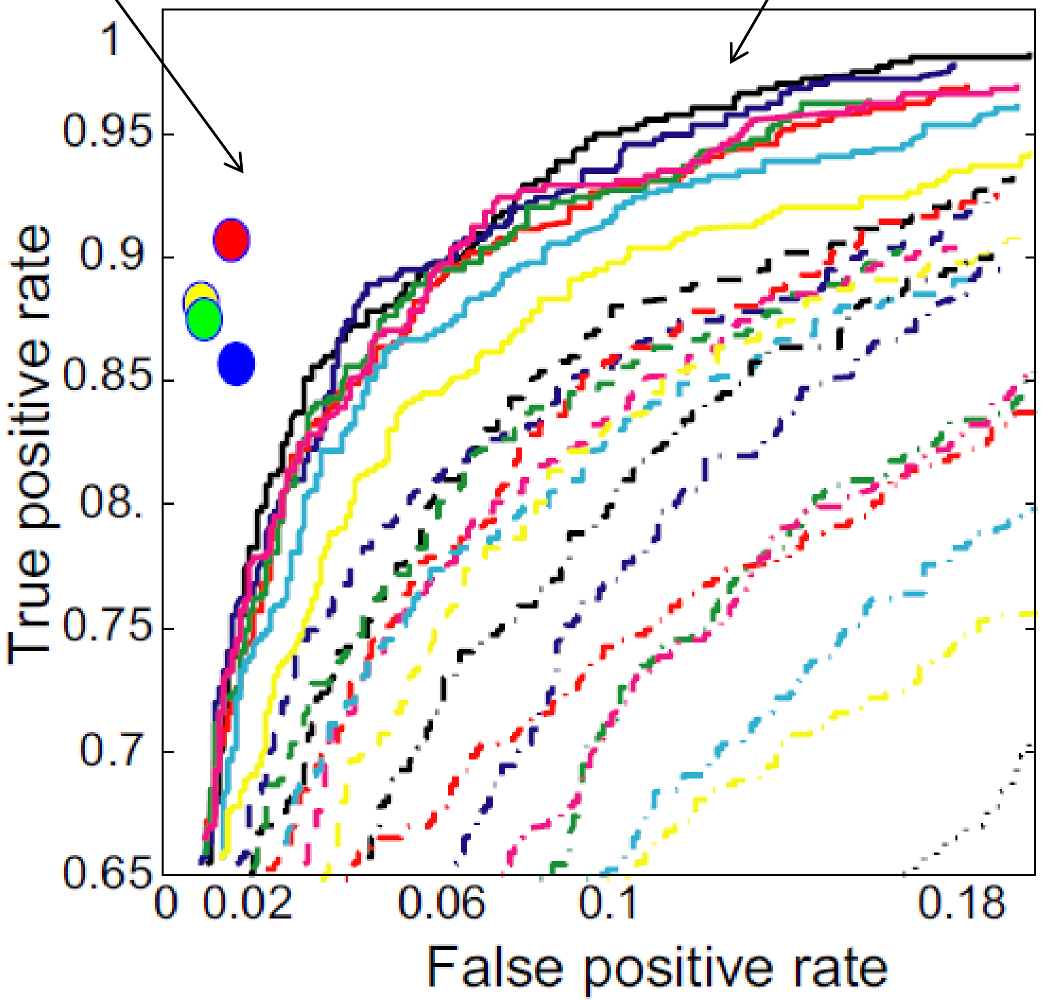


Torralba et al.

Humans vs. Computers: Car Classification

Humans for 32 pixel tall images

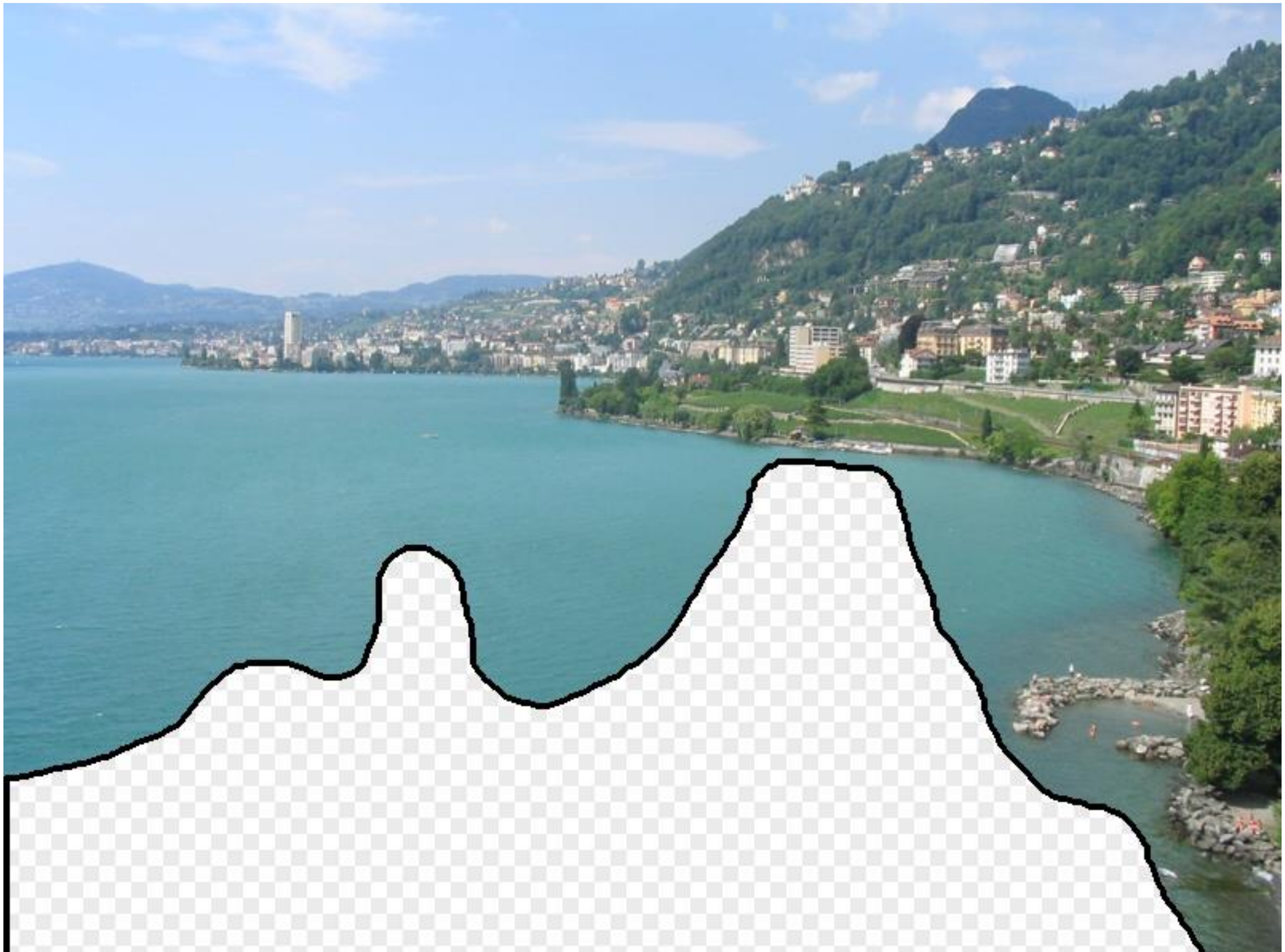
Various computer vision algorithms
for *full* resolution images



Example
test images

Torralba et al.

What should the missing region contain?



Which is the original?

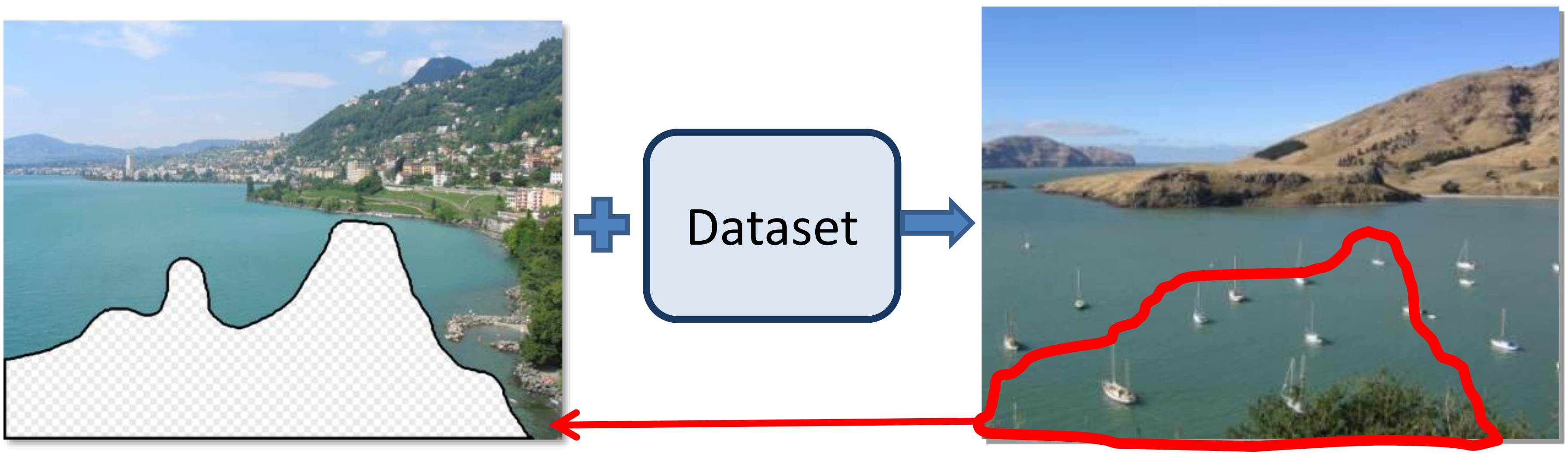
(a)

(b)

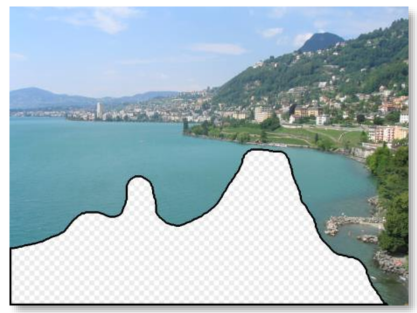
(c)

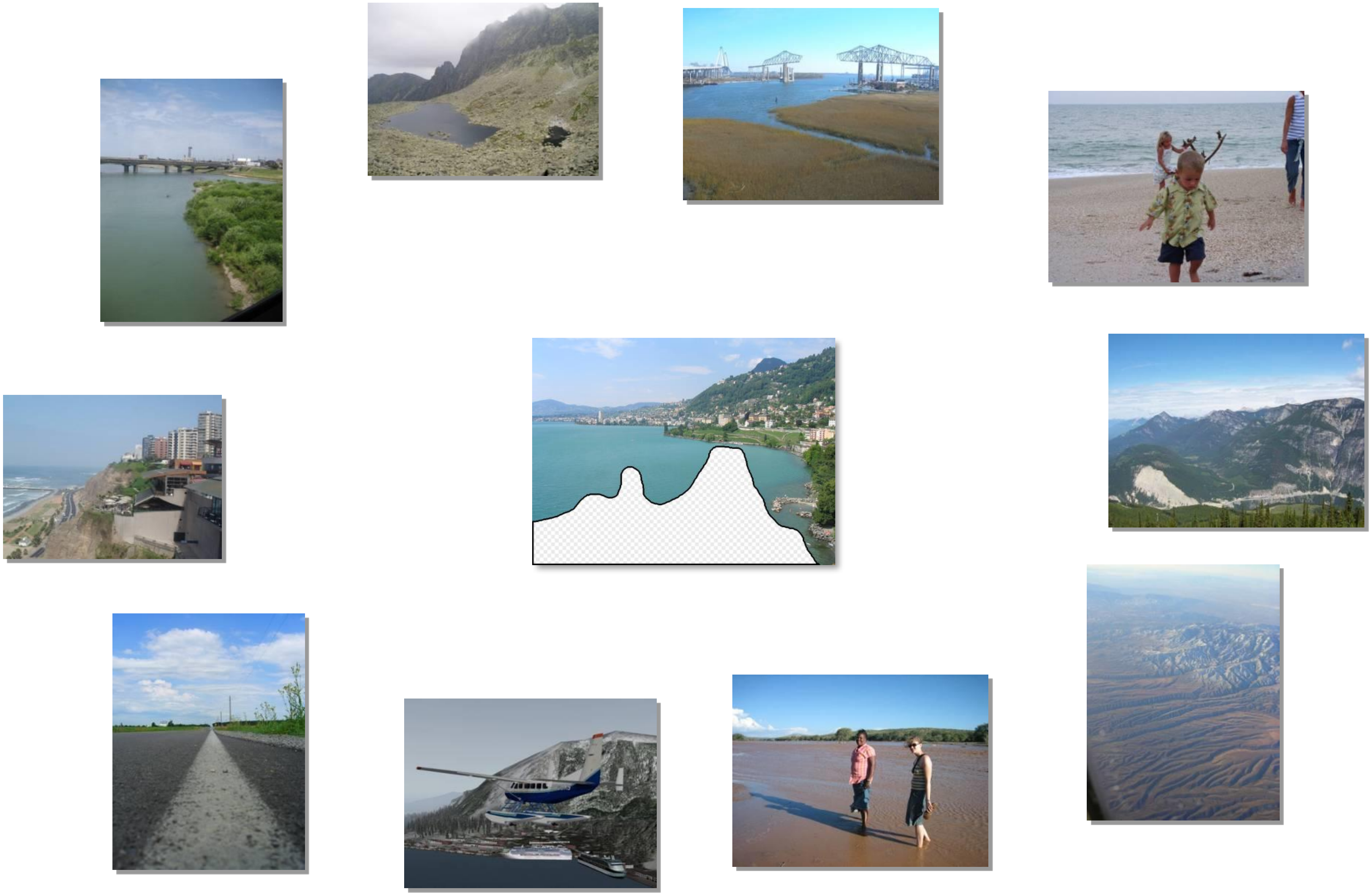
How it works

- Find a similar image from a large dataset
- Blend a region from that image into the hole

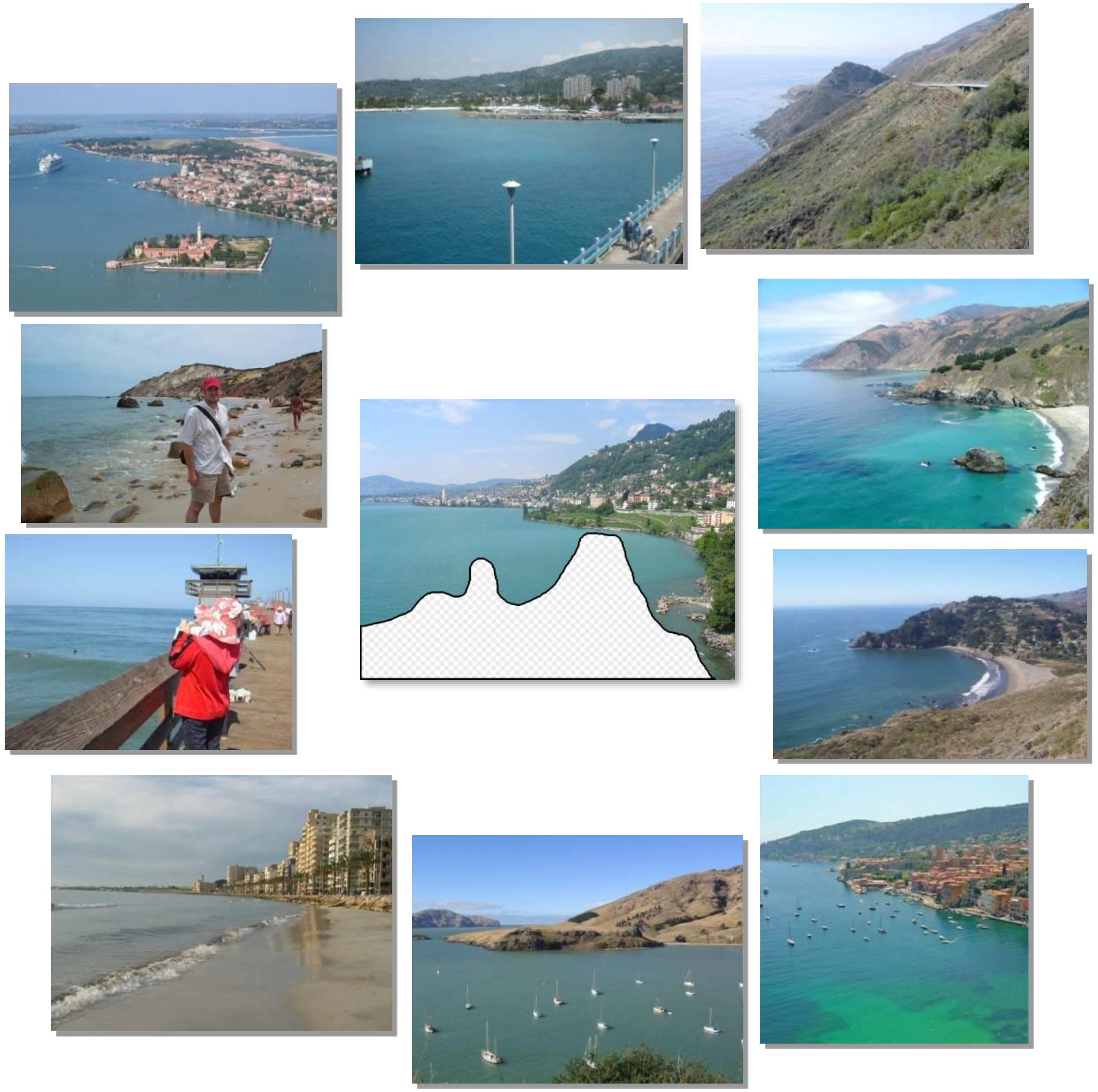


How many images is enough?





Nearest neighbors from a collection of 20 thousand images



Nearest neighbors from a
collection of 2 million images

Image Data on the Internet

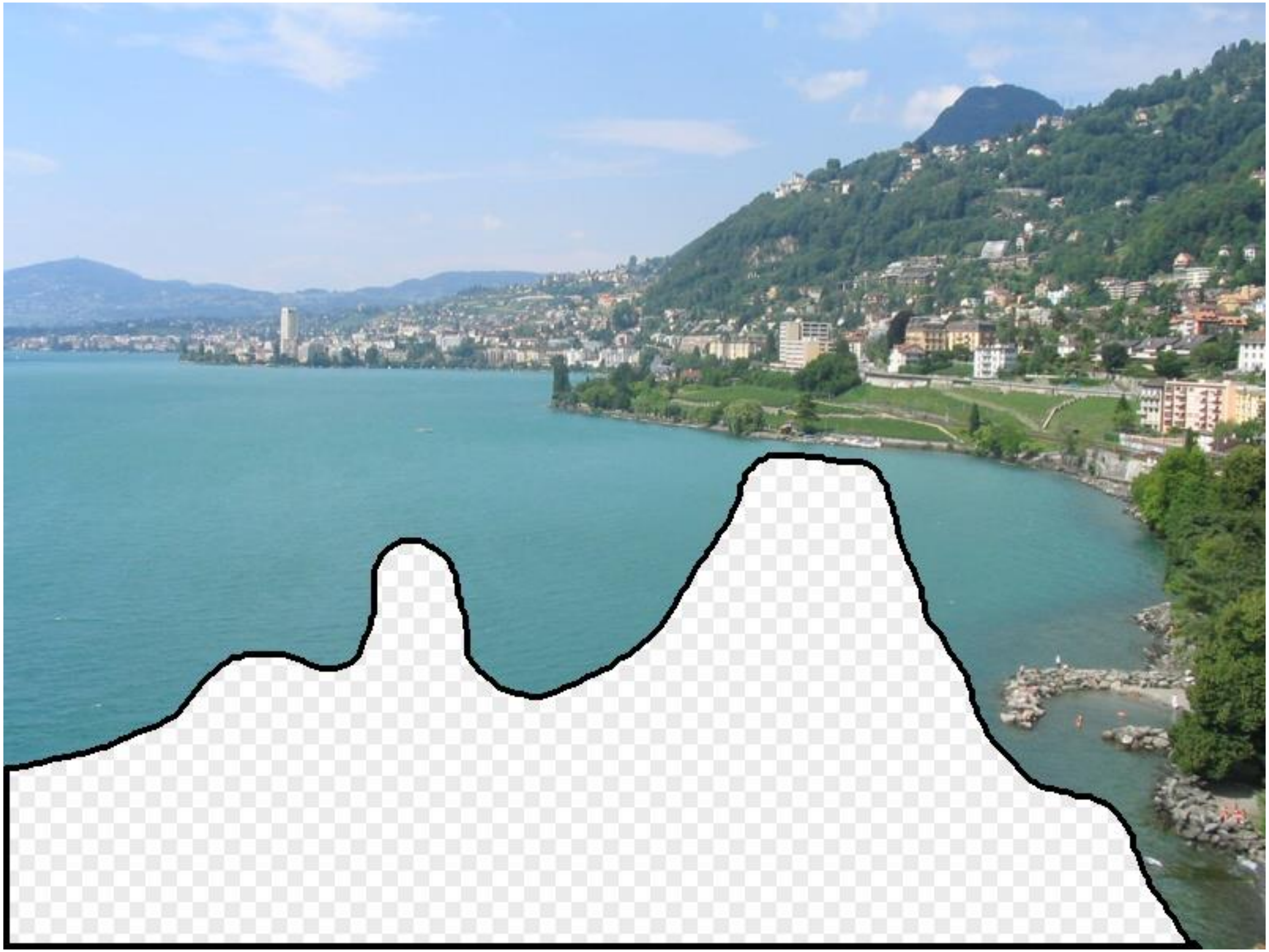
- Flickr (as of Nov 2013)
  - 10 billion photographs
  - 100+ million geotagged images
  - 3.5 million a day
- Facebook (as of Sept 2013)
  - 250 billion+
  - 300 million a day
- Instagram
  - 55 million a day

Image completion: how it works

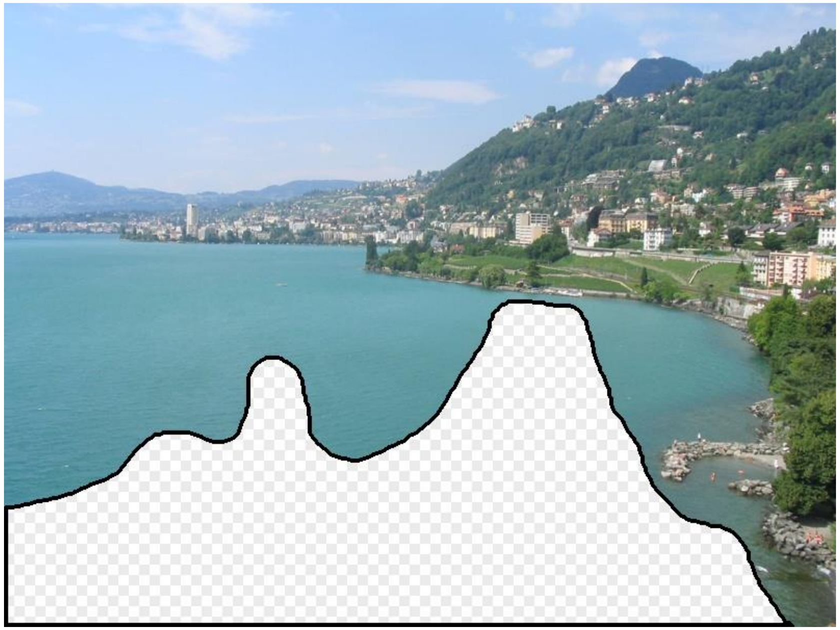
[Hays and Efros. Scene Completion Using Millions of Photographs. SIGGRAPH 2007 and CACM October 2008.]

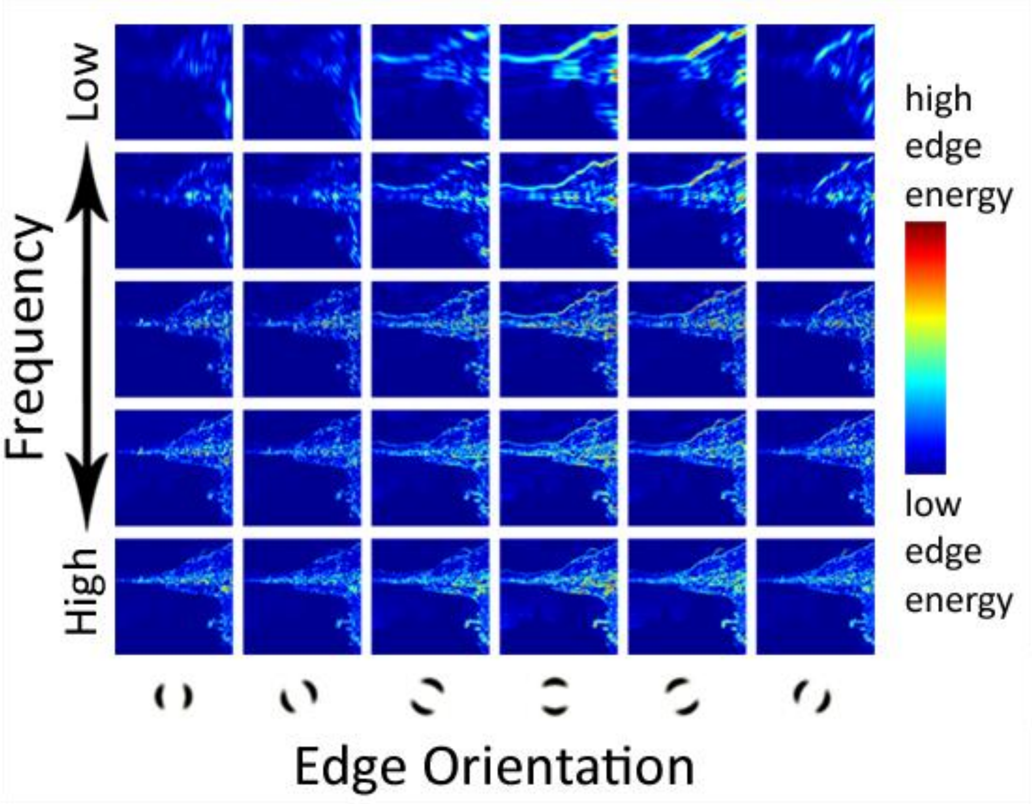
The Algorithm

Scene Matching

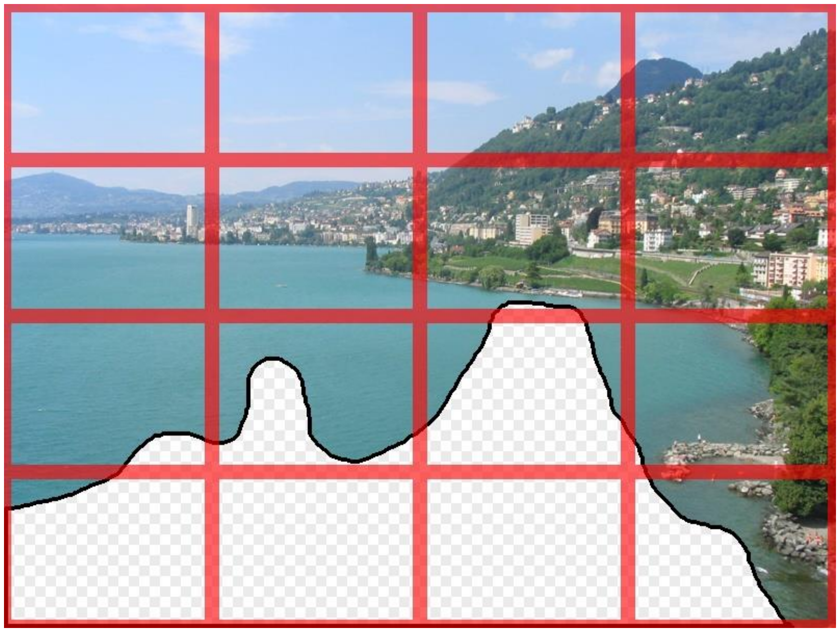


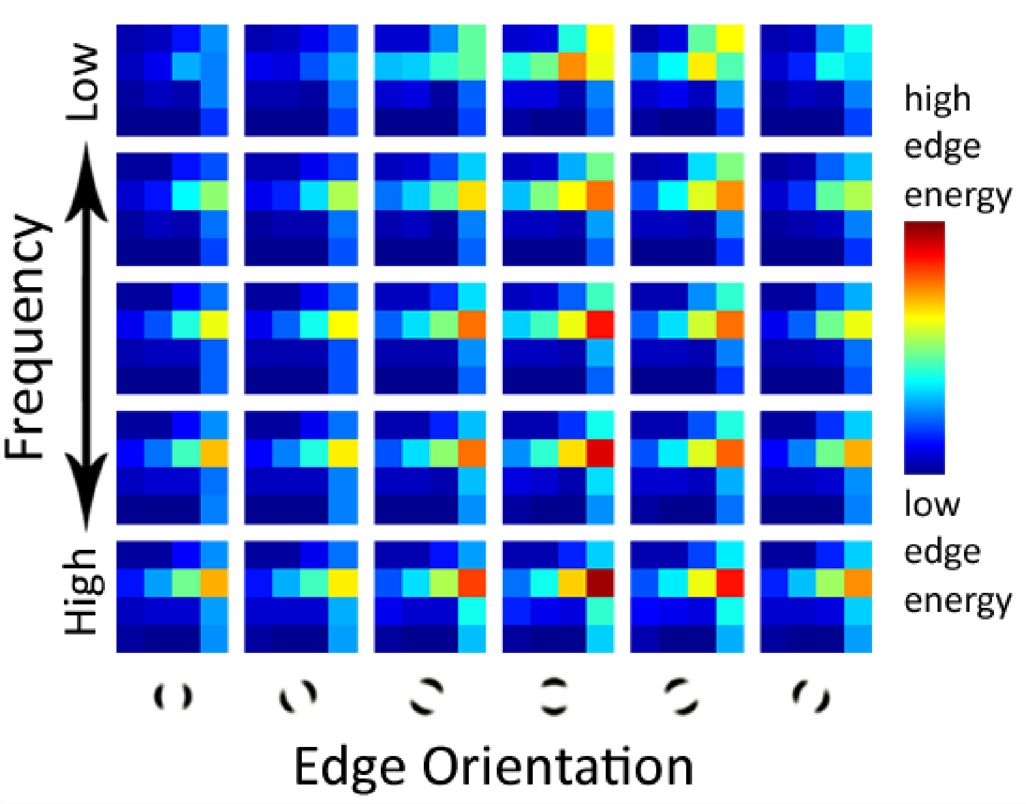
Scene Descriptor





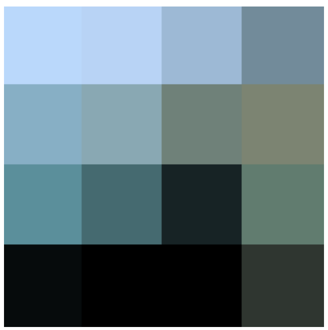
Scene Descriptor



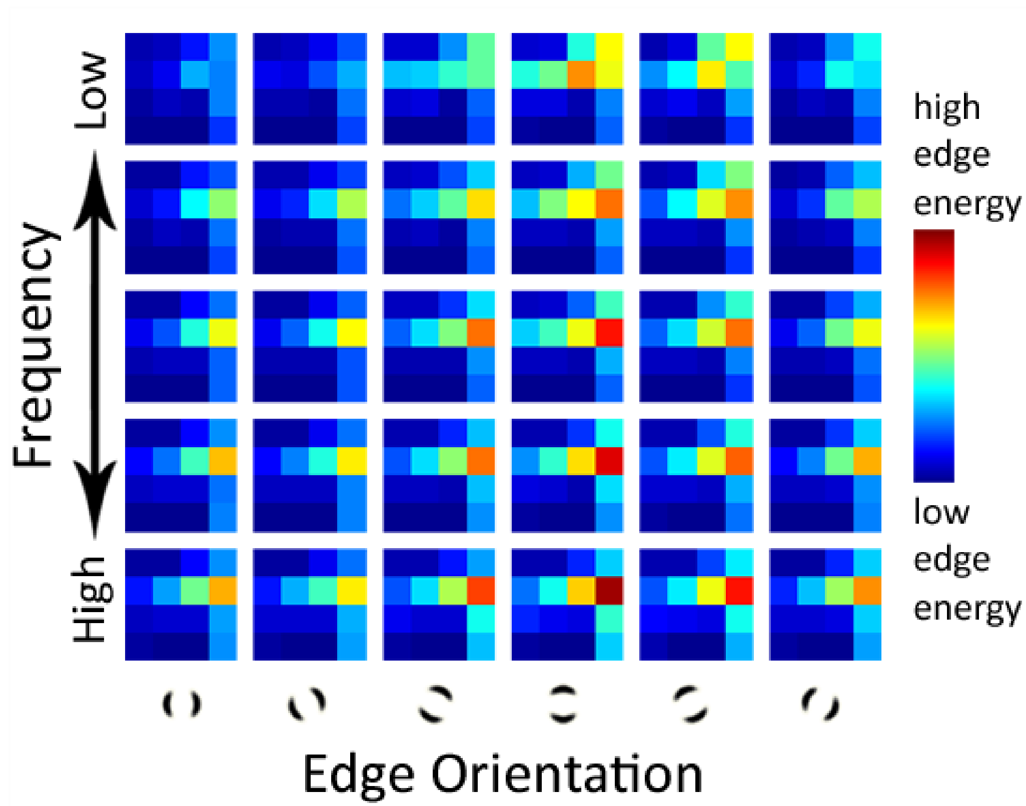


Scene Gist Descriptor
(Oliva and Torralba 2001)

Scene Descriptor

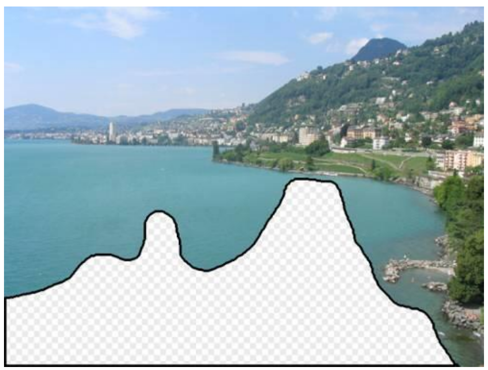


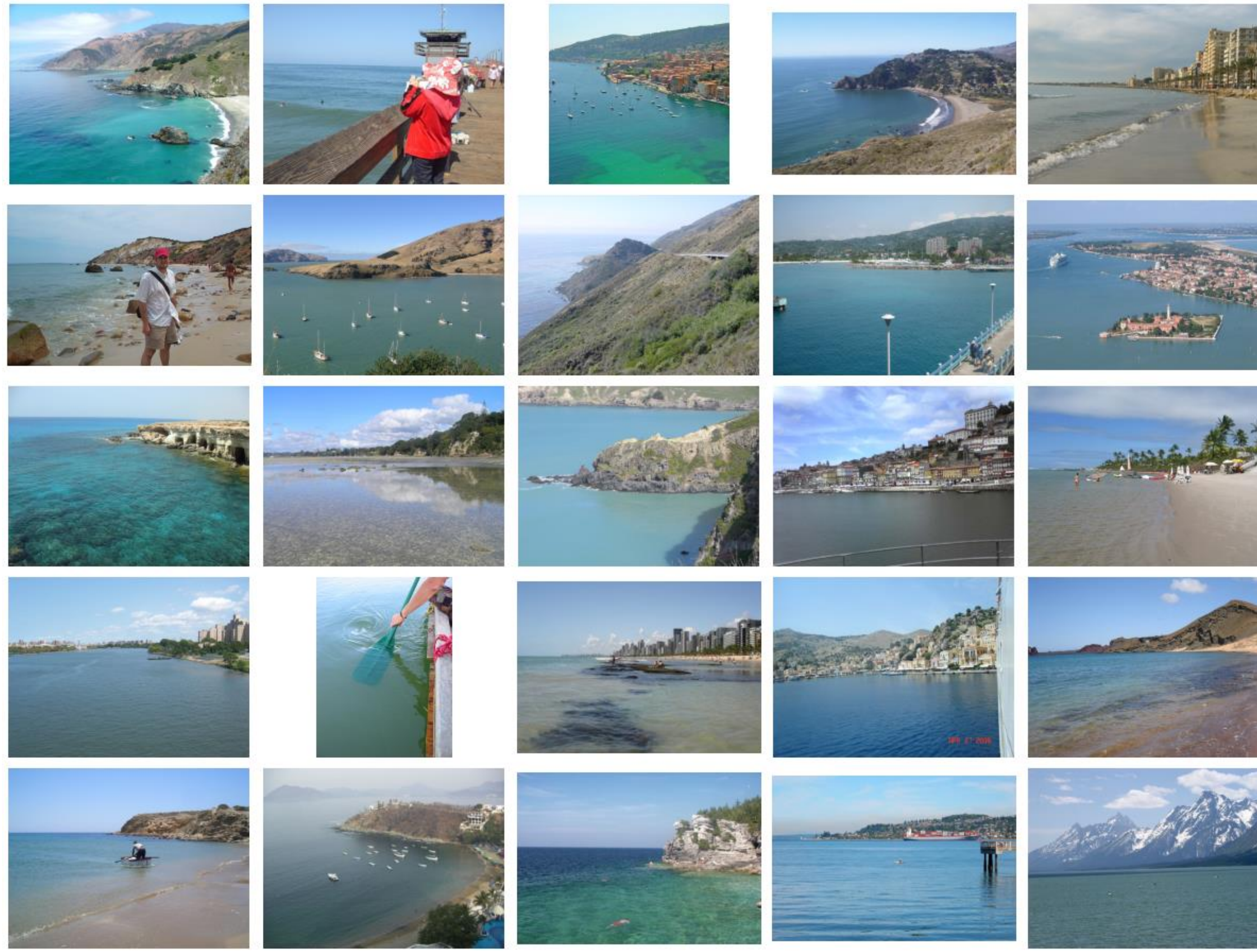
+



Scene Gist Descriptor
(Oliva and Torralba 2001)

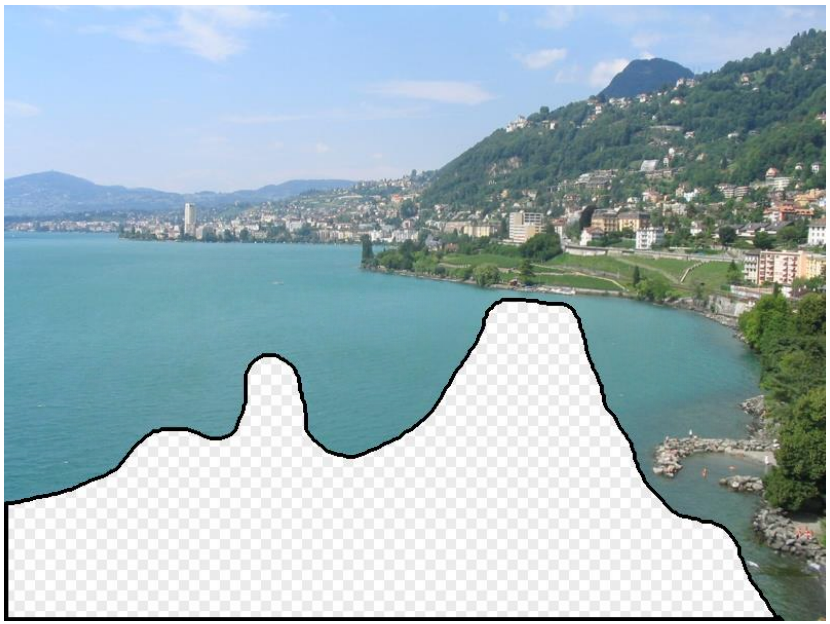
2 Million Flickr Images





... 200 total

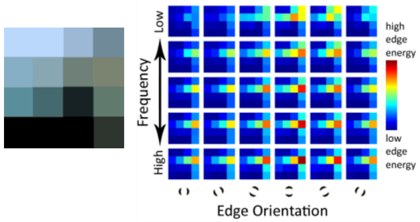
Context Matching



Graph cut + Poisson blending

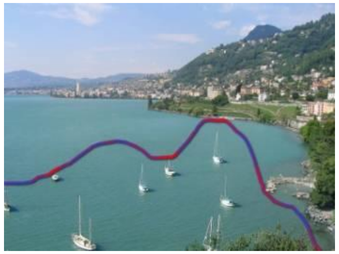
Result Ranking

We assign each of the 200 results a score which is the sum of:

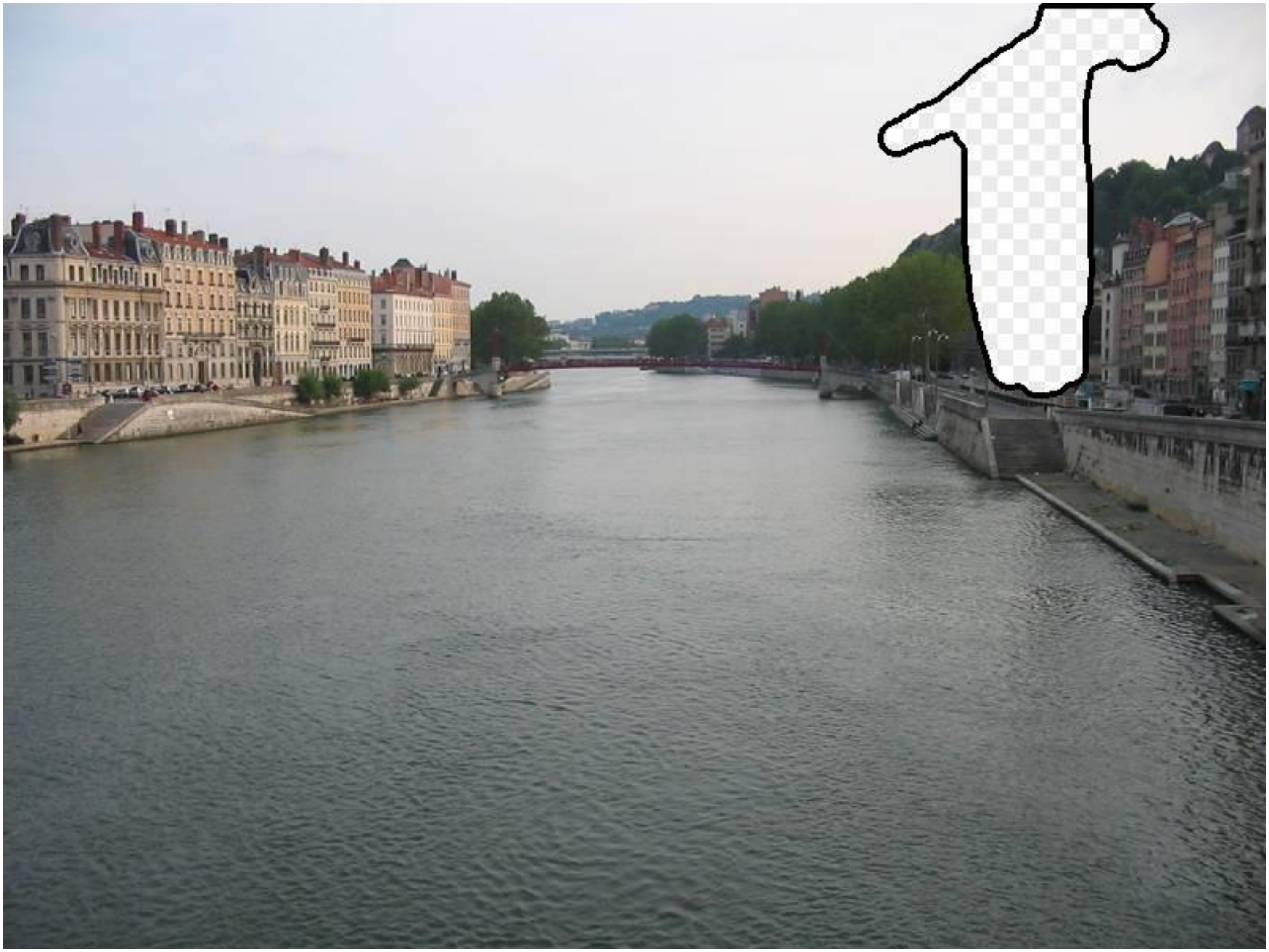


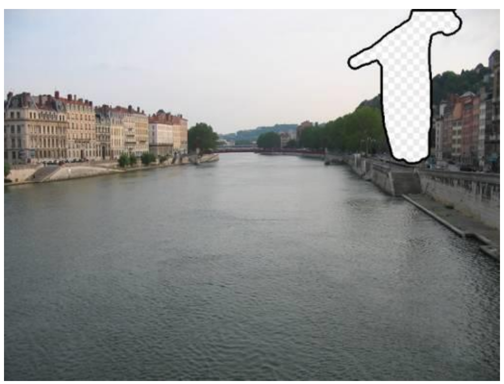
The scene matching distance

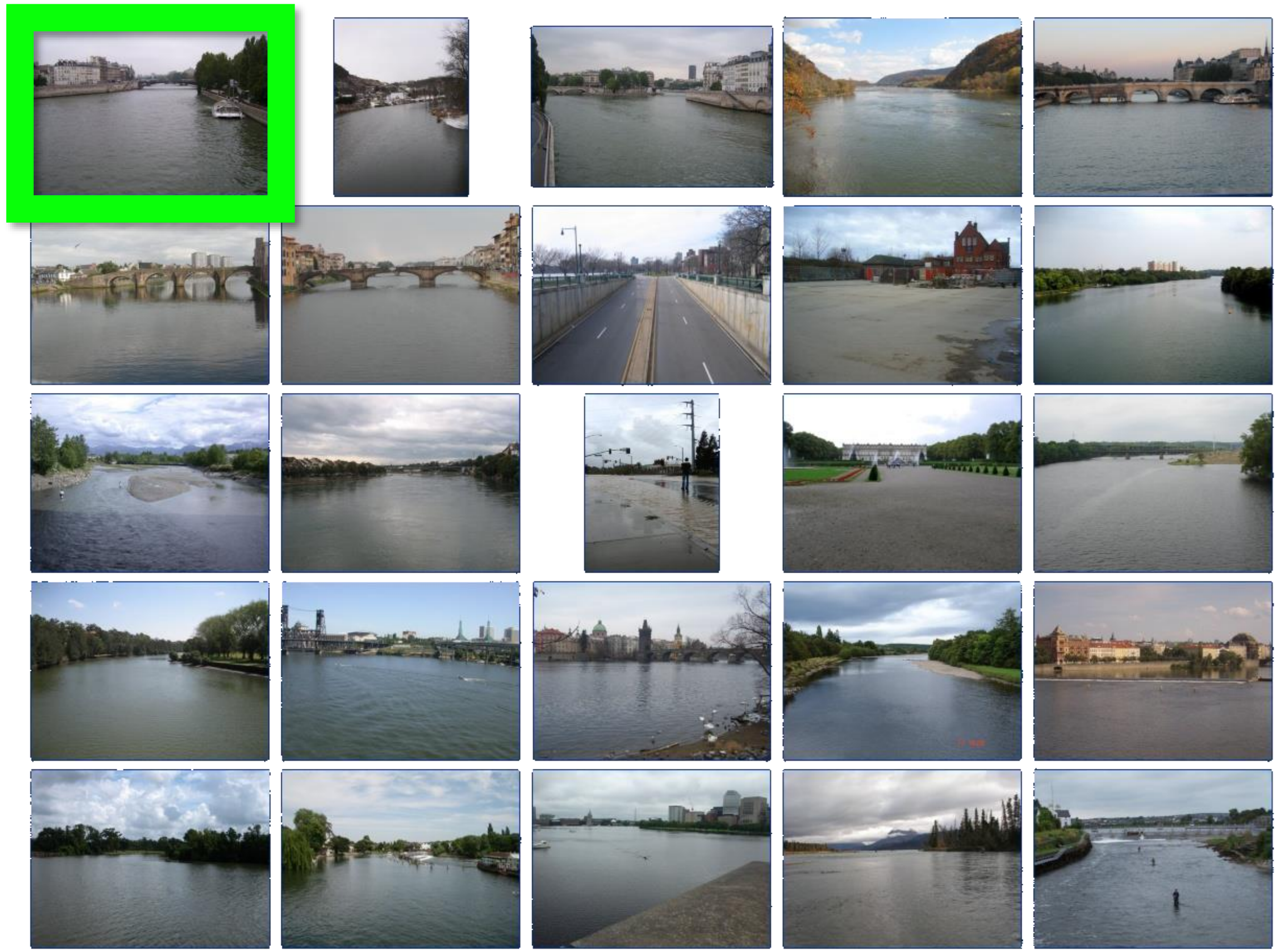
The context matching distance
(color + texture)



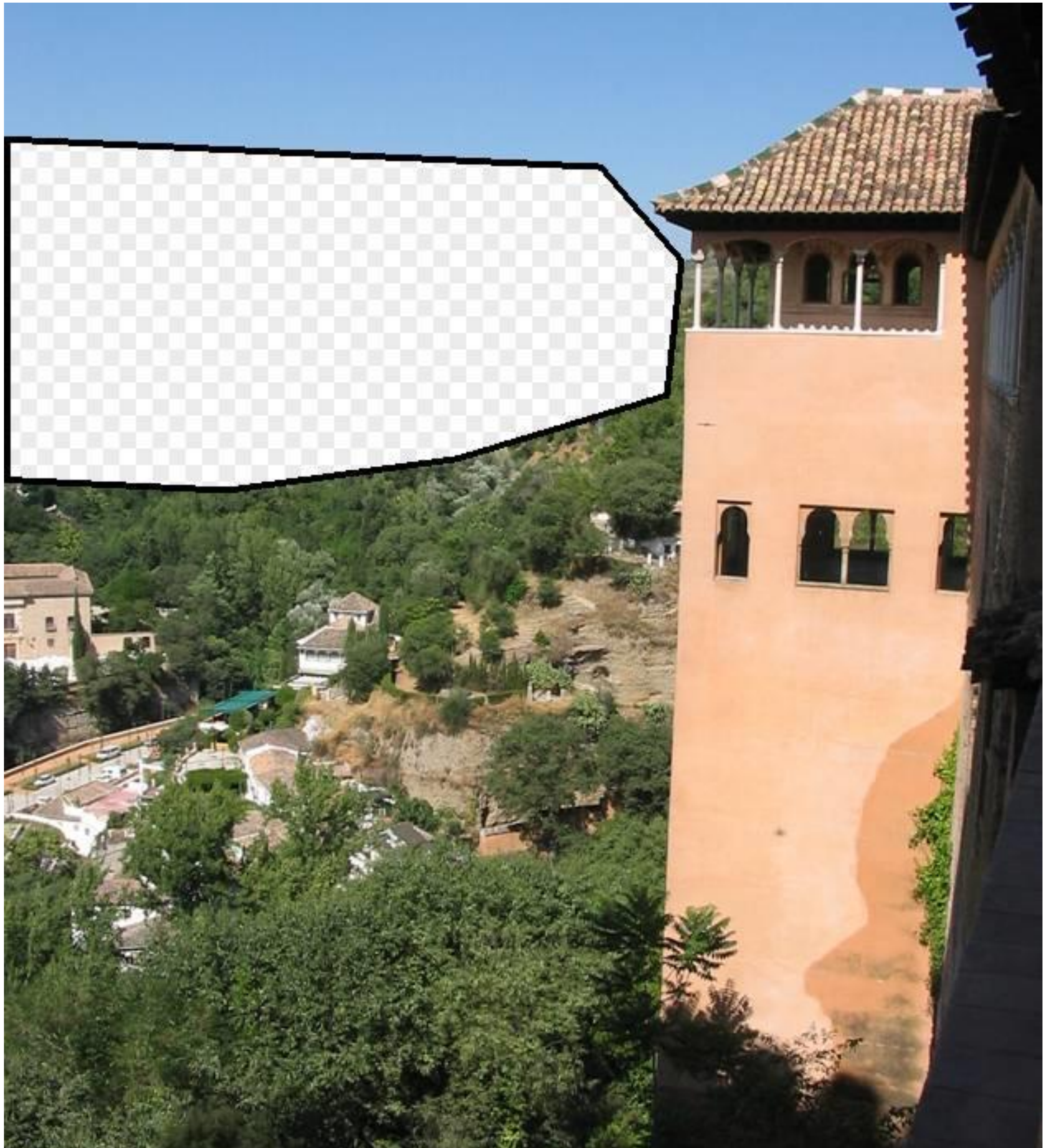
The graph cut cost







… 200 scene matches



Which is the original?

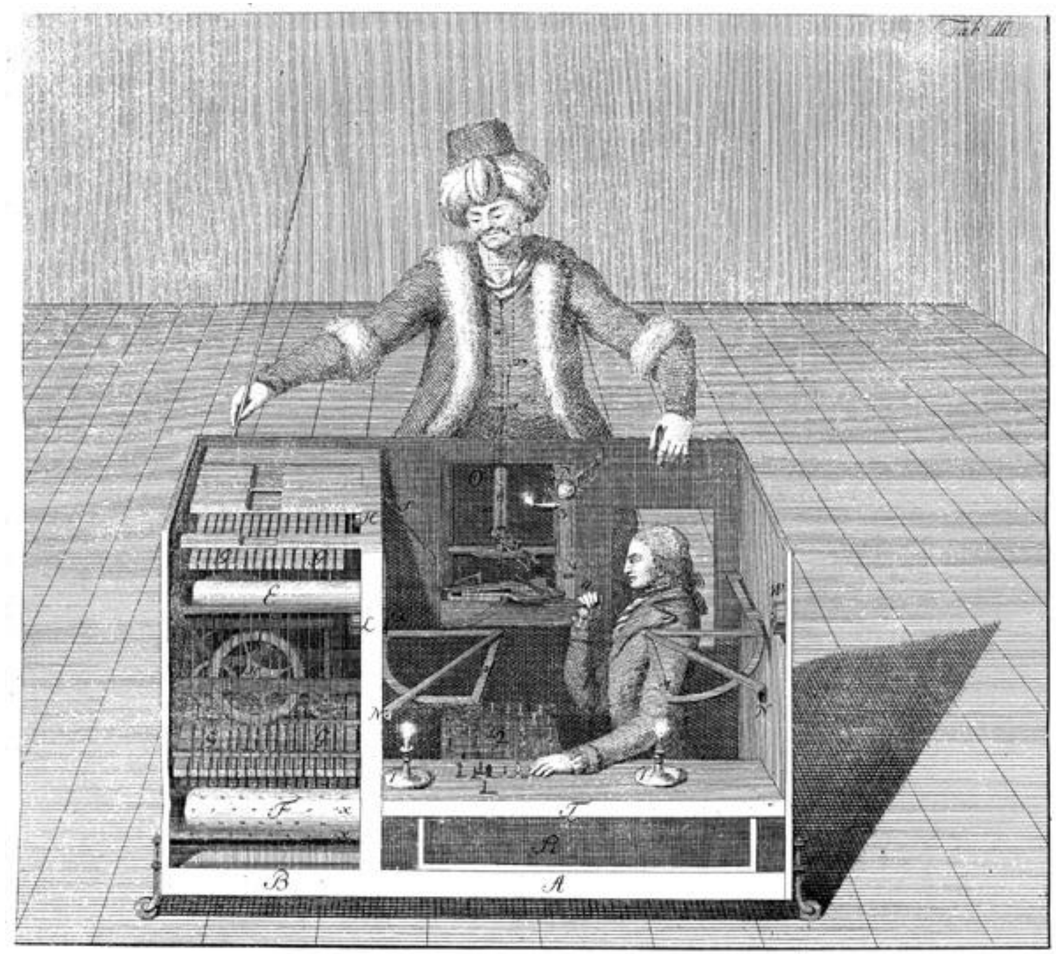
Mechanical Turk

- von Kempelen, 1770.
- Robotic chess player.
- Clockwork routines.
- Magnetic induction (not vision)

- Toured the world; played Napoleon Bonaparte and Benjamin Franklin.

Mechanical Turk

- It was all a ruse!
- Ho ho ho.



Amazon Mechanical Turk

Artificial artificial intelligence.

Launched 2005.
Small tasks, small pay.
Used extensively in data collection.

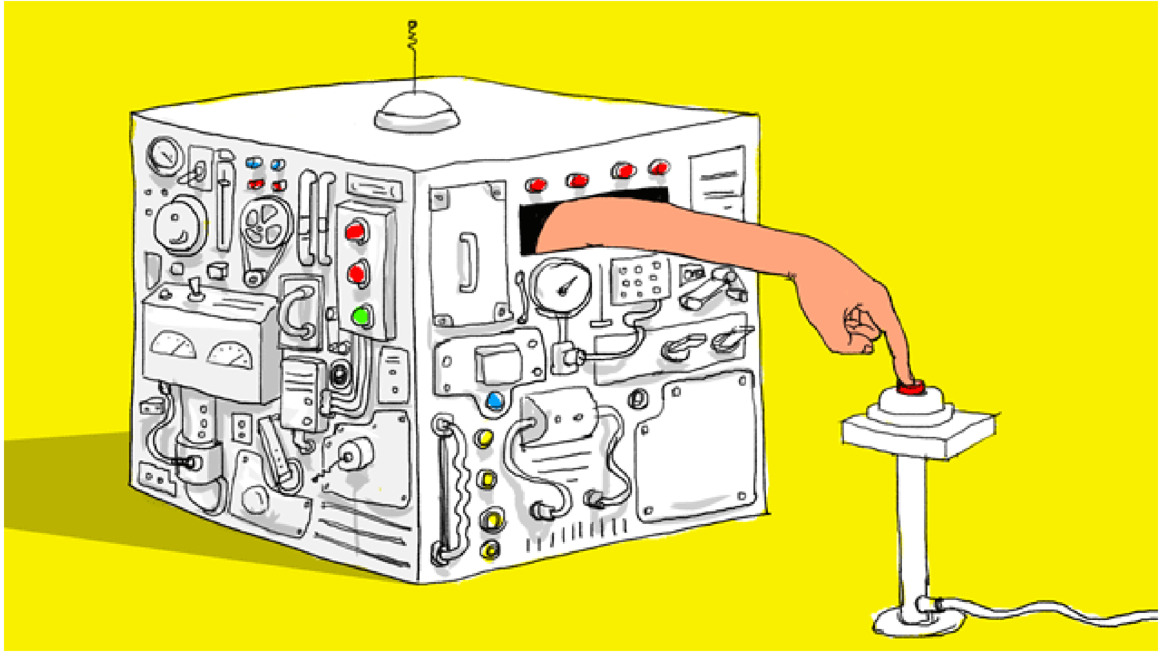
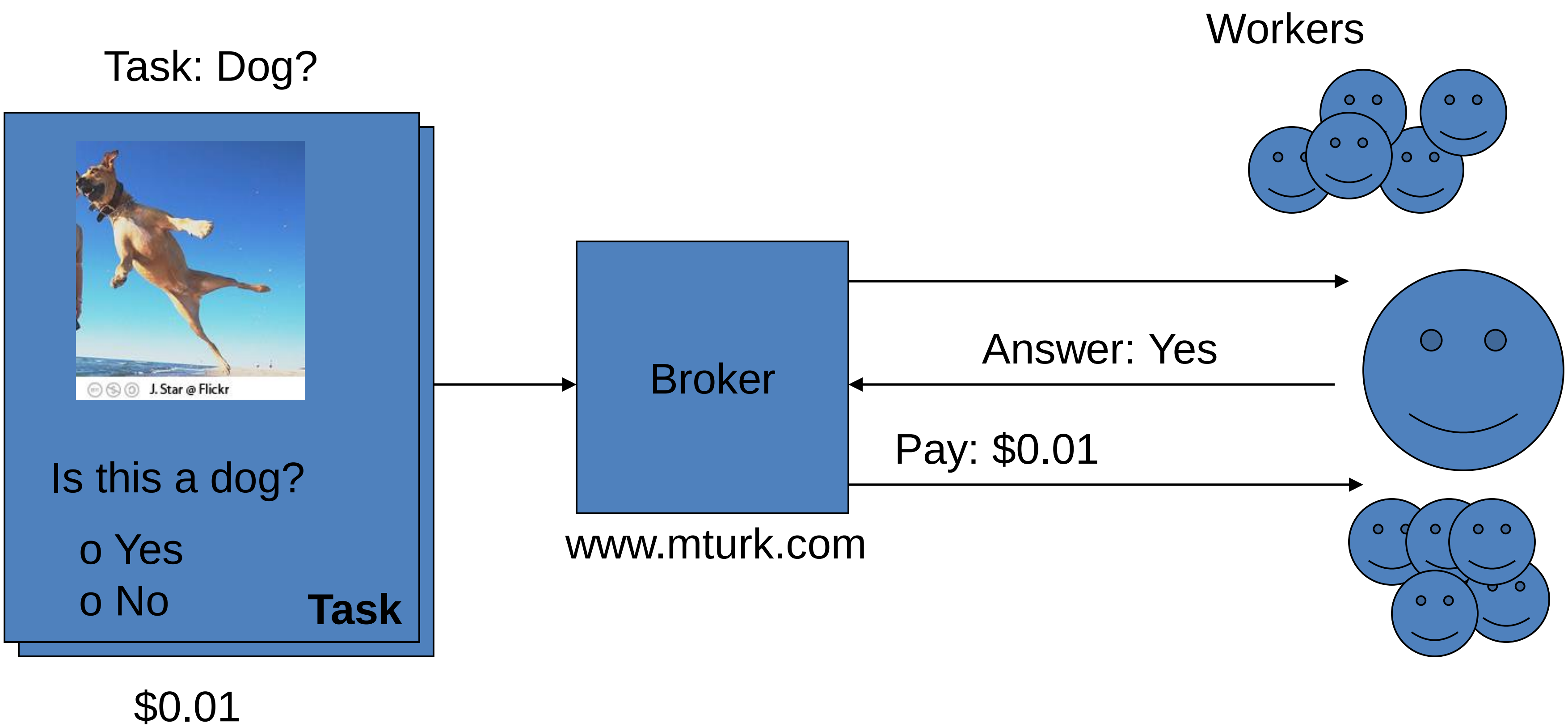


Image: Gizmodo

Amazon Mechanical Turk



Luis von Ahn and Laura Dabbish. [Labeling Images with a Computer Game](). ACM Conf. on Human Factors in Computing Systems, CHI 2004


ESP Game
Tag a Tune
Verbosity
Squigl
Matchin
score
0
ESP Game
Concentrate...
time
2:56
What do you see?
taboo words
student
guesses
submit
pass
Play Anonymously
Screencast-O-Matic.com

Vision (Segmentation): LabelMe

http://labelme.csail.mit.edu

"Open world" database annotated by the community*

Notes on Image Annotation, Barriuso and Torralba 2012. http://arxiv.org/abs/1210.3448

Utility data annotation via Amazon Mechanical Turk

 X 100 000 = $5000

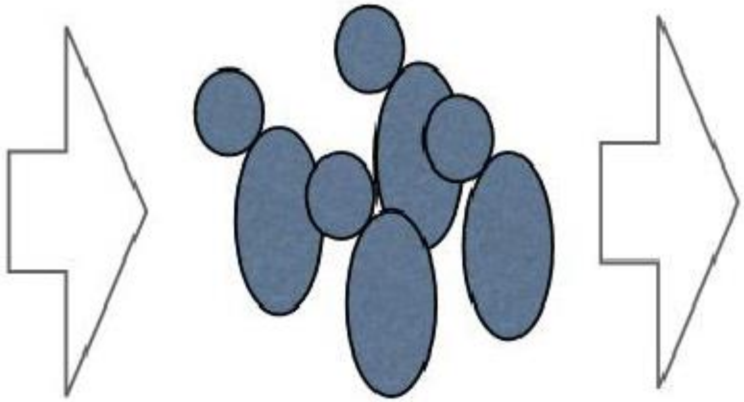
Alexander Sorokin
David Forsyth
CVPR Workshops 2008

Slides by Alexander Sorokin

Building datasets

6000 images
from flickr.com

Annotators



amazonmechanical turk
Artificial Artificial Intelligence

Is there an Indigo bunting in the image?

100s of
training images

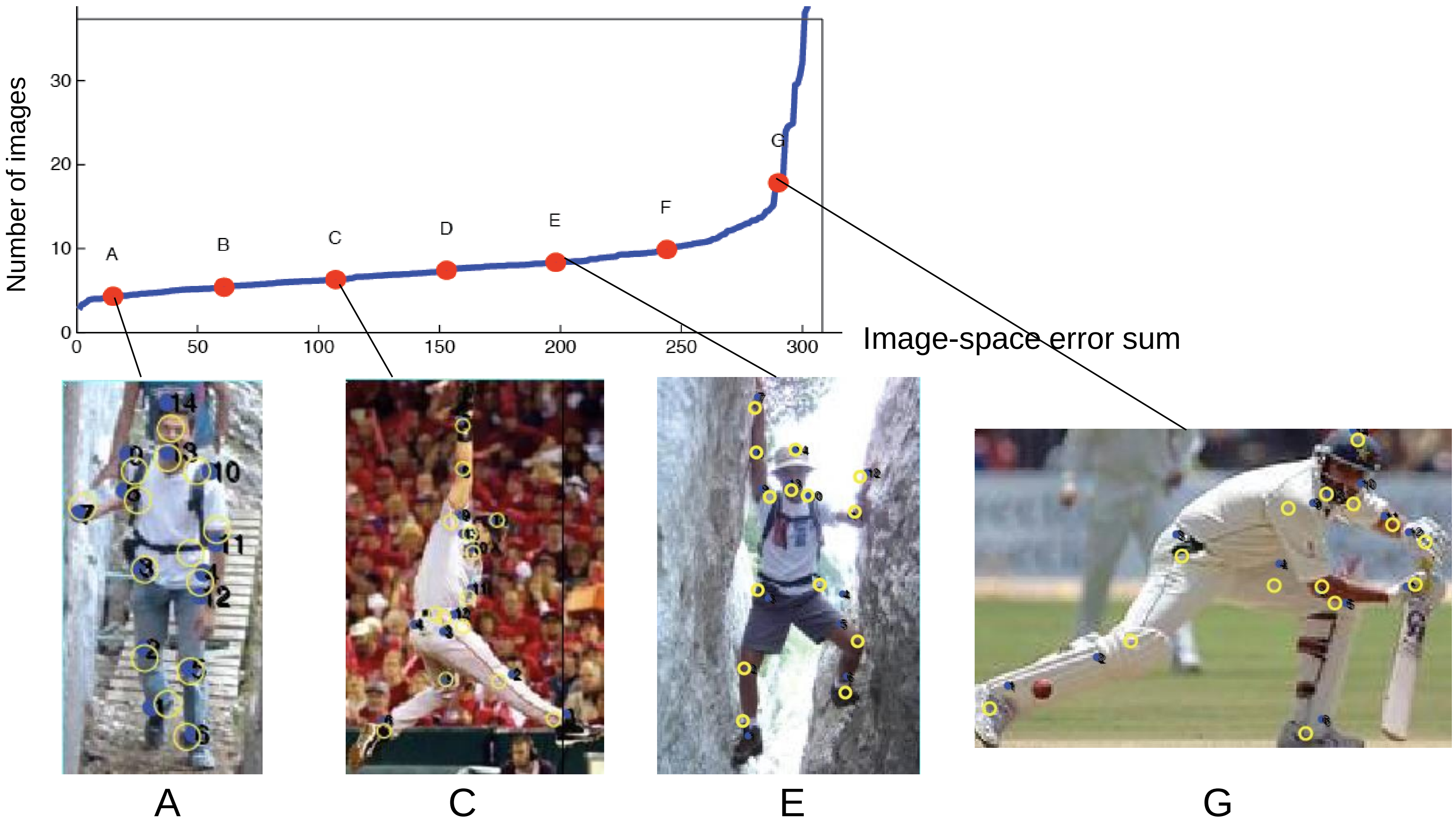
Slide credit: Welinder et al

Issues

- Quality?
  - How good is it?
  - How to be sure?
- Price?
  - Trade off between throughput and cost
    - *NOT* as much of a trade off with quality
  - Higher pay can actually attract scammers

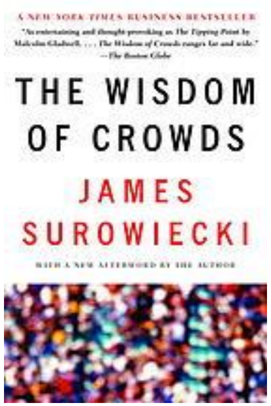
Annotation quality

How much agreement is there on ‘ground truth’ and turker-labeled joint positions?
Points must agree within 5-10 pixels on 500x500 image.



Yellow ring = ‘ground truth’; Blue circle = human labeled

Ensuring Annotation Quality



- Consensus in multiple annotations
  "Wisdom of the Crowds"
  Not enough on its own, but widely used

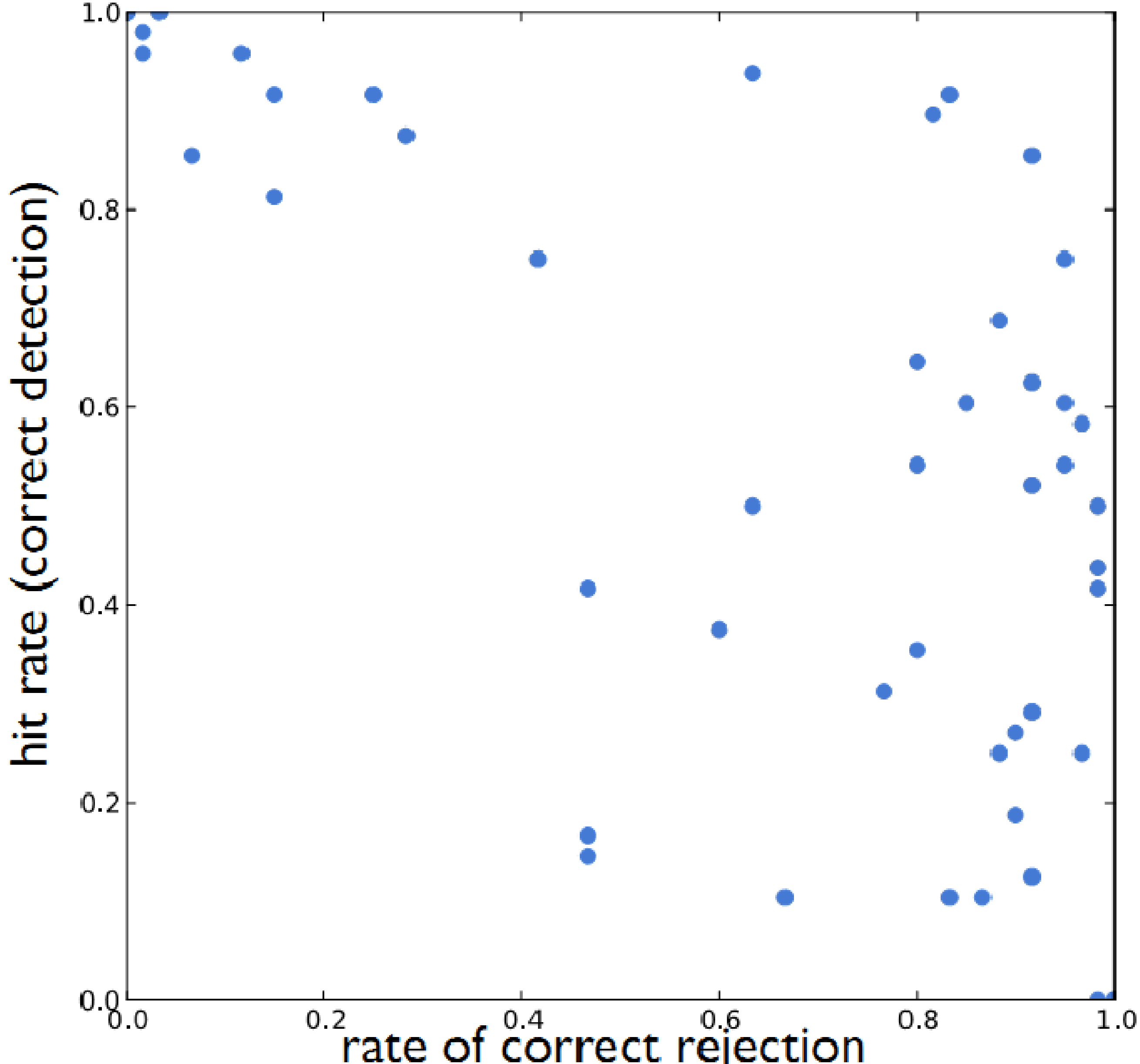
- Gold Standard / Sentinel
  - Special case: qualification exam

  Widely used & important. Find good annotators; keep them honest.

- Grading Tasks
  - A second tier of workers who grade others

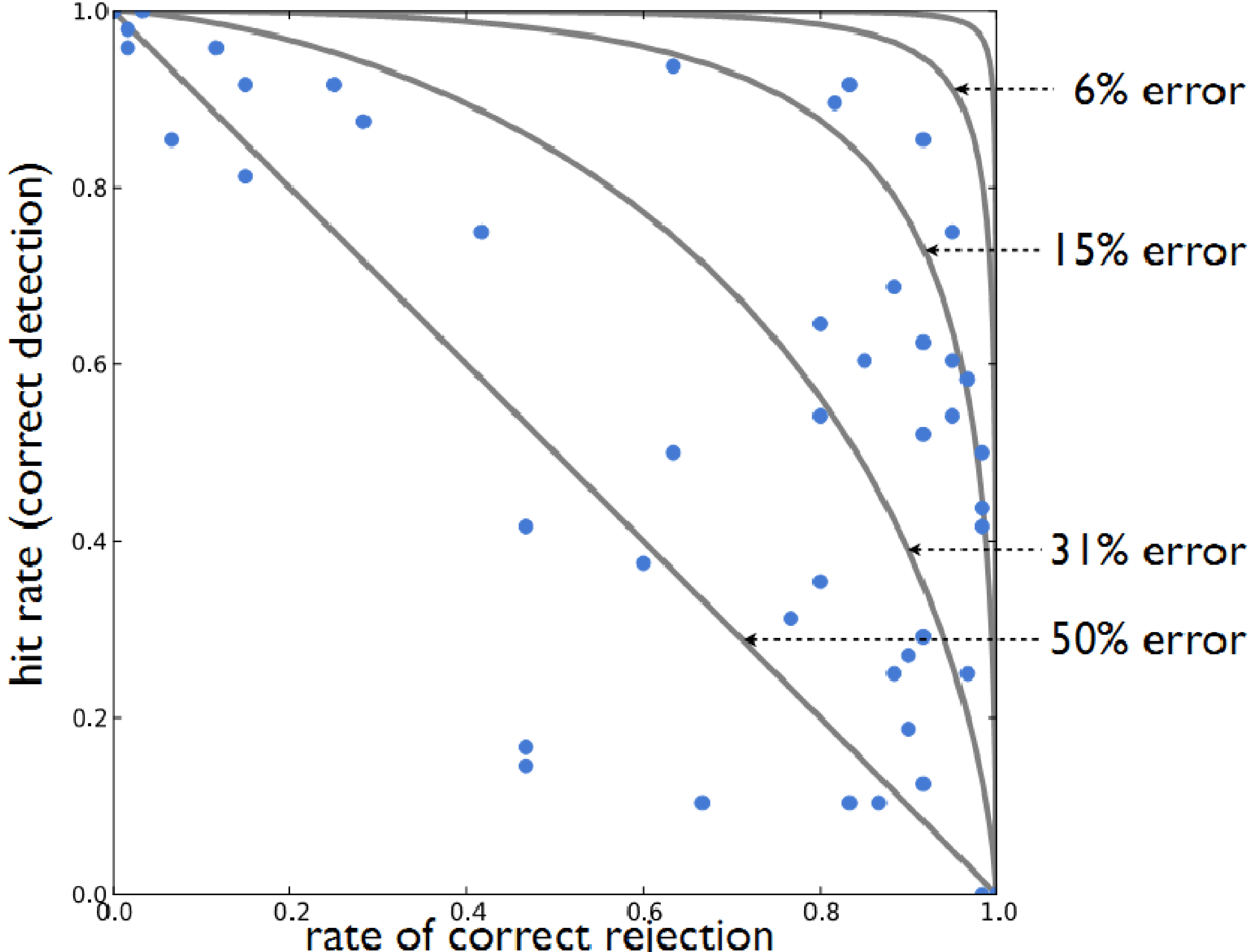
  Not widely used

Task: Find the Indigo Bunting



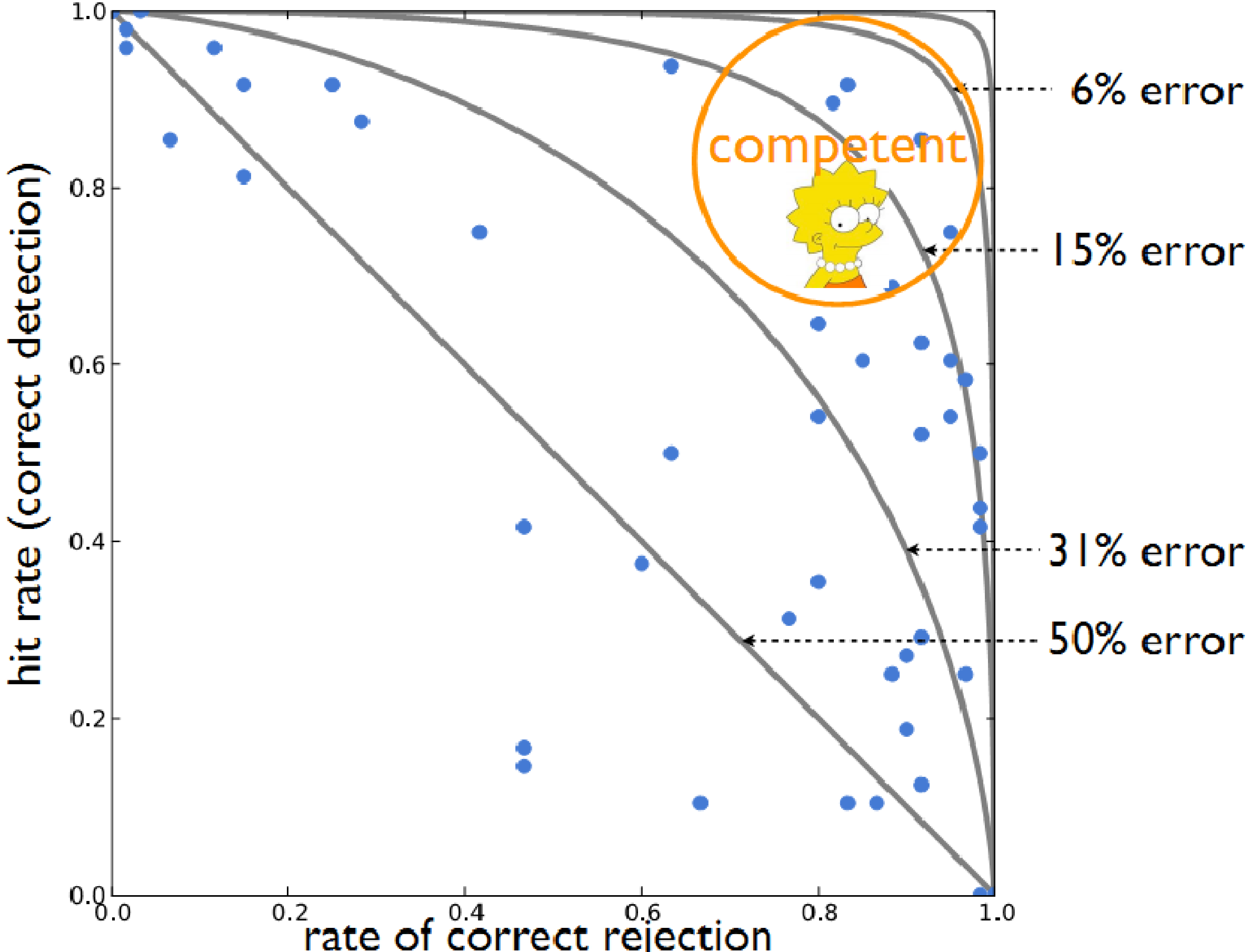
Slide credit: Welinder et al

Task: Find the Indigo Bunting



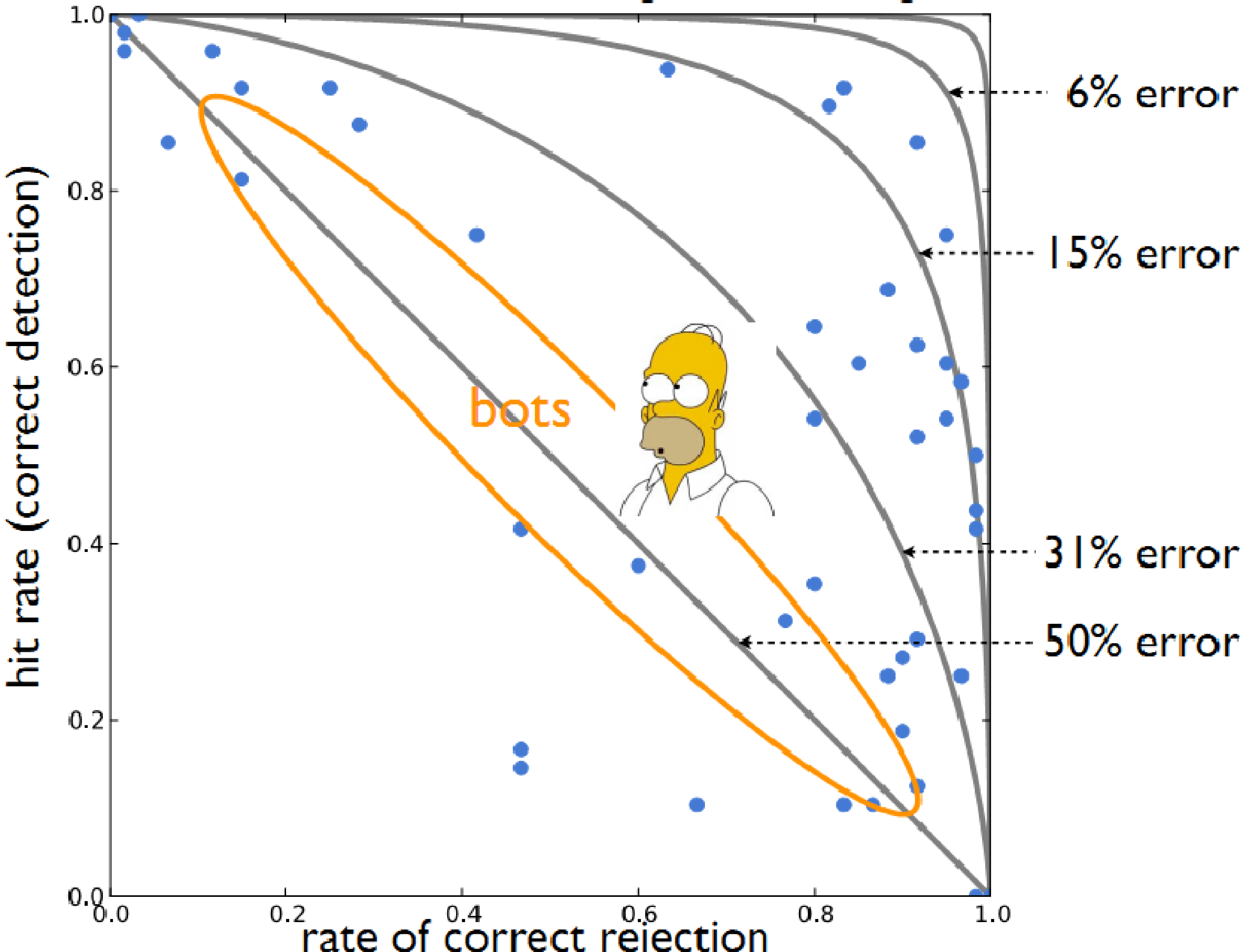
Slide credit: Welinder et al

Task: Find the Indigo Bunting



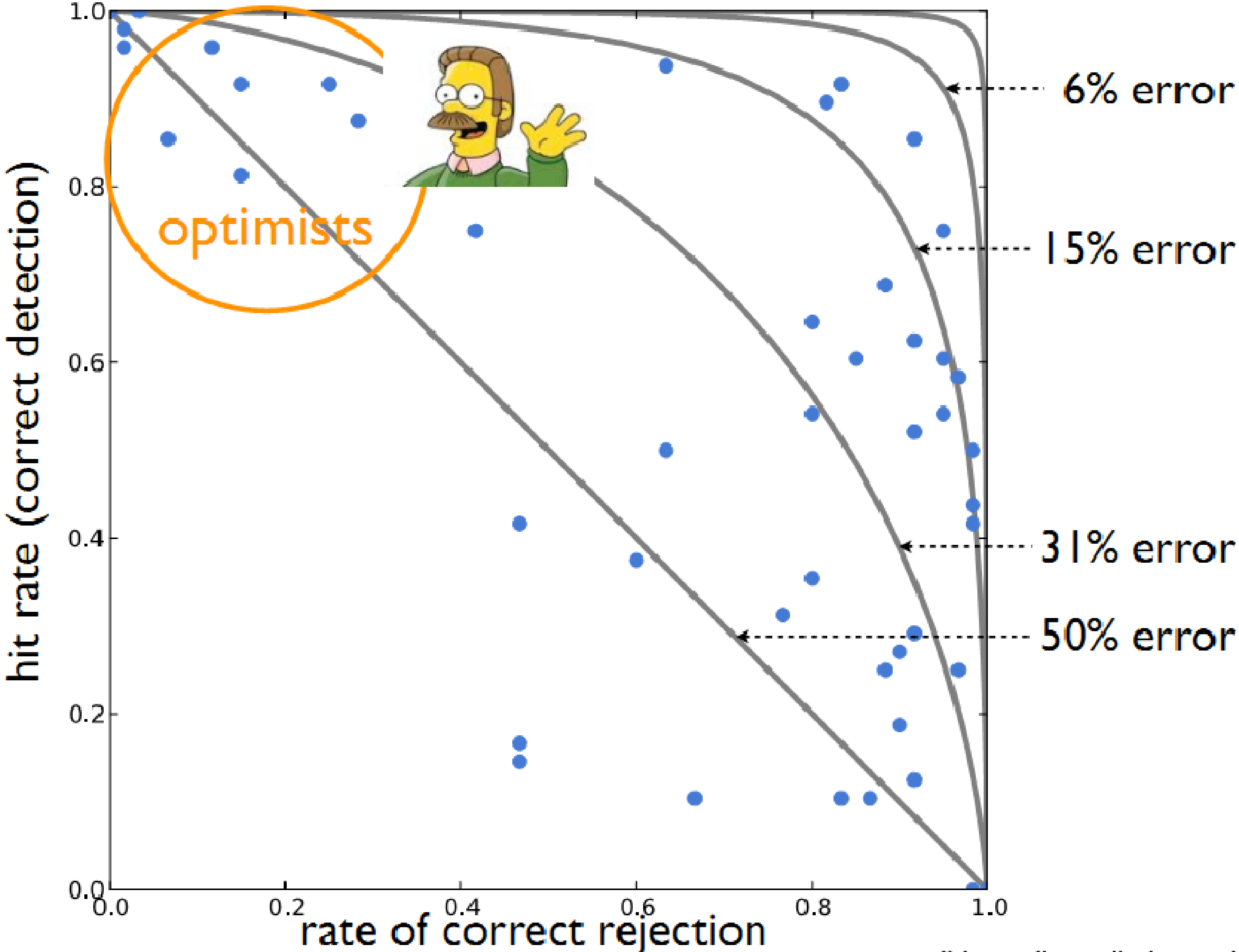
Slide credit: Welinder et al

Task: Find the Indigo Bunting



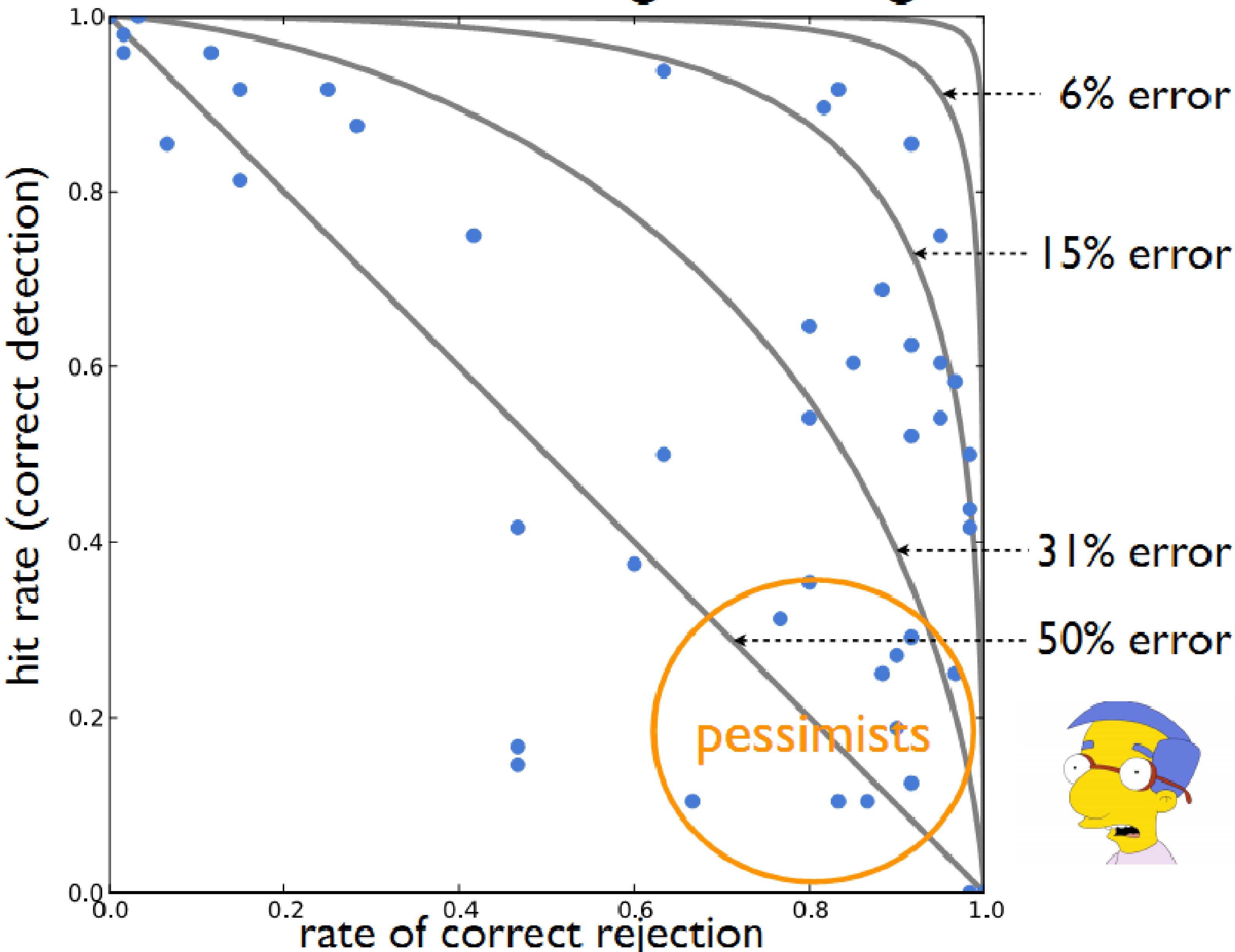
Slide credit: Welinder et al

Task: Find the Indigo Bunting



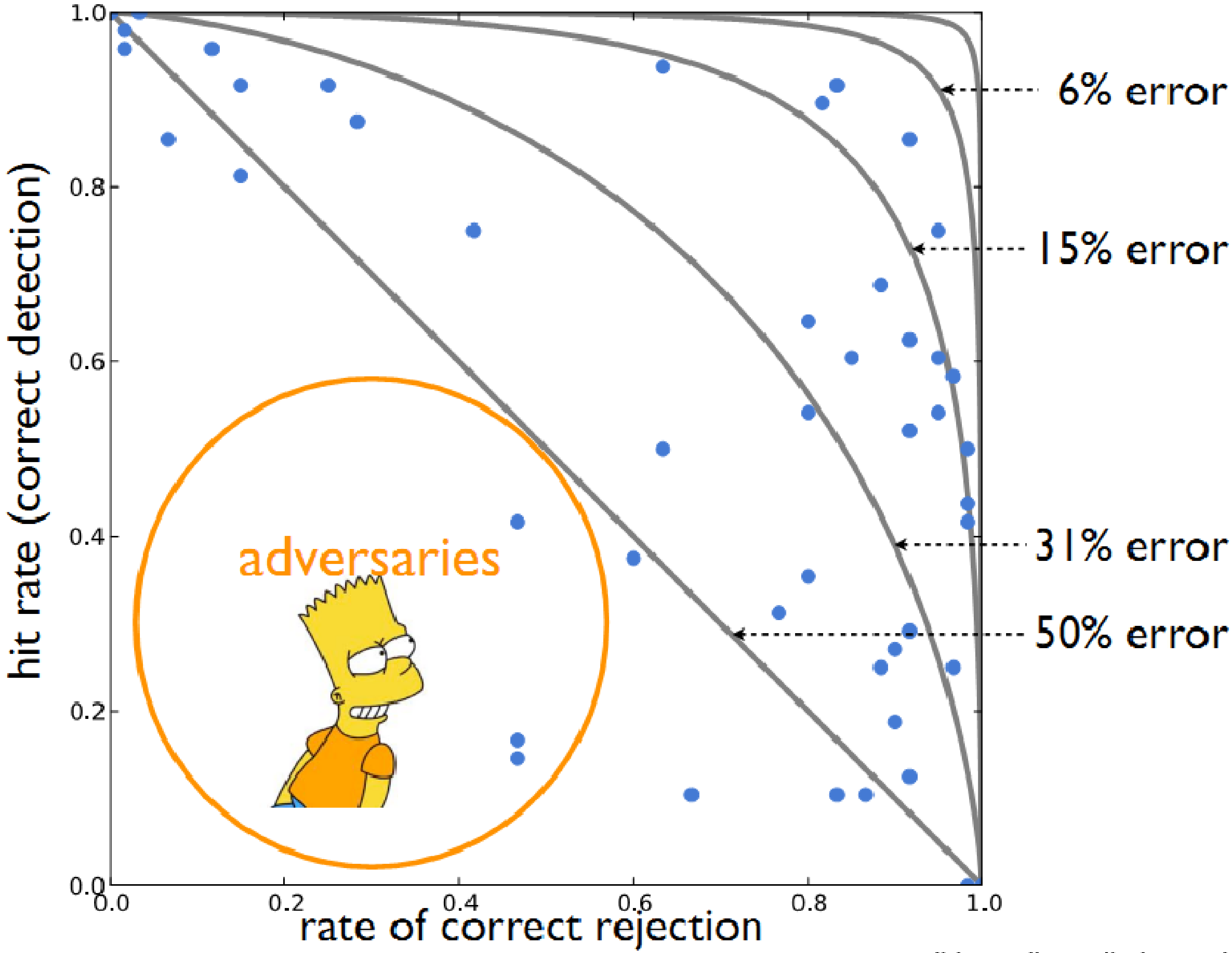
Slide credit: Welinder et al

Task: Find the Indigo Bunting



Slide credit: Welinder et al

Task: Find the Indigo Bunting



Slide credit: Welinder et al

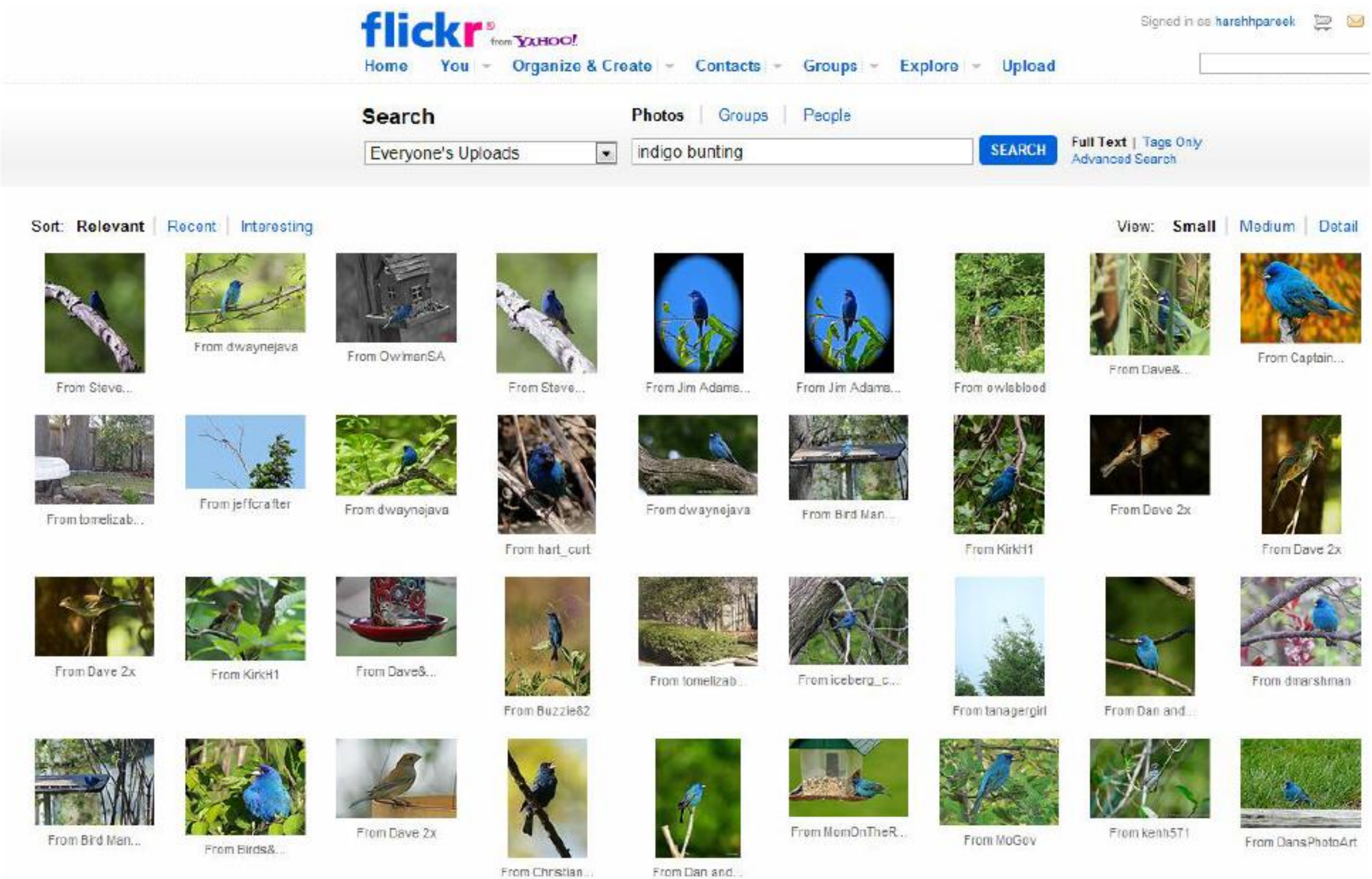


Image credit: Flickr.com

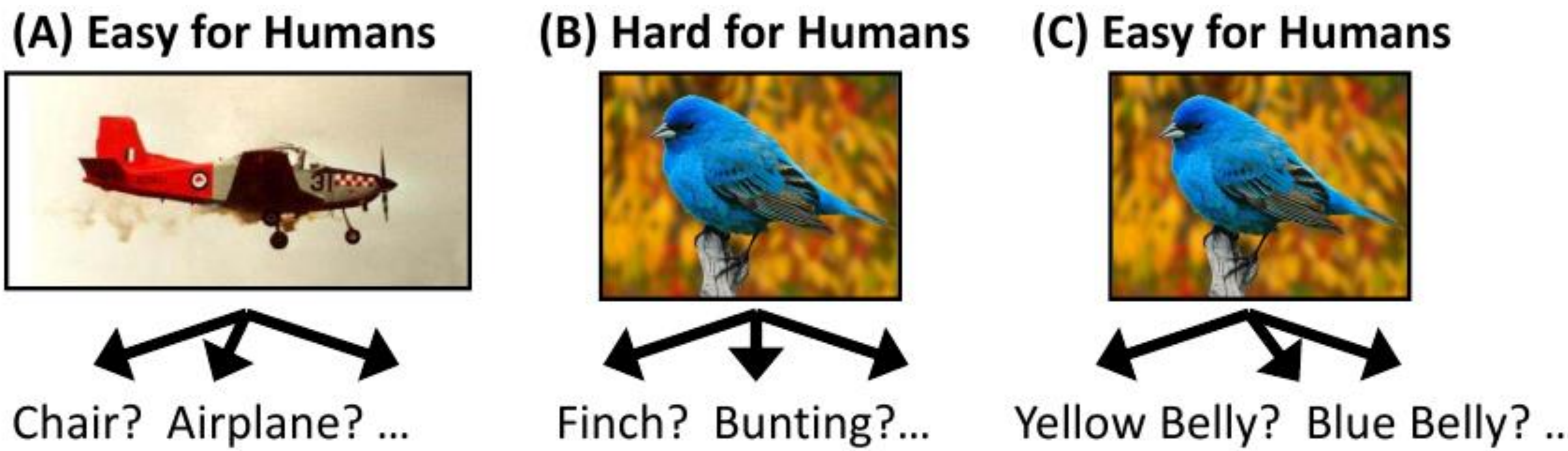
Visual Recognition with Humans in the Loop

Steve Branson, Catherine Wah, Florian Schroff, Boris Babenko, Peter Welinder, Pietro Perona, Serge Belongie

Part of the Visipedia project

Slides from Brian O'Neil

Introduction:



Computers starting to get good at this.

If it's hard for humans, it's probably too hard for computers.

Semantic feature extraction difficult for computers.

Combine strengths to solve this problem.

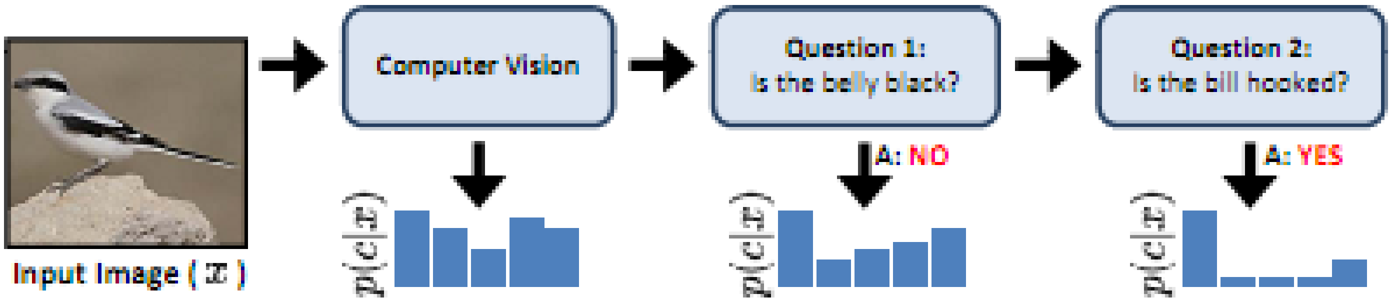
The Approach: What is progress?

- Supplement visual recognition with the human capacity for visual feature extraction to tackle difficult (fine-grained) recognition problems.
- Typical progress is viewed as increasing data difficulty while maintaining full autonomy

- Reduction in human effort on difficult data.

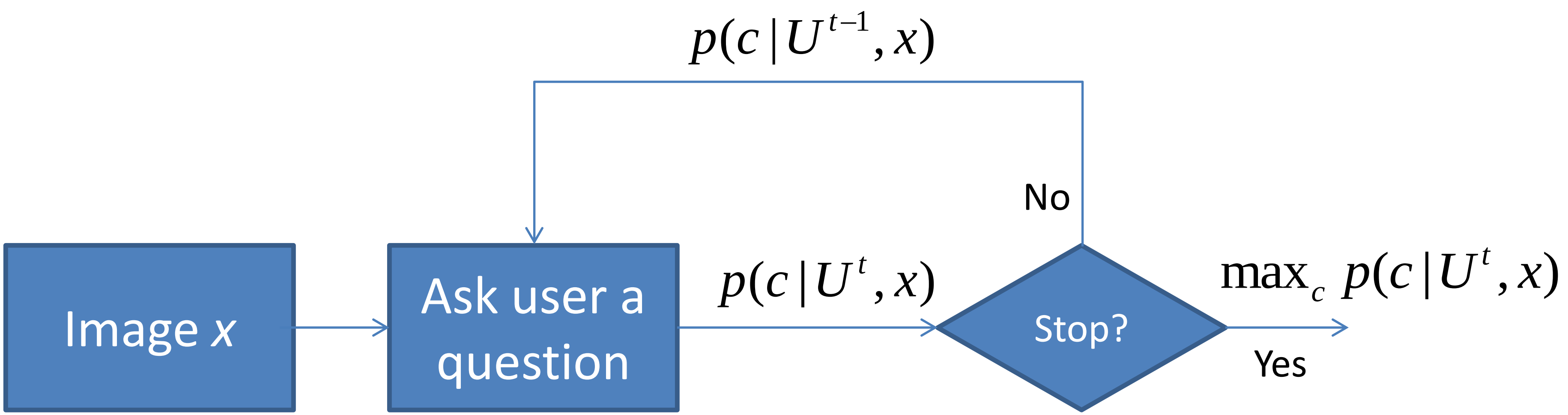
The Approach: 20 Questions

- Ask the user a series of discriminative visual questions to make the classification.



Which 20 questions?

- At each step, exploit the image itself and the user response history to select the most informative question to ask next.



Which question to ask?

- The question that will reduce entropy the most, taking into consideration the computer vision classifier confidences for each category.

Some definitions:

$Q = \{q_1 \ldots q_n\}$ • Set of possible questions

$a_i \in A_i$ • Possible answers to question *i*

$r_i \in V$ • Possible confidence in answer *i* (Guessing, Probably, Definitely)

$u_i = (a_i, r_i)$ • User response

U^t • History of user responses at time *t*

Question selection

- Seek the question that gives the maximum information gain (entropy reduction) given the image and the set of previous user responses.

$$I\left(c;u_i \mid x,U^{t-1}\right)=\sum_{u_i \in A_i \times V} \boxed{p\left(u_i \mid x,U^{t-1}\right)}\left(\boxed{H\left(c \mid x,u_i \cup U^{t-1}\right)}-\boxed{H\left(c \mid x,U^{t-1}\right)}\right)$$

Probability of obtaining
Response u_i given the image
And response history

Entropy when
response is
Added to history

Entropy before response
is added.

where $$H\left(c \mid x,U^{t-1}\right)=-\sum_{c=1}^{C} p\left(c \mid x,U^{t-1}\right)\log p\left(c \mid x,U^{t-1}\right)$$

Incorporating vision

- Bayes Rule
- A visual recognition algorithm outputs a probability distribution across all classes that is used as the prior.
- A posterior probability is then computed based on the probability of obtaining a particular response history given each class.

$$p(c \mid x, U) = \eta p(U \mid c, x) p(c \mid x) = \eta p(U \mid c) p(c \mid x)$$

Modeling user responses

- Assume that the questions are answered independently.

$$p\left(U^{t-1} \mid c\right) = \prod_{i}^{t-1} p\left(u_i \mid c\right)$$

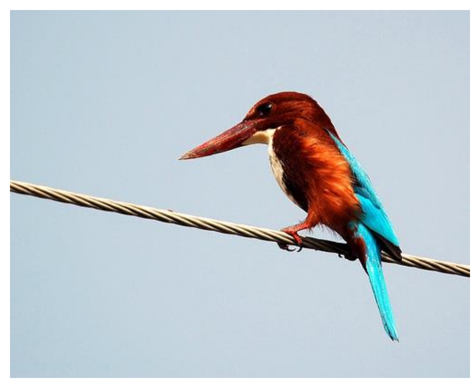
Required for posterior computation

$$p\left(u_i \mid x, U^{t-1}\right) = \sum_{c=1}^{C} p\left(u_i \mid c\right) p\left(c \mid x, U^{t-1}\right)$$

Required for information gain computation

The Dataset: Birds-200

- 6033 images of 200 species



Implementation

- Assembled 25 visual questions encompassing 288 visual attributes extracted from www.whatbird.com
- Mechanical Turk users asked to answer questions and provide confidence scores.

User Responses.

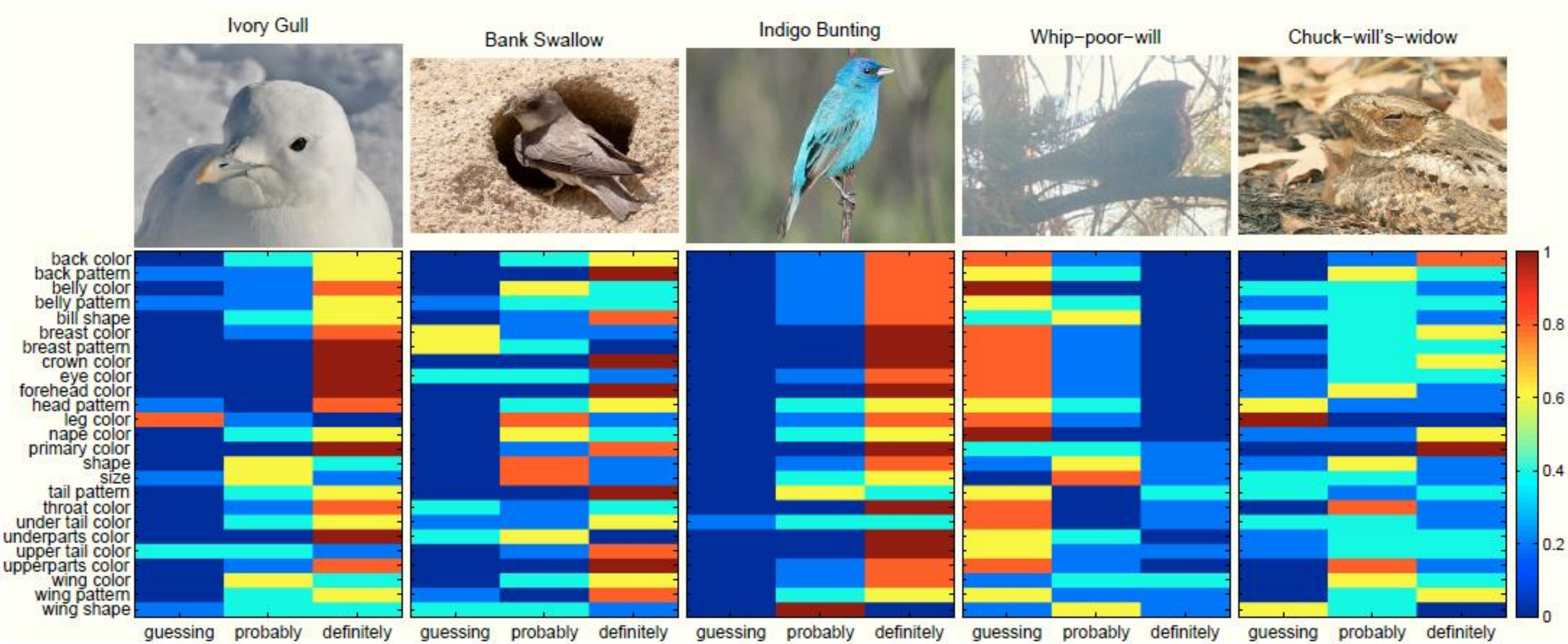
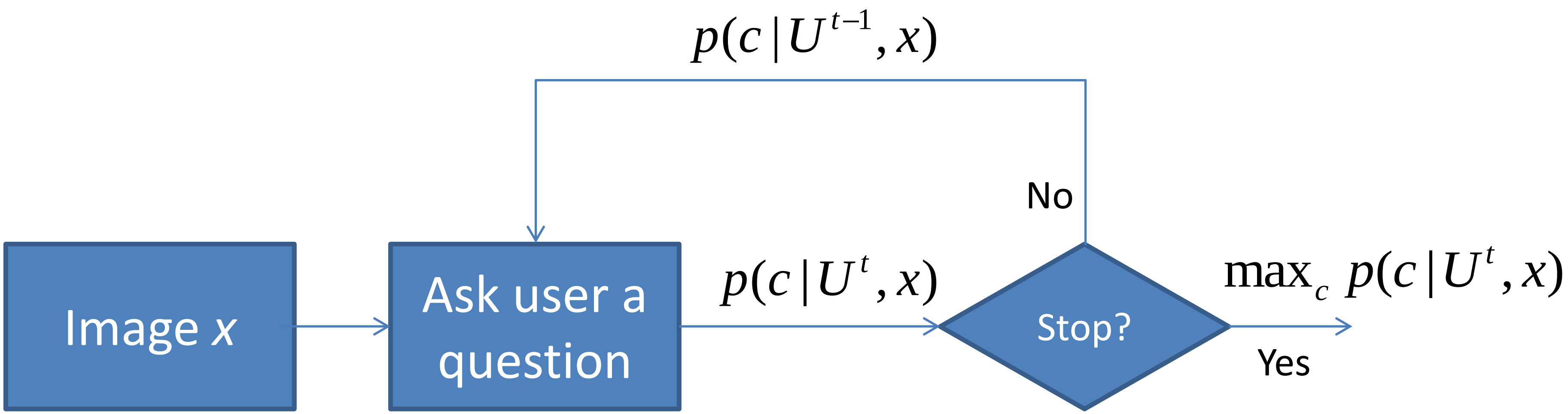


Fig. 4. Examples of user responses for each of the 25 attributes. The distribution over $\{Guessing, Probably, Definitely\}$ is color coded with blue denoting 0% and red denoting 100% of the five answers per image attribute pair.

Visual recognition

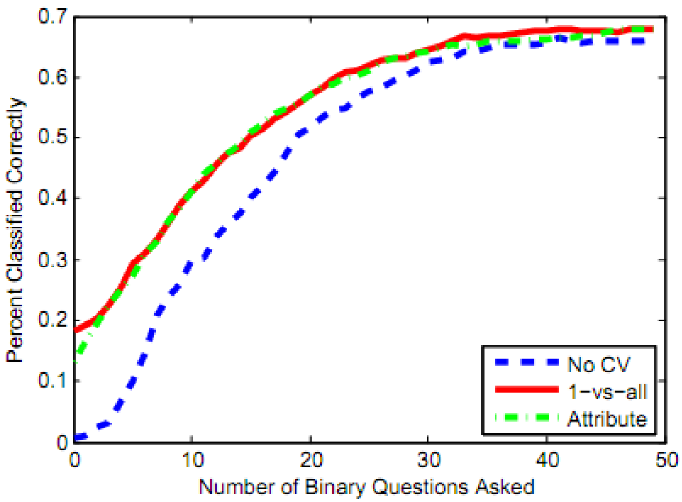
- Any vision system that can output a probability distribution across classes will work.
- Authors used Andrea Vedaldis's code.
  - Color/gray SIFT
  - VQ geometric blur
  - 1 v All SVM
- Authors added full image color histograms and VQ color histograms

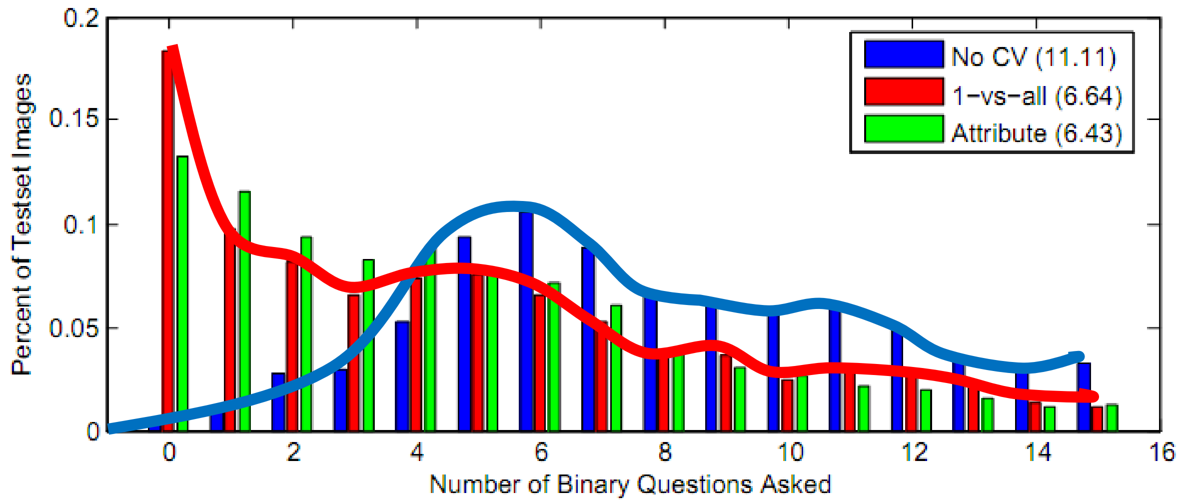
Experiments



- 2 Stop criteria:
  - Fixed number of questions – evaluate accuacy
  - User stops when bird identified – measure number of questions required.

Results





- Average number of questions to make ID reduced from 11.11 to 6.43
- Method allows CV to handle the easy cases, consulting with users only on the more difficult cases.

Key Observations

- Visual recognition reduces labor over a pure "20 Q" approach.
- Visual recognition improves performance over a pure "20 Q" approach. (69% vs 66%)
- User input dramatically improves recognition results. (66% vs 19%)

Strengths and weaknesses

- Handles very difficult data and yields excellent results.
- Plug-and-play with many recognition algorithms.
- Requires significant user assistance
- Reported results assume humans are perfect verifiers
- Is the reduction from 11 questions to 6 really that significant?

Hays