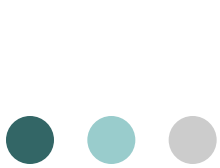


Microsoft Research Cambridge

Semantic Image Segmentation and Web-Supervised Visual Learning

Florian Schroff

Andrew Zisserman

University of Oxford, UK

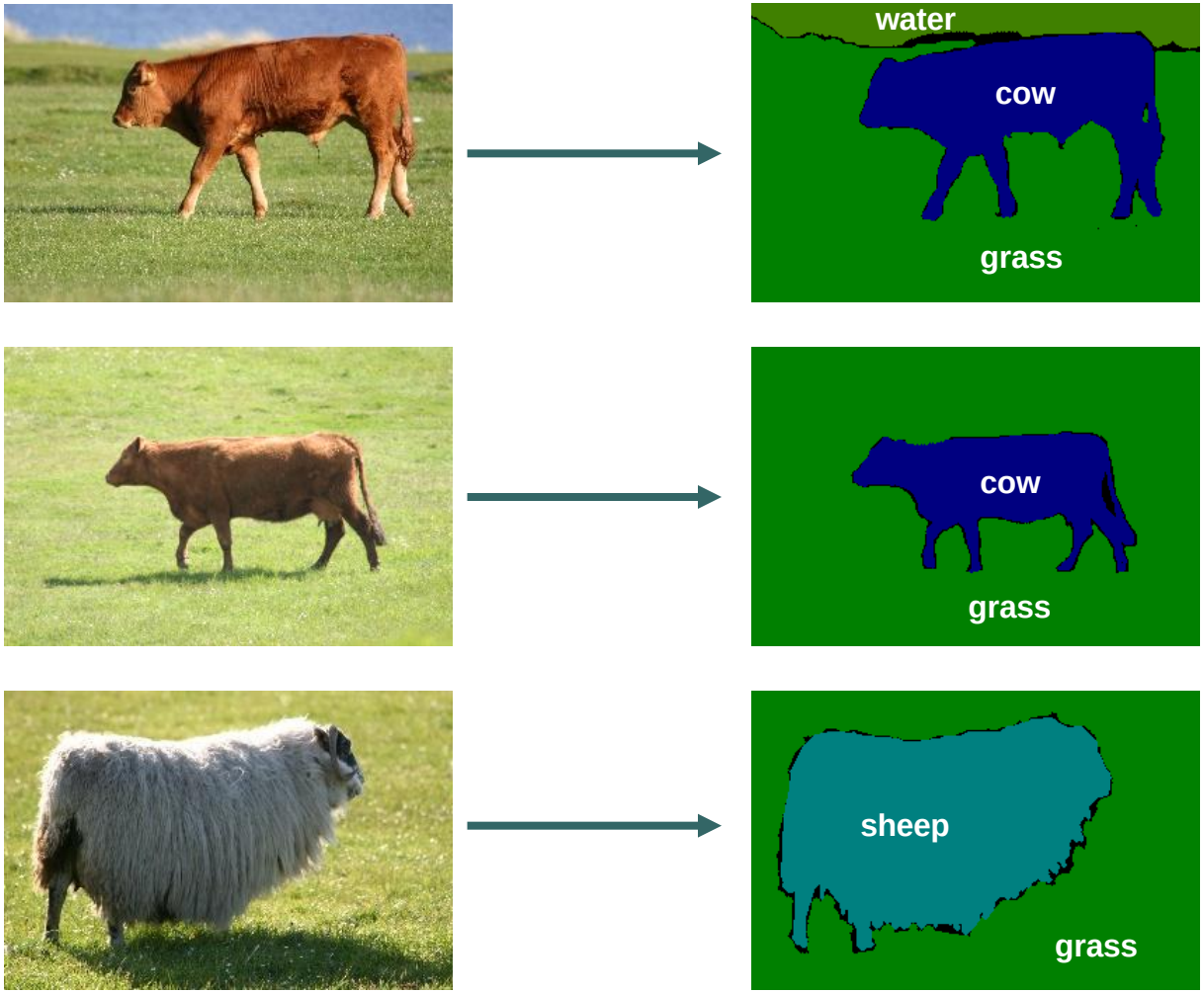
Antonio Criminisi

Microsoft Research Ltd, Cambridge, UK

Outline

- Part I: Semantic Image Segmentation
  - Goal: automatic segmentation into object regions
  - Texton-based Random Forest classifier
- Part II: Web-Supervised Visual Learning
  - Goal: harvest class specific images automatically
    - Use text & metadata from web-pages
    - Learn visual model
- Part III: Learn segmentation model from harvested images

Goal: Classification & Segmentation

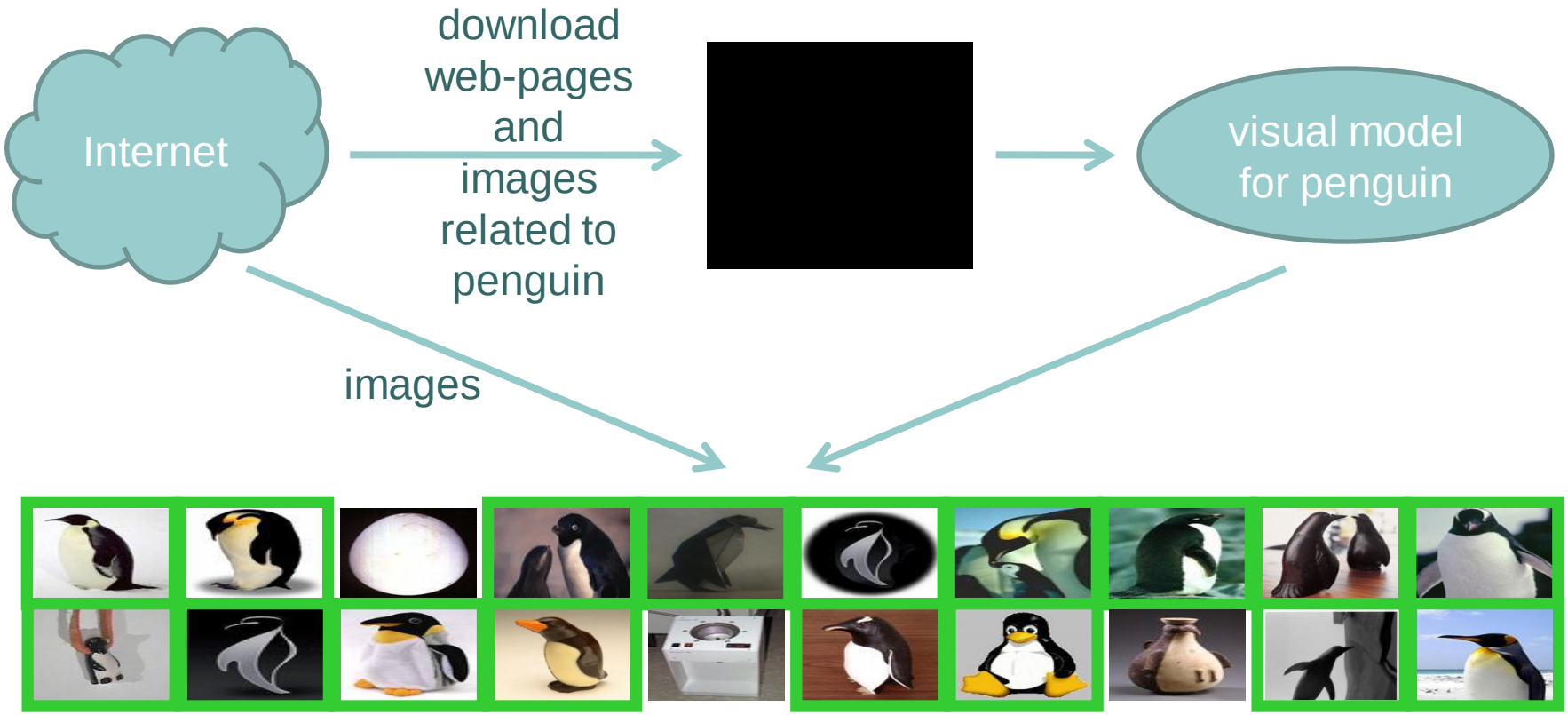


Image

Classification/Segmentation

Goal: Harvest images automatically

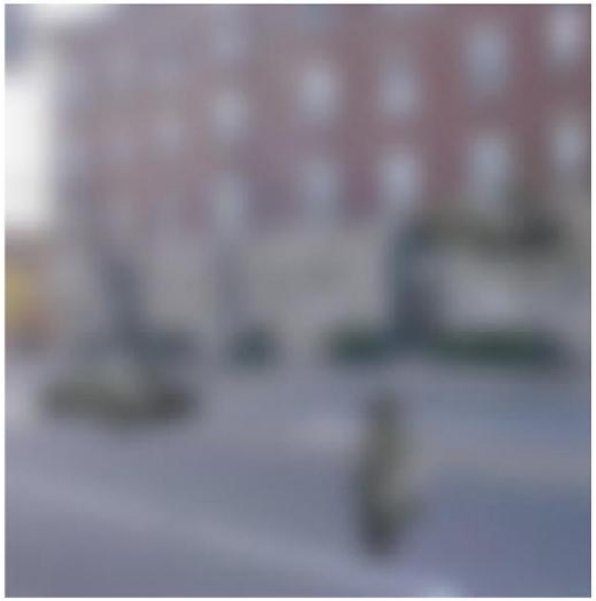
- Learn visual models w/o user interaction
- Specify object-class: e.g. penguin



Challenges in Object Recognition

- **Intra-class** variations: appearance differences/similarities among objects of the same class
- **Inter-class** variations: appearance differences/similarities between objects of different classes
- **Lighting and viewpoint**

Importance of Context

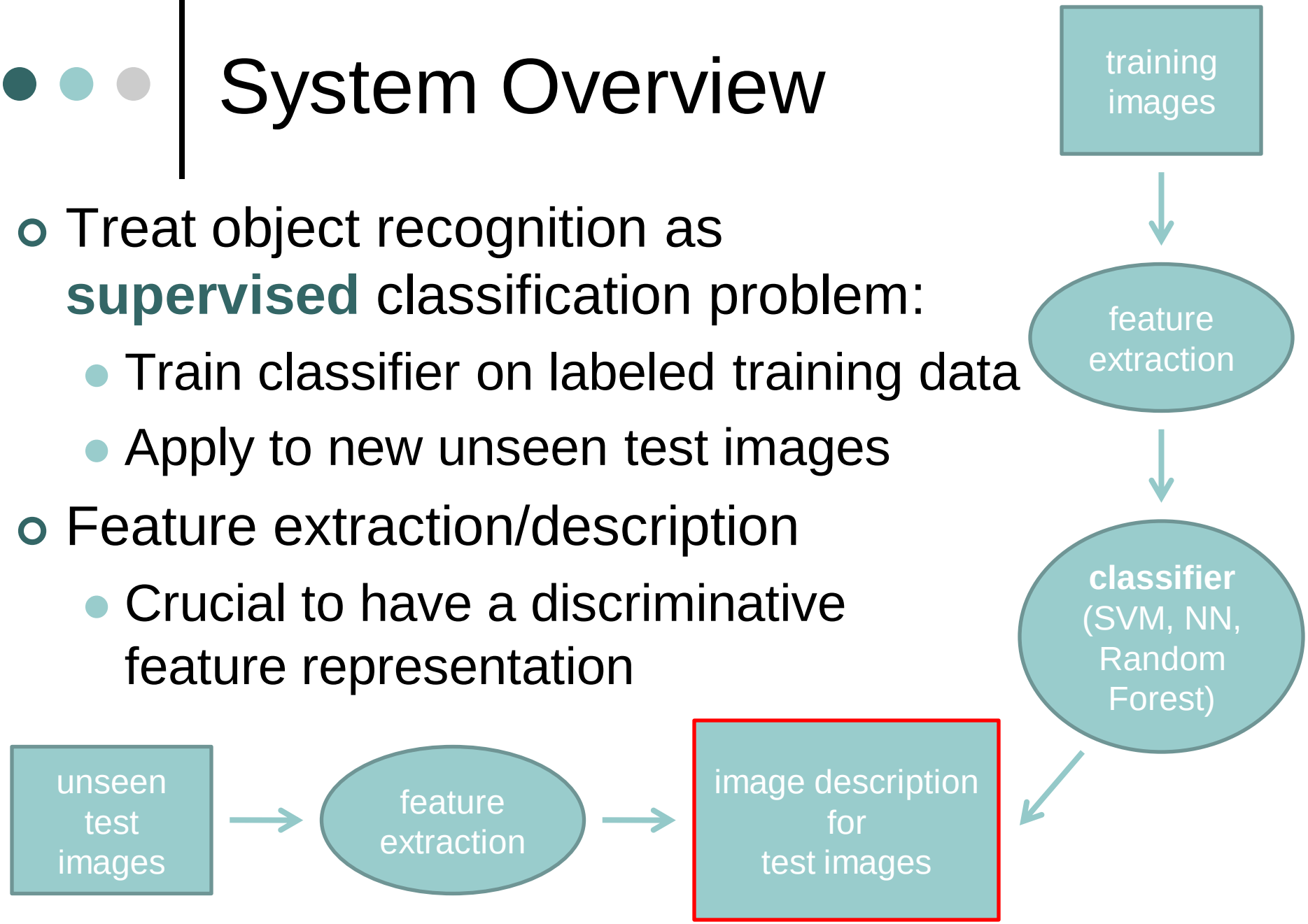


Oliva and Torralba (2007)

- Context often delivers important cues
- Human recognition heavily relies on context
- In ambiguous cases context is crucial for recognition

System Overview

- Treat object recognition as **supervised** classification problem:
  - Train classifier on labeled training data
  - Apply to new unseen test images
- Feature extraction/description
  - Crucial to have a discriminative feature representation



Part I: Image Segmentation

- Supervised classification problem:
  - Classify each pixel in the image

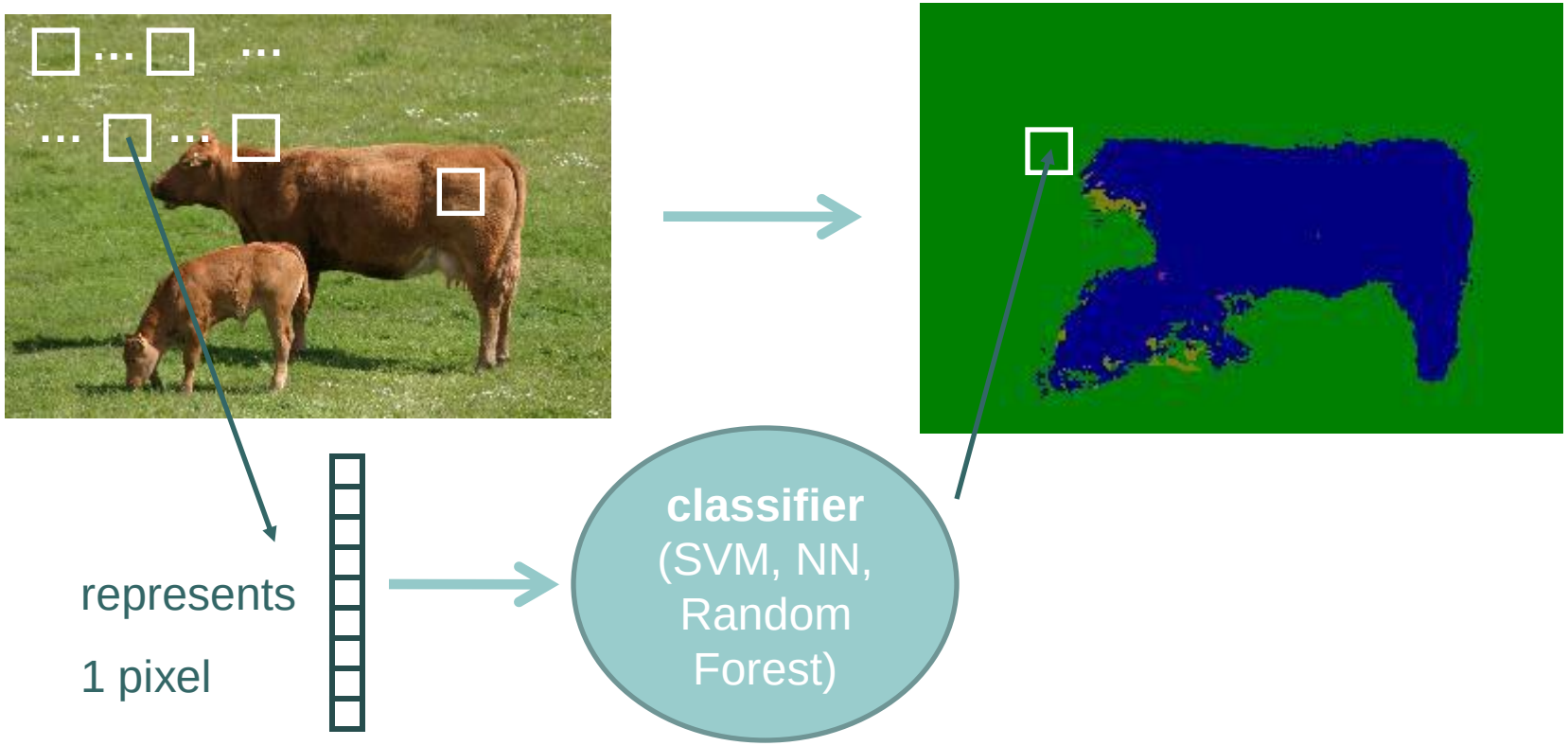


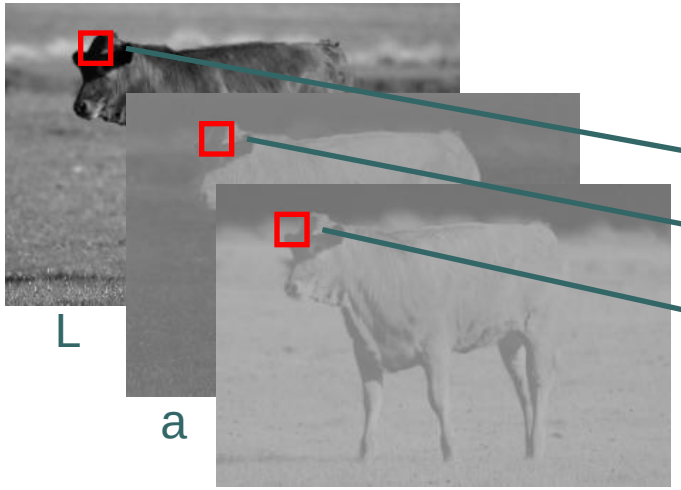
Image Segmentation

- Introduction to **textons** and **single-class histogram** models (SCHM)
- Comparison of nearest neighbour (NN) and Random Forest
- Show strength of Random Forests to combine multiple features

Background: Feature Extraction

Lab

colour-space

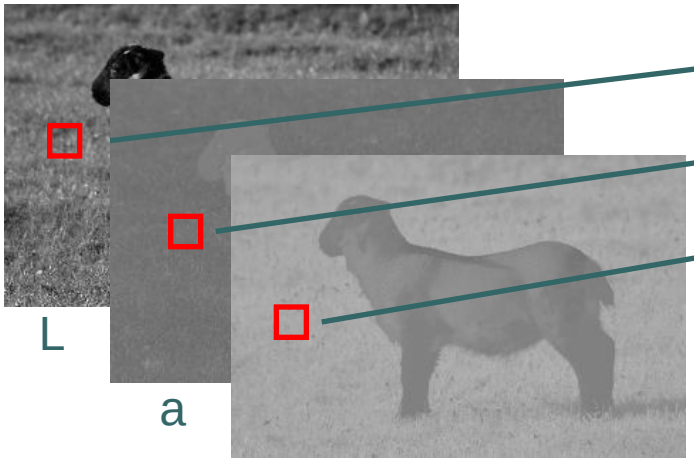


5x5 pixels neighbourhood

repr. 1 pixel

Lab

colour-space

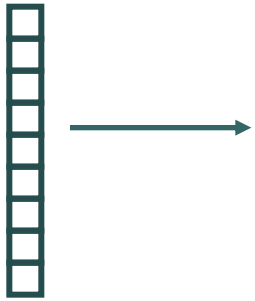


repr. 1 pixel

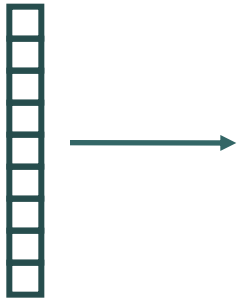
3x5x5=75 dim. feature vectors per pixel

Background: Texton Vocabulary

feature extraction

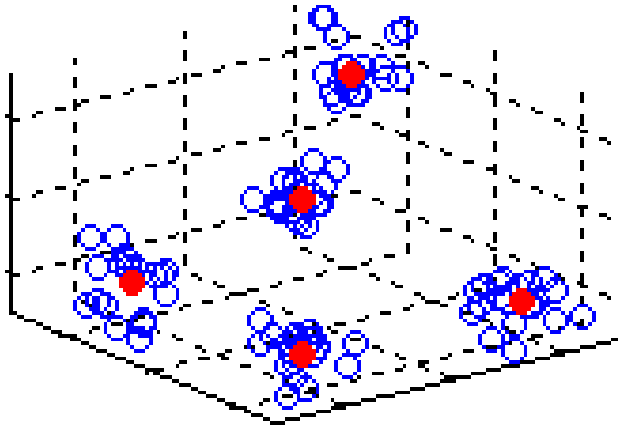


feature extraction



K-Means

75 dim.

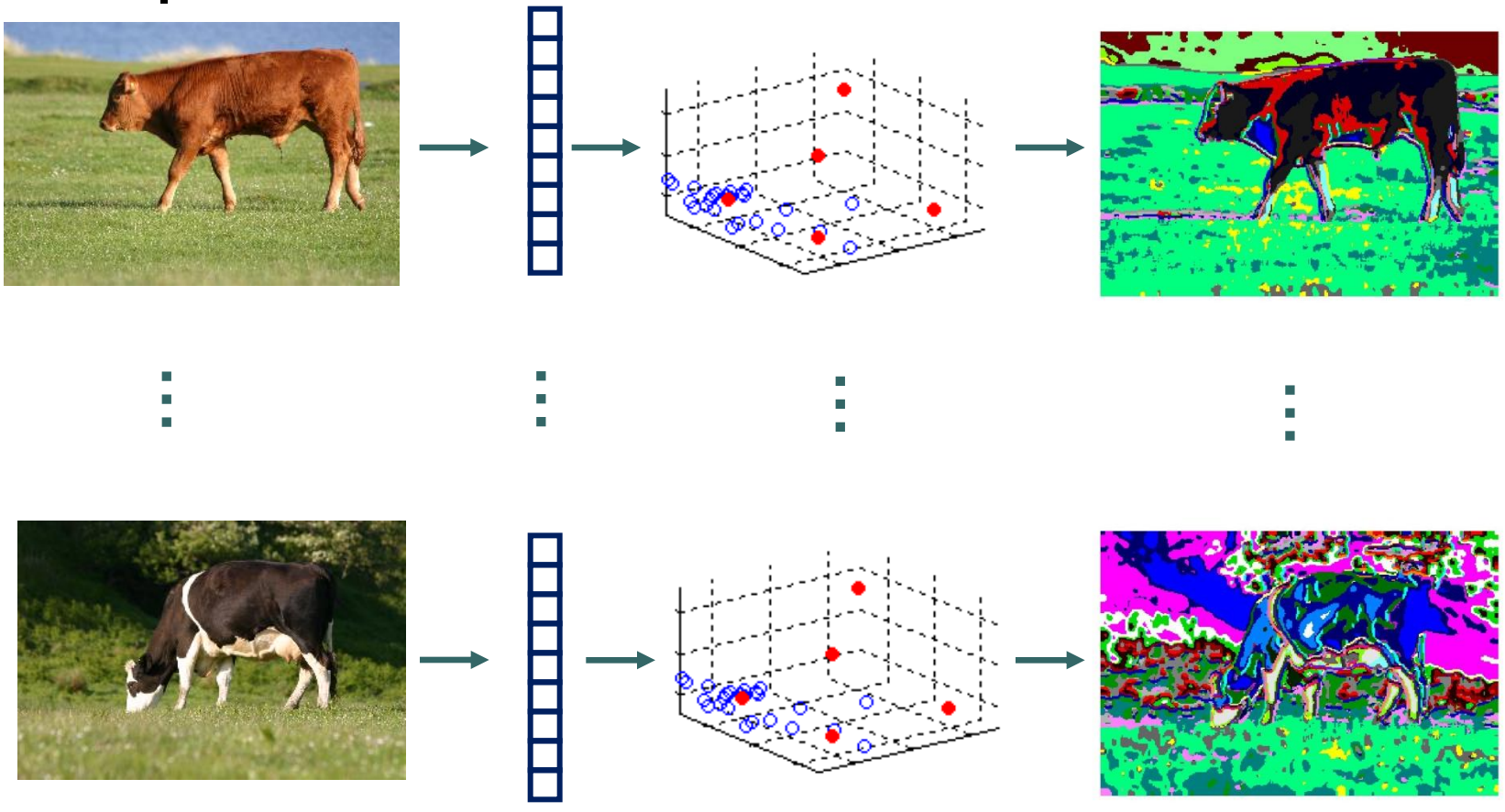


Training Images

Feature vectors
75 dim.

Texton vocabulary
V textons (#cluster centres)
V = K in K-means

Map Features to Textons



Training Images

Feature Vectors per pixel

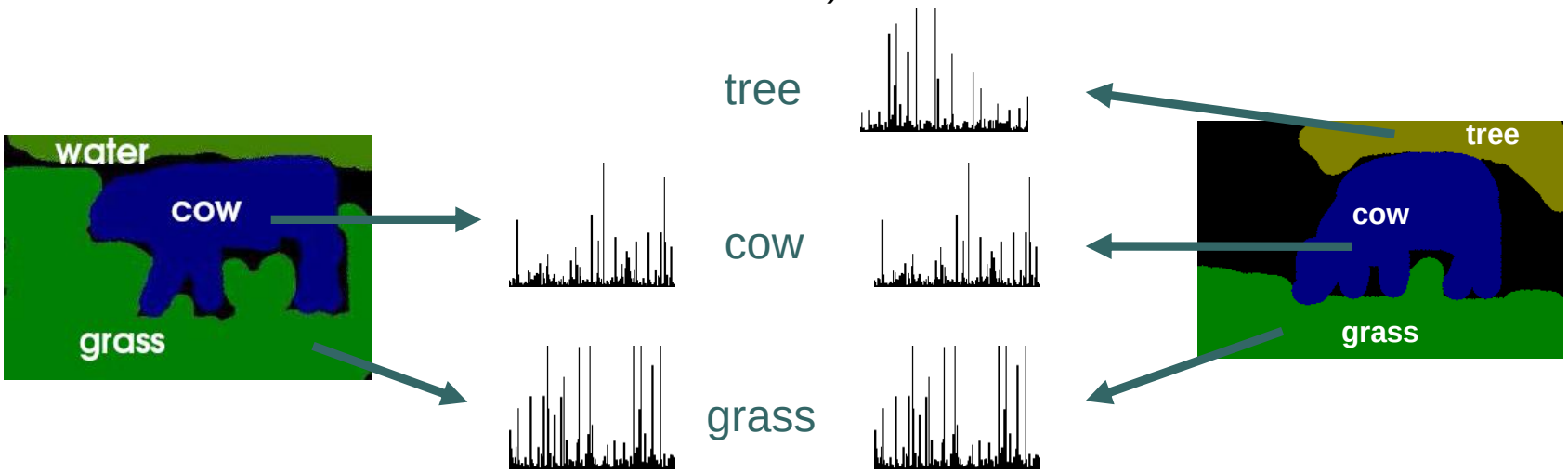
Map to textons (pre-clustered)

Resulting texton-maps

Texton-Based Class Models

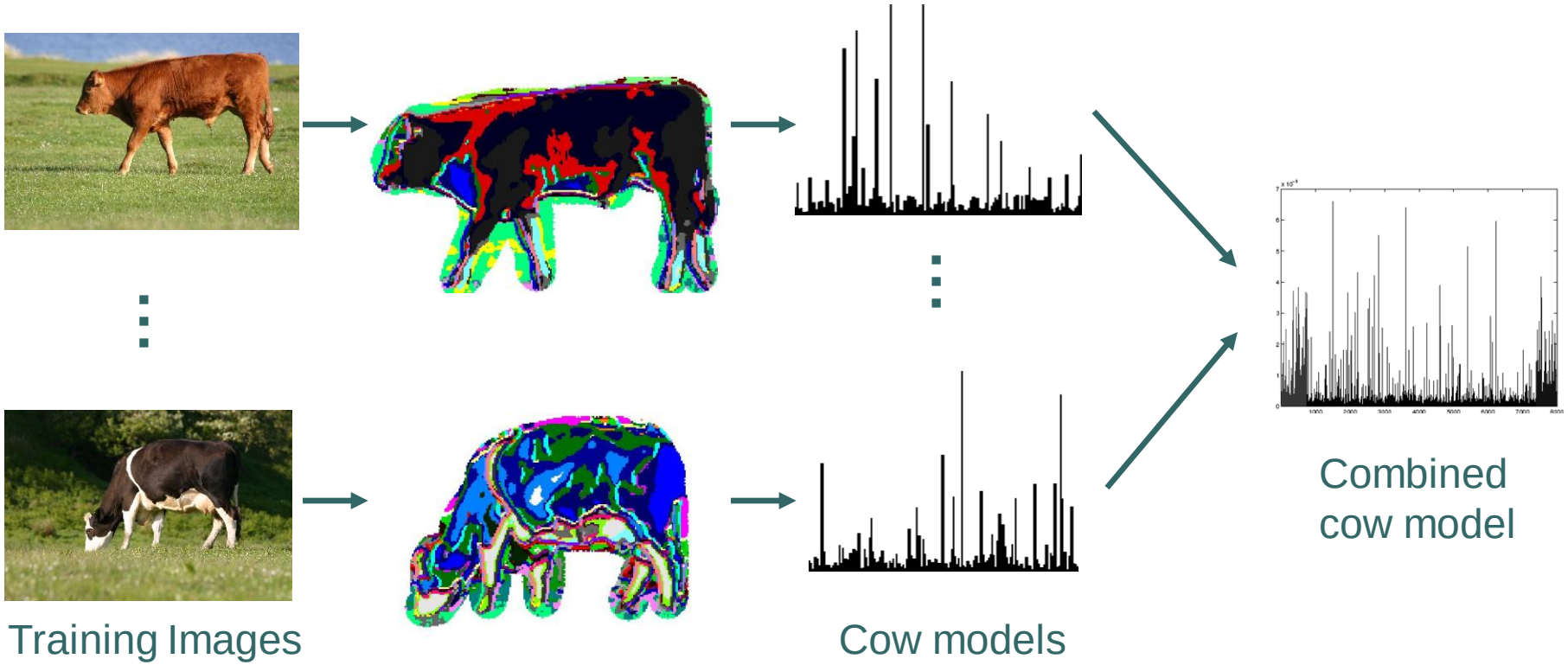
- Learn texton histograms given class regions
- Represent each class as a set of texton histograms
- Commonly used for texture classification (region ⇔ whole image)

(Leung&Malik ICCV99, Varma&Zisserman CVPR03, Cula&Dana SPIE01, Winn et al. ICCV05)



Exemplar based class models (Nearest Neighbour or SVM classifier)

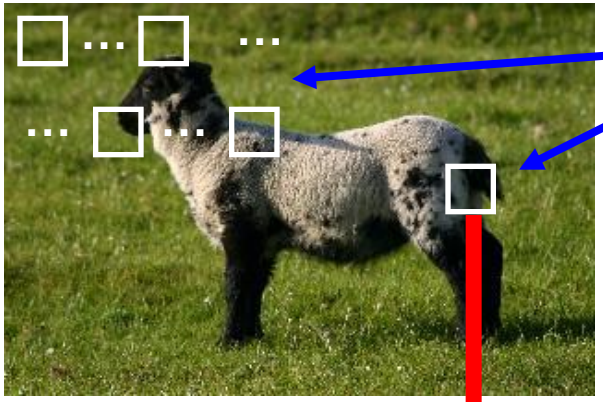
Single Histogram Class Model Histograms (SHCM)



Model each class by a **single** model! (Schroff *et al.* ICVGIP 06)
(rediscovered by Boiman, Shechtman, Irani CVPR 08)

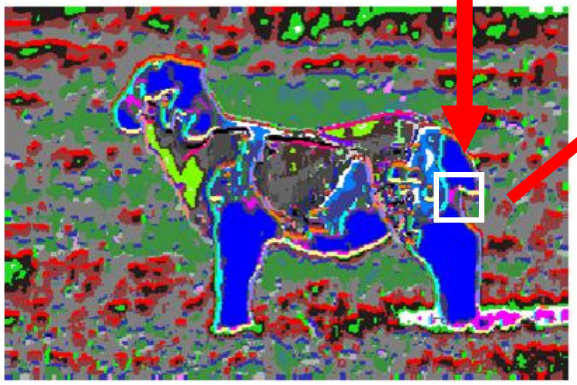
(SHCM improve generalization and speed)

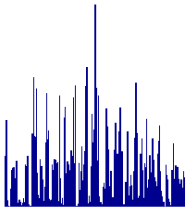
Pixelwise Classification (NN)



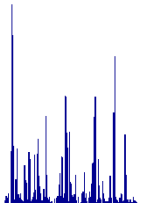
fixed size sliding window

assign textons





h



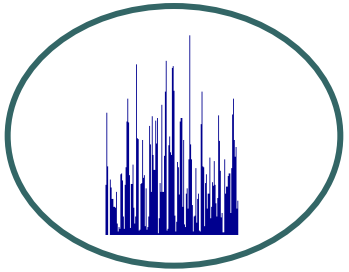
Cow model

$$\sum h_i \log\left(\frac{h_i}{q_i}\right)$$

=

Kullback-Leibler Divergence

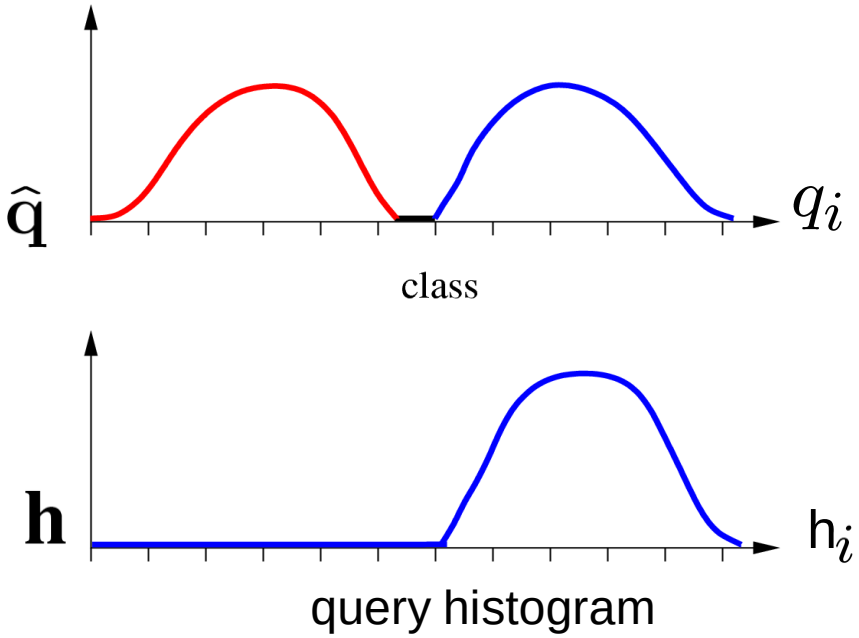
KL is better suited than χ^2



Sheep model

Kullback-Leibler Divergence: Testing

- KL does not penalize zero bins in the test histogram which are non-zero in the model histogram
- Thus, KL is better suited for single-histogram class models, which have many non-zero bins due to different class appearances
- This better suitability was shown by our experiments



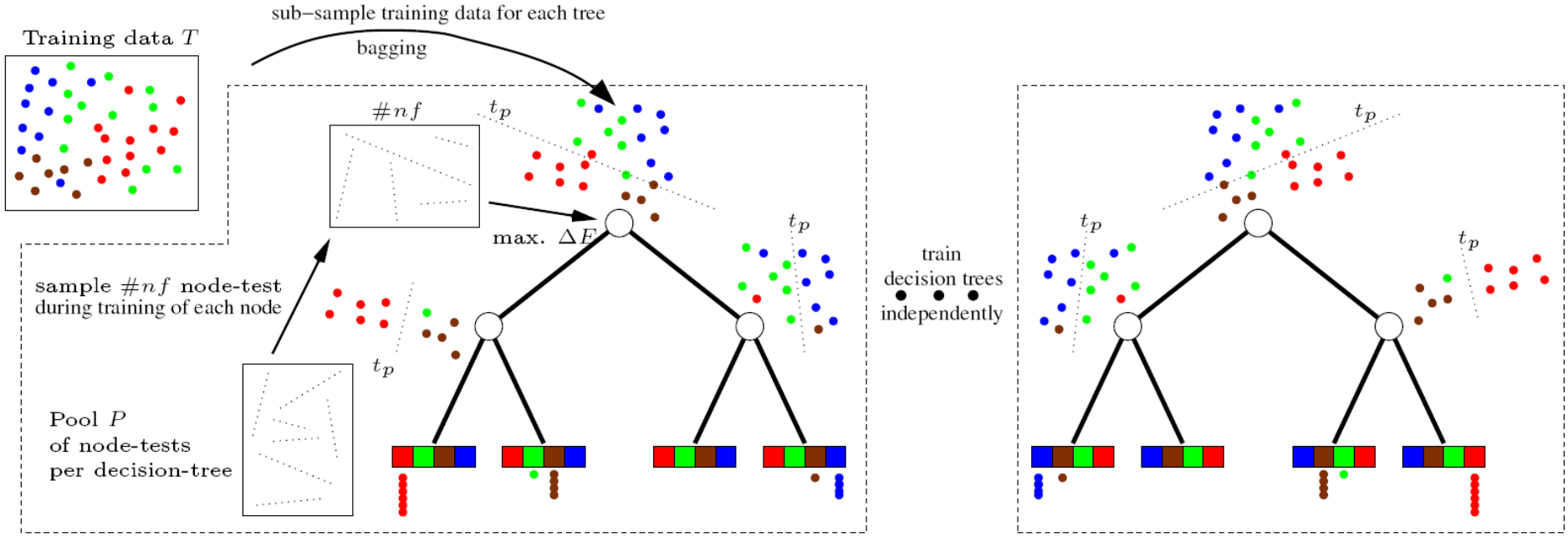
$$D_{KL}(\mathbf{h}||\mathbf{q}) = \sum h_i \log(\frac{h_i}{q_i})$$

Random Forest: Intro

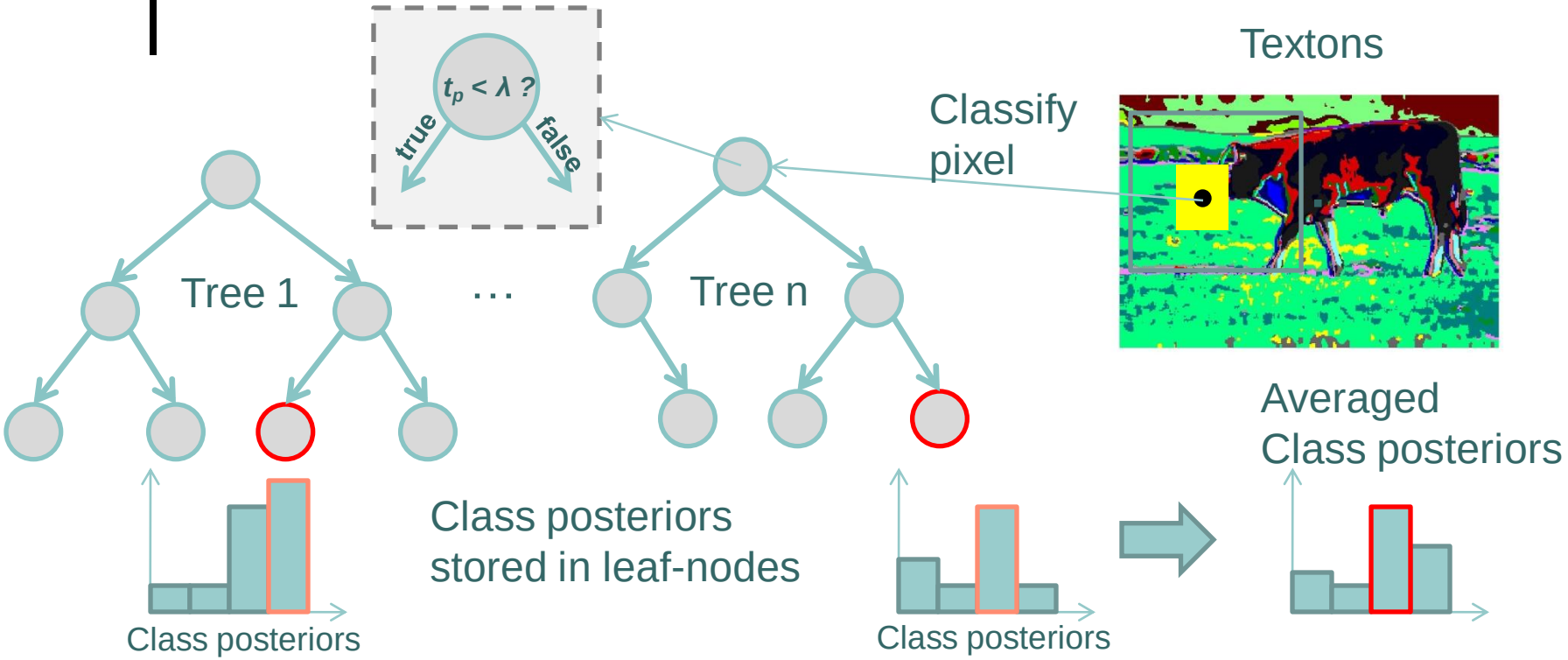
Combine
Single Histogram Class Model
and
Random Forest

Random Forest (Training)

- During training each node “selects” the feature from a precompiled feature pool that optimizes the information gain



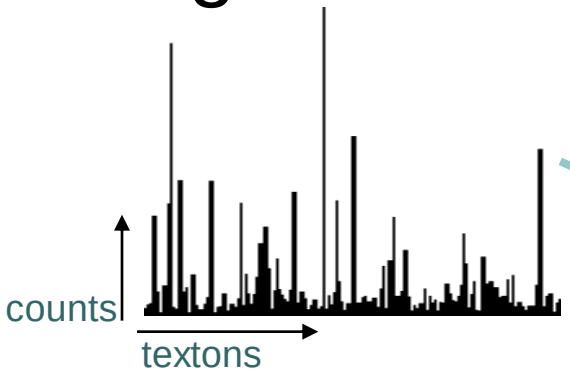
Random Forests (Testing)



- Combination of independent decision trees
- Emperical class posteriors in leaf nodes are averaged
  - Kleinberg, Stochastic Discrimination 90
  - Amit & Geman, Neural Computation 97; Breiman 01
  - Lepetit & Fua, PAMI06; Winn et al, CVPR06; Moosman et al., NIPS06

Single Histogram Class Model: Nearest Neighbour vs. node-tests

h test histogram
q^i class model histogram



Histogram: Sheep model

Combine to node-test

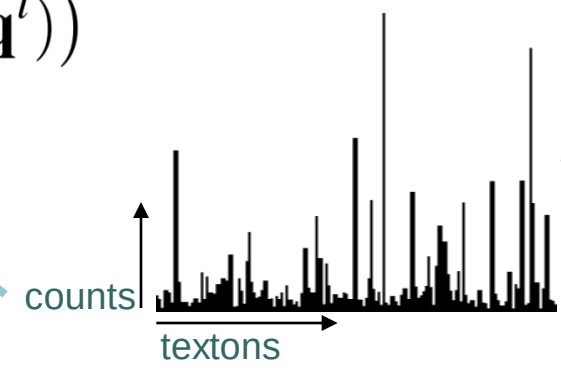
$$KL(\mathbf{h}_r||\mathbf{q}^i) < KL(\mathbf{h}_r||\mathbf{q}^j)$$

$$\mathbf{w}^{i,j} = \log\left(\frac{\mathbf{q}^i}{\mathbf{q}^j}\right)$$

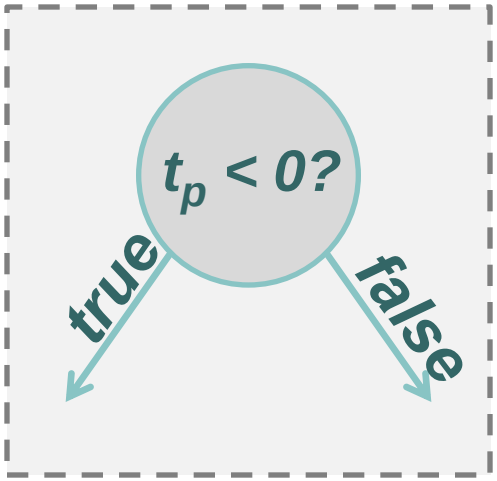
$$t_p < 0 \text{ with } t_p = \mathbf{w}^{i,j} \cdot \mathbf{h}_r$$

Nearest Neighbour

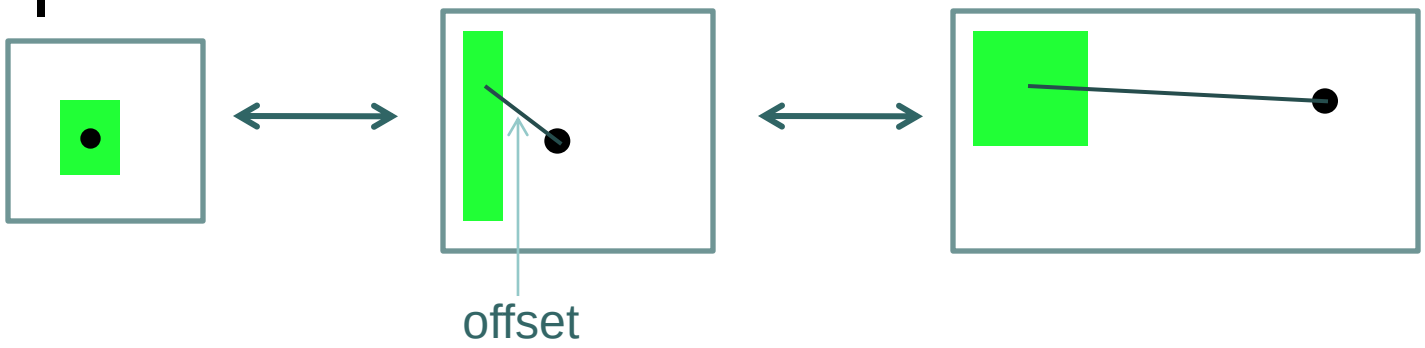
$$\hat{\mathbf{q}^i} = \arg\min_{\mathbf{q}^i}\left(KL(\mathbf{h}||\mathbf{q}^i)\right)$$



Histogram: Cow model



Flexible, learnt rectangles

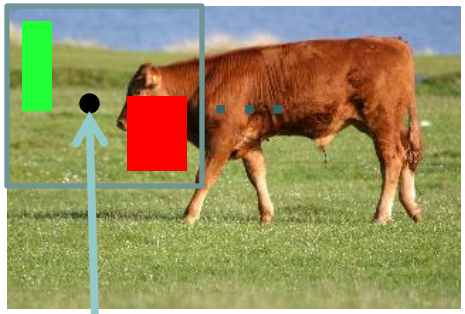


window size\offset	0	10	20	30
25:25	76.2%	78.8%	81.0%	82.2%
10:35	77.7%	79.3%	80.8%	82.0%

- Learning of offset and rectangle shapes/sizes, as well as the channels improves performance

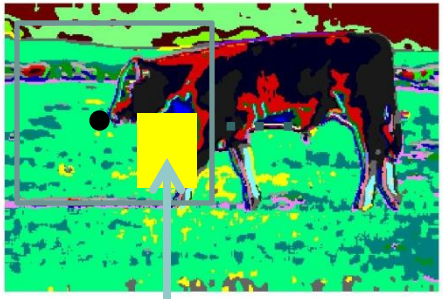
More Feature Types

RGB



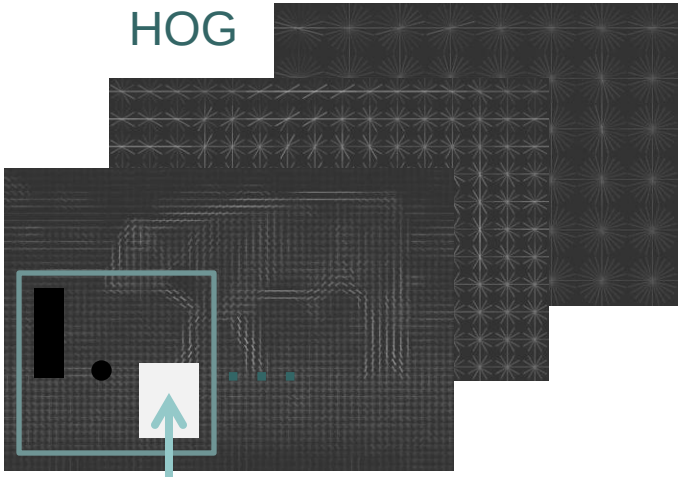
Pixel to be classified

Textons



Weighted sum of textons

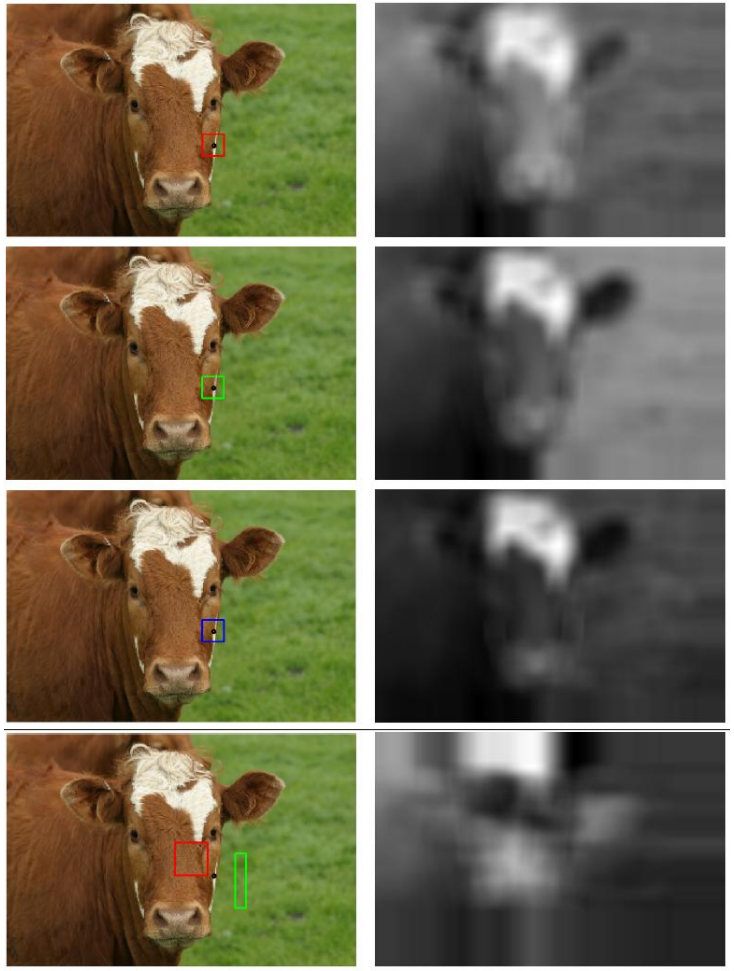
HOG



Difference of HOG responses

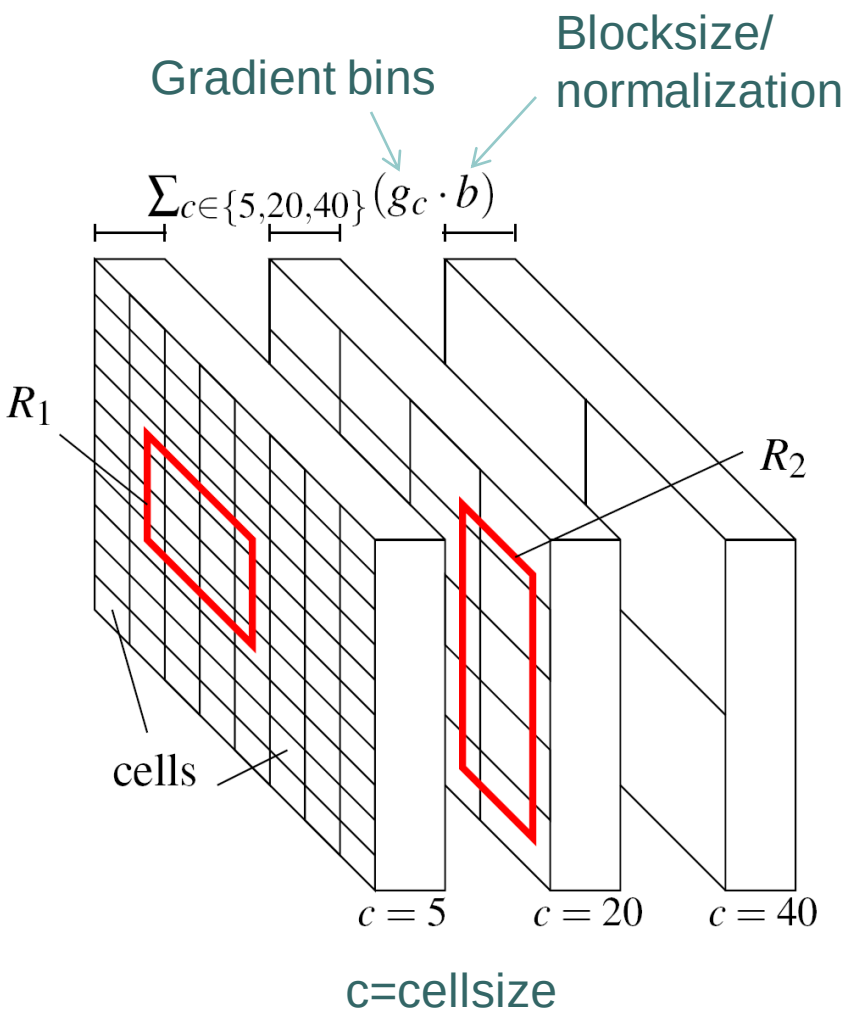
- Compute differences over various responses (RGB, textons, HOG)
- Use difference of rectangle responses together with a threshold as node-test $t_p < \lambda$?

Feature Response: Example



- Example of centered rectangle response:
  - Red-channel
  - Green-channel
  - Blue-channel
- Example of rectangle difference (red- and green-channel)

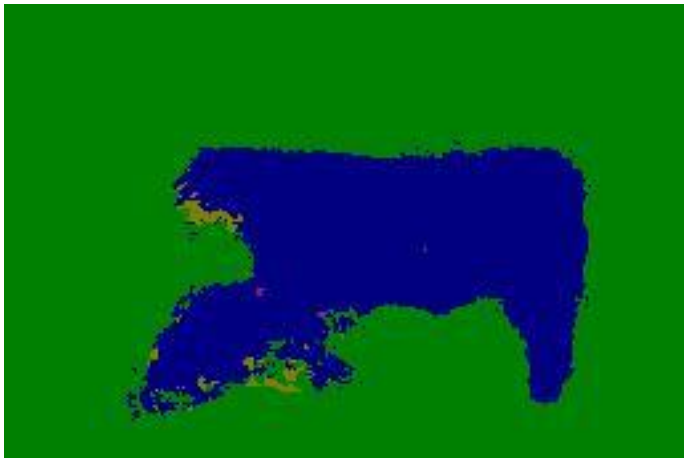
Features: HOG Detailed

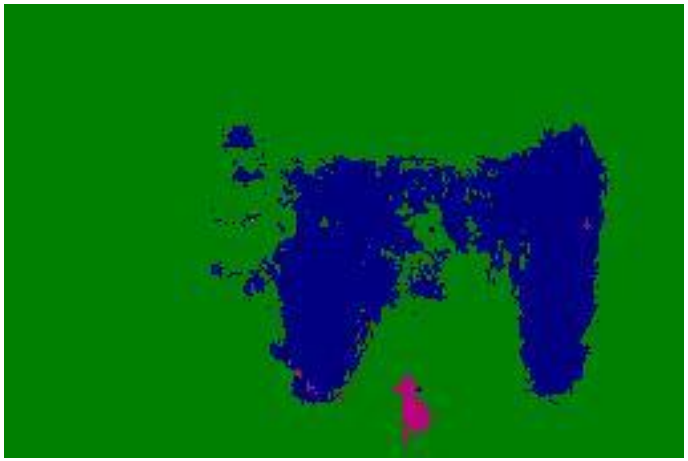


- Each pixel is discribed by a “stacked” hog descriptor with different parameters
- Difference computed over responses of one gradient bin with respect to a certain normalization and cellsize

Importance of different feature types

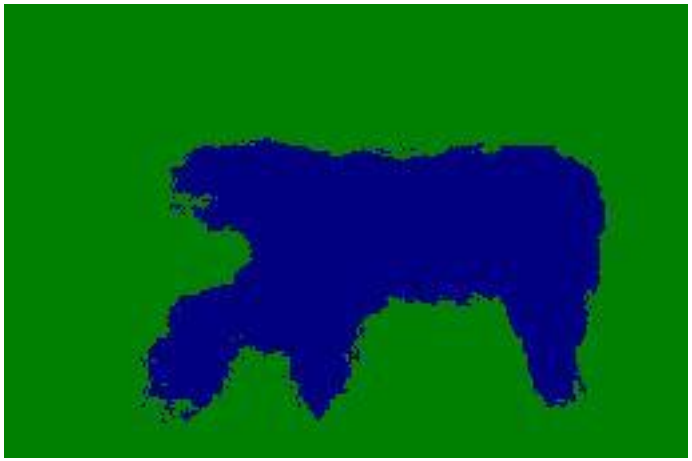
RGB





HOG

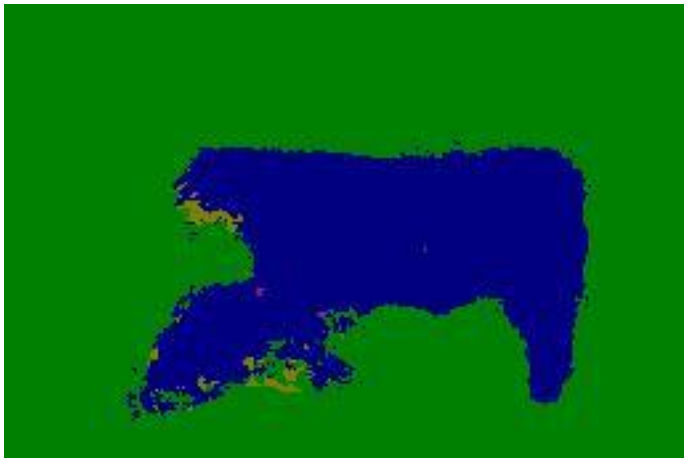
HOG & RGB

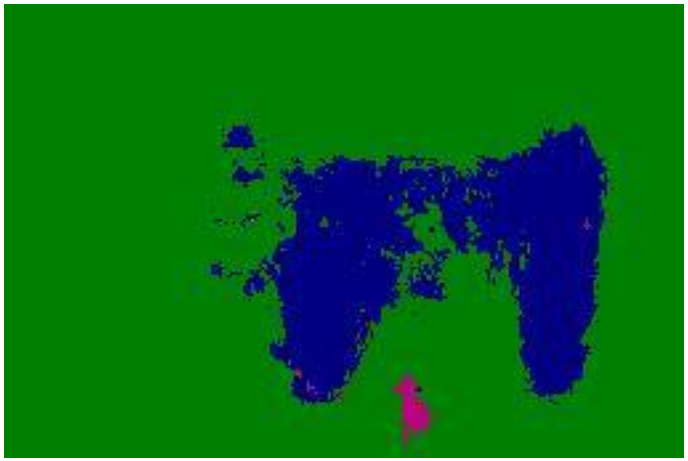


HOG & RGB

Importance of different feature types

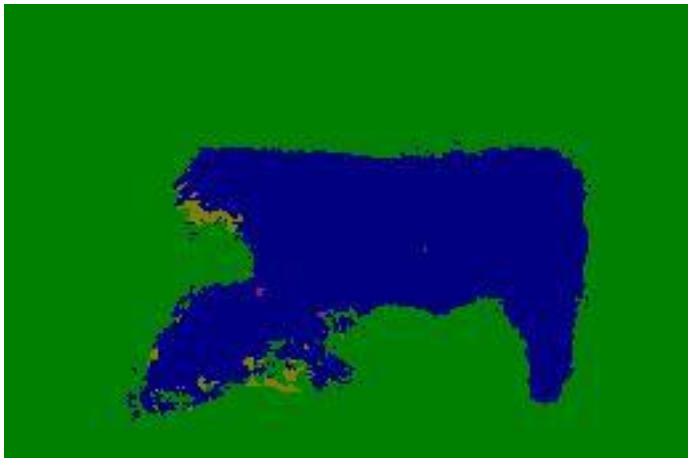
RGB





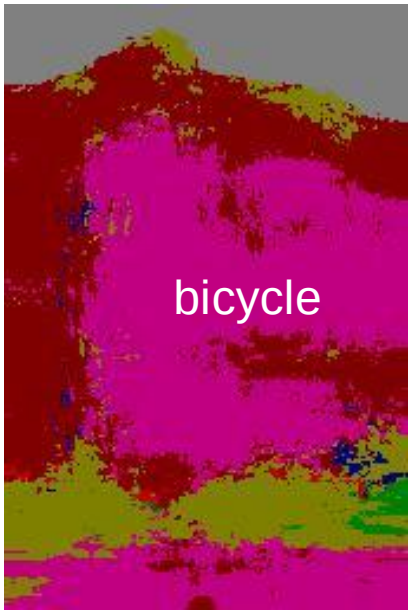
HOG

RGB



HOG & RGB

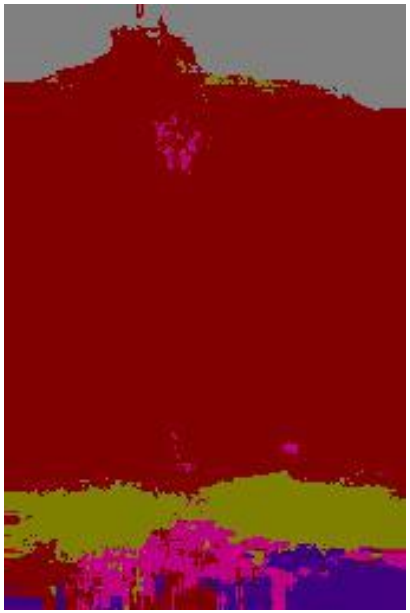
Importance of different feature types



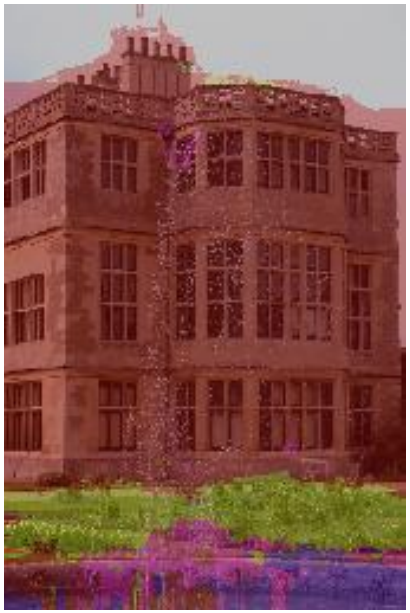
RGB

HOG

HOG & RGB



HOG & RGB



Conditional Random Field for Cleaner Object Boundaries

- Use global energy minimization instead of maximum a posteriori (MAP) estimate

Image Segmentation using Energy Minimization

Conditional Random Field (CRF)

- energy minimization using, e.g. Graph-Cut or TRW-S

$$E(\mathbf{c}) = \sum_i \psi_i(c_i, \mathbf{I}) + \sum_{ij \in I} \phi(c_i, c_j, \mathbf{I})$$

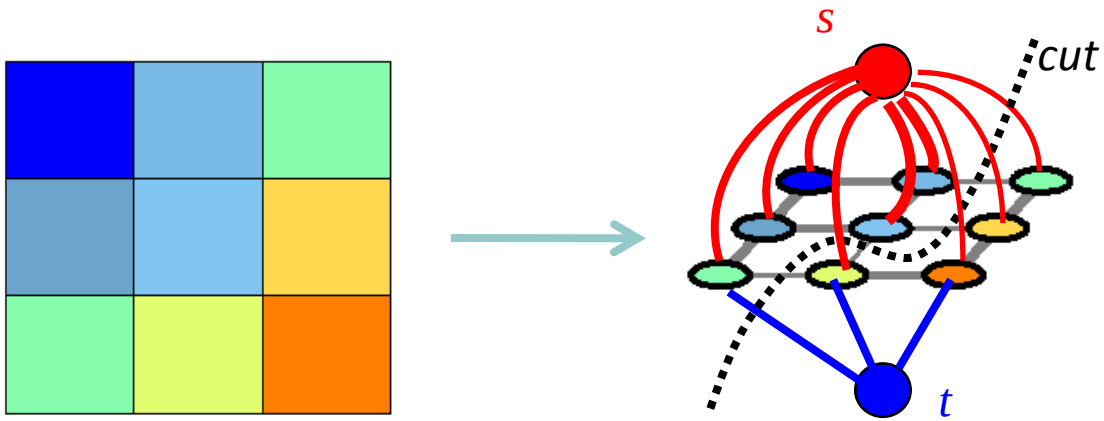
Unary likelihood

Contrast dependent Smoothness prior

$\mathbf{g}_{ij} \in \mathbb{R}^d$

Colour difference vector

c_i = binary variable representing label ('fg' or 'bg') of pixel i



Labelling problem

Graph Cut

CRF and Colour-Model

Test image specific colour-model

$$\log P(\mathbf{c}|\mathbf{I}) = \sum_i \left(\psi_i(c_i, \mathbf{I}) + \pi(c_i, \mathbf{I})\right) + \sum_{ij \in I} \phi(c_i, c_j, \mathbf{g}_{ij})$$

Class posteriors from Random Forest

Only for 2nd iteration

Contrast dependent smoothness prior

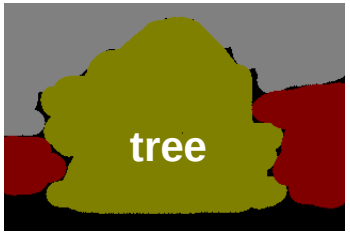
- CRF as commonly used (e.g. Shotton et al. ECCV06: TextonBoost)
- TRW-S is used to maximize this CRF
- Perform two iterations: one with one w/o colour model

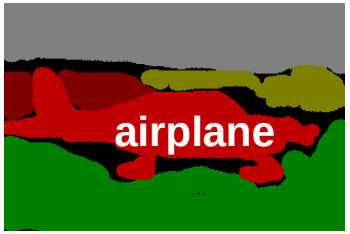
MSRC-Databases

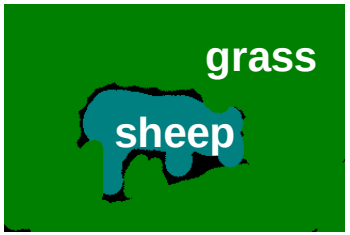
9-classes:
building,
grass,
tree,
cow,
sky,
airplane,
face,
car,
bicycle

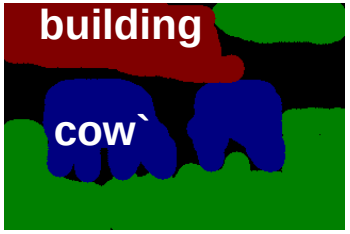
120 training-
120 test-
images

Similar:
21-classes

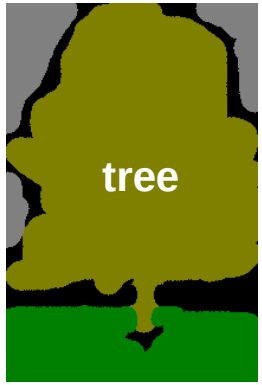


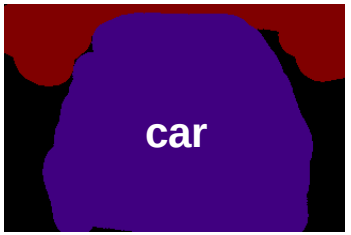



Images

Groundtruth

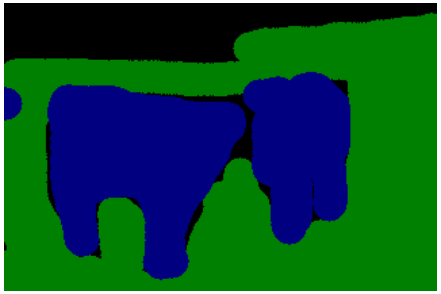
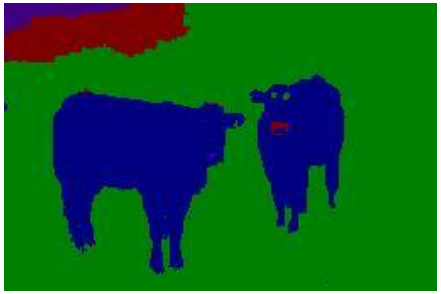
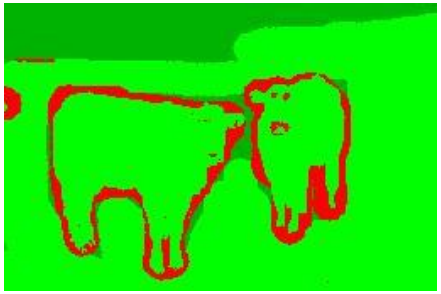
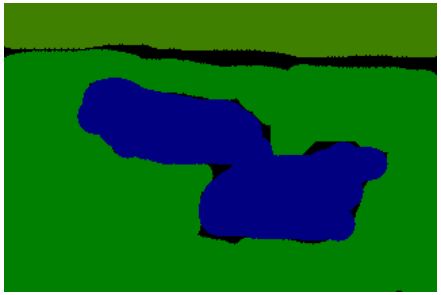
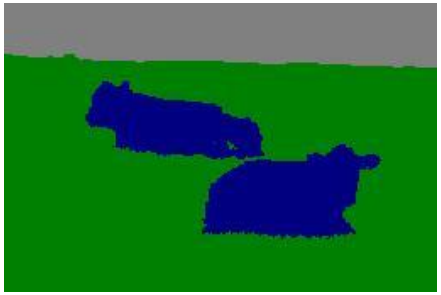
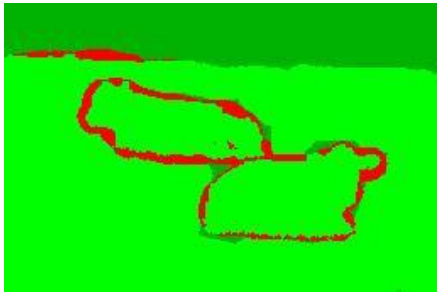
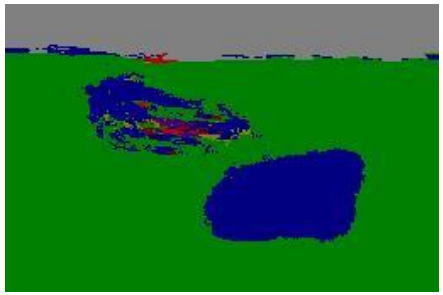
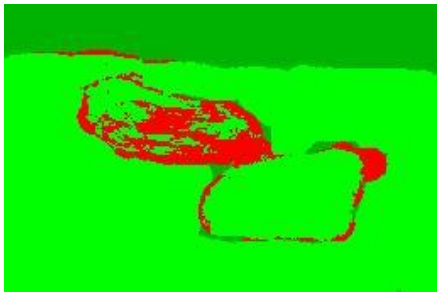




Images

Groundtruth

Segmentation Results (MSRC-DB) with Colour-Model

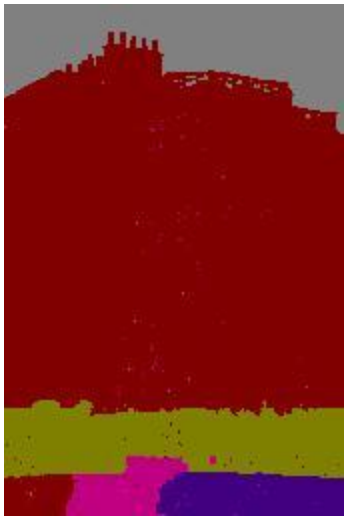
Image	Groundtruth	Classification	Classification Quality
			
			
w/o CRF Class posteriors only			

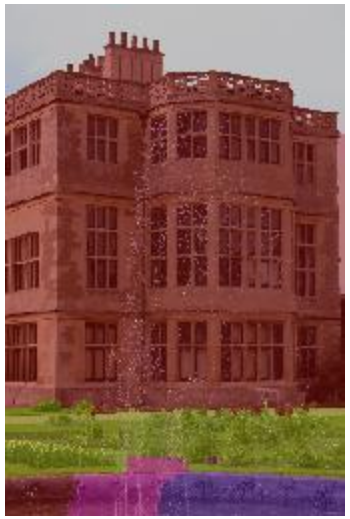
Segmentation Results (MSRC-DB) with Colour-Model

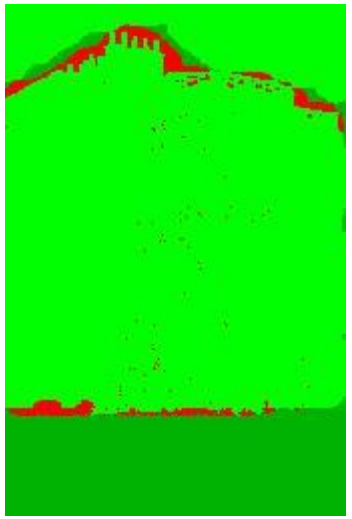
Classification

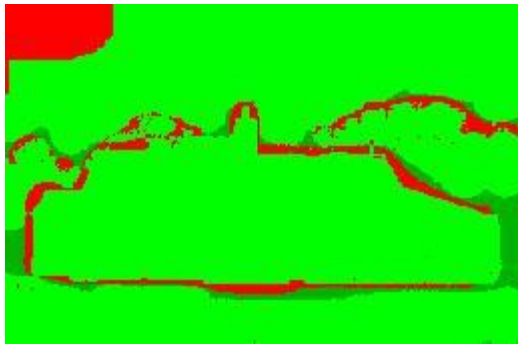
Image

Classification Quality

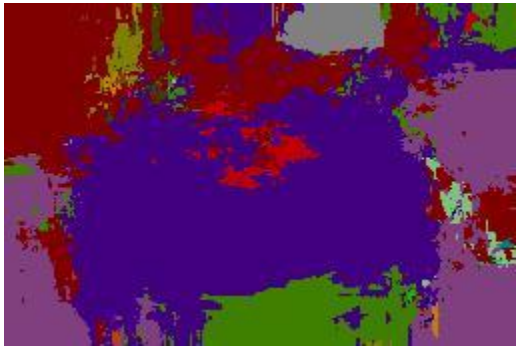
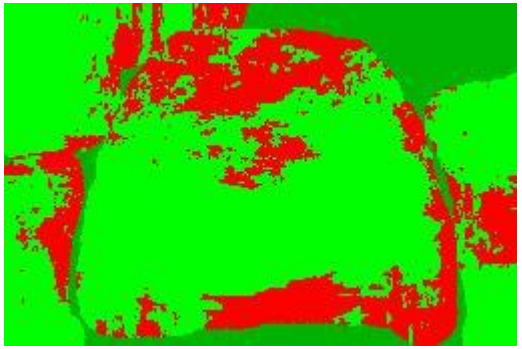
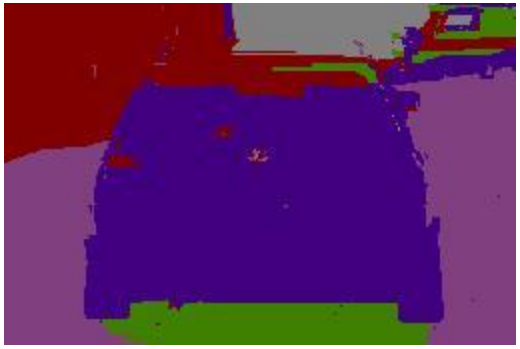




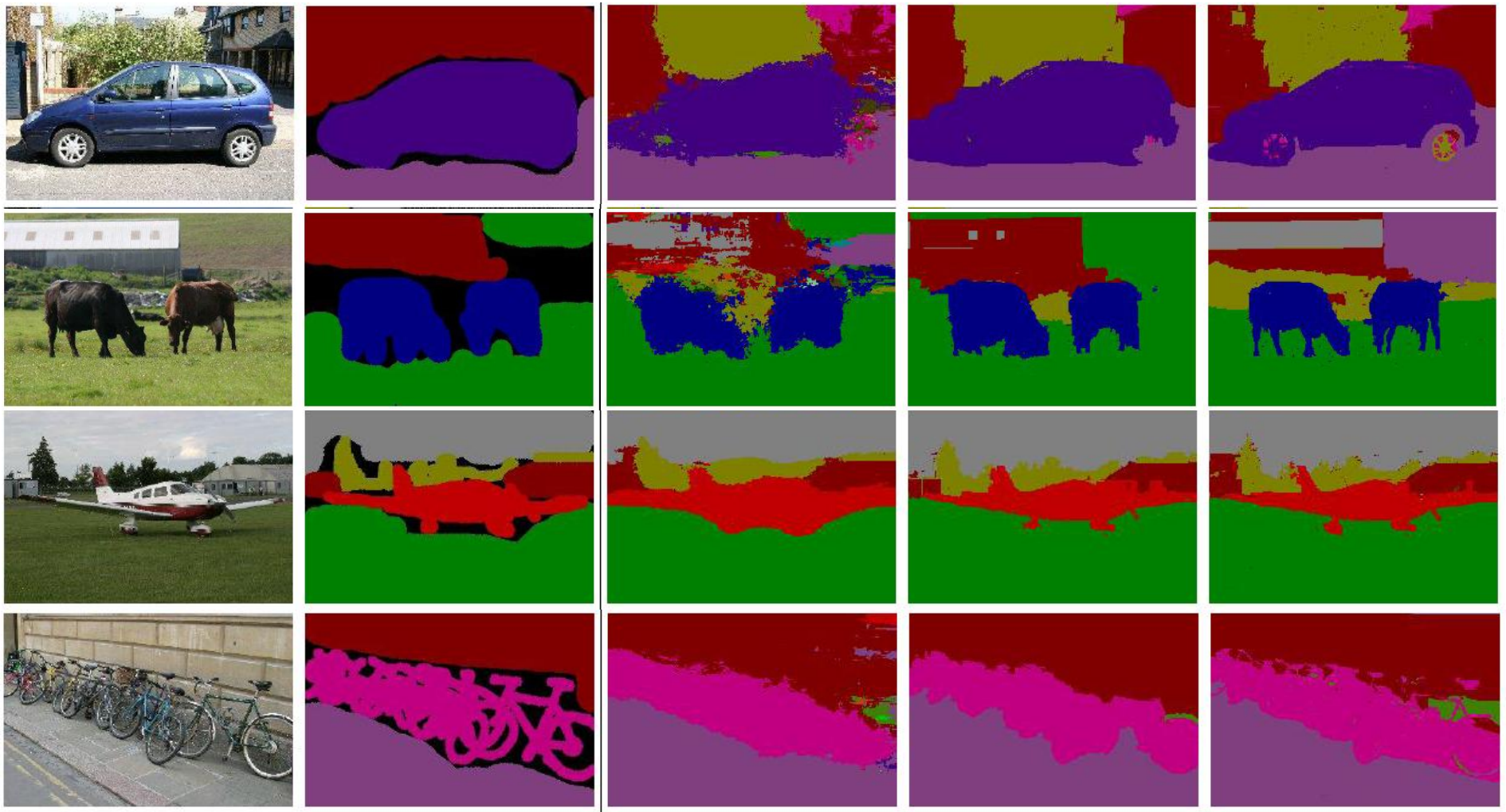




Segmentation Results (MSRC-DB 21 classes)

	Classification	Image overlay	Classification Quality
MAP w/o CRF			
CRF			

21-class MSCR dataset



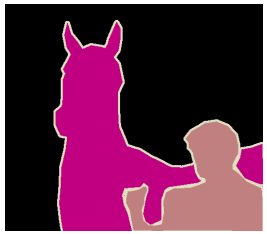
VOC2007-Database

20 classes:
Aeroplane
Bicycle
Bird
Boat
Bottle
Bus
Car
Cat
Chair
Cow
Diningtable
Dog
Horse
Motorbike
Person
Pottedplant
Sheep
Sofa
Train
Tvmonitor





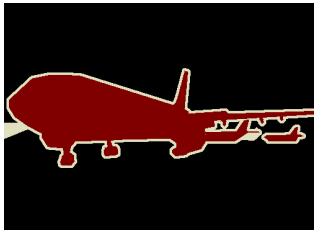


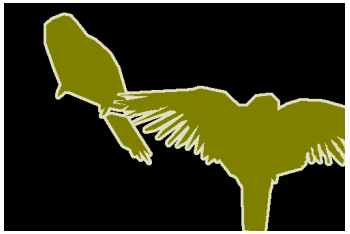



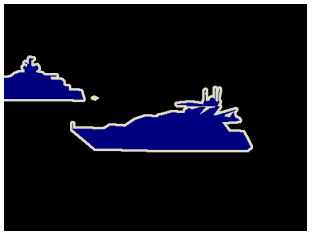
Images

Groundtruth







Images

Groundtruth

VOC 2007

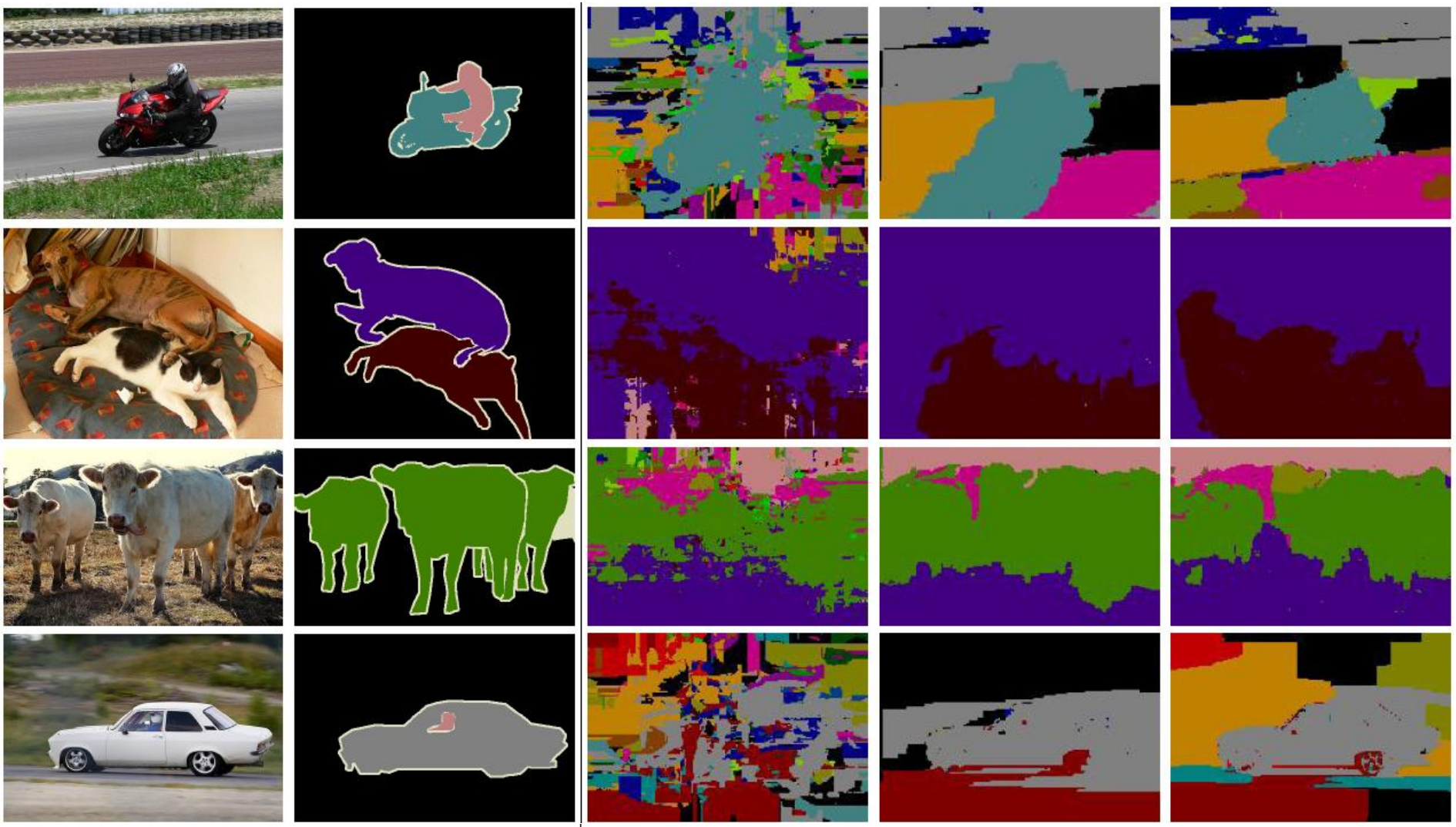


image
groundtruth
MAP
(21.4%)
w/o colour model
(24.4%)
w/ colour model
(22.7%)
plane 33.2
bike 21.3
bird 23.0
boat 16.2
bottle 6.4
bus 25.6
car 29.9
cat 6.0
chair 15.3
cow 17.5
table 31.3
dog 40.9
horse 63.3
mbike 37.6
prs 57.3
plant 9.7
sheep 5.8
sofa 8.5
train 33.0
tv 26.8
bg 4.1

Results

	textons	RGB	HOG	HOG,RGB	HOG,RGB, F17	HOG,RGB, SHCMs	CRF	Others
9-class (pixelwise)	72.8%	72.2%	75.9%	84.5%	86.0%	84.9%	**87.3%**	84.9% [1]
21-class (pixelwise)	40.3%	54.0%	56.3%	69.9%	71.5%	71.7%	**73.7%**	72.7% [2]
VOC2007 (class avg.)	13.6%	10.1%	17.4%	17.7%	19.7%	21.4%	**24.4%**	20.0% [3]

[1] Verbeek et al. NIPS2008; [2] Shotton et al. ECCV2006;
[3] Shotton et al. CVPR 2008 (raw results w/o image level prior)

- Combination of features improves performance
- CRF improves performance and most importantly visual quality

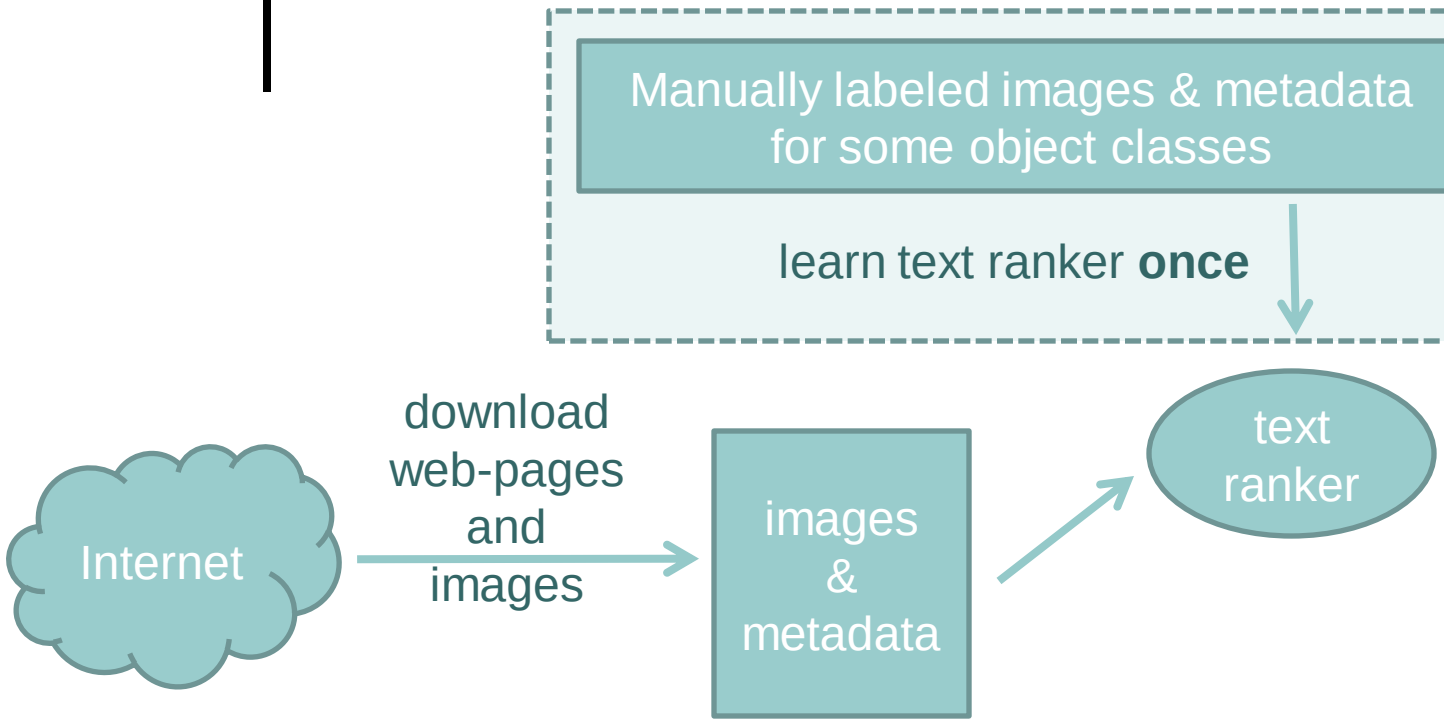
Summary

- Discriminative learning of rectangle shapes and offsets improves performance
- Different feature types can easily be combined in the random forest framework
- Combining different feature types improves performance

Part II: Web-Supervised Visual Learning

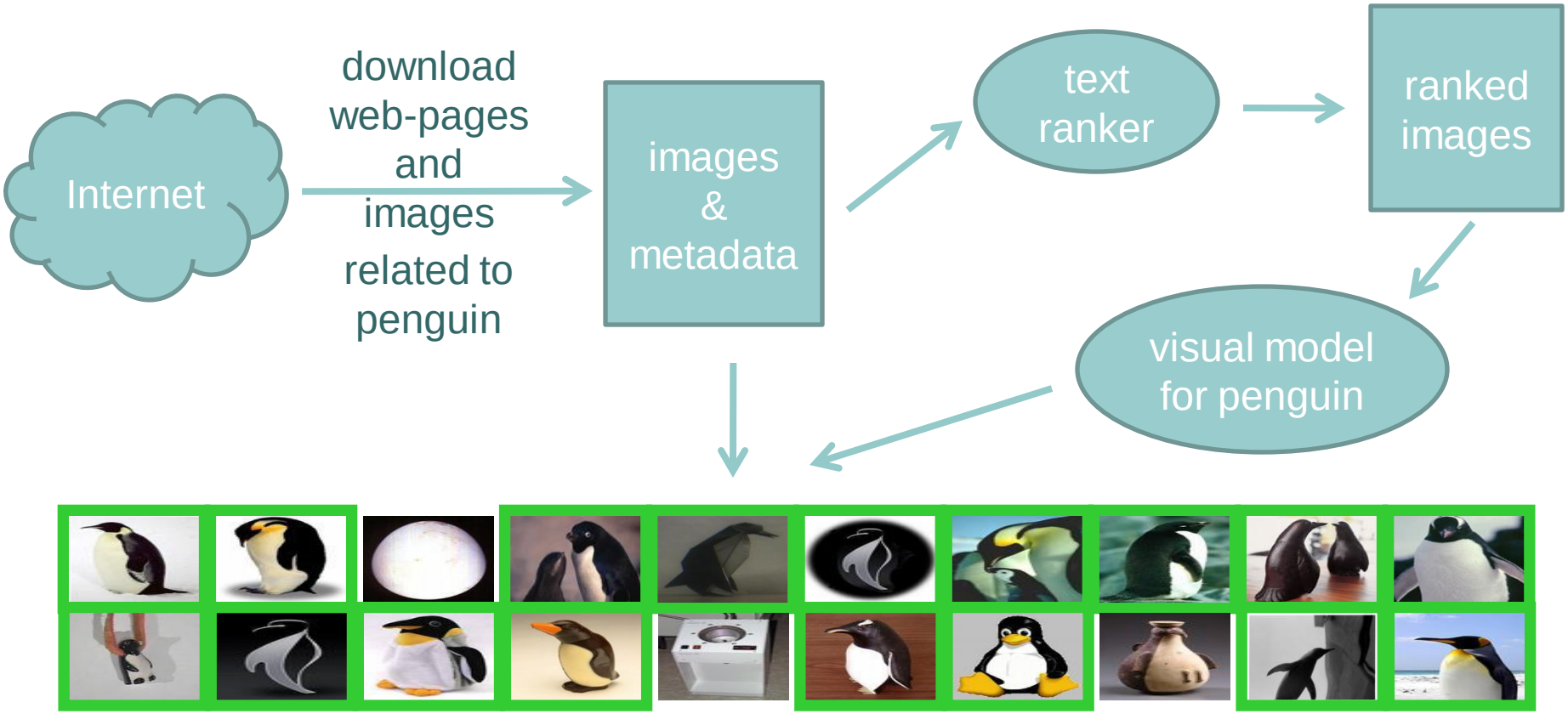
- Goal: retrieve class specific images from the web
  - No user interaction (fully automatic)
- Images are ranked using a multi-modal approach:
  - Text & metadata from the web-pages
  - Visual features
- Previous work on learning relationships between words and images:
  - Barnard et al. JMLR 03 (Matching Words and Pictures)
  - Berg et al. CVPR 04, CVPR 06

Overview: Harvesting Algorithm



Overview: Harvesting Algorithm

User specifies: penguin



Text&Metadata Ranker

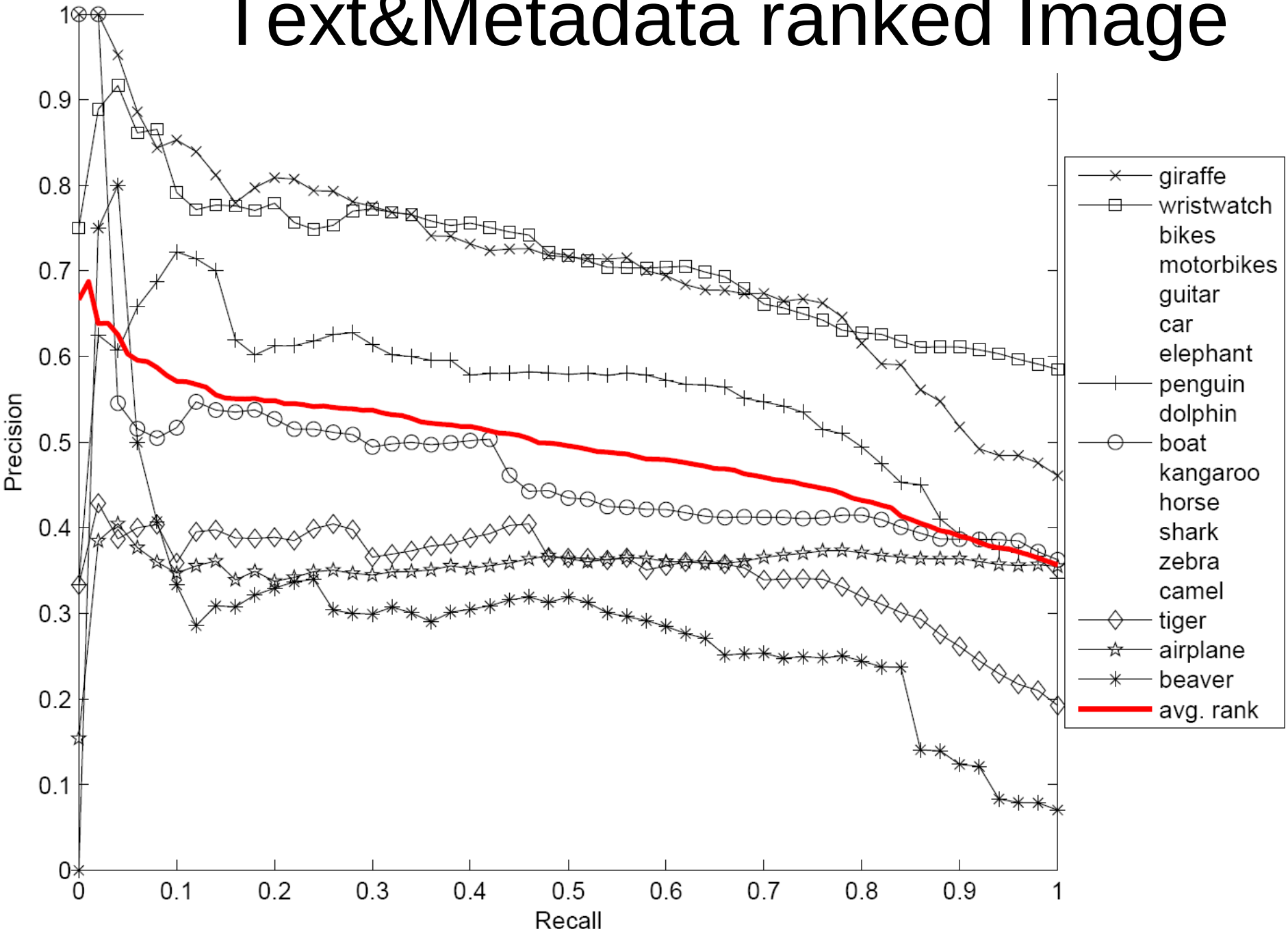
- Why don't we start with Google image search?
  - Limited return (only 1000 images)
- Goal: object class independent ranker

- Rank images using Bayes model on binary feature vector:

a=(context10, context50, filename, filedir, imagealt, imagetitle, websitetitle)

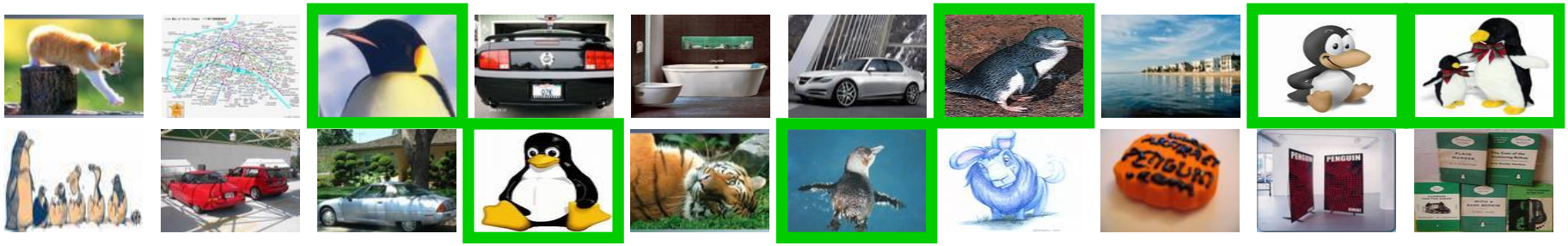
$$P(y|\mathbf{a}) = P(a_1, \ldots, a_4|y) \prod_{5}^{7} P(a_i|y) \cdot P(y)/Z$$

Text&Metadata ranked Image



Visual Ranking

- How to learn visual model from these noisy images?
- Where do we get the training data from?



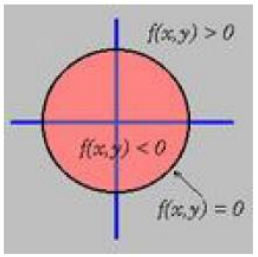
- Train on top text ranked images → positive data
- Randomly sample images → negative data
- Support Vector Machine (SVM)
  - robust to noise

Filter drawings & abstract Images

drawing&symbolic









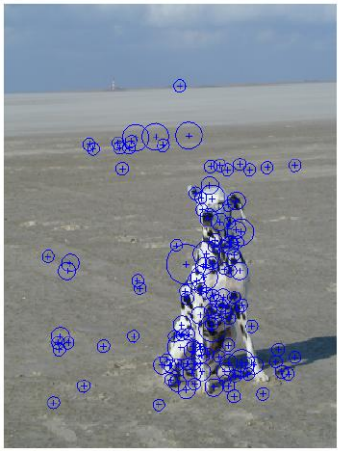
non drawing&symbolic






- Gradient- & colour-histograms
- RBF-SVM

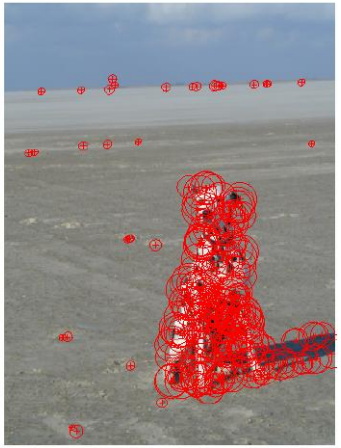
Visual Features



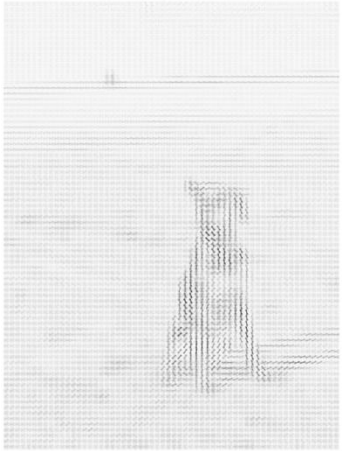
Difference of Gaussians

Multiscale Harris

Kadir's saliency



Canny edge points

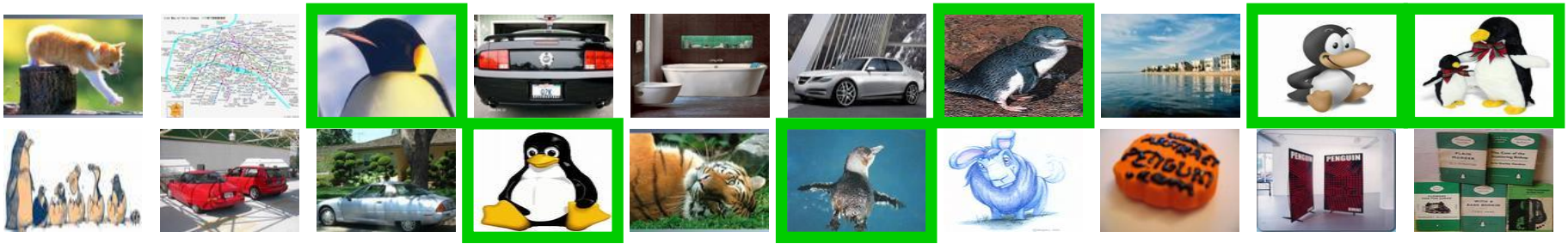


HOG

- 400 visual-words from four interest point detectors
- HOG descriptor to represent shape
- RBF-SVM on "stacked" feature vector

Example: Penguin

1. Enter “penguin”
2. Retrieve images from web pages returned by Google web search on **penguin**
   - 522 in-class, 1771 non-class



3. Remove drawings & abstract images
   - 391 in-class, 784 non-class

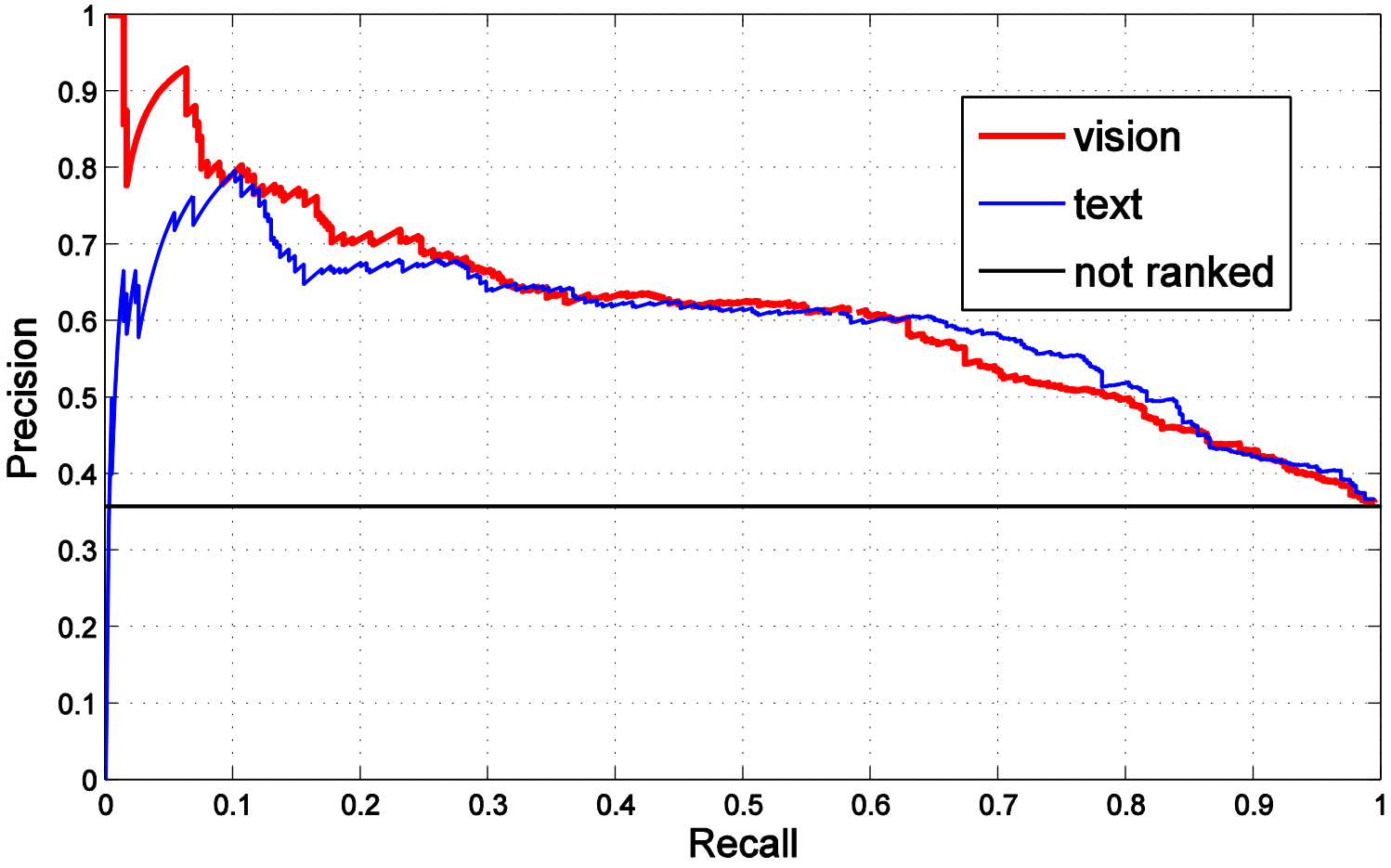
Example: Penguin continued

4. rank images using naïve Bayes metadata ranker

5. Train SVM on visual features using ranked images as noisy training data

6. Final re-ranking using trained SVM

Example: Penguin continued



Text+visual ranked images

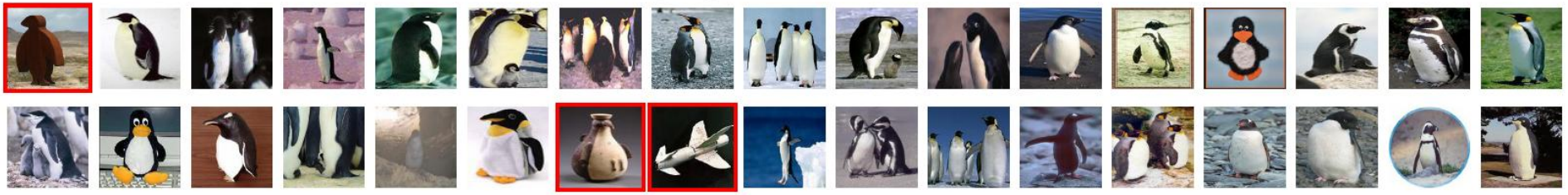
- Text ranker:
  - rank images for new requested object-class
- Visual ranker:
  - Train visual classifier and re-rank images

airplane: text ranker

airplane: HOG+BOW t+v

Examples continued

penguin: text ranker



penguin: HOG+BOW t+v

shark: text ranker

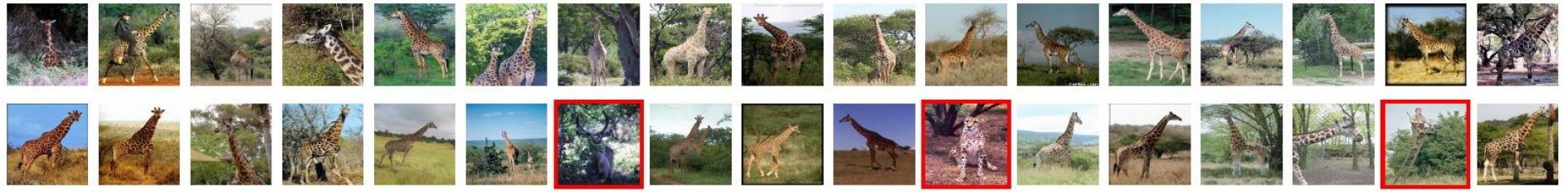
shark: HOG+BOW t+v

Examples continued

elephant: text ranker

elephant: HOG+BOW t+v

giraffe: text ranker



giraffe: HOG+BOW t+v

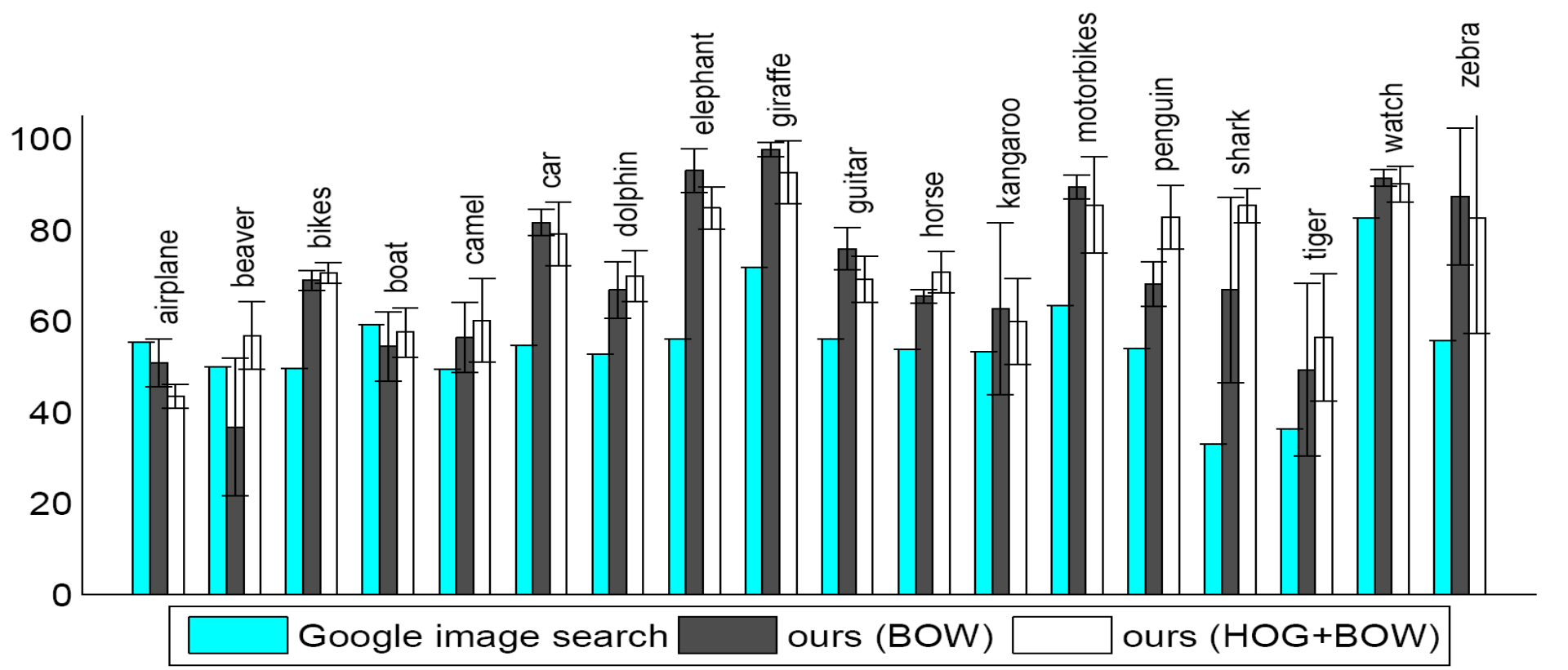
Examples continued

wristwatch: text ranker

wristwatch: HOG+BOW t+v

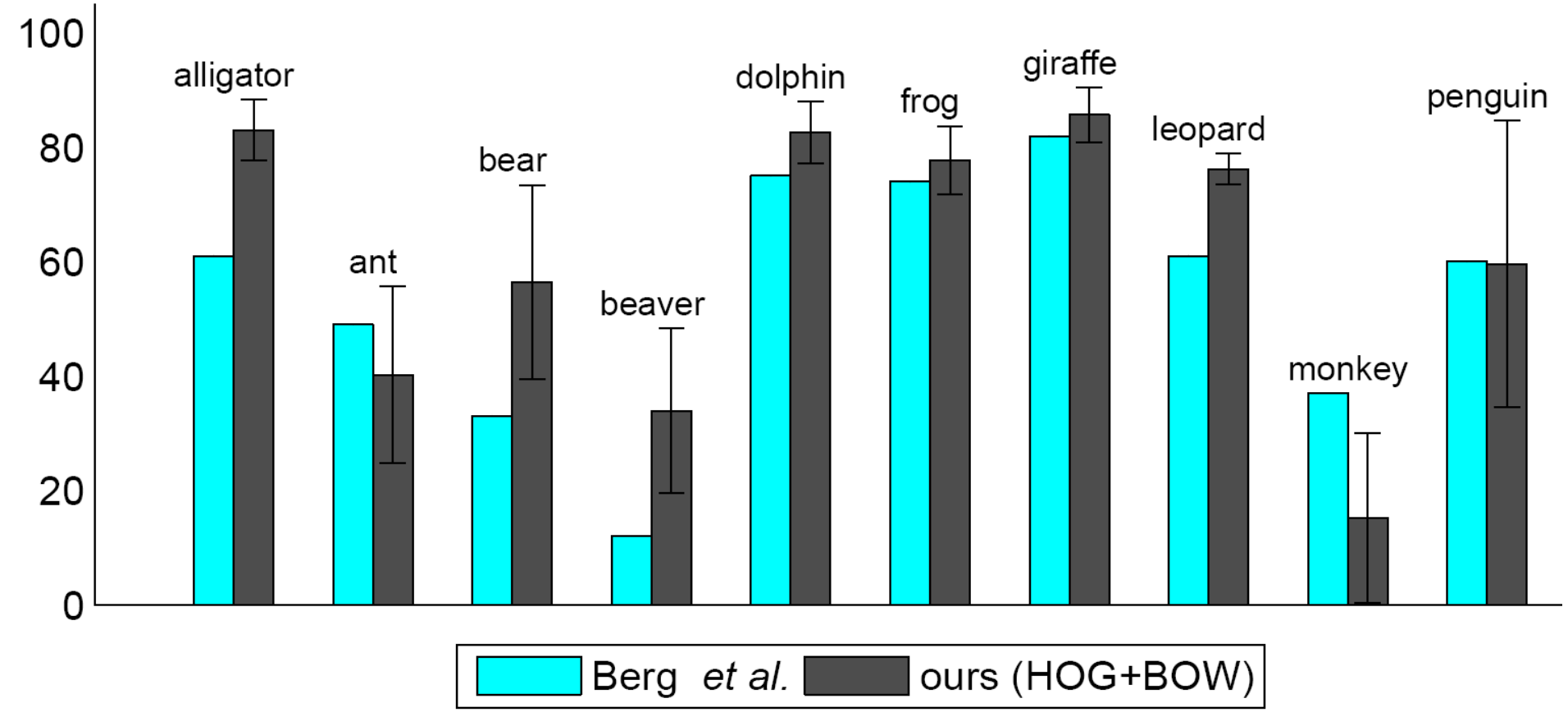
zebra: text ranker

zebra: HOG+BOW t+v

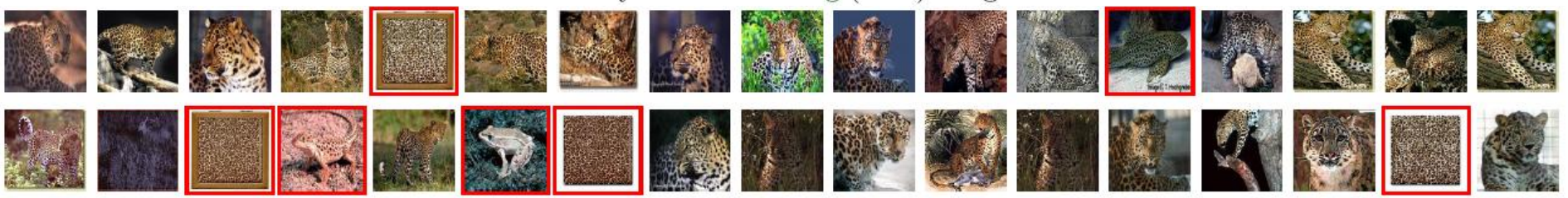


car: returned by Google image search

car: re-ranking Google images using our visual re-ranking



monkey: re-ranked Berg (2006) images



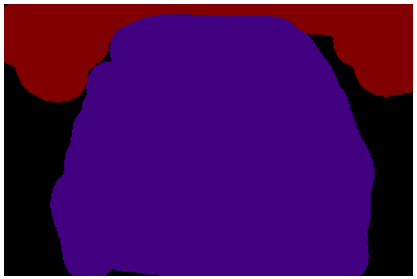
leopard: re-ranked Berg (2006) images

Summary

- Use object-class **independent** text ranker to retrieve training data
- Train visual classifier on top text ranked images
- Show applicability on different datasets
  - Google image search
  - Berg et al. (Animals on the Web)

Part III: Segmentation from Harvested Images

- Random Forest pixelwise classification
- Use weak supervision
  - No segmented training data
  - Per image classlabels are used



- Segment images in 21-class MSRC dataset
  - Weak supervision: 52.1% (w/o CRF)
  - Strong supervision: 71.5% (w/o CRF)

(following images with CRF)

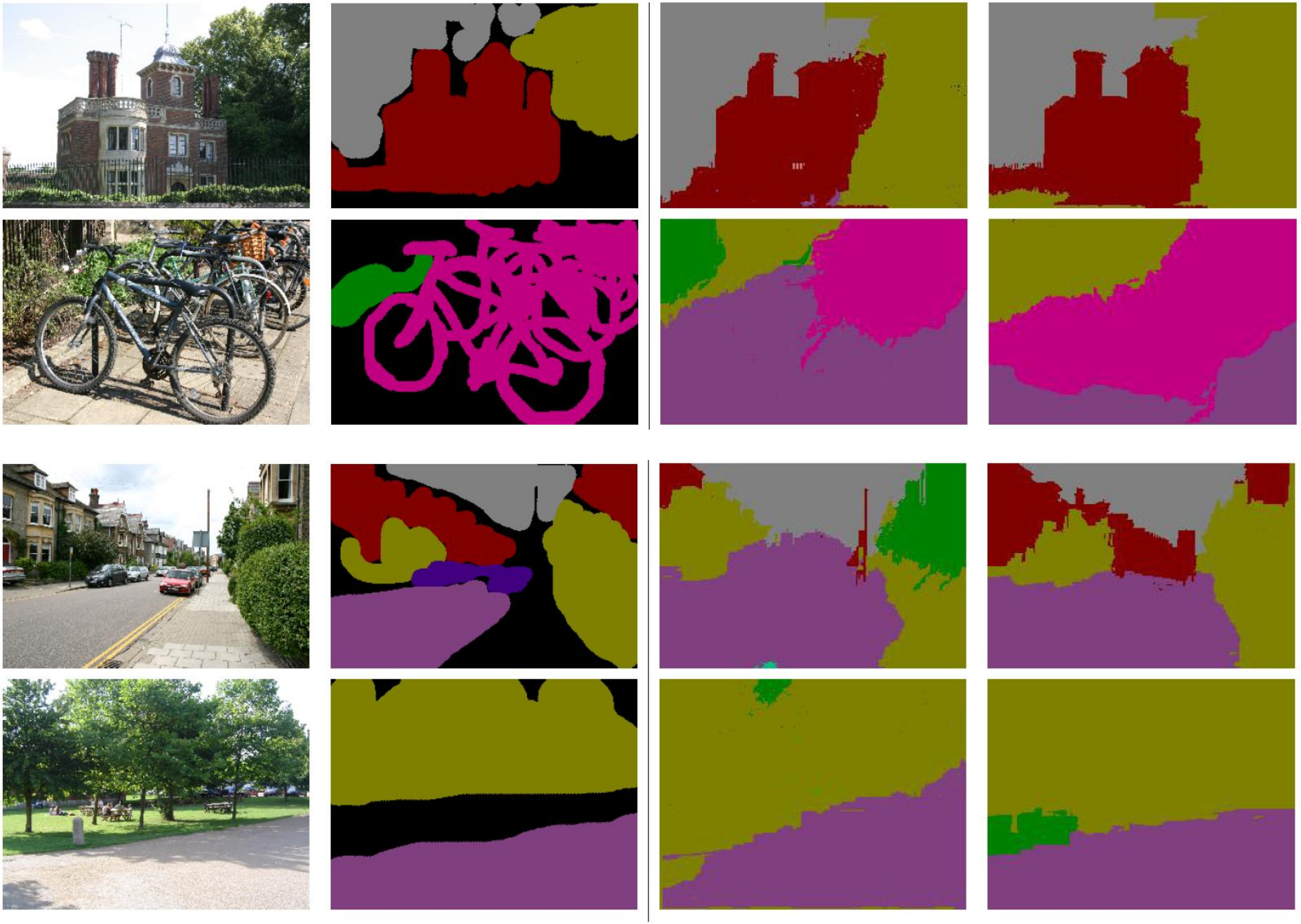


image
groundtruth
weak supervision
strong supervision

Learn Segmentation Model

- Train Random Forest on top ranked
  100 car images and
  200 randomly sampled background images
- Segment images in 21-class MSRC dataset
  (using CRF with colour-model)

car: text ranker

car: HOG+BOW t+v

image groundtruth automatic segmentation

Summary

- Show that Random Forest can be trained on weakly labelled training data
- Combine strong Random Forest segmentation with unsupervised visual learning
- This allows learning of segmentation models w/o requiring manually labeled training data

Discussion & Future Work

- Image level class priors (Shotton *et al.* CVPR08) can improve performance dramatically
- Incorporate a more global shape into the decision trees
- Hierarchy of trees
  - Top trees classifying interesting image subareas
  - Subsequent trees perform fine grained segmentation