Ecole Normale Supérieure Lyon
Winter School Vision and Machine Learning
January 24-28, 2011

# Category-level localization

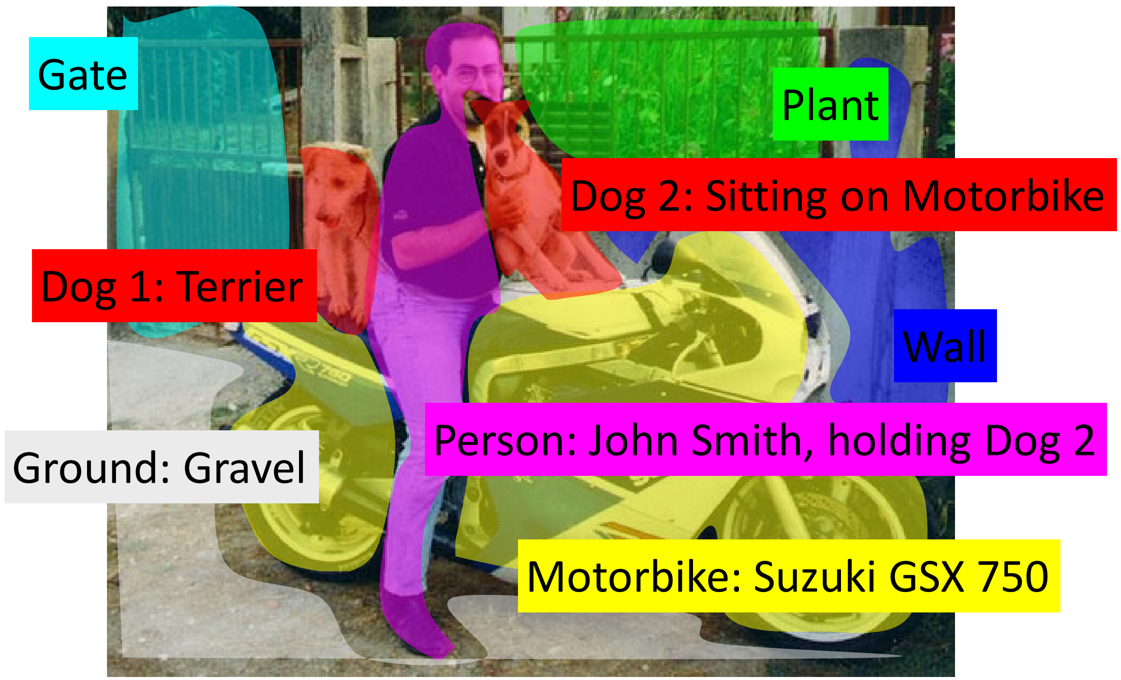

**Ivan Laptev**

INRIA, WILLOW, ENS/INRIA/CNRS UMR 8548
Laboratoire d'Informatique, Ecole Normale Supérieure, Paris


Includes slides from: Ondra Chum, Alyosha Efros, Mark Everingham, Pedro Felzenszwalb, Rob Fergus, Kristen Grauman, Bastian Leibe, Ivan Laptev, Fei-Fei Li, Marcin Marszalek, Pietro Perona, Deva Ramanan, Bernt Schiele, Jamie Shotton, Andrea Vedaldi and Andrew Zisserman

# What we would like to be able to do…

- Visual scene understanding
- **<u>What</u>** is in the image and **<u>where</u>**



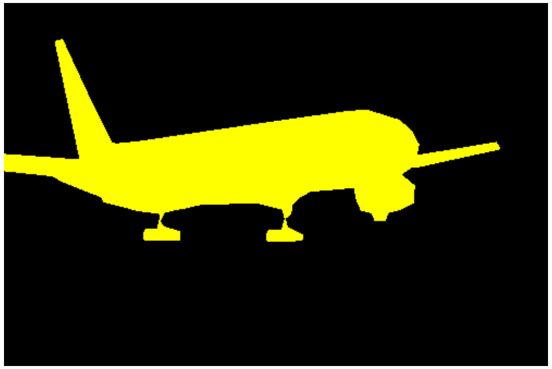

- Object categories, identities, properties, activities, relations, …

# Recognition Tasks

- **Image Classification**
  - Does the image contain an aeroplane?

- **Object Class Detection/Localization**
  - Where are the aeroplanes (if any)?

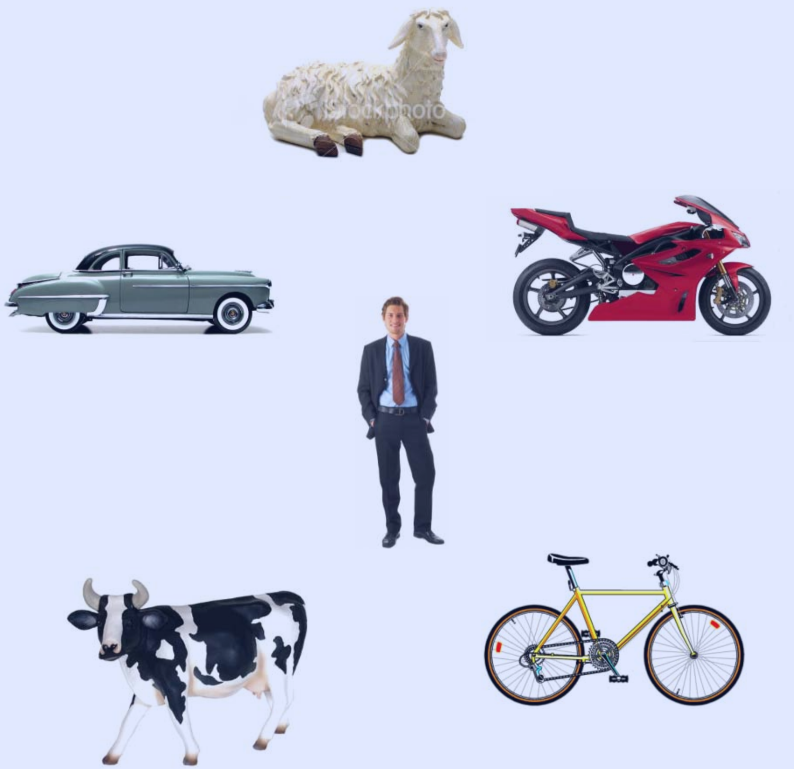
- **Object Class Segmentation**
  - Which pixels are part of an aeroplane (if any)?



# Things vs. Stuff

Ted Adelson, Forsyth et al. 1996.

Thing (n): An object with a specific size and shape.



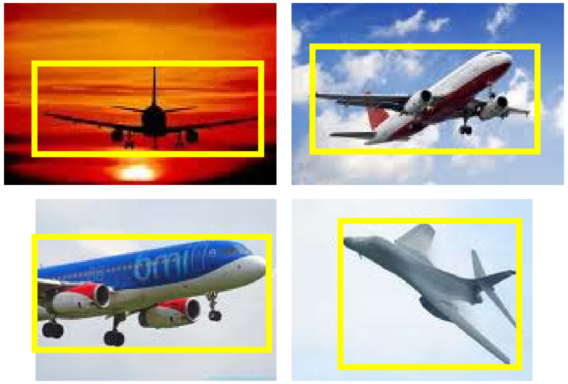
Stuff (n): Material defined by a homogeneous or repetitive pattern of fine-scale properties, but has no specific or distinctive spatial extent or shape.

Slide: Geremy Heitz

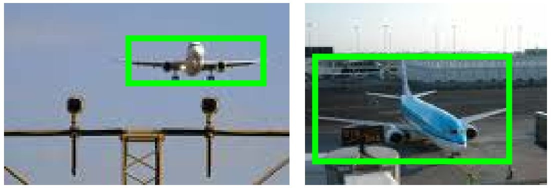
# Recognition Task

- **Object Class Detection/Localization**
  - Where are the aeroplanes (if any)?

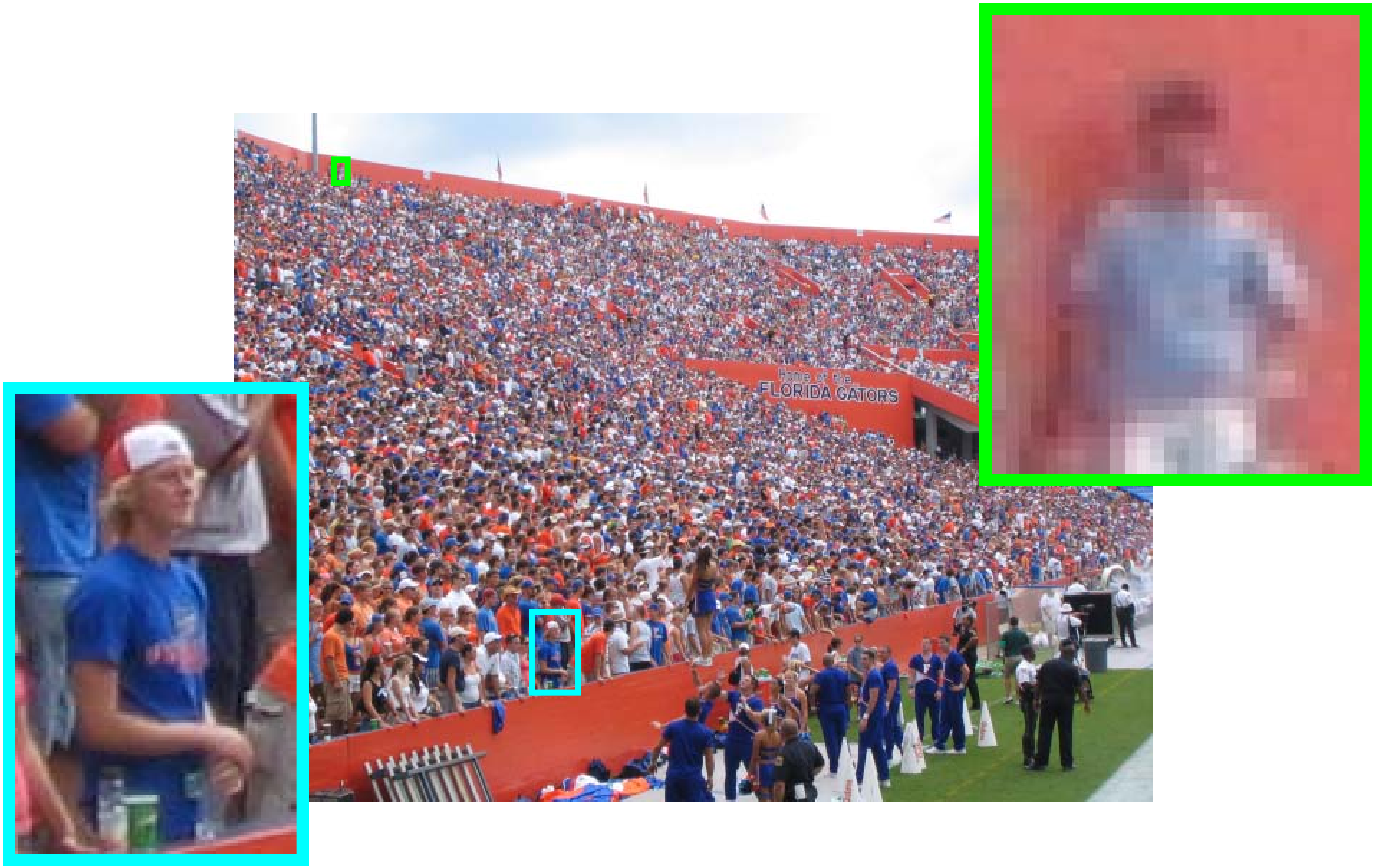

- **Challenges**
  - Imaging factors e.g. lighting, pose, occlusion, clutter
  - Intra-class variation



- **Compared to Classification**
  - Detailed prediction e.g. bounding box
  - Location usually provided for training



# Challenges: Scale



# Challenges: Background Clutter

# Challenges: Occlusion and truncation

# Challenges: Intra-class variation

# Object Category Recognition by Learning

- Difficult to define model of a category. Instead, **learn** from **example images**

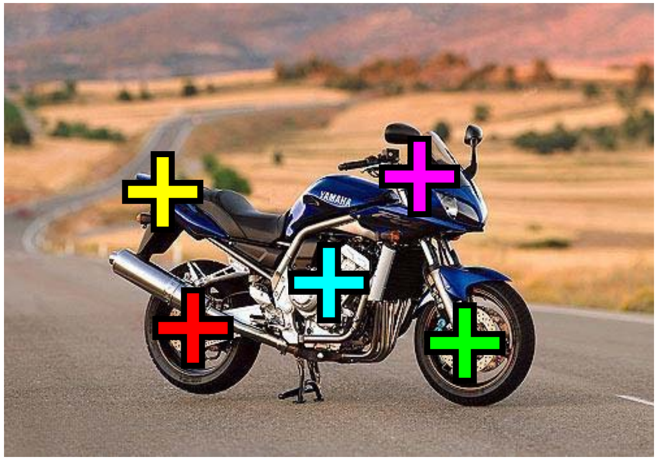
# Level of Supervision for Learning

Image-level label

Bounding box

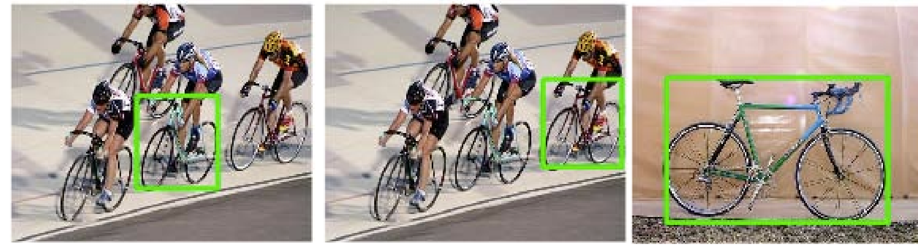
Pixel-level segmentation

“Parts”



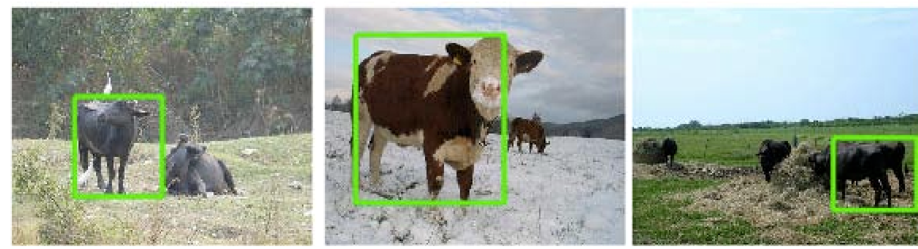
# Preview of typical results

aeroplane



bicycle

car



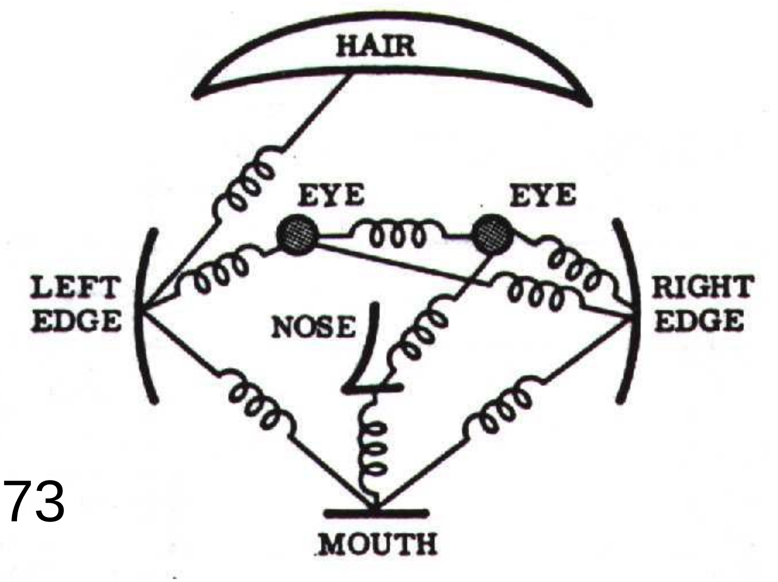
cow

horse

motorbike

# Class of model: Pictorial Structure

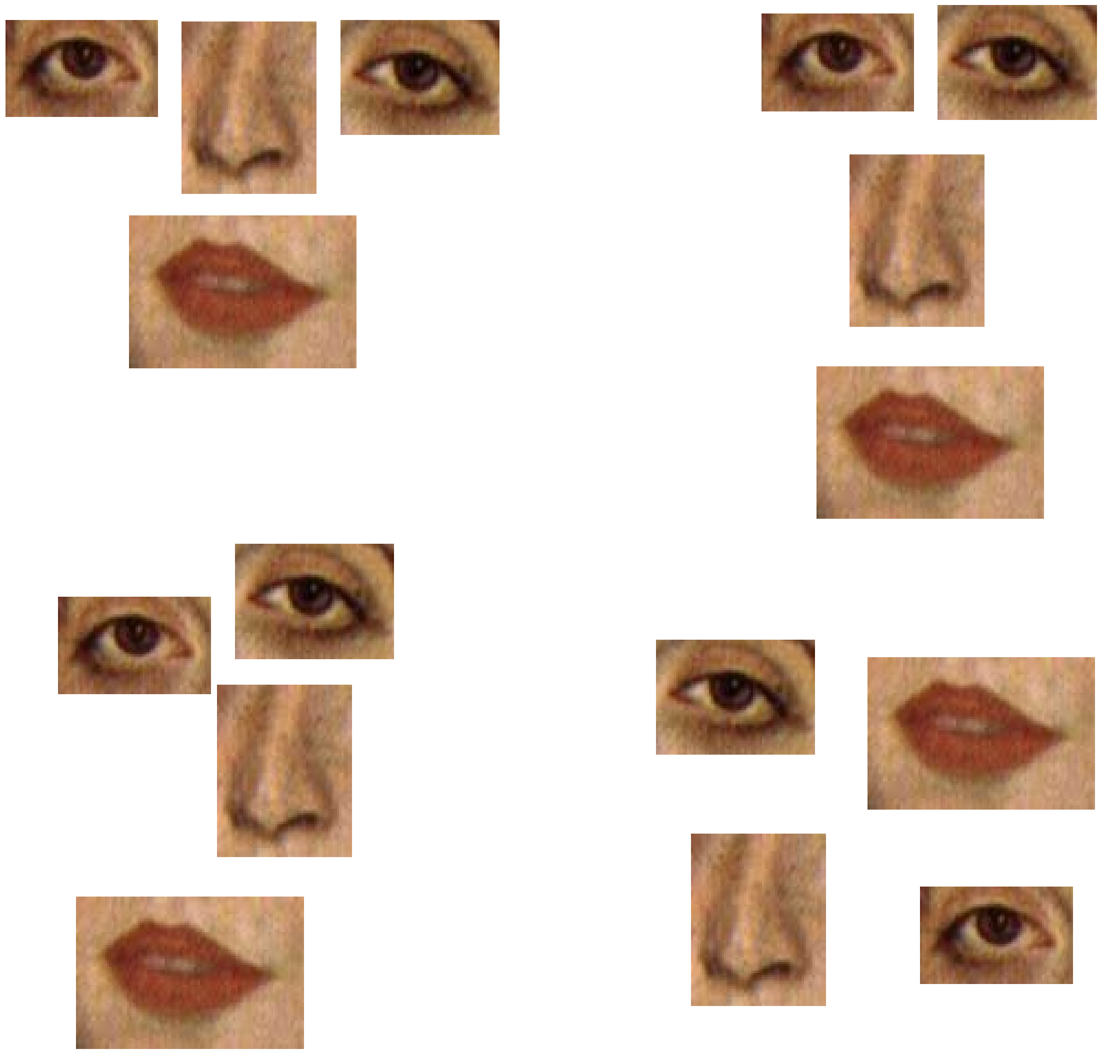
- Intuitive model of an object
- Model has two components
  1. parts (2D image fragments)
  2. structure (configuration of parts)
- Dates back to Fischler & Elschlager 1973




Is this complexity of representation necessary ?

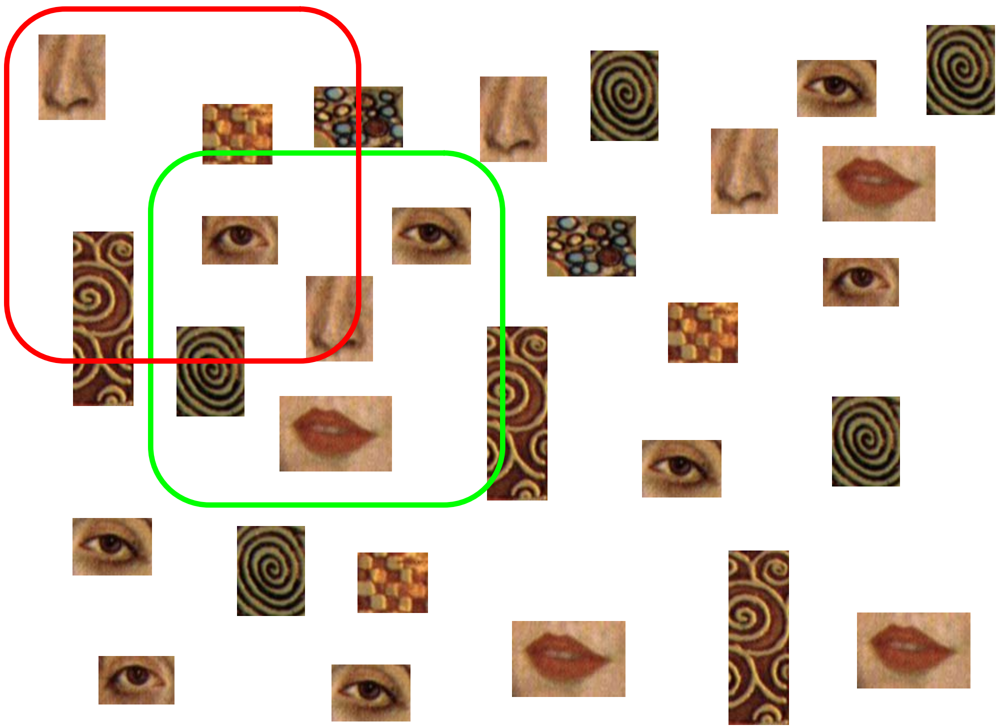
Which features?

# Restrict deformations



# Problem of background clutter

- Use a sub-window
  - At correct position, no clutter is present
  - Slide window to detect object
  - Change size of window to search over scale



# Outline

1. Sliding window detectors

2. Features and adding spatial information

3. Histogram of Oriented Gradients (HOG)

4. Two state of the art algorithms and PASCAL VOC

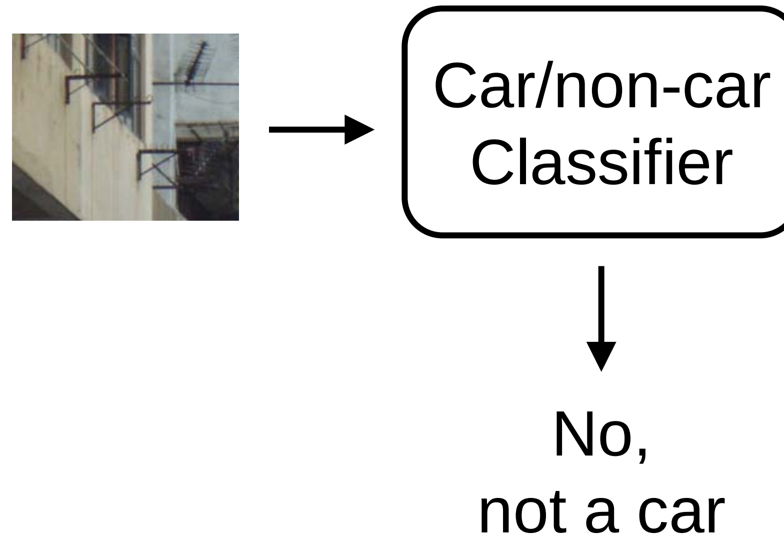
5. The future and challenges

# Outline

1. Sliding window detectors
   - Start: feature/classifier agnostic
   - Method
   - Problems/limitations
2. Features and adding spatial information
3. Histogram of Oriented Gradients (HOG)
4. Two state of the art algorithms and PASCAL VOC
5. The future and challenges

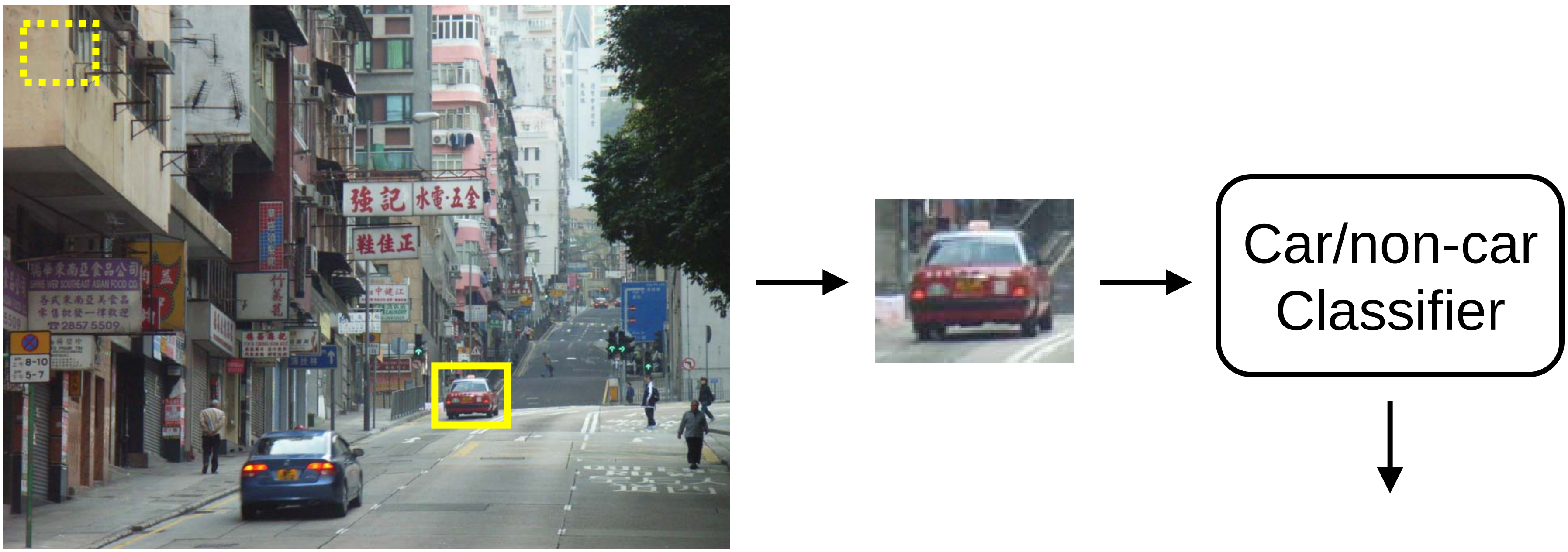
# Detection by Classification

- Basic component: binary classifier



# Detection by Classification

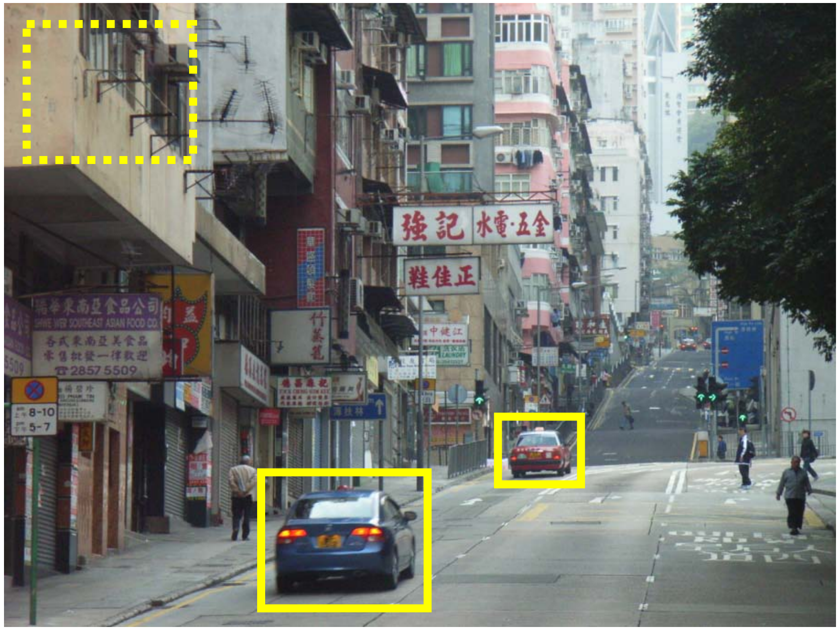
• Detect objects in clutter by **search**




• **Sliding window**: exhaustive search over position and scale

# Detection by Classification

- Detect objects in clutter by **search**

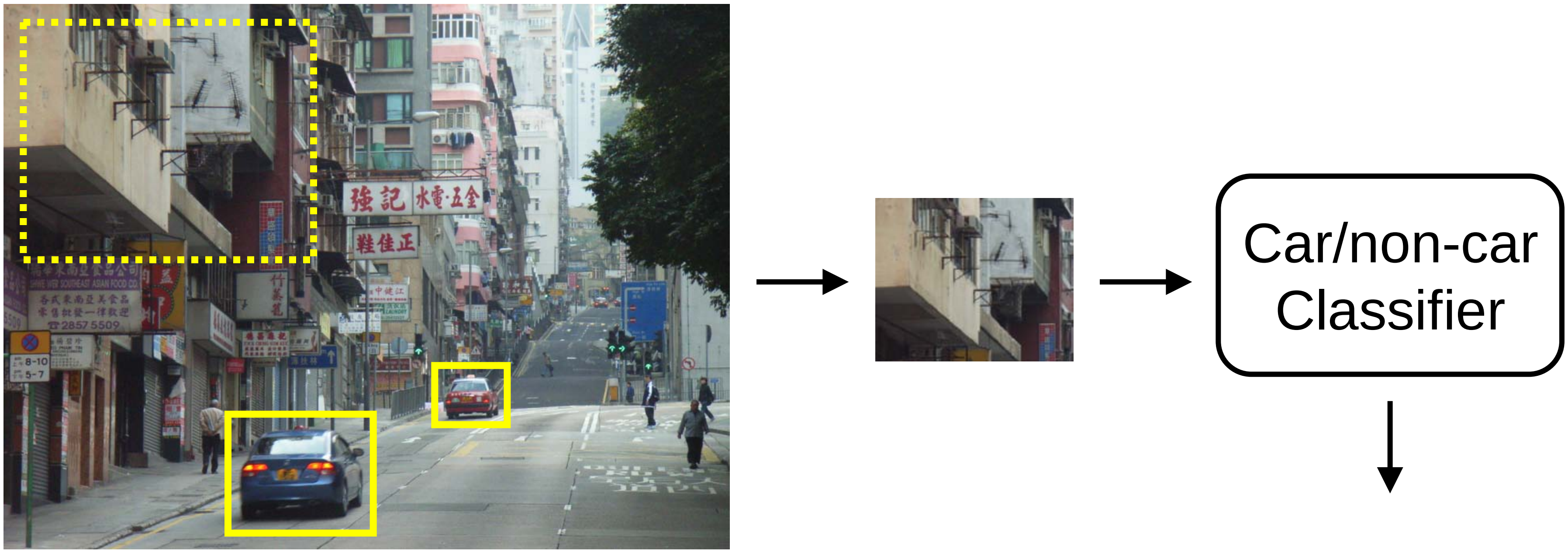


Car/non-car Classifier

- **Sliding window**: exhaustive search over position and scale

# Detection by Classification

• Detect objects in clutter by **search**

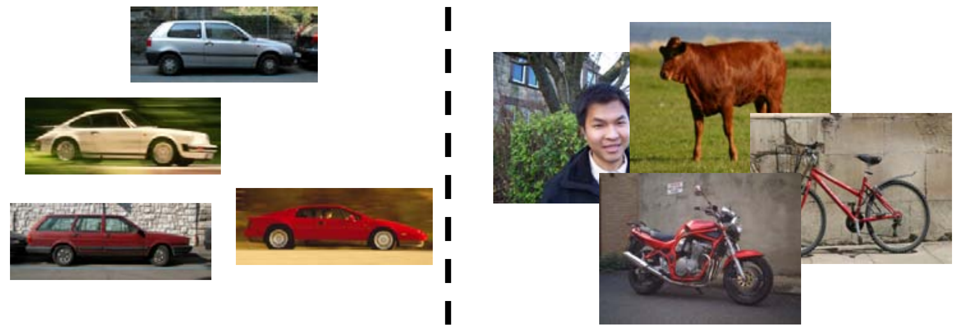



• **Sliding window**: exhaustive search over position and scale

(can use same size window over a spatial pyramid of images)

# Window (Image) Classification

Training Data



 → Feature Extraction → $\begin{bmatrix} \vdots \\ \vdots \end{bmatrix}$ → Classifier $F(x)$

$\mathbf{x}$

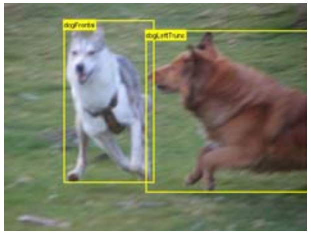
Car/Non-car

$P(c|\mathbf{x}) \propto F(\mathbf{x})$

- Features usually engineered
- Classifier learnt from data

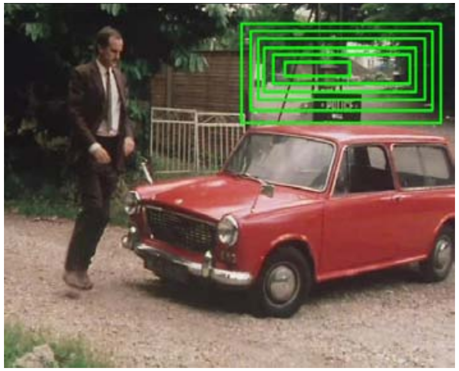
# Problems with sliding windows …

- aspect  ratio
- granularity (finite grid)
- partial occlusion
- multiple responses




See recent work  by

- Christoph Lampert et al CVPR 08, ECCV 08



# Outline

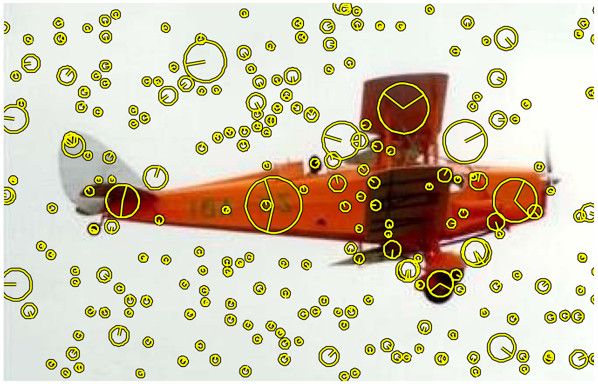
1. Sliding window detectors
2. Features and adding spatial information
    - Bag of visual word (BoW) models
    - Beyond BoW I: Constellation and ISM models
    - Beyond BoW II: Grids and spatial pyramids
3. Histogram of Oriented Gradients (HOG)
4. Two state of the art algorithms and PASCAL VOC
5. The future and challenges

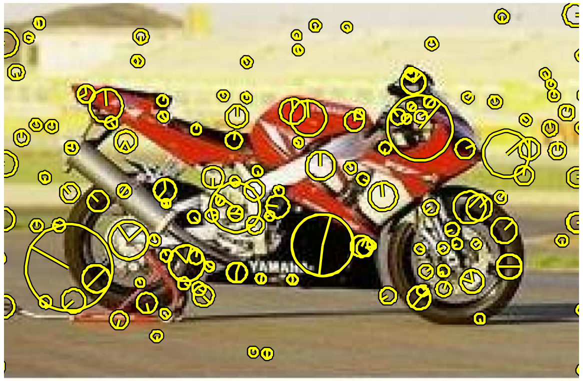
# Recap: Bag of (visual) Words representation

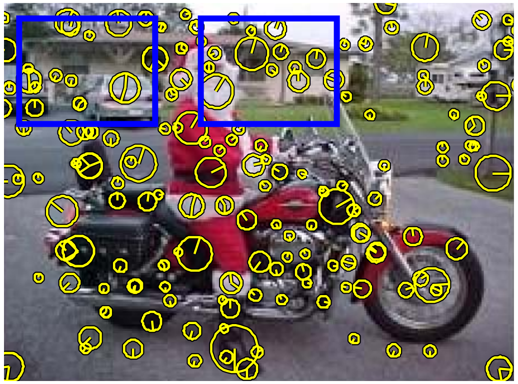
- Detect affine invariant local features (e.g. affine-Harris)

- Represent by high-dimensional descriptors, e.g. 128-D for SIFT

- How to summarize sliding window content in a fixed-length vector for classification?



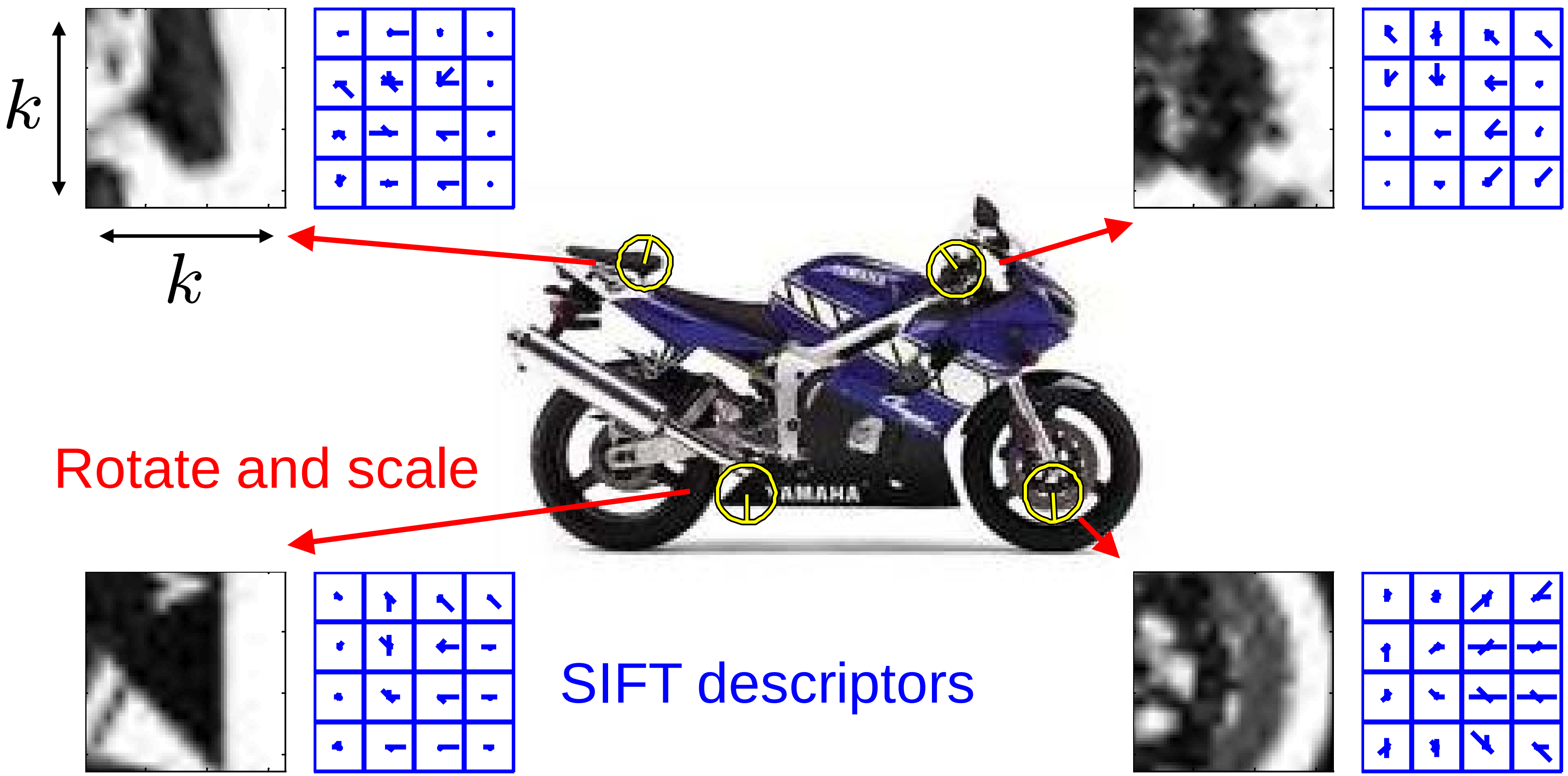




1. Map descriptors onto a common vocabulary of **visual words**

2. Represent image as a histogram over visual words – a **bag of words**

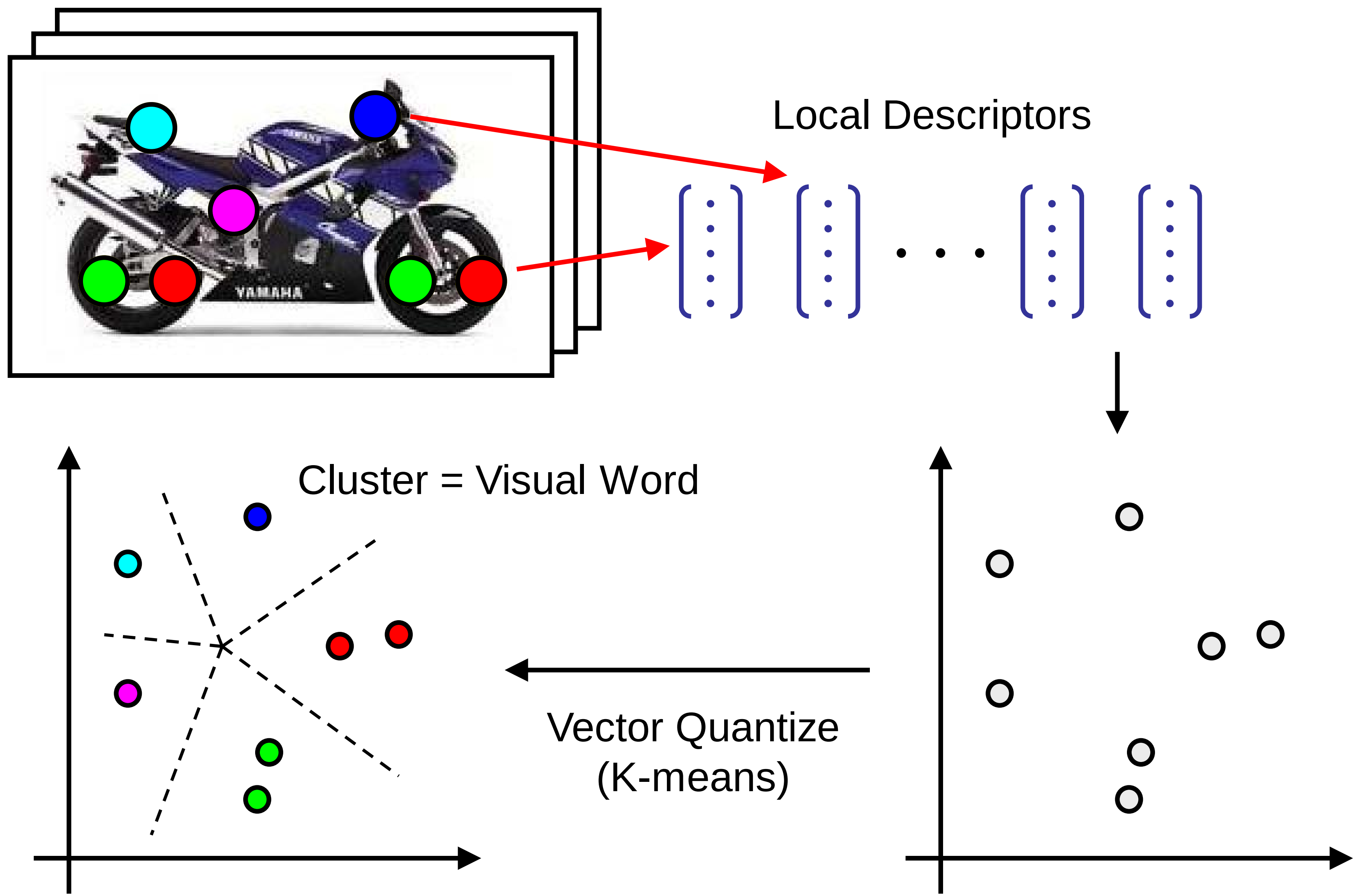
# Local region descriptors and visual words



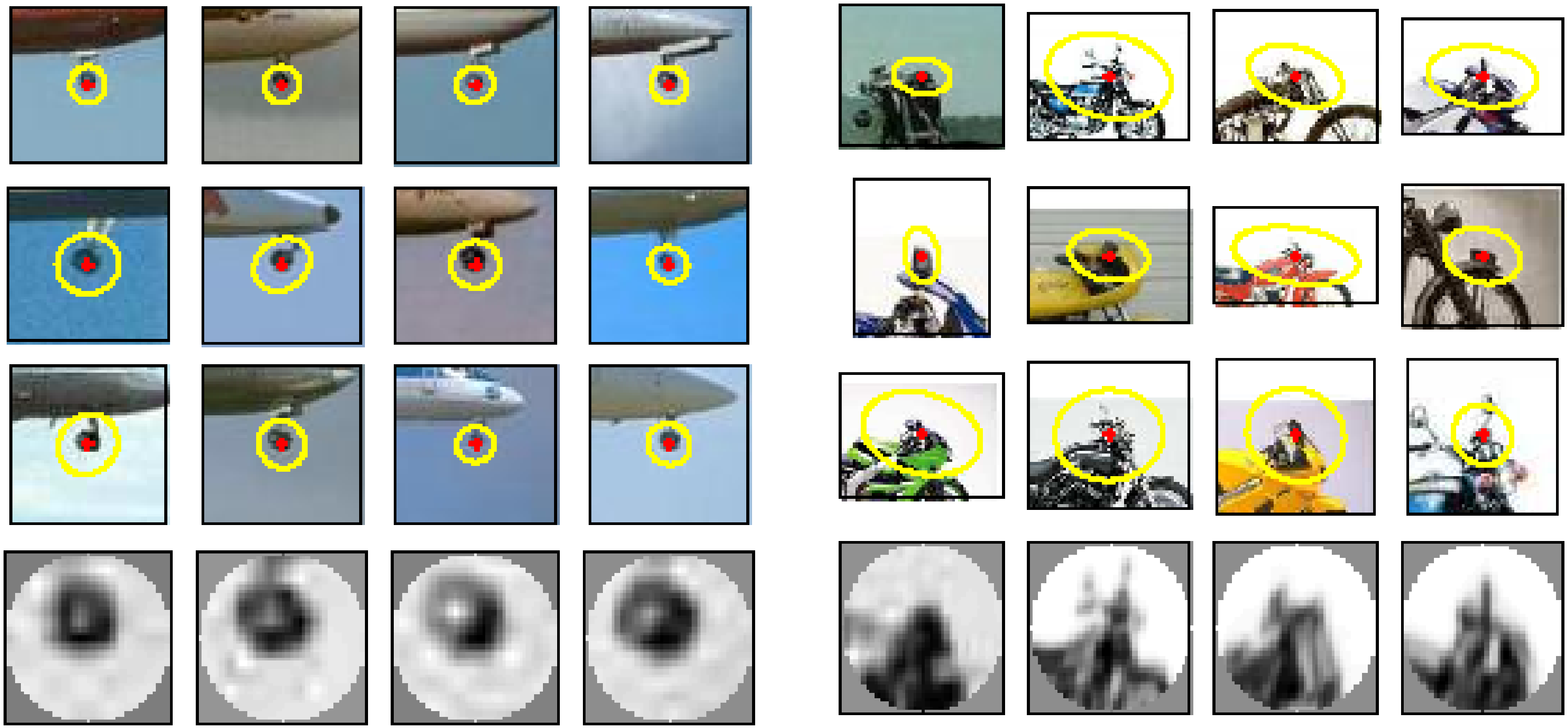

- Normalize regions to fixed size and shape
- Describe each region by a SIFT descriptor
- Vector quantize into visual words, e.g. using k-means

NB: aff. detectors/SIFT/visual words originally for view point invariant matching

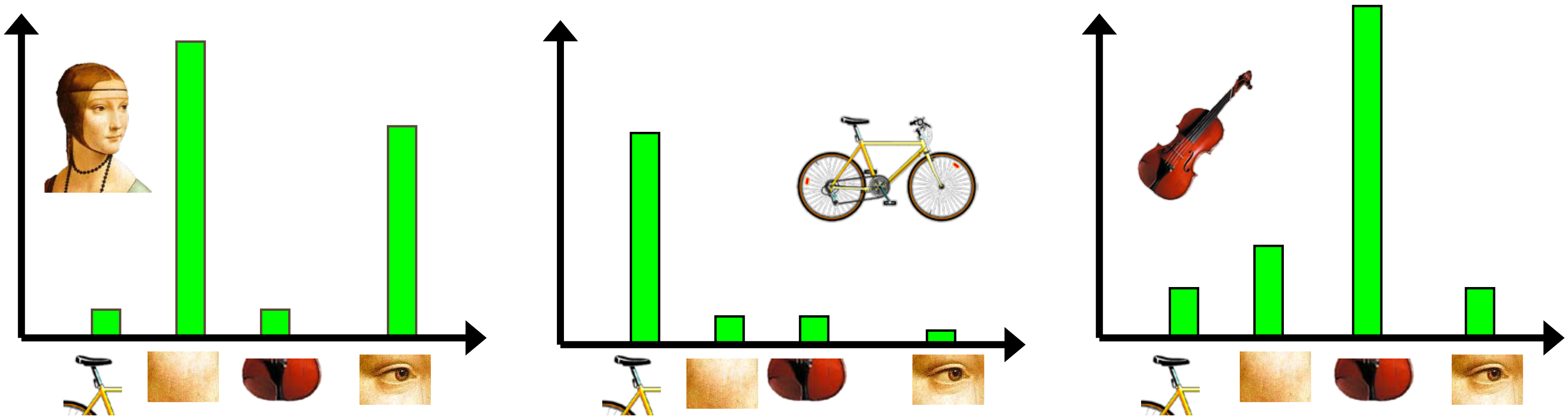
# Visual Words



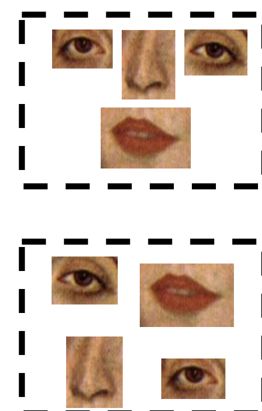
# Example Visual Words



# Intuition

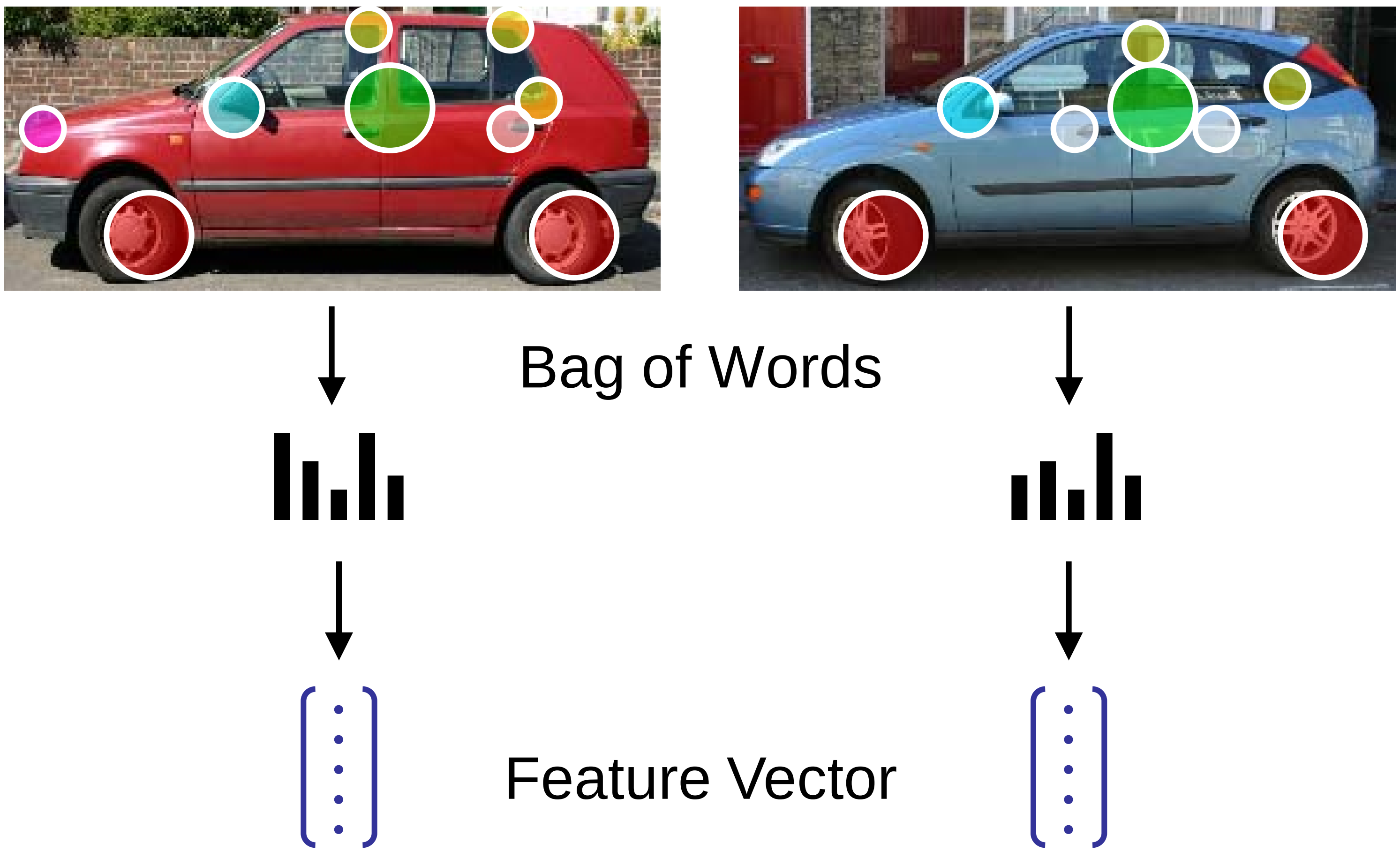


Visual Vocabulary



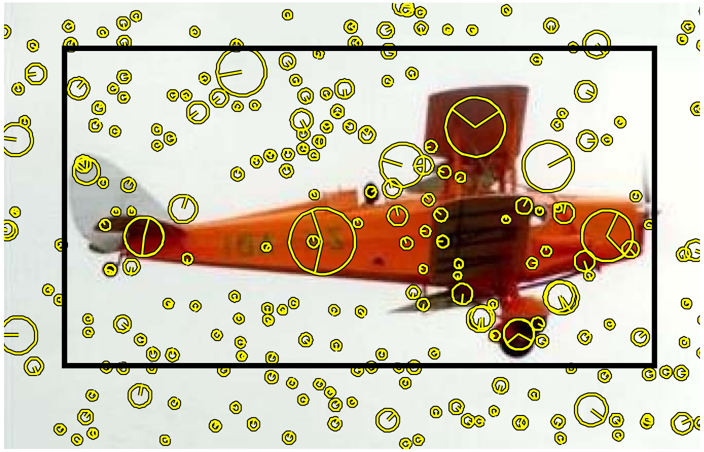
- Visual words represent “iconic” image fragments
- Feature detectors and SIFT give invariance to local rotation and scale
- Discarding spatial information gives configuration invariance

# Learning from positive ROI examples

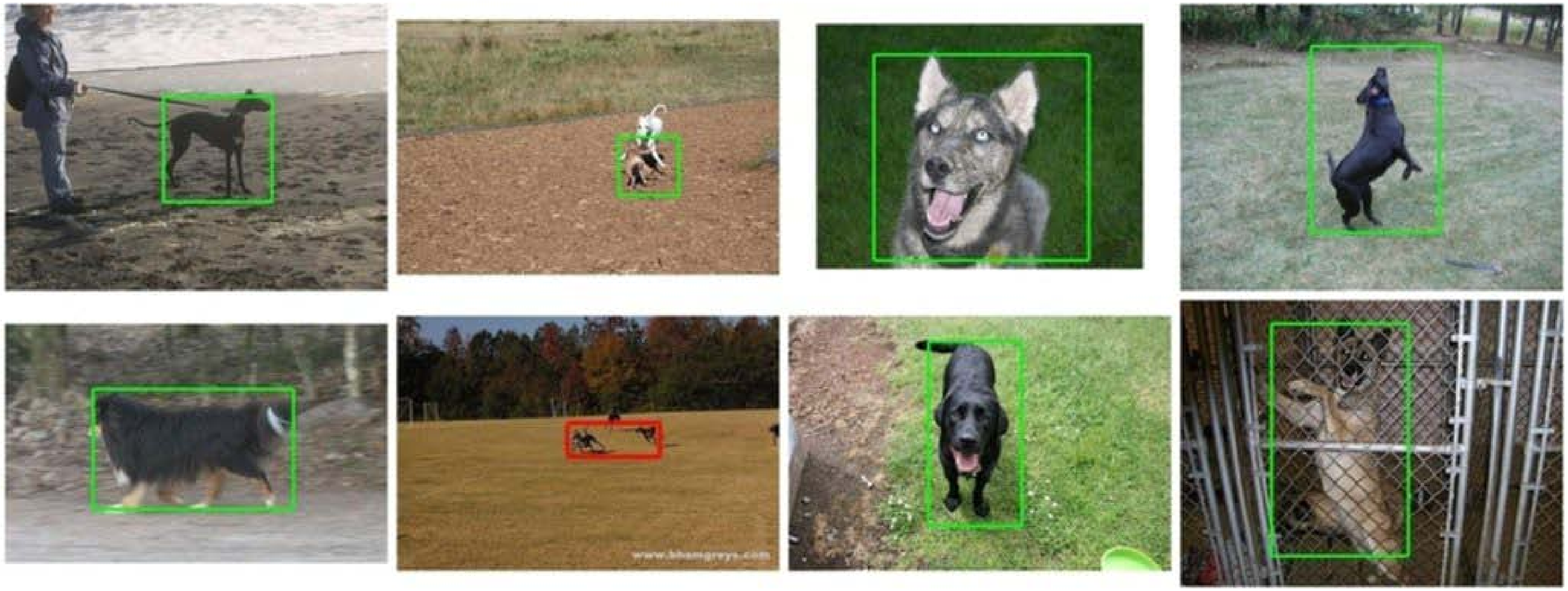


# Sliding window detector

- Classifier: SVM with linear kernel
- BOW representation for ROI



Example detections for dog



Lampert et al CVPR 08

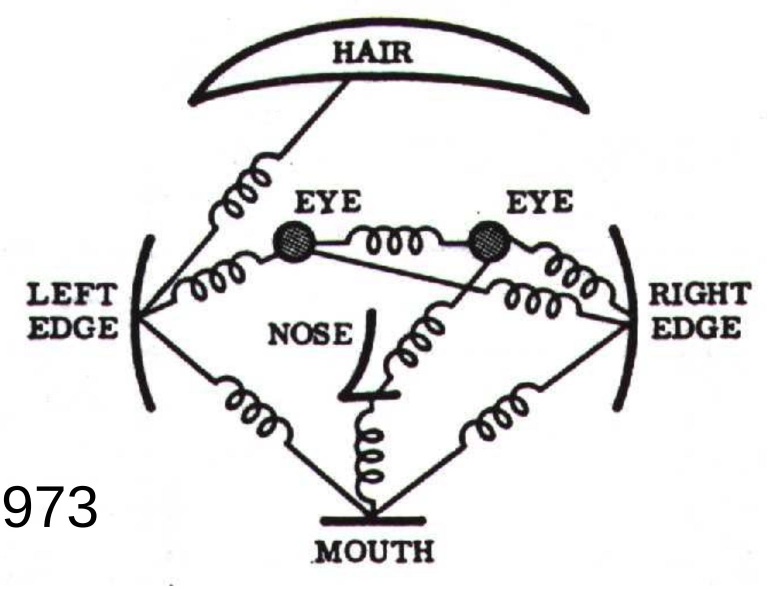
# Discussion: ROI as a Bag of Visual Words

- Advantages
  - No explicit modelling of spatial information ⇒ high level of invariance to position and orientation in image
  - Fixed length vector ⇒ standard machine learning methods applicable

- Disadvantages
  - No explicit modelling of spatial information ⇒ less discriminative power
  - Inferior to state of the art performance

# Beyond BOW I: Pictorial Structure

- Intuitive model of an object
- Model has two components
  1. parts (2D image fragments)
  2. structure (configuration of parts)
- Dates back to Fischler & Elschlager 1973

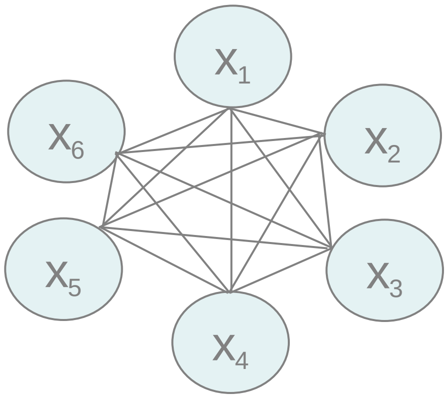



Two approaches that have investigated this spring like model:

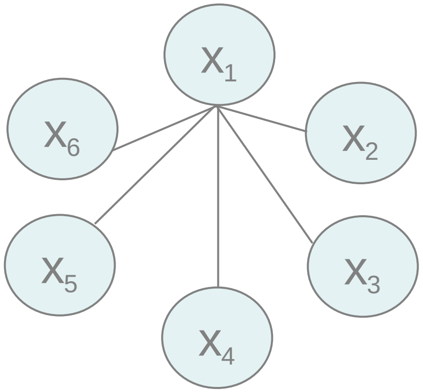
- Constellation model
- Implicit shape model

# Spatial Models Considered

Fully connected shape model




"Star" shape model



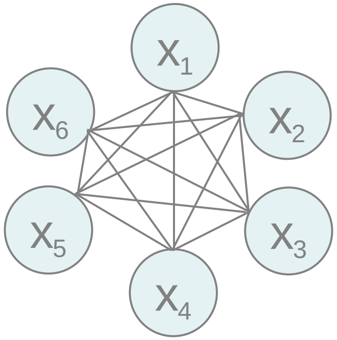

e.g. Constellation Model
Parts fully connected
Recognition complexity: $O(N^P)$
Method: Exhaustive search

e.g. ISM
Parts mutually independent
Recognition complexity: O(NP)
Method: Gen. Hough Transform

# Constellation model

Fergus, Perona & Zisserman,CVPR 03

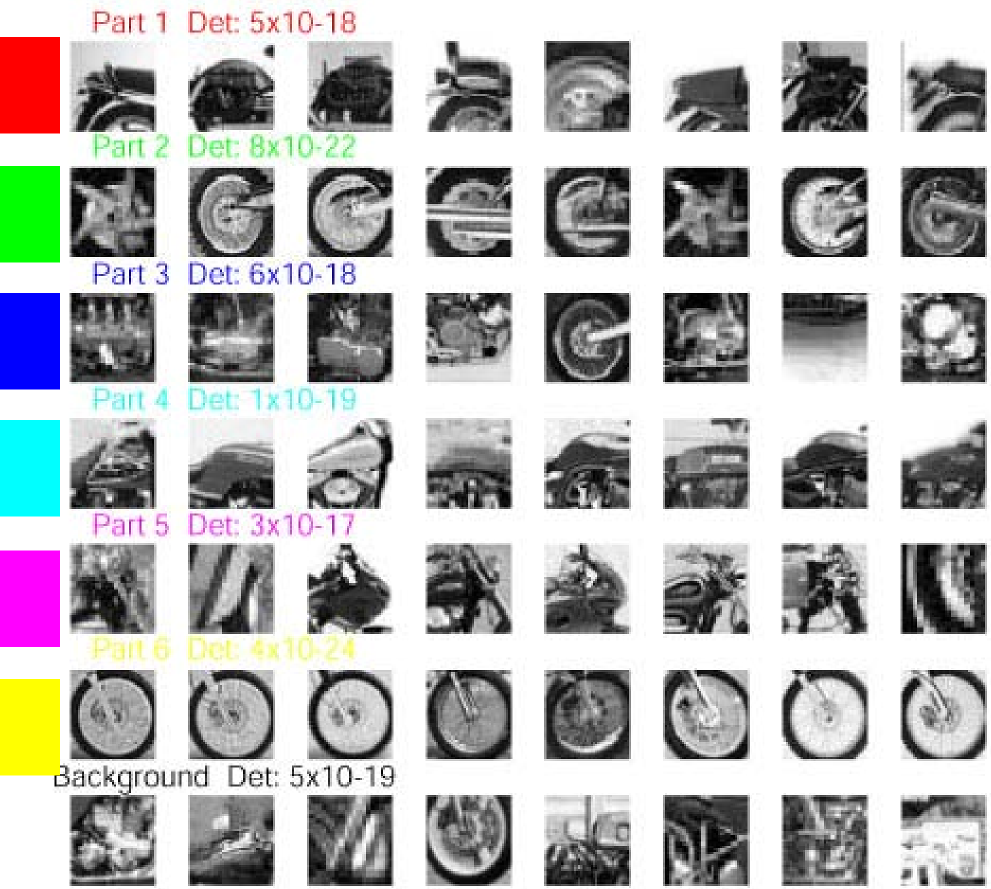
- Explicit structure model – Joint Gaussian over all part positions
- Part detector determines position *and* scale
- Simultaneous learning of parts and structure
- Learn from images alone using EM algorithm



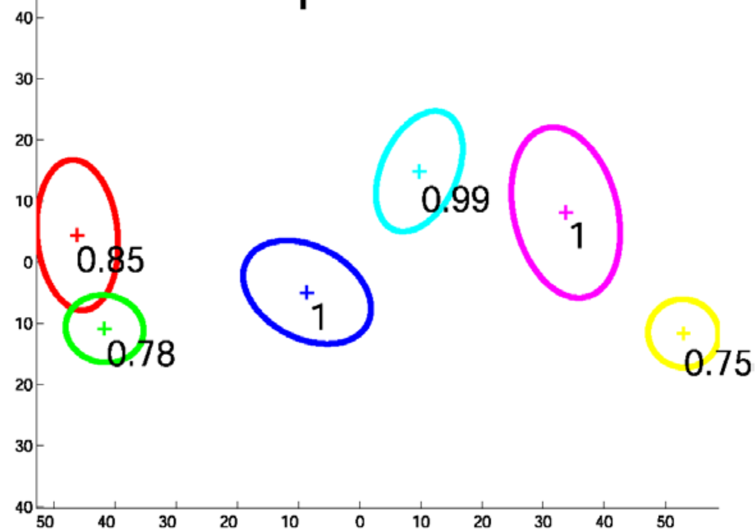

Given detections: learn a six part model by optimizing part and configuration similarity

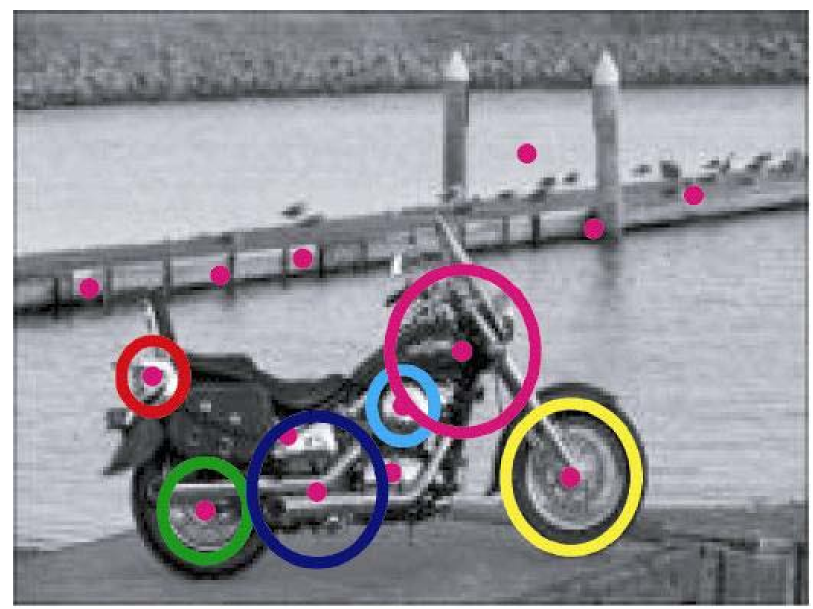
# Example – Learnt Motorbike Model

## Samples from appearance model

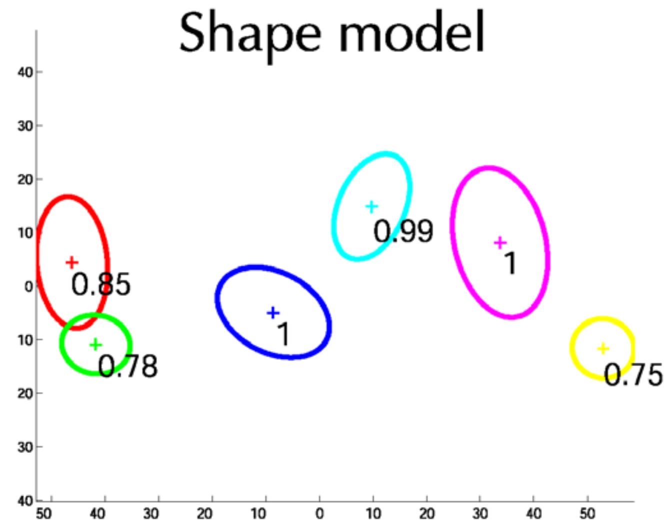


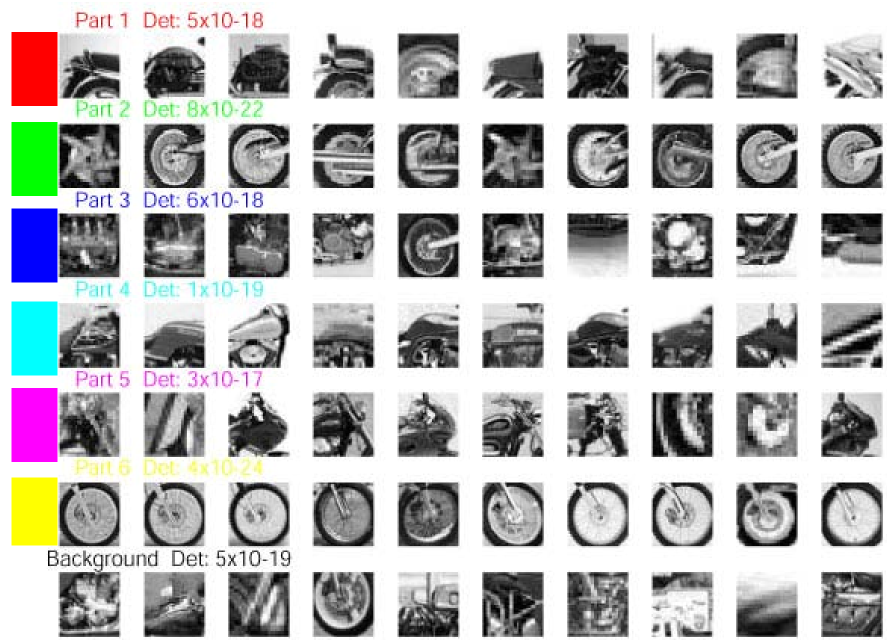

## Shape model



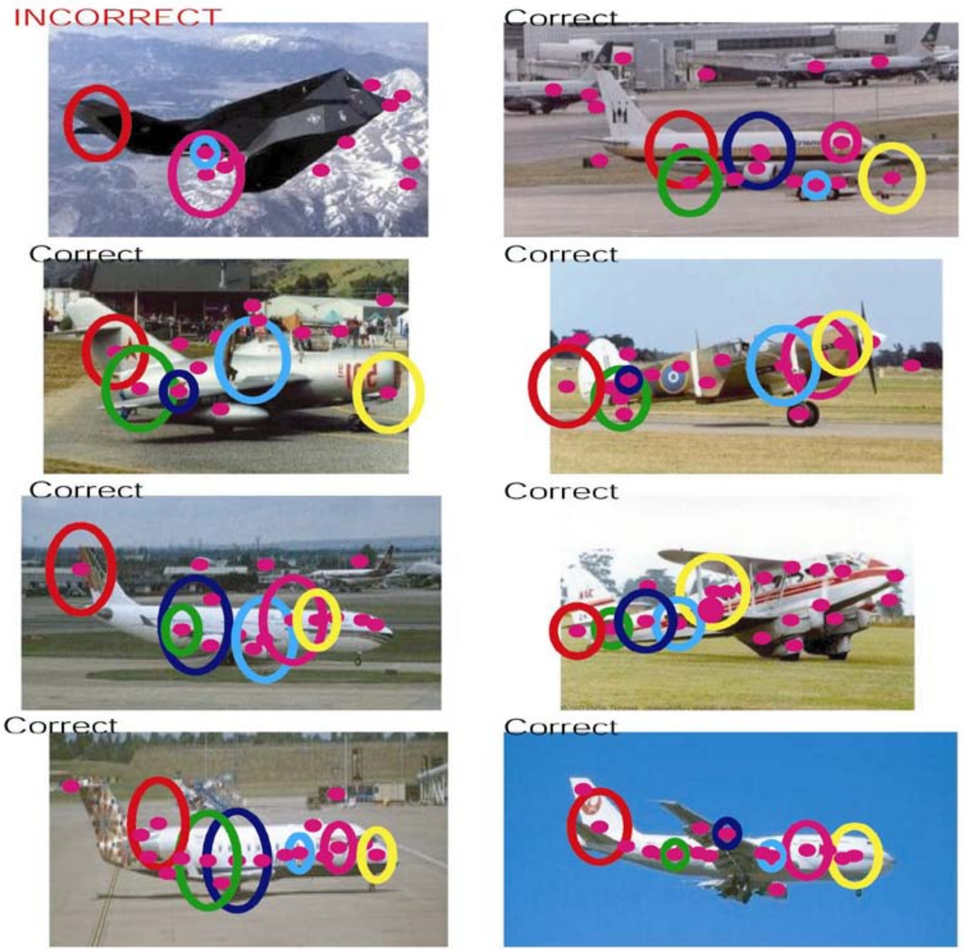
# Recognized Motorbikes



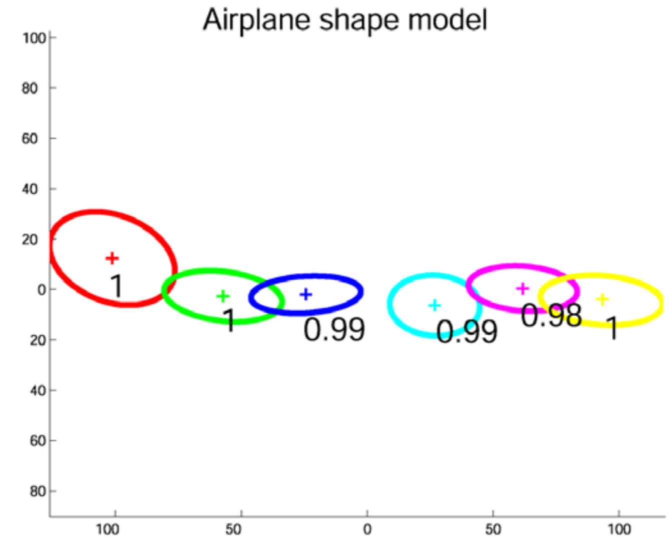




position of object determined

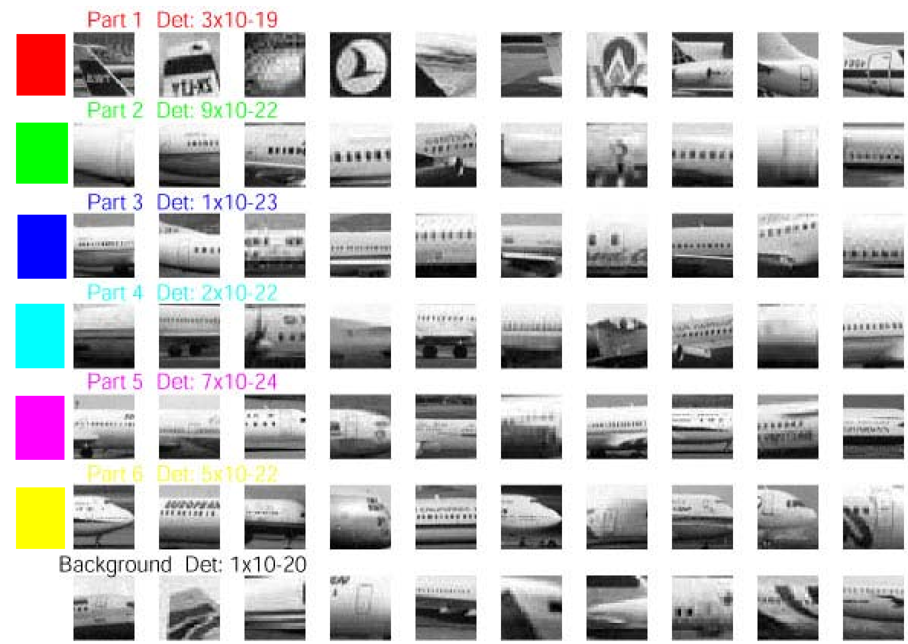
# Airplanes



INCORRECT
Correct
Correct
Correct
Correct
Correct
Correct
Correct

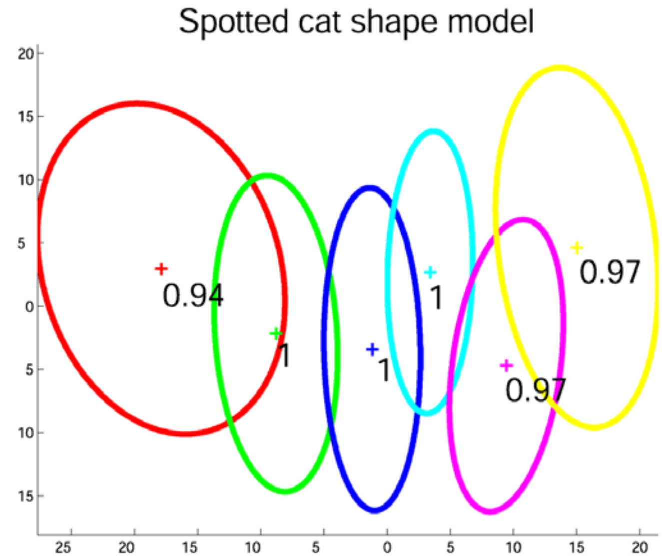



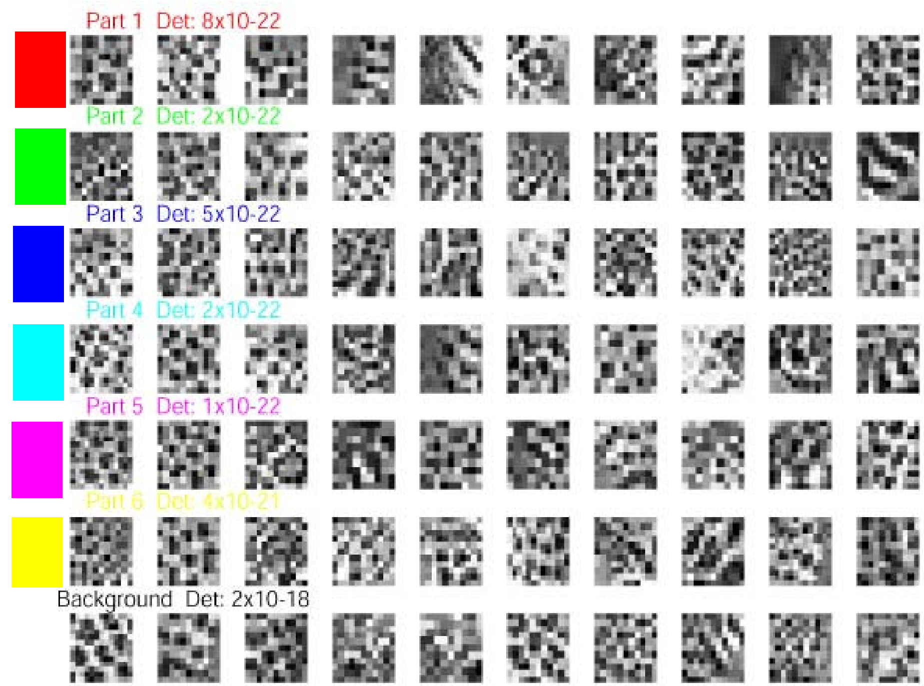
Airplane shape model
100
80
60
40
20
0
20
40
60
80
100
50
0
50
100
1
1
0.99
0.99
0.98
1

Part 1 Det: 3x10-19
Part 2 Det: 9x10-22
Part 3 Det: 1x10-23
Part 4 Det: 2x10-22
Part 5 Det: 7x10-24
Part 6 Det: 5x10-22
Background Det: 1x10-20

# Spotted cats



# Discussion: Constellation Model

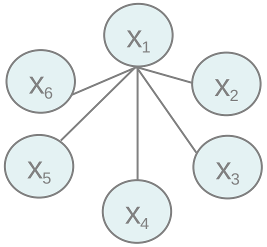
- Advantages
  - Works well for many different object categories
  - Can adapt well to categories where
    - Shape is more important
    - Appearance is more important
  - Everything is learned from training data
  - Weakly-supervised training possible

- Disadvantages
  - Model contains many parameters that need to be estimated
  - Cost increases exponentially with increasing number of parameters

  $\Rightarrow$ Fully connected model restricted to small number of parts.

# Implicit Shape Model (ISM)

**Leibe, Leonardis, Schiele, 03/04**



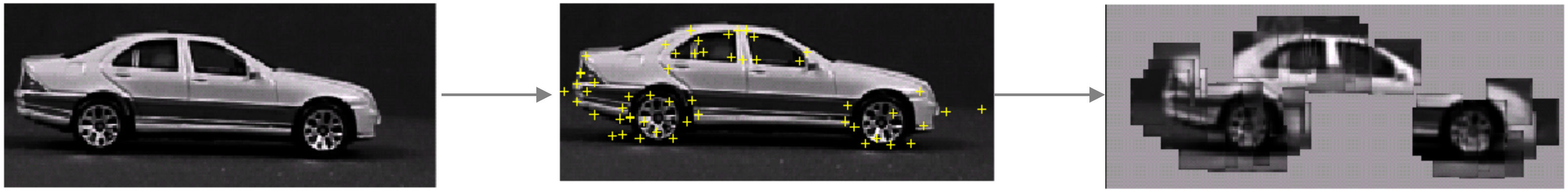

- Basic ideas
  - Learn an appearance codebook
  - Learn a star-topology structural model
    - Features are considered independent given object centre

- Algorithm: probabilistic Generalized Hough Transform

  Good engineering:
  - Soft assignment
  - Probabilistic voting
  - Continuous Hough space

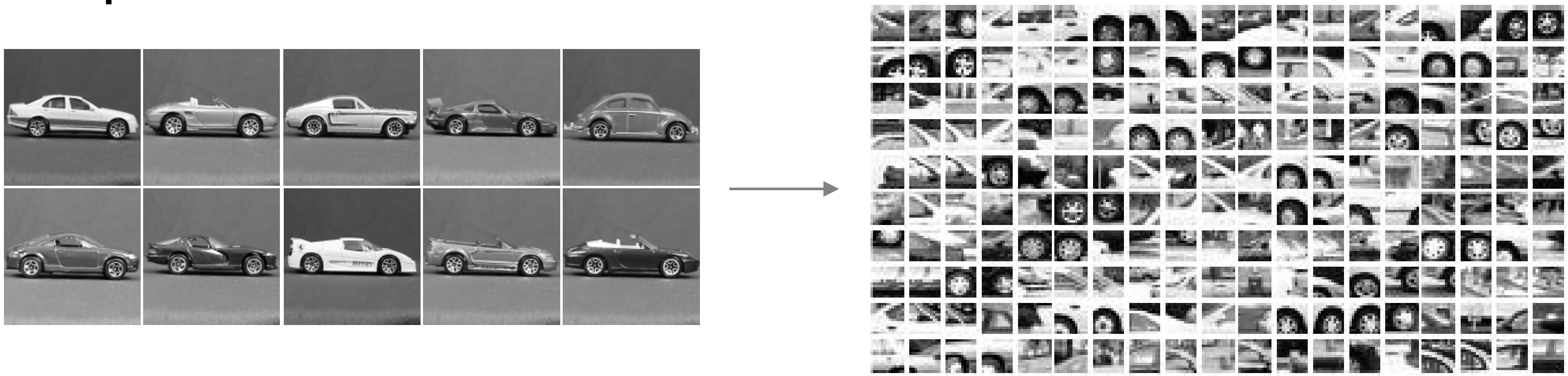
# Codebook Representation

- Extraction of local object features
  - Interest Points (e.g. Harris detector)
  - Sparse representation of the object appearance



- Collect features from whole training set

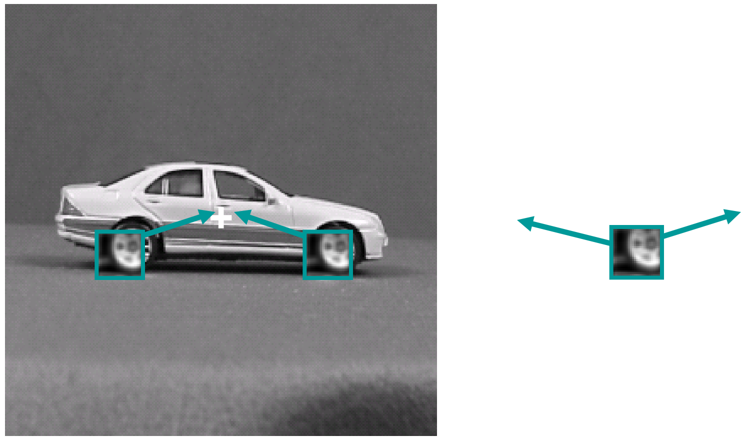
- Example:



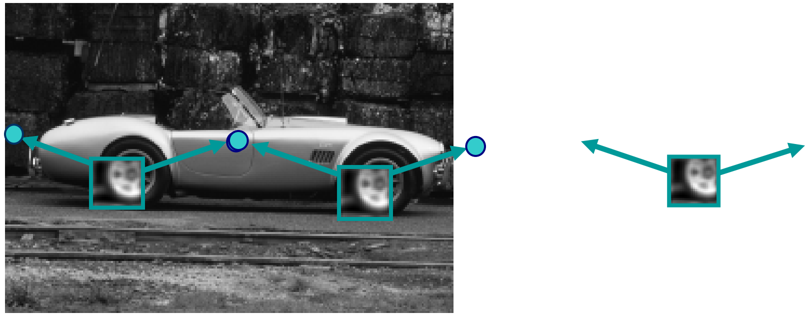
Class specific vocabulary

# Leibe & Schiele 03/04: Generalized Hough Transform

- Learning: for every cluster, store possible “occurrences”

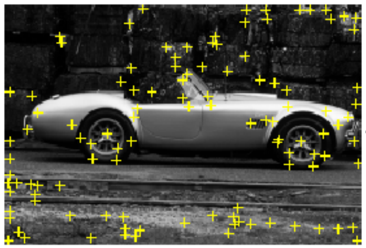


- Recognition: for new image, let the matched patches vote for possible object positions

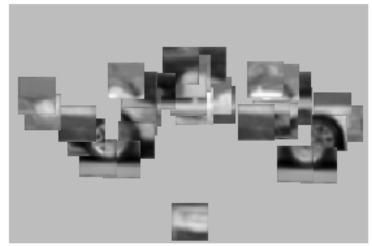


# Leibe & Schiele 03/04: Generalized Hough Transform

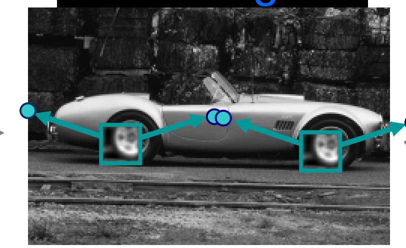
Interest Points

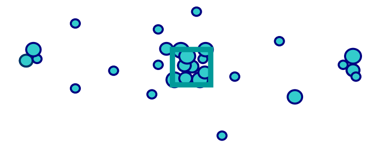


Matched Codebook Entries

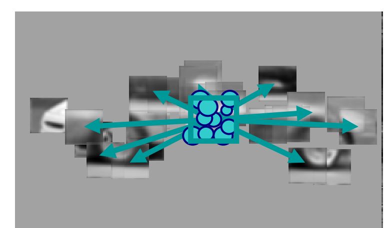


Probabilistic Voting



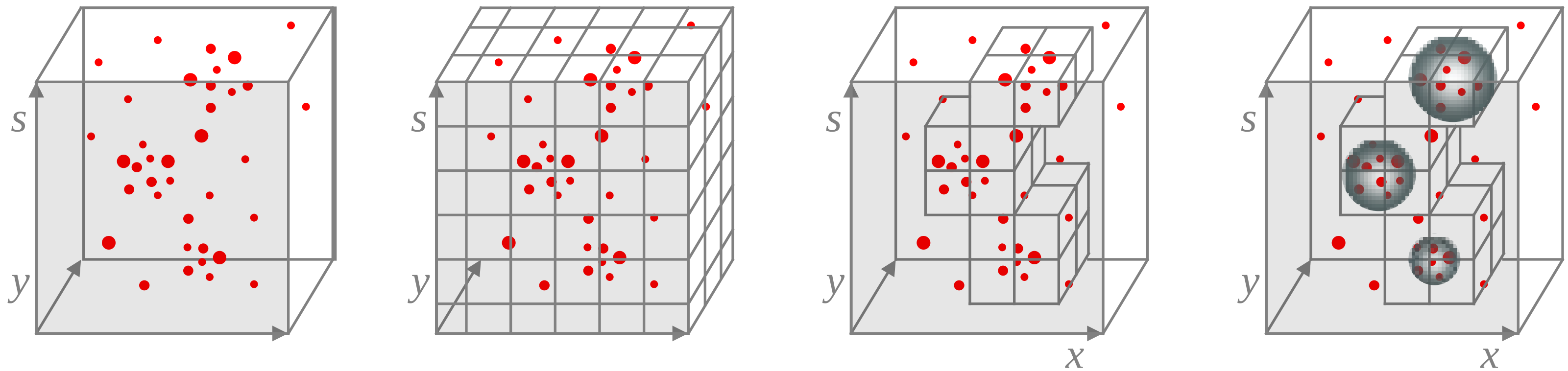


Voting Space (continuous)



Backprojection of Maximum

# Scale Voting: Efficient Computation




Scale votes

Binned accum. array

Candidate maxima

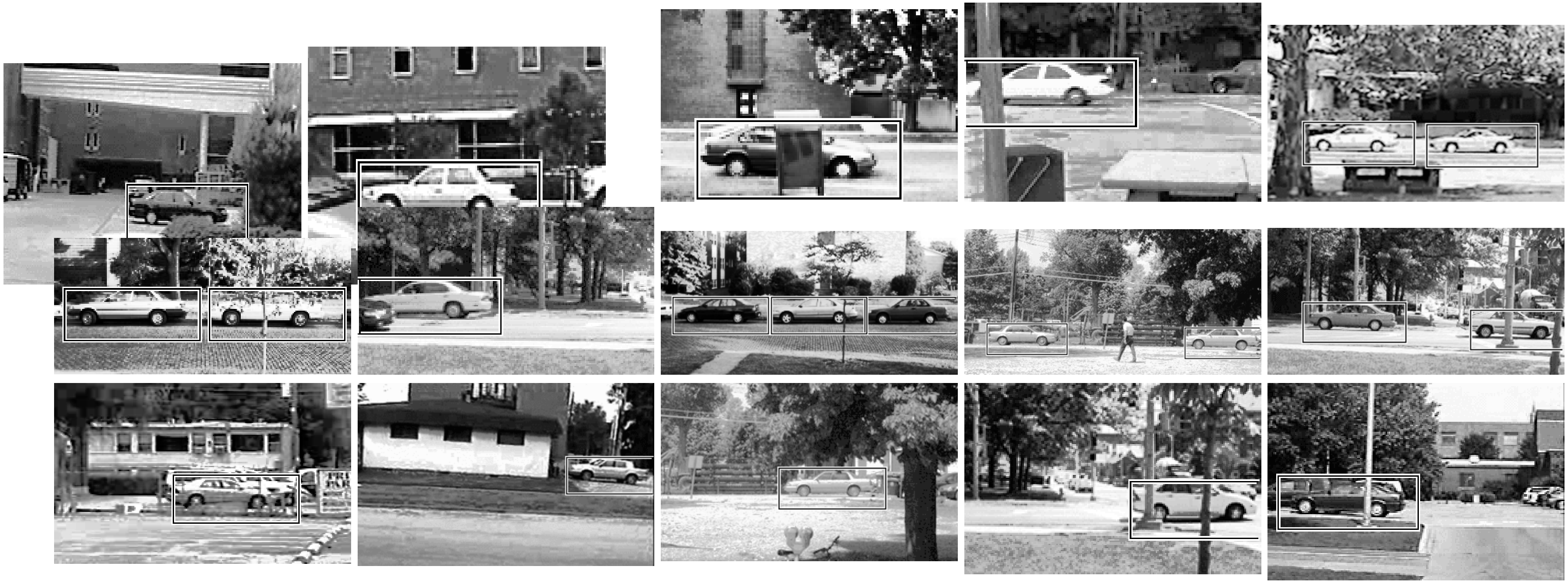
Refinement (MSME)

- Mean-Shift formulation for refinement
  - Scale-adaptive *balloon density estimator*

$$\hat{p}(o_n, x) = \frac{1}{V_b} \sum_k \sum_j p(o_n, x_j | f_k, \ell_k) K\left(\frac{x - x_j}{b}\right)$$

# Detection Results

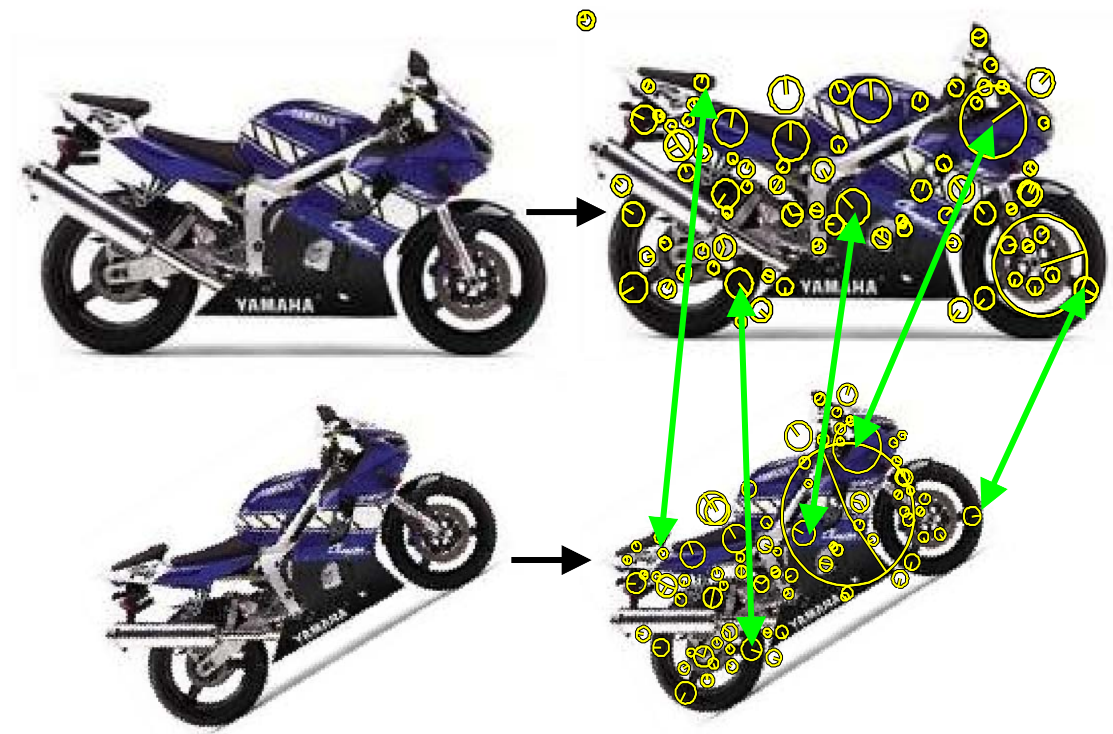
- Qualitative Performance
  - Recognizes different kinds of cars
  - Robust to clutter, occlusion, low contrast, noise



# Discussion: ISM and related models

## Advantages

- Scale and rotation invariance can be built into the representation from the start
- Relatively cheap to learn and test (inference)
- Works well for many different object categories
- Max-margin extensions possible, Maji & Malik, CVPR09



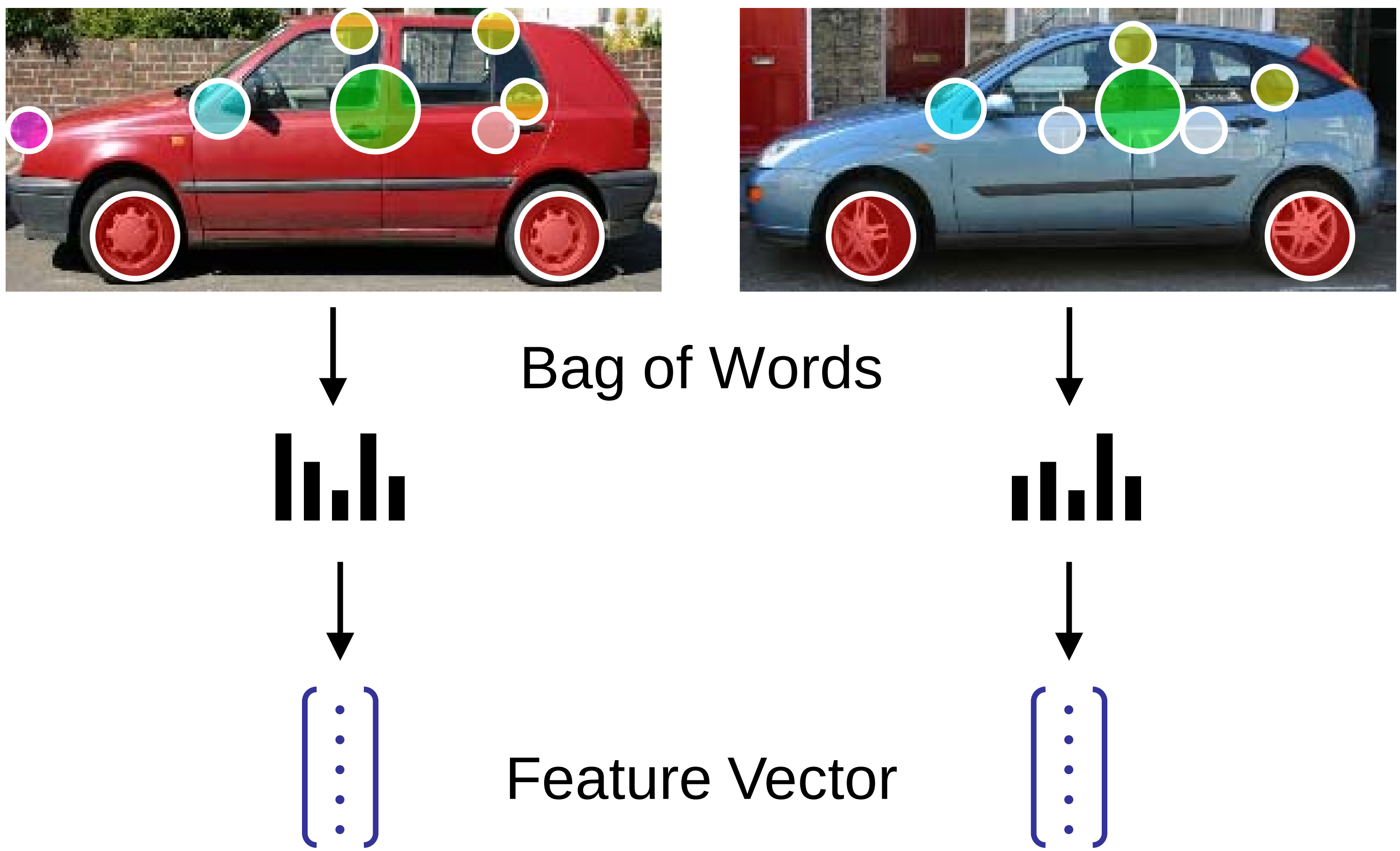

## Disadvantages

- Requires searching for modes in the Hough space
- Similar to sliding window in this respect
- Is such a degree of invariance required? (many objects are horizontal)

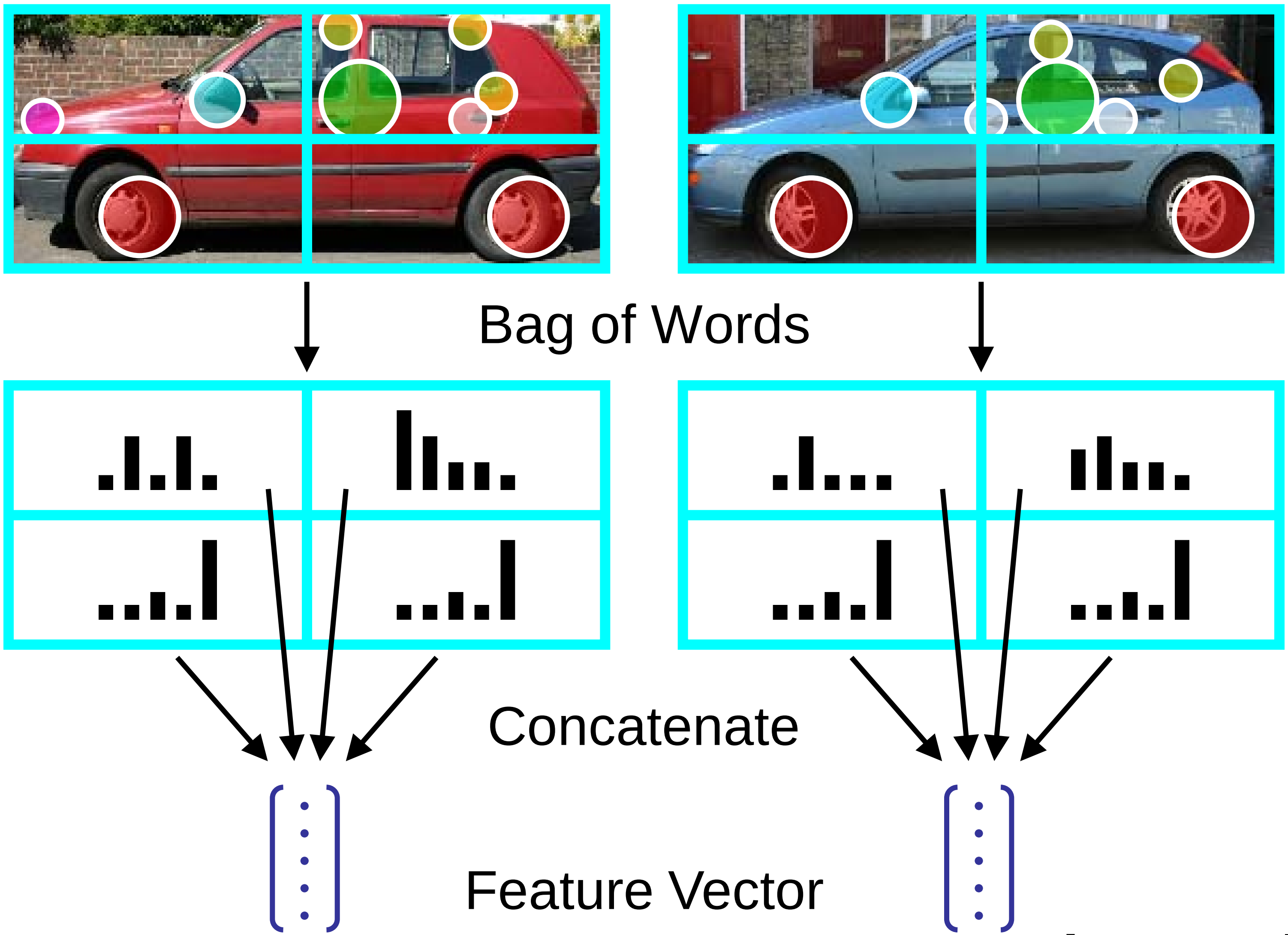
# Beyond BOW II: Grids and spatial pyramids

Start from BoW for ROI

- no spatial information recorded
- sliding window detector



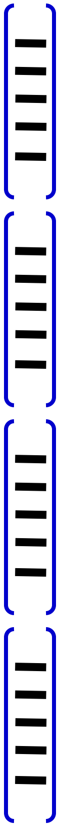
# Adding Spatial Information to Bag of Words



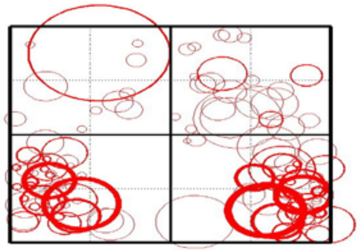

[Fergus et al, 2005]

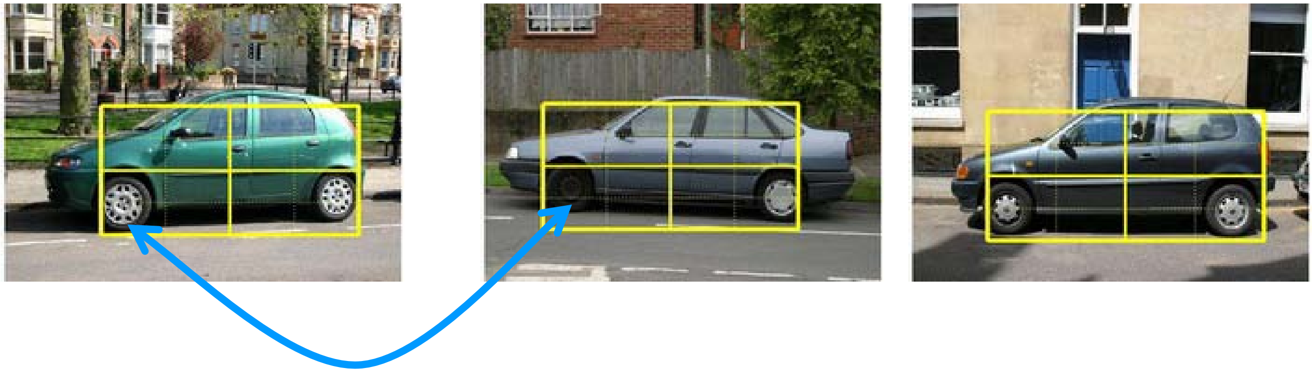
Keeps fixed length feature vector for a window

# Tiling defines (records) the spatial correspondence of the words



- parameter: number of tiles

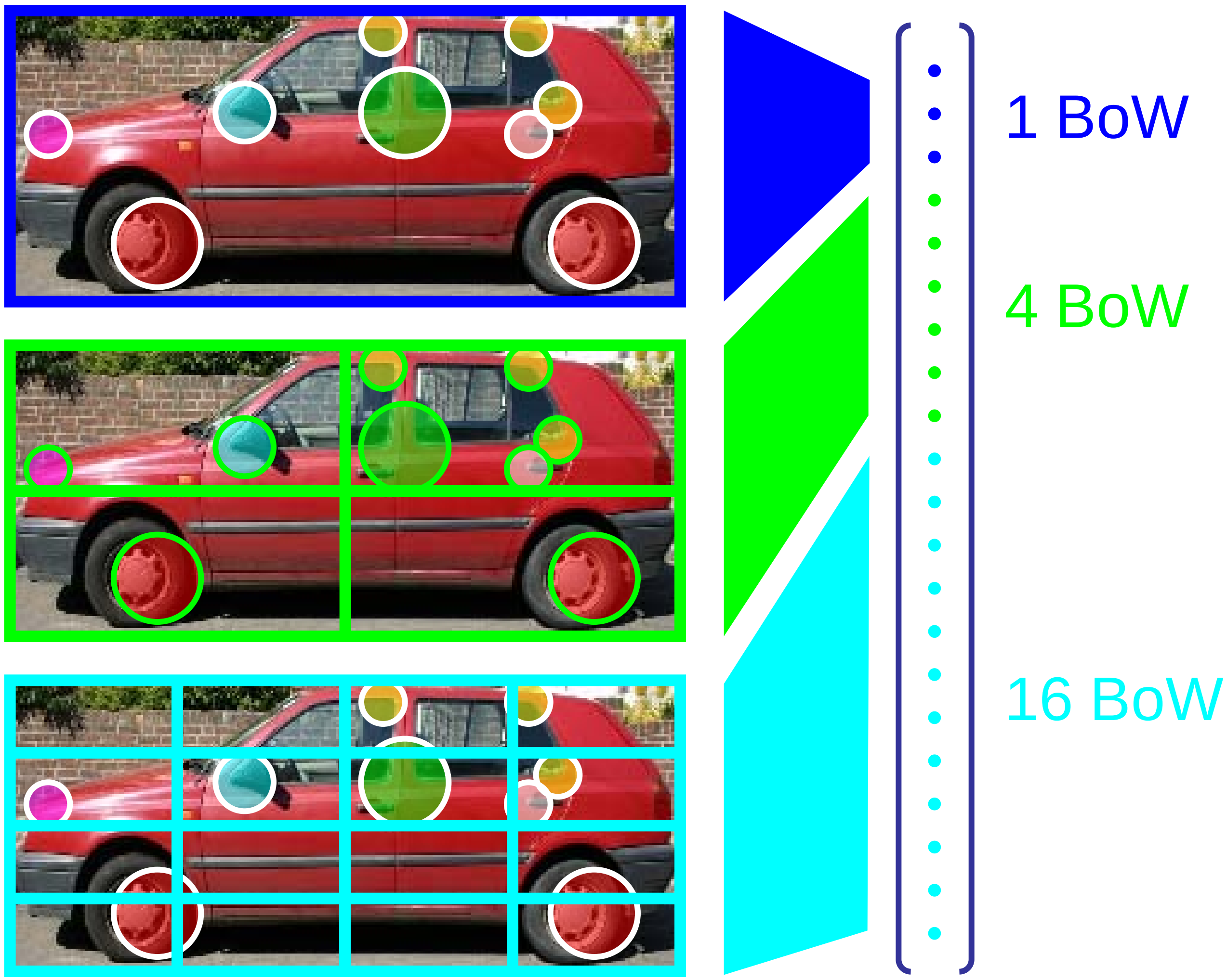




If codebook  has V visual words, then representation has dimension 4V

Fergus et al ICCV 05

# Spatial Pyramid – represent correspondence

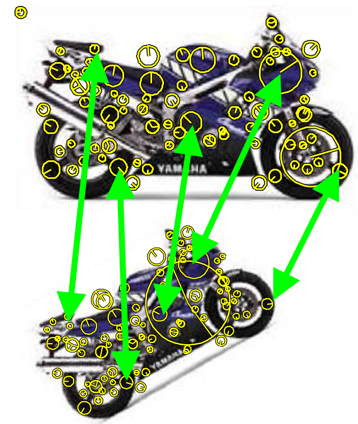



- As in scene/image classification can use pyramid kernel

[Grauman & Darrell, 2005] [Lazebnik et al, 2006]

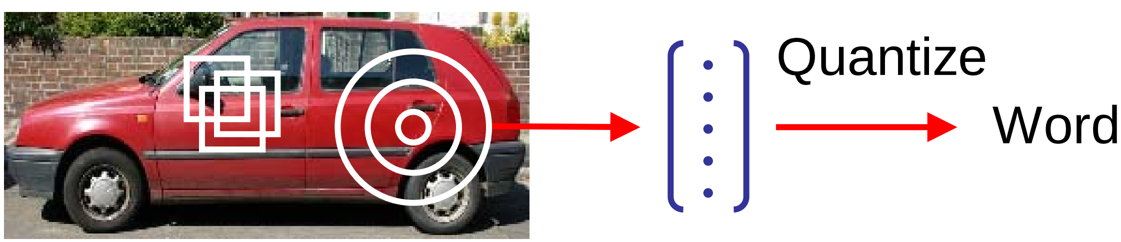
# Dense Visual Words

- Why extract only **sparse** image fragments?



- Good where lots of invariance is needed, but not relevant to sliding window detection?

- Extract **dense** visual words on an overlapping grid




[Luong & Malik, 1999]
[Varma & Zisserman, 2003]
[Vogel & Schiele, 2004]
[Jurie & Triggs, 2005]
[Fei-Fei & Perona, 2005]
[Bosch et al, 2006]

- More “detail” at the expense of invariance
- Pyramid histogram of visual words (PHOW)

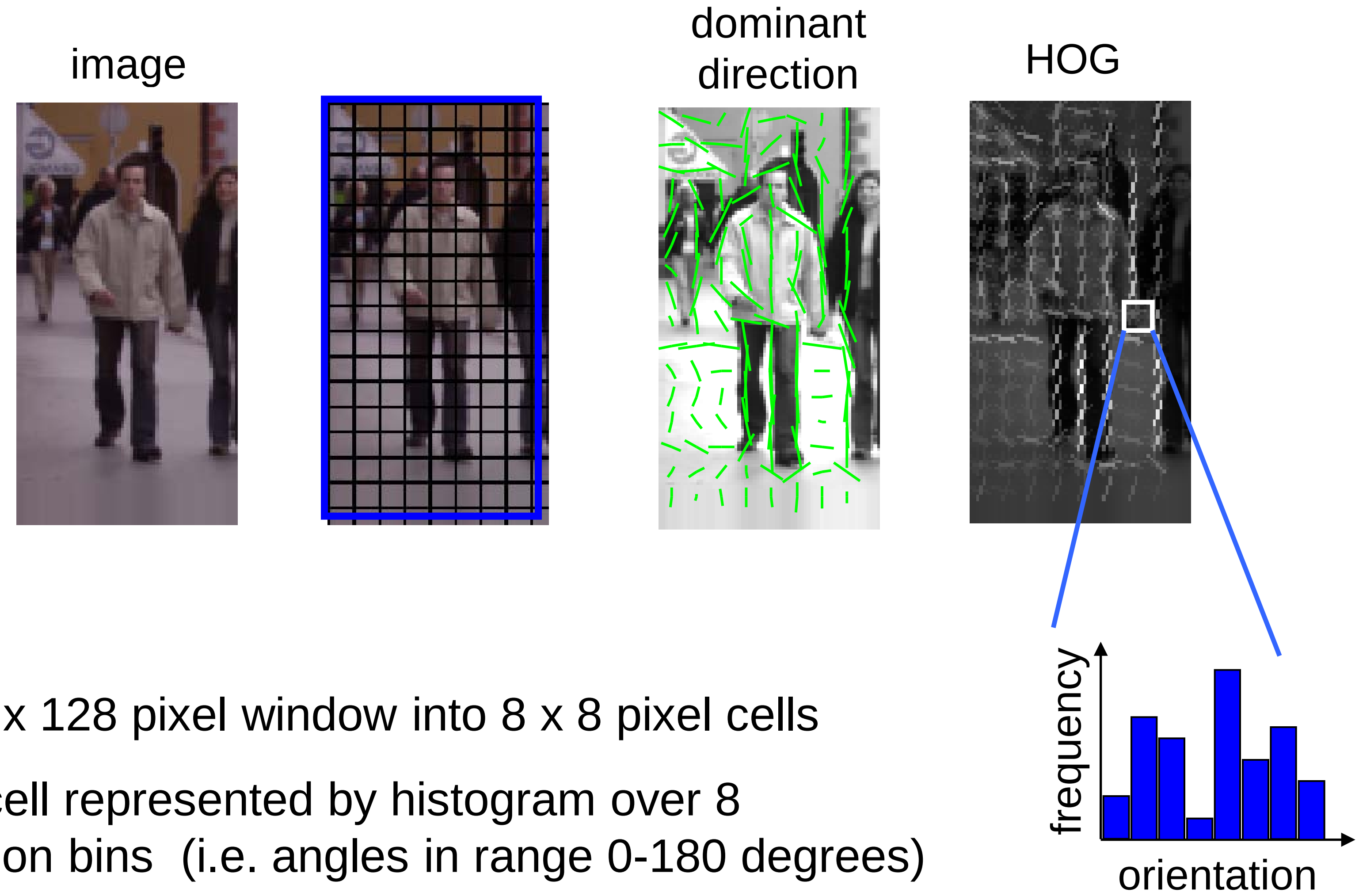
# Outline

1. Sliding window detectors
2. Features and adding spatial information
3. Histogram of Oriented Gradients + linear SVM classifier
   - Dalal & Triggs pedestrian detector
   - HOG and history
   - Training an object detector
4. Two state of the art algorithms and PASCAL VOC
5. The future and challenges

# Dalal & Triggs CVPR 2005 Pedestrian detection

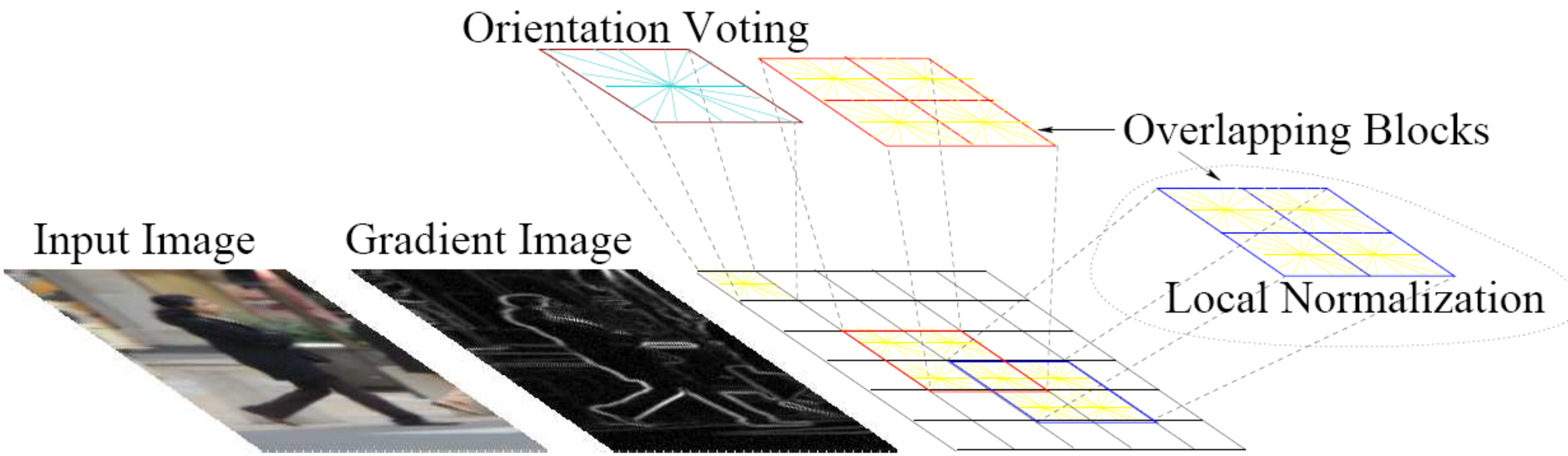
- Objective: detect (localize) standing humans in an image
- sliding window classifier
- train a binary classifier on whether a window contains a standing person or not
- Histogram of Oriented Gradients (HOG) feature
- although HOG + SVM originally introduced for pedestrians has been used very successfully for many object categories

# Feature: Histogram of Oriented Gradients (HOG)




- tile 64 x 128 pixel window into 8 x 8 pixel cells
- each cell represented by histogram over 8 orientation bins (i.e. angles in range 0-180 degrees)

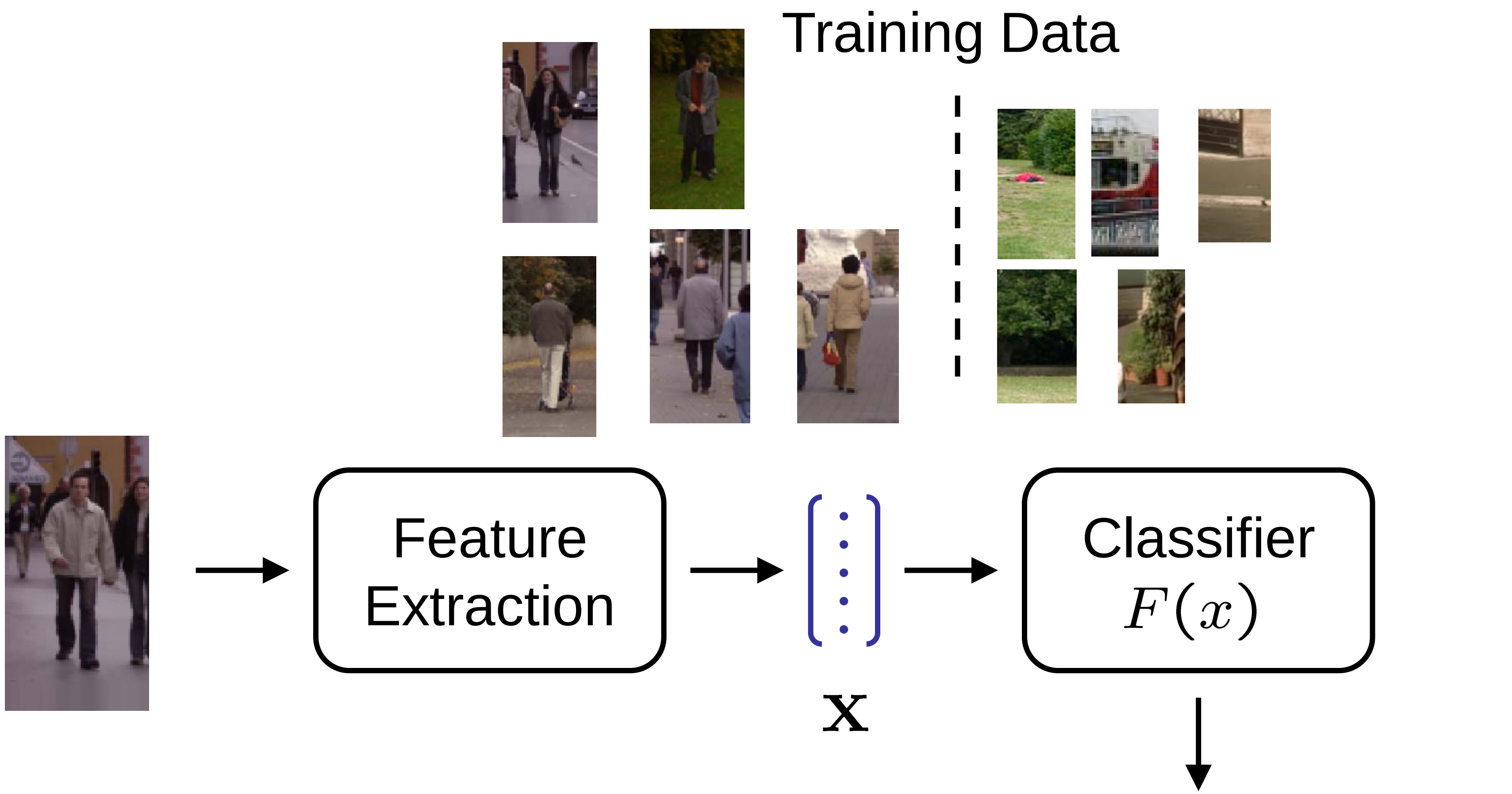
# Histogram of Oriented Gradients (HOG) continued




- Adds a second level of overlapping spatial bins re-normalizing orientation histograms over a larger spatial area

- Feature vector dimension (approx) = 16 x 8 (for tiling) x 8 (orientations) x 4 (for blocks) = 4096

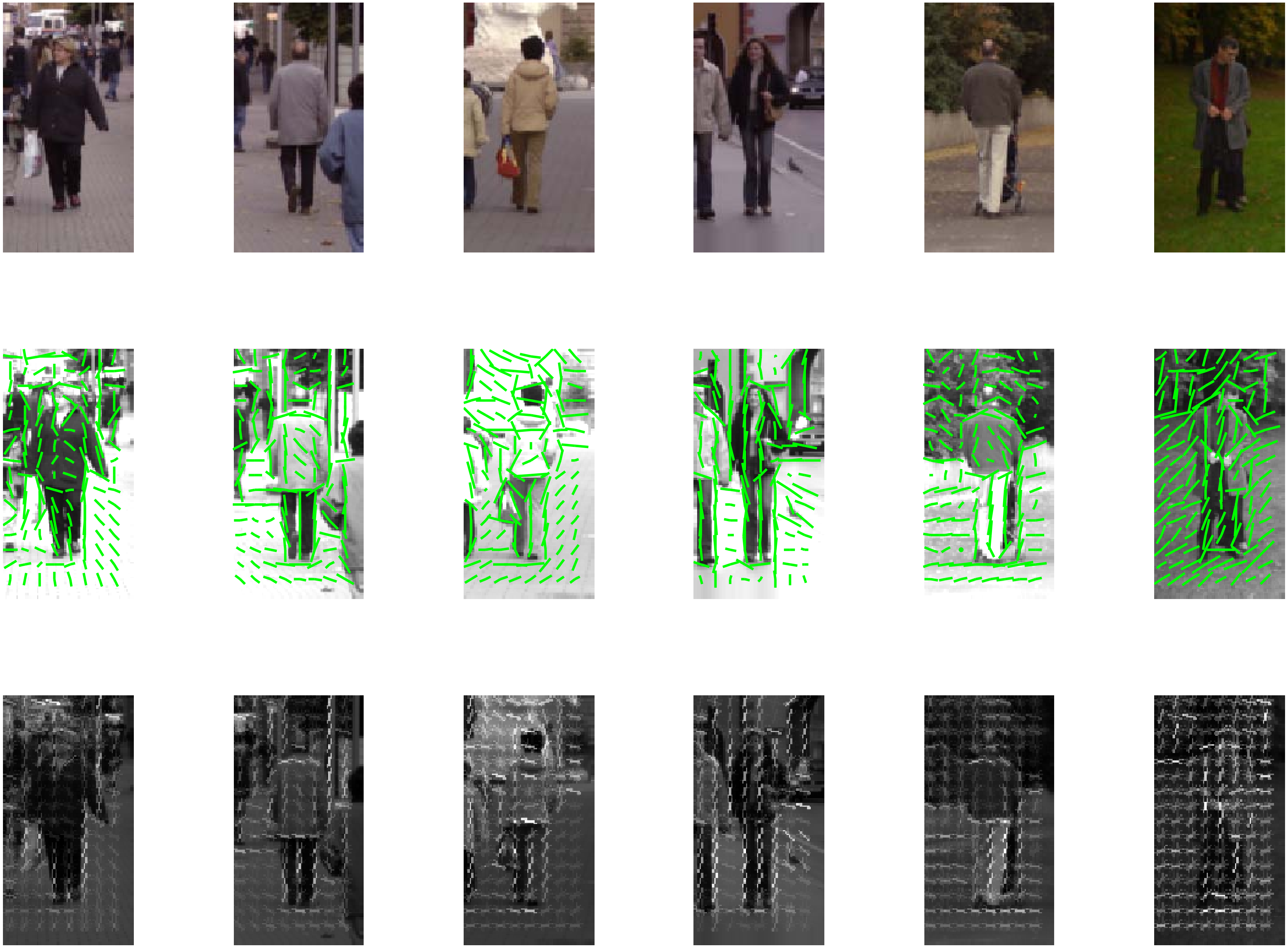
# Window (Image) Classification



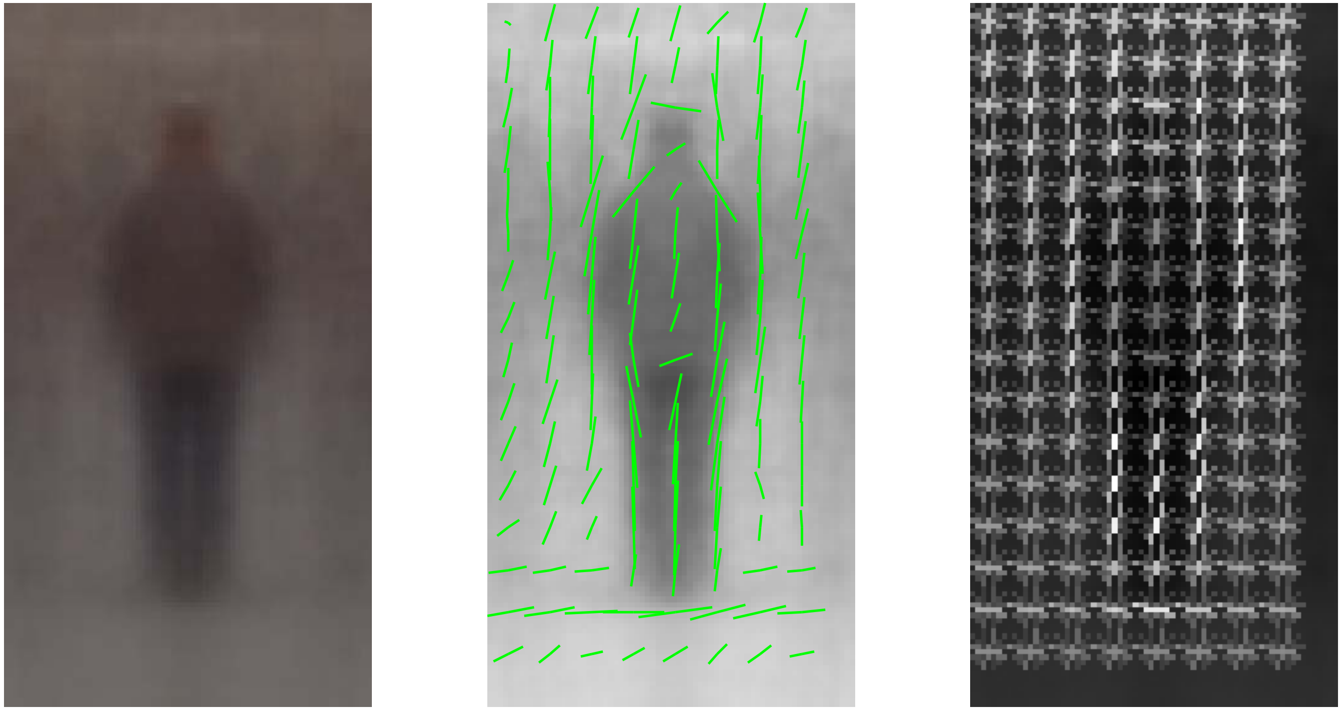

pedestrian/Non-pedestrian

$$P(c|\mathbf{x}) \propto F(\mathbf{x})$$

- HOG Features
- Linear SVM classifier



# Averaged examples



# Classifier: linear SVM

Advantages of linear SVM: $$\mathrm{f}(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$

- Training (Learning)
  - Very efficient packages for the linear case, e.g. LIBLINEAR for batch training and Pegasos for on-line training.
  - Complexity O(N) for N training points (cf O(N^3) for general SVM)
- Testing (Detection)

Non-linear $$\mathrm{f}(\mathbf{x}) = \sum_i^S \alpha_i \mathrm{k}(\mathbf{x}_i, \mathbf{x}) + b$$

S = # of support vectors

= (worst case ) N

size of training data

linear $$\mathrm{f}(\mathbf{x}) = \sum_i^S \alpha_i \mathbf{x}_i{}^T \mathbf{x} + b$$

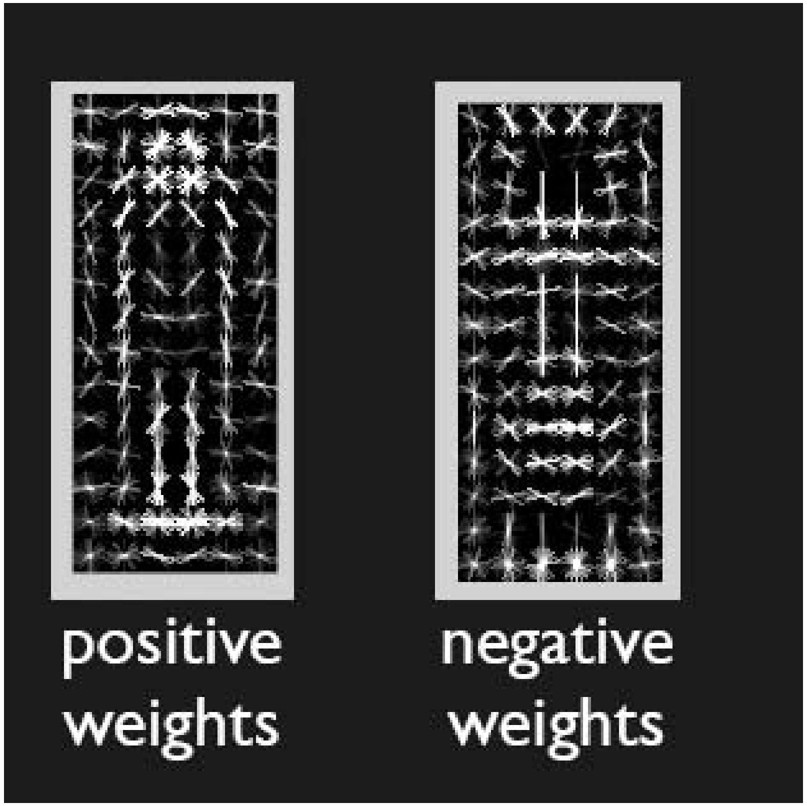
$$= \mathbf{w}^T \mathbf{x} + b$$

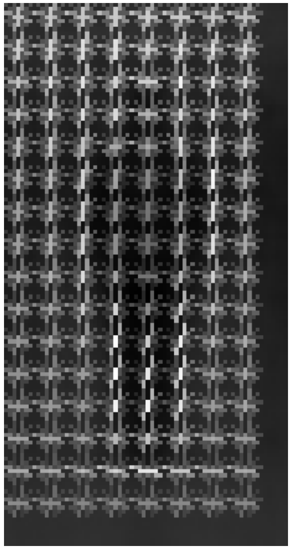
Independent of size of training data


Dalal and Triggs, CVPR 2005

# Learned model

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$



average over
positive training data

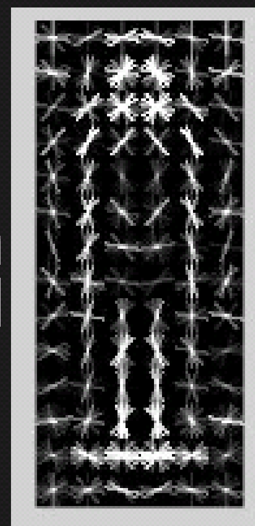
# What do negative weights mean?

$$wx > 0$$

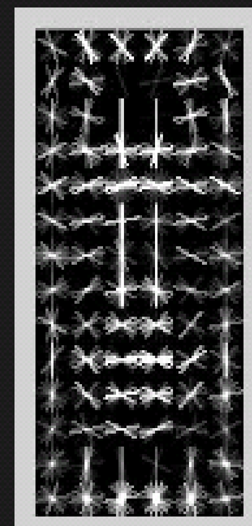
$$(w_{+} - w_{-})x > 0$$

$$w_{+} > w_{-}x$$

pedestrian
model



>



pedestrian
background
model

Complete system should compete pedestrian/pillar/doorway models

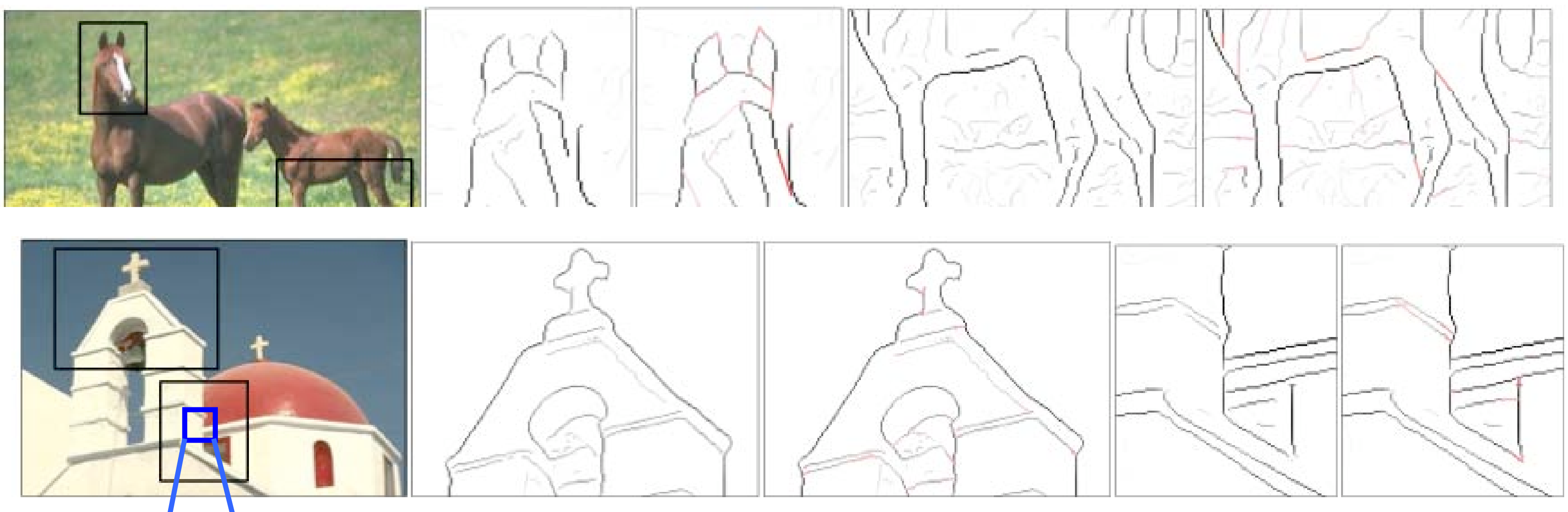
Discriminative models come equipped with own bg

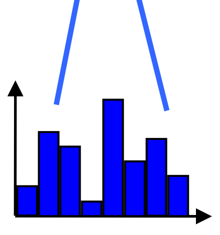
(avoid firing on doorways by penalizing vertical edges)

Slide from Deva Ramanan

# Why does HOG + SVM work so well?

- Similar to SIFT, records spatial arrangement of histogram orientations
- Compare to learning only edges:
  - Complex junctions can be represented
  - Avoids problem of early thresholding
  - Represents also soft internal gradients
- Older methods based on edges have become largely obsolete





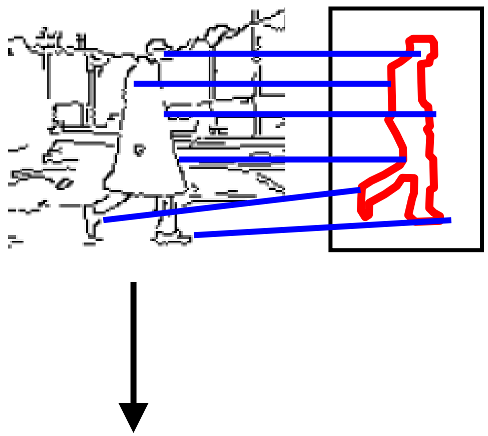
- HOG gives fixed length vector for window, suitable for feature vector for SVM

# Chamfer Matching

Input

Edges

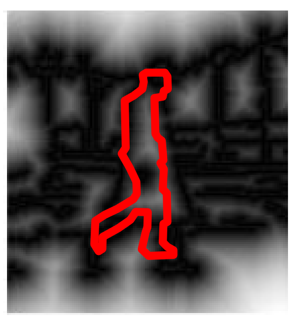
Template



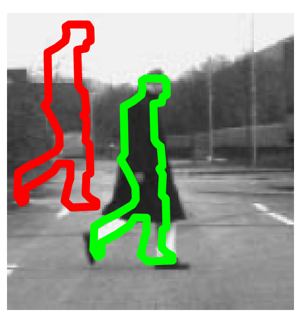
- Match points between template and image
- Measure mean distance
- Template edgel matches **nearest** image edgel

$$D(T, I) = \frac{1}{|T|} \sum_{\mathbf{p} \in T} \min_{\mathbf{q} \in I} d(\mathbf{p}, \mathbf{q})$$

Distance Transform



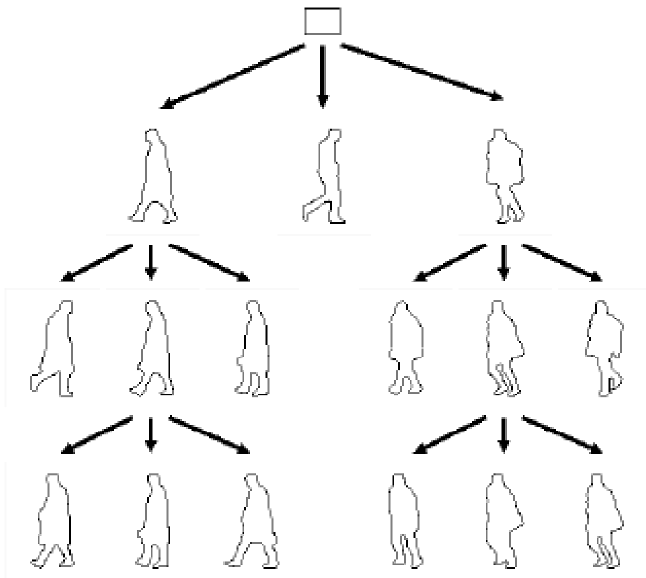
Best match



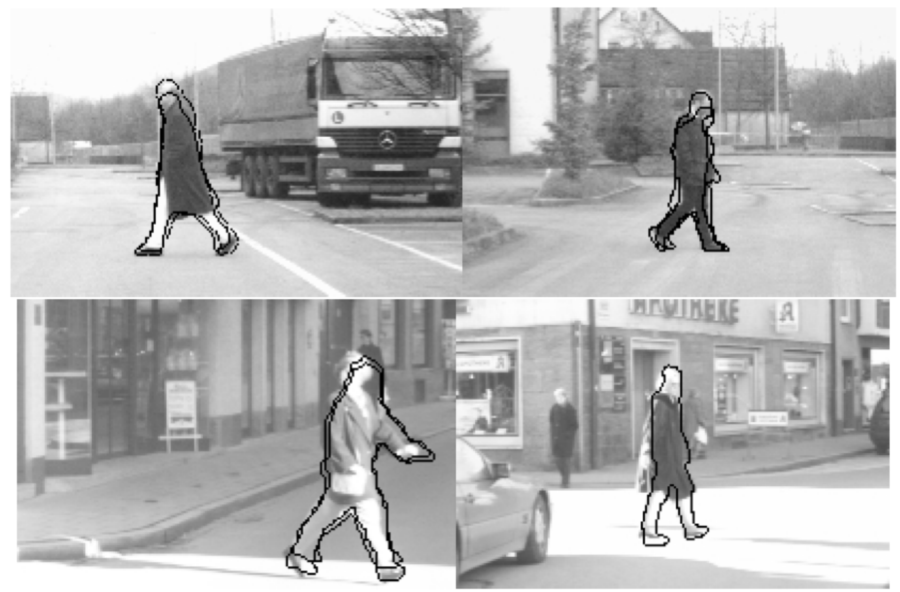
- Distance transform reduces min operation to array lookup
- Computable in linear time
- Localize by sliding window search

[Gavrila & Philomin, 1999]

# Chamfer Matching



Hierarchy of Templates



Detections

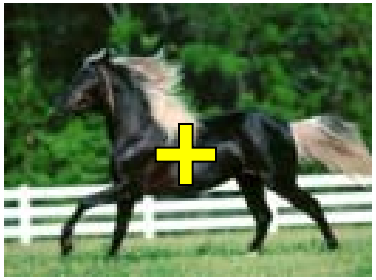
- In practice performs poorly in clutter
- Unoriented edges are not discriminative enough (too easy to find…)

[Gavrila & Philomin, 1999]

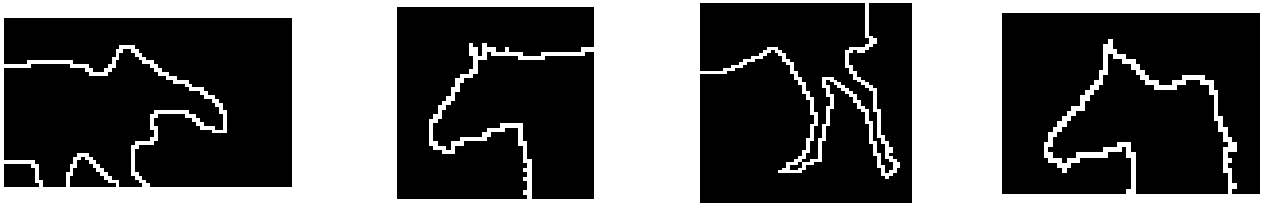
# Contour-fragment models

Shotton et al ICCV 05, Opelt et al ECCV 06

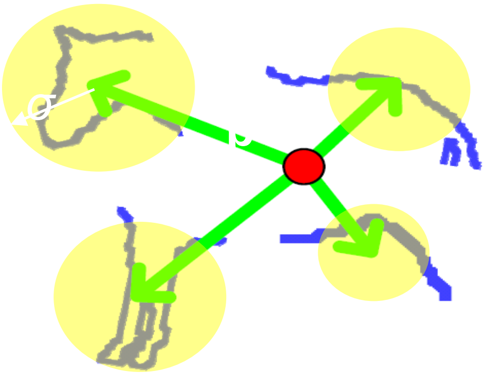
- Generalized Hough like representation using contour fragments

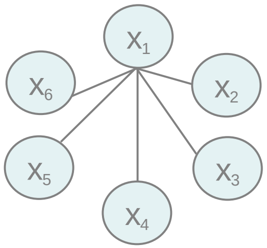


- Contour fragments learnt from edges of training images



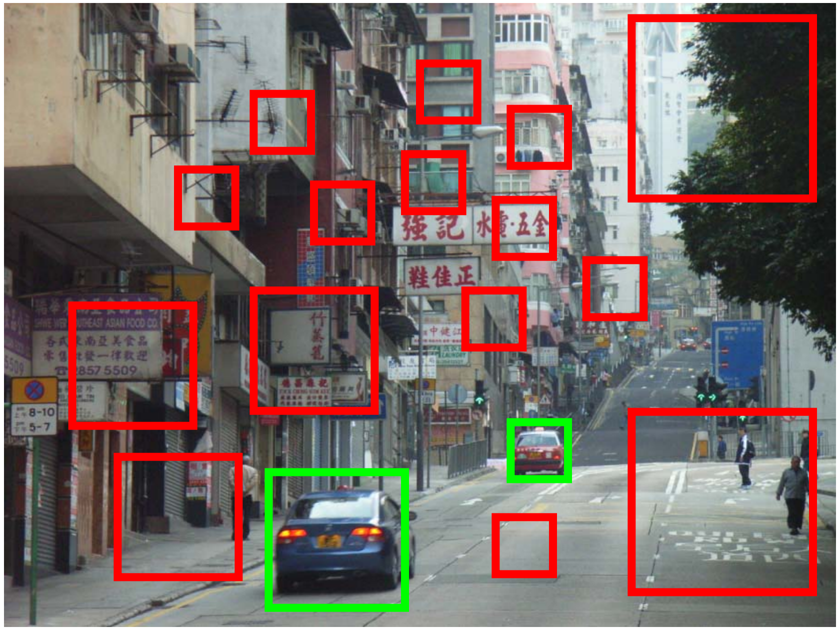
- Hough like voting for detection





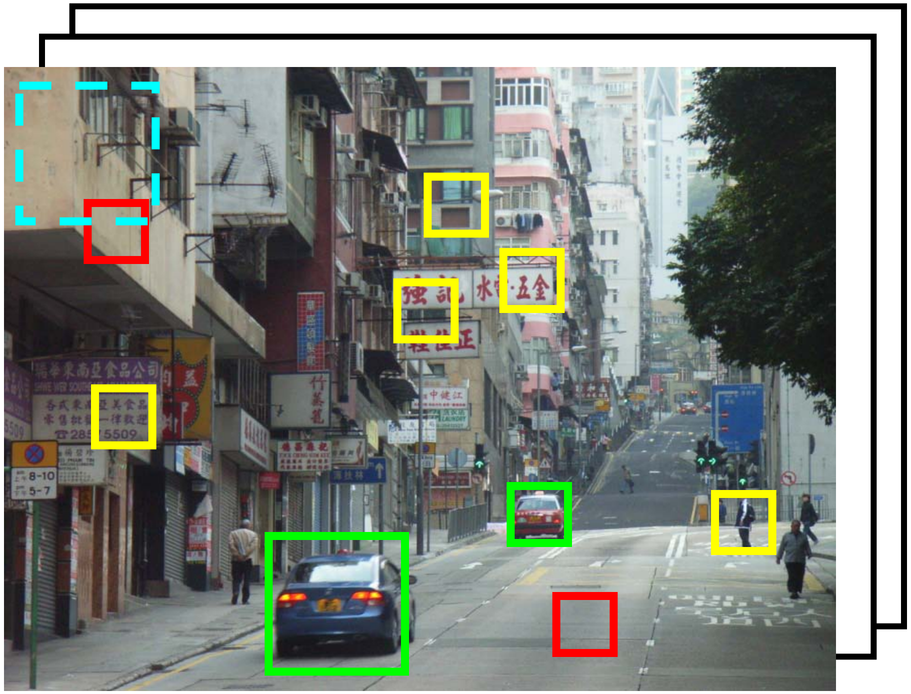
# Training a sliding window detector

- Object **detection** is inherently asymmetric: much more "non-object" than "object" data



- Classifier needs to have very low false positive rate
- Non-object category is very complex – need lots of data

# Bootstrapping



1. Pick negative training set at random
2. Train classifier
3. Run on training data
4. Add false positives to training set
5. Repeat from 2

- Collect a finite but diverse set of non-object windows
- Force classifier to concentrate on **hard negative** examples

- For some classifiers can ensure equivalence to training on entire data set

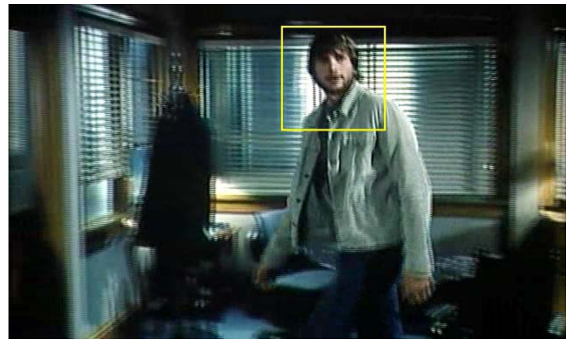
# Example: train an upper body detector

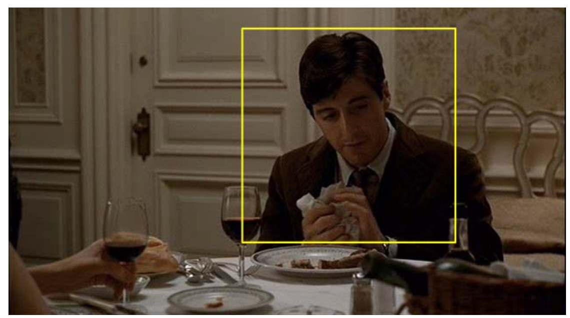
- Training data – used for training and validation sets
  - 33 Hollywood2 training movies
  - 1122 frames with upper bodies marked

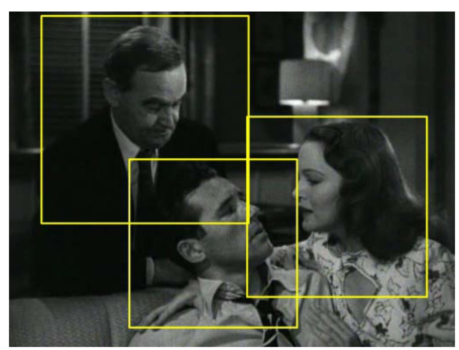
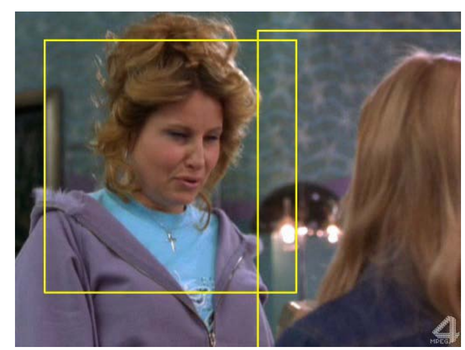
- First stage training (bootstrapping)
  - 1607 upper body annotations jittered to 32k positive samples
  - 55k negatives sampled from the same set of frames

- Second stage training (retraining)
  - 150k hard negatives found in the training data

# Training data – positive annotations

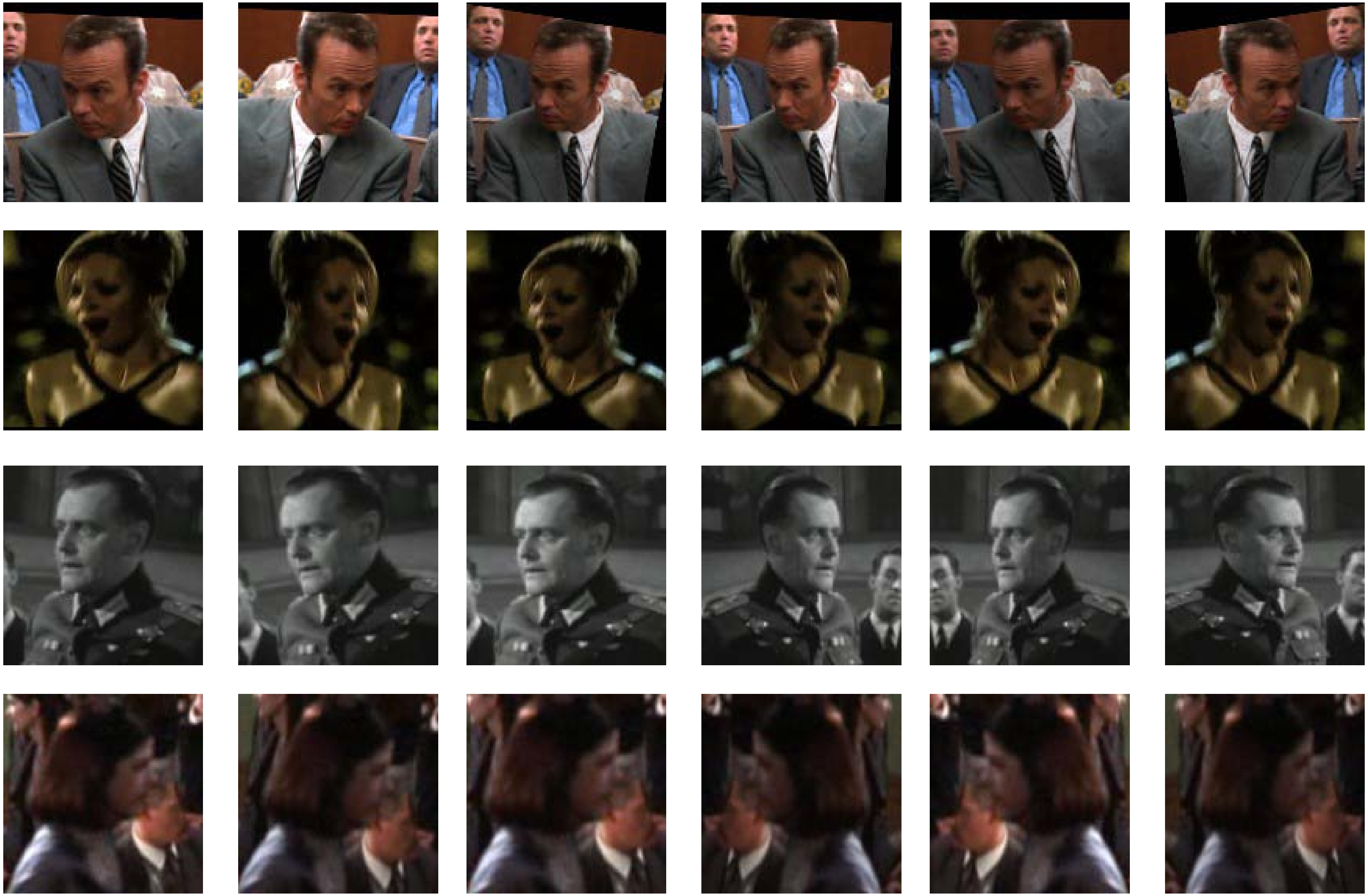



# Positive windows

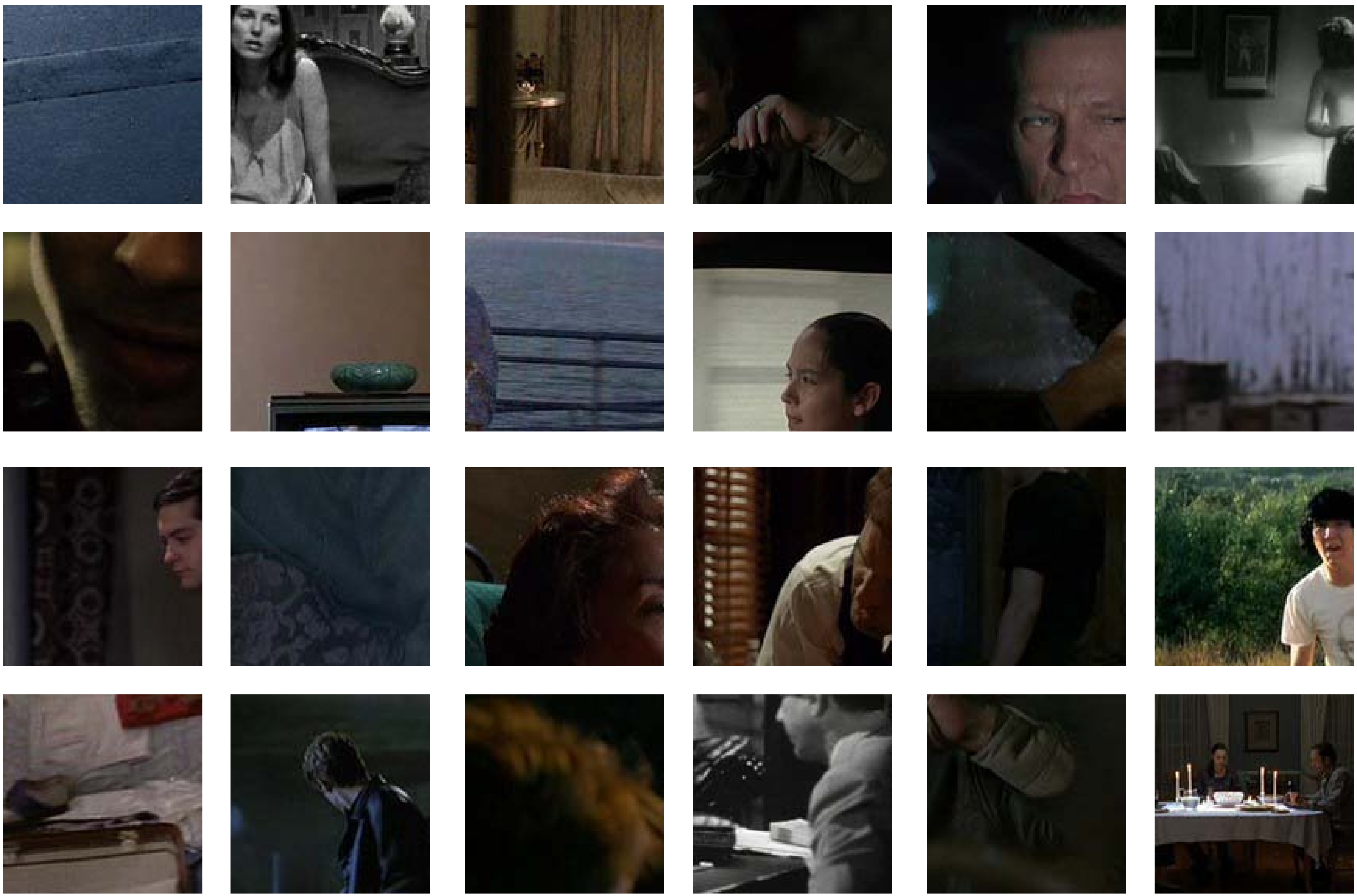

Note: common size and alignment

# Jittered positives



# Jittered positives

# Random negatives



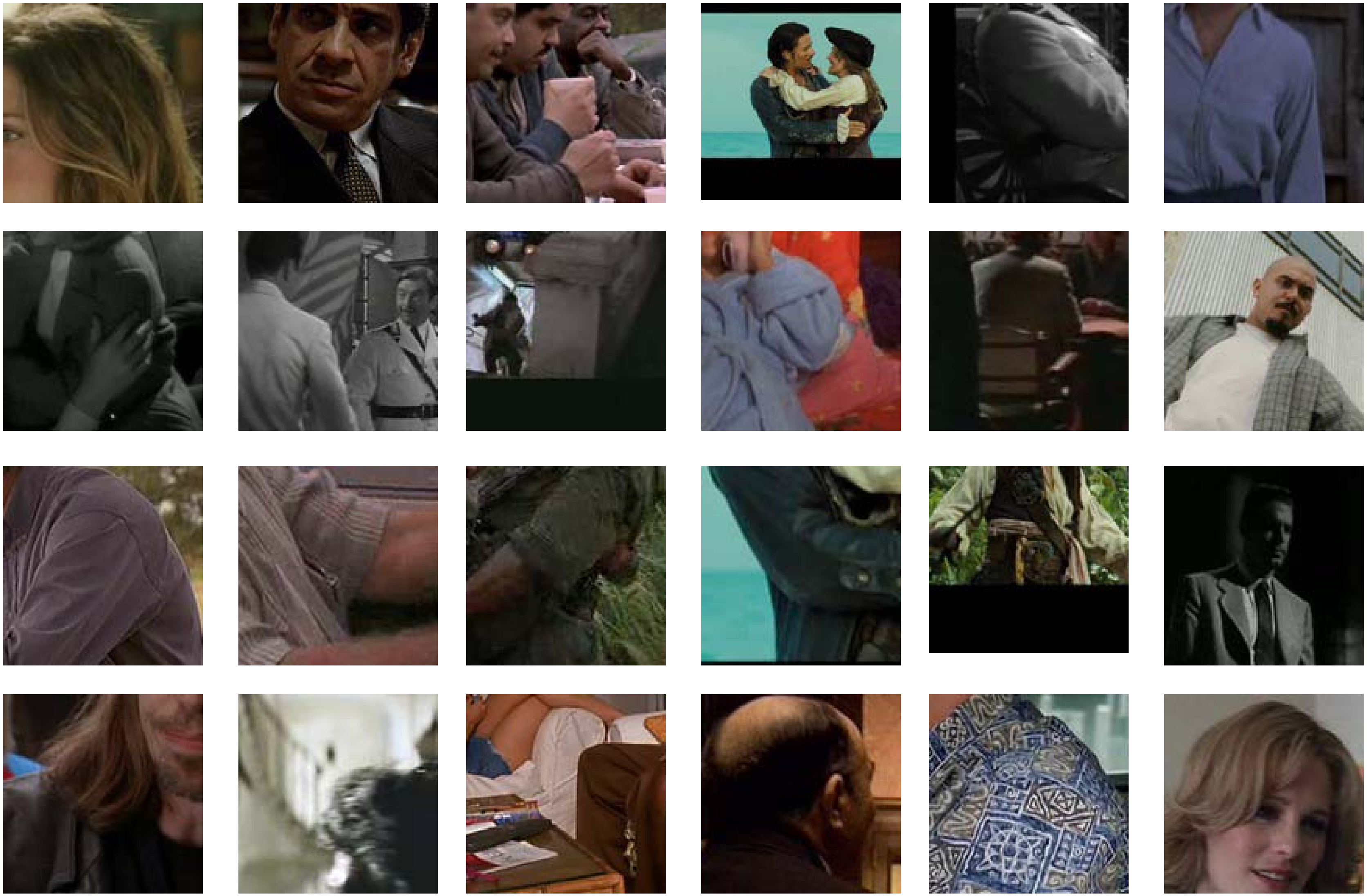
# Random negatives

# Window (Image) first stage classification

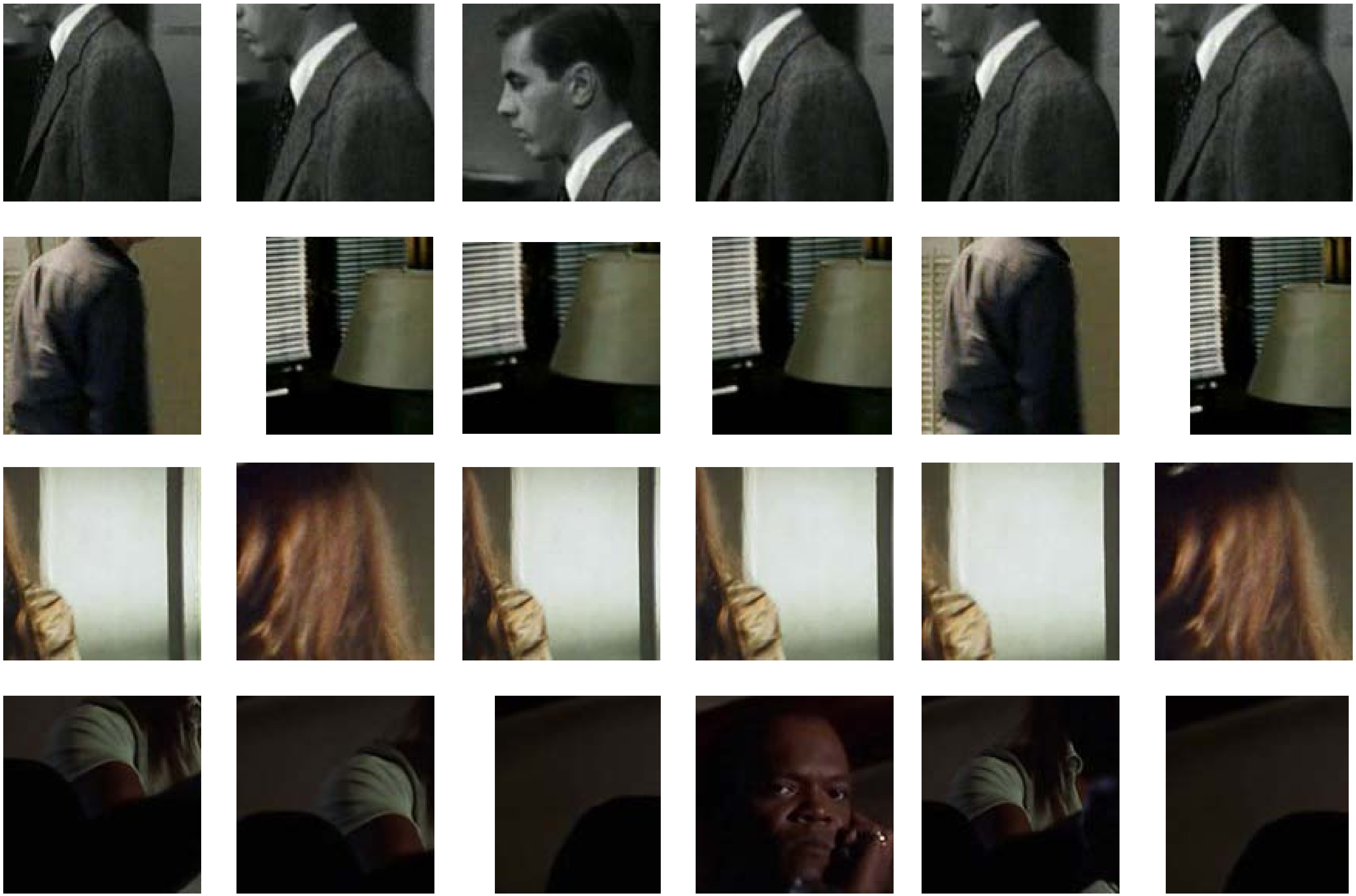
Jittered positives
random negatives → HOG Feature Extraction → $\mathbf{x}$ → Linear SVM Classifier $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$

- find high scoring false positives detections
- these are the hard negatives for the next round of training
- cost = # training images x inference on each image

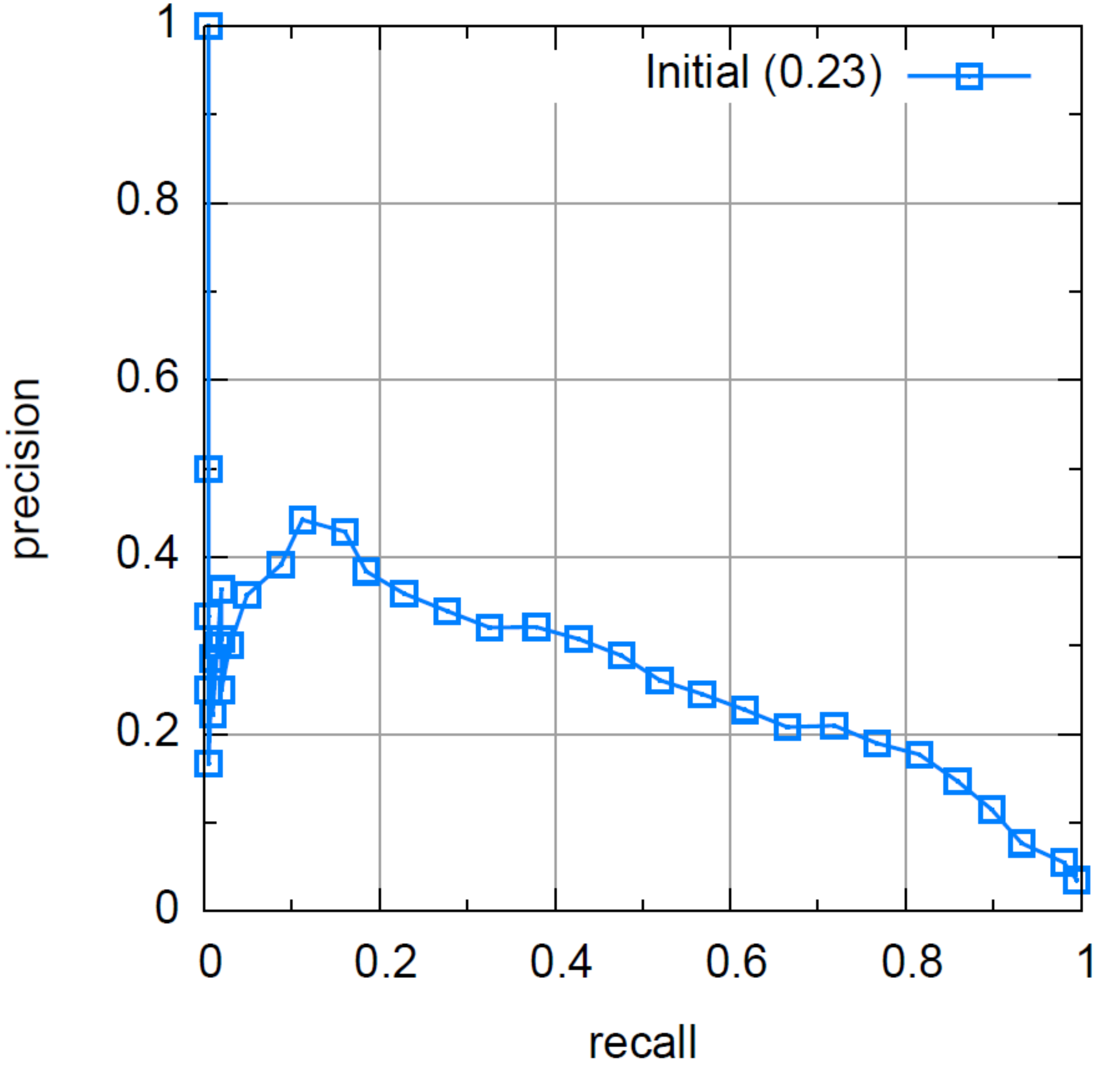
# Hard negatives



# Hard negatives

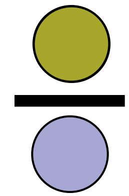


# First stage performance on validation set

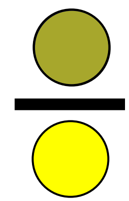


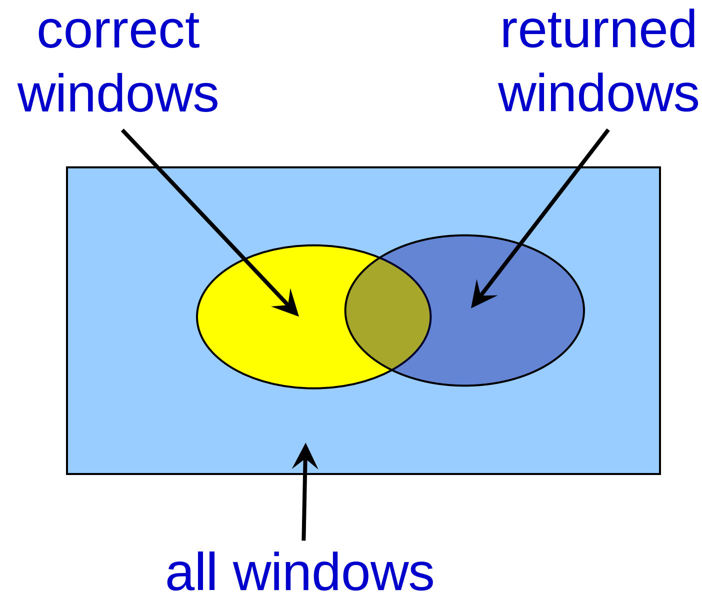
# Precision – Recall curve

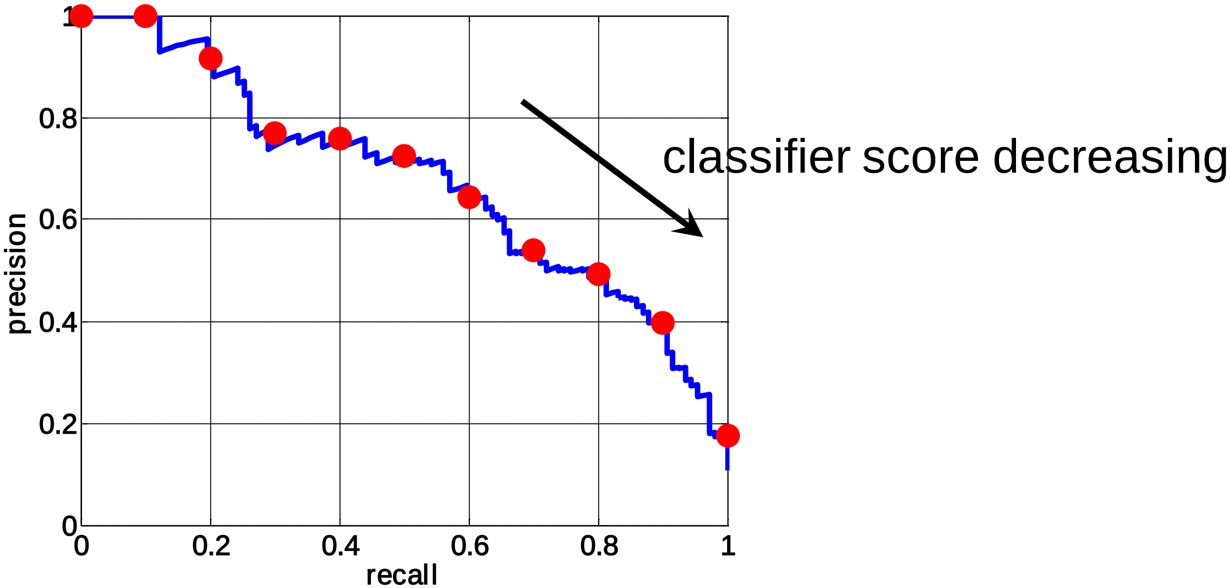
- Precision: % of returned windows that are correct



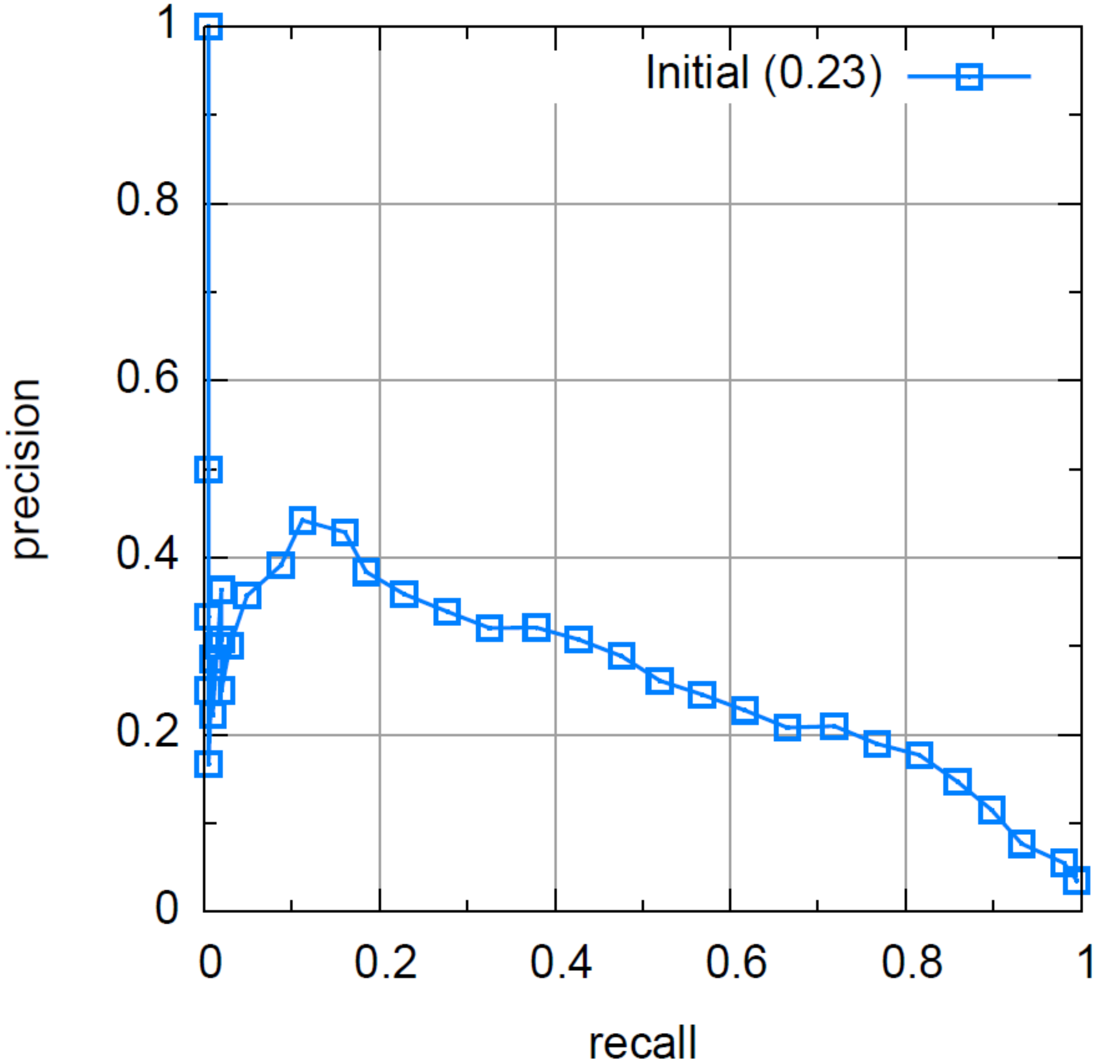
- Recall: % of correct windows that are returned



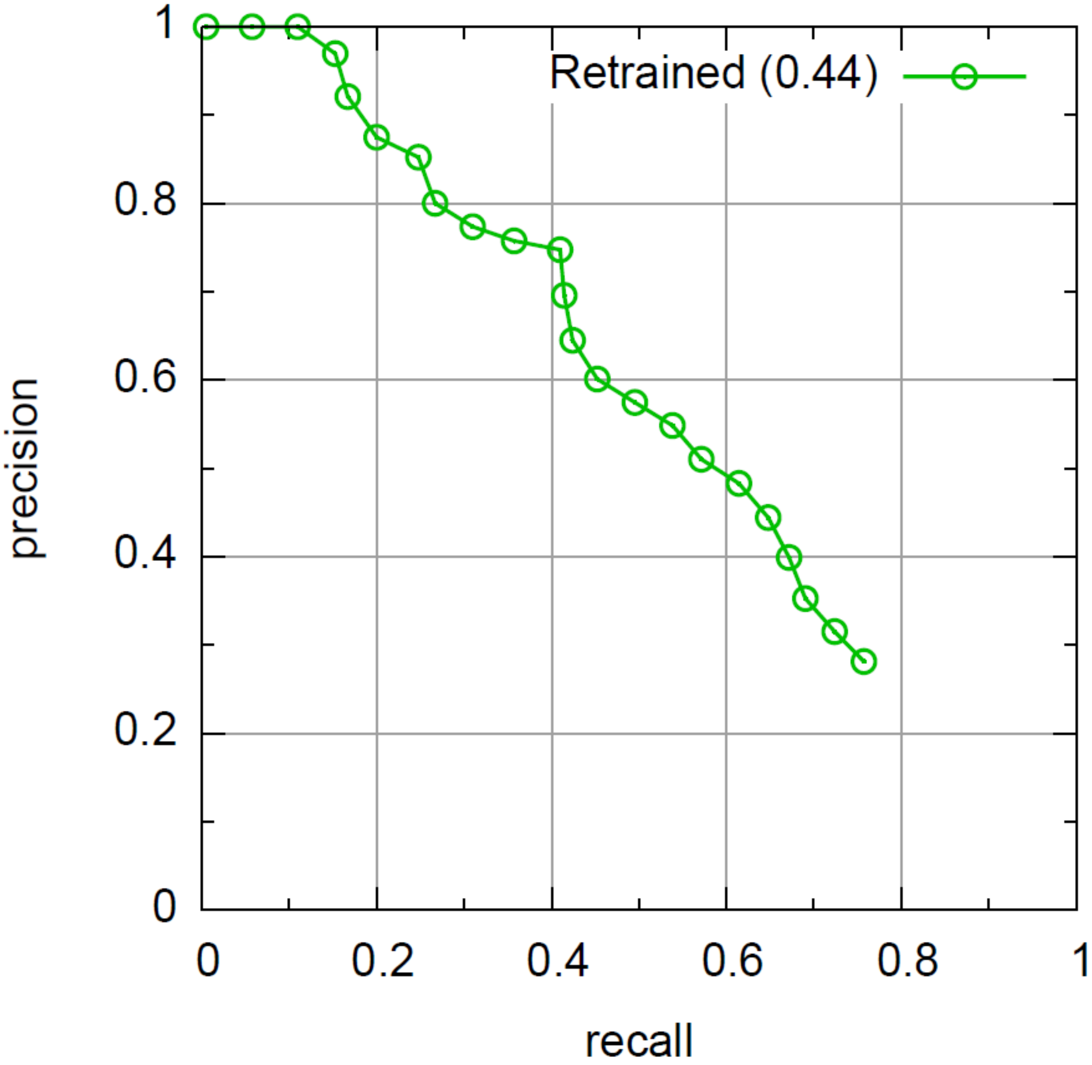


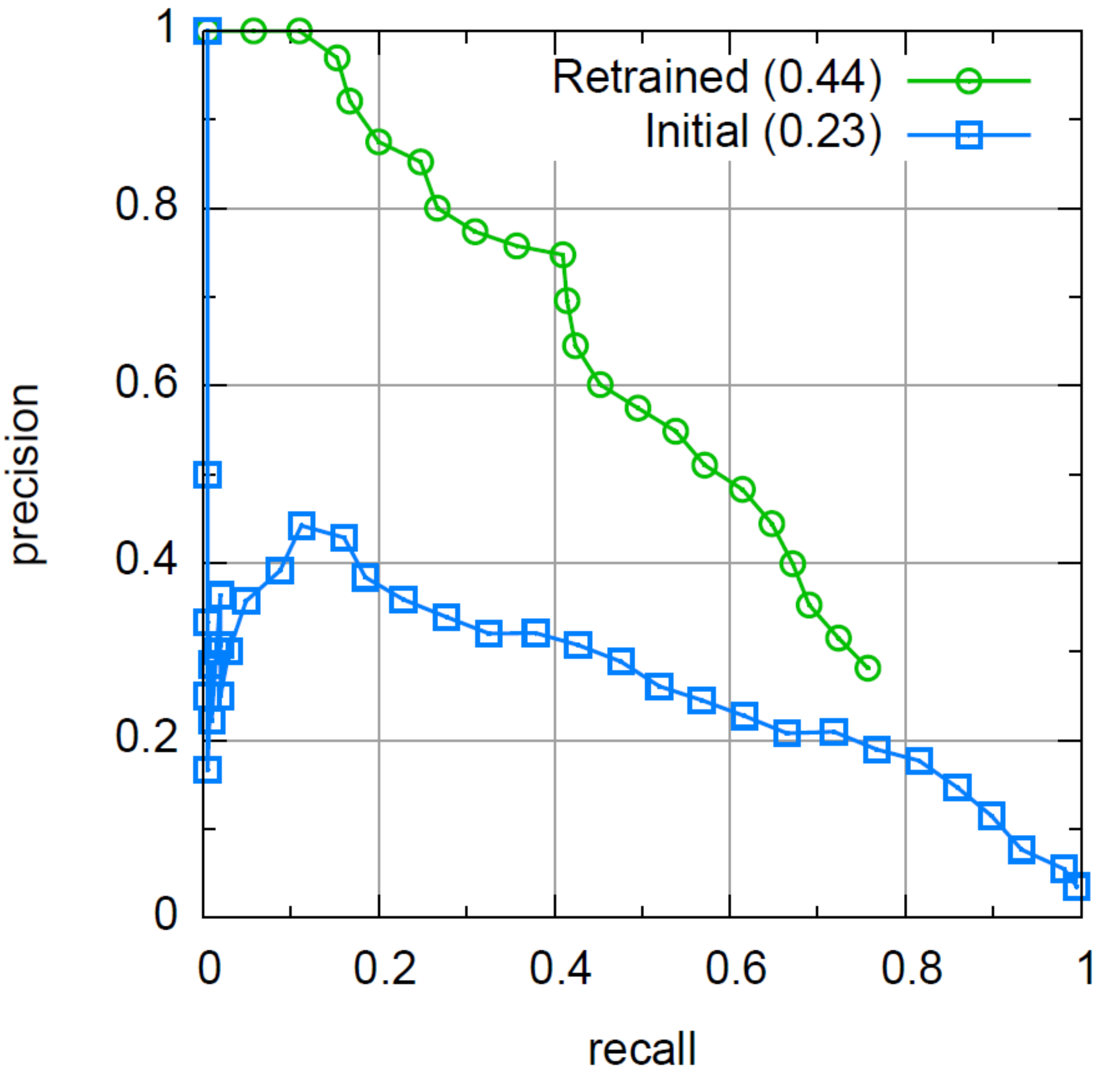
# First stage performance on validation set



# Performance after retraining



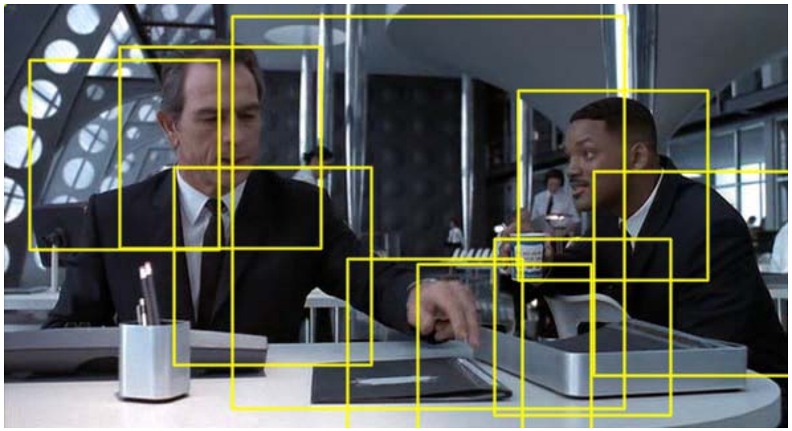
# Effects of retraining

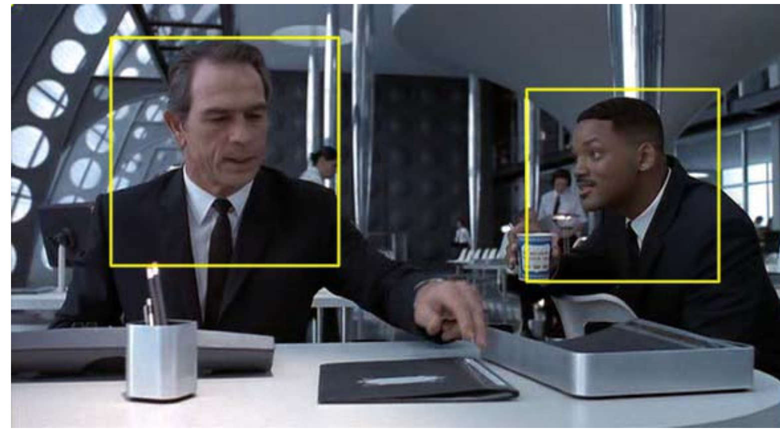


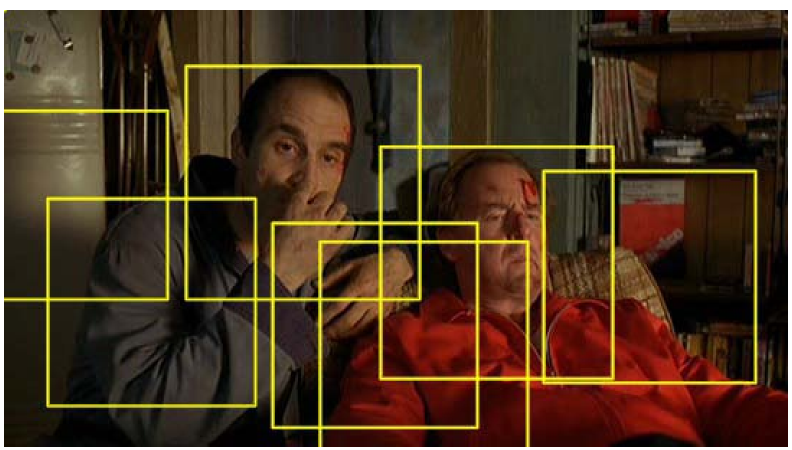
# Side by side

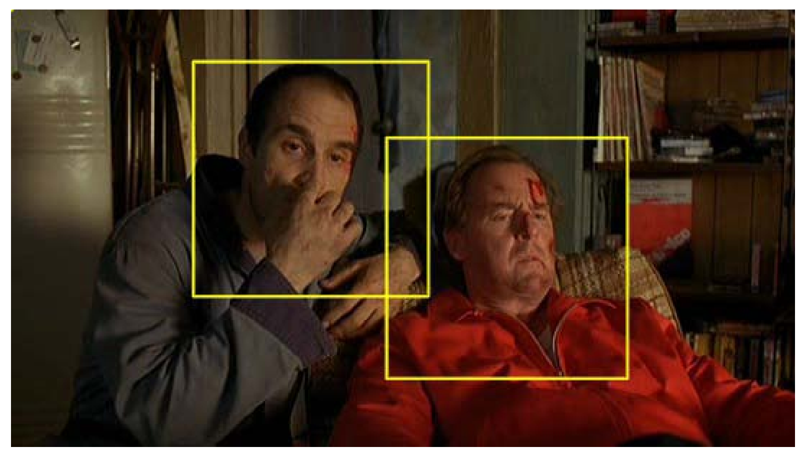
before retraining

after retraining





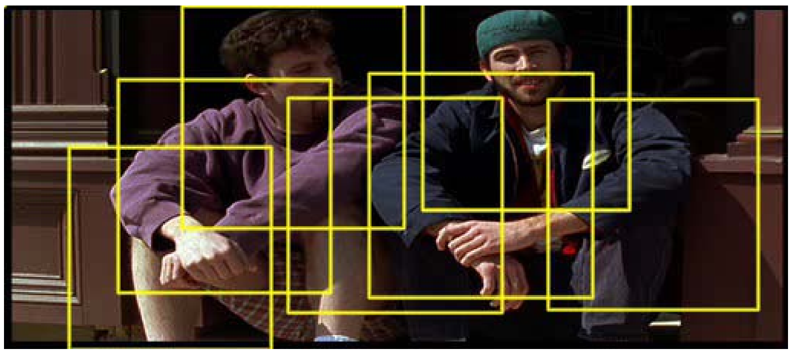


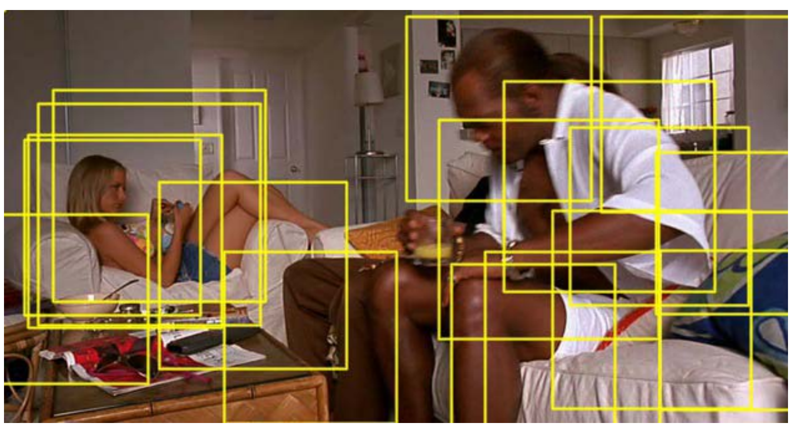


# Side by side

before retraining

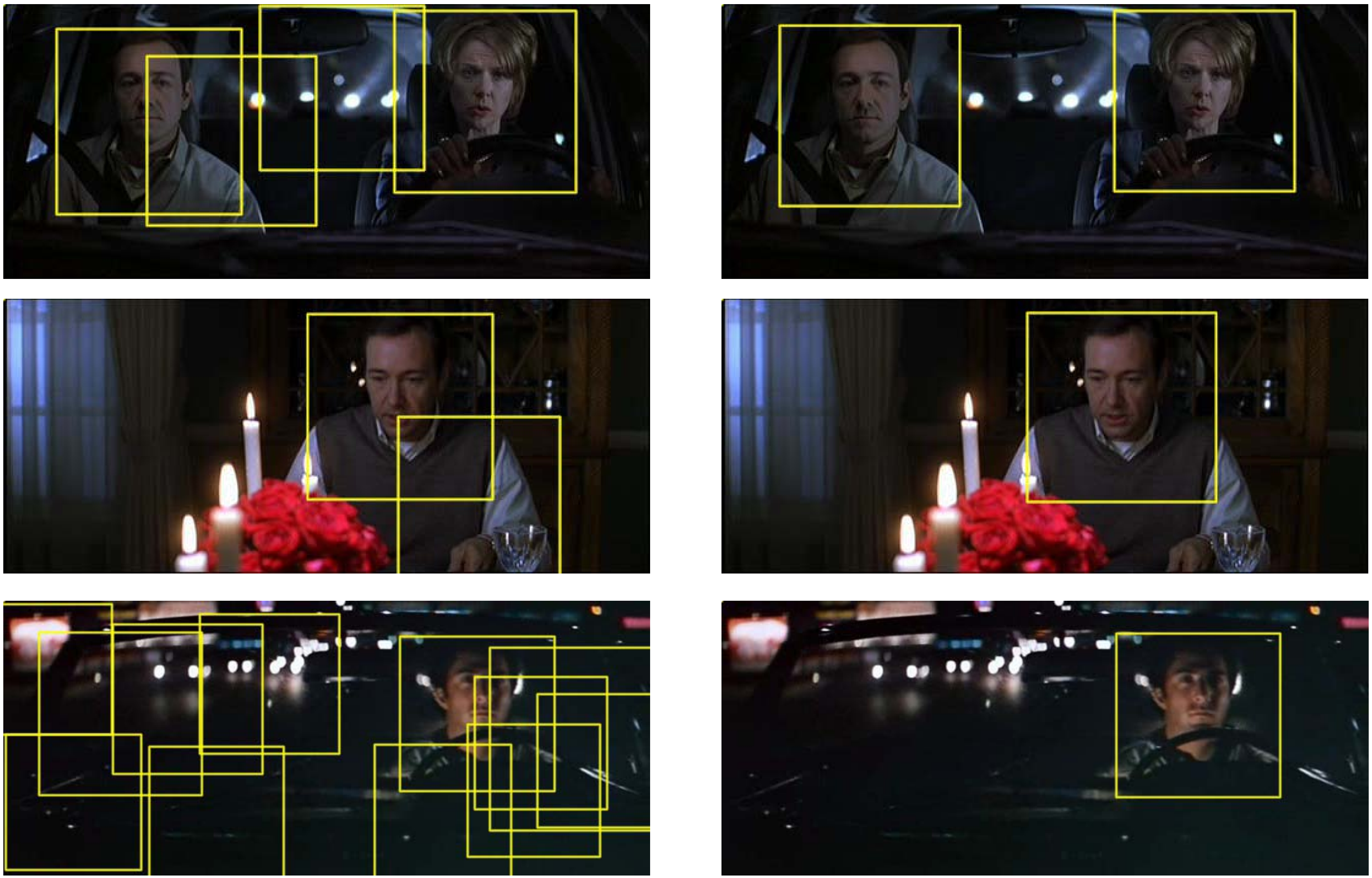
after retraining



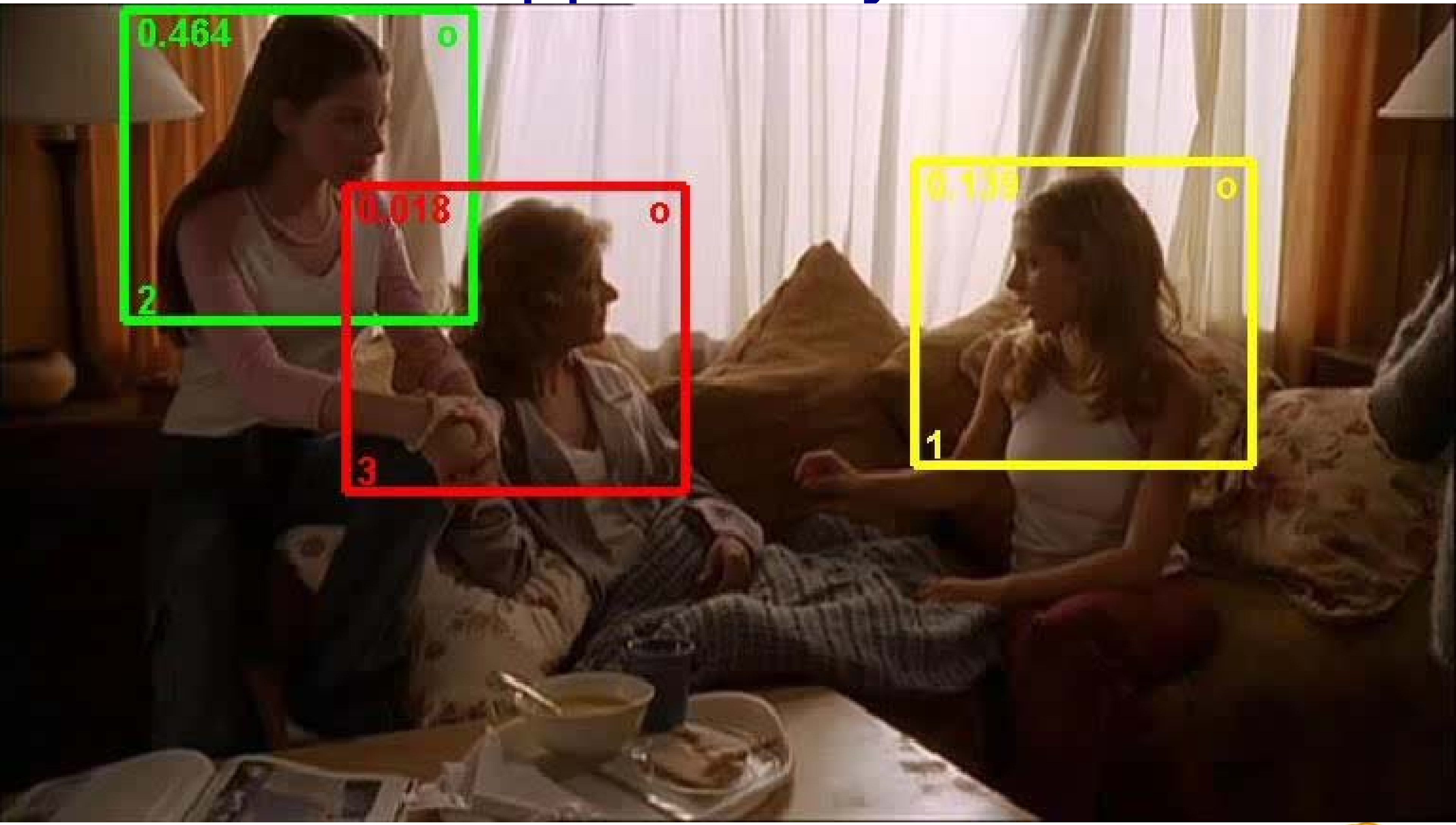


# Side by side

before retraining | after retraining

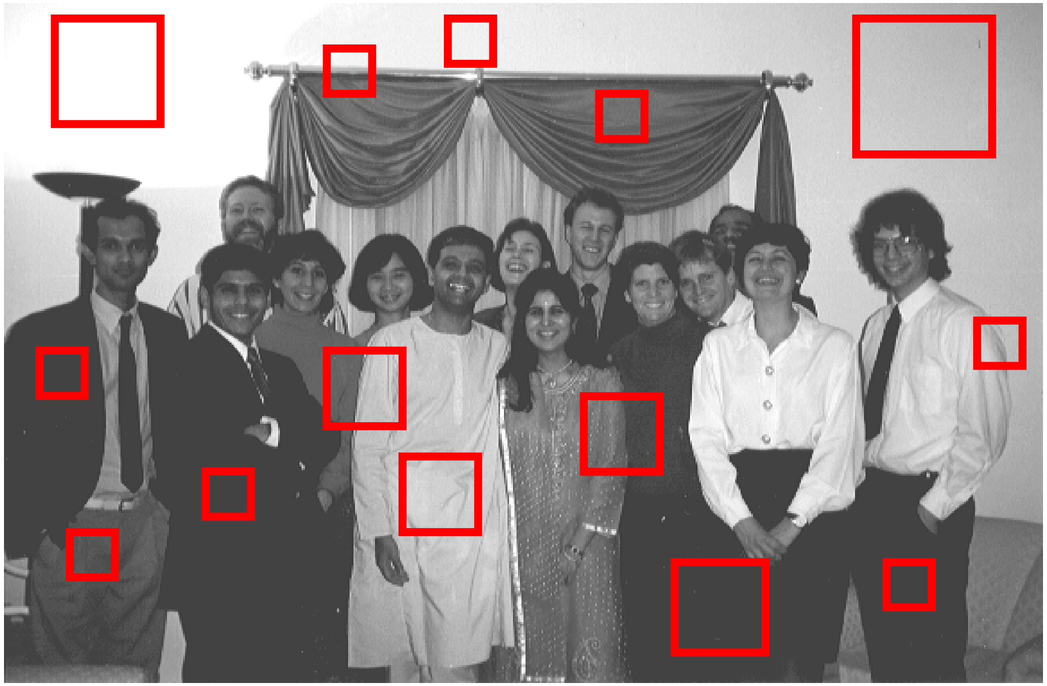


# Tracked upper body detections



# Accelerating Sliding Window Search

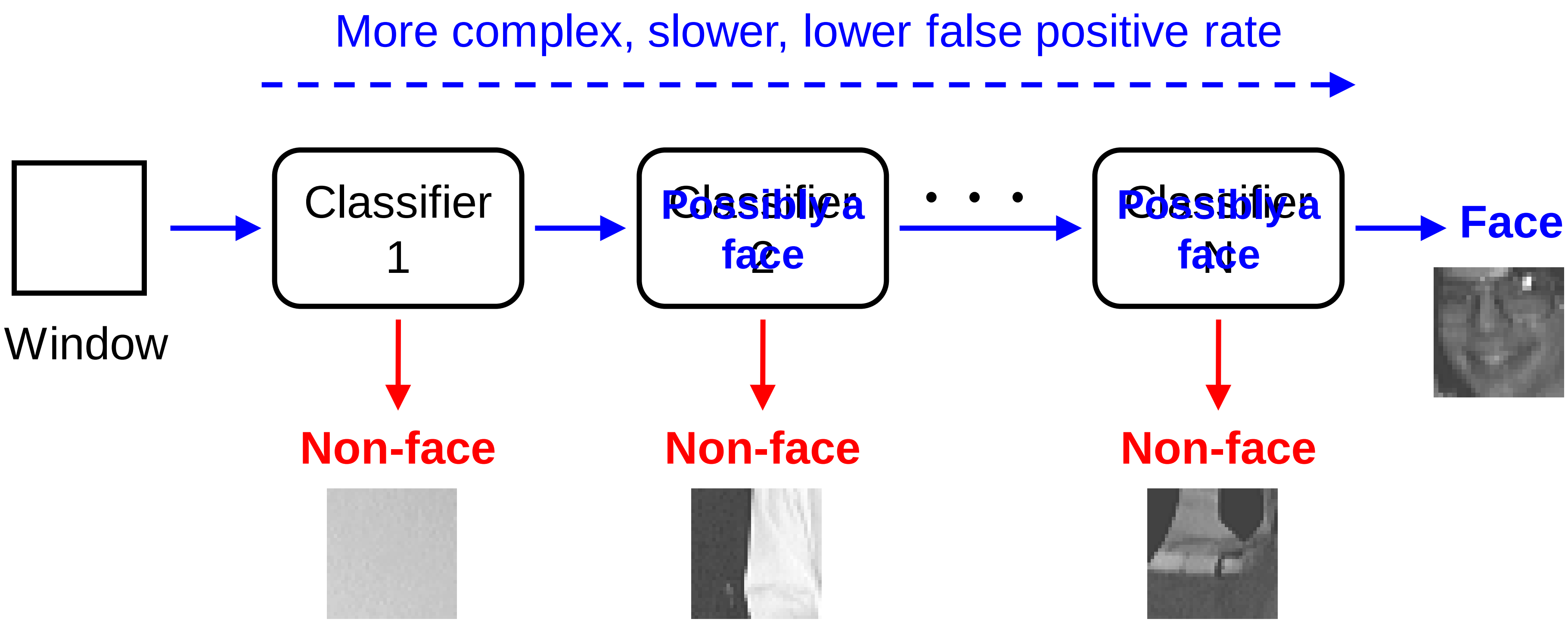
- Sliding window search is slow because so many windows are needed e.g. x × y × scale ≈ 100,000 for a 320×240 image



- Most windows are clearly not the object class of interest
- Can we speed up the search?

# Cascaded Classification

- Build a sequence of classifiers with increasing complexity




- Reject easy non-objects using simpler and faster classifiers

# Cascaded Classification

- Slow expensive classifiers only applied to a few windows ➔ significant speed-up

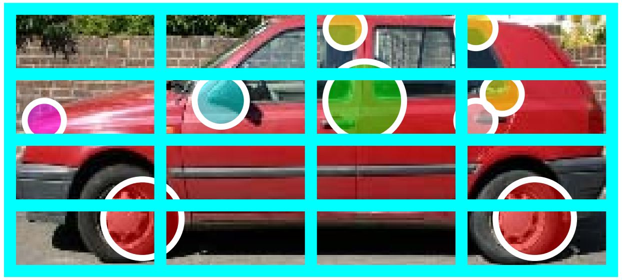
- Controlling classifier complexity/speed:
  - Number of support vectors [Romdhani et al, 2001]
  - Number of features [Viola & Jones, 2001]
  - Type of SVM kernel [Vedaldi et al, 2009]

# Summary: Sliding Window Detection

- Can convert any image classifier into an object detector by sliding window. Efficient search methods available.

- Requirements for invariance are reduced by searching over e.g. translation and scale

- Spatial correspondence can be "engineered in" by spatial tiling



# Outline

1. Sliding window detectors
2. Features and adding spatial information
3. HOG + linear SVM classifier
4. Two state of the art algorithms and PASCAL VOC
    - VOC challenge
    - Vedaldi et al – multiple kernels and features, cascade
    - Felzenswalb et al – multiple parts, latent SVM
5. The future and challenges

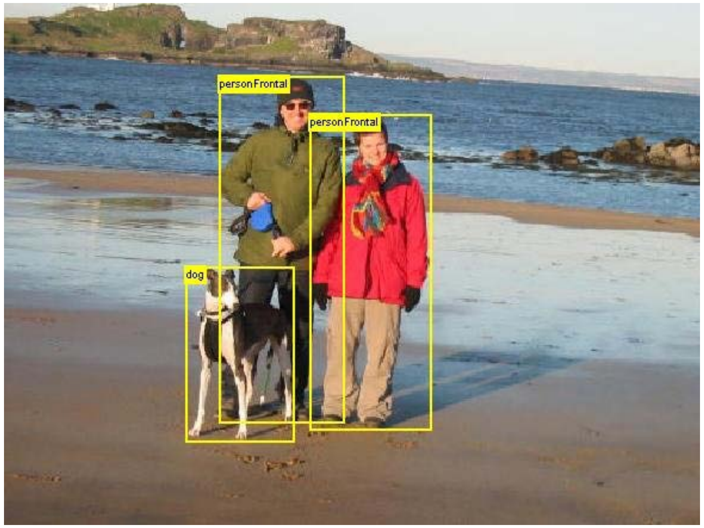
# The PASCAL Visual Object Classes (VOC) Dataset and Challenge

Mark Everingham
Luc Van Gool
Chris Williams
John Winn
Andrew Zisserman

# The PASCAL VOC Challenge

- Challenge in visual object recognition funded by PASCAL network of excellence

- Publicly available dataset of annotated images




- Main competitions in classification (is there an X in this image), detection (where are the X's), and segmentation (which pixels belong to X)

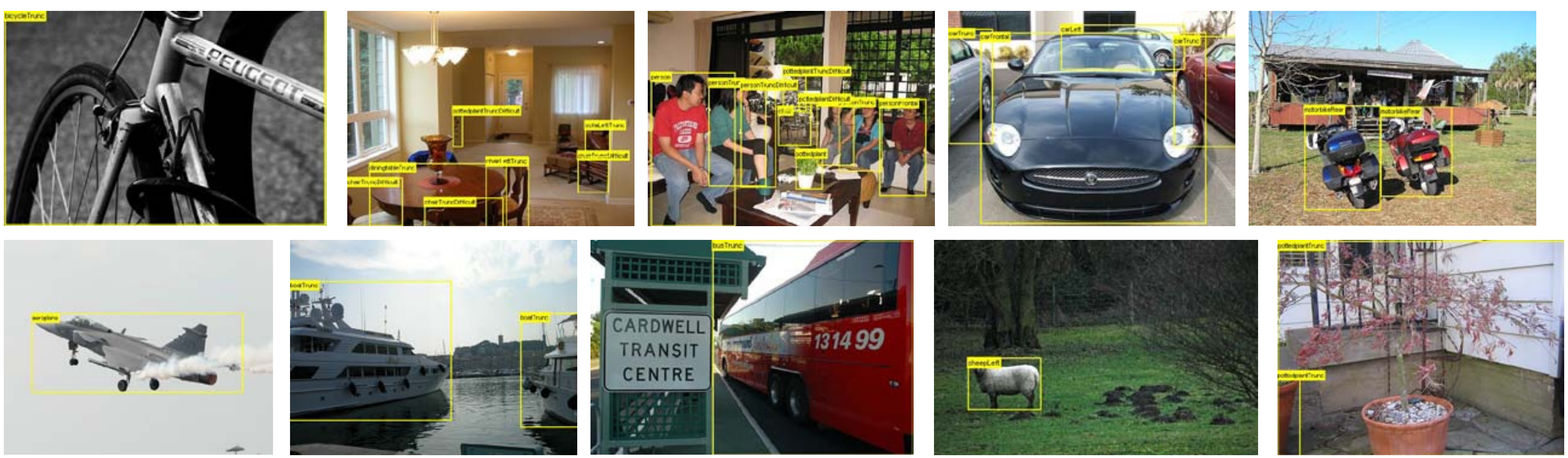
- "Taster competitions" in 2-D human "pose estimation" (2007-present) and static action classes

- Standard evaluation protocol (software supplied)

# Dataset Content

- 20 classes: aeroplane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, TV

- Real images downloaded from flickr, not filtered for “quality”




- Complex scenes, scale, pose, lighting, occlusion, ...

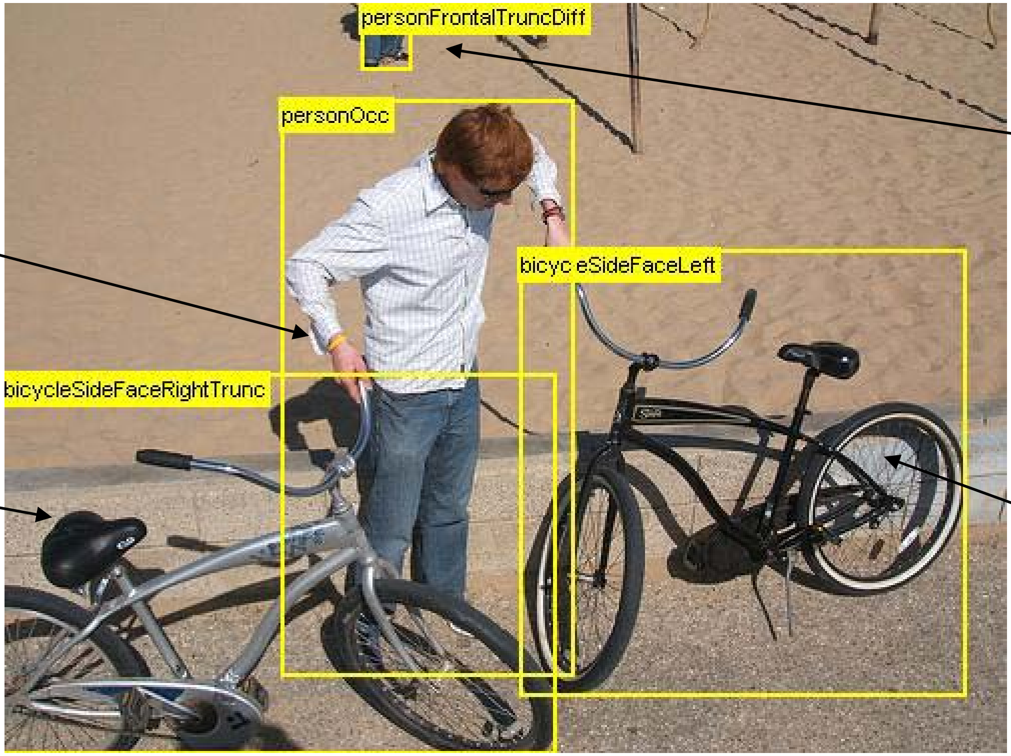
# Annotation

- Complete annotation of all objects

- Annotated in one session with written guidelines

**Occluded**
Object is significantly occluded within BB

**Truncated**
Object extends beyond BB




**Difficult**
Not scored in evaluation

**Pose**
Facing left

# Examples

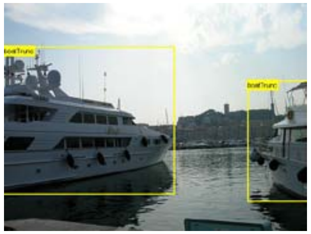
Aeroplane

Bicycle

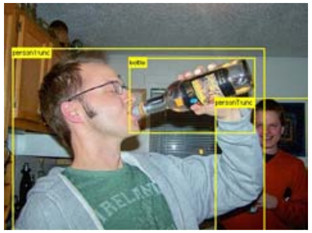


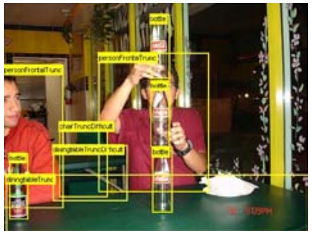
Bird

Boat



Bottle



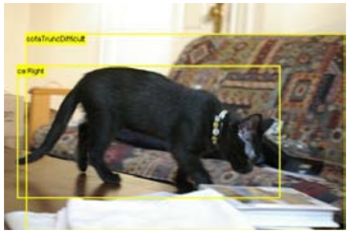


Bus

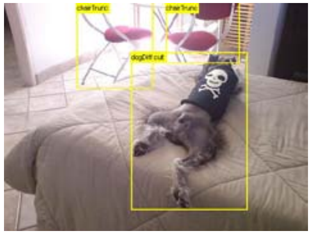


Car

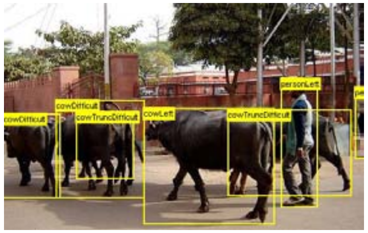
Cat



Chair



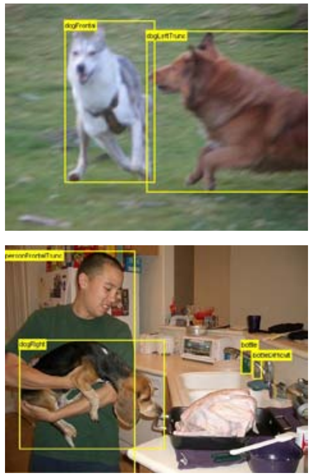
Cow



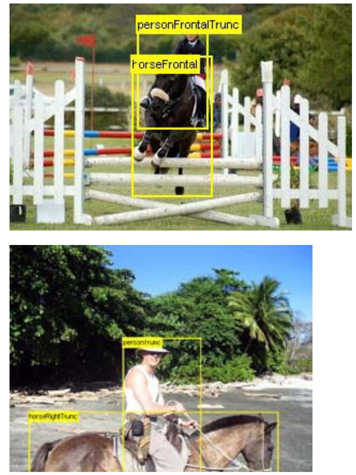
# Examples

Dining Table

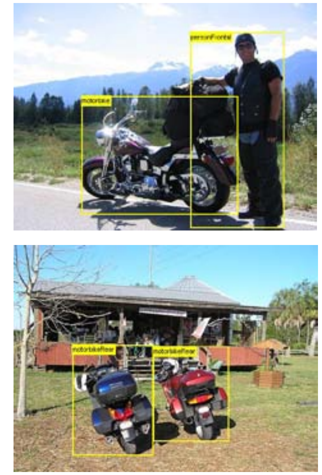
Dog



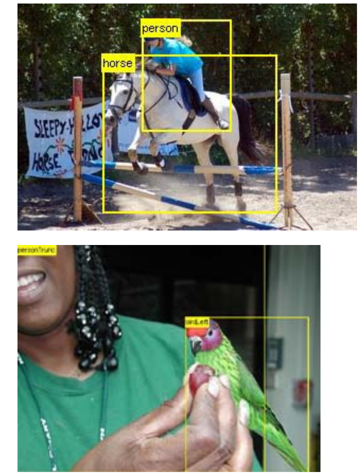
Horse




Motorbike

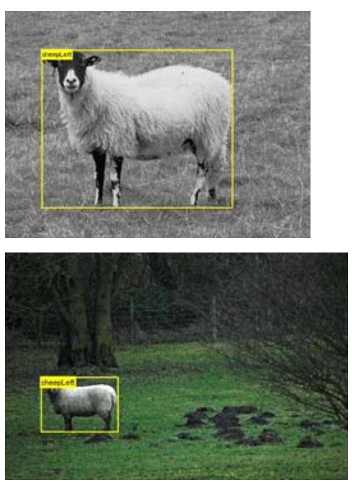


Person

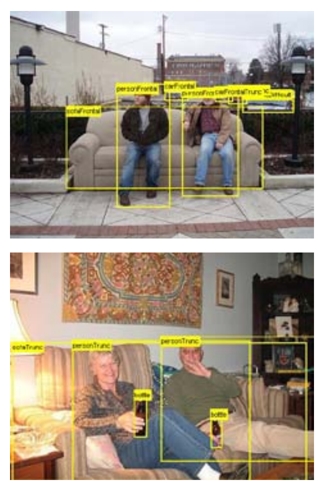



Potted Plant

Sheep



Sofa



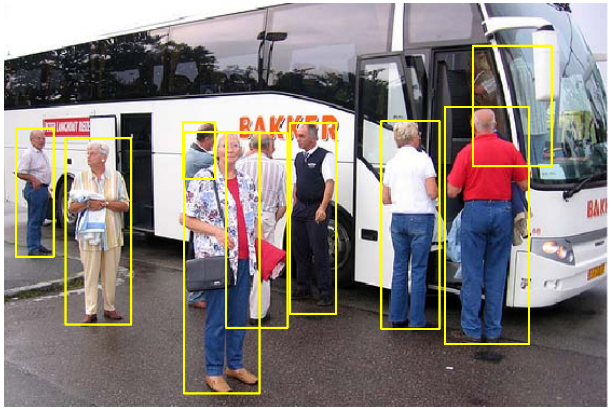
Train

TV/Monitor

# Main Challenge Tasks

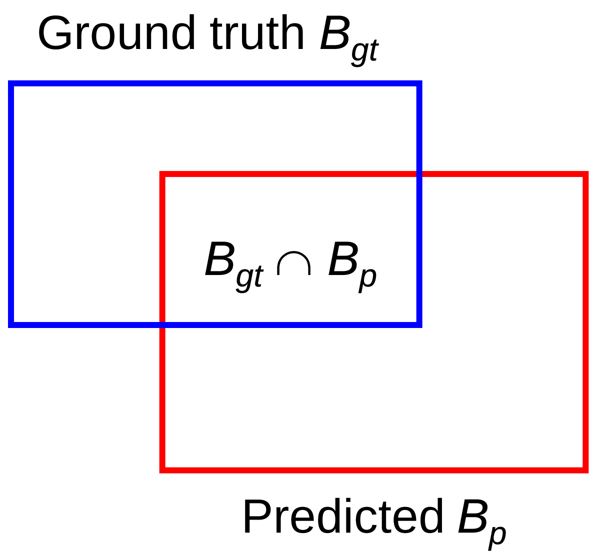
- **Classification**
  - Is there a dog in this image?
  - Evaluation by precision/recall

- **Detection**
  - Localize all the people (if any) in this image
  - Evaluation by precision/recall based on bounding box overlap



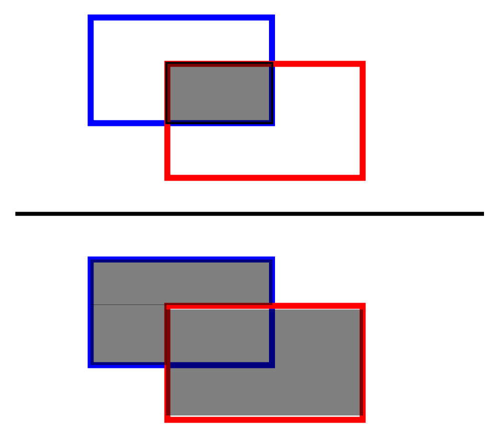
# Detection: Evaluation of Bounding Boxes

- Area of Overlap (AO) Measure




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

Detection if



> Threshold

50%

# Dataset Statistics

| | train | | val | | trainval | | test | |
|---|---|---|---|---|---|---|---|---|
| | Images | Objects | Images | Objects | Images | Objects | Images | Objects |
| Aeroplane | 201 | 267 | 206 | 266 | 407 | 533 | | |
| Bicycle | 167 | 232 | 181 | 236 | 348 | 468 | | |
| Bird | 262 | 381 | 243 | 379 | 505 | 760 | | |
| Boat | 170 | 270 | 155 | 267 | 325 | 537 | | |
| Bottle | 220 | 394 | 200 | 393 | 420 | 787 | | |
| Bus | 132 | 179 | 126 | 186 | 258 | 365 | | |
| Car | 372 | 664 | 358 | 653 | 730 | 1,317 | | |
| Cat | 266 | 308 | 277 | 314 | 543 | 622 | | |
| Chair | 338 | 716 | 330 | 713 | 668 | 1,429 | | |
| Cow | 86 | 164 | 86 | 172 | 172 | 336 | | |
| Diningtable | 140 | 153 | 131 | 153 | 271 | 306 | | |
| Dog | 316 | 391 | 333 | 392 | 649 | 783 | | |
| Horse | 161 | 237 | 167 | 245 | 328 | 482 | | |
| Motorbike | 171 | 235 | 167 | 234 | 338 | 469 | | |
| Person | 1,333 | 2,819 | 1,446 | 2,996 | 2,779 | 5,815 | | |
| Pottedplant | 166 | 311 | 166 | 316 | 332 | 627 | | |
| Sheep | 67 | 163 | 64 | 175 | 131 | 338 | | |
| Sofa | 155 | 172 | 153 | 175 | 308 | 347 | | |
| Train | 164 | 190 | 160 | 191 | 324 | 381 | | |
| Tvmonitor | 180 | 259 | 173 | 257 | 353 | 516 | | |
| Total | 3,473 | 8,505 | 3,581 | 8,713 | 7,054 | 17,218 | 6,650 | 16,829 |

# True Positives - Bicycle

UoCTTI_LSVM-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT

# False Positives - Bicycle

UoCTTI_LSVM-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT

# True Positives – TV/monitor

OXFORD_MKL

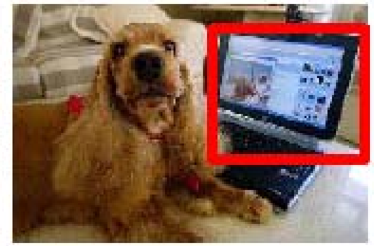
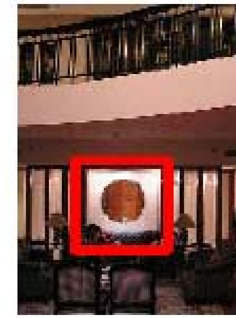
UoCTTI_LSVM-MDPM


LEAR_CHI-SVM-SIFT-HOG-CLS

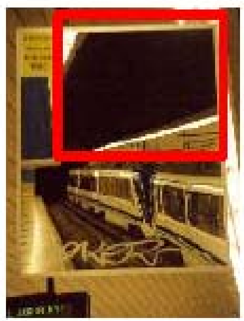
# False Positives – TV/monitor

OXFORD_MKL

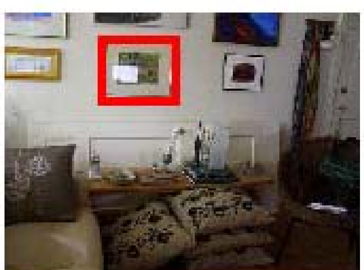


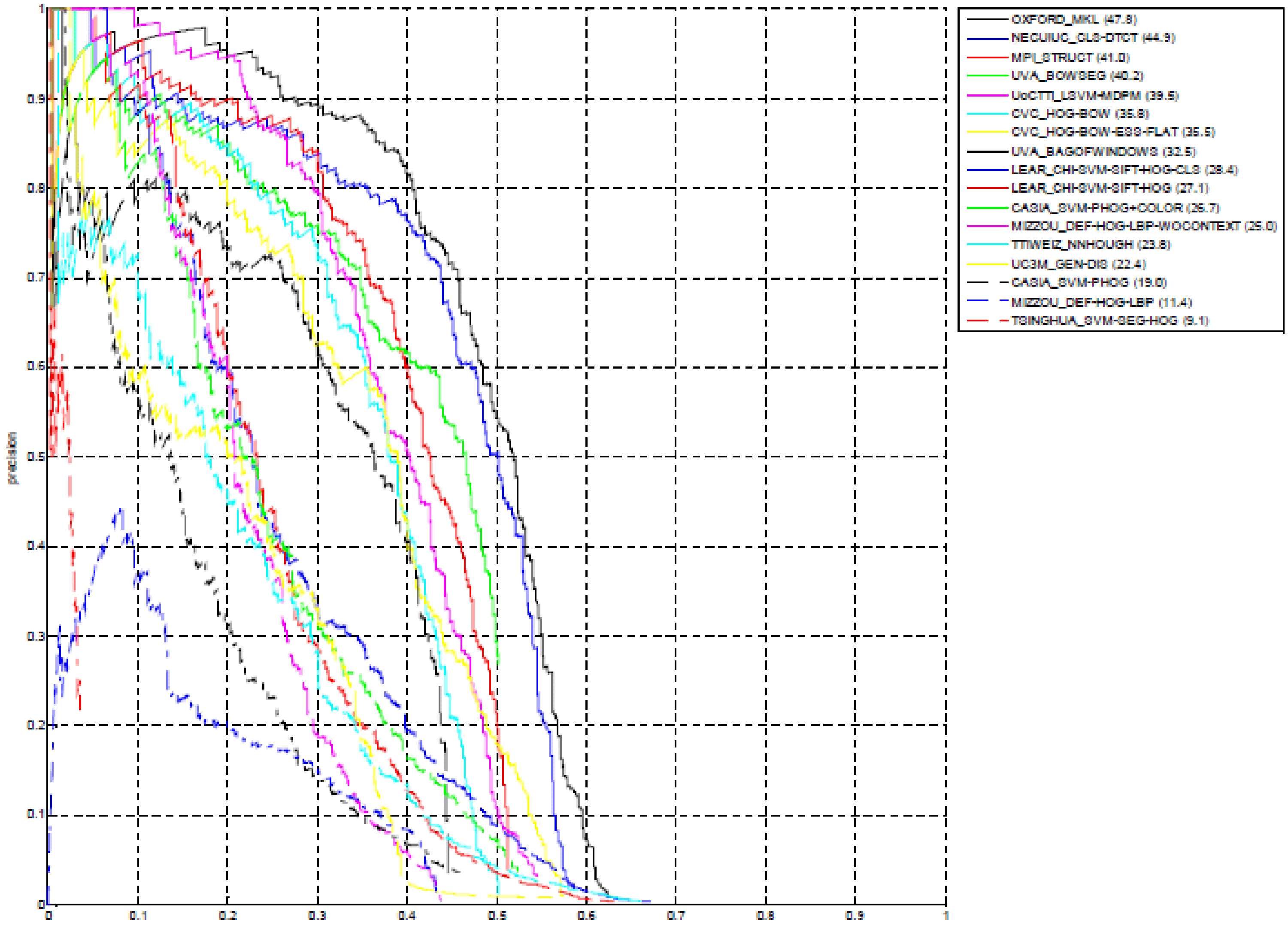
UoCTTI_LSVM-MDPM

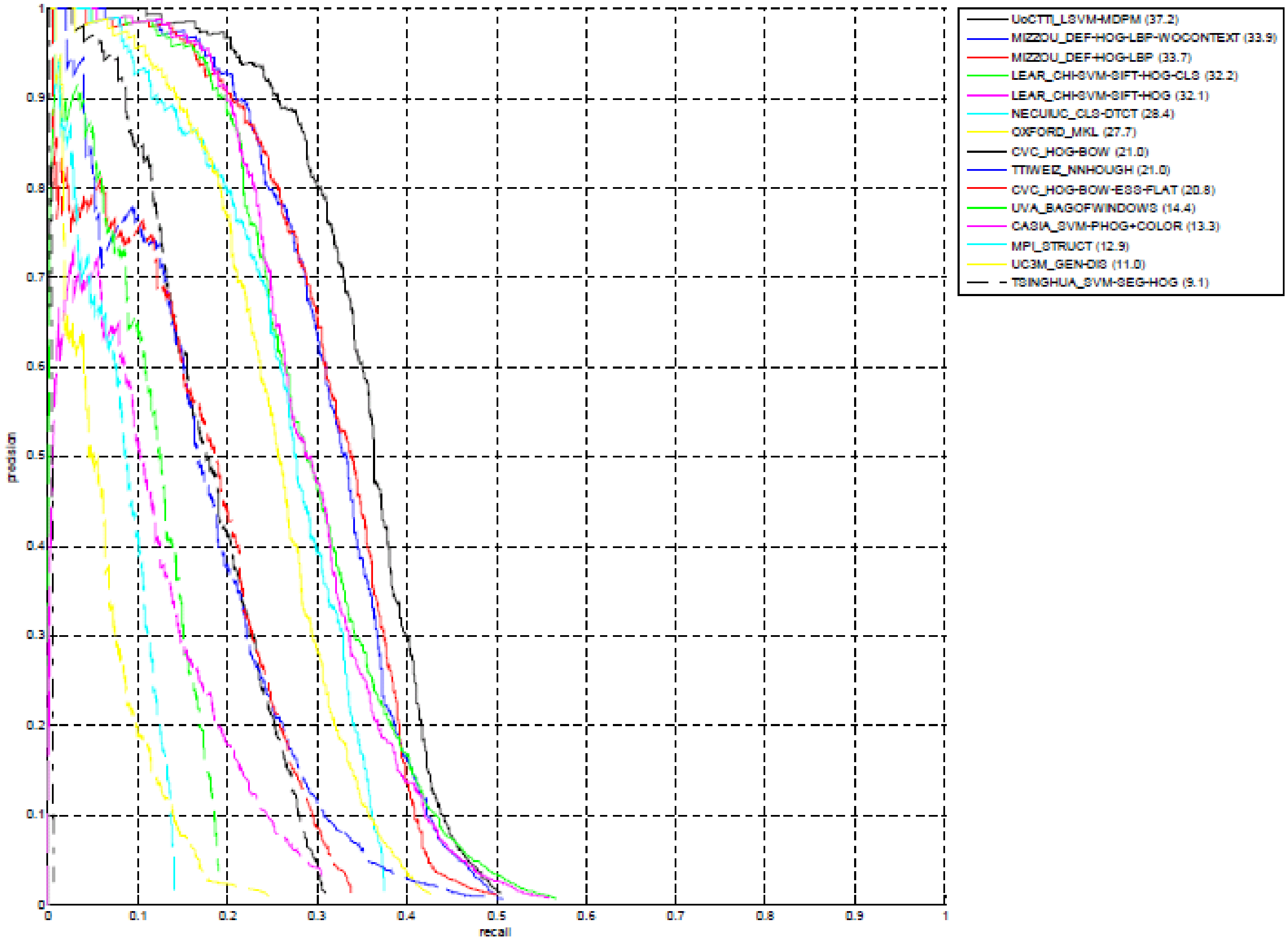

LEAR_CHI-SVM-SIFT-HOG-CLS

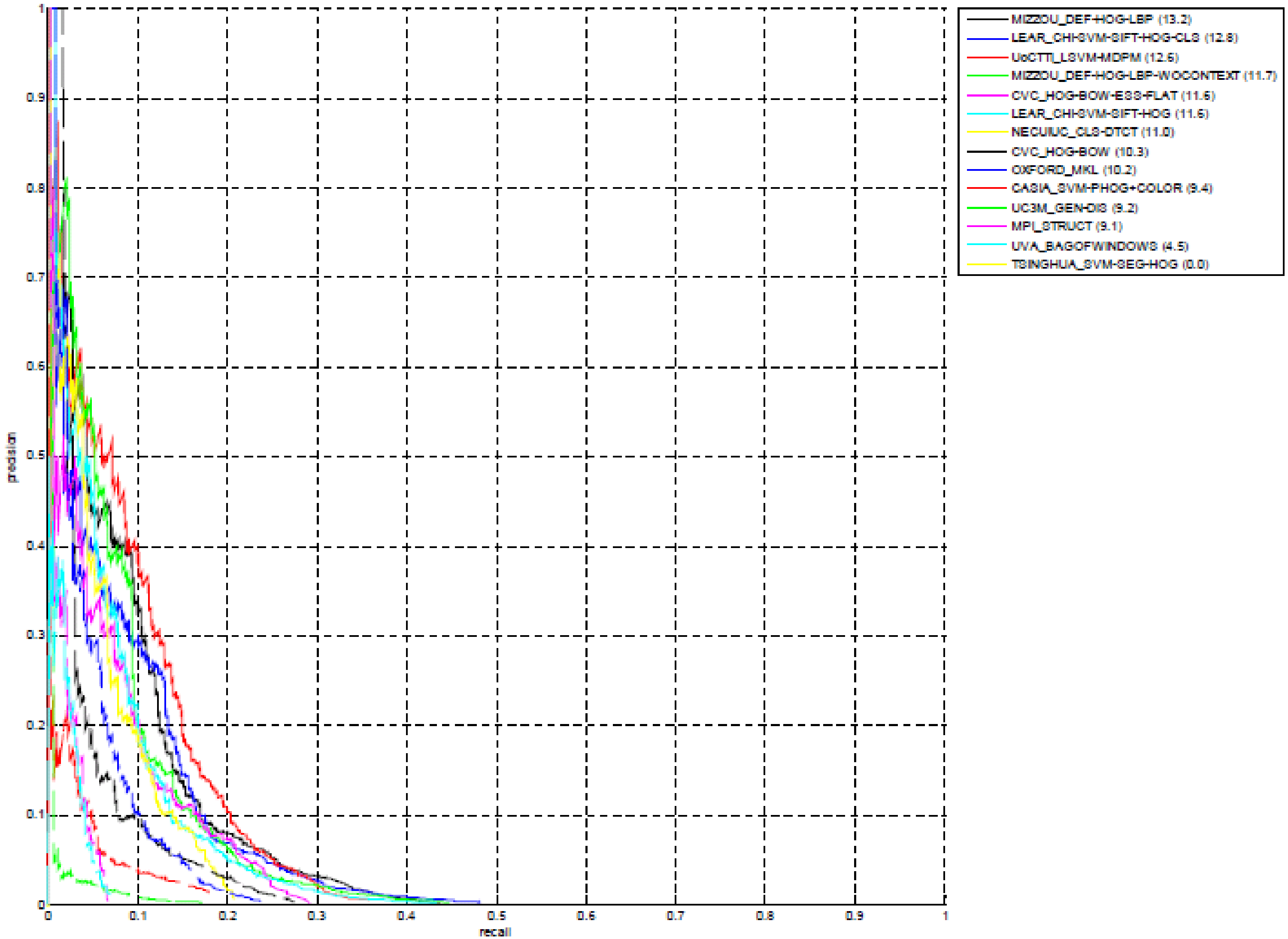

# Precision/Recall - Aeroplane



# Precision/Recall - Car

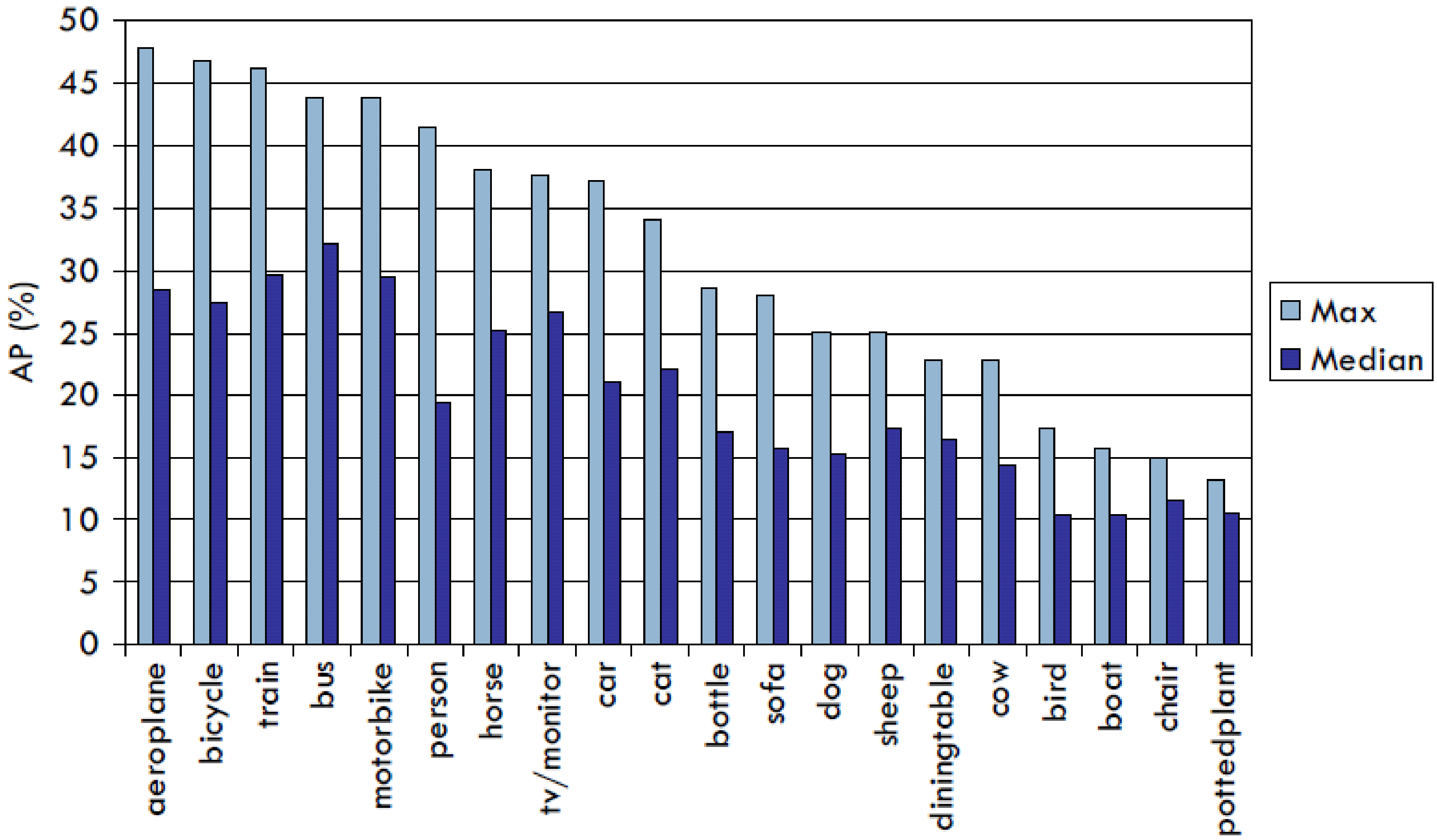


# Precision/Recall – Potted plant



# AP by Class

# Detection



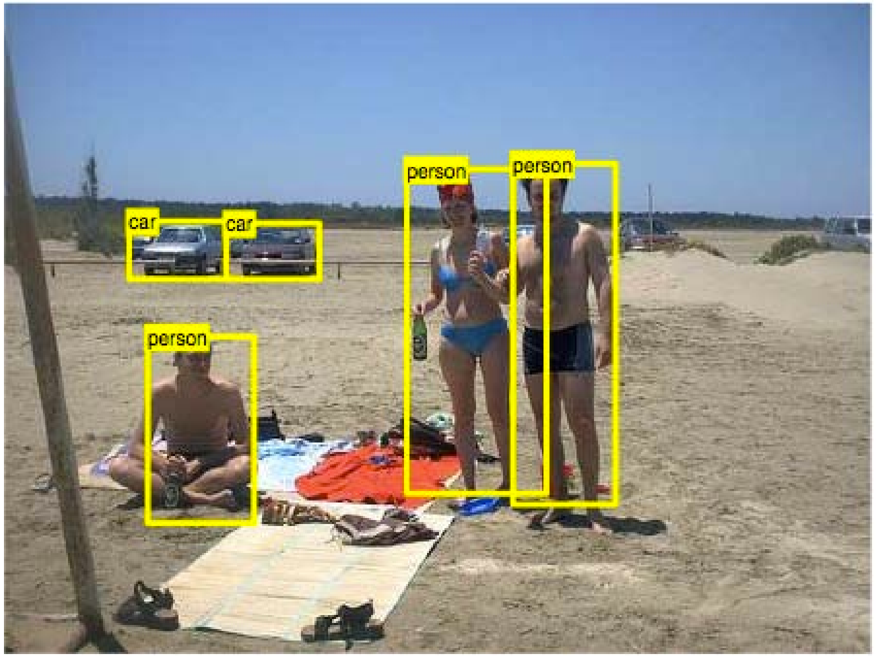

Wide variety of methods: sliding window, combination with whole image classifiers, segmentation based

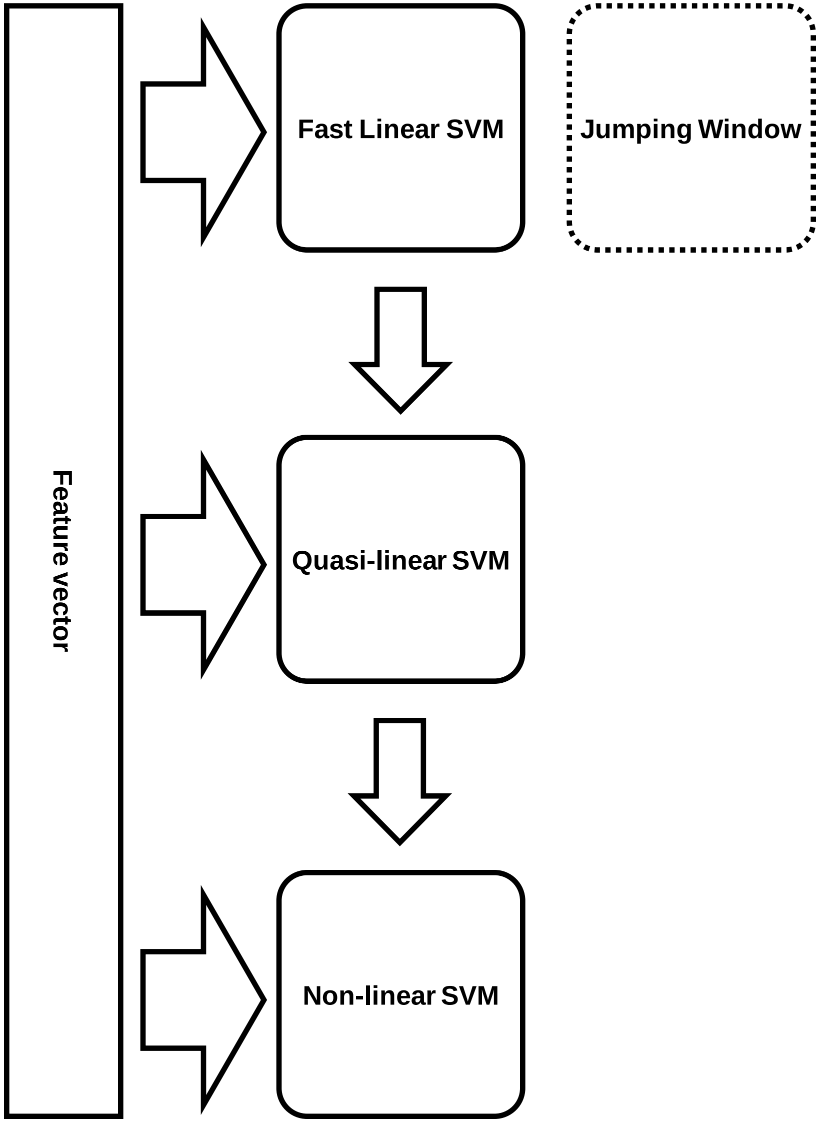
# Multiple Kernels for Object Detection

Andrea Vedaldi, Varun Gulshan,
Manik Varma, Andrew Zisserman

ICCV 2009

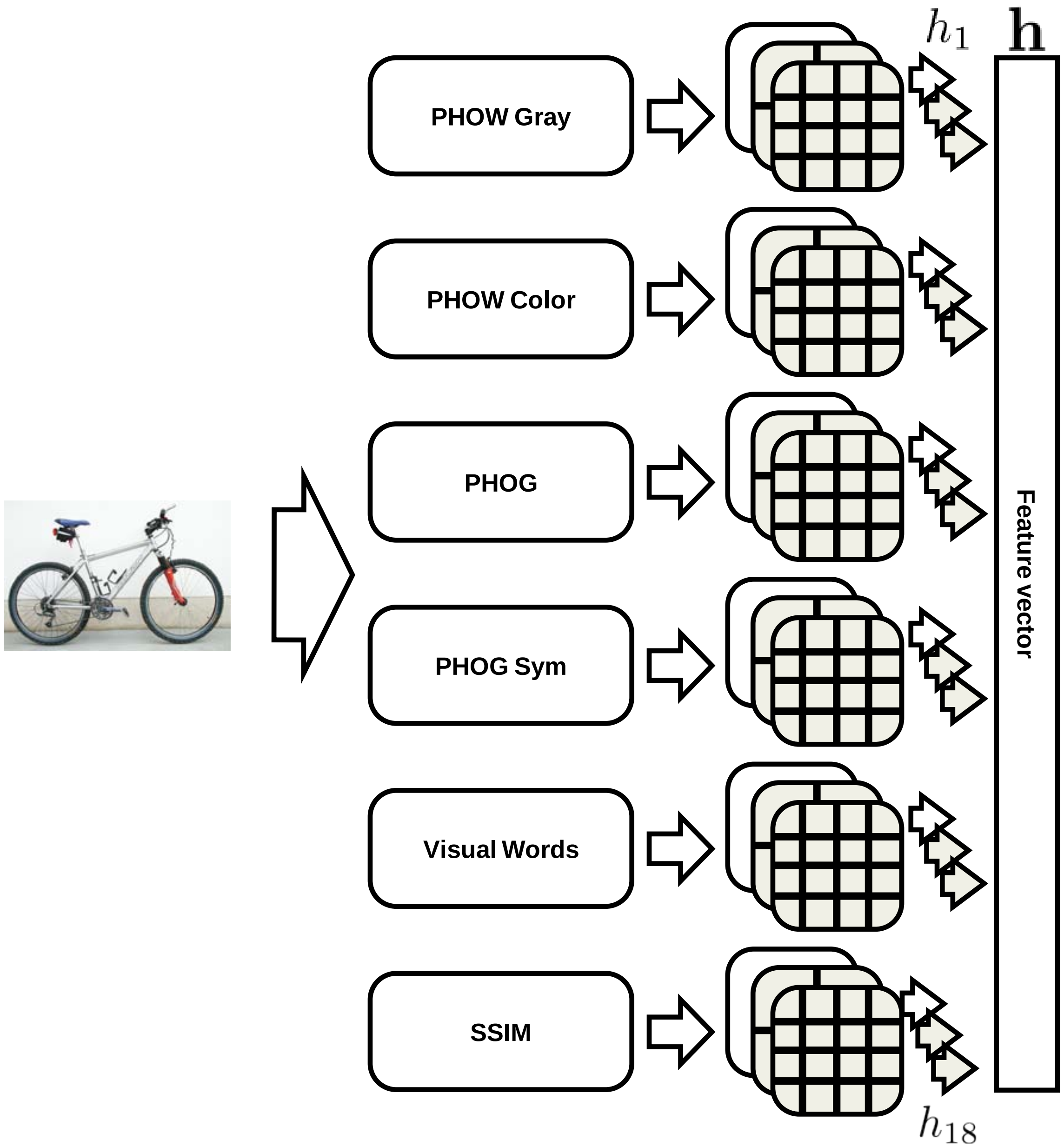
# Approach




- Three stage cascade
  - Each stage uses a more powerful and more expensive classifier
- Multiple kernel learning for the classifiers over multiple features
- Jumping window first stage

# Multiple Kernel Classification



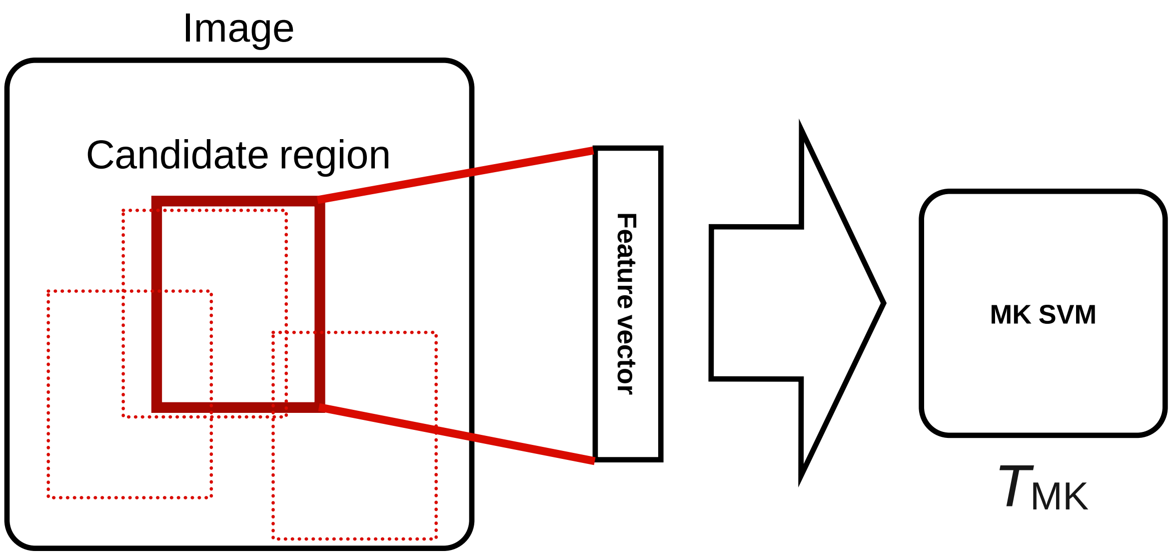

combine one kernel per histogram

$$K(\mathbf{h}, \mathbf{h}') = \sum_{i=1}^{F} d_i K(h_i, h_i')$$

[Varma & Rai, 2007]
[Gehler & Nowozin, 2009]

# Multiple Kernel Detection: Challenges

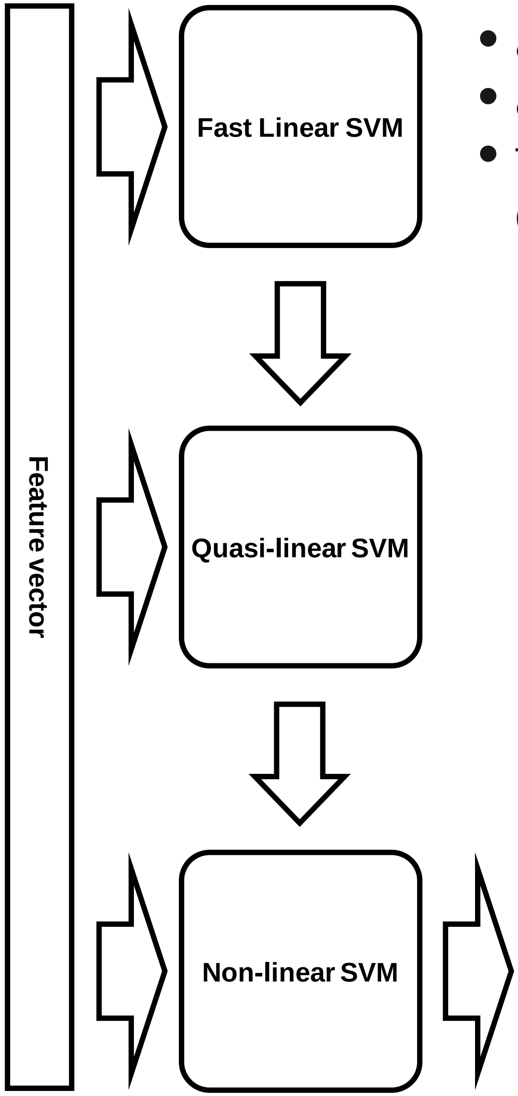
- Goal: sliding window MK classifier
  - Inference space is huge
  - #windows = 100 millions
  - TMK = seconds




Time required:
$T_{MK}$ × #windows

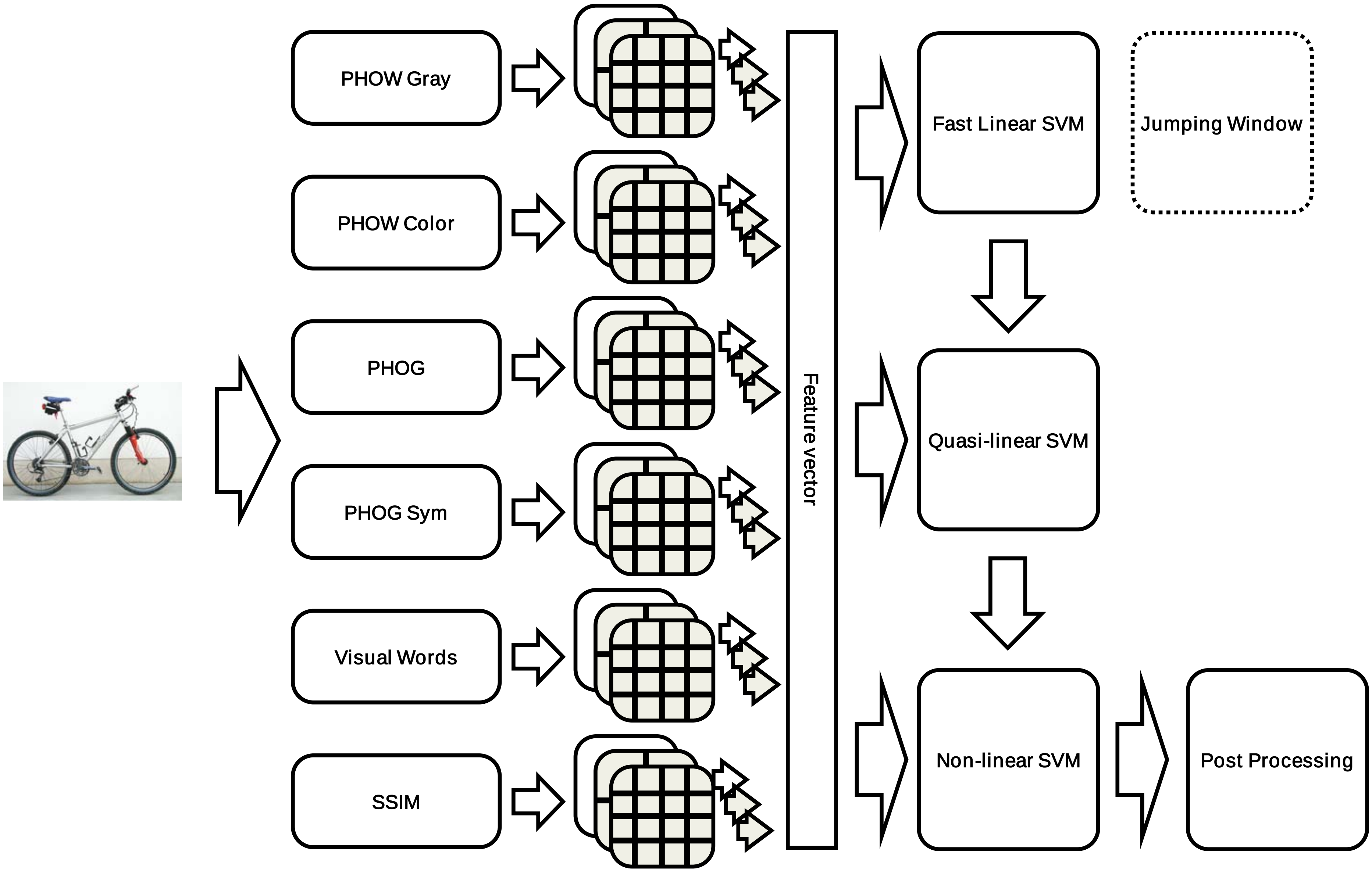
Excruciatingly slow (days per image)

# Cascade

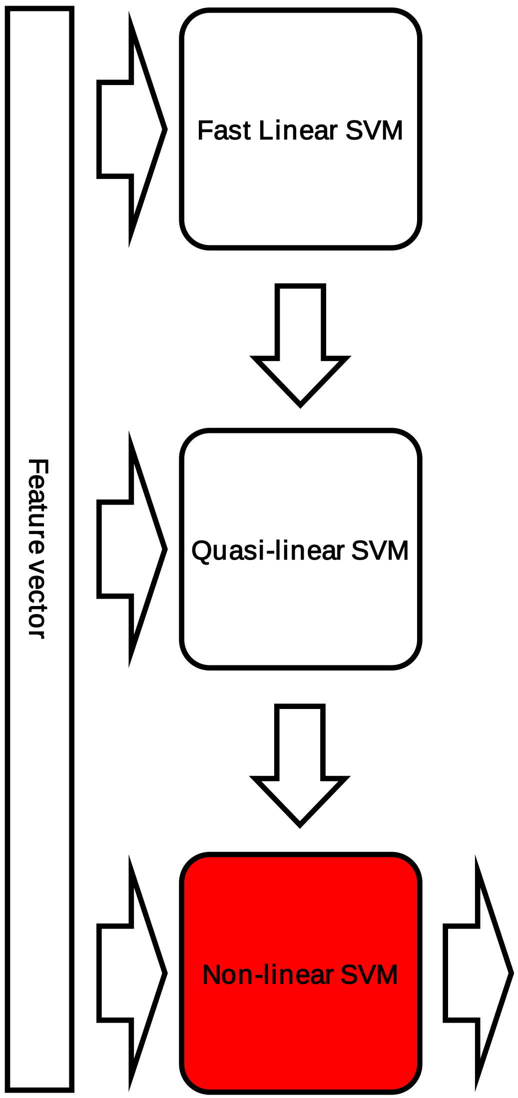



- all full MK SVMs
- all look at all features
- trade-off speed and power by choosing the kernel structure

# Architecture

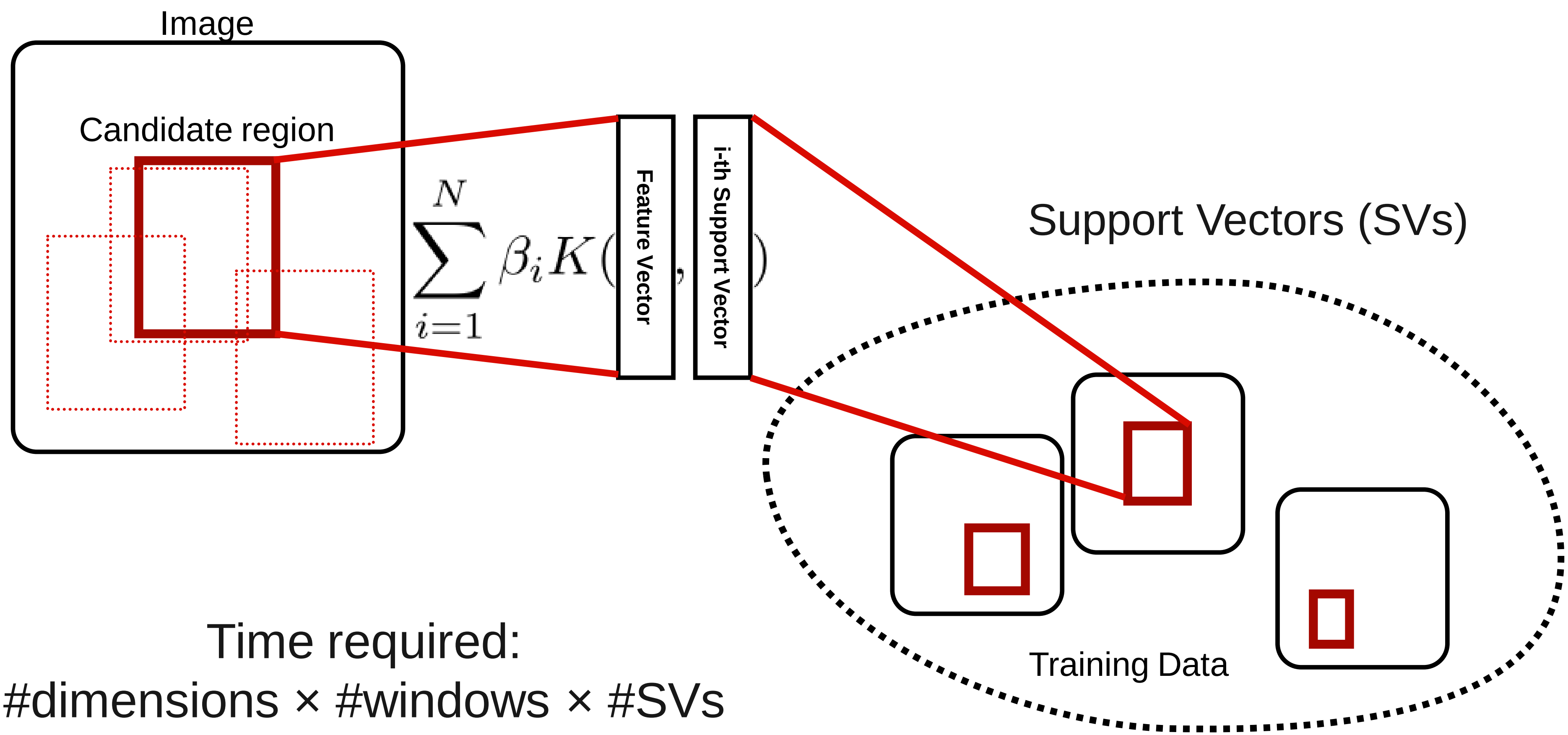


# Cascade



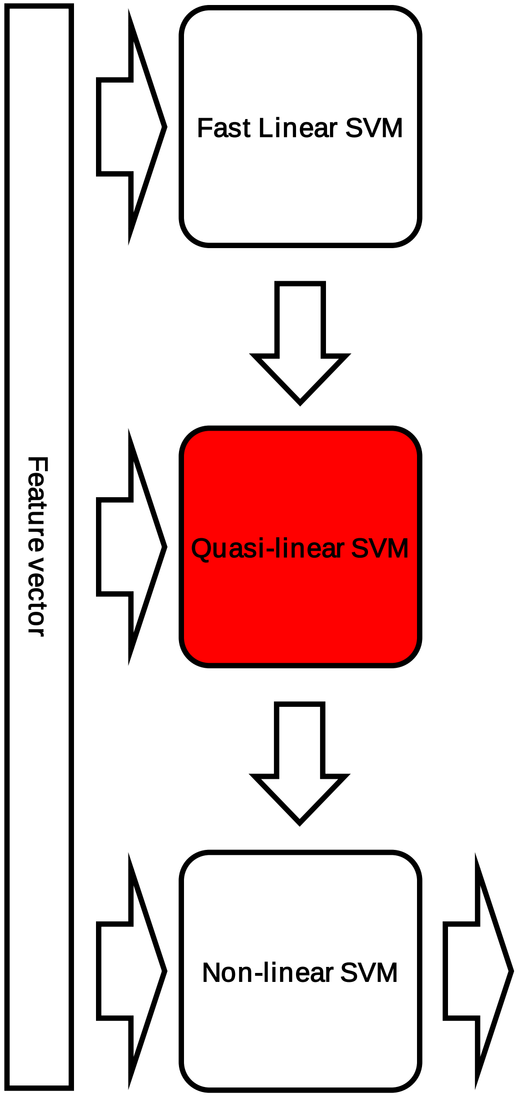
Feature vector
Fast Linear SVM
Quasi-linear SVM
Non-linear SVM

# Non-linear sliding SVM



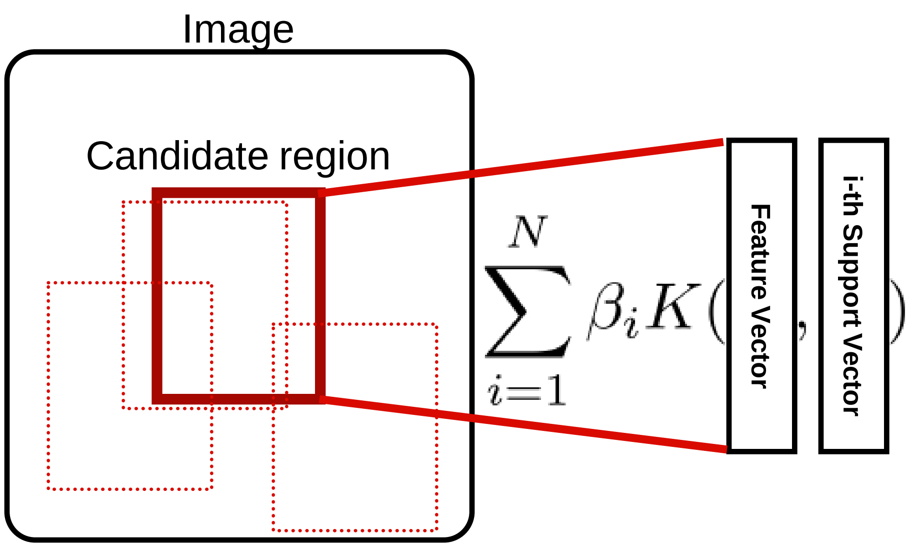

Time required:
#dimensions × #windows × #SVs

# Cascade



Feature vector
Fast Linear SVM
Quasi-linear SVM
Non-linear SVM

# Quasi-linear SVM




Time required:
#dimensions × #windows × ~~#SVs~~

#dimensions × #windows

Quasi-linear (or additive) kernel decompose as:

$$K(x, y) = \sum_{j=1}^{d} k(x_j, y_j)$$

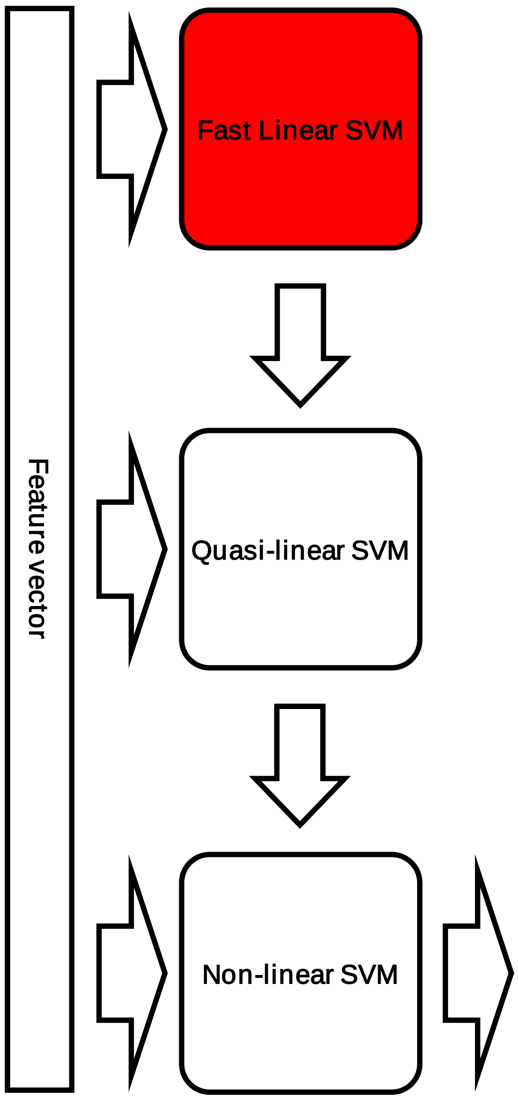
Thus SVM score rewrites:

$$\sum_{j=1}^{d} \underbrace{\sum_{i=1}^{N} \beta_j k(x_j, y_{ij})}_{\psi_j(x_j)}$$

Pre-compute look-up table.

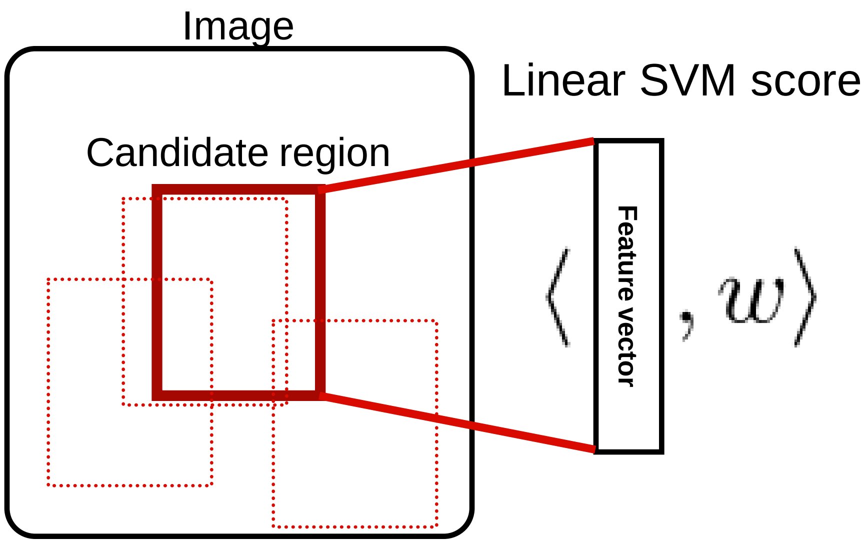
Maji, Berg, Malik, CVPR 08

# Cascade



Feature vector
Fast Linear SVM
Quasi-linear SVM
Non-linear SVM

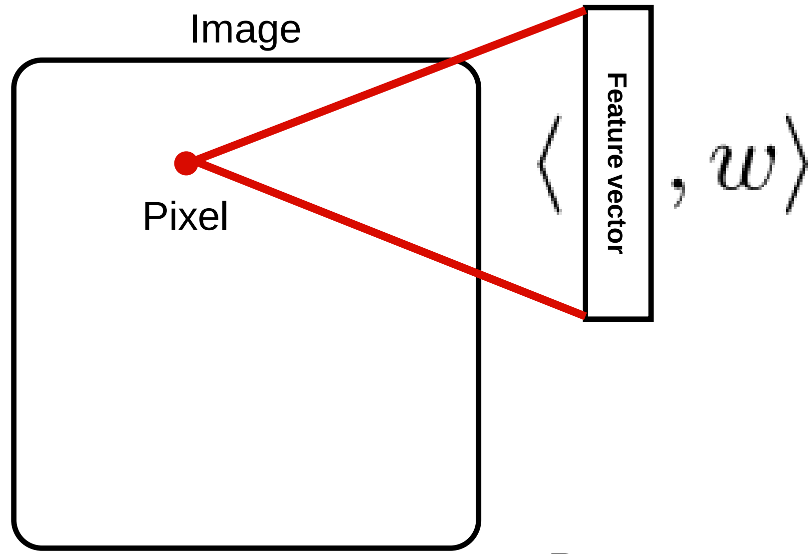
# Fast linear SVM



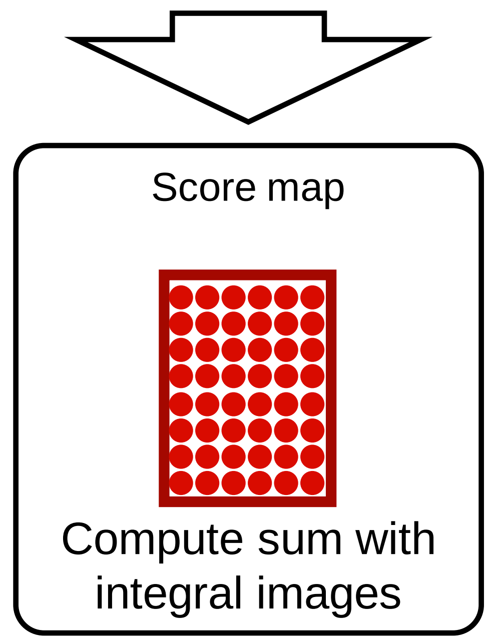

Time required:

#~~dimensions~~ × #windows × #~~SVs~~

#windows

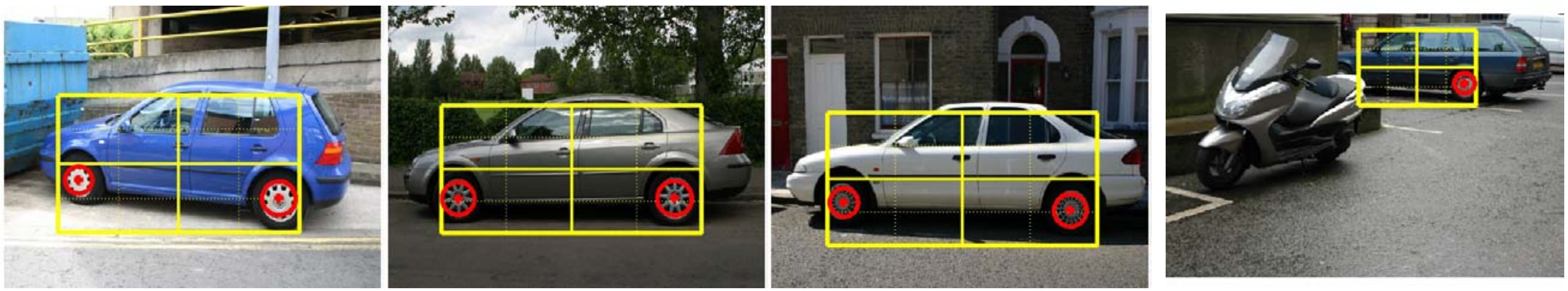



Pre-compute scores for each pixel.

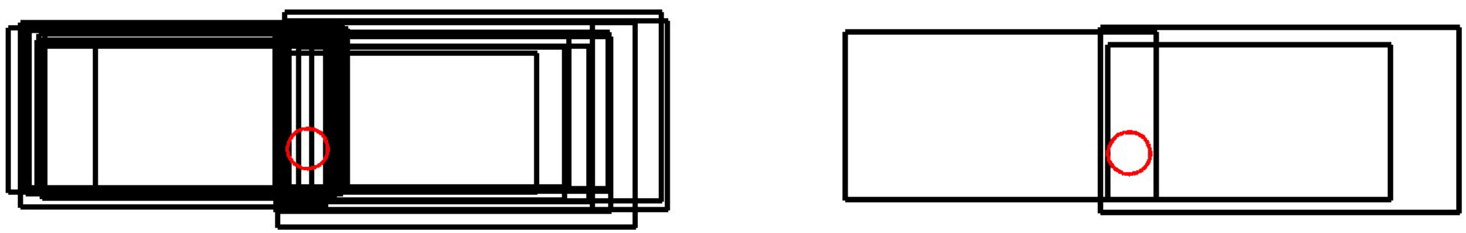


# Jumping window

Training

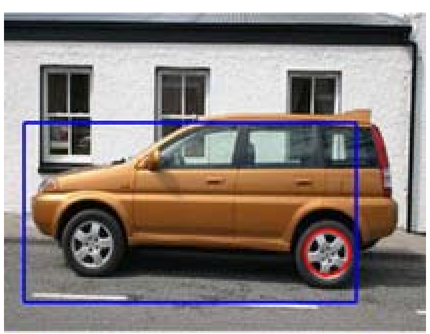


Position of visual word with respect to the object



learn the position/scale/aspect ratio of the ROI with respect to the visual word

Detection

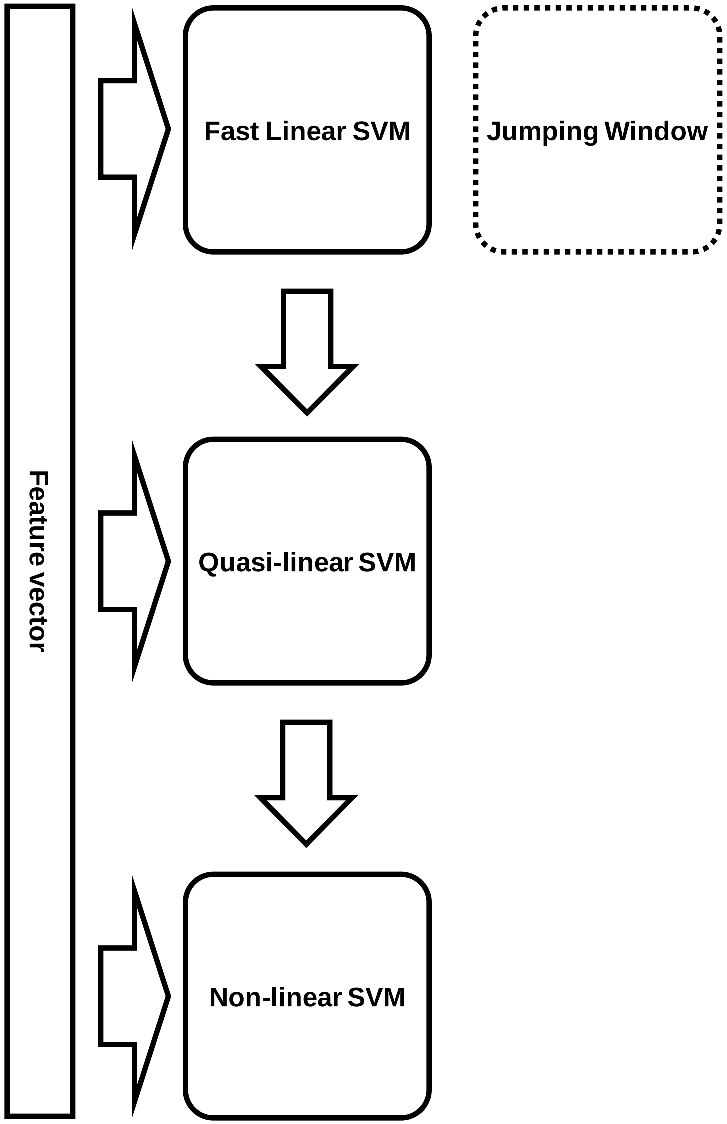


Hypothesis

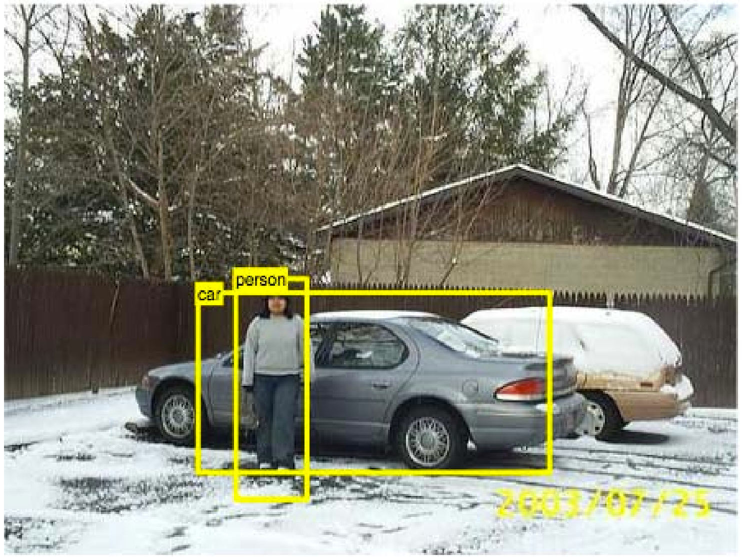
Handles change of aspect ratio

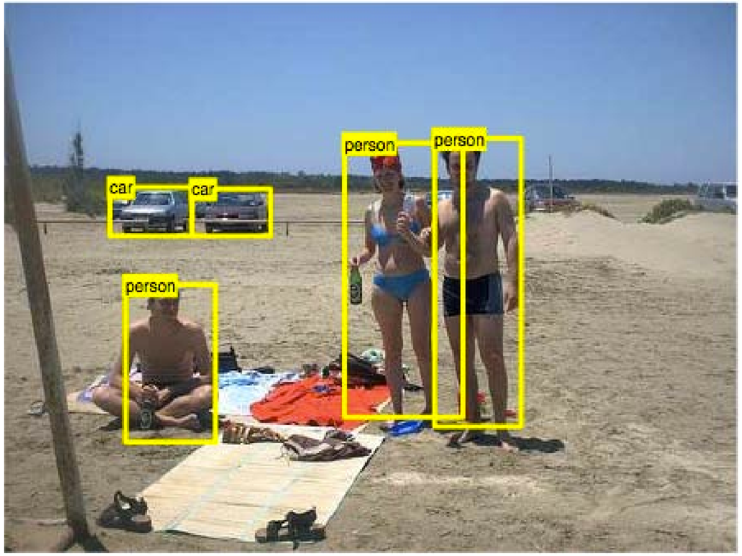
# SVMs overview

- **First stage**
  - linear SVM
  - (or jumping window)
  - time: #windows
- **Second stage**
  - quasi-linear SVM
  - $\chi^2$ kernel
  - time: #windows × #dimensions
- **Third stage**
  - non-linear SVM
  - $\chi^2$-RBF kernel
  - time:
    #windows × #dimensions × #SVs



# Results

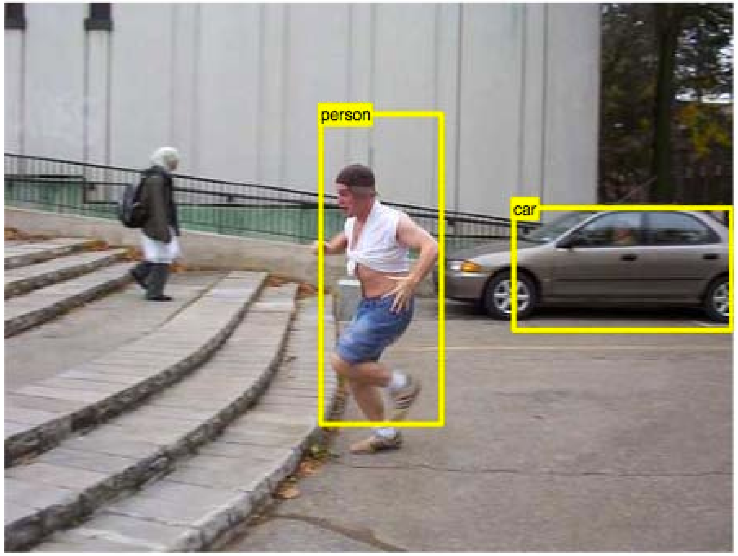

car
person
2003/07/25


person
person
car
car
person

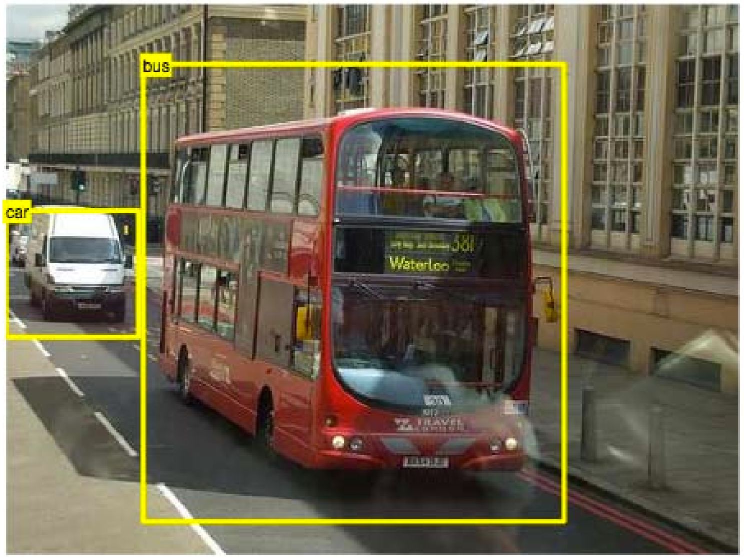

car
person
HONDA
50


person
person
car

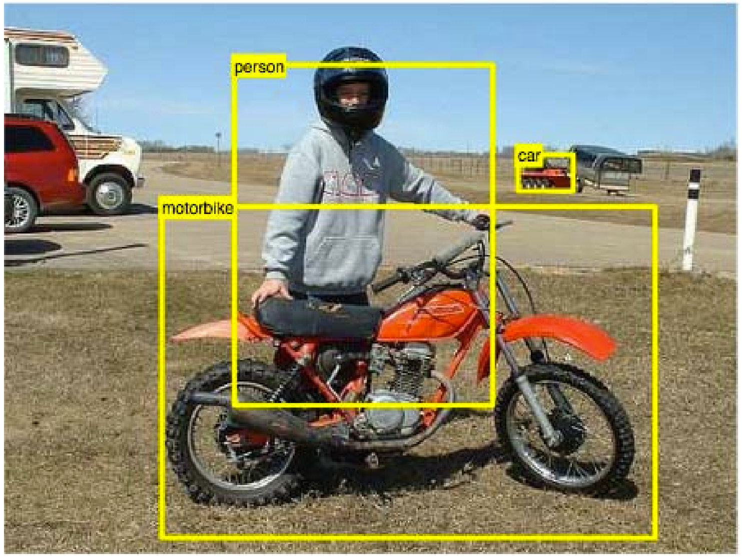
# Results

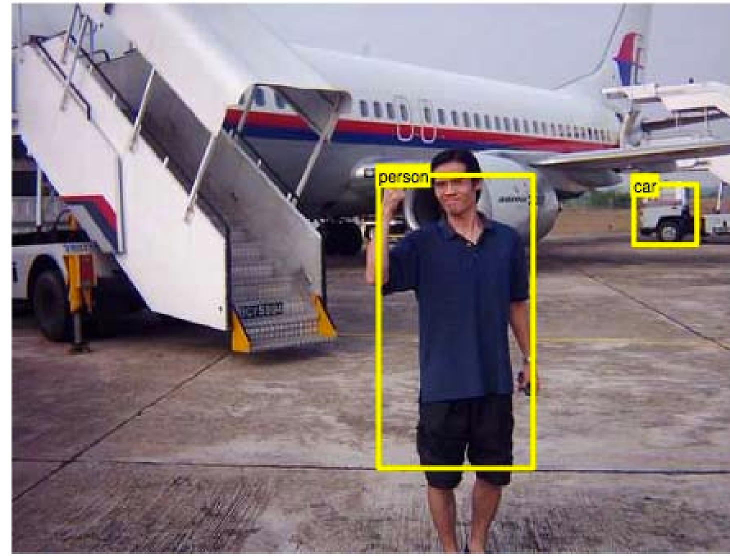


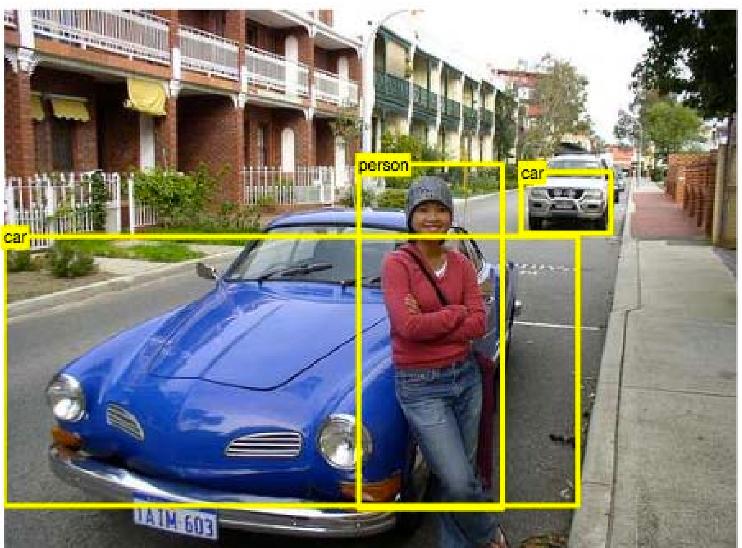


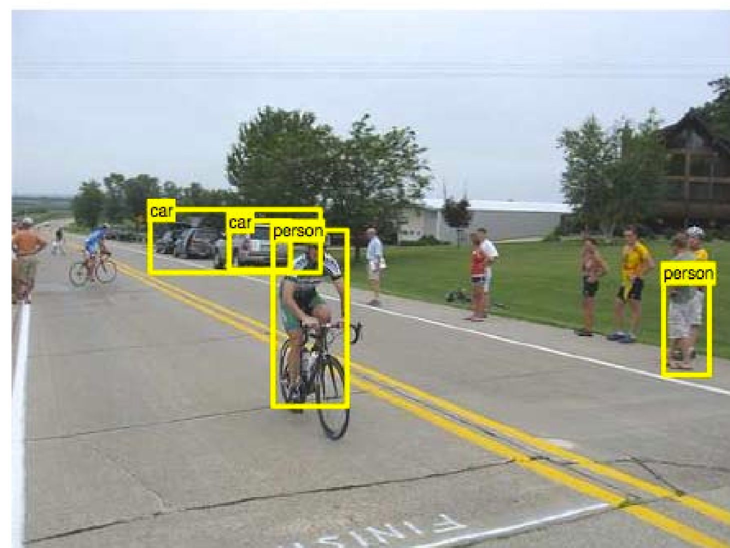



# Results


person
car
motorbike

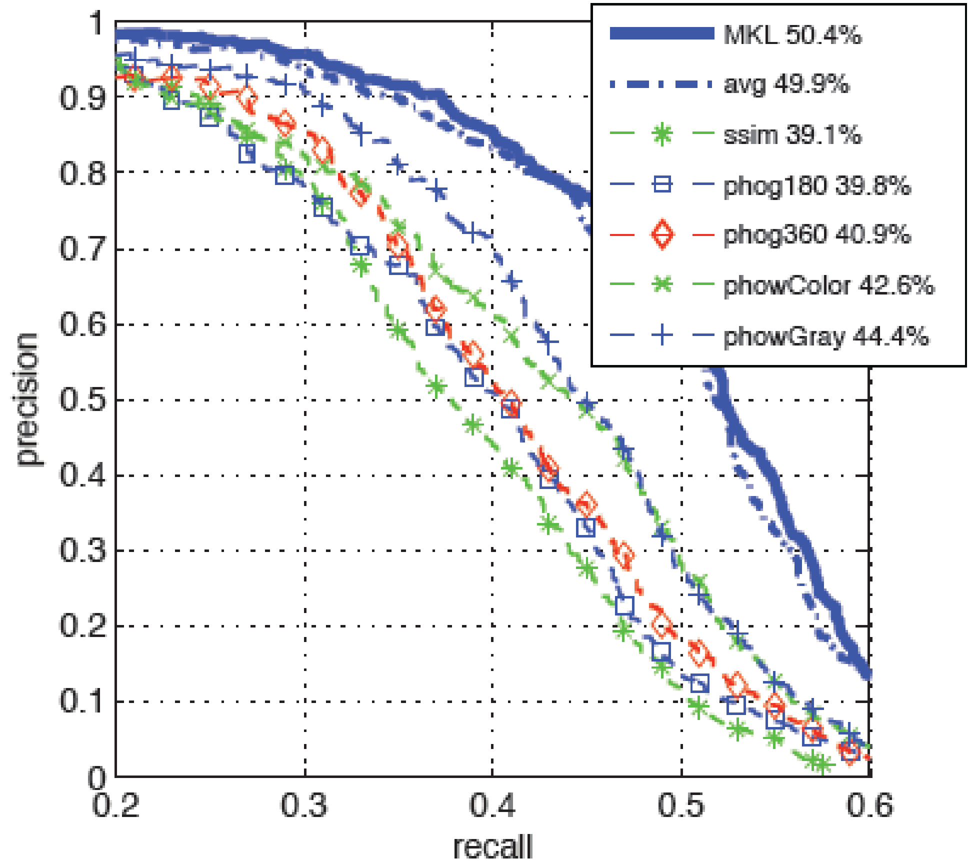

person
car


person
car
car

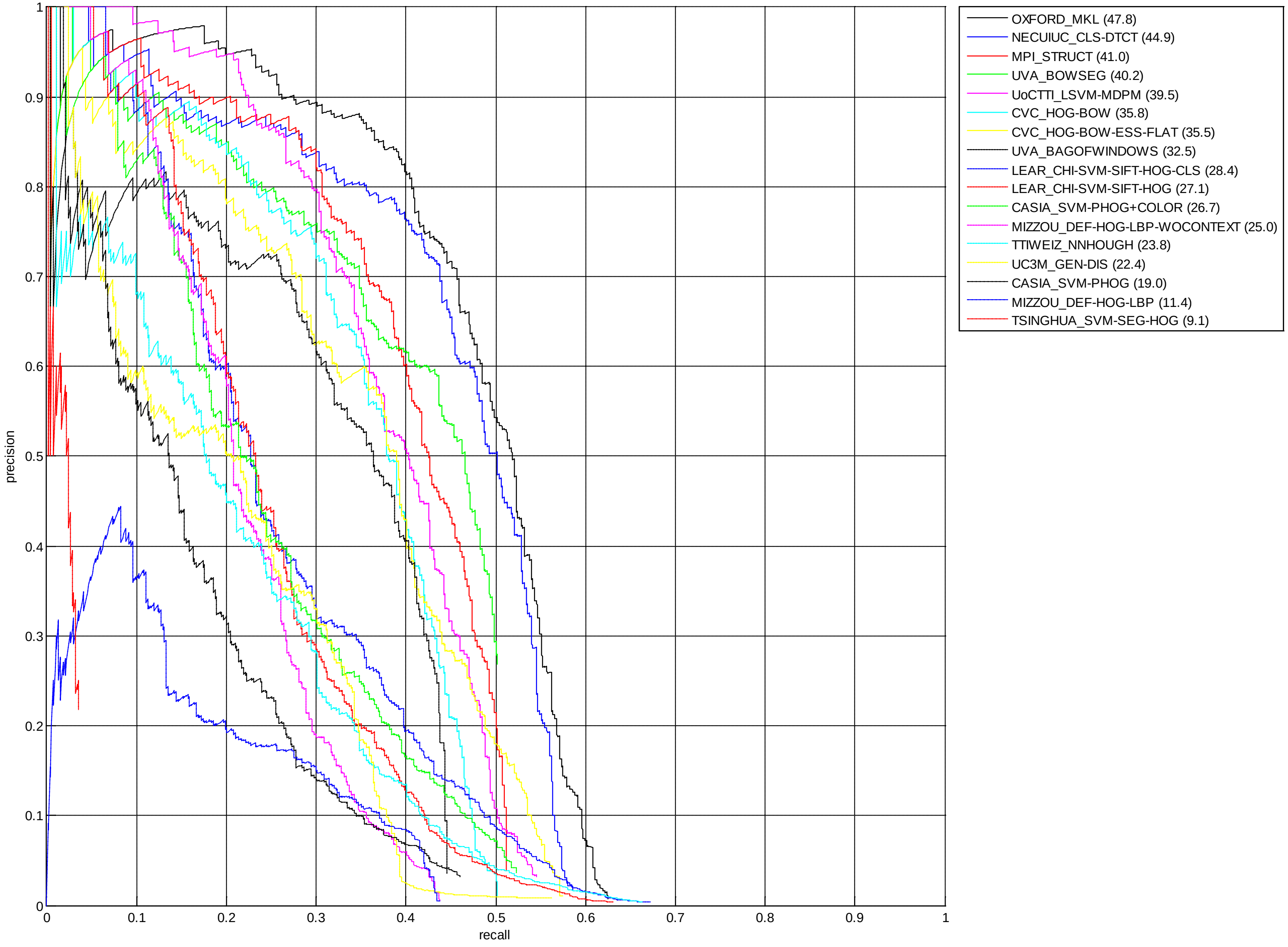

car
car
person
person

# Single Kernel vs. Multiple Kernels

- Multiple Kernels gives substantial boost
- Multiple Kernel Learning:
  - small improvement over averaging
  - sparse feature selection



# Precision/Recall: VOC2009 Aeroplane

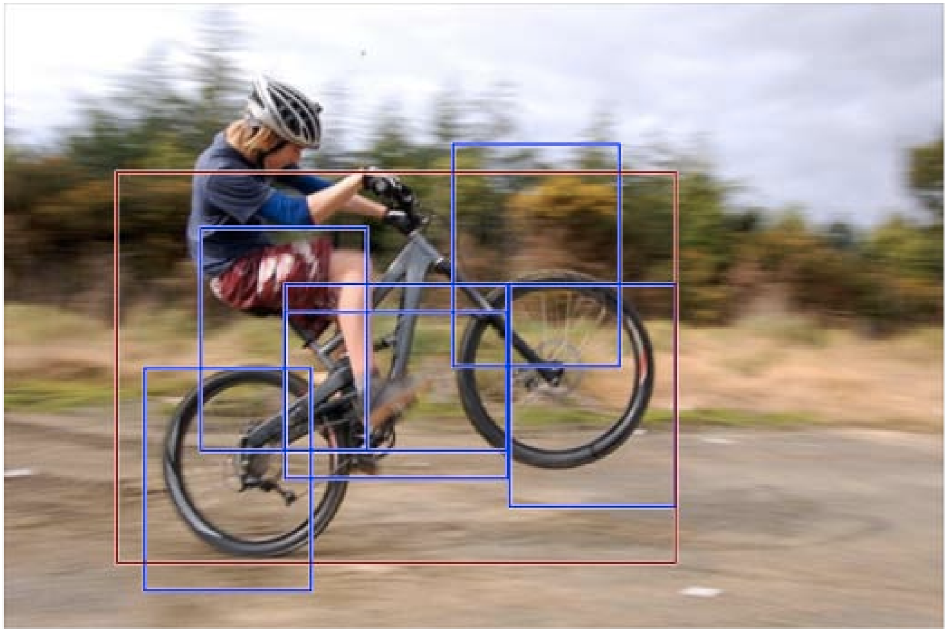


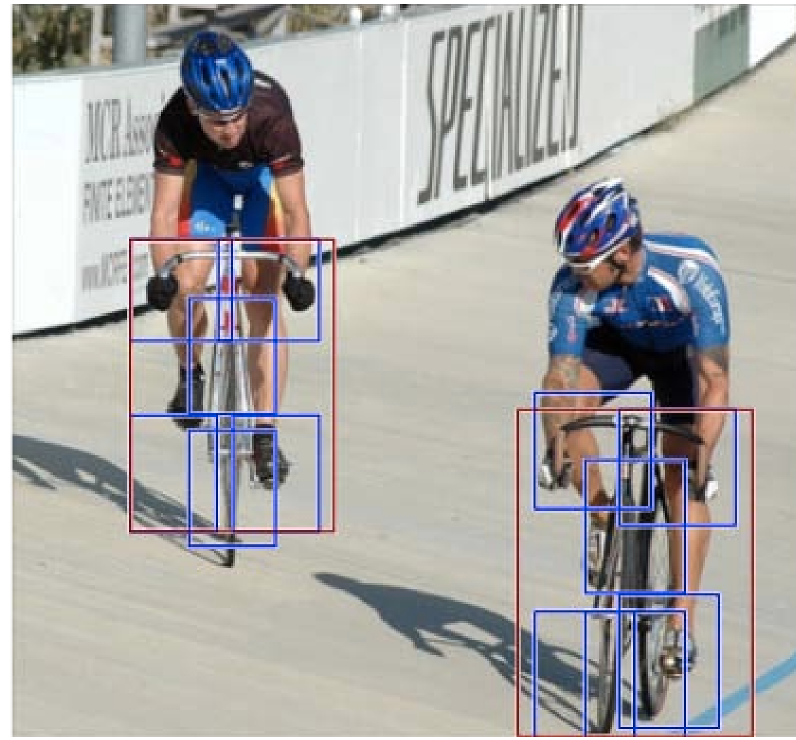
# Object Detection with Discriminatively Trained Part Based Models

Pedro F. Felzenszwalb, David Mcallester,
Deva Ramanan, Ross Girshick

PAMI 2010

# Approach

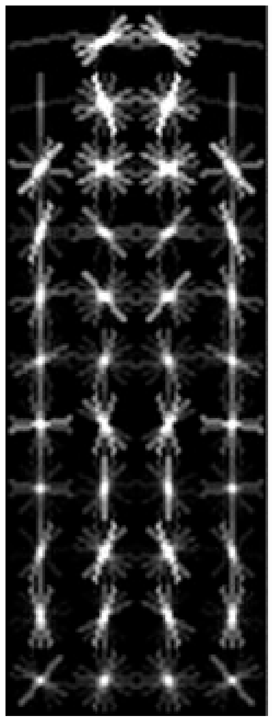


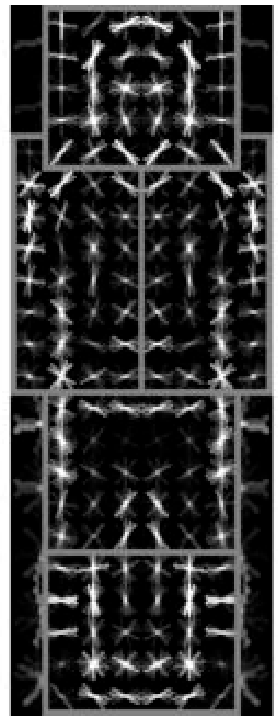


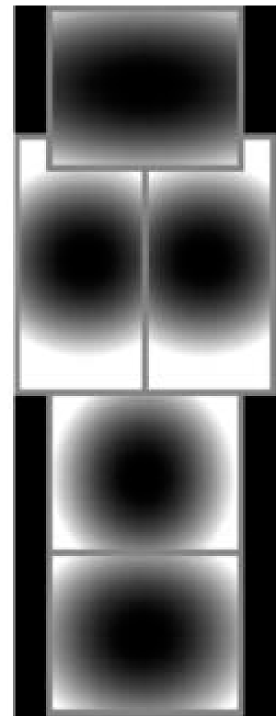
- Mixture of deformable part-based models
  - One component per “aspect” e.g. front/side view
- Each component has global template + deformable parts
- Discriminative training from bounding boxes alone

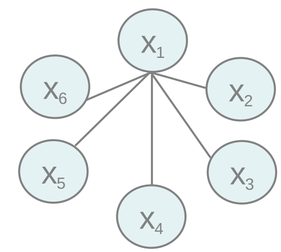
# Example Model

- One component of person model







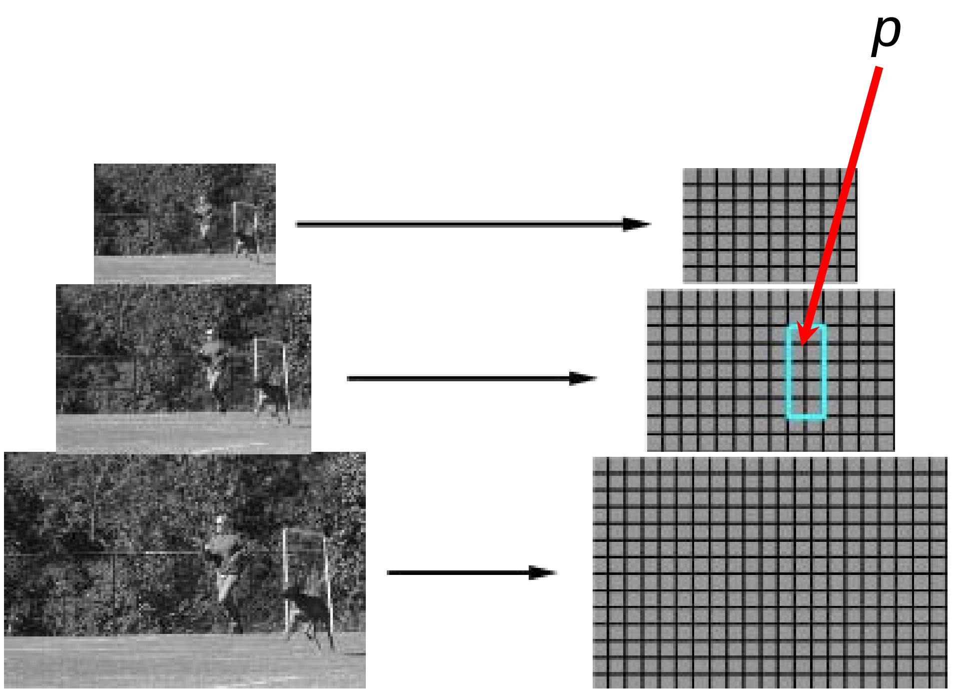



root filters
coarse resolution

part filters
finer resolution

deformation
models

# Starting Point: HOG Filter




HOG pyramid $H$

Filter $F$

Score of $F$ at position $p$ is
$F \cdot \varphi(p, H)$

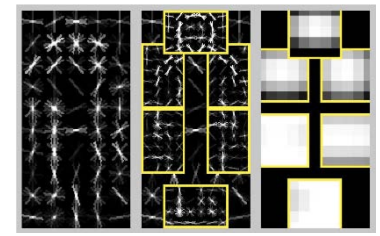
$\varphi(p, H)$ = concatenation of HOG features from subwindow specified by $p$

- Search: sliding window over position and scale
- Feature extraction: HOG Descriptor
- Classifier: Linear SVM

Dalal & Triggs [2005]

# Object Hypothesis

- Position of root + each part
- Each part: HOG filter (at higher resolution)



$z = (p_0, \ldots, p_n)$

$p_0$ : location of root

$p_1, \ldots, p_n$ : location of parts

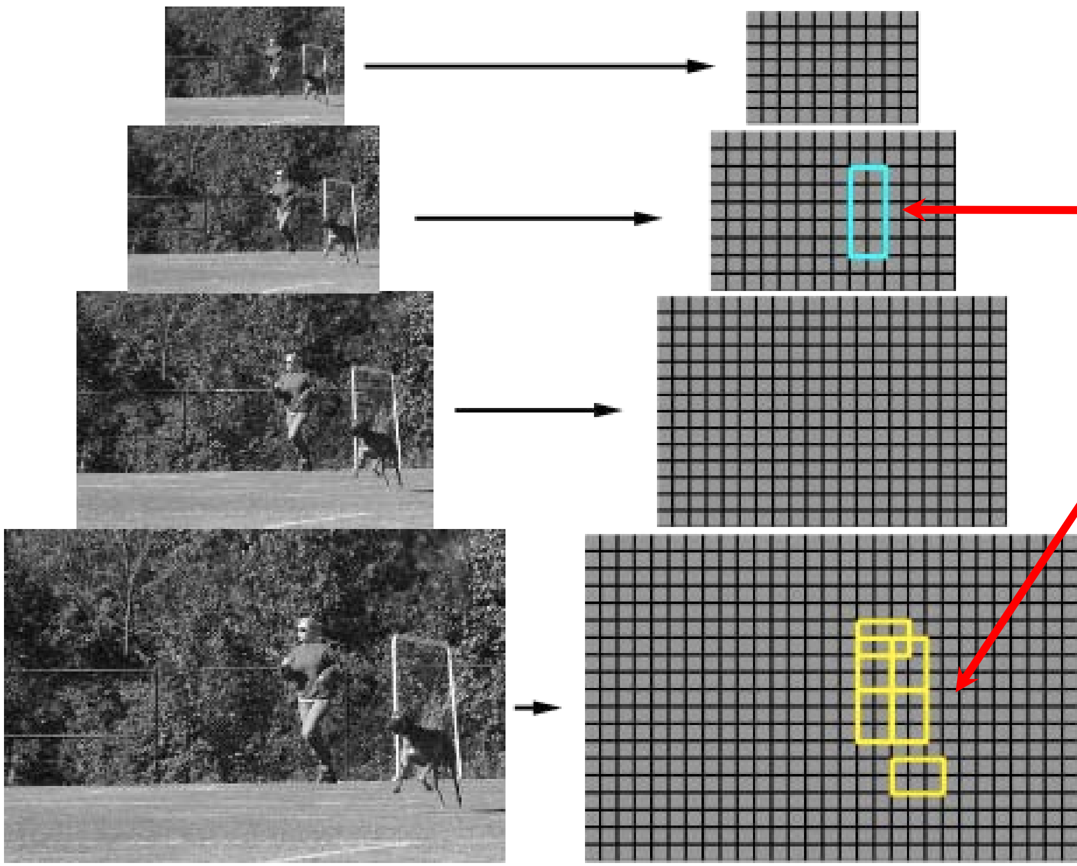


Image pyramid

HOG feature pyramid

Score is sum of filter scores minus deformation costs

# Score of a Hypothesis

Appearance term | Spatial prior

$$\text{score}(p_0, \ldots, p_n) = \sum_{i=0}^{n} F_i \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot (dx_i^2, dy_i^2)$$

filters

displacements

deformation parameters

$$\text{score}(z) = \beta \cdot \Psi(H, z)$$

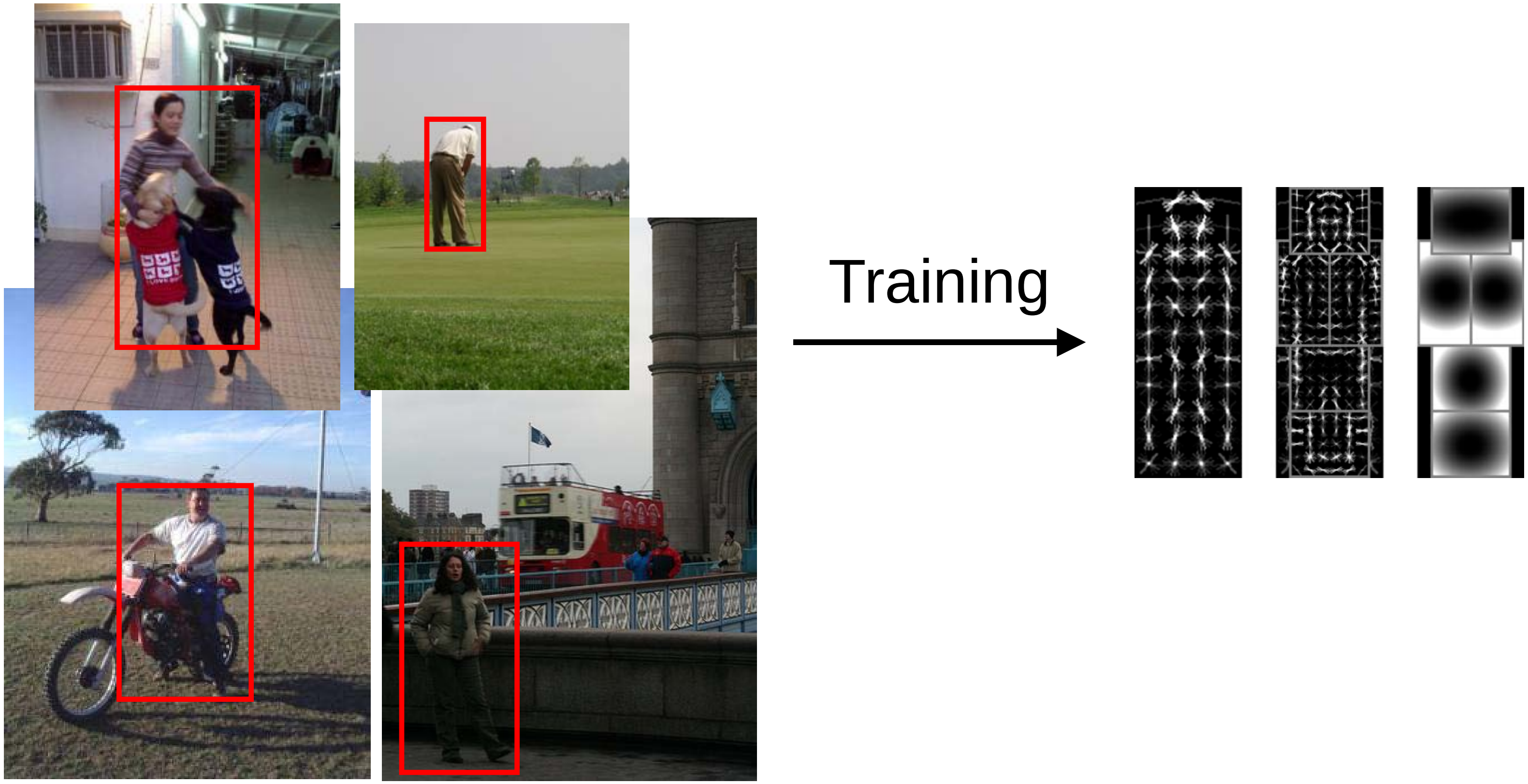
concatenation of filters and deformation parameters

concatenation of HOG features and part displacement features

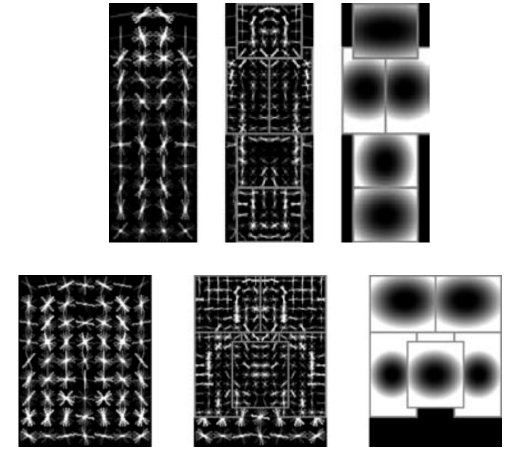
- Linear classifier applied to feature subset defined by hypothesis

# Training

- Training data = images + bounding boxes
- Need to learn: model structure, filters, deformation costs



# Latent SVM (MI-SVM)



Classifiers that score an example $x$ using

$$f_\beta(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z)$$

$\beta$ are model parameters

$z$ are latent values

- Which component?
- Where are the parts?

Training data $D = (\langle x_1, y_1 \rangle, \ldots, \langle x_n, y_n \rangle) \quad y_i \in \{-1, 1\}$

We would like to find $\beta$ such that: $y_i f_\beta(x_i) > 0$

Minimize

$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C \sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

SVM objective

# Latent SVM Training

$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n} \max(0, 1 - y_i f_\beta(x_i))$$

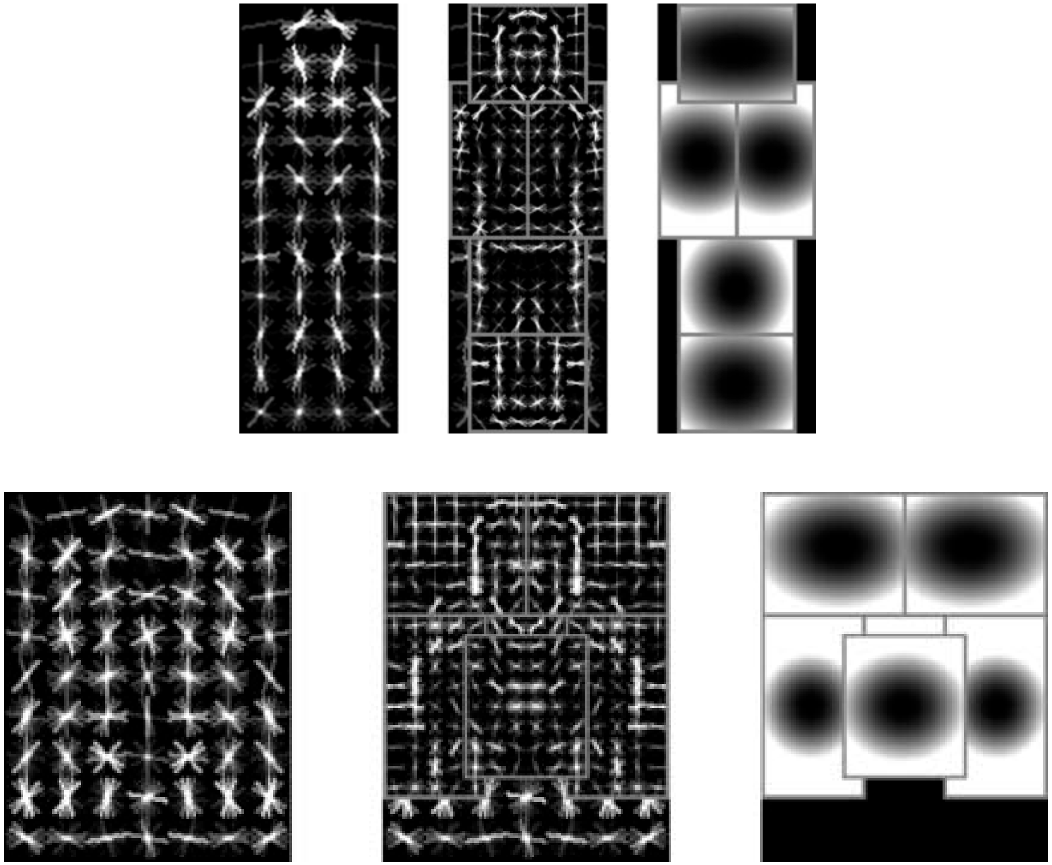
- Convex if we fix z for positive examples

- Optimization:
  - Initialize $\beta$ and iterate:
    - Pick best $z$ for each positive example
    - Optimize $\beta$ with $z$ fixed

Alternation strategy

- Local minimum: needs good initialization
  - Parts initialized heuristically from root

# Person Model



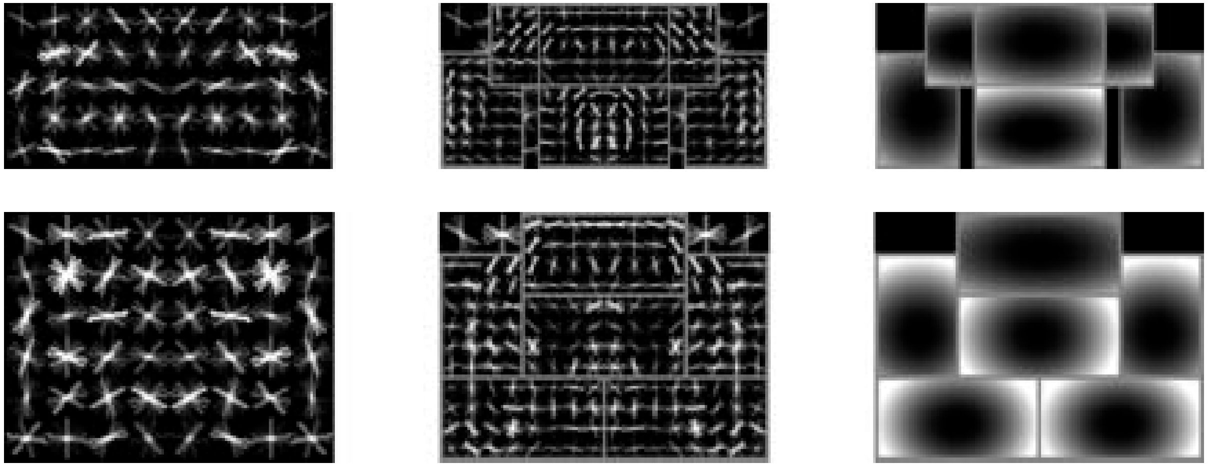
root filters
coarse resolution

part filters
finer resolution

deformation
models

Handles partial occlusion/truncation

# Car Model



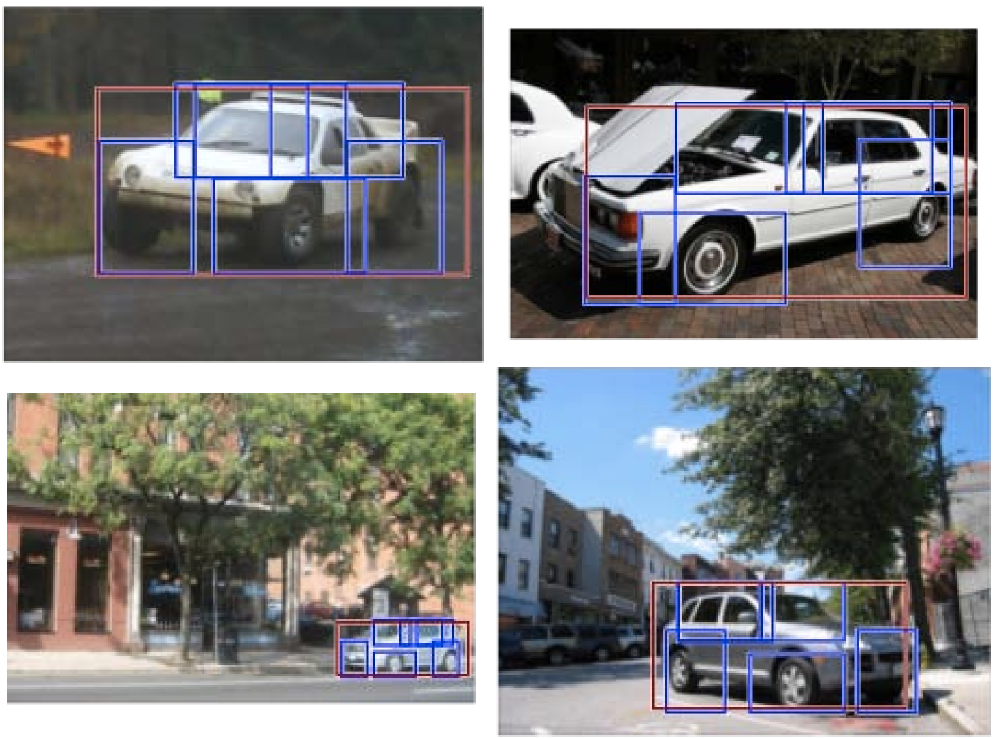
root filters
coarse resolution

part filters
finer resolution

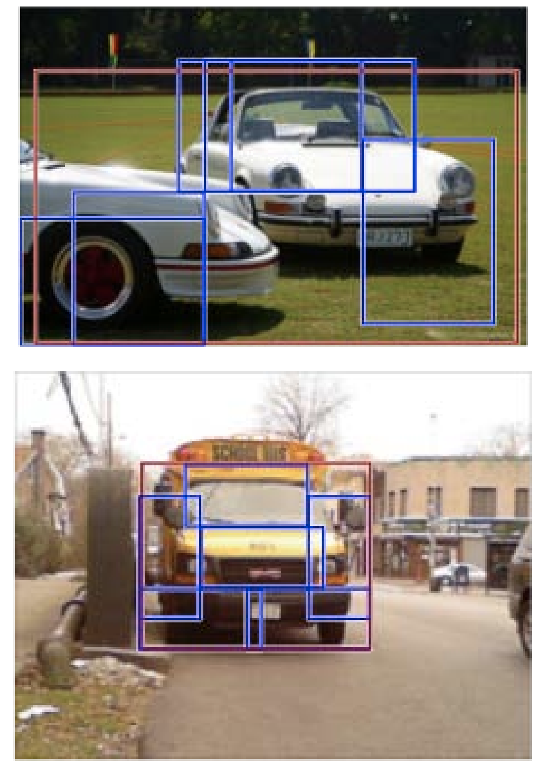
deformation
models

# Car Detections

high scoring true positives

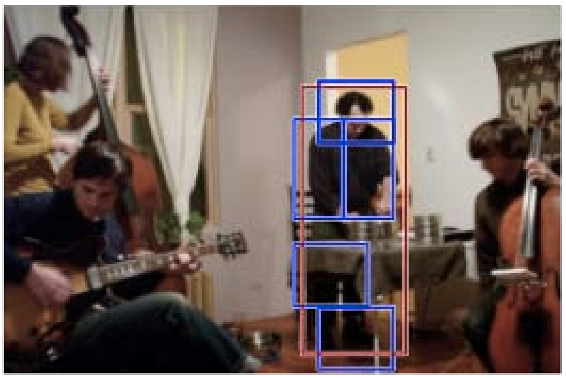


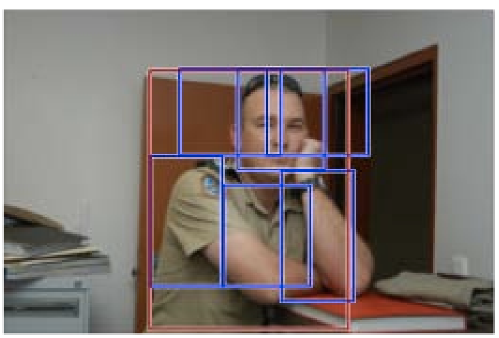
high scoring false positives

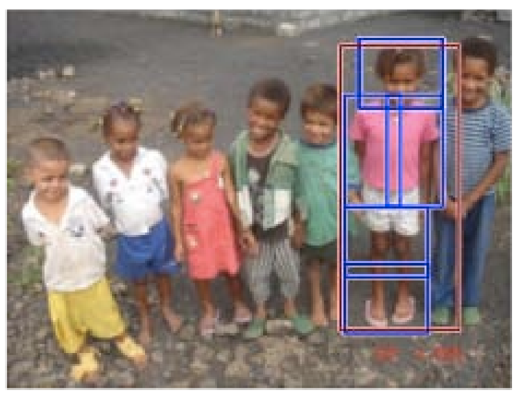


# Person Detections

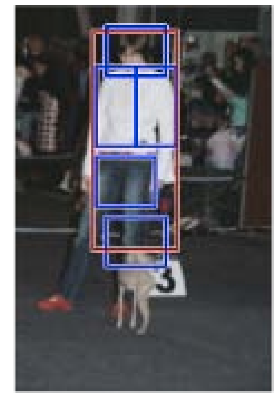
high scoring true positives



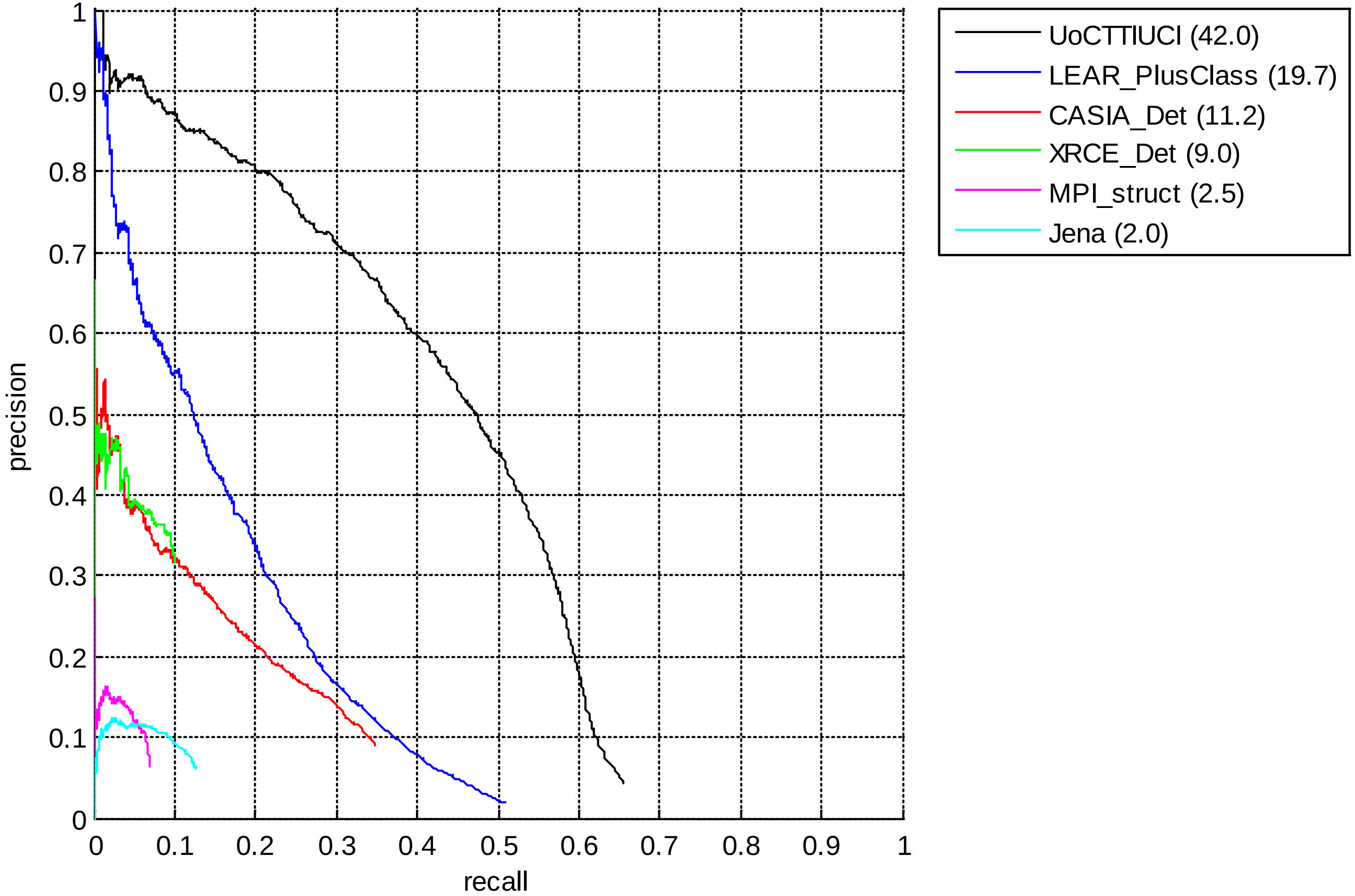




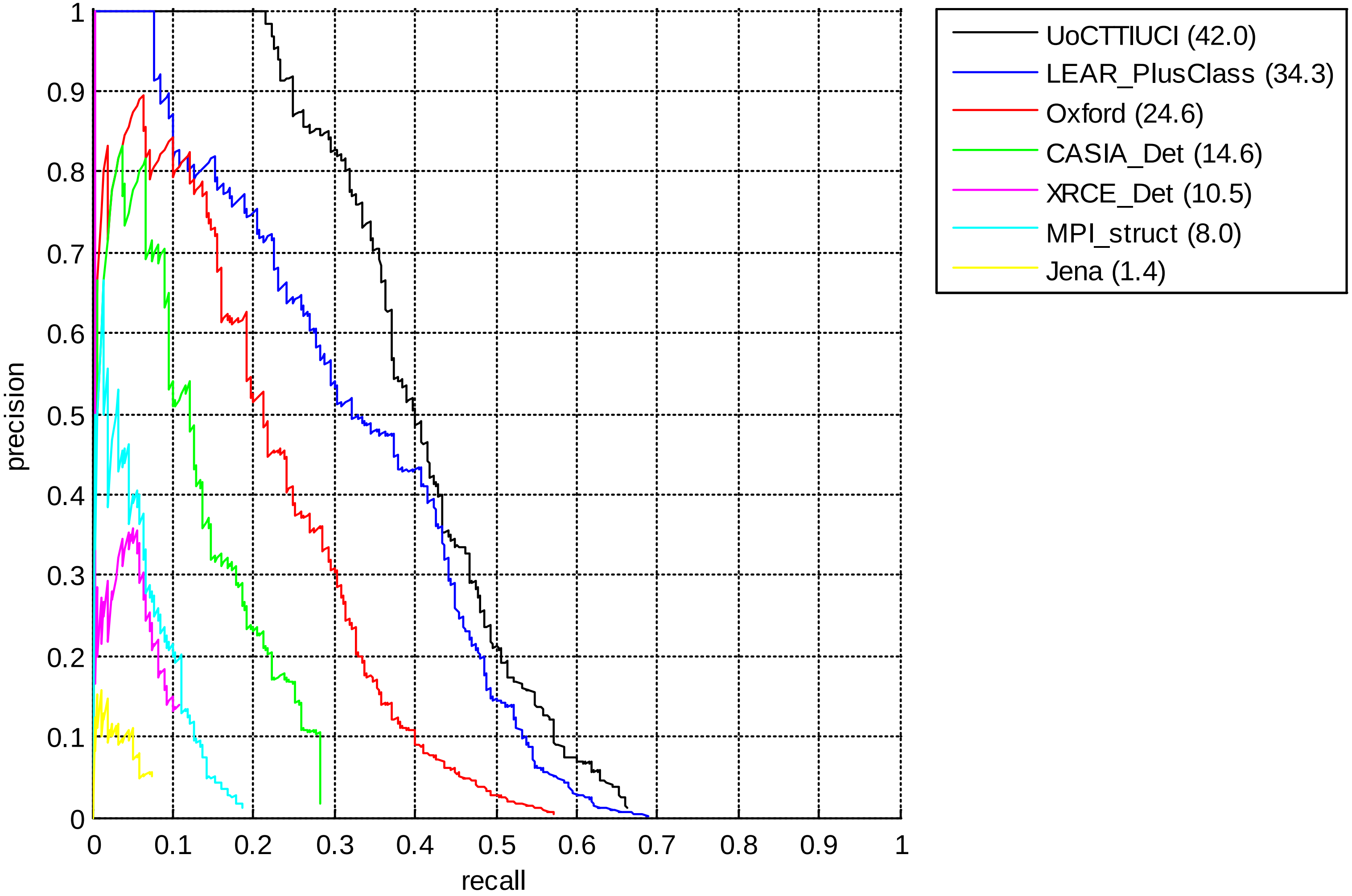
high scoring false positives
(not enough overlap)



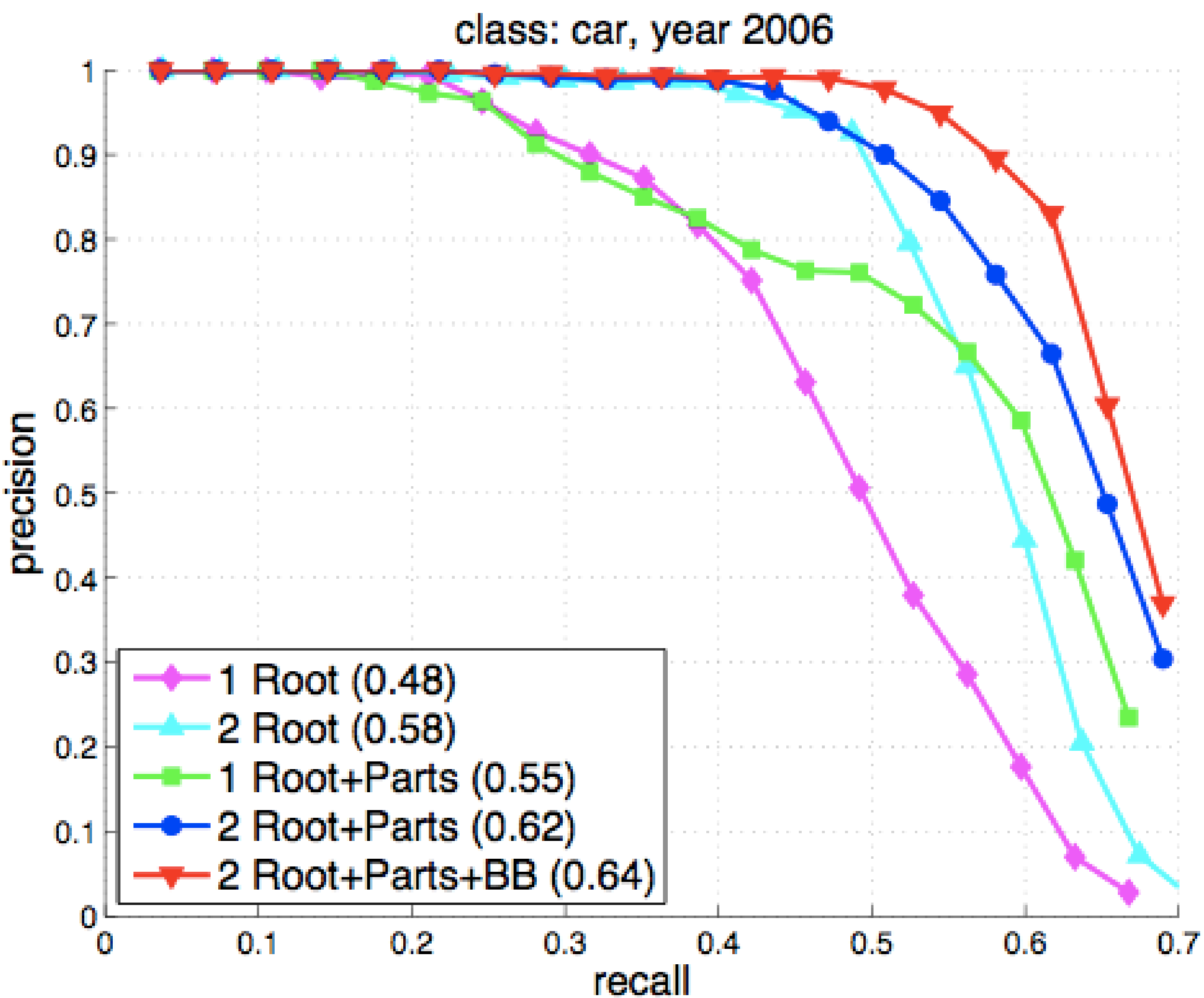
# Precision/Recall: VOC2008 Person



# Precision/Recall: VOC2008 Bicycle

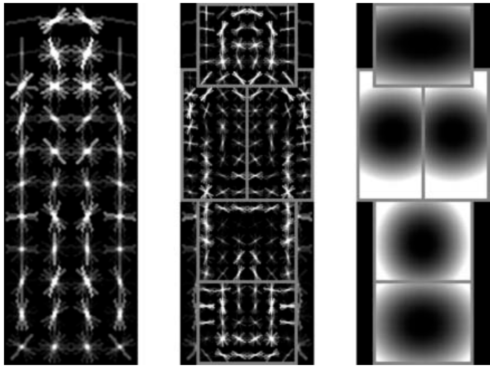


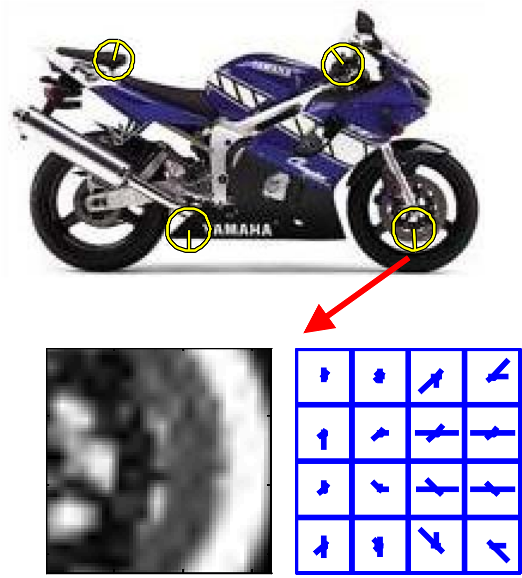
# Comparison of Models

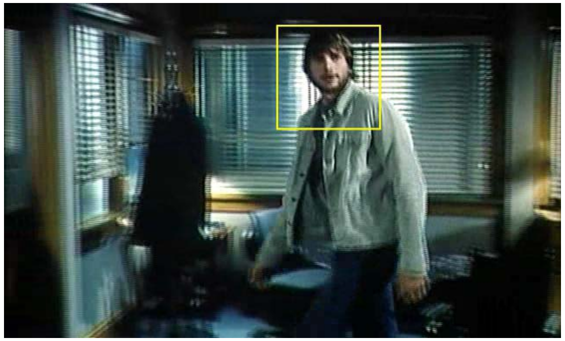


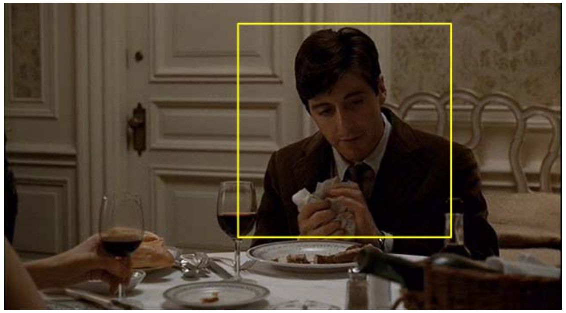
# Summary

- Multiple features and multiple kernels boost performance
- Discriminative learning of model with latent variables for single feature (HOG):
  - Latent variables can learn best alignment in the ROI training annotation
  - Parts can be thought of as local SIFT vectors
  - Some similarities to Implicit Shape Model/Constellation models but with discriminative/careful training throughout









NB: Code available for latent model !

# Outline

1. Sliding window detectors

2. Features and adding spatial information

3. HOG + linear SVM classifier

4. Two state of the art algorithms and PASCAL VOC

5. The future and challenges

# Current Research Challenges

- Context
  - from scene properties: GIST, BoW, stuff
  - from other objects
  - from geometry of scene, e.g. Hoiem et al CVPR 06
- Occlusion/truncation
  - Winn & Shotton, Layout Consistent Random Field, CVPR 06
  - Vedaldi & Zisserman, NIPS 09
  - Yang et al, Layered Object Detection, CVPR 10
- 3D
- Scaling up – thousands of classes
  - Torralba et al, Feature sharing
  - ImageNet
- Weak and noisy supervision