CAP 4453 Robot Vision

Dr. Gonzalo Vaca-Castaño

gonzalo.vacacastano@ucf.edu

Administrative details

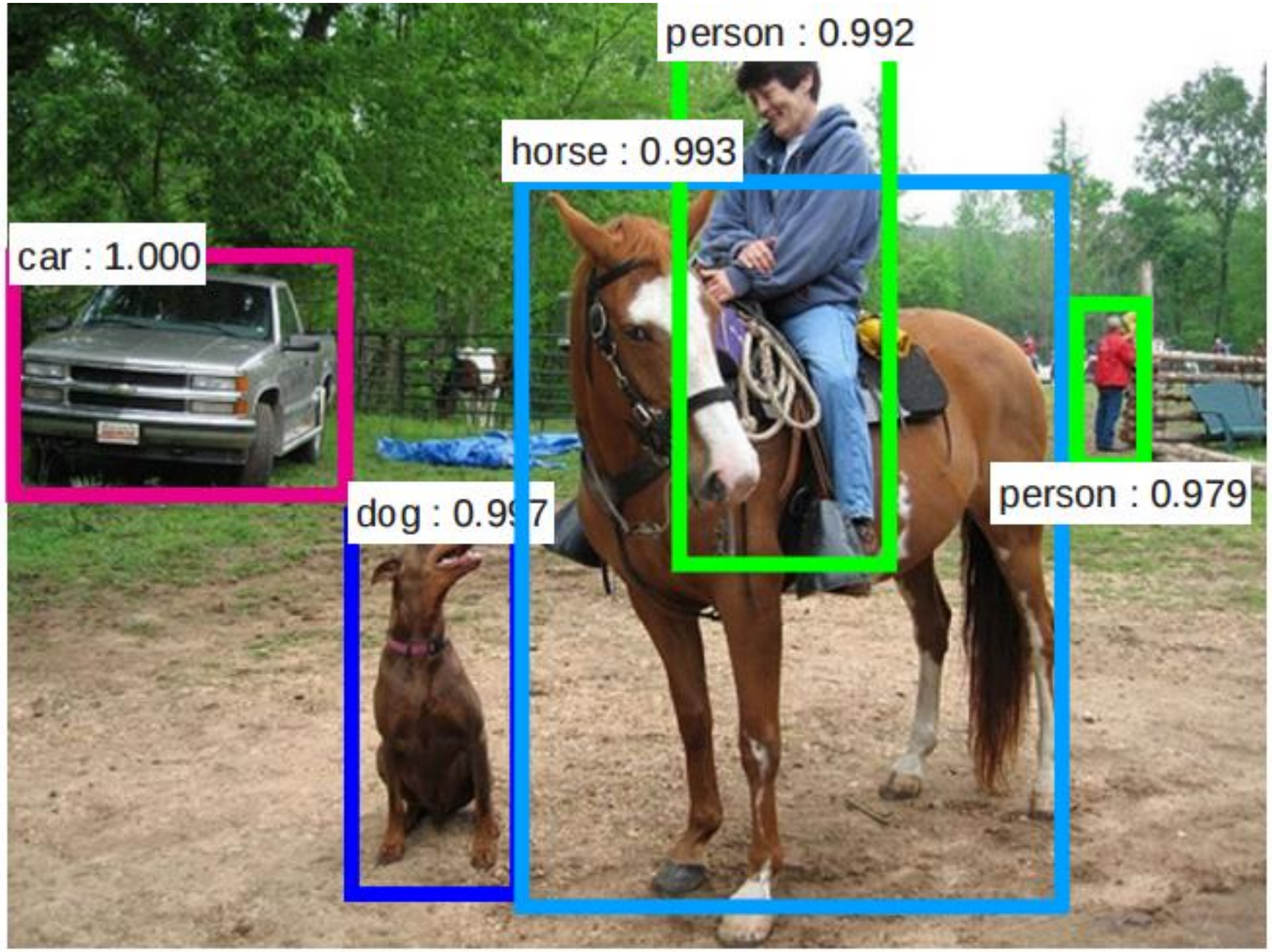
- Issues submitting homework

Credits

- Some slides comes directly from:
  - Ross B. Girshick
  - Pedro F. Felzenszwalb

Short Review from last class

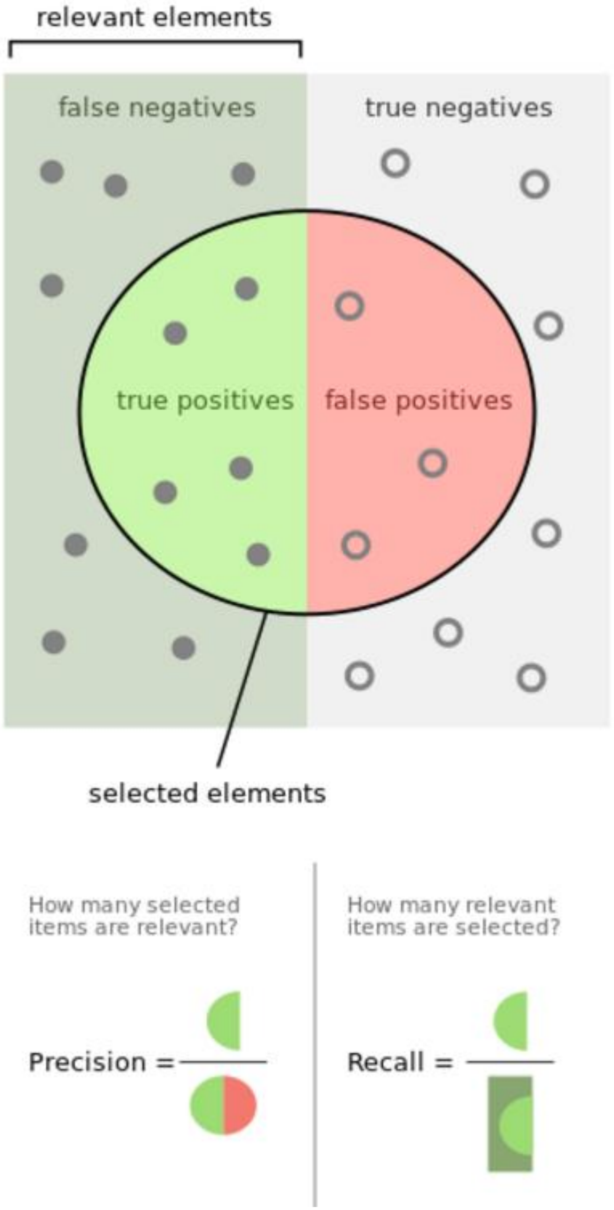
Object detection



- Multiple outputs
  - Bounding box
  - Label
  - Score

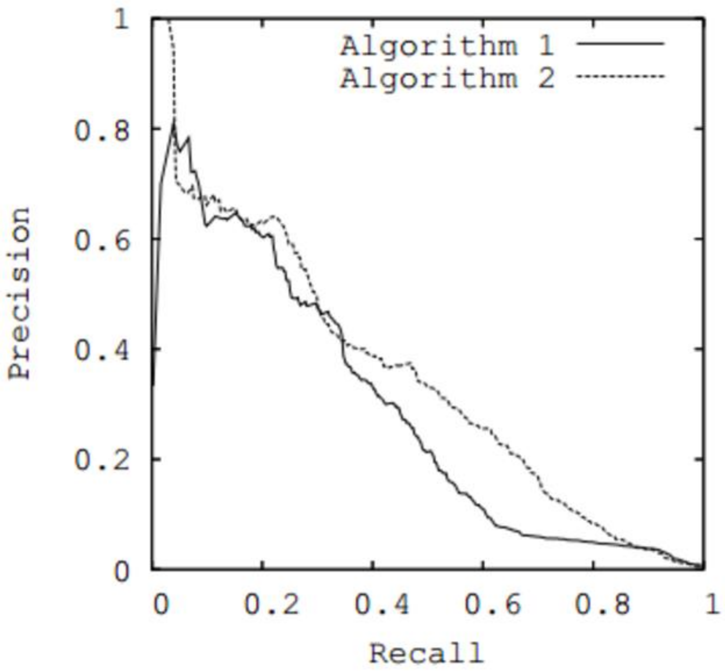
Terms

Recall
Precision
mAP
IoU



Possible detection

- Bounding box
- Label
- ***score***



Average precision (AP): Area under curve

Histograms of Oriented Gradients for Human Detection

Navneet Dalal and Bill Triggs
INRIA Rhône-Alps, 655 avenue de l'Europe, Montbonnot 38334, France
{Navneet.Dalal,Bill.Triggs}@inrialpes.fr, http://lear.inrialpes.fr

Abstract

We study the question of feature sets for robust visual object recognition, adopting linear SVM based human detection as a test case. After reviewing existing edge and gradient based descriptors, we show experimentally that grids of Histograms of Oriented Gradient (HOG) descriptors significantly outperform existing feature sets for human detection. We study the influence of each stage of the computation on performance, concluding that fine-scale gradients, fine orientation binning, relatively coarse spatial binning, and high-quality local contrast normalization in overlapping descriptor blocks are all important for good results. The new approach gives near-perfect separation on the original MIT pedestrian database, so we introduce a more challenging dataset containing over 1800 annotated human images with a large range of pose variations and backgrounds.

1 Introduction

We briefly discuss previous work on human detection in §2, give an overview of our method §3, describe our data sets in §4 and give a detailed description and experimental evaluation of each stage of the process in §5–6. The main conclusions are summarized in §7.

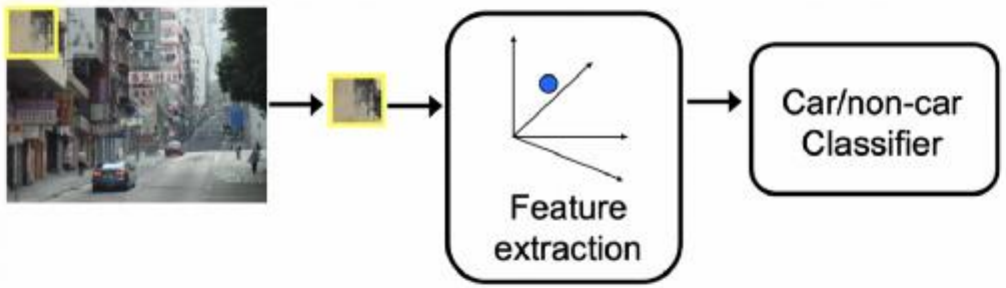
2 Previous Work

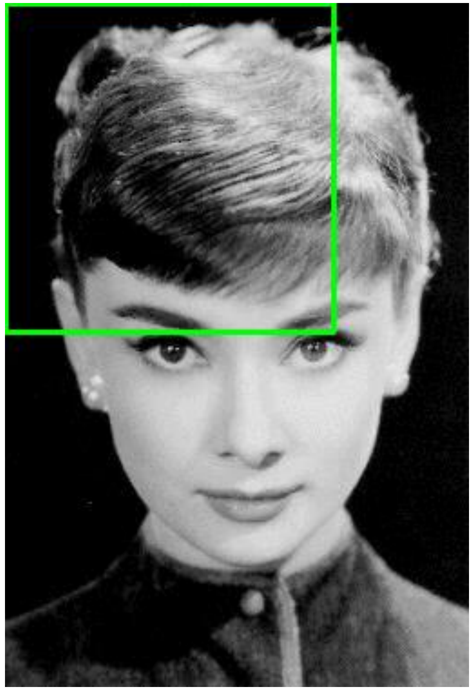
There is an extensive literature on object detection, but here we mention just a few relevant papers on human detection [18,17,22,16,20]. See [6] for a survey. Papageorgiou *et al* [18] describe a pedestrian detector based on a polynomial SVM using rectified Haar wavelets as input descriptors, with a parts (subwindow) based variant in [17]. Depoortere *et al* give an optimized version of this [2]. Gavrila & Philomen [8] take a more direct approach, extracting edge images and matching them to a set of learned exemplars using chamfer distance. This has been used in a practical real-time pedestrian detection system [7]. Viola *et al* [22] build an efficient

- CVPR 2005

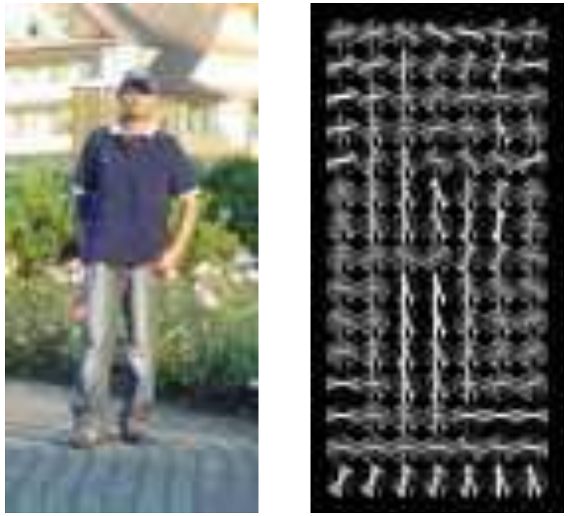
Sliding Window Technique

- Score every subwindow
  - extract features from the image window
  - classifier decides based on the given features.
- It is a brute-force approach





Person detection with HoG's & linear SVM's (so far)

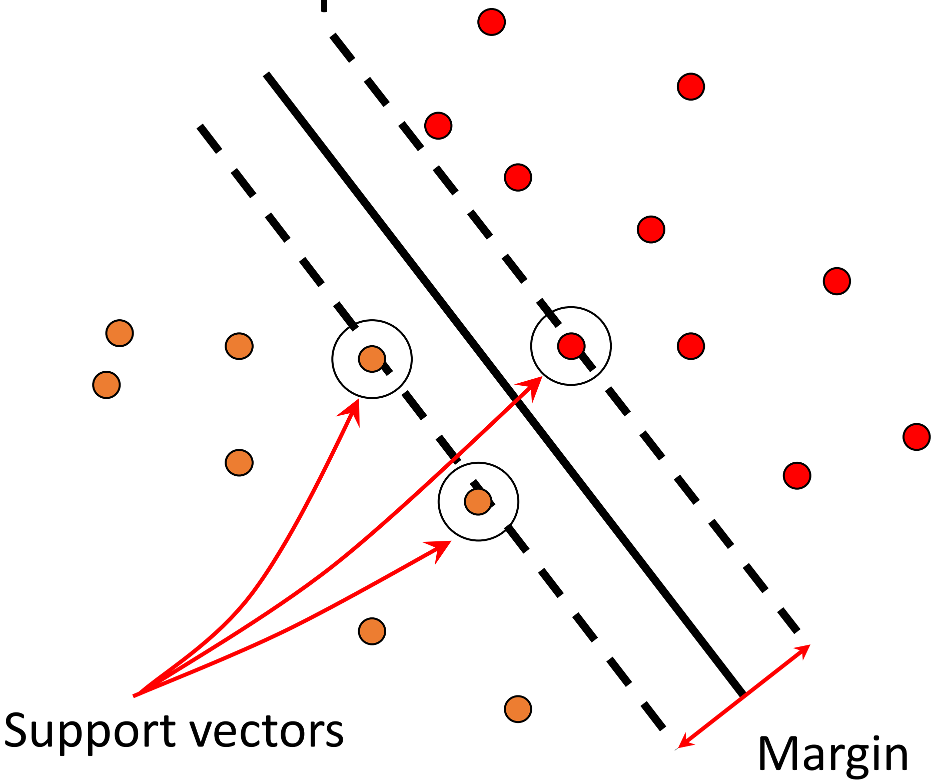


- Histogram of oriented gradients (HoG): Map each grid cell in the input window to a histogram counting the gradients per orientation.
- Train a linear SVM using training set of pedestrian vs. non-pedestrian windows.

Dalal & Triggs, CVPR 2005

Support vector machines

- Find hyperplane that maximizes the *margin* between the positive and negative examples



$$\mathbf{x} \text{ positive } (y=1): \quad \mathbf{x}\cdot\mathbf{w}+b \geq 1$$

$$\mathbf{x} \text{ negative } (y=-1): \quad \mathbf{x}\cdot\mathbf{w}+b \leq -1$$

For support vectors, $\mathbf{x}\cdot\mathbf{w}+b = \pm 1$

Distance between point and hyperplane: $\dfrac{|\mathbf{x}\cdot\mathbf{w}+b|}{\|\mathbf{w}\|}$

Therefore, the margin is $2 / \|\mathbf{w}\|$

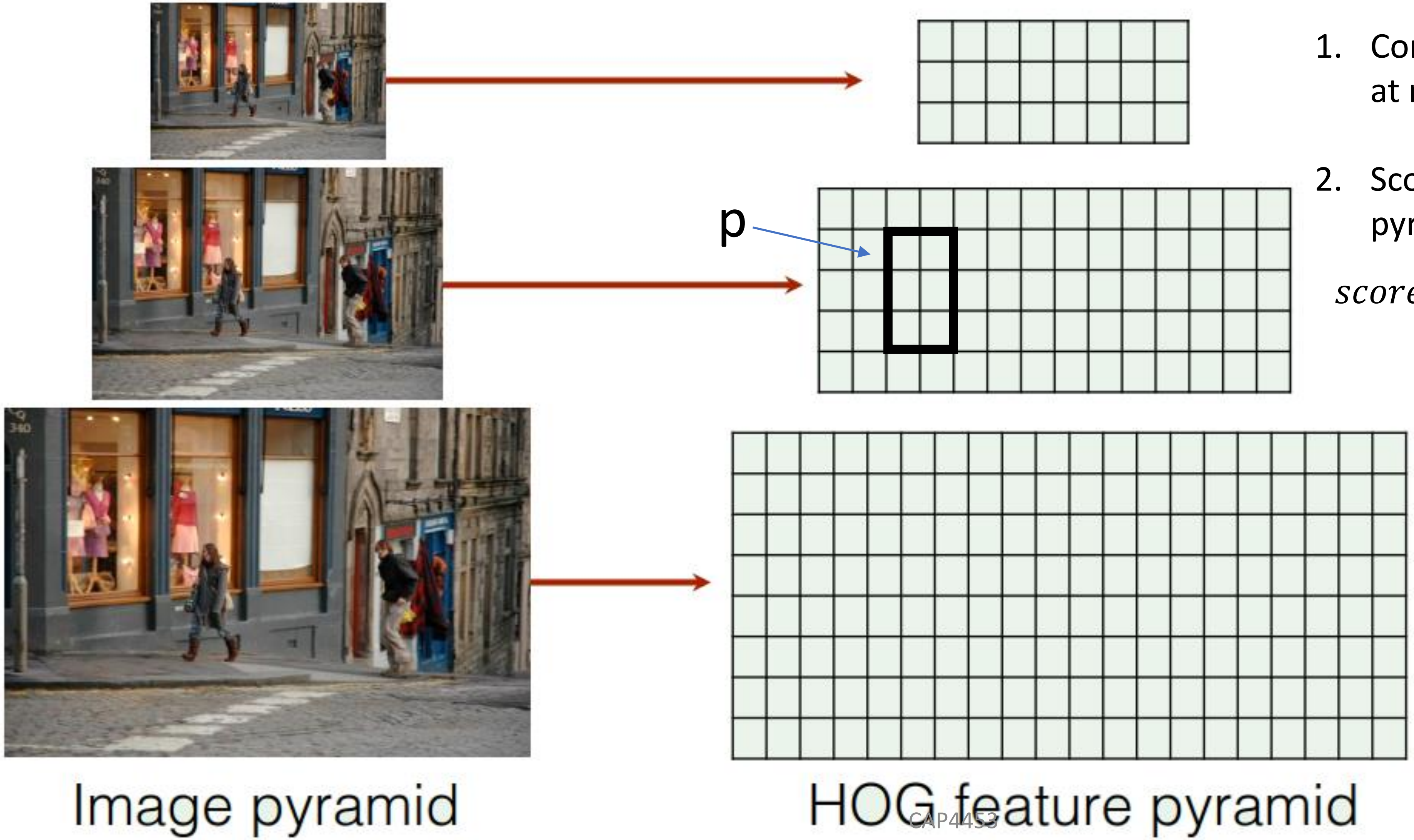
C. Burges, A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, 1998

SVMs: Pros and cons

- Pros
  - Kernel-based framework is very powerful, flexible
  - Training is convex optimization, globally optimal solution can be found
  - Amenable to theoretical analysis
  - SVMs work very well in practice, even with very small training sample sizes

- Cons
  - No “direct” multi-class SVM, must combine two-class SVMs (e.g., with one-vs-others)
  - Computation, memory (esp. for nonlinear SVMs)

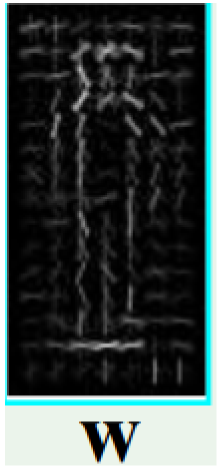
The Dalal & Triggs detector



1. Compute HOG of the whole image at multiple resolutions!
2. Score every window of the feature pyramid

$$score(I,p) = w \cdot \phi(I,p)$$

FROM TRAINING



3. Apply non-maximal suppression (NMS)

Robot Vision

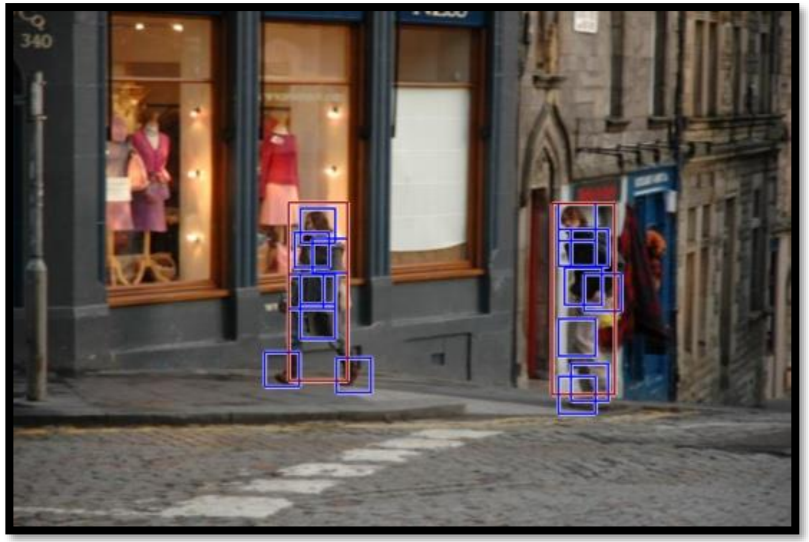
14. Object detection II

Outline

- Overview: What is Object detection?
- Top methods for object detection
- Object detection with Sliding Window and Feature Extraction(HoG)
    - Sliding Window technique
    - HOG: Gradient based Features
    - Machine Learning
        - Support Vector Machine (SVM)
    - Non-Maximum Suppression (NMS)
- Implementation examples
- **Deformable Part-based Model (DPM)**

Motivation

- Problem: Detecting and localizing generic objects from categories (e.g. people, cars, etc.) in static images.



- Issues to overcome:
  - Changes in illumination or viewpoint
  - Non-rigid deformations, e.g. pose
  - Intraclass variability, e.g. types of cars

Previous Works

Dalal & Triggs '05

- Histogram of Oriented Gradients (HOG)
- Support Vector Machines (SVM) Training
- Sliding window detection

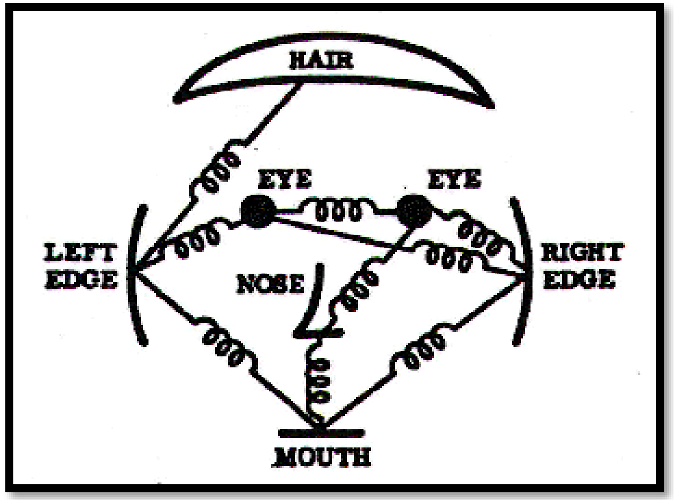
Original Image

Histogram of Oriented Gradients

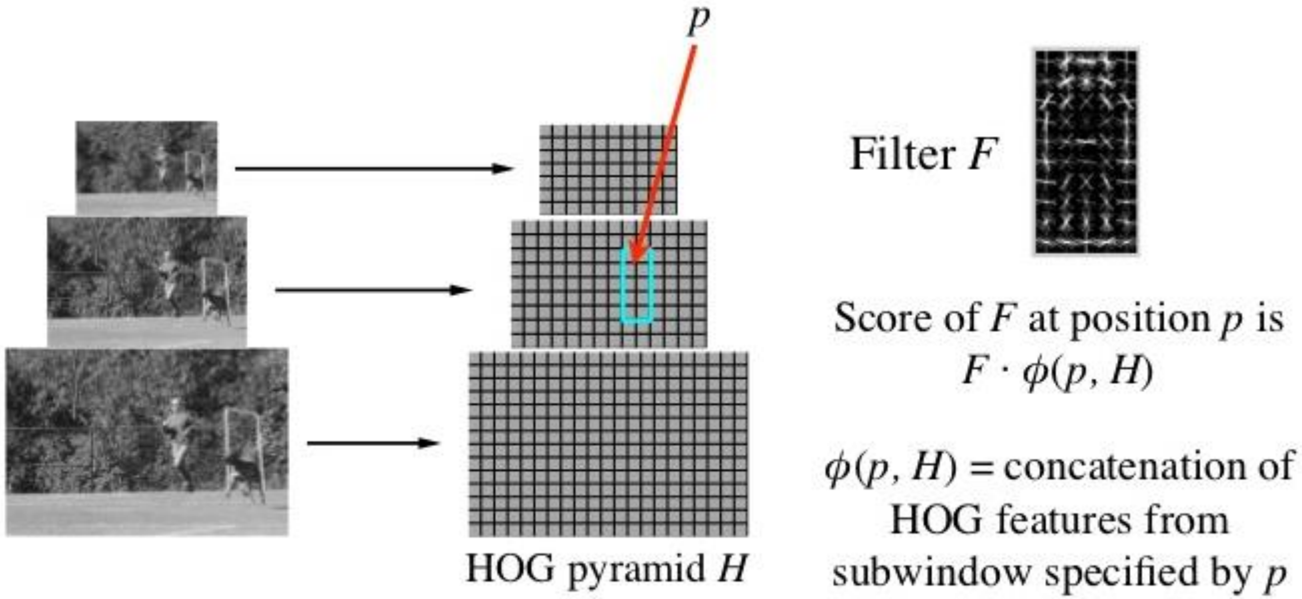
Fischler & Elschlager '73
Felzenszwalb & Huttenlocher '00

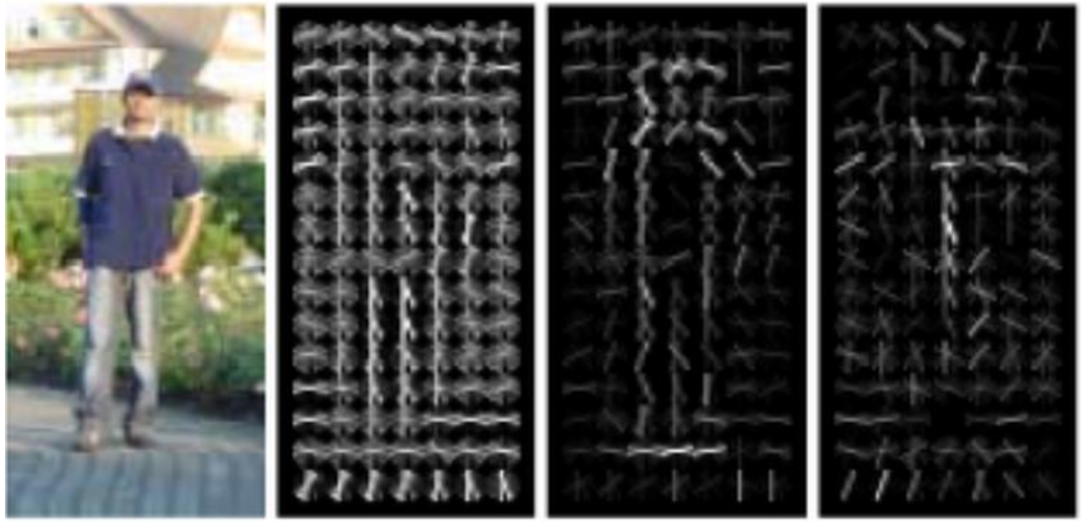
- Pictorial structures
- Weak appearance models
- Non-Discriminative training

Pictorial Structures Model of a Face



Object Detection with Histogram of Oriented gradients





Original Image Extracted Gradient Positive Weights Negative Weights

Combine HOG and Linear SVM

Detects objects using weighted HOG filters

Inspect both positive and negative weighted results

Human or not?

Object Detection with Discriminatively Trained Part Based Models

Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester and Deva Ramanan

Abstract—We describe an object detection system based on mixtures of multiscale deformable part models. Our system is able to represent highly variable object classes and achieves state-of-the-art results in the PASCAL object detection challenges. While deformable part models have become quite popular, their value had not been demonstrated on difficult benchmarks such as the PASCAL datasets. Our system relies on new methods for discriminative training with partially labeled data. We combine a margin-sensitive approach for data-mining hard negative examples with a formalism we call *latent SVM*. A latent SVM is a reformulation of MI-SVM in terms of latent variables. A latent SVM is semi-convex and the training problem becomes convex once latent information is specified for the positive examples. This leads to an iterative training algorithm that alternates between fixing latent values for positive examples and optimizing the latent SVM objective function.

Index Terms—Object Recognition, Deformable Models, Pictorial Structures, Discriminative Training, Latent SVM

1 Introduction

Object recognition is one of the fundamental challenges in computer vision. In this paper we consider the problem of detecting and localizing generic objects from categories such as people or cars in static images. This is a difficult problem because objects in such categories can vary greatly in appearance. Variations arise not only from changes in illumination and viewpoint, but also due to non-rigid deformations, and intraclass variability in shape and other visual properties. For example, people wear different clothes and take a variety of poses while cars come in a various shapes and colors.

We describe an object detection system that represents highly variable objects using mixtures of multiscale deformable part models. These models are trained using a discriminative procedure that only requires bounding boxes for the objects in a set of images. The resulting system is both efficient and accurate, achieving state-of-the-art results on the PASCAL VOC benchmarks [11]–[13] and the INRIA Person dataset [10].

it has been difficult to establish their value in practice. On difficult datasets deformable part models are often outperformed by simpler models such as rigid templates [10] or bag-of-features [44]. One of the goals of our work is to address this performance gap.

While deformable models can capture significant variations in appearance, a single deformable model is often not expressive enough to represent a rich object category. Consider the problem of modeling the appearance of bicycles in photographs. People build bicycles of different types (e.g., mountain bikes, tandems, and 19th-century cycles with one big wheel and a small one) and view them in various poses (e.g., frontal versus side views). The system described here uses mixture models to deal with these more significant variations.

We are ultimately interested in modeling objects using "visual grammars". Grammar based models (e.g. [16], [24], [45]) generalize deformable part models by representing objects using variable hierarchical structures. Each part in a grammar based model can be defined directly or in terms of other parts. Moreover, grammar

CVPR 2008

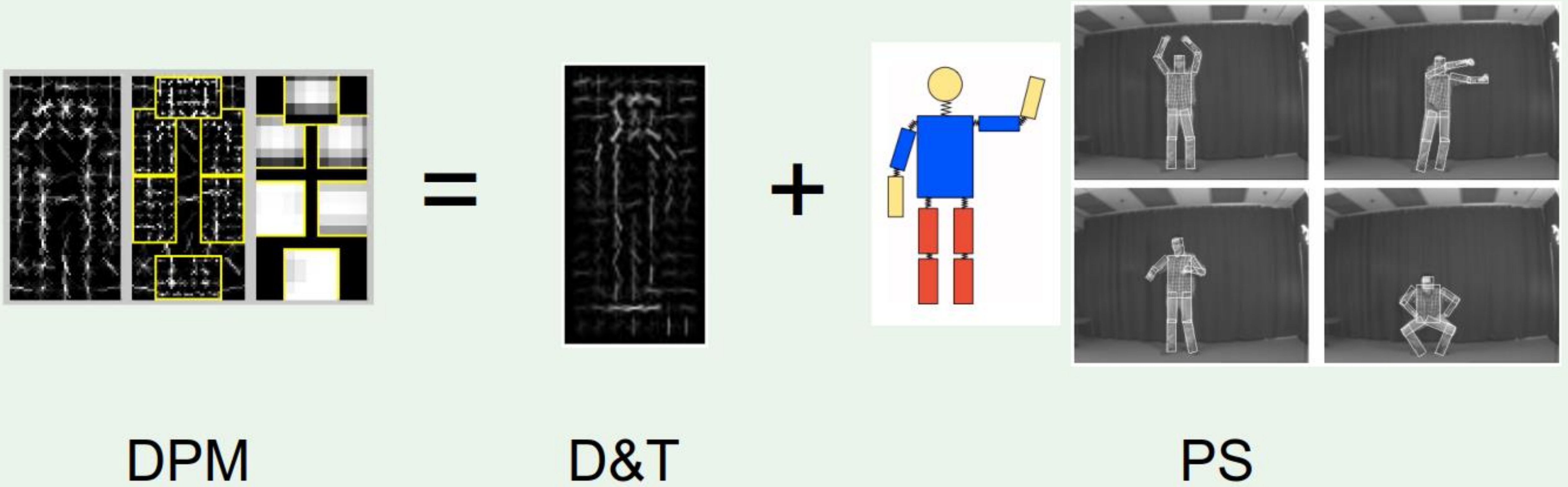
Tpami 2010

Successful detection method

- Joint winner in 2009 Pascal VOC challenge with the Oxford Method.
- Award of "lifetime achievement“ in 2010.
- Mixture of deformable part models
- Each component has global template + deformable parts
  - HOG feature templates
- Fully trained from bounding boxes alone

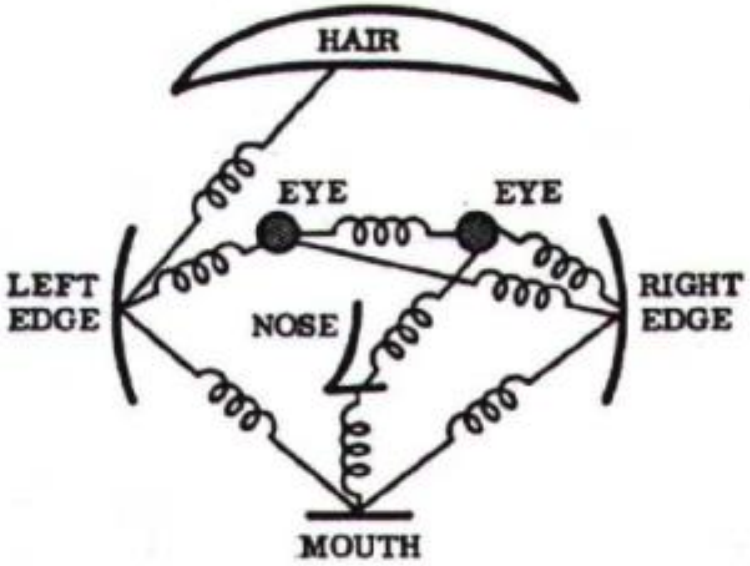
Key idea

- Port the success of Dalal & Triggs into a part-based model



Part-based models

- **Origins in Fischler & Elschlager 1973**
- **Model has two components**
  - **parts (2D image fragments)**
  - **structure (configuration of parts)**

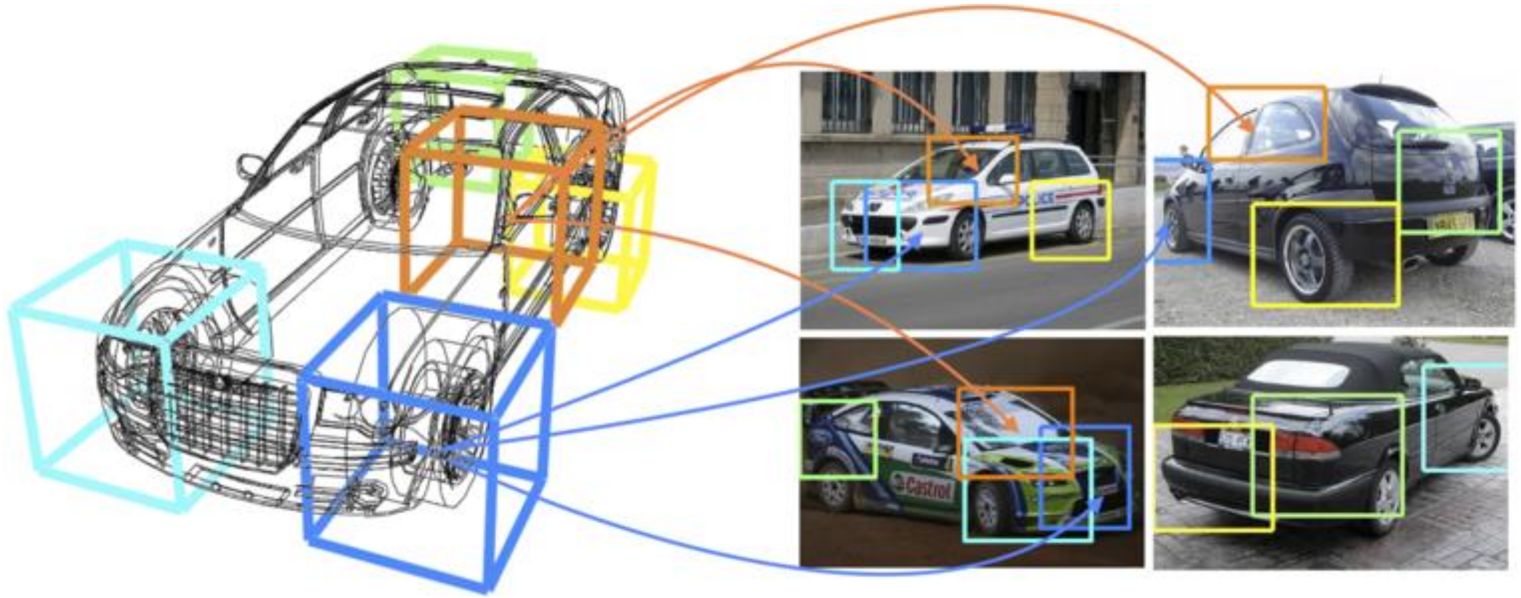


MODELS

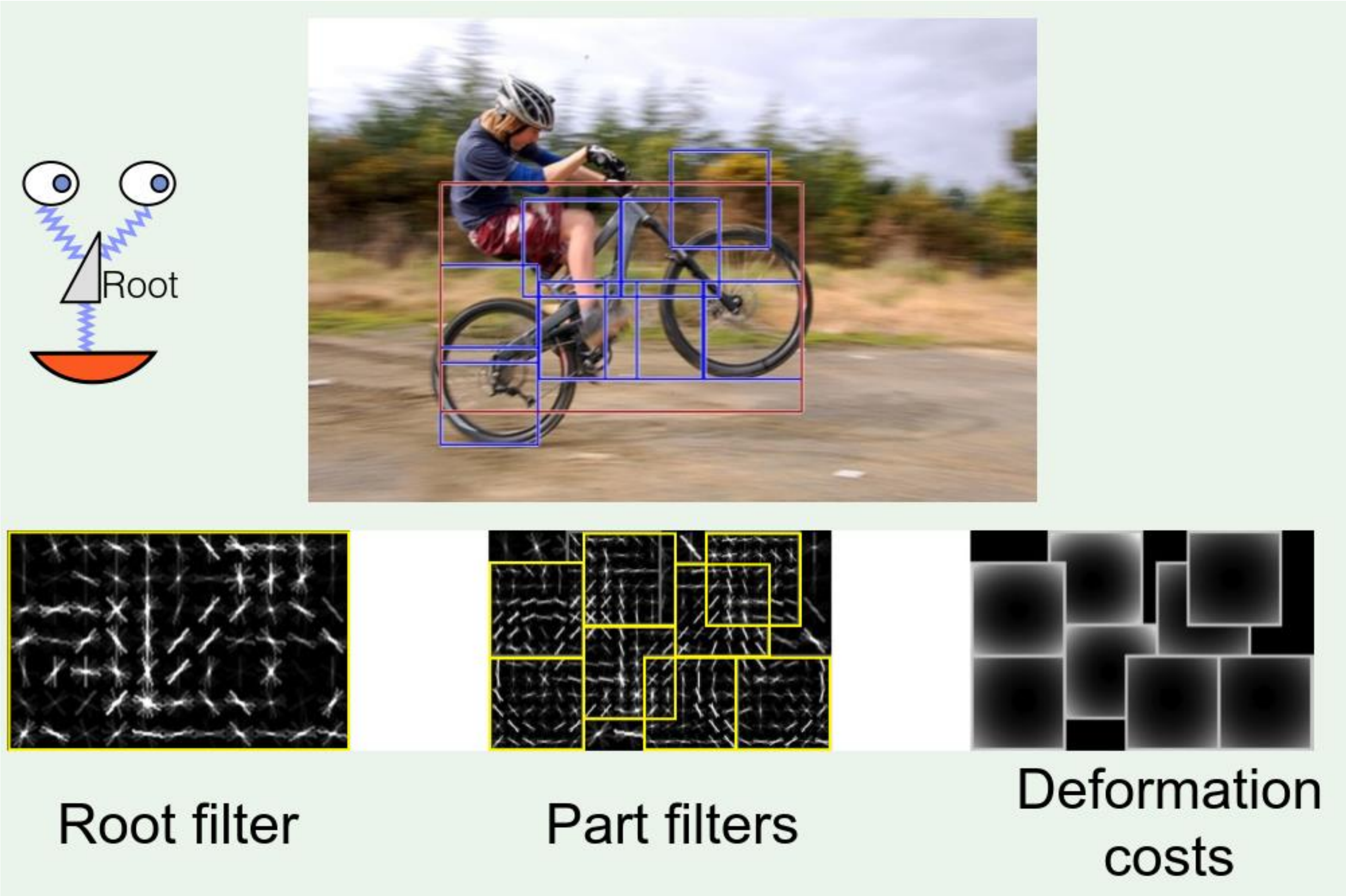
- Deformable Part Models (DPM)
- Matching
- Mixture Models

Deformable Part Models (DPM)

- Represent object by several parts
- Model is deformable, i.e. parts can move independently of each other
- Parts are “punished” for being far away from their origin



DPM Idea



The Dalal & Triggs detector

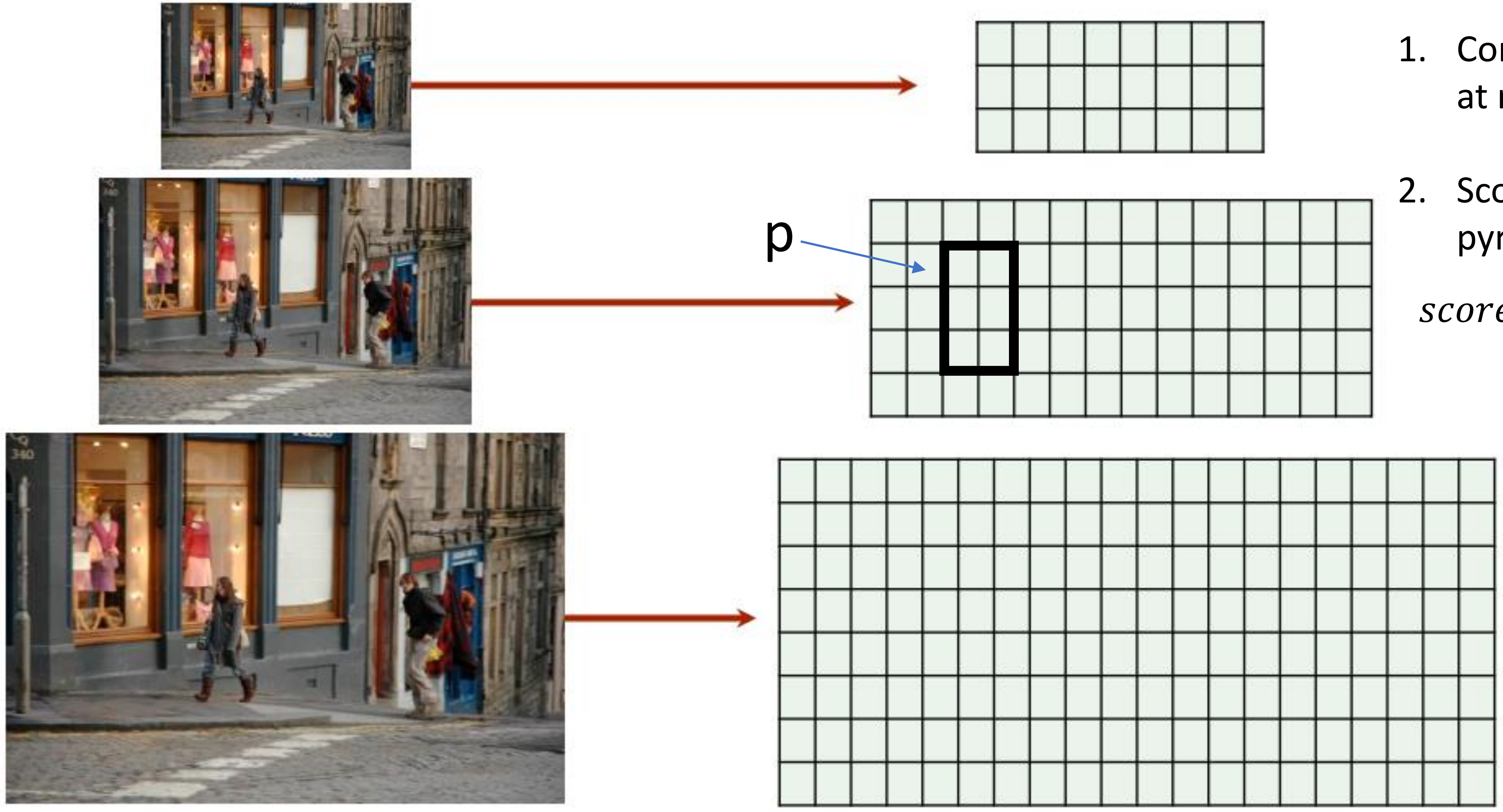


Image pyramid

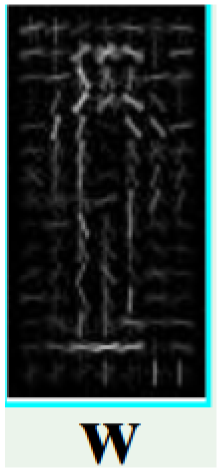
HOG feature pyramid

1. Compute HOG of the whole image at multiple resolutions

2. Score every window of the feature pyramid

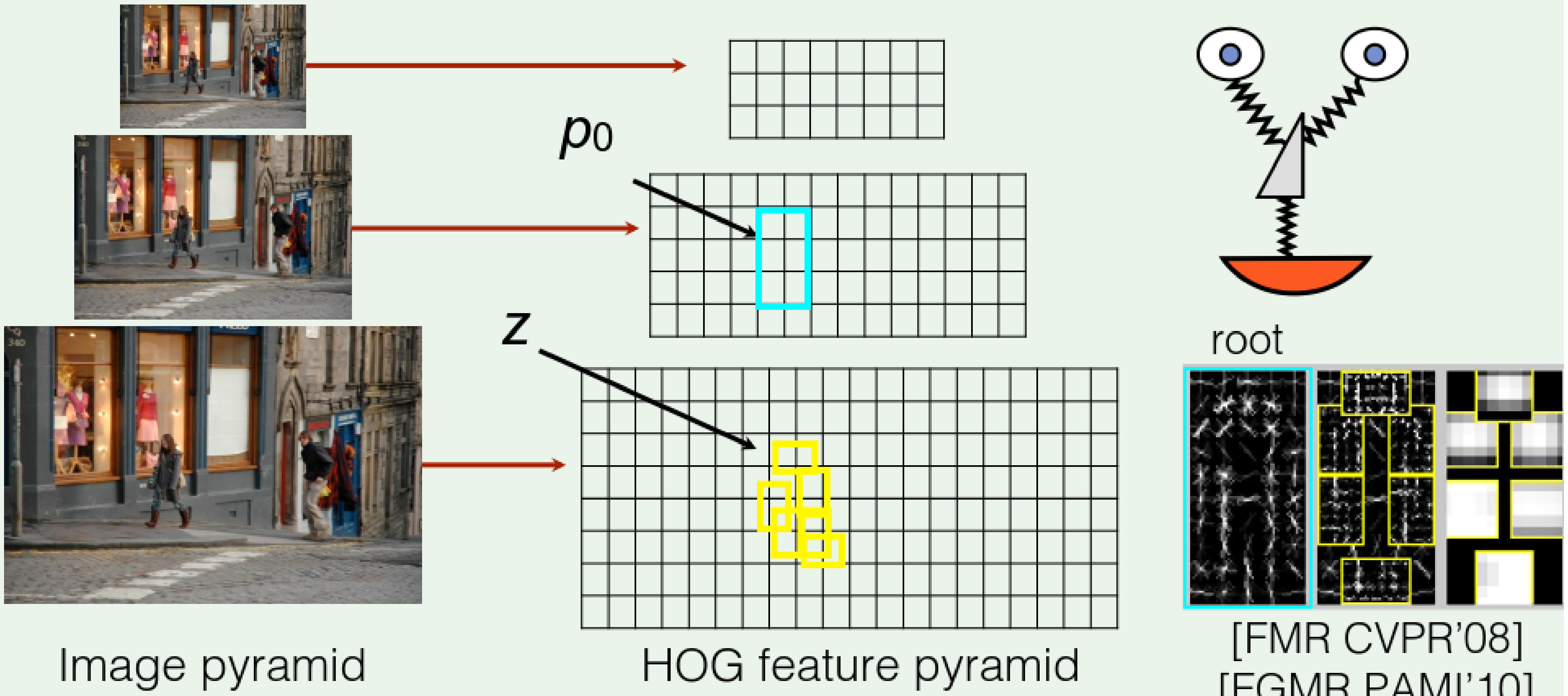
$$score(I,p) = w \cdot \phi(I,p)$$

FROM TRAINING



3. Apply non-maximal suppression (NMS)

DPM = D&T + parts



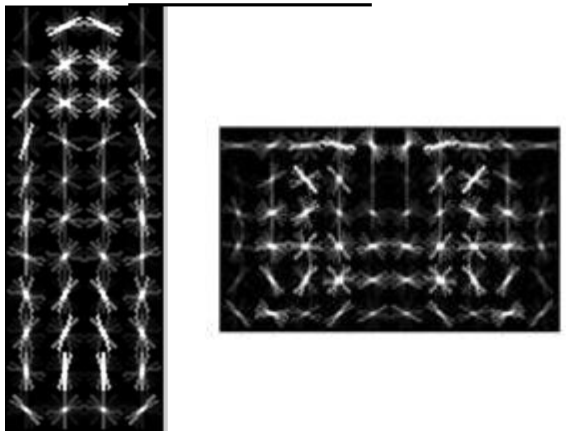
[FMR CVPR'08]
[FGMR PAMI'10]

- Add parts to the Dalal & Triggs detector
  - HOG features
  - Linear filters / sliding-window detector
  - Discriminative training

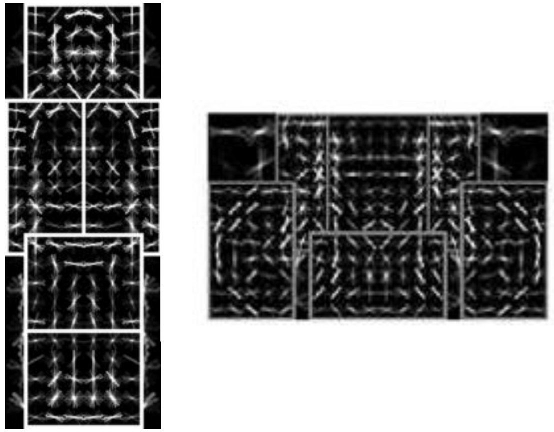
Deformable Part Models (DPM)

- Model has a root filter F_O and n part models represented by (F_i, v_i, d_i)
  - F_i is the i-th part filter
  - v_i is the is the origin of the i-th part relative to the root
  - d_i is the deformation parameter

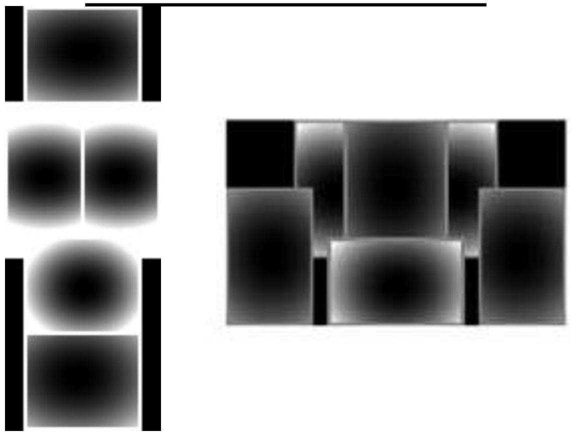
Coarse Filter



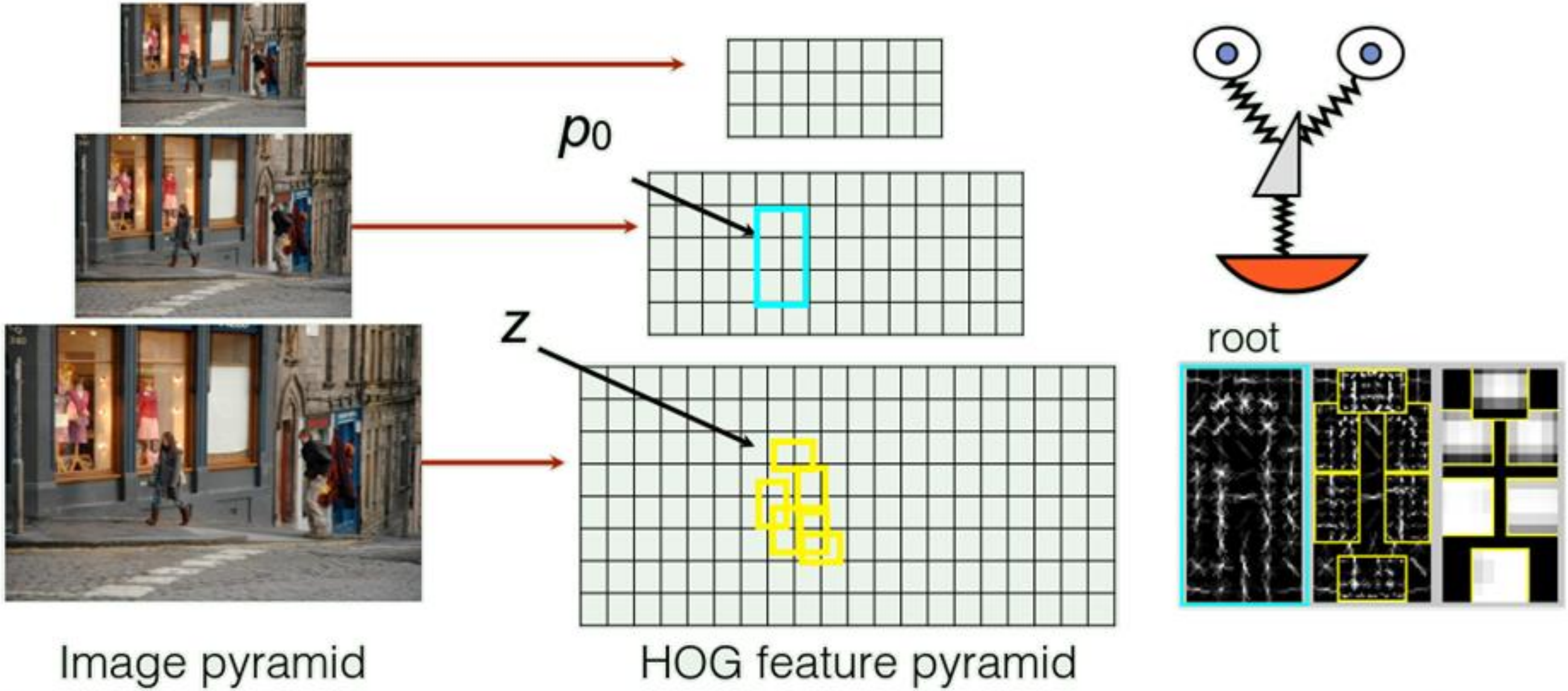
High-res Part Filter



Deformation models



Sliding window detection with DPM



$$z = (p_1, \ldots, p_n)$$

$$\mathrm{score}(I, p_0) = \max_{p_1,\ldots,p_n} \underbrace{\sum_{i=0}^{n} m_i(I, p_i)}_{\text{Filter scores}} - \underbrace{\sum_{i=1}^{n} d_i(p_0, p_i)}_{\text{Spring costs}}$$

Deformable Part Models (DPM)

$$score(p_o, \dots, p_n) = \sum_{i=0}^{n} F'_i \cdot \phi(H, p_i) - \sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i) + b$$

Bias

Filters

Feature of subwindow at location p_i

Deformation Parameters

Displacement of part i

- Score of hypothesis z… $$score(z) = \beta \cdot \psi(H, z)$$
- Unknown… $$\beta = (F_0, \dots, F_n, d_1, \dots, d_n, b)$$
- Known… $$\psi(H, z) = (\phi(H, p_0), \dots, \phi(H, p_n), -\phi(dx_1, dy_1), \dots, -\phi(dx_n, dy_n), 1)$$

Deformable Part Models (DPM)

$$score(p_o, \dots, p_n) = \underbrace{\sum_{i=0}^{n} F'_i \cdot \phi(H, p_i)}_{\text{Data term}} - \underbrace{\sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i)}_{\text{Spatial info}} + b$$

Data term

Spatial info

Bias

Filters

Feature of subwindow at location p_i

Deformation Parameters

Displacement of part i

- Initial condition: $d_i = (0,0,1,1)$
- Displacement Function: $\phi_d(dx, dy) = (dx, dy, dx^2, dy^2)$

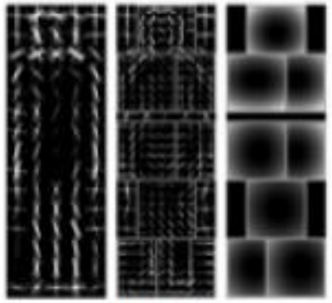
Matching

- The overall score of a root location is computed according to the best possible placement of parts
  - High scoring root locations define detections
  - High scoring part roots define object hypothesis

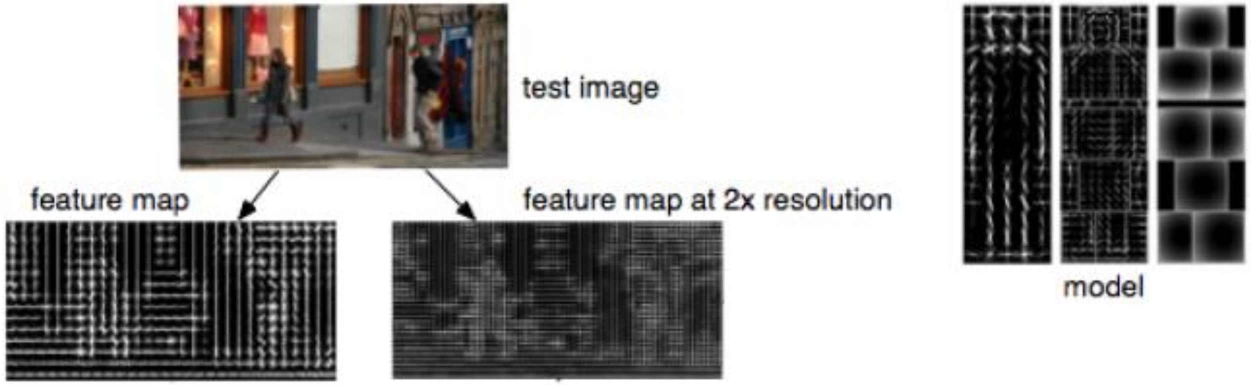
$$score(p_0) = \max_{p_1,\ldots,p_n} score(p_0, \ldots, p_n)$$

DPM detection


test image


model

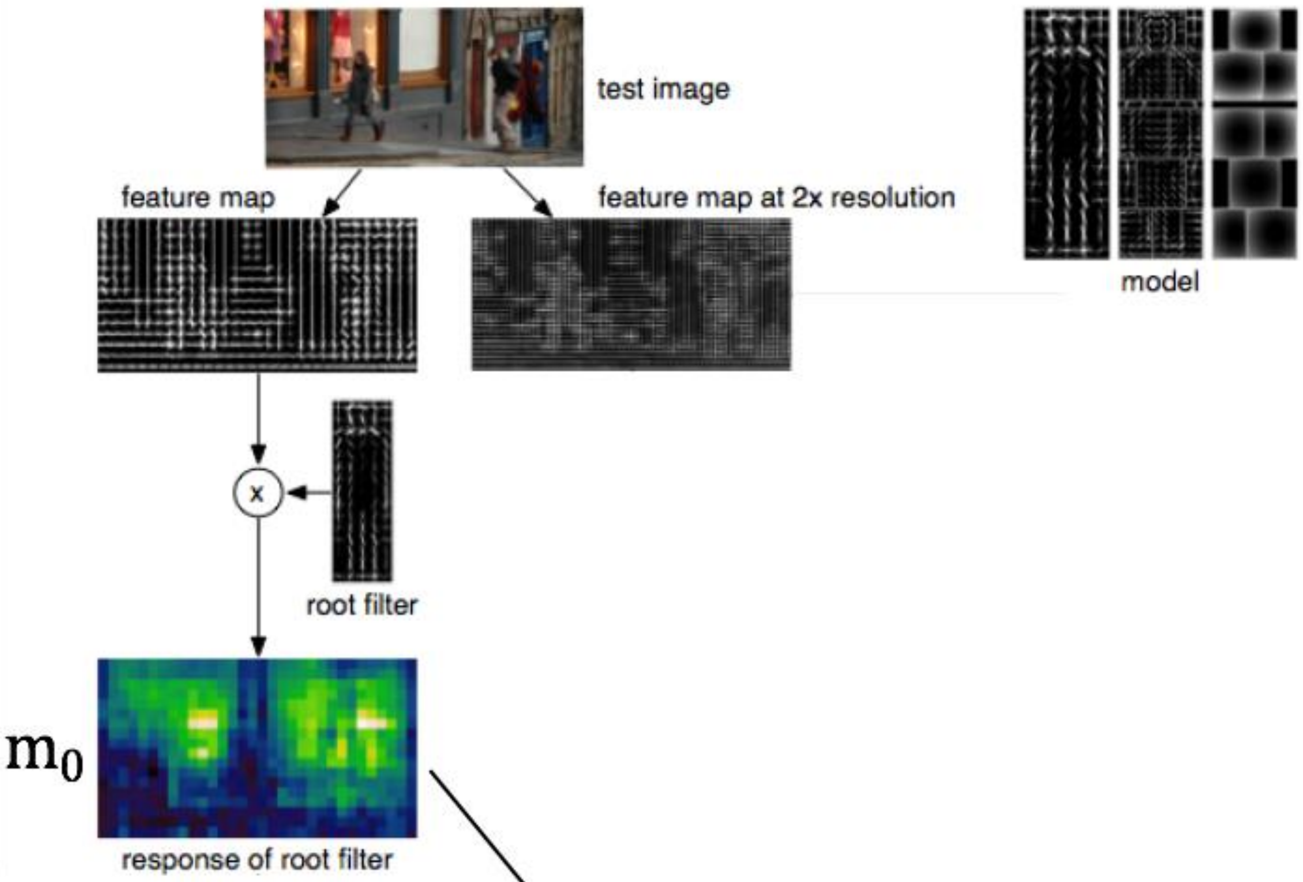
DPM detection



Root scale   Part scale

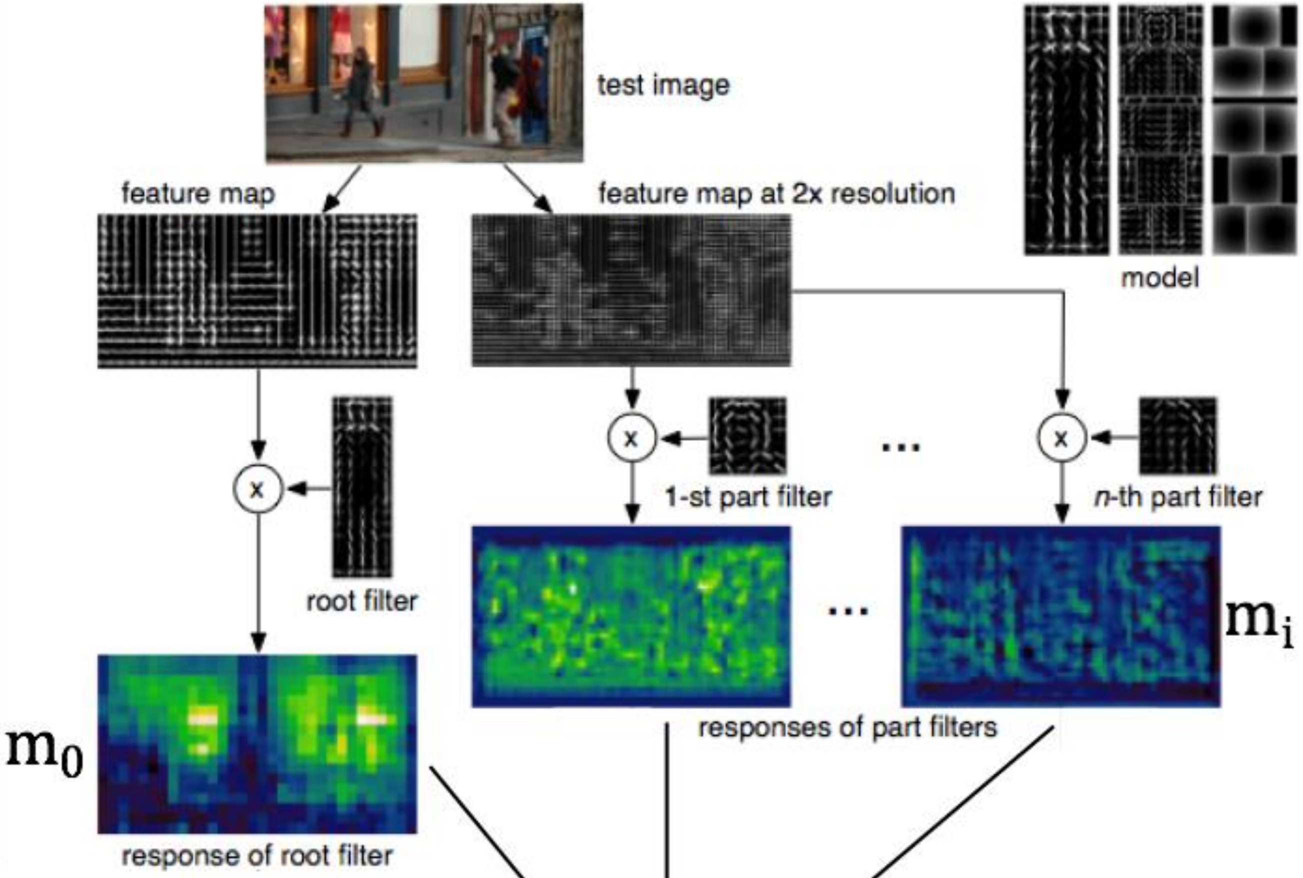
repeat for each level in pyramid

DPM detection



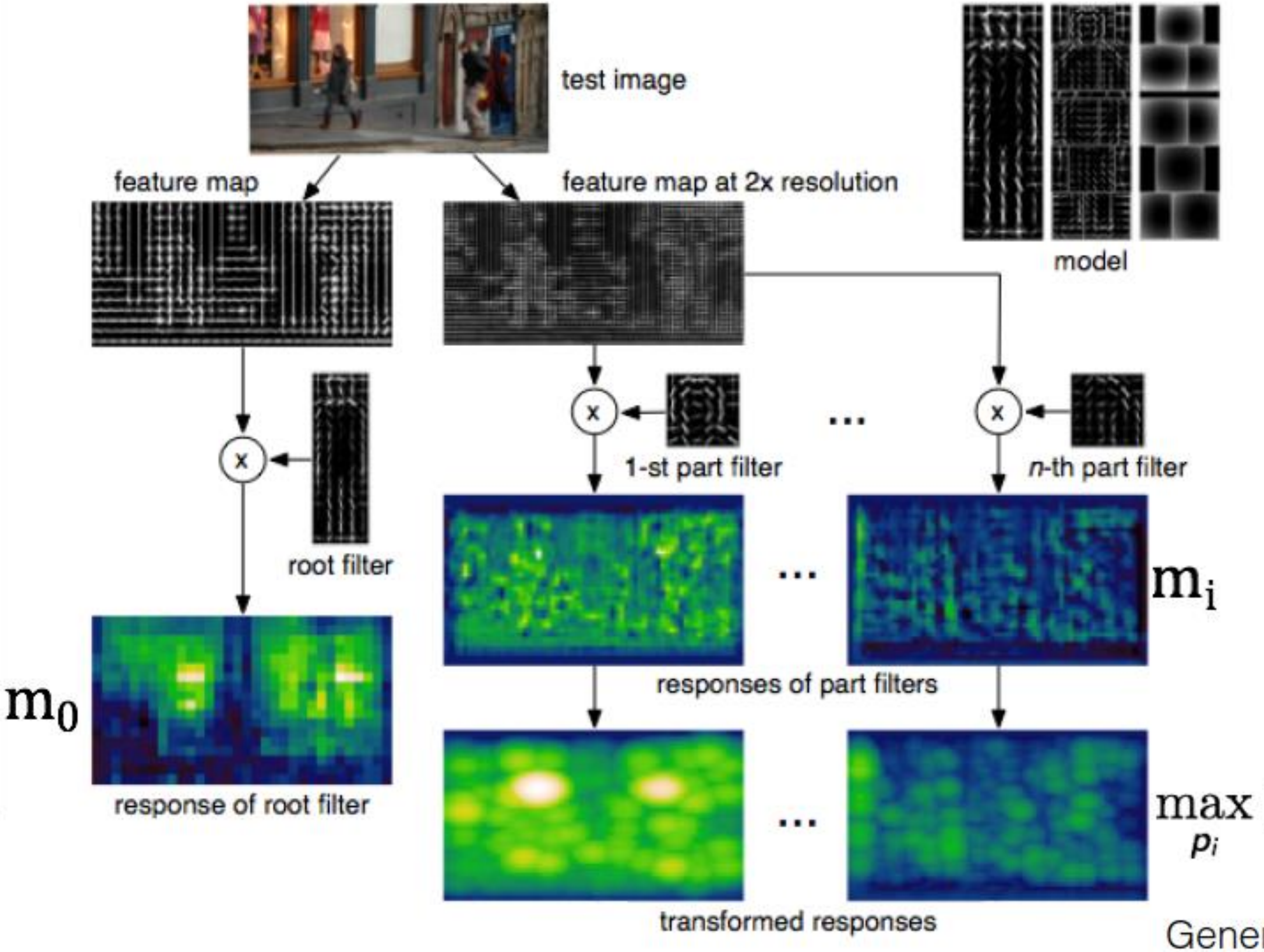
$$\text{score}(I, p_0) = \max_{p_1,\ldots,p_n} \sum_{i=0}^{n} m_i(I, p_i) - \sum_{i=1}^{n} d_i(p_0, p_i)$$

DPM detection



$$\text{score}(I, p_0) = \max_{p_1,\ldots,p_n} \sum_{i=0}^{n} m_i(I, p_i) - \sum_{i=1}^{n} d_i(p_0, p_i)$$

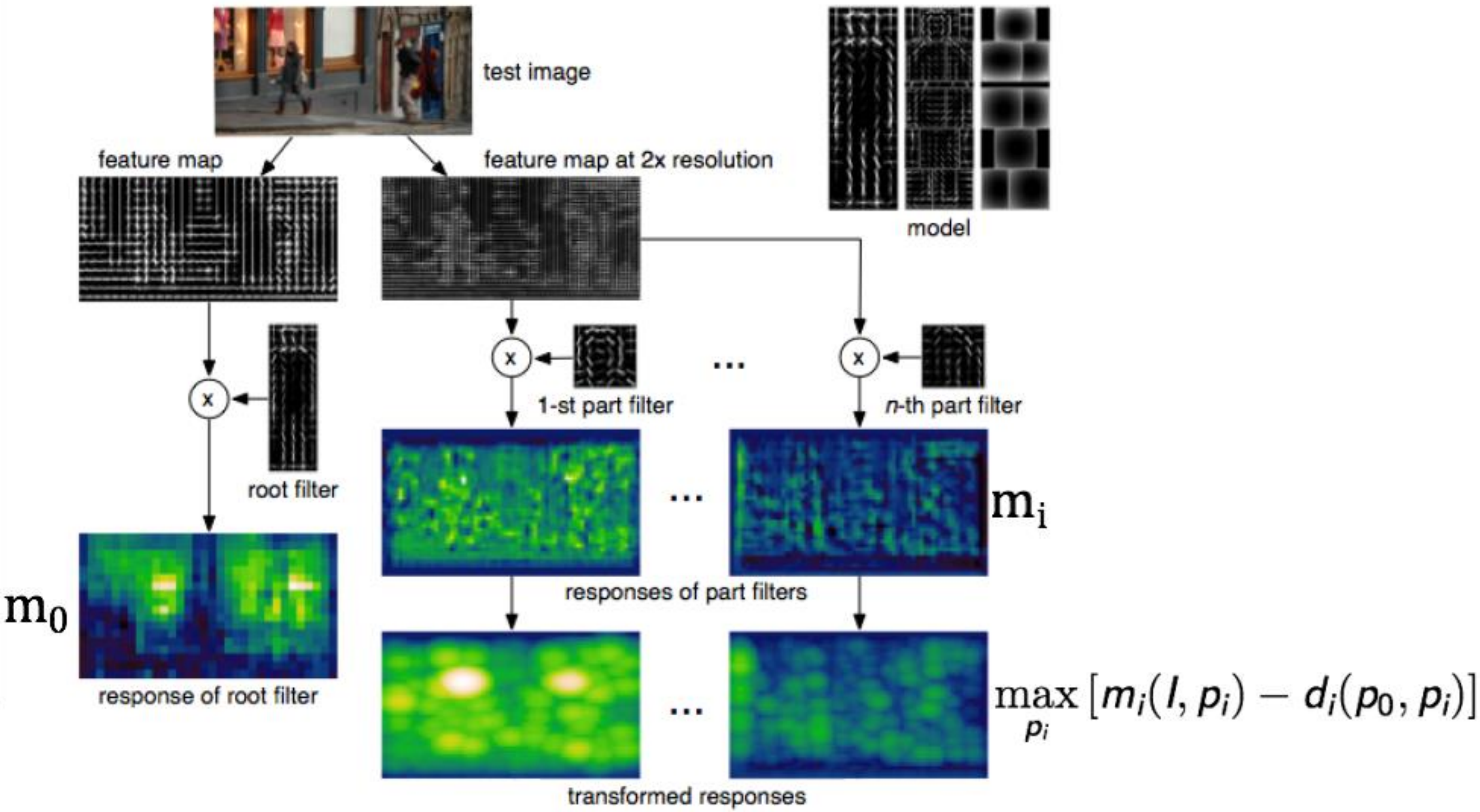
DPM detection



$$\max_{p_i} \left[m_i(I, p_i) - d_i(p_0, p_i) \right]$$

Generalized distance transform
Felzenszwalb & Huttenlocher '00

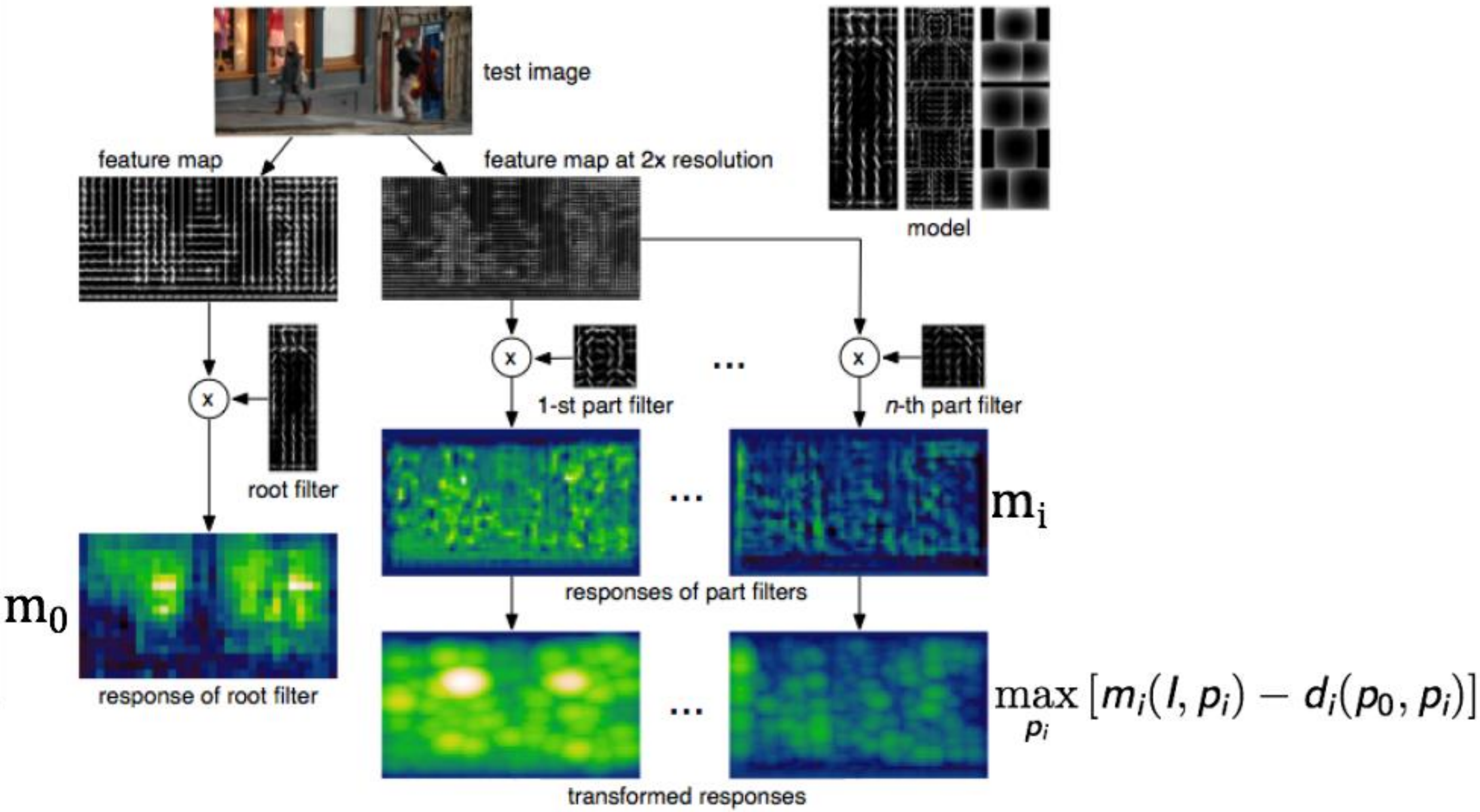
DPM detection



$$\text{score}(I, p_0) = \max_{p_1,\ldots,p_n} \sum_{i=0}^{n} m_i(I, p_i) - \sum_{i=1}^{n} d_i(p_0, p_i)$$

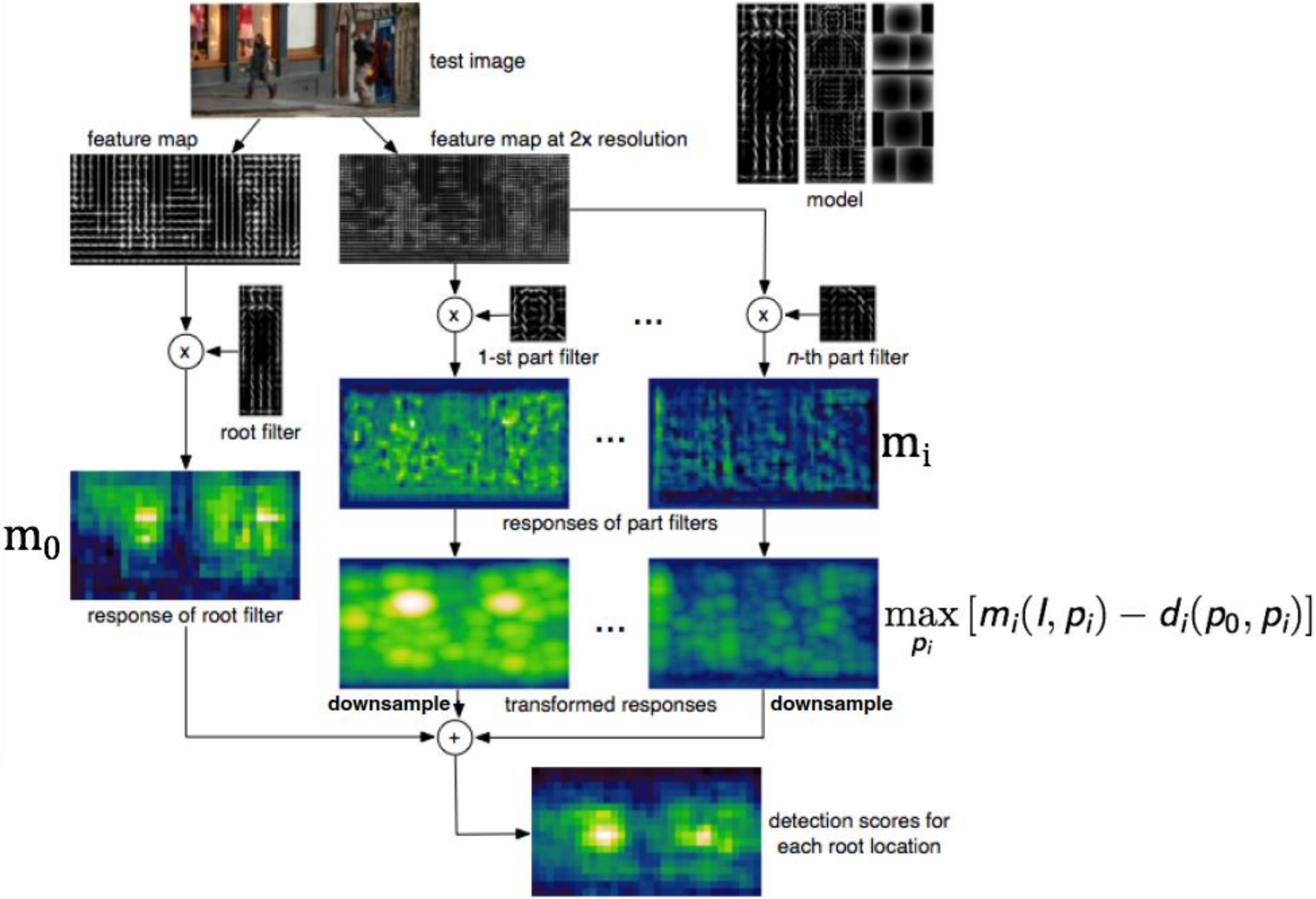
$$= m_0(I, p_0) + \sum_{i=1}^{n} \max_{p_i} [m_i(I, p_i) - d_i(p_0, p_i)]$$

DPM detection



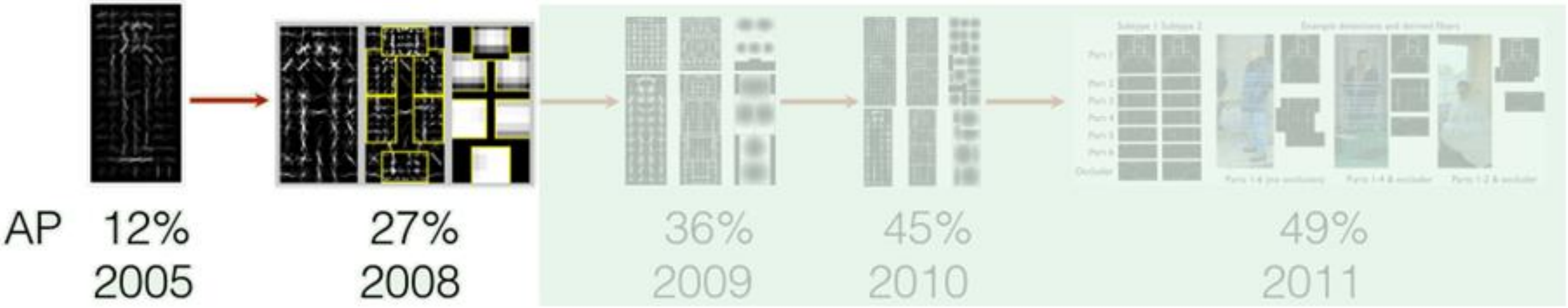
All that's left: combine evidence

DPM detection



Person detection progress

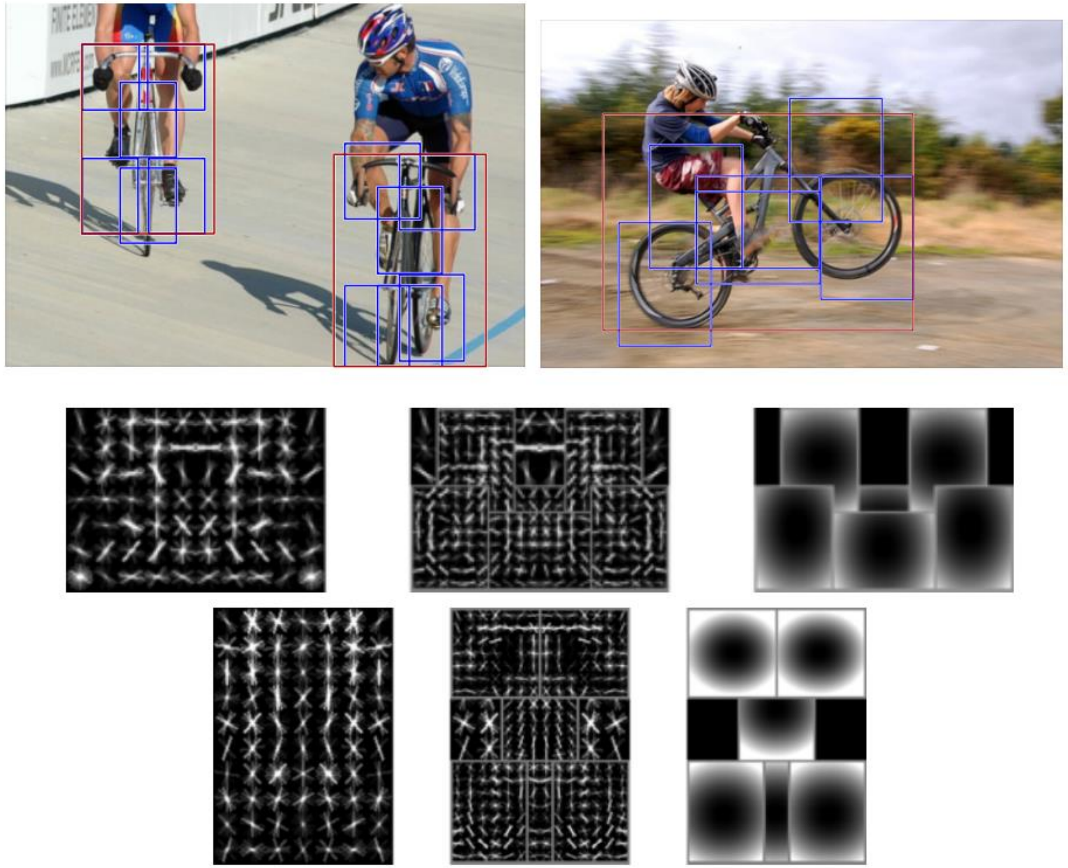
Progress bar:



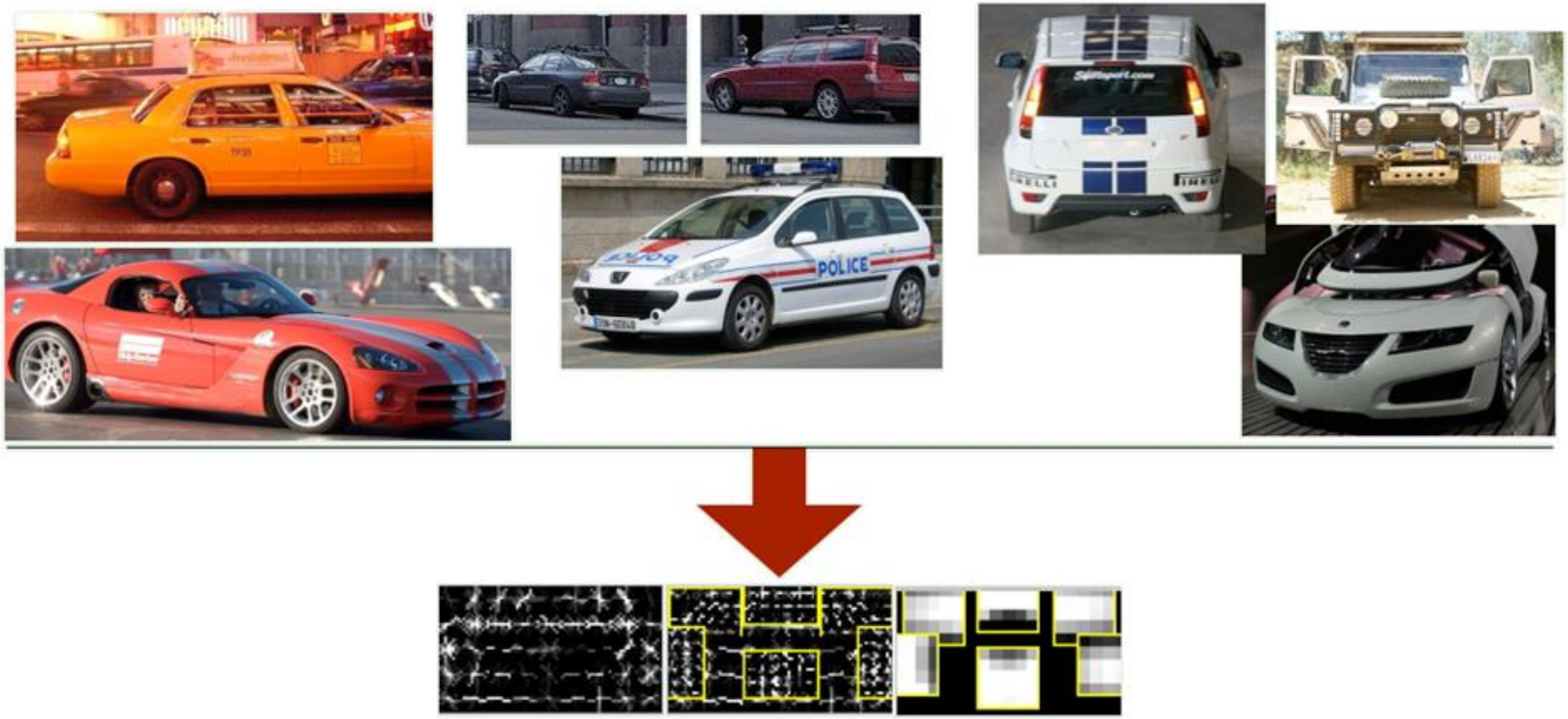
One DPM is not enough: **What are the parts?**

Mixture Models

- Modelling for objects is done using multiple orientations
- Models subject to translation and rotation around the axis perpendicular to the page



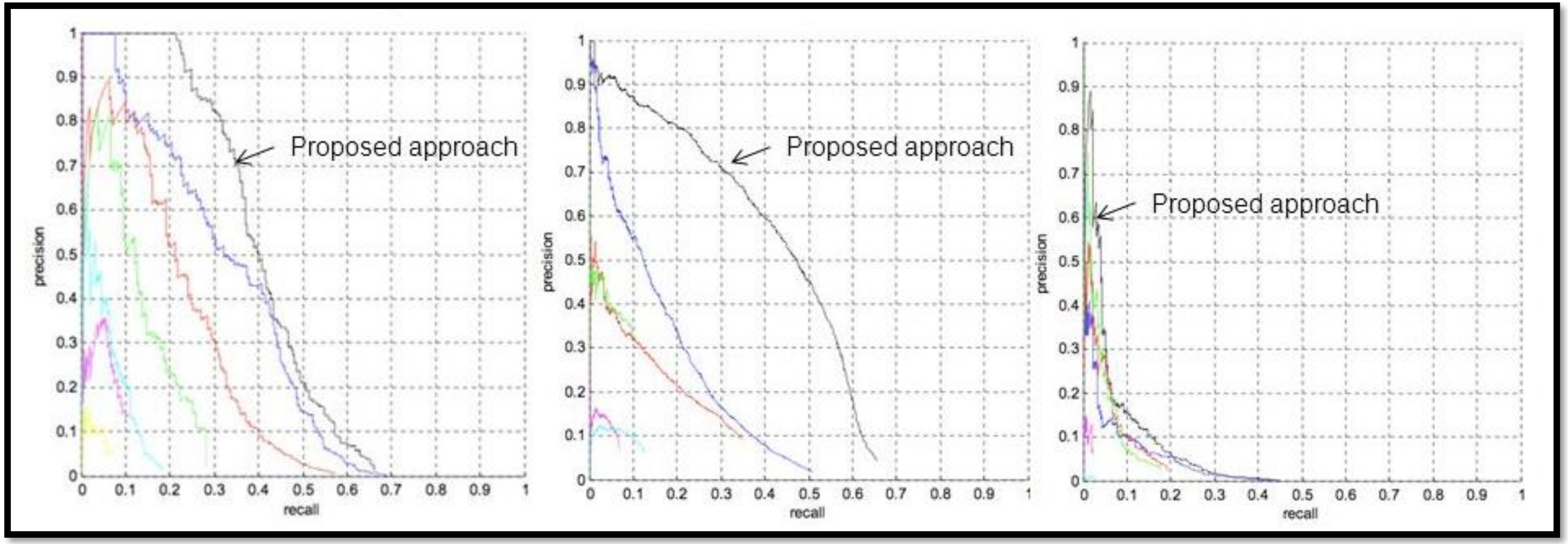
Aspect soup



General philosophy: enrich models to better represent the data

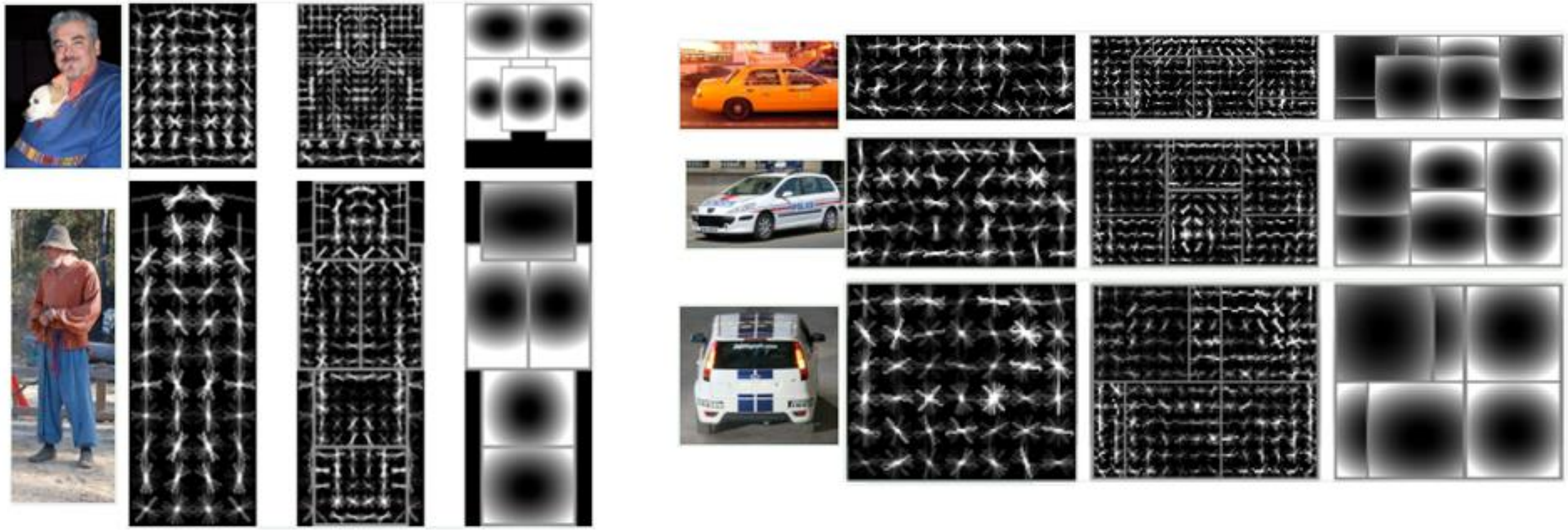
Results (PASCAL VOC 2008)

- Seven total systems competed
- DPM placed first in 7/20 categories

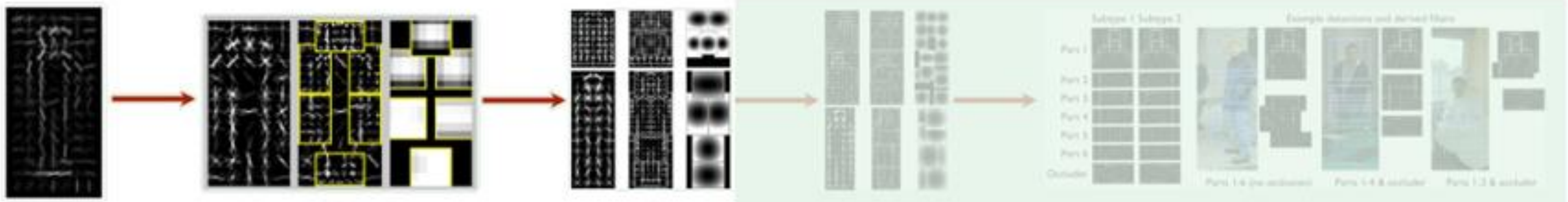


Mixture models

Data driven: aspect, occlusion modes, subclasses



Progress bar:

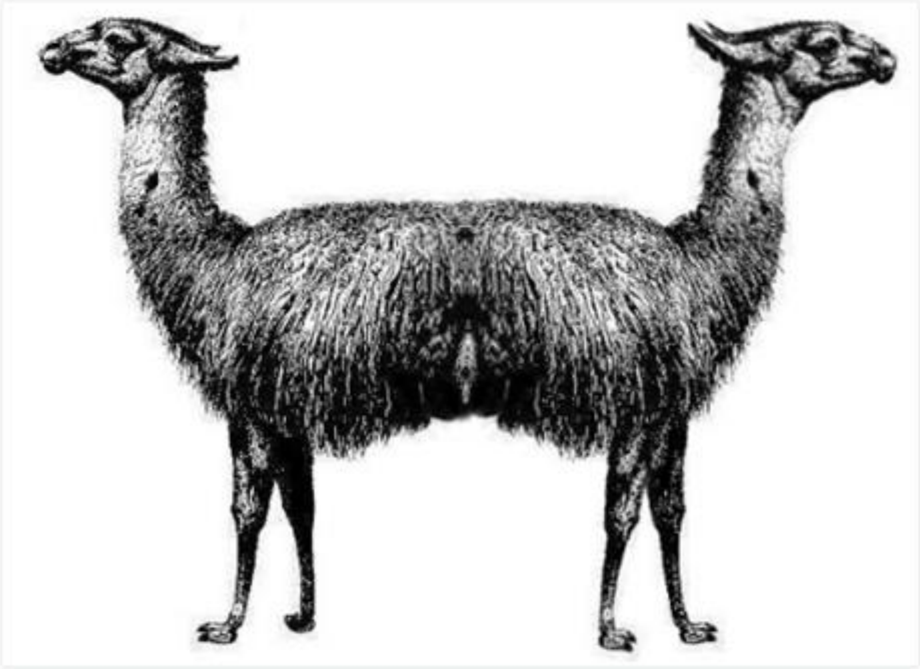


AP 12% 2005 | 27% 2008 | 36% 2009 | 45% 2010 | 49% 2011

Pushmi–pullyu?

Good generalization properties on Doctor Dolittle's farm

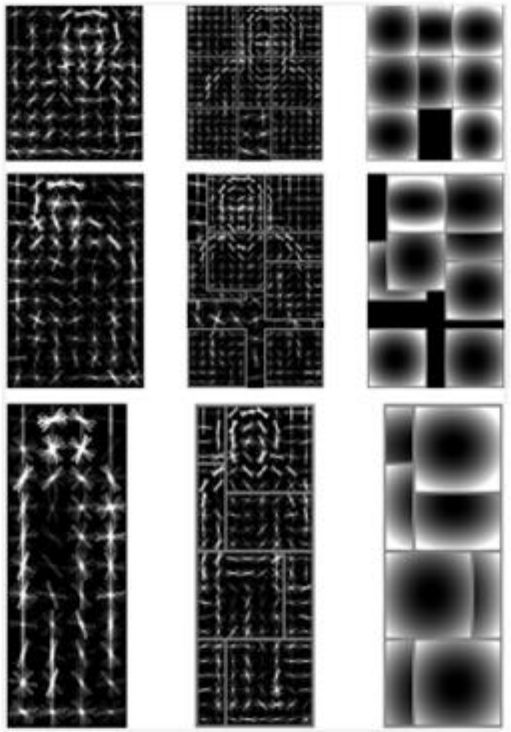
( + ) / 2 =



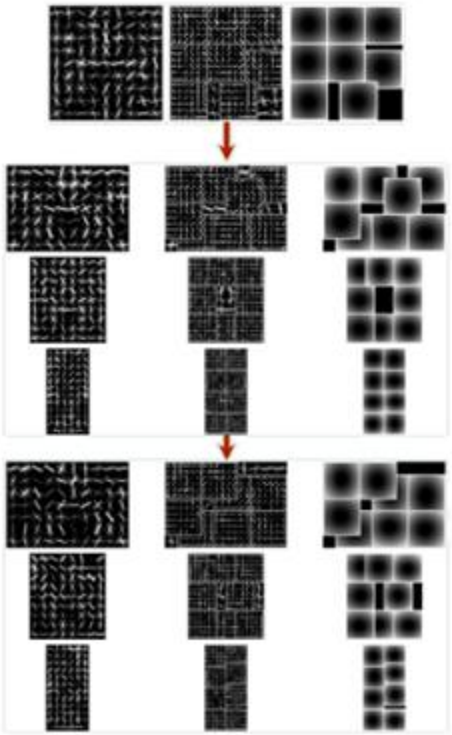
This was supposed to detect horses

Latent orientation

Unsupervised left/right orientation discovery



horse AP

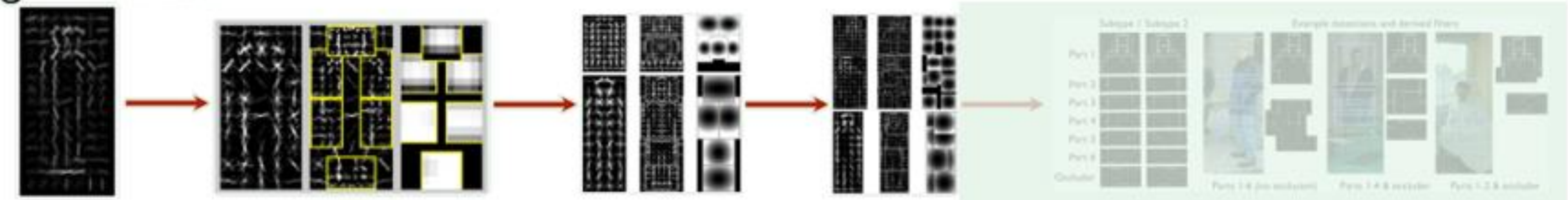


0.42

0.47

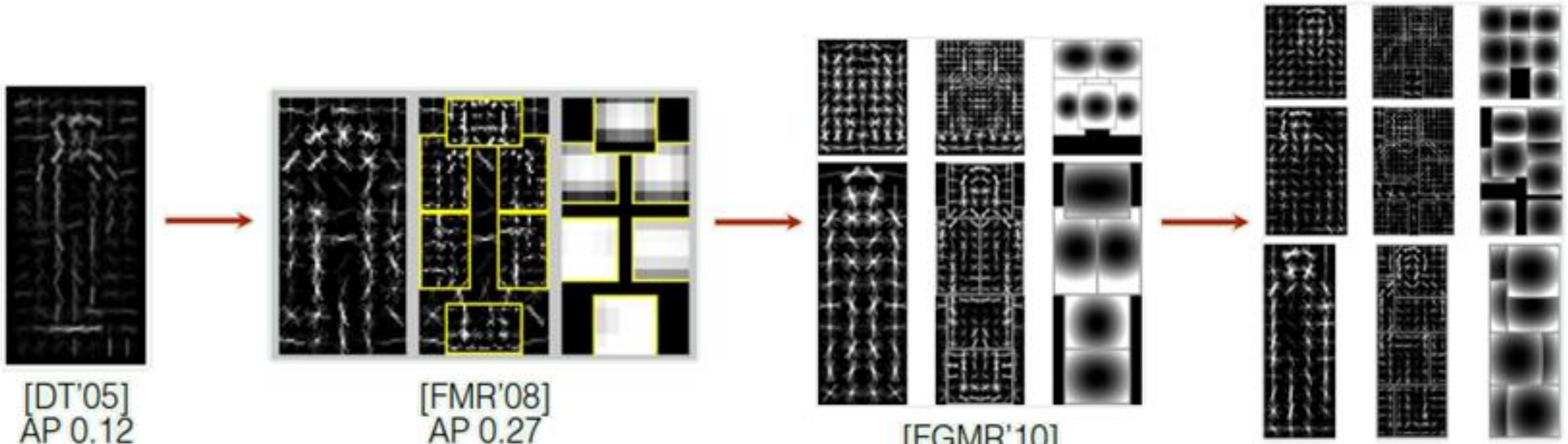
0.57

Progress bar:



AP	12%	27%	36%	45%	49%
	2005	2008	2009	2010	2011

Summary of results

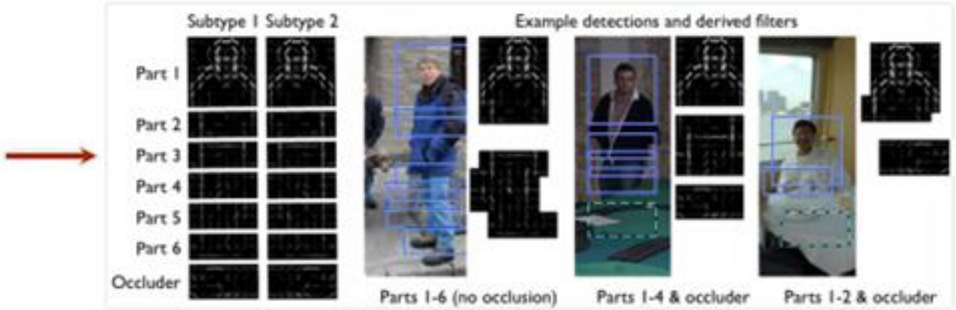


[DT'05]
AP 0.12

[FMR'08]
AP 0.27

[FGMR'10]
AP 0.36

[GFM voc-release5]
AP 0.45



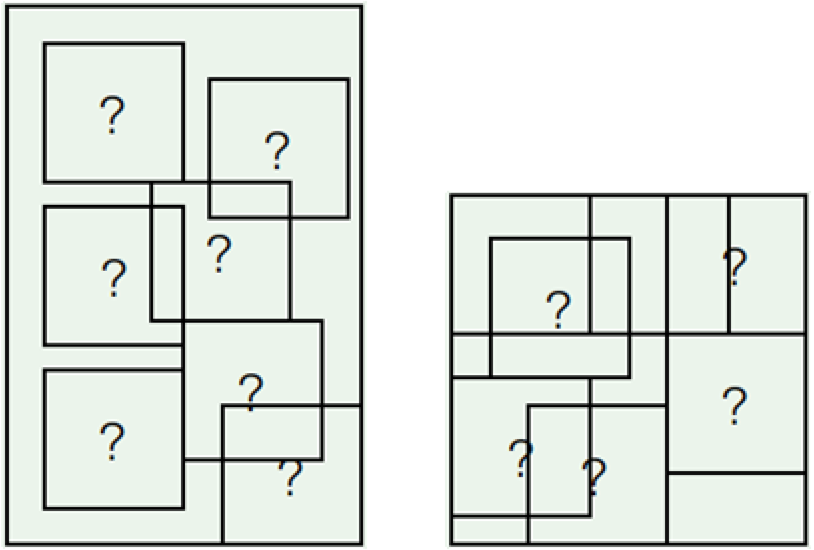
[Girshick, Felzenszwalb, McAllester '11]
AP 0.49

Object detection with grammar models

Code at www.cs.berkeley.edu/~rbg/voc-release5

Part 2: DPM parameter learning

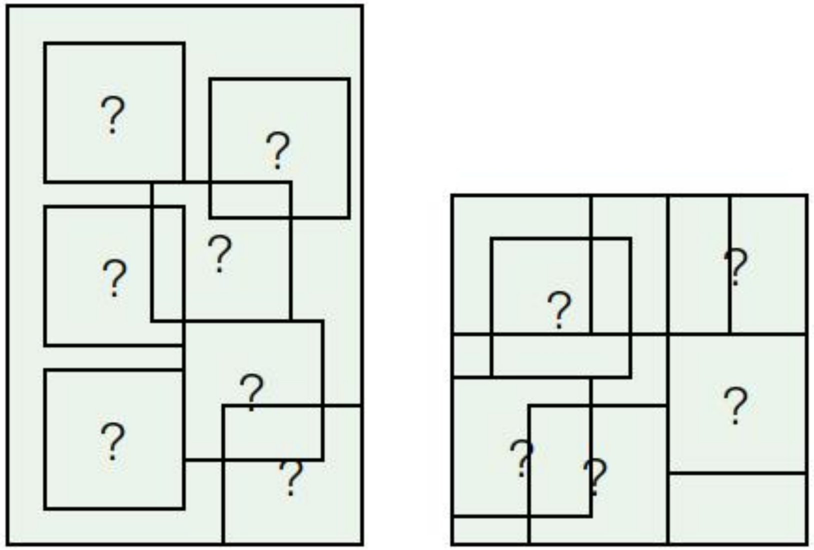
given fixed model *structure*



component 1 component 2

Part 2: DPM parameter learning

given fixed model *structure*



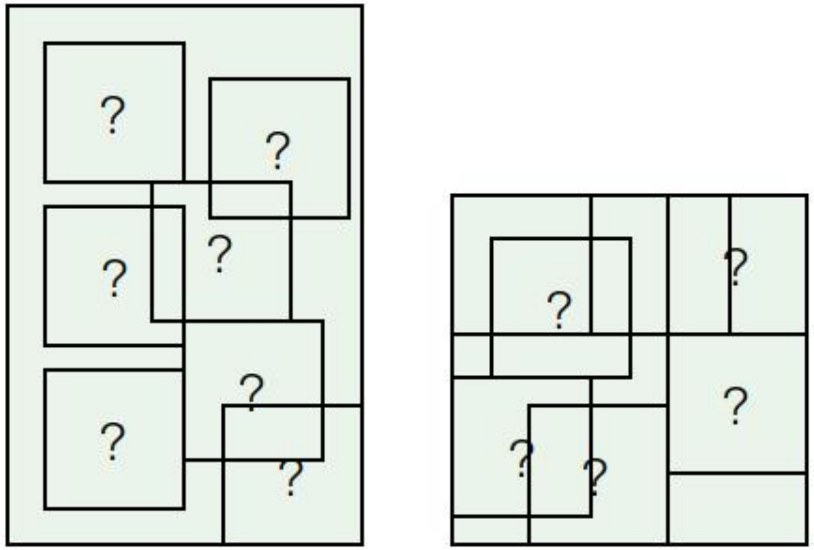
component 1 component 2

training images y

+1

Part 2: DPM parameter learning

given fixed model *structure*



component 1 component 2

training images

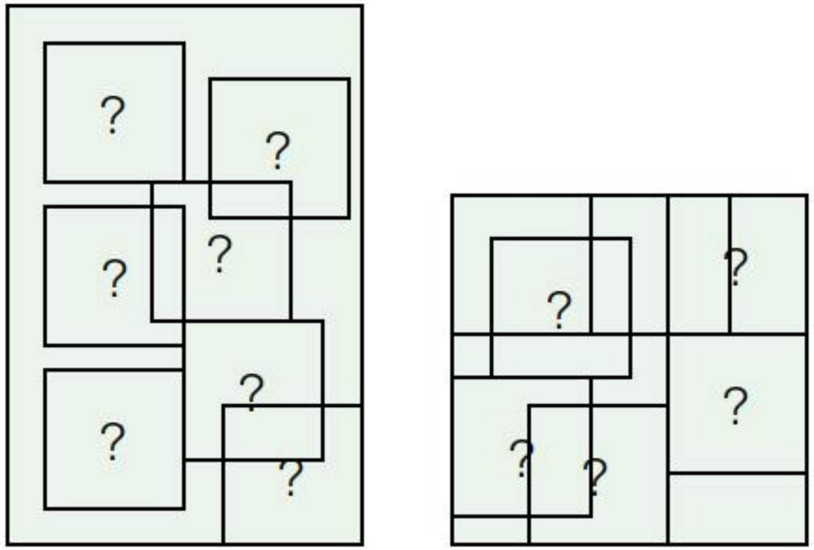
y

+1

-1

Part 2: DPM parameter learning

given fixed model *structure*



component 1 component 2

training images *y*

+1

-1

Parameters to learn:

- biases (per component)
- deformation costs (per part)
- filter weights

Linear parameterization of sliding window score

$$z = (p_1, \ldots, p_n)$$

$$\text{score}(I, p_0) = \max_{p_1,\ldots,p_n} \underbrace{\sum_{i=0}^{n} m_i(I, p_i)}_{\text{Filter scores}} - \underbrace{\sum_{i=1}^{n} d_i(p_0, p_i)}_{\text{Spring costs}}$$

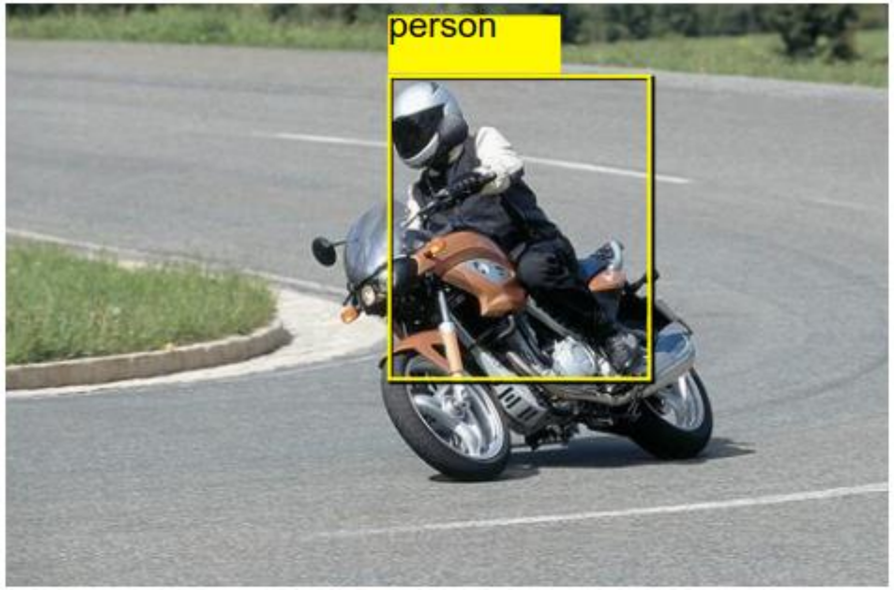
Filter scores $$m_i(I, p_i) = \mathbf{w}_i \cdot \phi(I, p_i)$$

Spring costs $$d_i(p_0, p_i) = \mathbf{d}_i \cdot (dx^2, dy^2, dx, dy)$$

$$\boxed{score(I, p_0) = \max_{z} \mathbf{w} \cdot \Phi(I, (p_0, z))}$$

Positive examples ($y = +1$)

x specifies an image and bounding box



We want

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

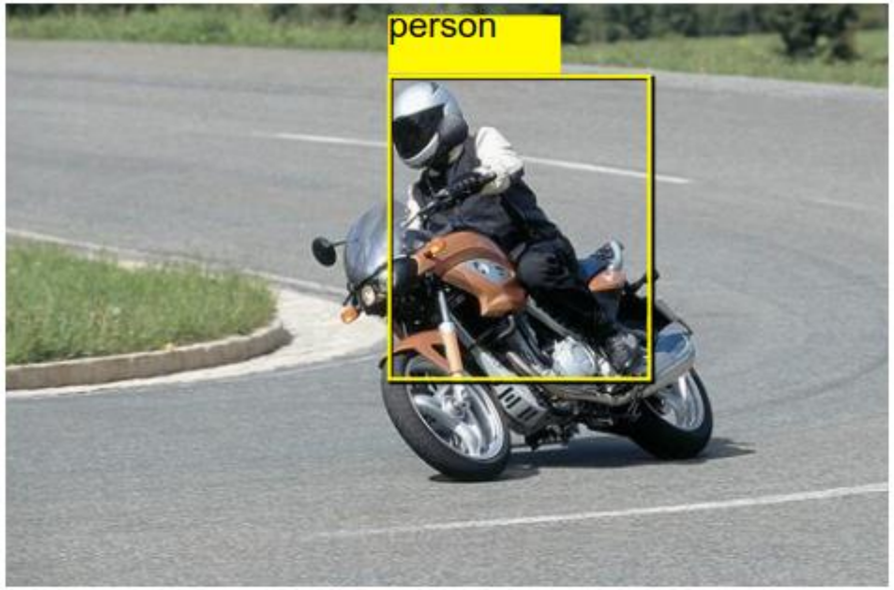
to score >= +1

$Z(x)$ includes all z with more than 70% overlap with ground truth

Positive examples ($y = +1$)

Positive examples ($y = +1$)

x specifies an image and bounding box



We want

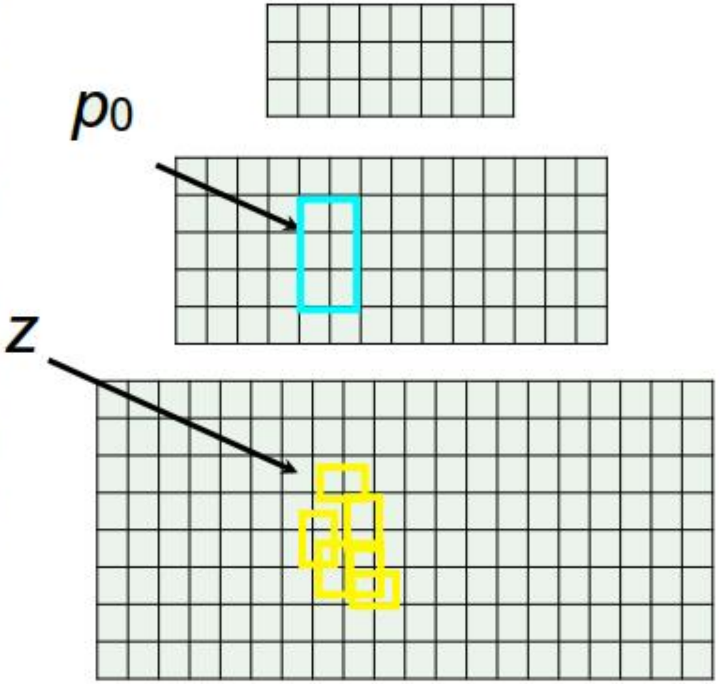
$$f_{\mathbf{w}}(x) = \max_{z \in Z(x)} \mathbf{w} \cdot \Phi(x, z)$$

to score >= +1

At least one configuration scores high

Negative examples (y = -1)

x specifies an image and a HOG pyramid location p_0



We want

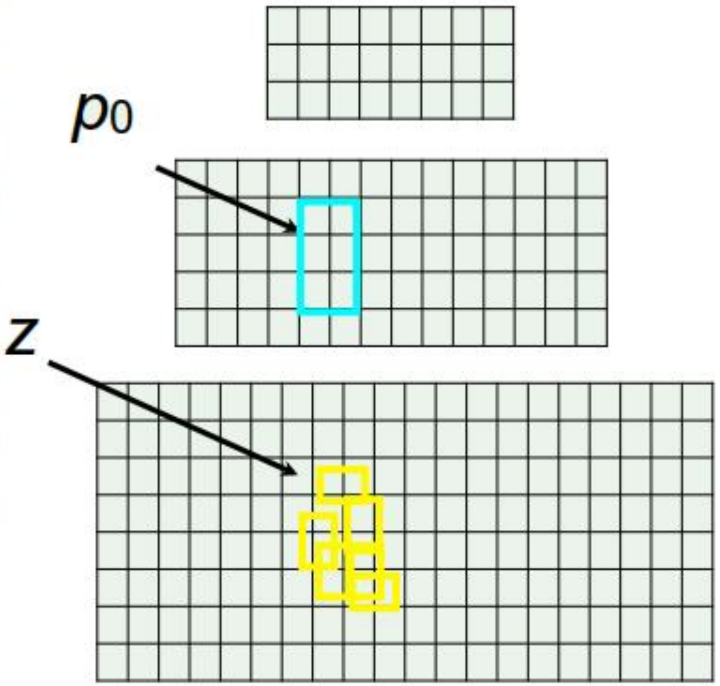
$$f_{\mathbf{w}}(x) = \max_{z \in Z(x)} \mathbf{w} \cdot \Phi(x, z)$$

to score <= -1

$Z(x)$ restricts the root to p_0 and allows *any* placement of the other filters

Negative examples (y = -1)

x specifies an image and a HOG pyramid location p_0



We want

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

to score <= -1

All configurations score low

$Z(x)$ restricts the root to p_0 and allows *any* placement of the other filters

Typical dataset

300 – 8,000 positive examples

500 million to 1 billion negative examples
(not including latent configurations!)

Large-scale optimization!

How we learn parameters: latent SVM

$$E(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \max\{0, 1 - y_i f_{\mathbf{w}}(x_i)\}$$

How we learn parameters: latent SVM

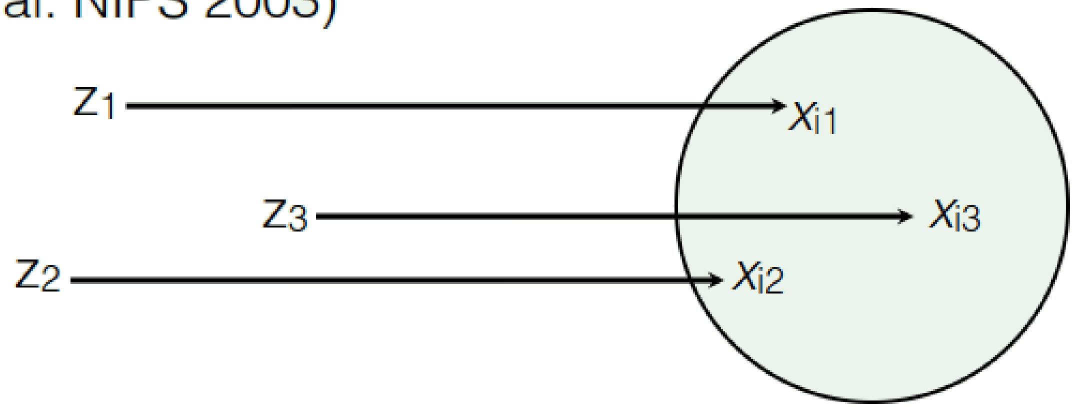
$$E(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_i \max\{0, 1 - y_i f_{\mathbf{w}}(x_i)\}$$

$$E(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i\in P} \max\{0, 1 - \max_{z\in Z(x)} \mathbf{w}\cdot\Phi(x_i, z)\}$$
$$+ C\sum_{i\in N} \max\{0, 1 + \max_{z\in Z(x)} \mathbf{w}\cdot\Phi(x_i, z)\}$$

P: set of positive examples
N: set of negative examples

Latent SVM and Multiple Instance Learning via MI-SVM

Latent SVM is mathematically equivalent to MI-SVM
(Andrews et al. NIPS 2003)



latent labels for x_i bag of instances for x_i

Latent SVM can be written as a latent structural SVM
(Yu and Joachims ICML 2009)

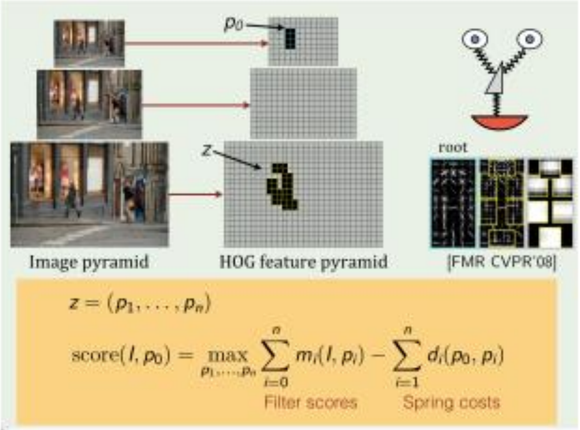
– natural optimization algorithm is concave-convex procedure

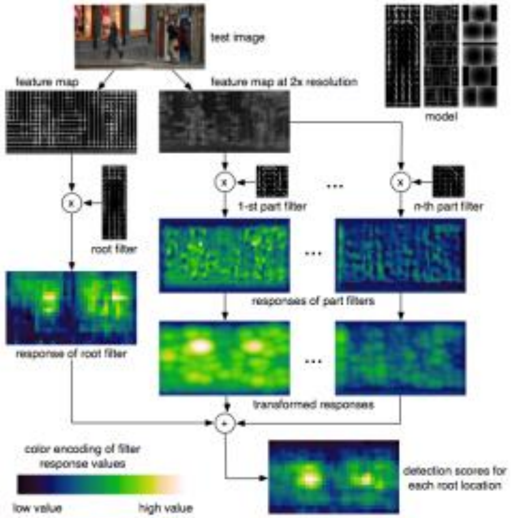
– similar to, but not exactly the same as, coordinate descent

Step 1

$$Z_{P_i} = \underset{z \in Z(x_i)}{\operatorname{argmax}} \, \mathbf{w}_{(t)} \cdot \Phi(x_i, z) \quad \forall i \in P$$

This is just detection:





We know how to do this!

Step 2

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i\in P}\max\{0, 1-\mathbf{w}\cdot\Phi(x_i, Z_{Pi})\}$$
$$+ C\sum_{i\in N}\max\{0, 1+\max_{z\in Z(x)}\mathbf{w}\cdot\Phi(x_i, z)\}$$

Convex!

Step 2

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i \in P} \max\{0, 1 - \mathbf{w} \cdot \Phi(x_i, Z_{Pi})\}$$
$$+ C\sum_{i \in N} \max\{0, 1 + \max_{z \in Z(x)} \mathbf{w} \cdot \Phi(x_i, z)\}$$

Convex!

Similar to a structural SVM

Step 2

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i\in P} \max\{0, 1 - \mathbf{w}\cdot\Phi(x_i, Z_{Pi})\}$$
$$+ C\sum_{i\in N} \max\{0, 1 + \max_{z\in Z(x)} \mathbf{w}\cdot\Phi(x_i, z)\}$$

Convex!

Similar to a structural SVM

But, recall 500 million to 1 billion negative examples!

Step 2

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i\in P}\max\{0, 1-\mathbf{w}\cdot\Phi(x_i, Z_{Pi})\}$$

$$+ C\sum_{i\in N}\max\{0, 1+\max_{z\in Z(x)}\mathbf{w}\cdot\Phi(x_i, z)\}$$

Convex!

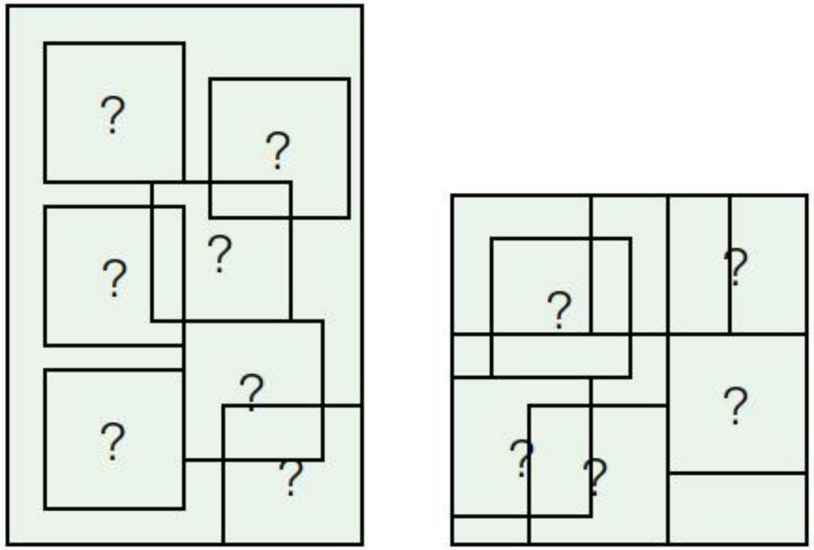
Similar to a structural SVM

But, recall 500 million to 1 billion negative examples!

Can be solved by a working set method
– "bootstrapping"
– "data mining" / "hard negative mining"
– "constraint generation"
– requires a bit of engineering to make this fast

What about the model structure?

<u>Given fixed model *structure*</u>



component 1 component 2

Model structure

– # components
– # parts per component
– root and part filter shapes
– part anchor locations

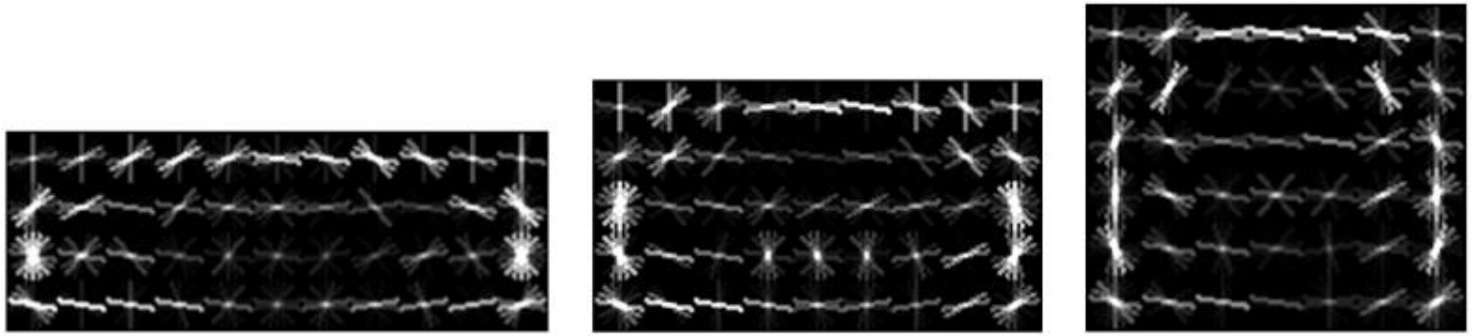
training images y

+1

-1

Learning model structure

1a. Split positives by aspect ratio

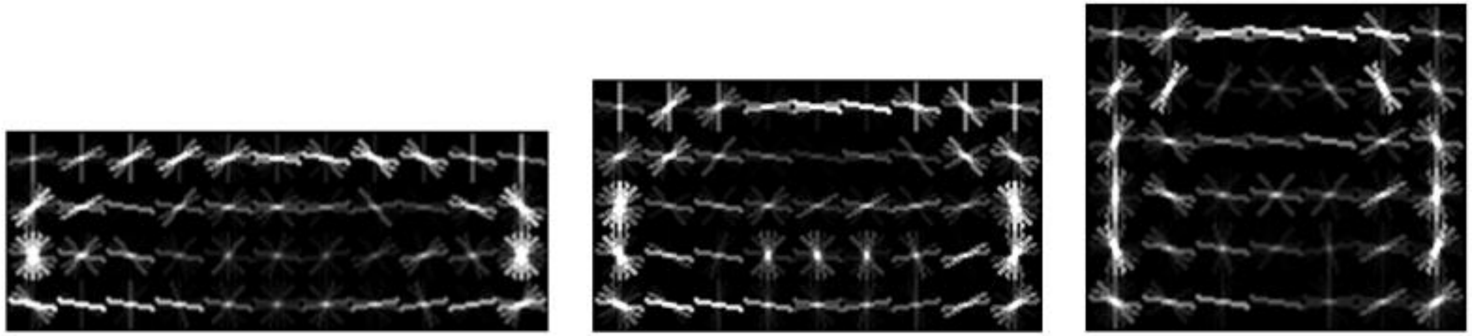


(a) Car component 1 (Phase 1) (b) Car component 2 (Phase 1) (c) Car comp. 3 (Phase 1)

1b. Warp to common size

1c. Train Dalal & Triggs model for each aspect ratio on its own

Learning model structure



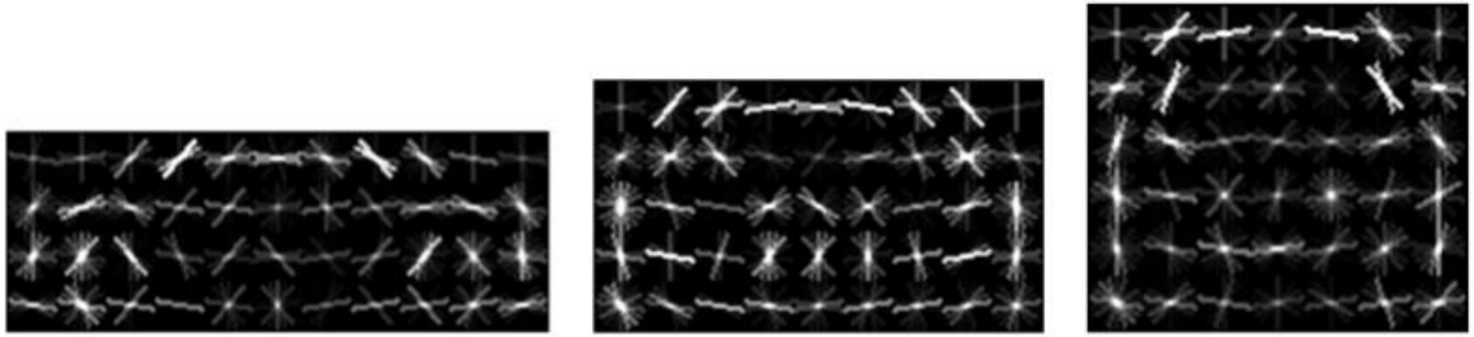
(a) Car component 1 (Phase 1) (b) Car component 2 (Phase 1) (c) Car comp. 3 (Phase 1)

2a. Use D&T filters as initial **w** for LSVM training

Merge components

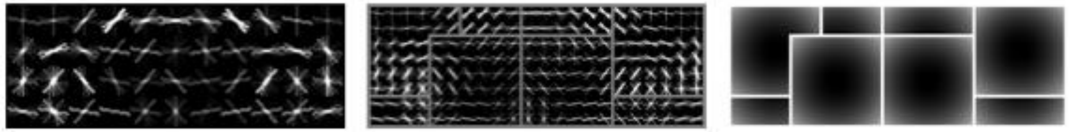
2b. Train with latent SVM

Root filter placement and component choice are latent

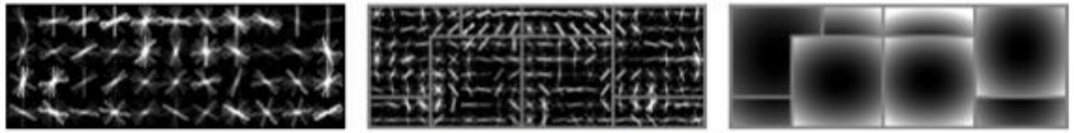


(d) Car component 1 (Phase 2) (e) Car component 2 (Phase 2) (f) Car comp. 3 (Phase 2)

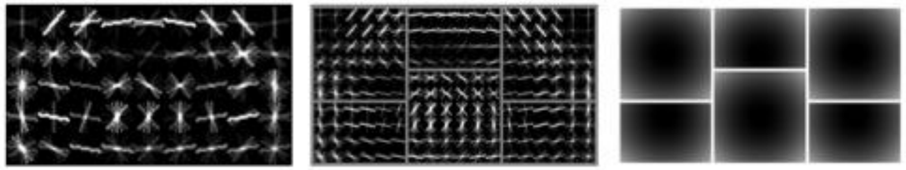
Learning model structure



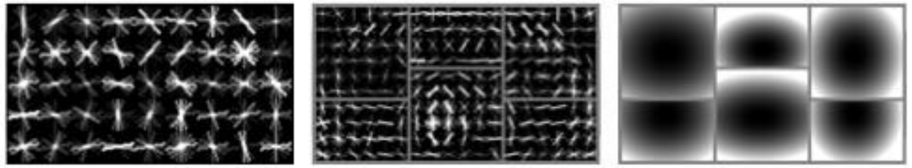
(a) Car component 1 (initial parts)



(b) Car component 1 (trained parts)



(c) Car component 2 (initial parts)

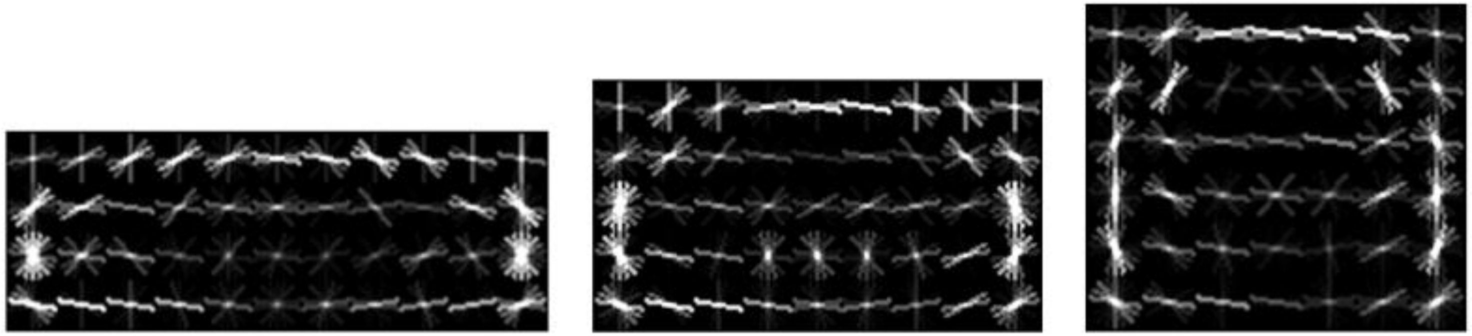


(d) Car component 2 (trained parts)

3a. Add parts to cover high-energy areas of root filters

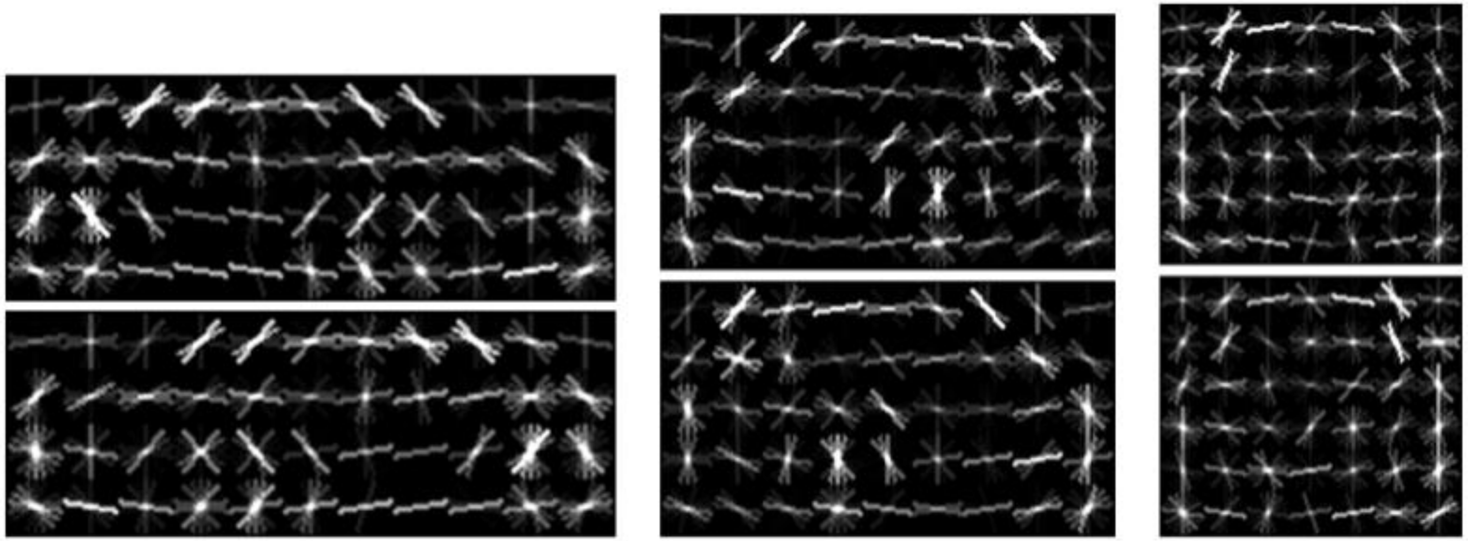
3b. Continue training model with LSVM

Learning model structure



(a) Car component 1 (Phase 1) (b) Car component 2 (Phase 1) (c) Car comp. 3 (Phase 1)

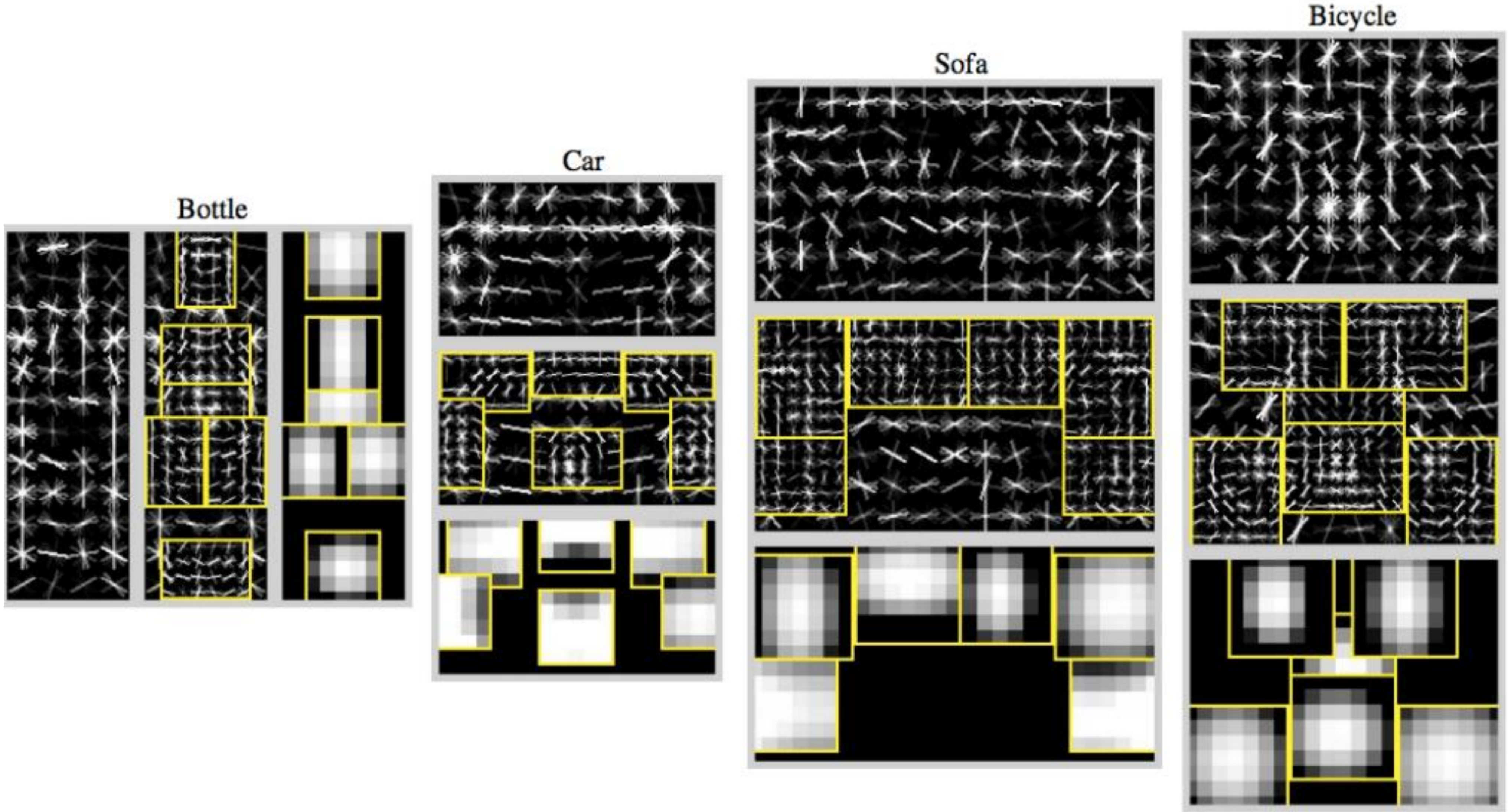
without orientation clustering



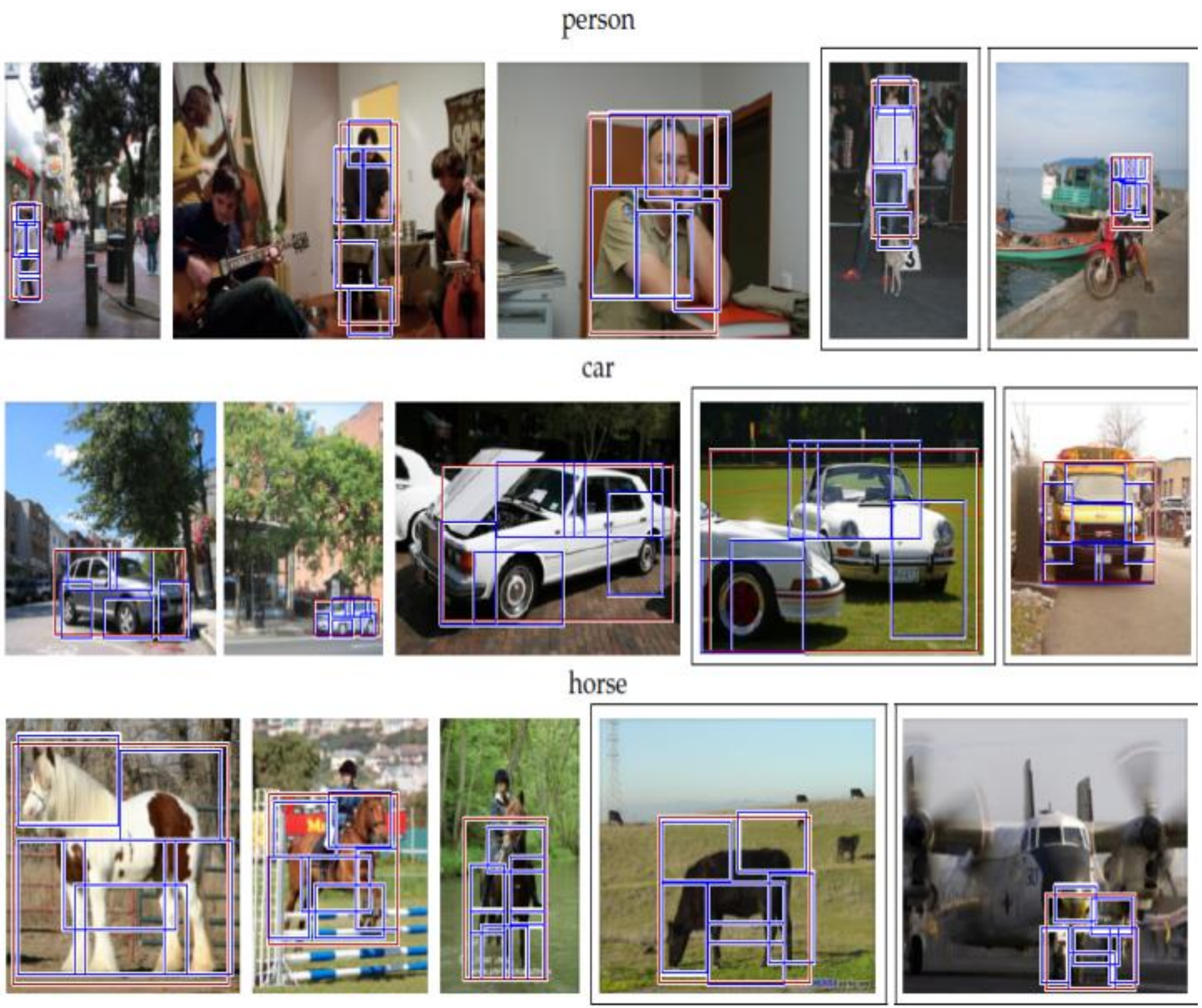
(a) Car component 1 (b) Car component 2 (c) Car component 3

with orientation clustering

DPM learnt models

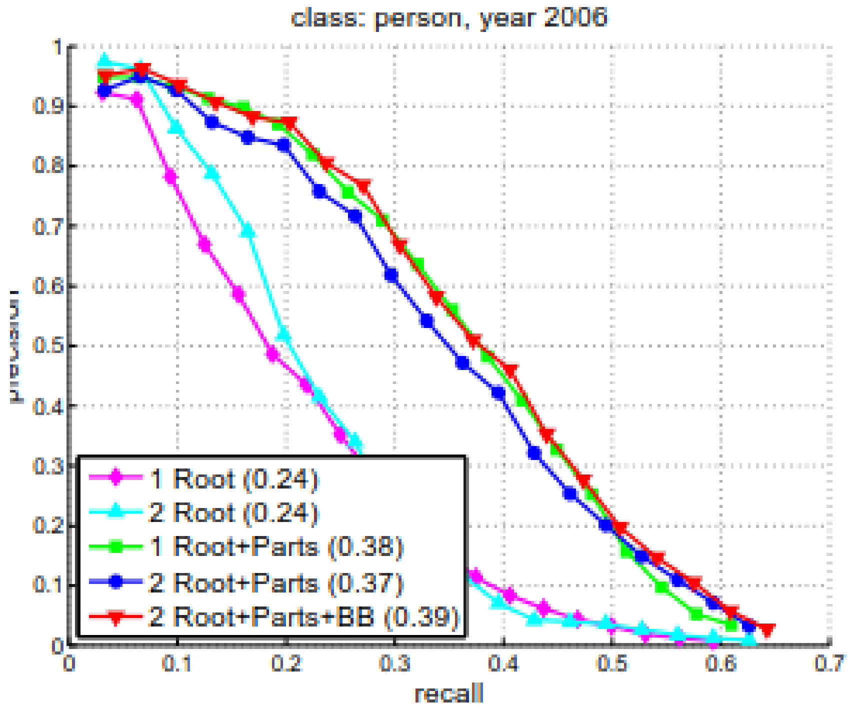


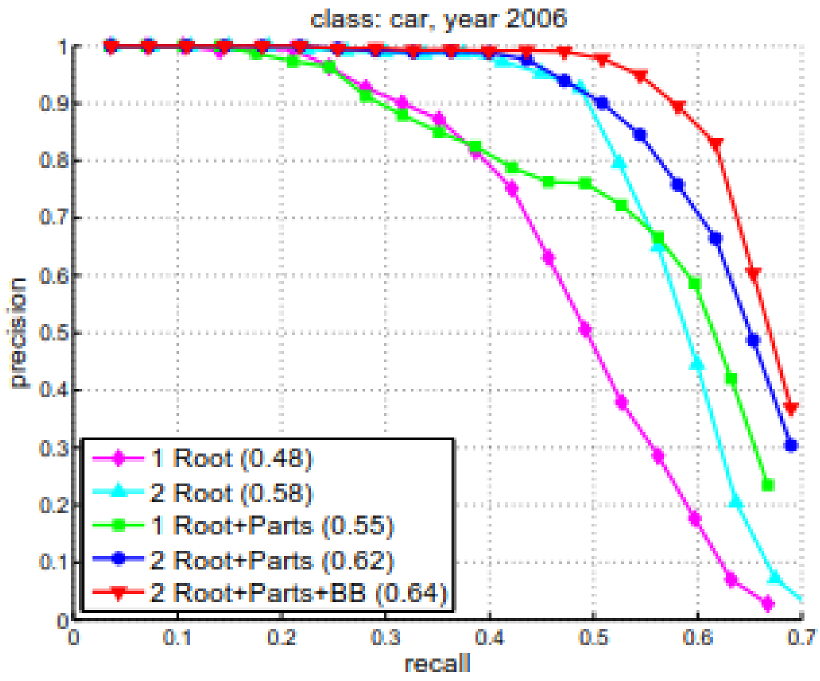
Results



Results

Effects of multiple models + parts





Questions?