LSVM-MDPM

LSVM - Mixtures of Deformable Part Models

Pedro Felzenszwalb, Ross Girshick
University of Chicago

David McAllester
TTI-C

Presented by Deva Ramanan, UC Irvine

Reference

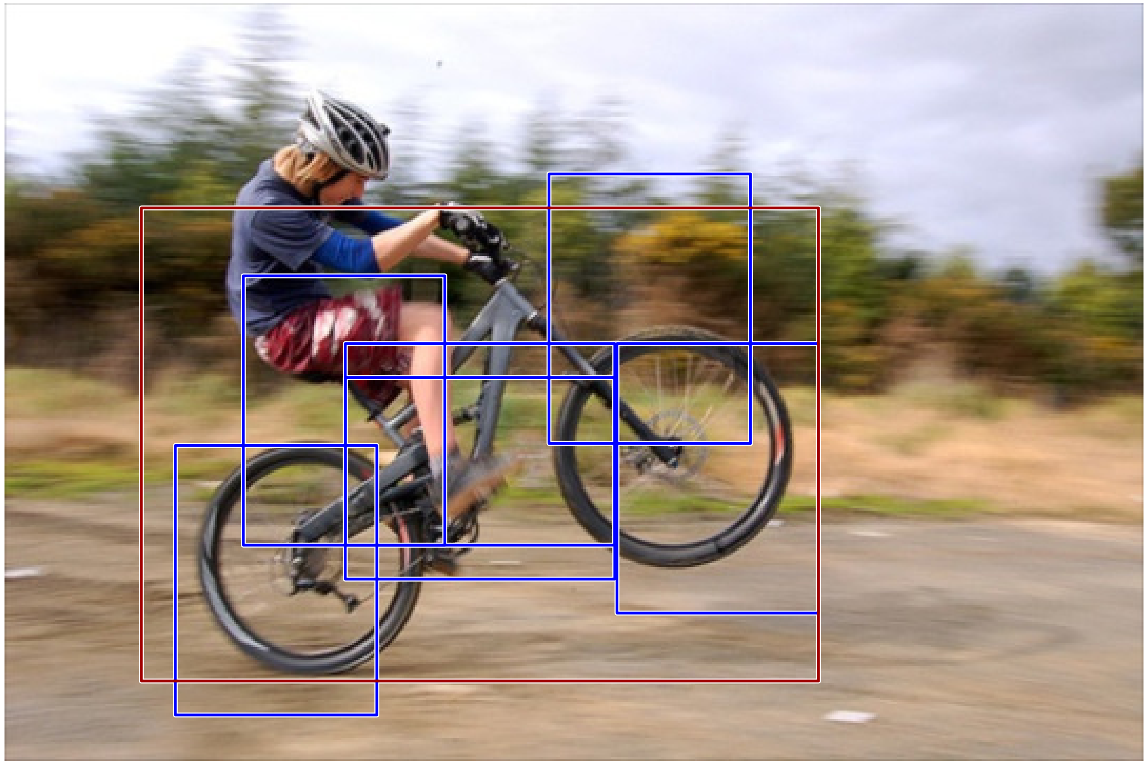
Object Detection with Discriminatively Trained Part Based Models
Pedro Felzenszwalb, Ross Girshick, David McAllester, Deva Ramanan
IEEE Transactions on Pattern Analysis and Machine Intelligence (preprint)

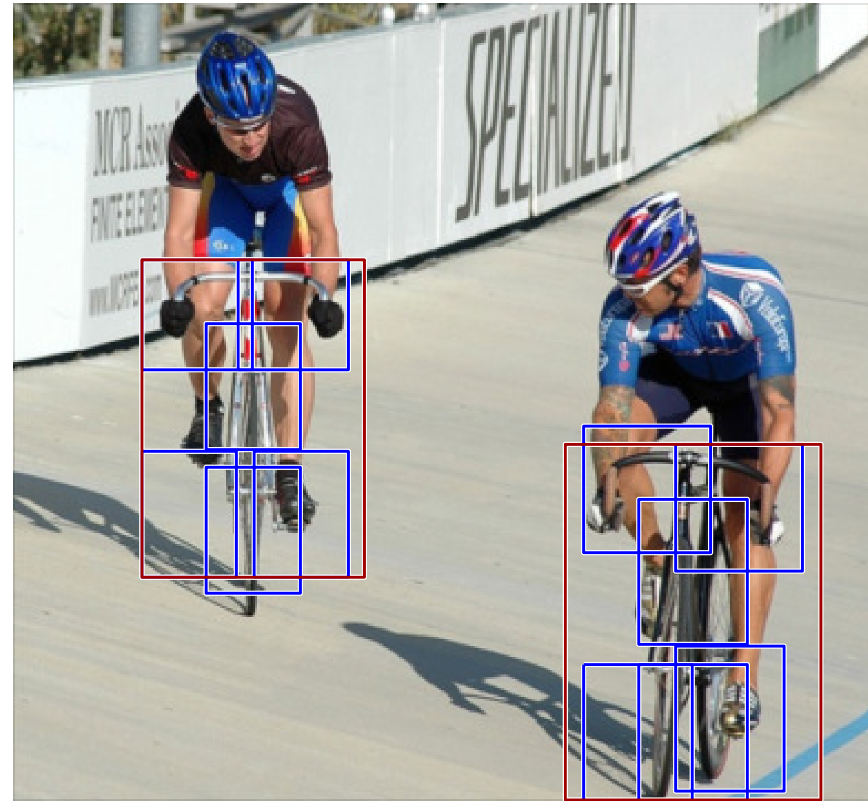
Paper describes general approach and results with 2 component models
For the 2009 competition we trained 6 component models

Code for 2 component models is available online
(new version will be available "soon")

http://www.cs.uchicago.edu/~pff/latent

Overview of our models

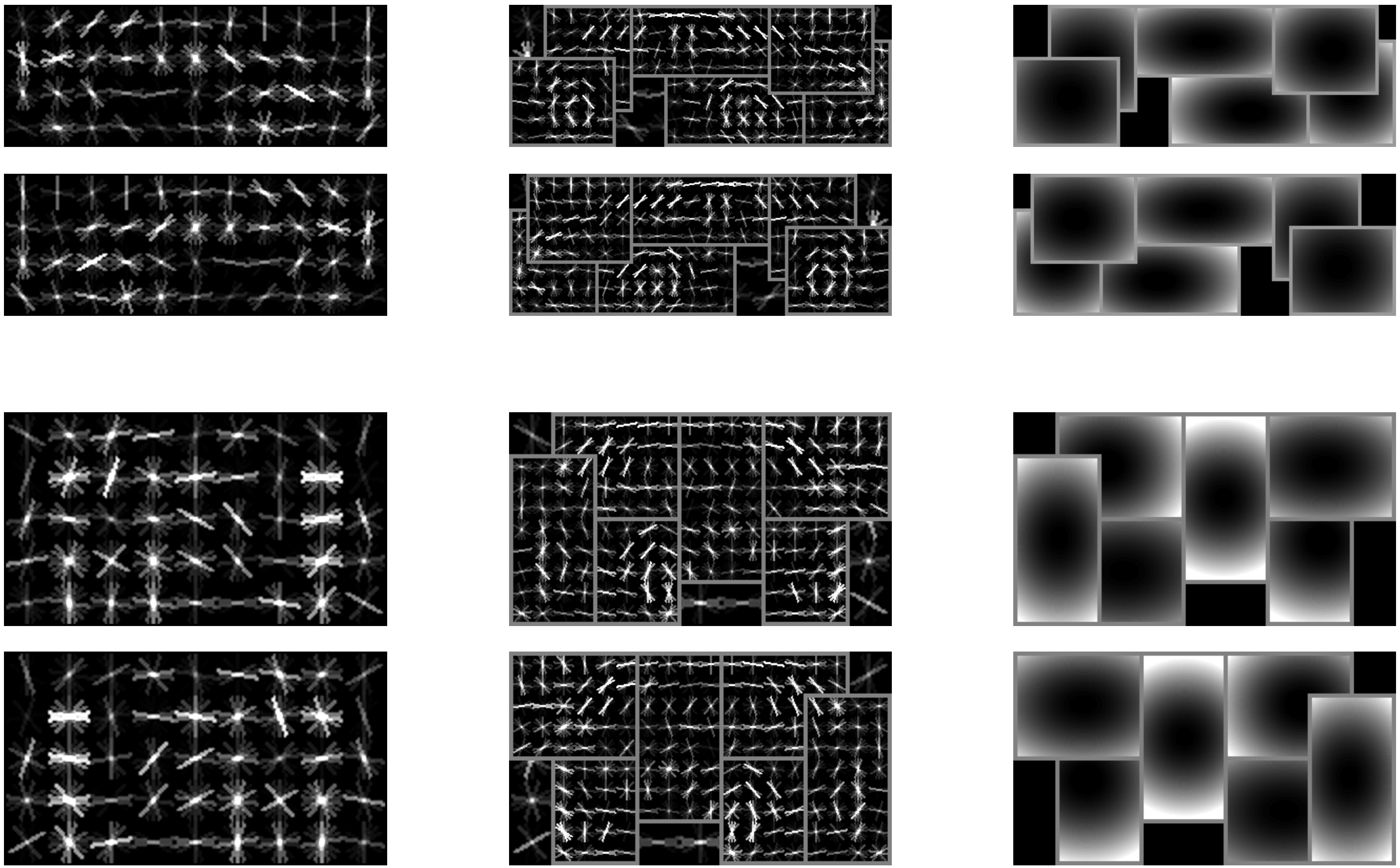




- Mixture of deformable part models (pictorial structures)
- Each component has global template + deformable parts
  - Templates model HOG features
- Fully trained from bounding boxes alone

6 component car model

2 of 3 symmetric pairs shown



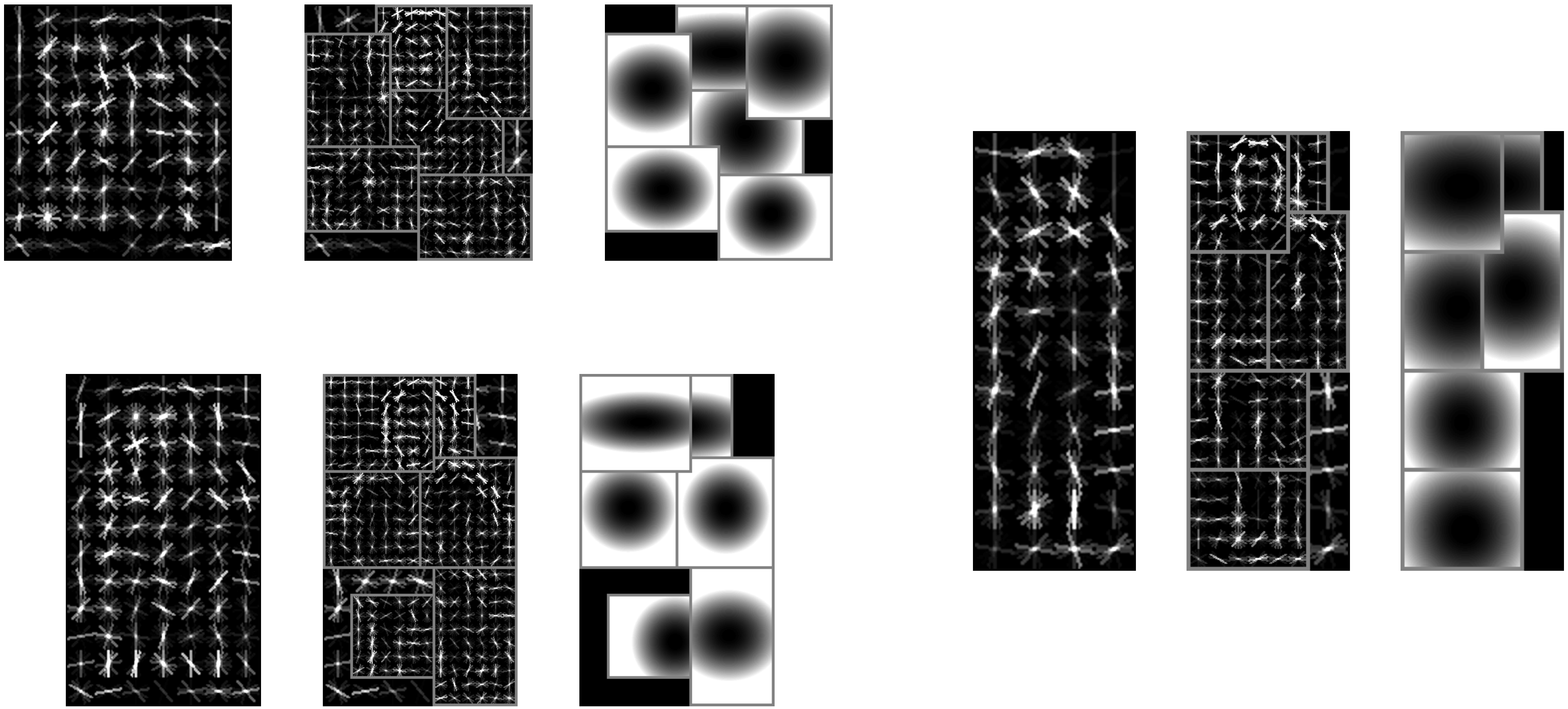
root filters
coarse resolution

part filters
finer resolution

deformation
models

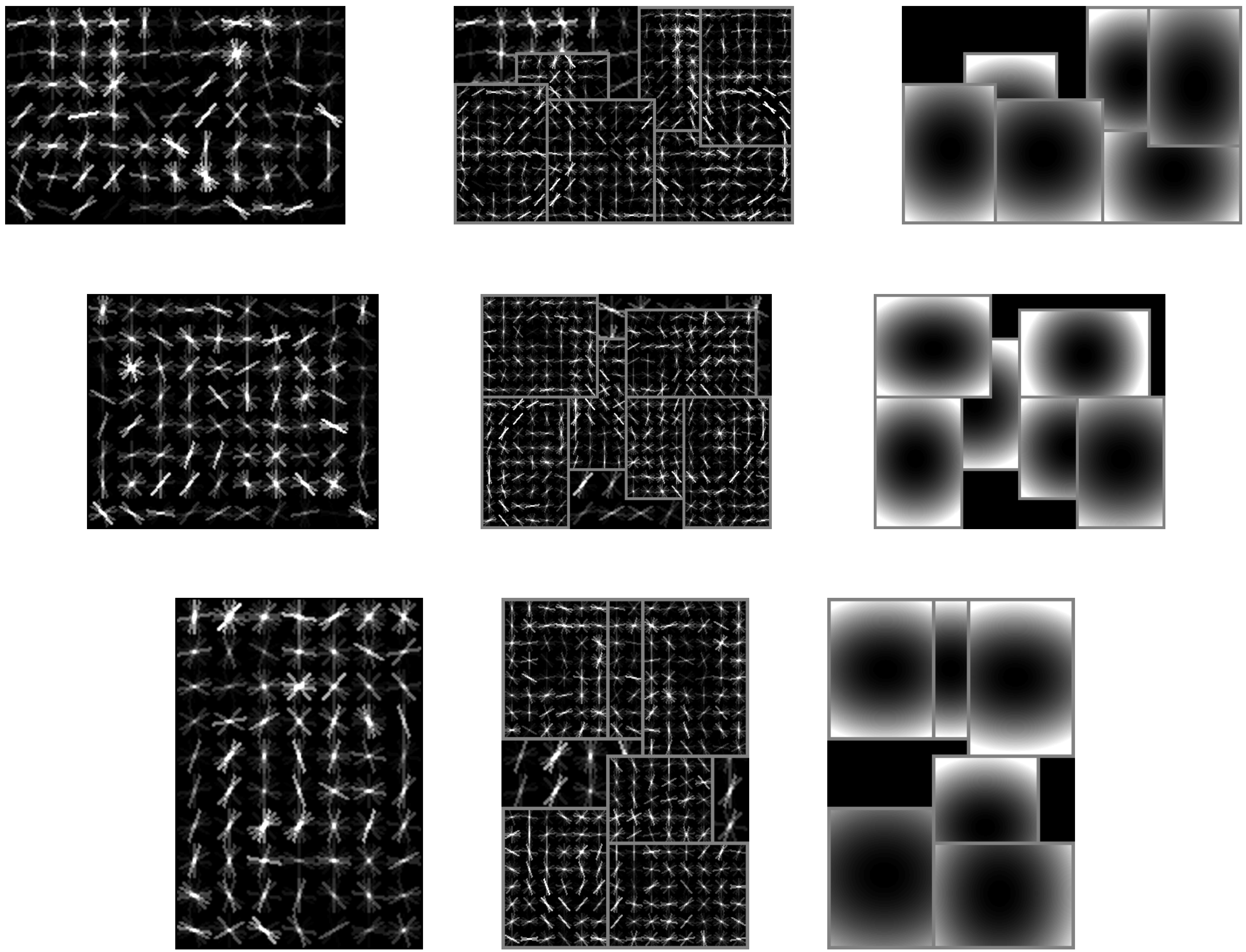
6 component person model

1 component from of each symmetric pair

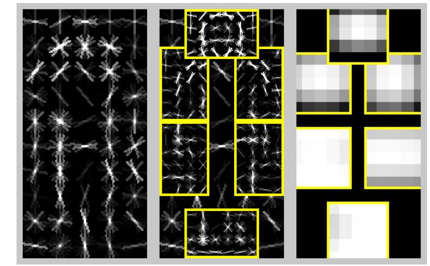


6 component bicycle model

1 component from of each symmetric pair



Object hypothesis



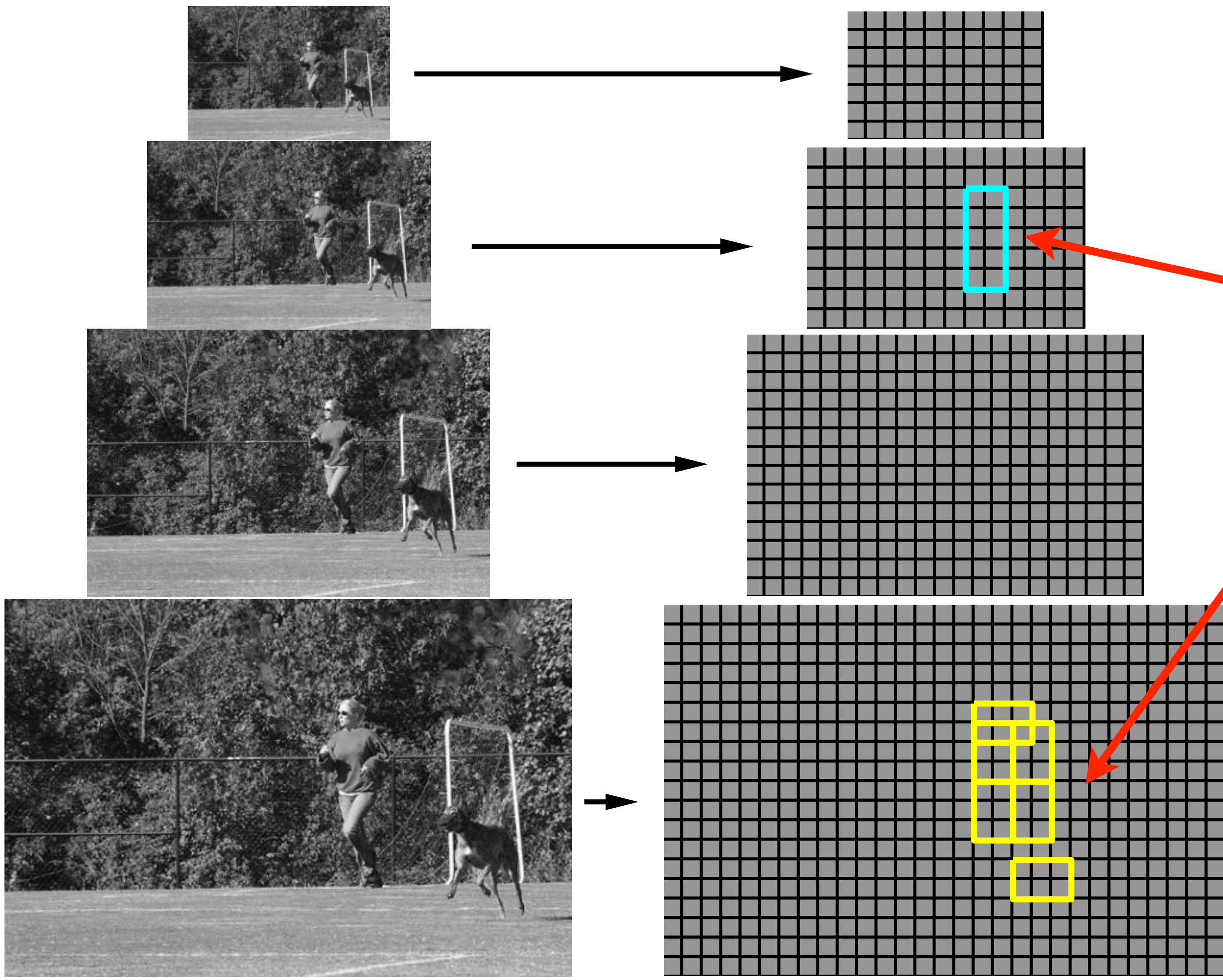


Image pyramid

HOG feature pyramid

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

c : component label

p_0 : location of root

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

Score is sum of filter scores minus deformation costs

Multiscale model captures features at two-resolutions

Score of a hypothesis (single component)

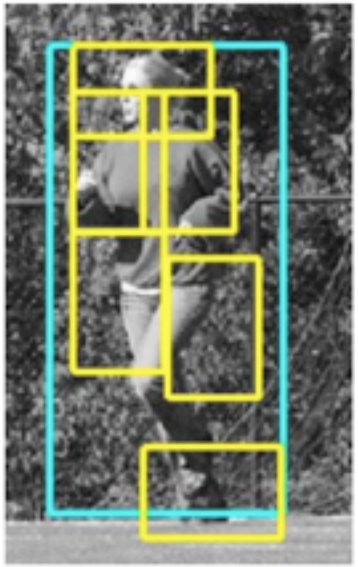
"data term" "spatial prior"

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

filters

deformation parameters

displacements



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

concatenation filters and deformation parameters

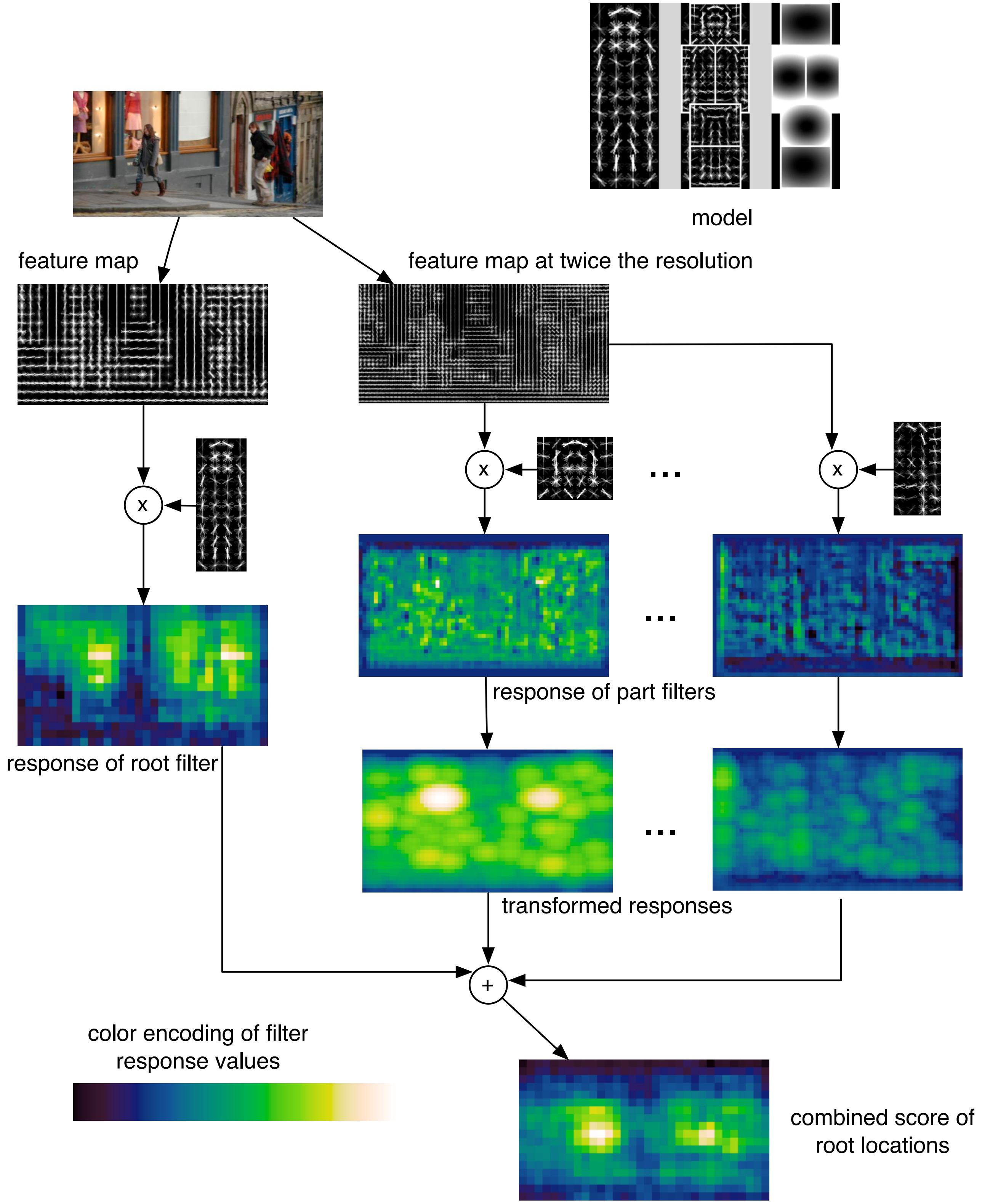
concatenation of HOG features and part displacement features

Matching

- Define an overall score for each root location
  - Based on best placement of parts

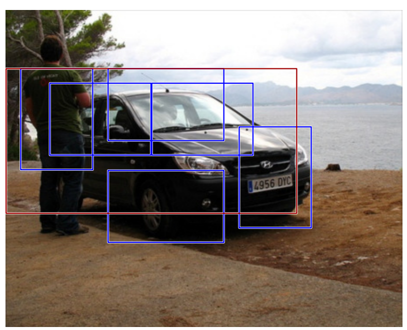
$$\text{score}(p_0) = \max_{p_1,\ldots,p_n} \text{score}(p_0,\ldots,p_n).$$

- High scoring root locations define detections
  - “sliding window approach”
- Efficient computation: dynamic programming + generalized distance transforms (max-convolution)


model
feature map
feature map at twice the resolution
x
x
x
...
...
...
response of part filters
response of root filter
transformed responses
+
color encoding of filter
response values
combined score of
root locations

Post-processing detections

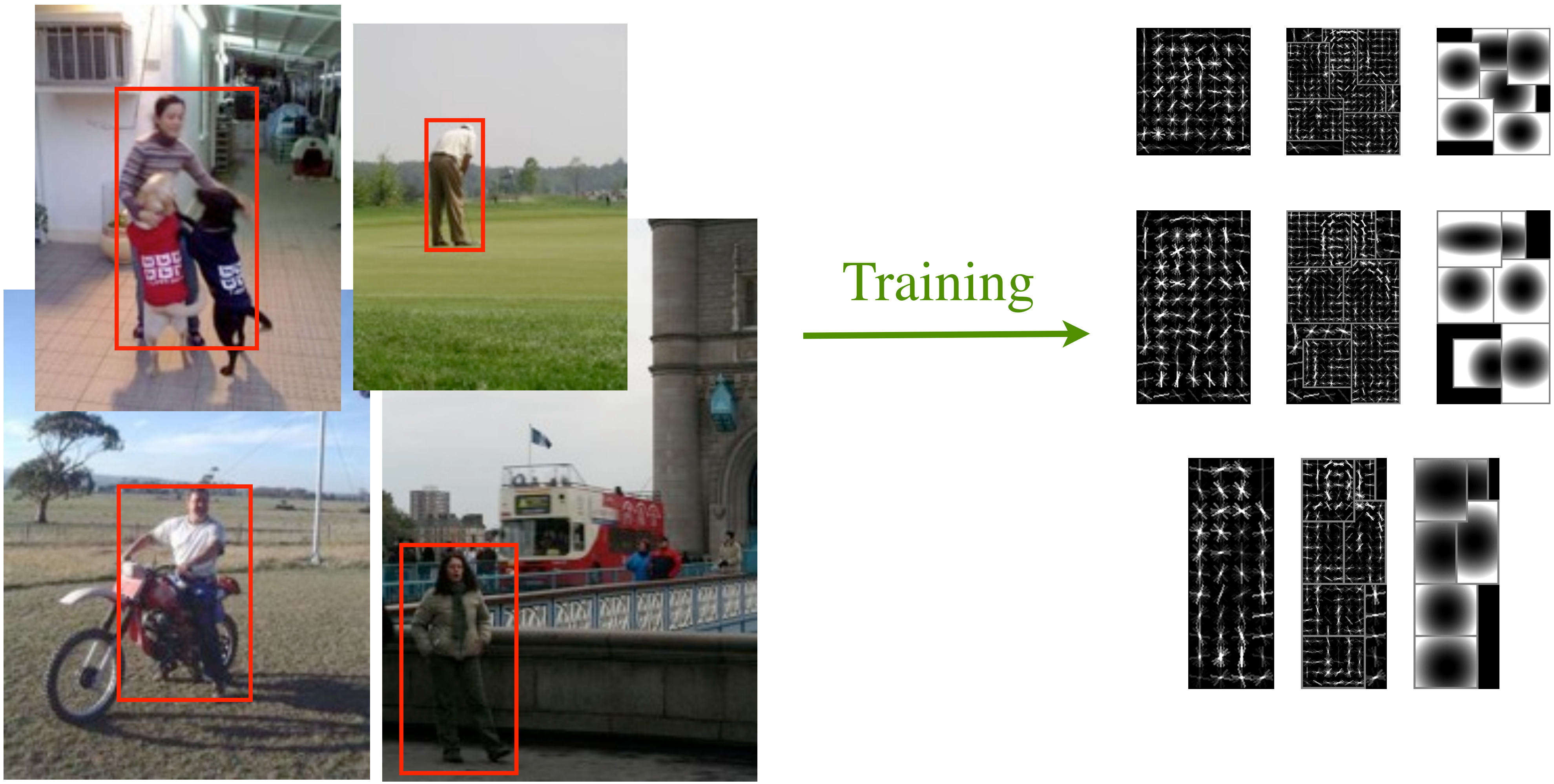
- NMS
  - Remove multiple detections using overlap criteria
- Bounding box prediction
  - Use part locations to predict object bounding box



- "Context rescoring"
  - Rescore a detection based on its location within the image and maximum score of detections from each class

Training

- Training data consists of images with labeled bounding boxes
- Need to learn the model structure, filters and deformation costs



Latent SVM (MI-SVM)

Classifiers that score an example x using

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

β are model parameters
z are latent values

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 β such that: $y_i f_\beta(x_i) > 0$

Minimize

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

Latent SVM training

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

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

- Convex if we fix z for positive examples
- Optimization:
  - Initialize β and iterate:
    - Pick best z for each positive example
    - Optimize β via gradient descent with data-mining

Training 6 component models

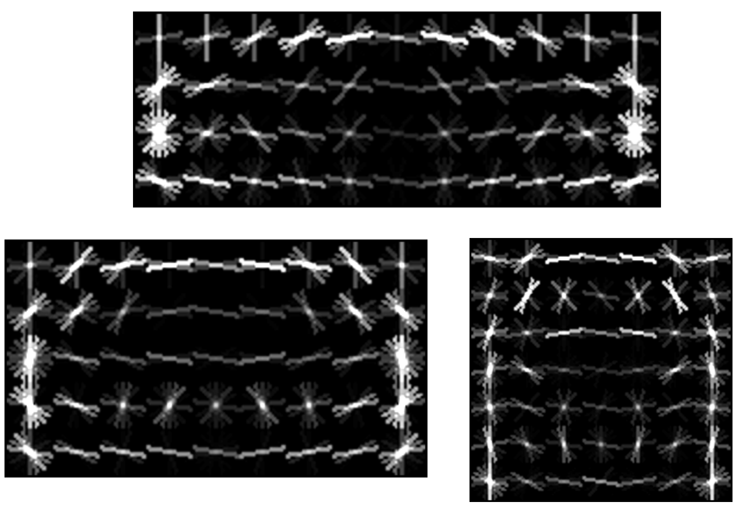
(1) Train 3 self-symmetric root filters

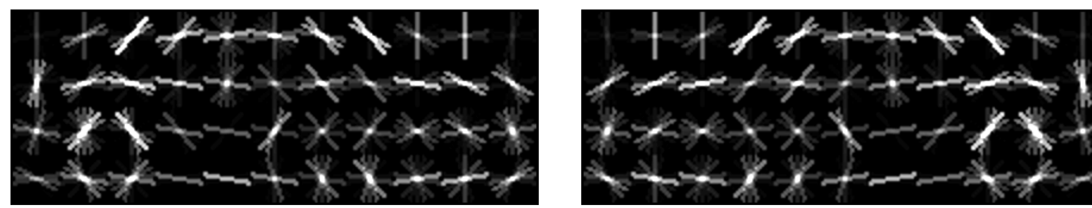
- Split positive examples using bbox aspect ratio

(2) Split each root into pair of symmetric filters

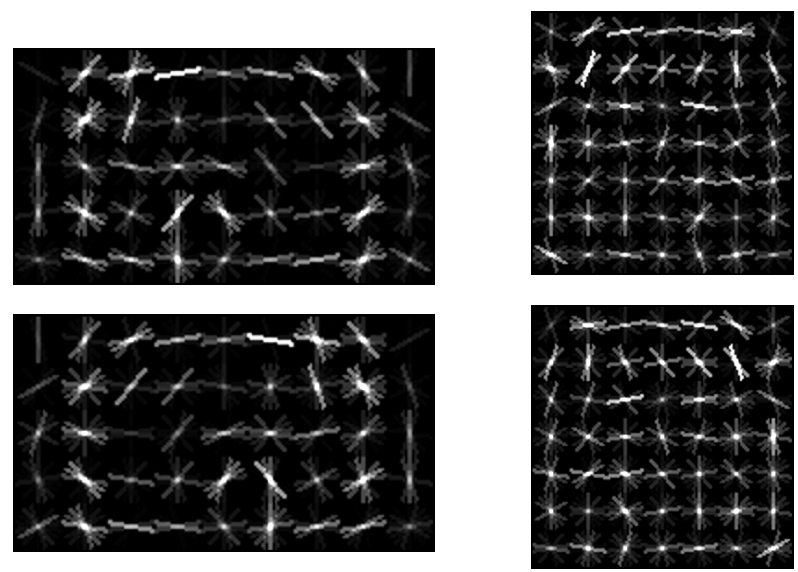
- Duplicate filters, add noise, retrain with latent component labels

(1)





(2)



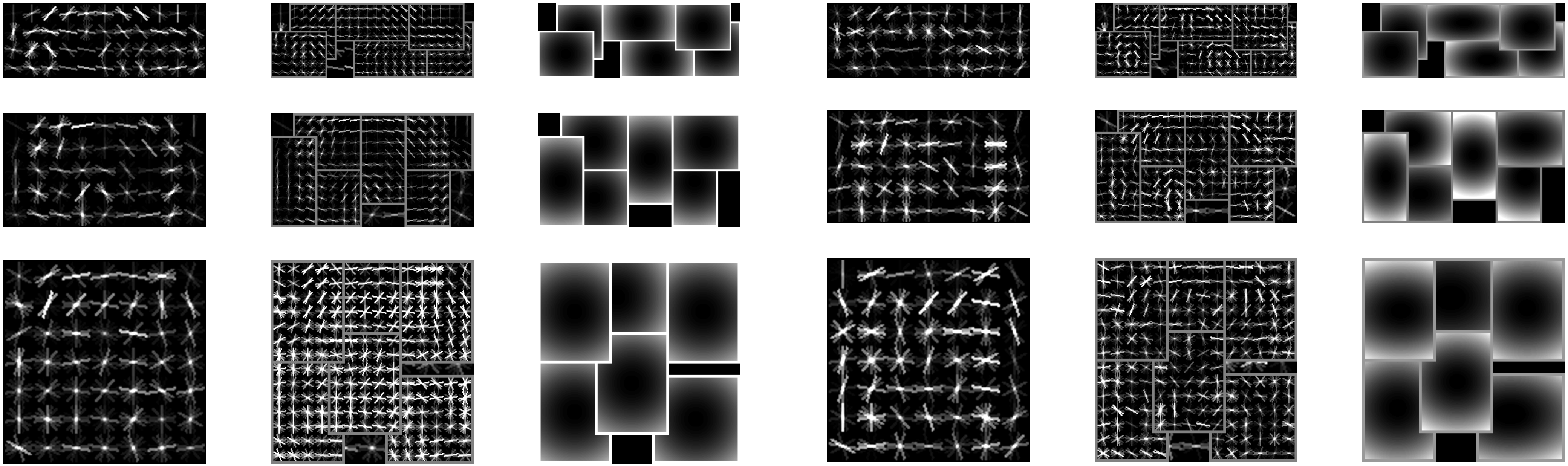
Training 6 component models

(3) Initialize parts

- Pick high energy regions in root, interpolate filter

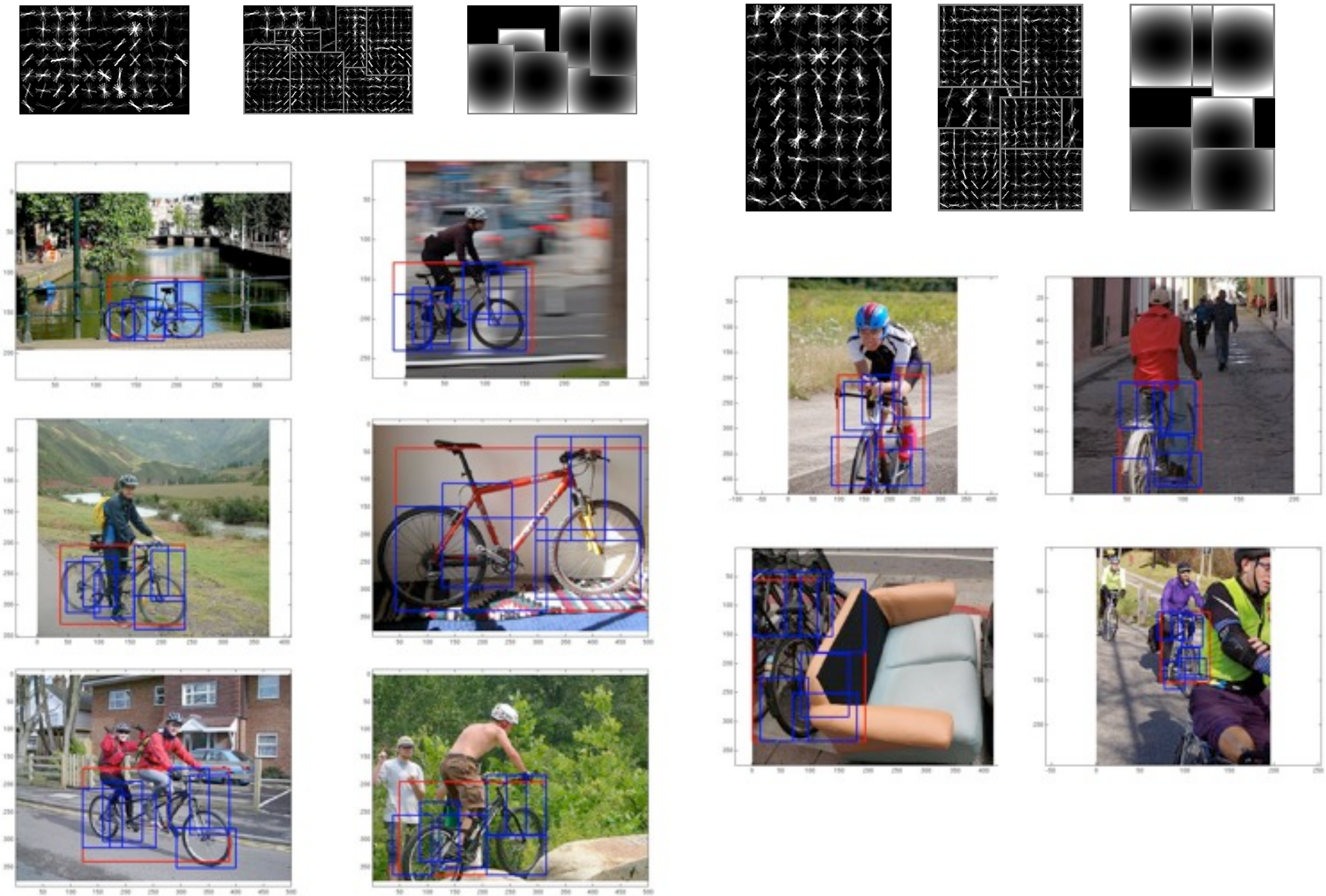
(4) Retrain parameters using model from (3)

(3) (4)

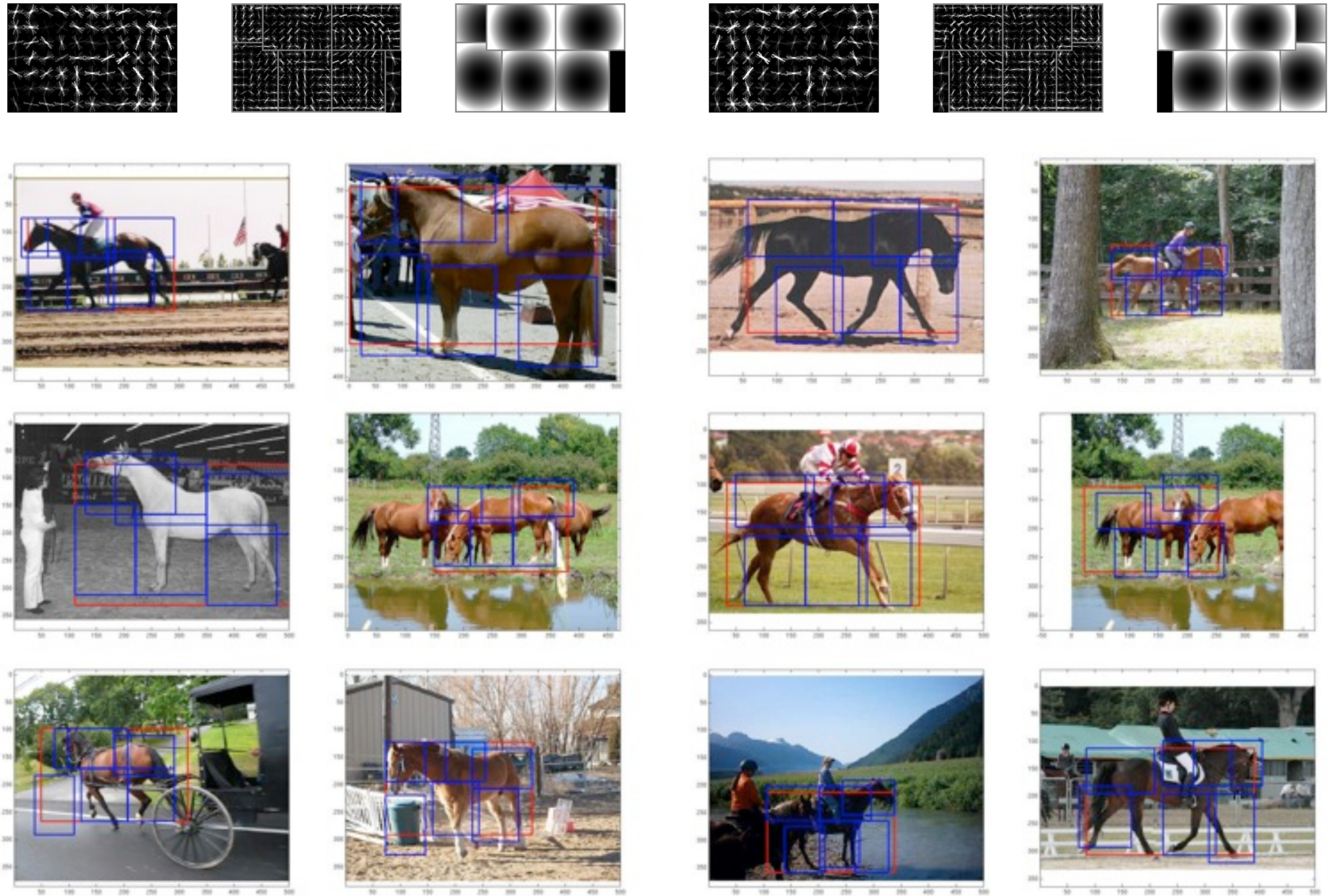


Car detections

Bicycle detections



Horse detections



Summary

- Deformable models for object detection
  - Fast matching algorithms
  - Learning from weakly-labeled data
- Current and future work:
  - Visual grammars
  - AO* search (coarse-to-fine)
  - Non-linear models