Object Detection II

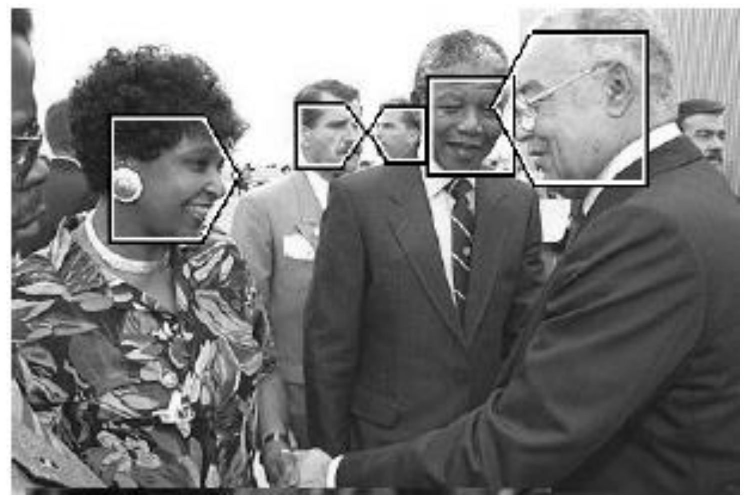
COS 429

Princeton University

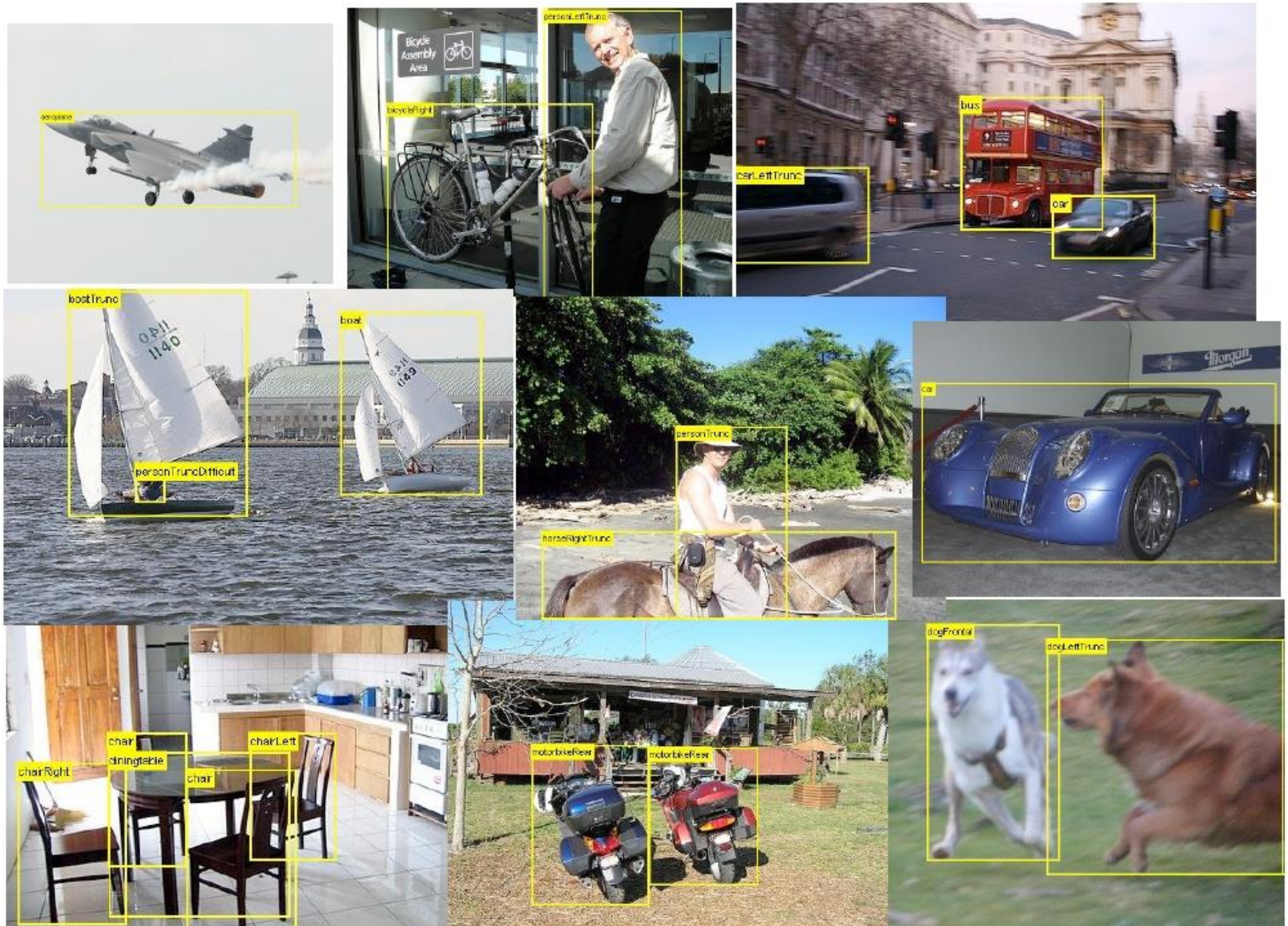
Object detection

Given an image, find all instances of a basic object category (e.g., car, face, etc.)

- Report the object locations (e.g., bounding boxes) or report that there is none



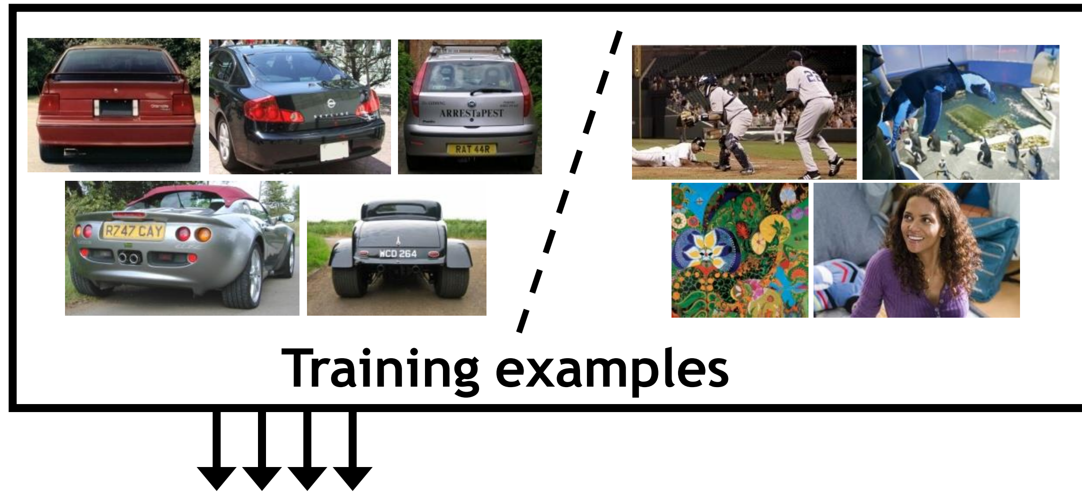
Goal: single method, many object classes

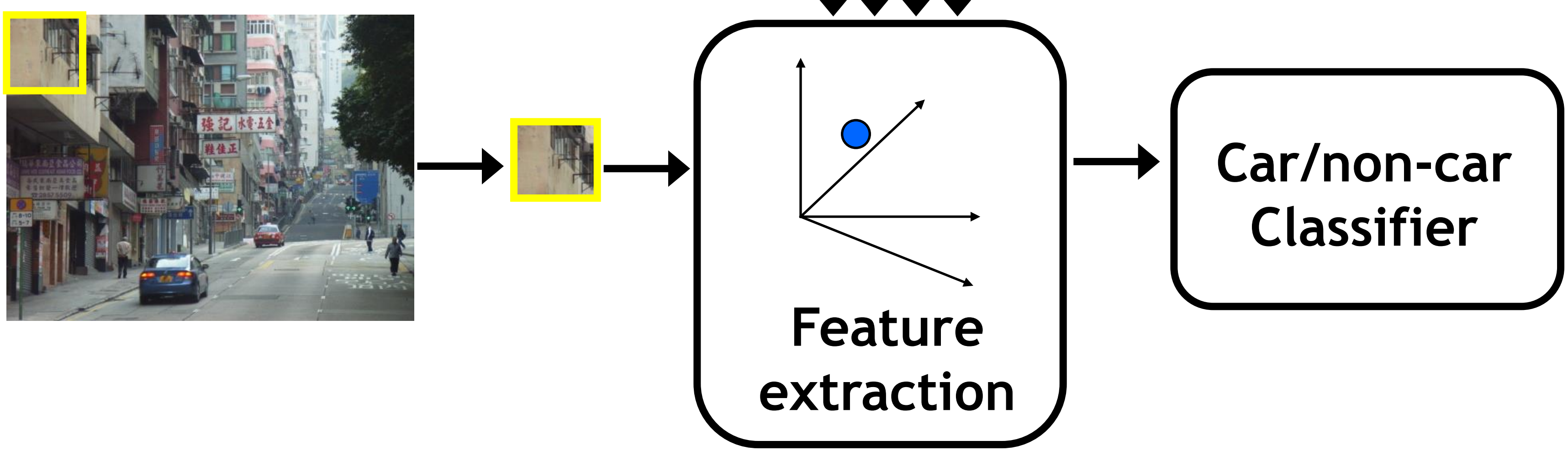


Last time: sliding window detection

Training:
1. Obtain training data
2. Define features
3. Define classifier

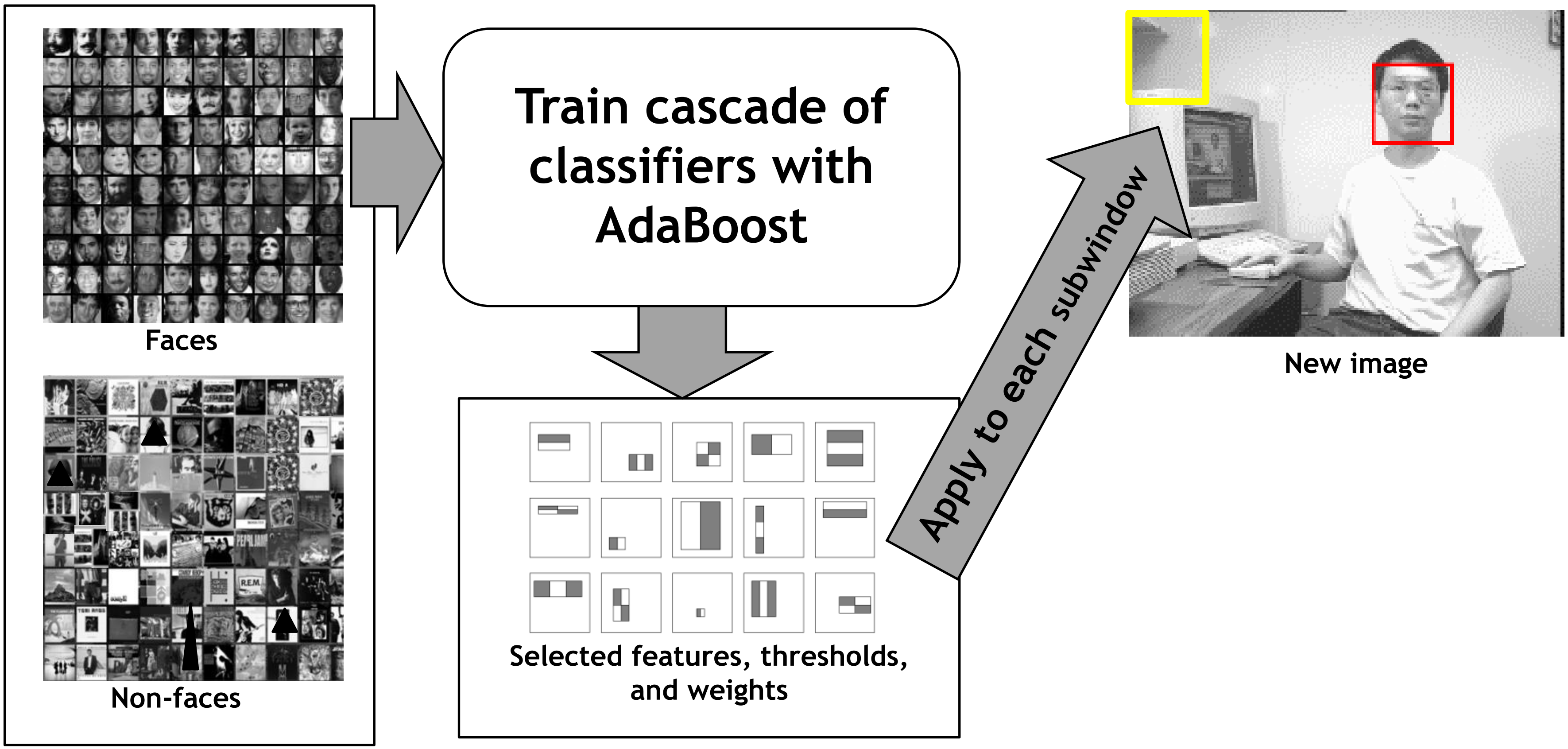
Given new image:
1. Slide window
2. Score by classifier





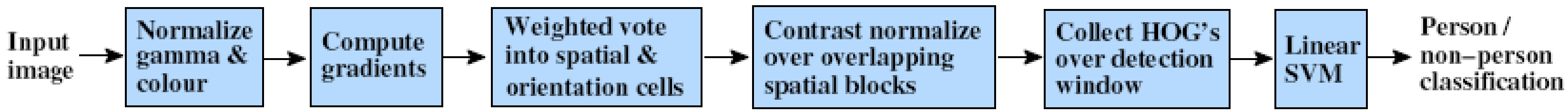
Kristen Grauman

Last time: Viola & Jones



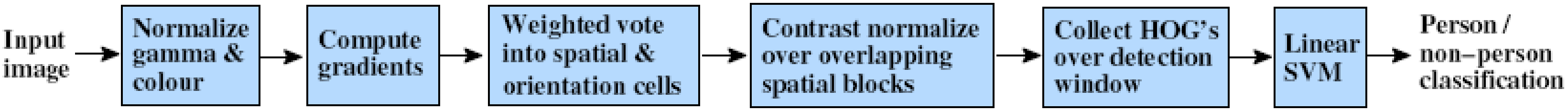
Kristen Grauman

Dalal & Triggs

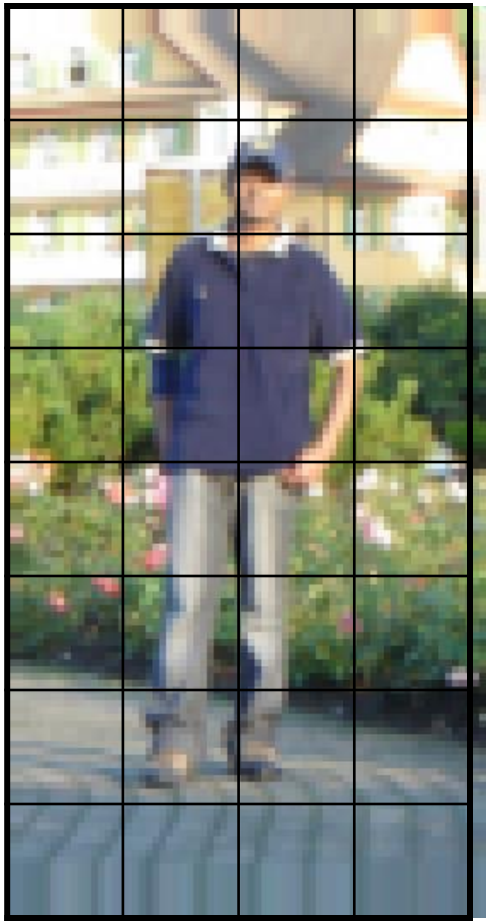


Similar in concept to Viola & Jones,
but different features (HOG),
different classifier (SVM),
and better results

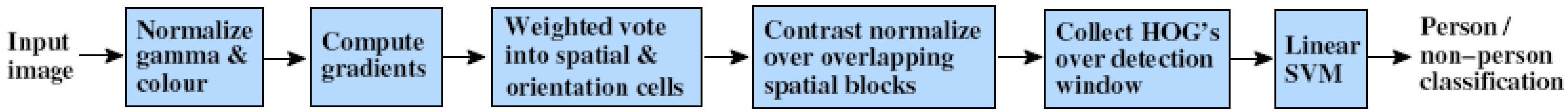
Dalal & Triggs



1) Decompose window into blocks

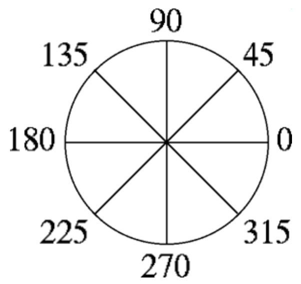


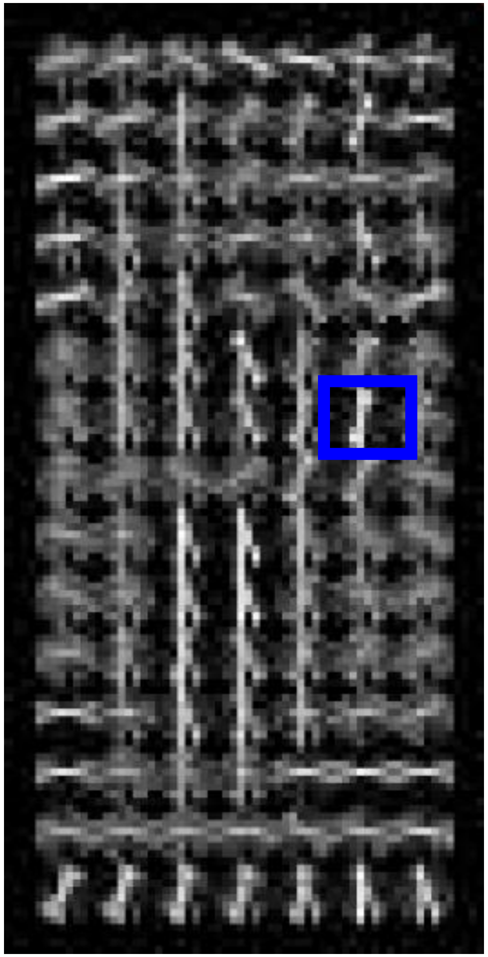
Dalal & Triggs



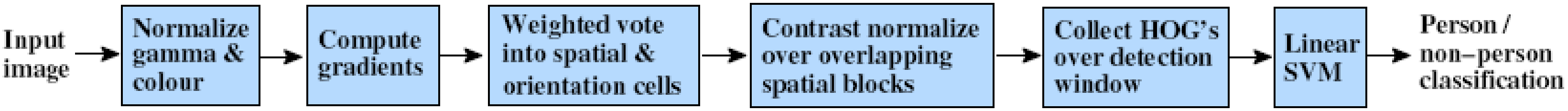
1) Decompose window into blocks
2) Compute block features

Histogram of oriented gradients (HOG)

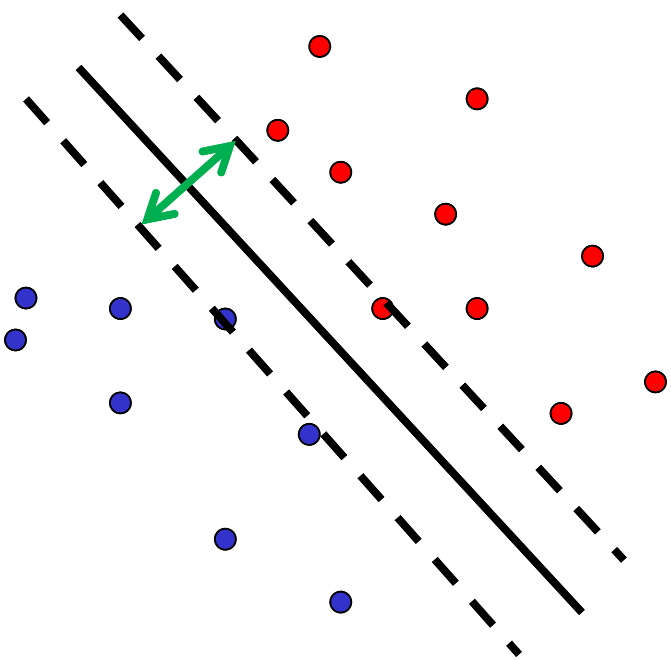


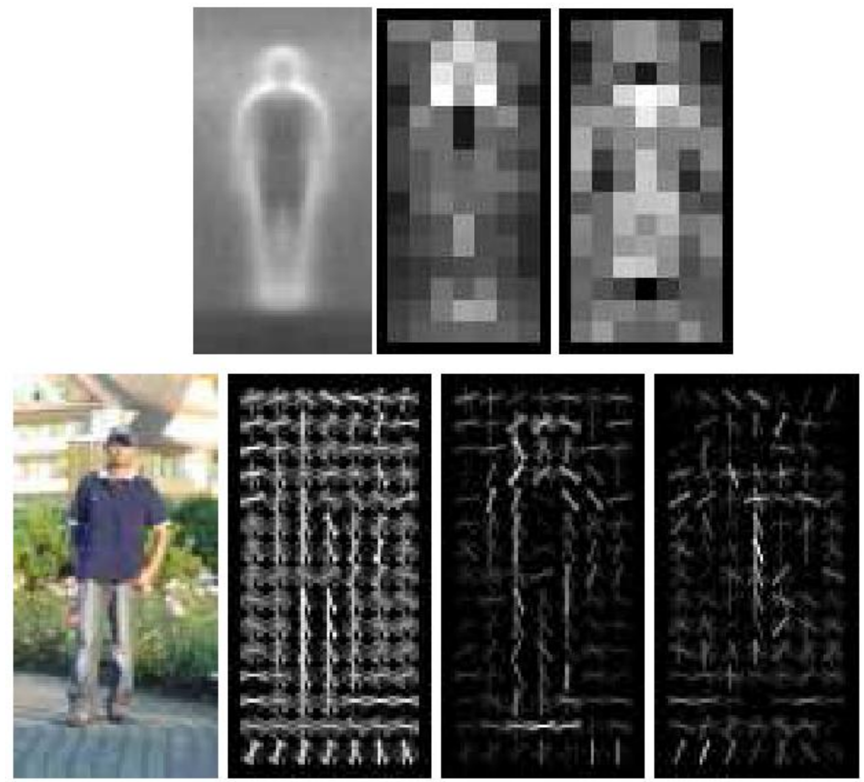


Dalal & Triggs

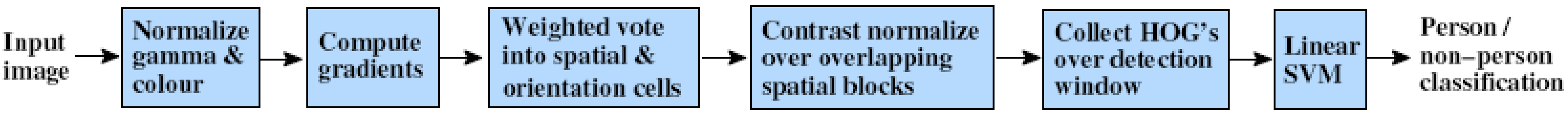


1) Decompose window into blocks
2) Compute block features
3) Classify with linear SVM

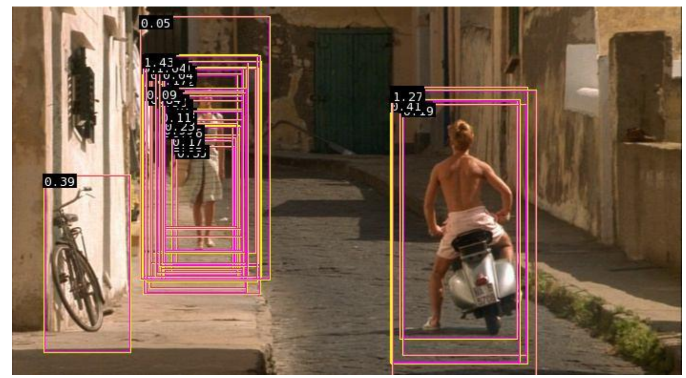


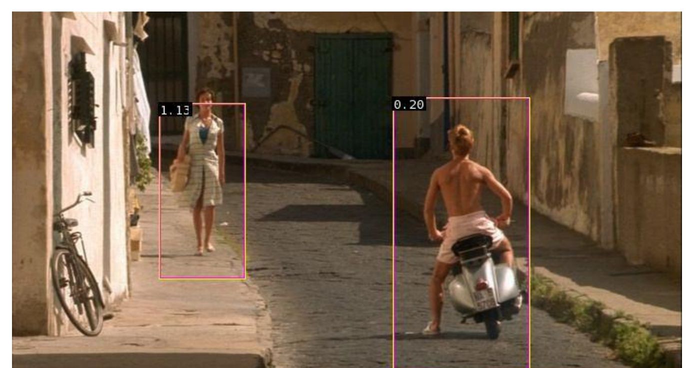


Dalal & Triggs

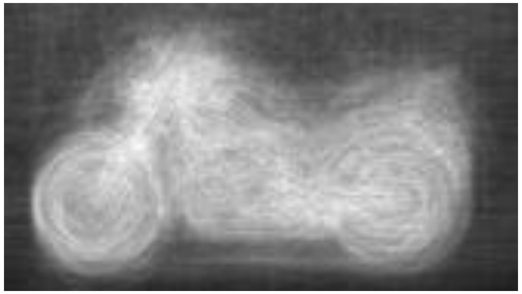


1) Decompose window into blocks
2) Compute block features
3) Classify with linear SVM
4) Extract bounding boxes

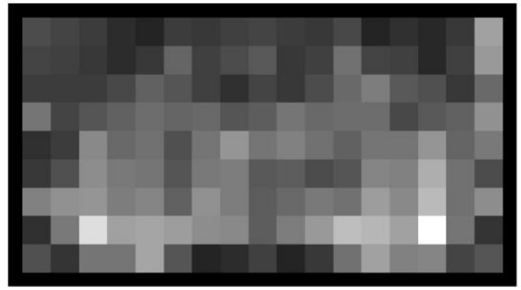




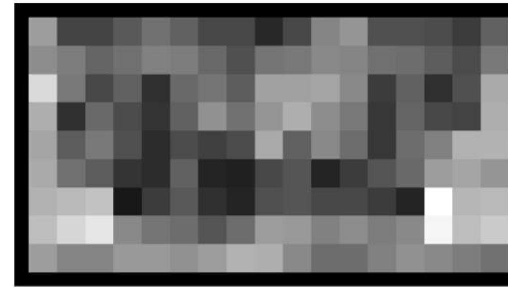
Dalal & Triggs



Average gradients

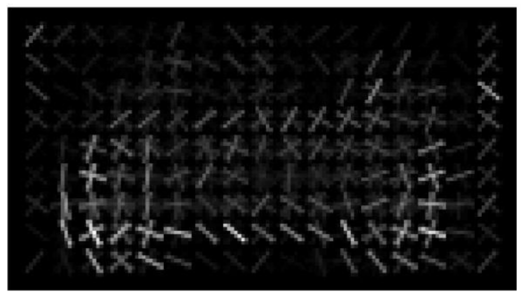


Weighted pos wts

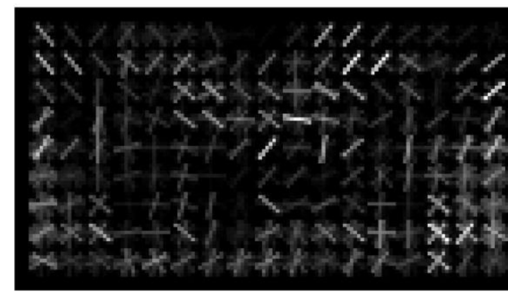


Weighted neg wts

Input window

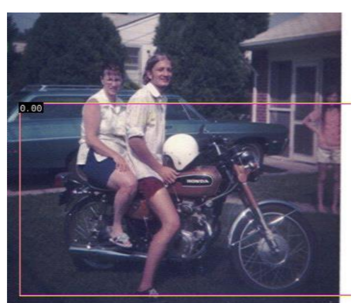


Dominant pos orientations



Dominant neg orientations

Detection Examples



Dalal & Triggs 2005

Are we done?

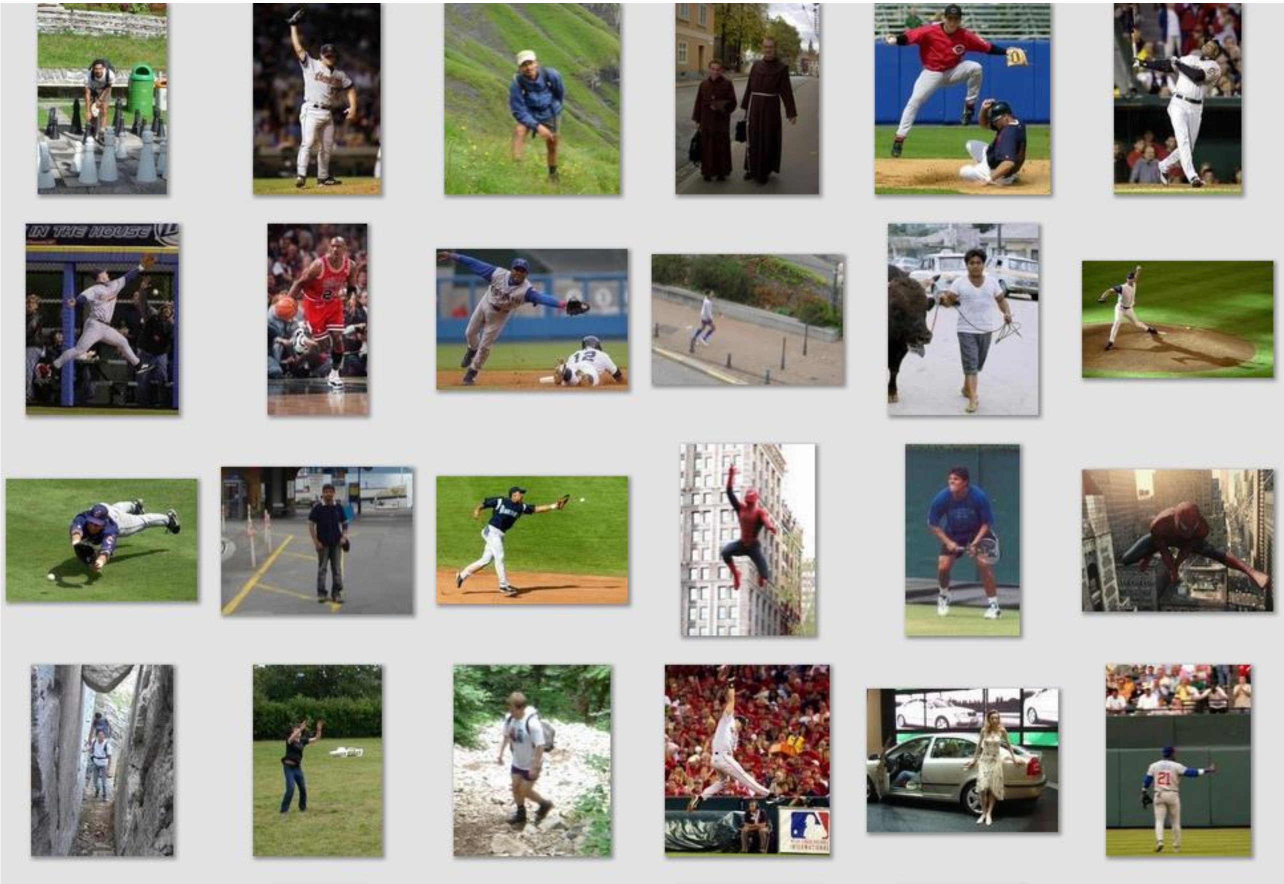
Are we done?

Single, rigid template usually not enough to represent a category

- Many objects (e.g. humans) are articulated, or have parts that can vary in configuration

- Many object categories look very different from different viewpoints, or from instance to instance

Deformable objects



Images from D. Ramanan's dataset

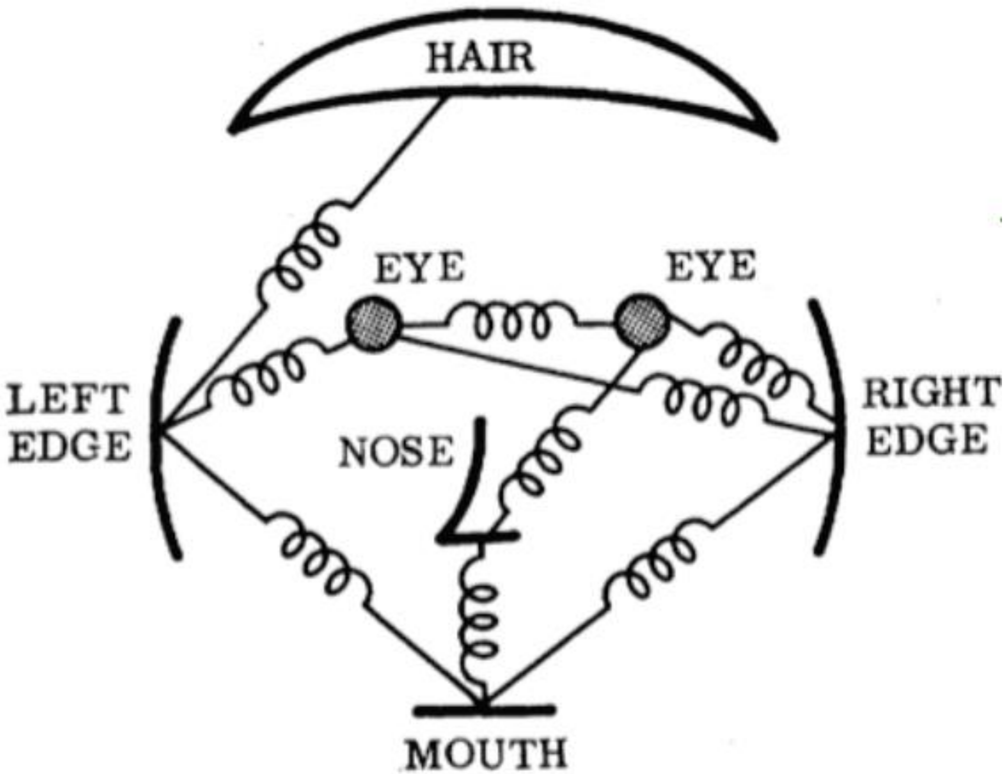
Non-rigid objects

Images from Caltech-256

Deformable object representation?

Part-based Models

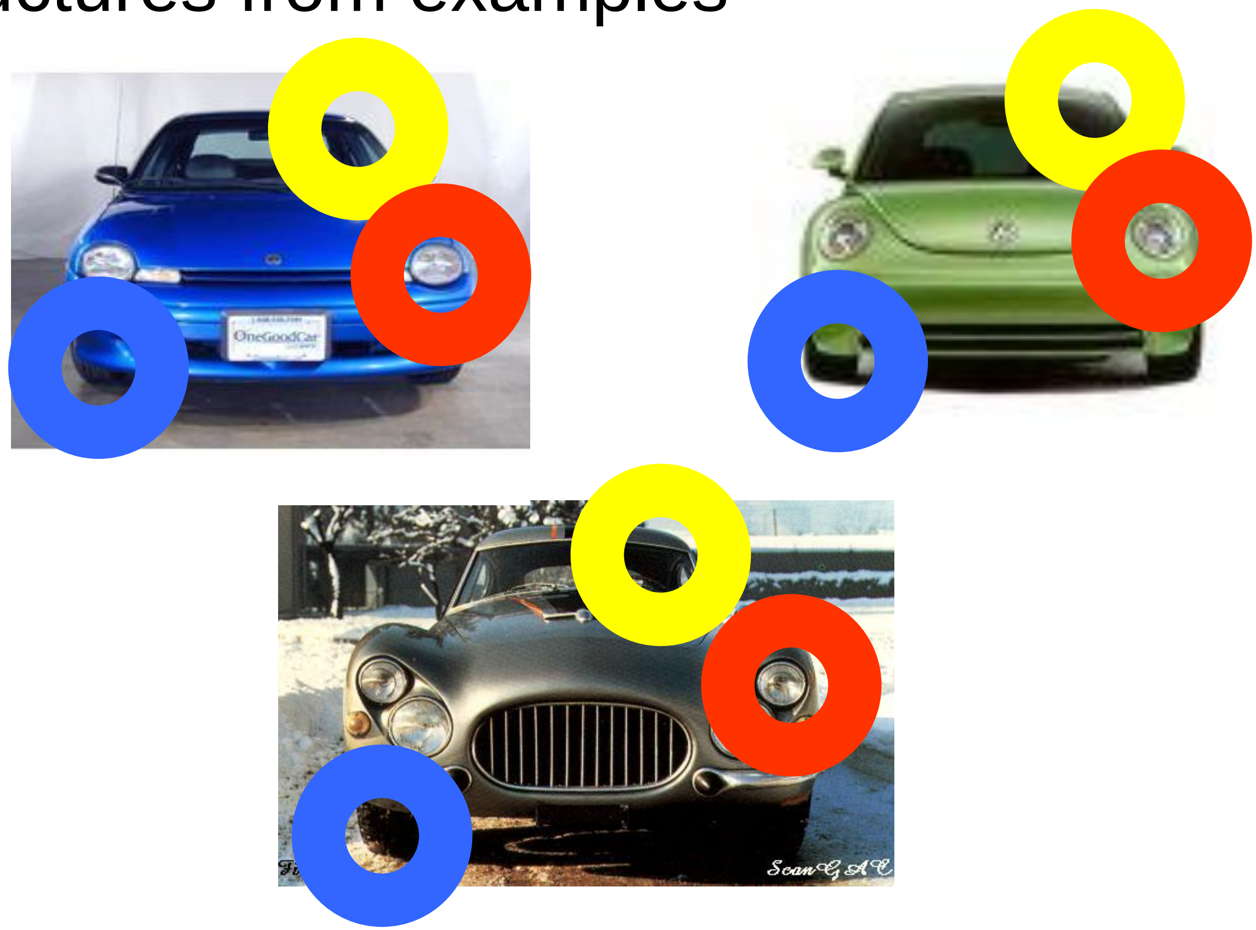
Objects are represented by features of parts and spatial relations between parts



Face model by Fischler and Elschlager '73

Generative part-based models

Training problem: find the most salient part structures from examples



R. Fergus, P. Perona and A. Zisserman,
[Object Class Recognition by Unsupervised Scale-Invariant Learning](), CVPR 2003

Generative part-based models

Recognition problem: find the most probable part layout $l_1, \ldots, l_n$ in new image

Part 1

Part 3

Part 2

R. Fergus, P. Perona and A. Zisserman,
[Object Class Recognition by Unsupervised Scale-Invariant Learning](), CVPR 2003

Generative part-based models

$$P(image \mid object) = P(appearance, shape \mid object)$$

Part descriptors

Part locations

Candidate parts

Generative part-based models

$$P(image \mid object) = P(appearance, shape \mid object)$$

Candidate parts

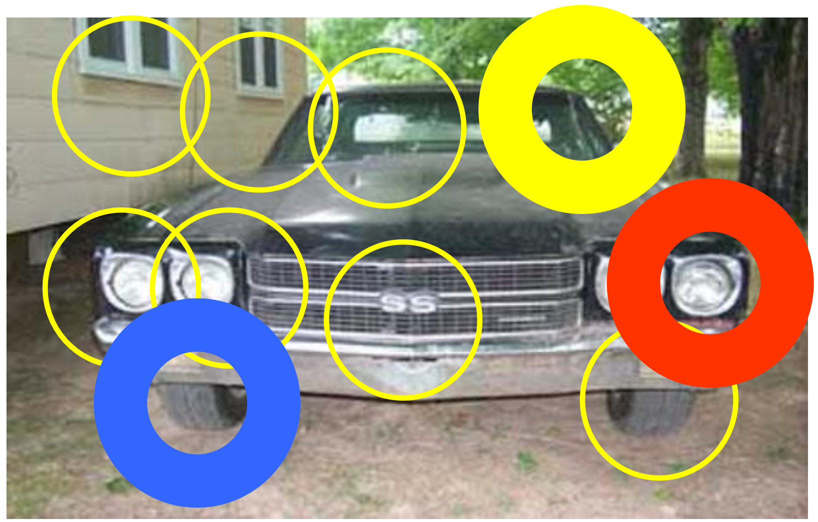
Generative part-based models

$$P(image \mid object) = P(appearance, shape \mid object)$$

$$= \max_h P(appearance \mid L, object)\, p(shape \mid L, object)\, p(L \mid object)$$

L: assignment of features to parts

Part 1



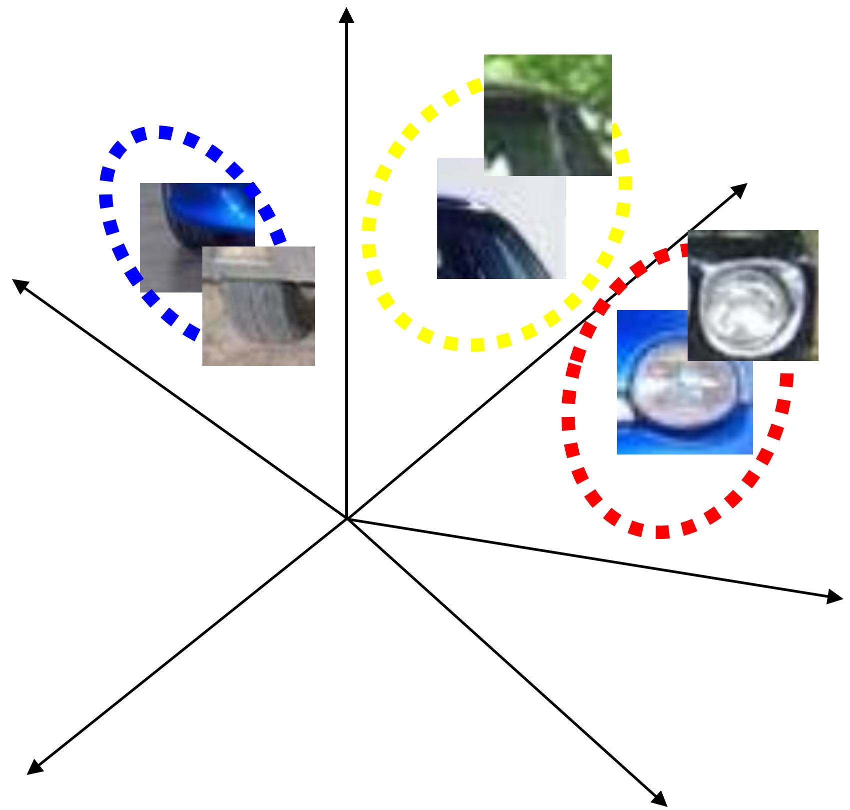
Part 3

Part 2

Generative part-based models

$$P(image \mid object) = P(appearance, shape \mid object)$$

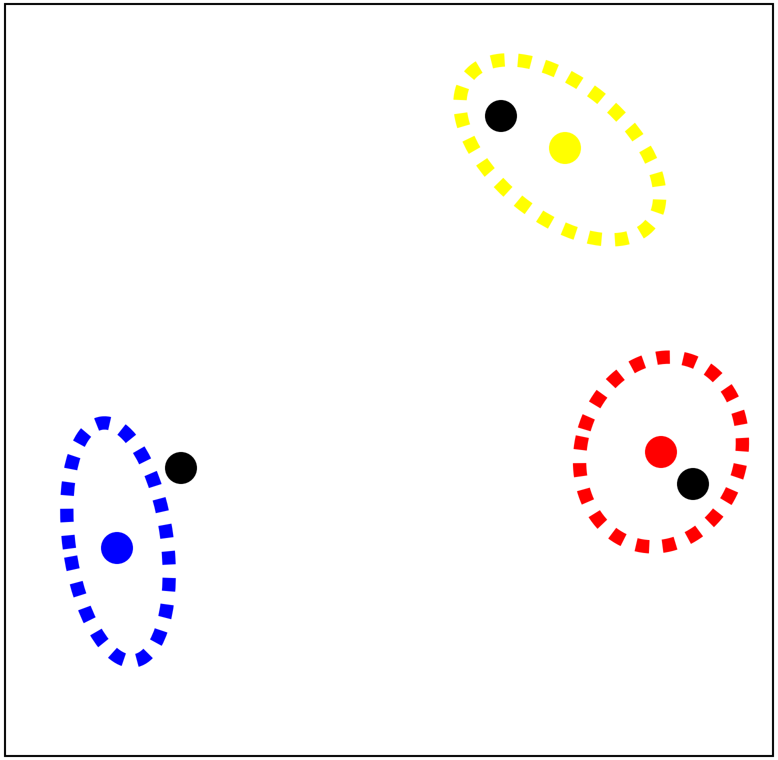
$$= \max_h \boxed{P(appearance \mid L, object)}\, p(shape \mid L, object)\, p(L \mid object)$$



Generative part-based models

$$P(image \mid object) = P(appearance, shape \mid object)$$

$$= \max_h P(appearance \mid L, object) \boxed{p(shape \mid L, object)} p(L \mid object)$$



2D image space

Distribution over joint part positions

Generative part-based models

Energy-based formulation for detection:

$$L^* = \arg\min_L \left(\sum_{i=1}^{n} m_i(l_i) + \sum_{(v_i, v_j) \in E} d_{ij}(l_i, l_j) \right)$$

$m_i(l_i)$: matching cost for part I

$d_{ij}(l_i,l_j)$: deformation cost for connected parts

(v_i,v_j): connection between part i and j

Generative part-based models

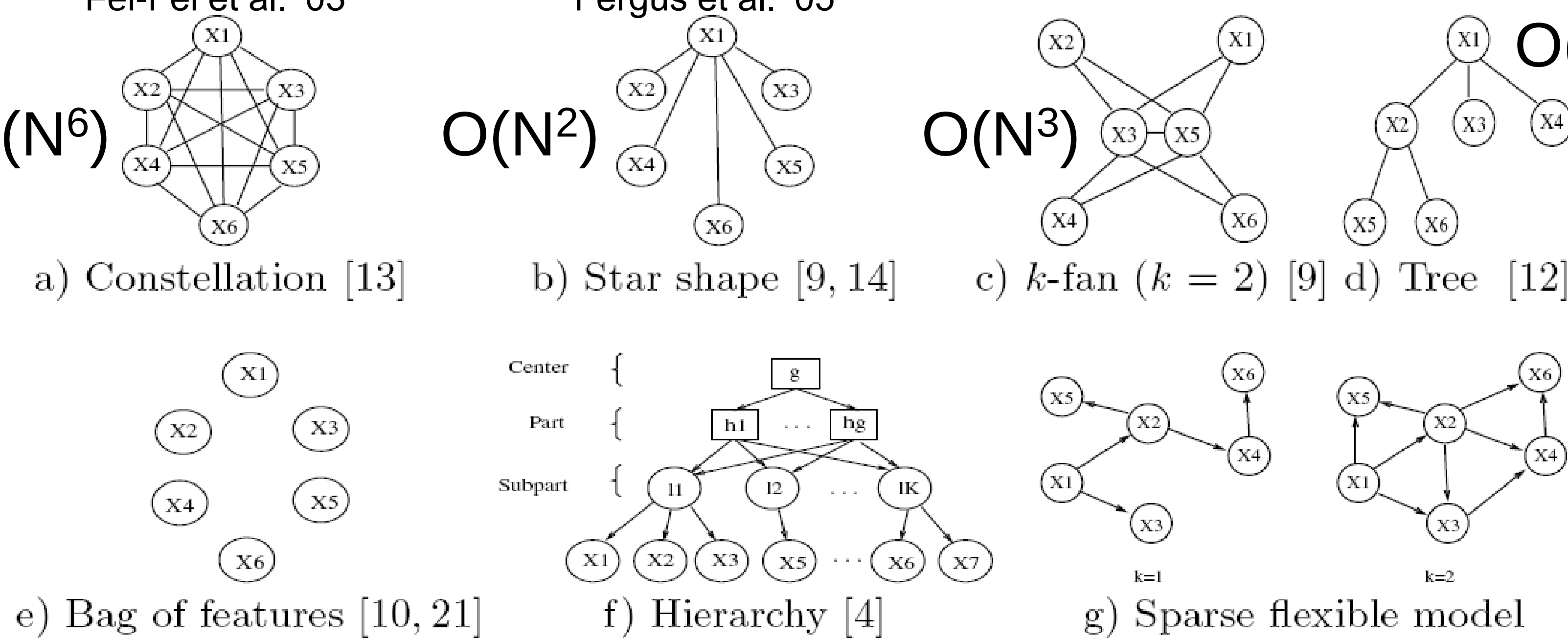
Complexity of finding minimal energy depends on topology of part model

Fergus et al. '03
Fei-Fei et al. '03
$O(N^6)$

Crandall et al. '05
Fergus et al. '05
$O(N^2)$

Crandall et al. '05
$O(N^3)$

Felzenszwalb & Huttenlocher '00
$O(N^2)$



Csurka '04
Vasconcelos '00

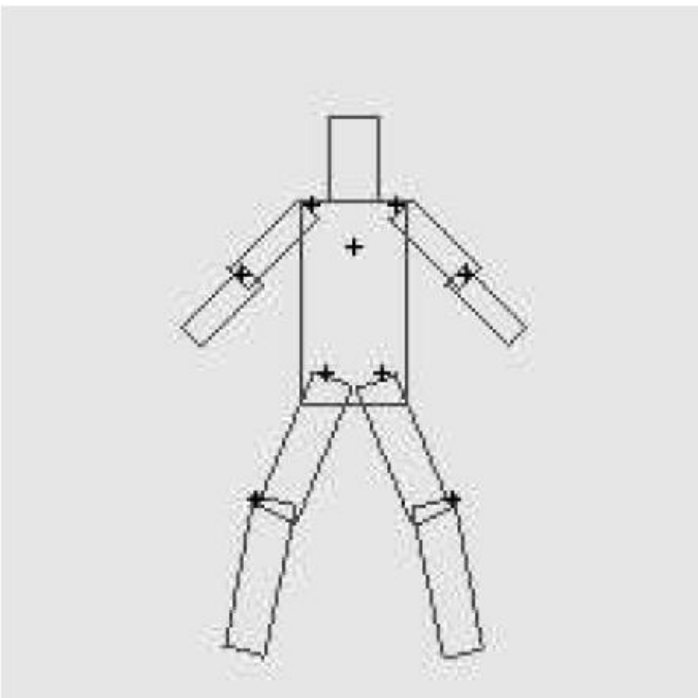
Bouchard & Triggs '05

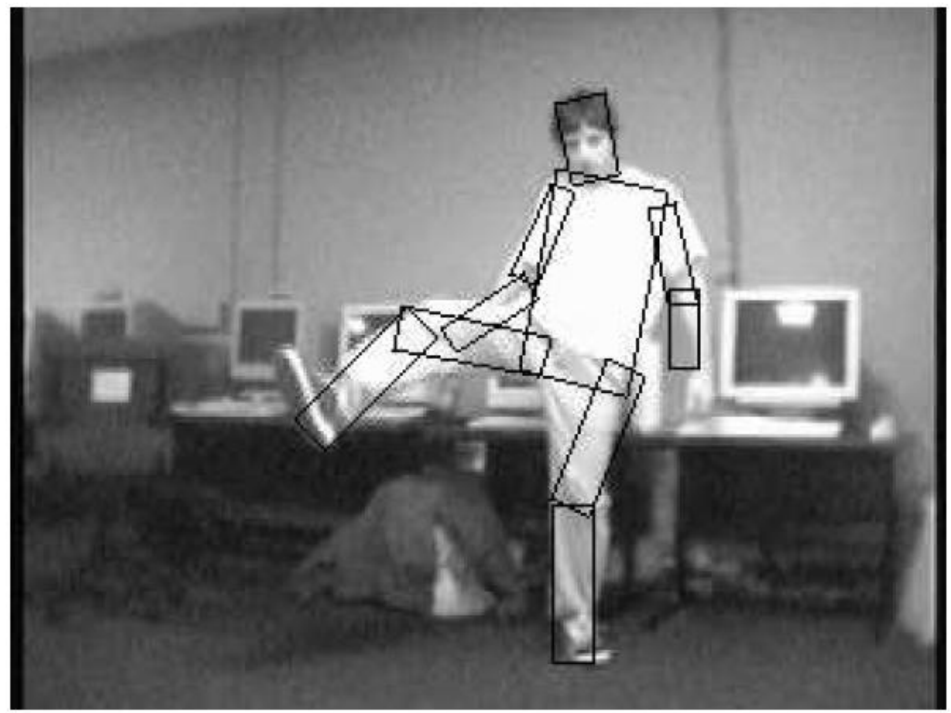
Carneiro & Lowe '06

Sparse Flexible Models of Local Features
Gustavo Carneiro and David Lowe, ECCV 2006

Generative part-based models

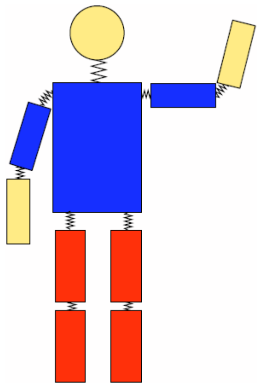
Tree-structured models can solved optimally in $O(N^2)$ with dynamic programming

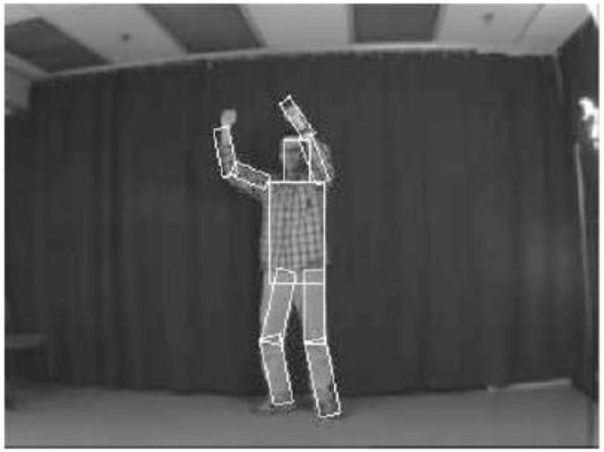




Generative part-based models

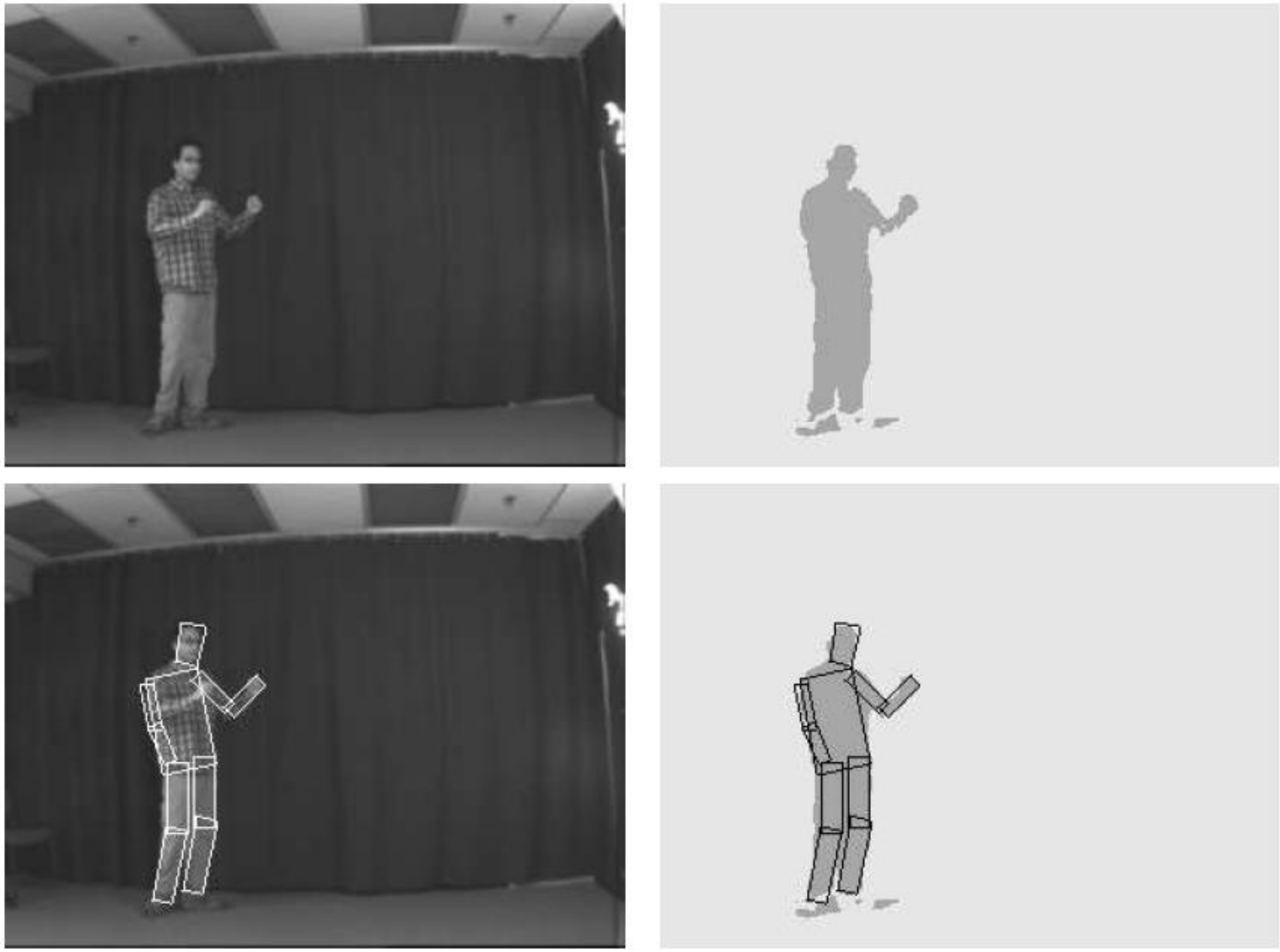
Sample result on matching human





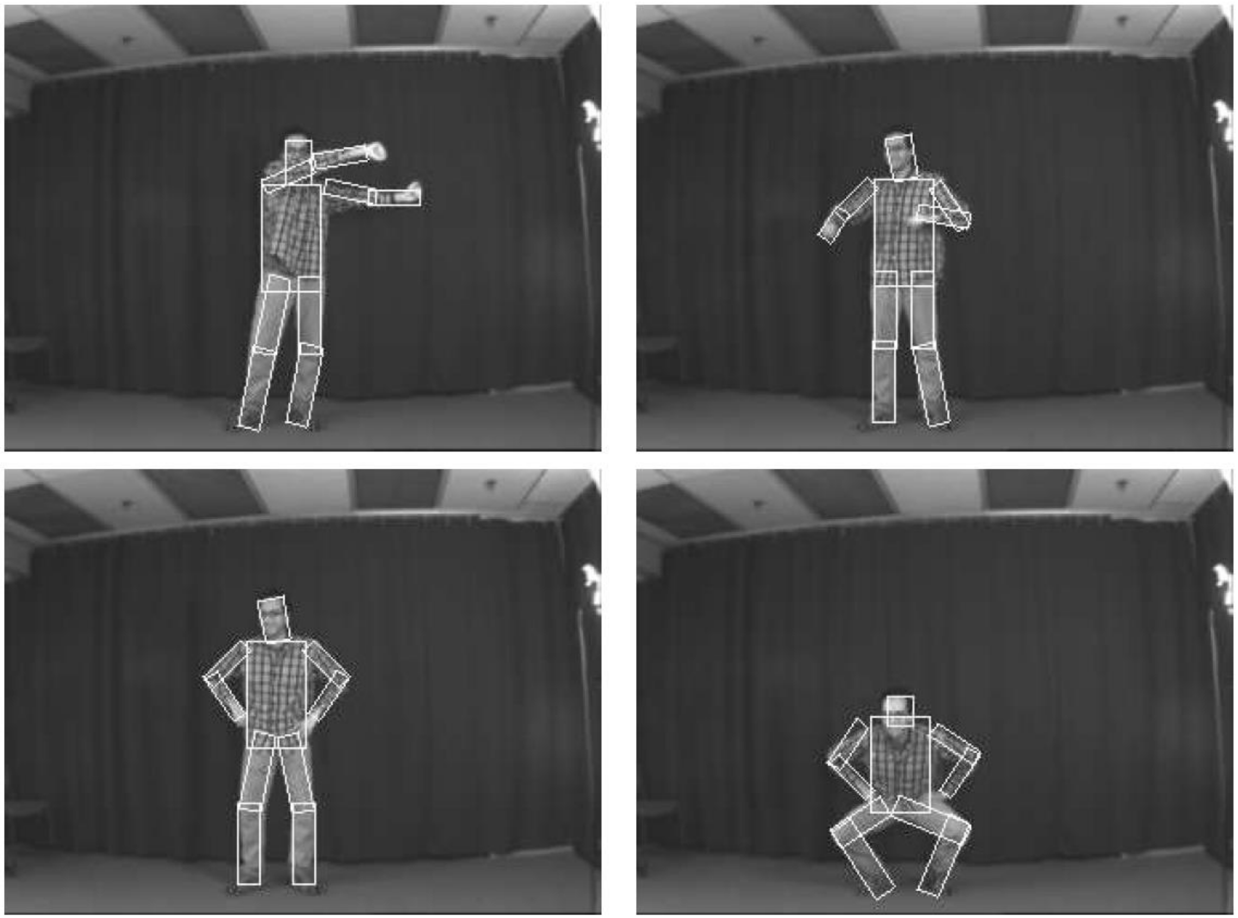
Generative part-based models

Sample result on matching human

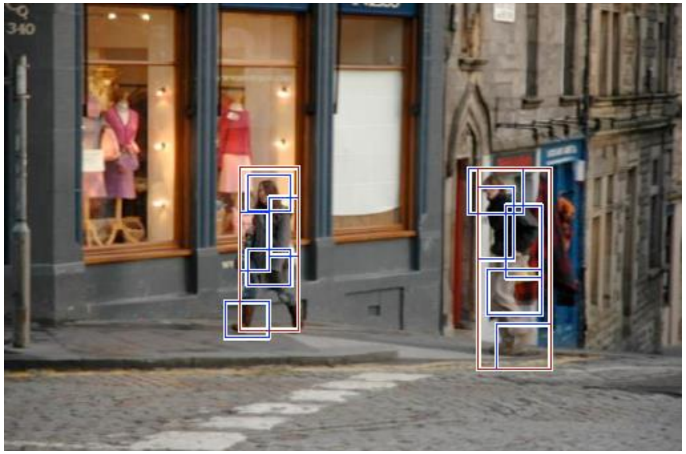


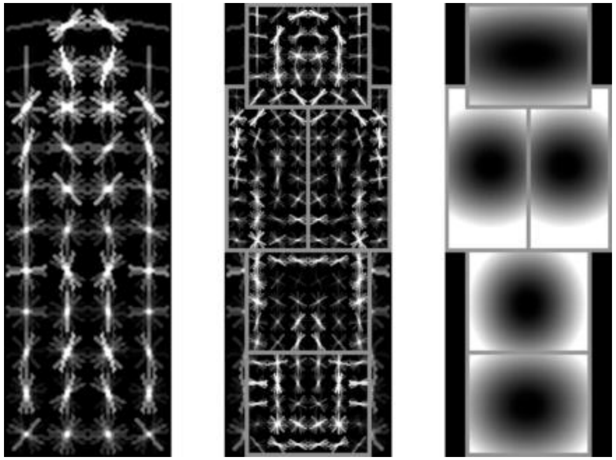
Generative part-based models

Sample result on matching human



Discriminative Part-based Models



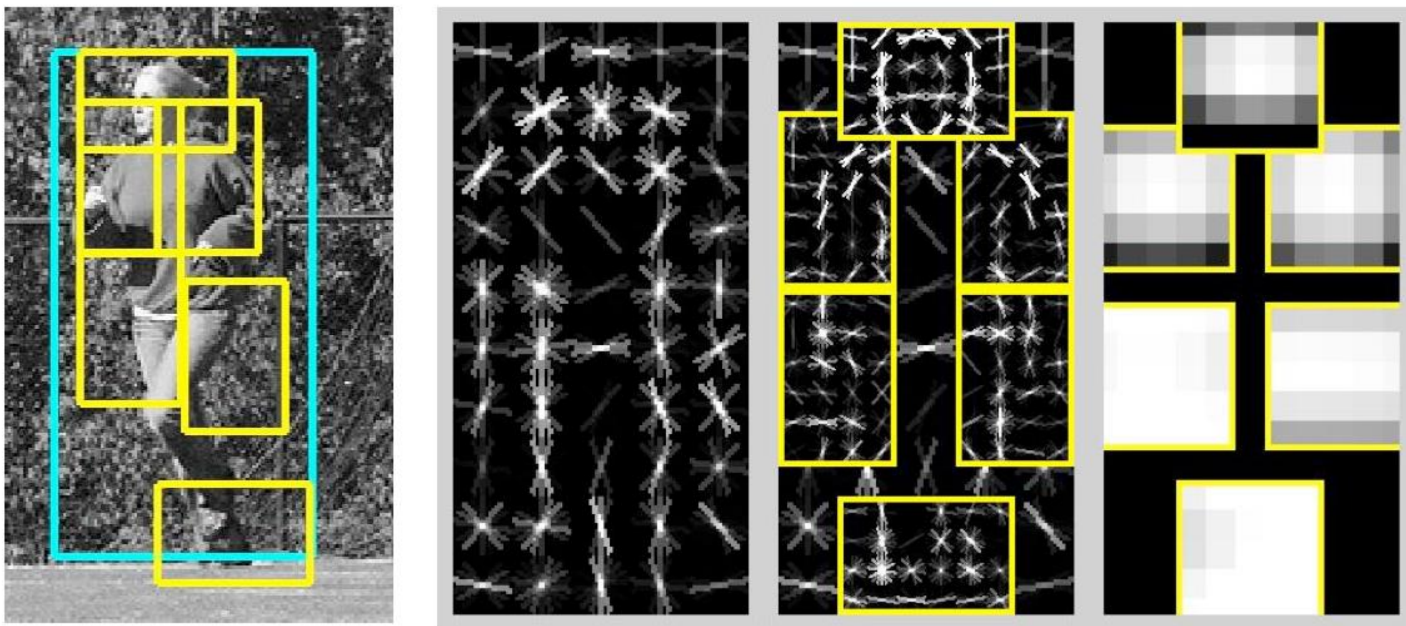


P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part Based Models, PAMI 32(9), 2010

Descriminative part-based models

Represent object as feature vector representing

- Appearance of root and parts
- Spatial relationships between root and parts

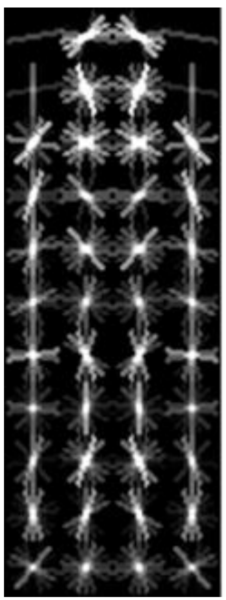


detection root filter part filters deformation models

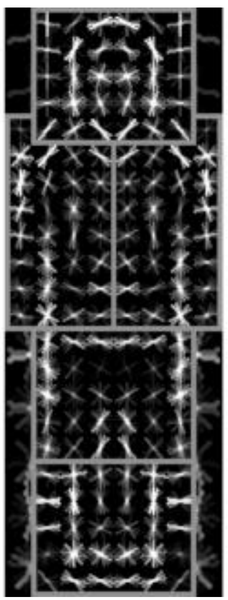
Discriminative part-based models

At detection time, consider object hypotheses (windows) at multiple shifts and scales

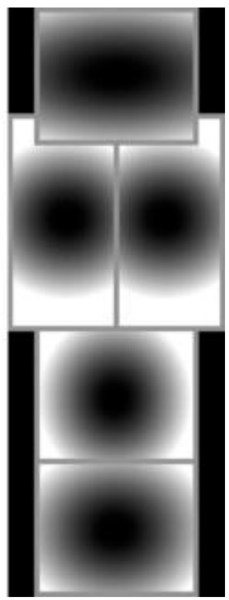
Root filter



Part filters



Deformation weights



Scoring an object hypothesis

The score of a hypothesis is the sum of filter scores minus the sum of deformation costs

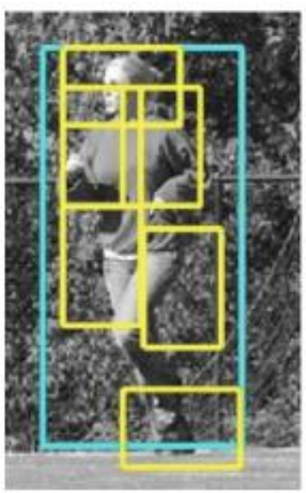
Subwindow features

Displacements

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

Filters

Deformation weights



Scoring an object hypothesis

The score of a hypothesis is the sum of filter scores minus the sum of deformation costs

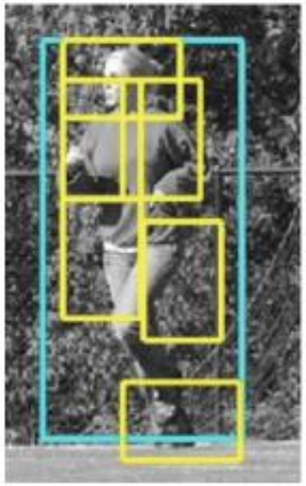
Subwindow features

Displacements

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

Filters

Deformation weights



$$score(z) = w \cdot H(z)$$

Concatenation of filter and deformation weights

Concatenation of subwindow features and displacements

Detection

Define the score of each root filter location as the score given the best part placements:

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

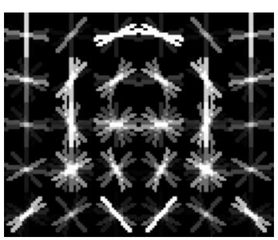
Detection

Define the score of each root filter location as the score given the best part placements:

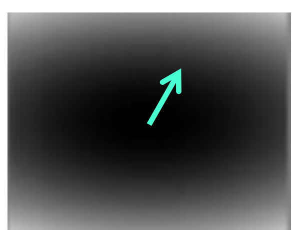
$$score(p_0) = \max_{p_1,\ldots,p_n} score(p_0,\ldots,p_n)$$

- Efficient computation: *generalized distance transforms*
  - For each "default" part location, find the score of the "best" displacement

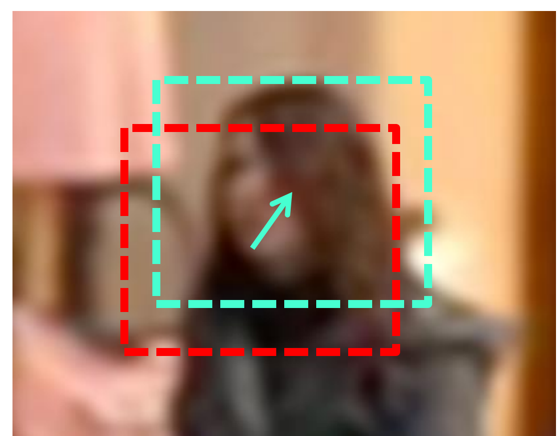
$$R_i(x,y) = \max_{dx,dy}\Big(F_i \cdot H(x+dx, y+dy) - D_i \cdot (dx, dy, dx^2, dy^2)\Big)$$



Head filter



Deformation cost



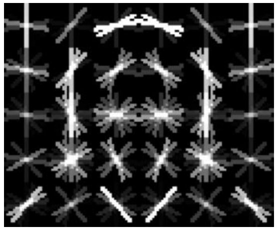
Detection

Define the score of each root filter location as the score given the best part placements:

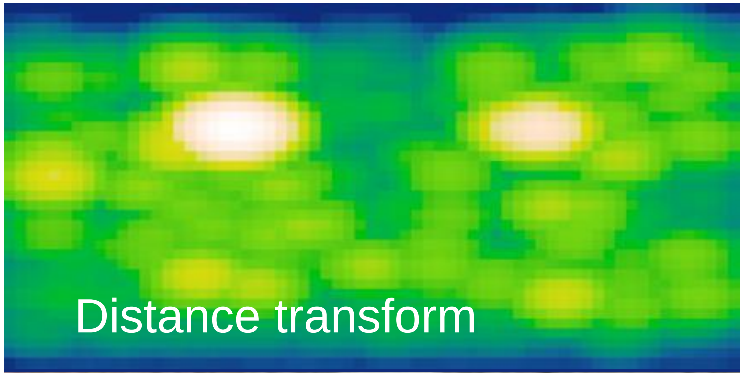
$$score(p_0) = \max_{p_1,\ldots,p_n} score(p_0,\ldots,p_n)$$

- Efficient computation: *generalized distance transforms*
  - For each "default" part location, find the score of the "best" displacement

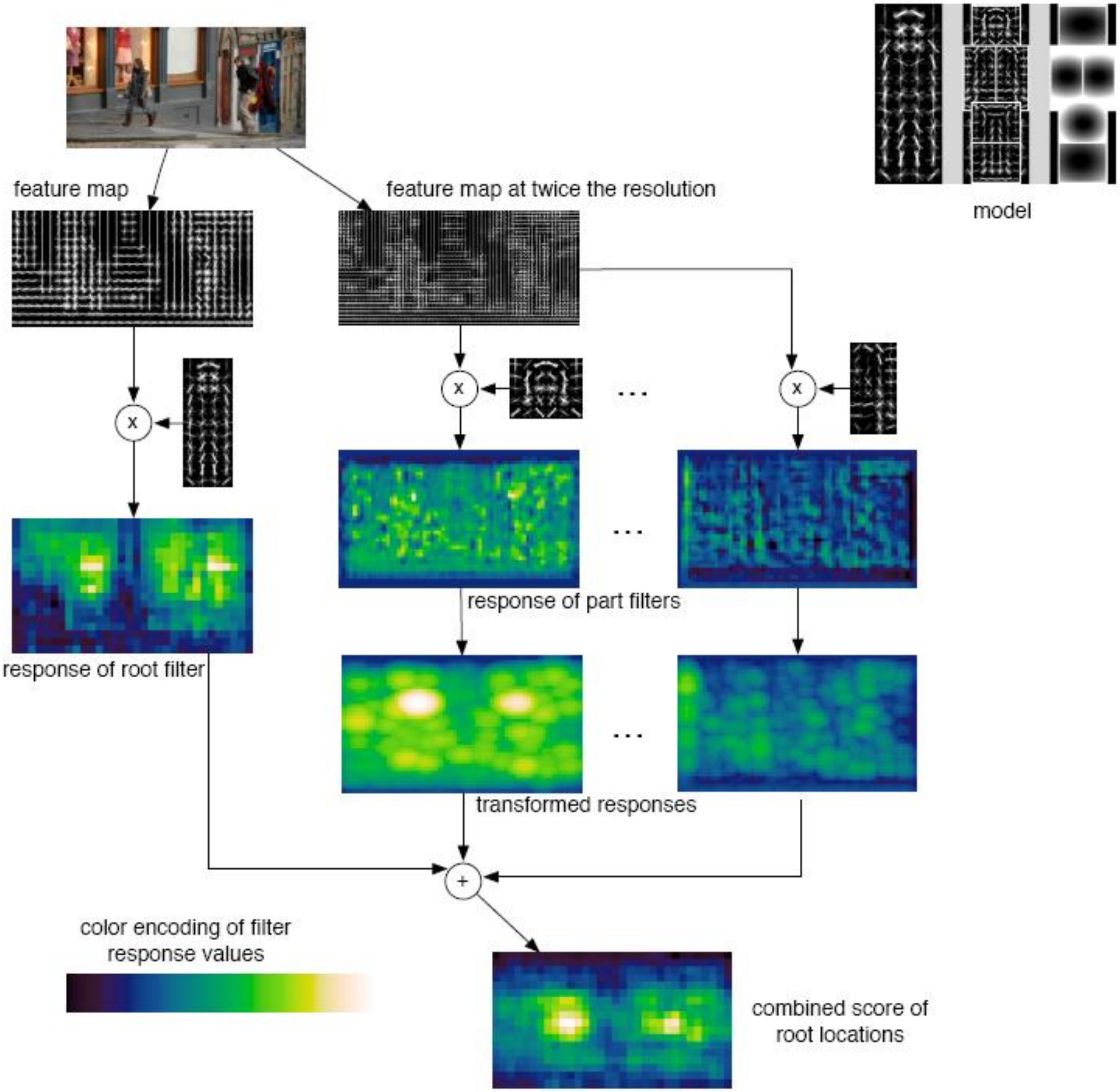
$$R_i(x,y) = \max_{dx,dy}\left(F_i \cdot H(x+dx, y+dy) - D_i \cdot (dx, dy, dx^2, dy^2)\right)$$



Head filter

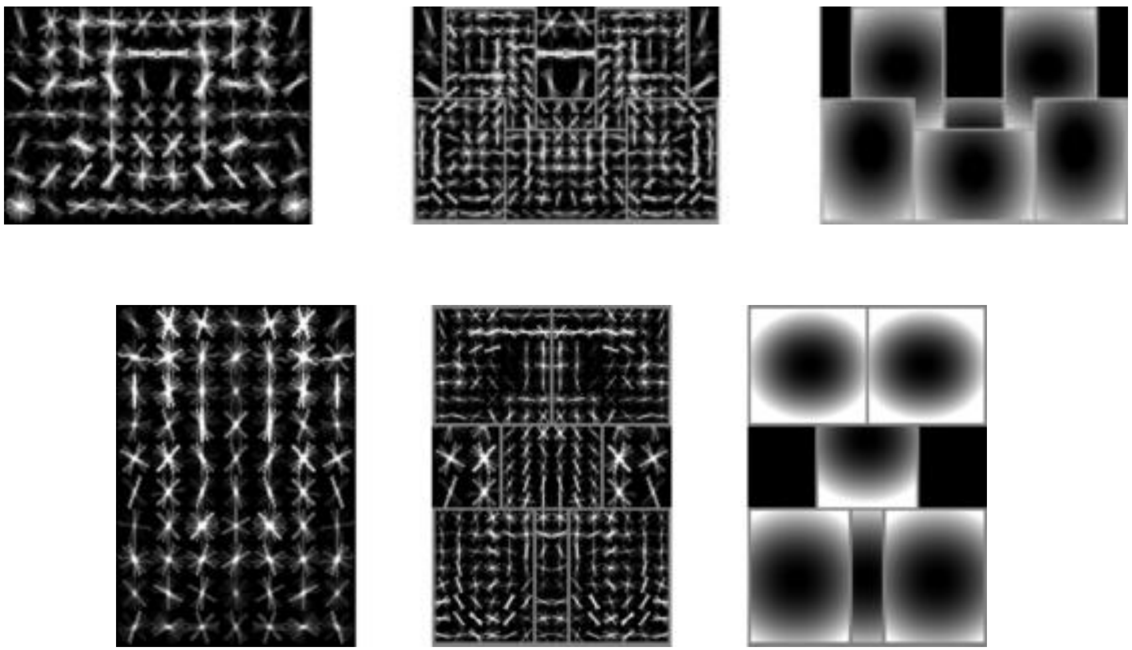


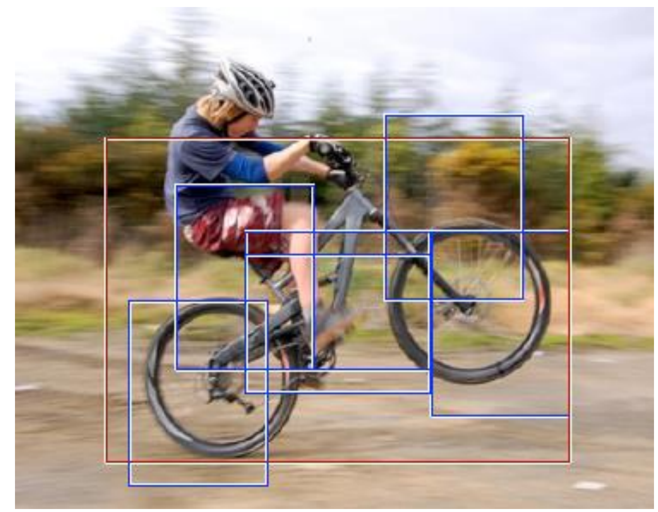
Detection



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

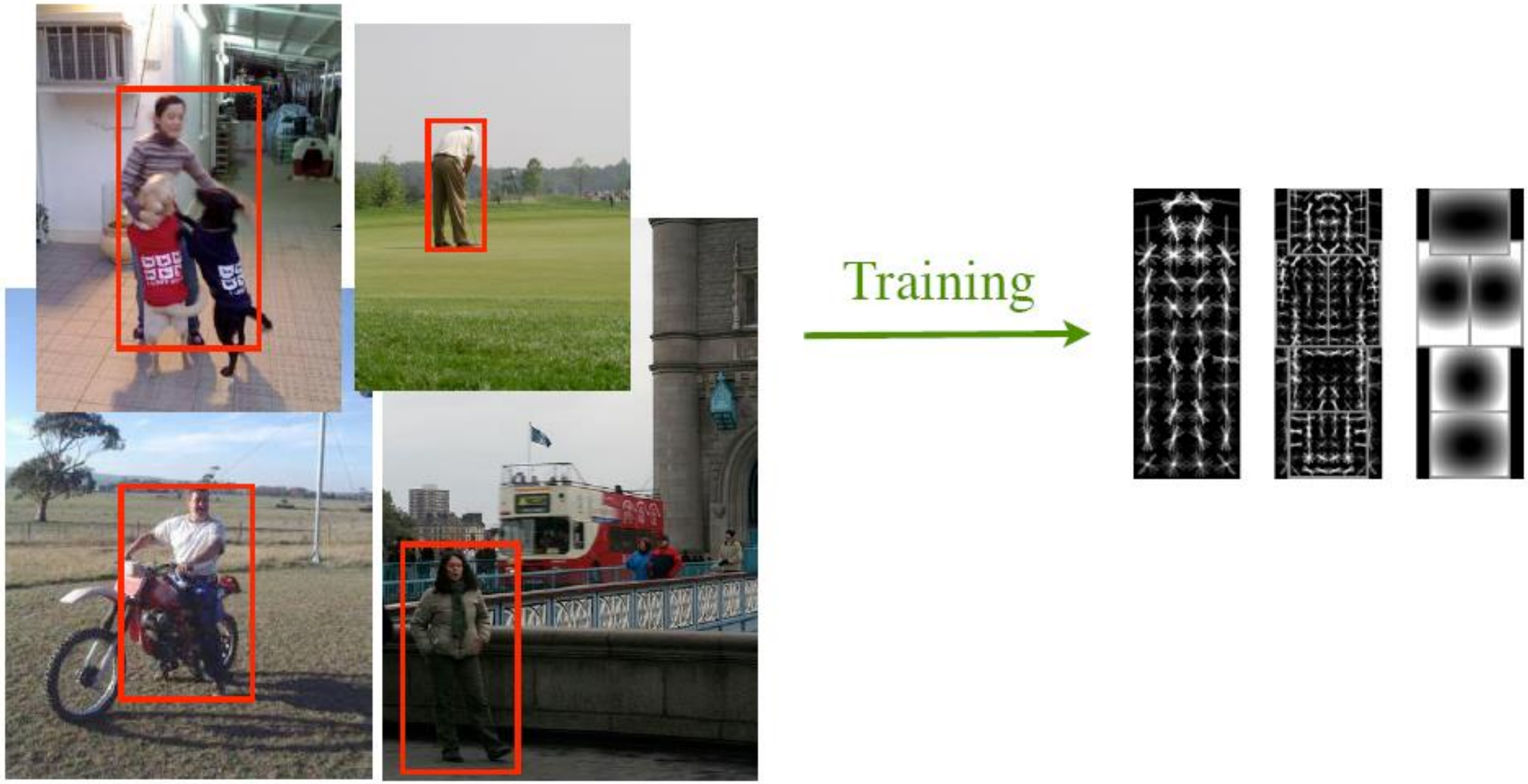
Discriminative part-based models





Training

- Training data consists of labeled bounding boxes
- Need to learn the filters and deformation parameters



Training

- Classifier has the form

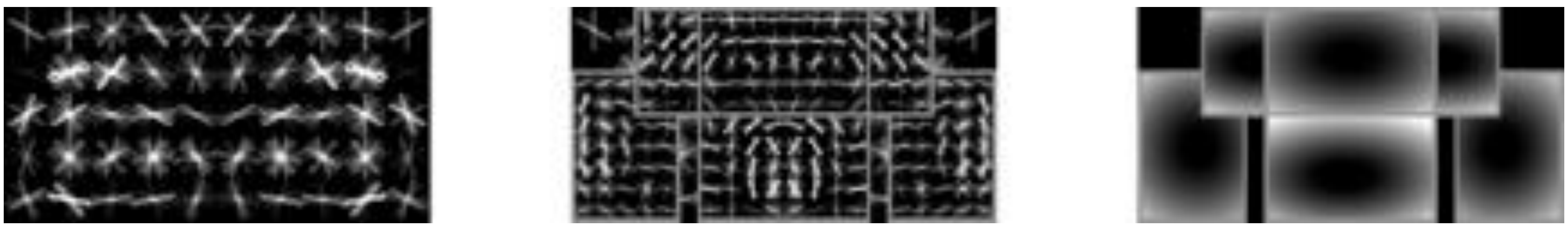
$$f(x) = \max_z w \cdot H(x, z)$$

- *w* are model parameters, *z* are *latent* hypotheses

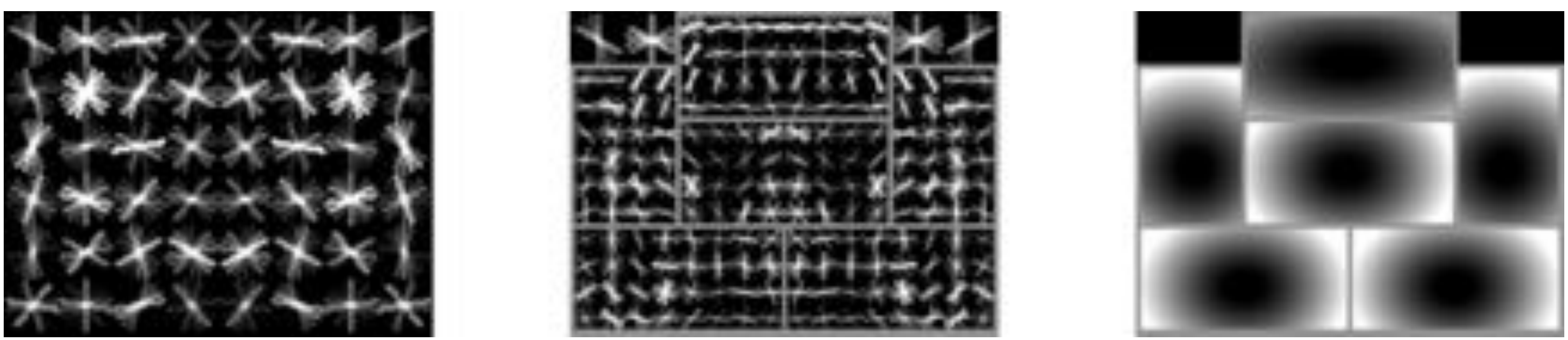
- **Latent SVM** training:
  - Initialize *w* and iterate:
    - Fix *w* and find the best *z* for each training example (detection)
    - Fix *z* and solve for *w* (standard SVM training)

Car model

Component 1

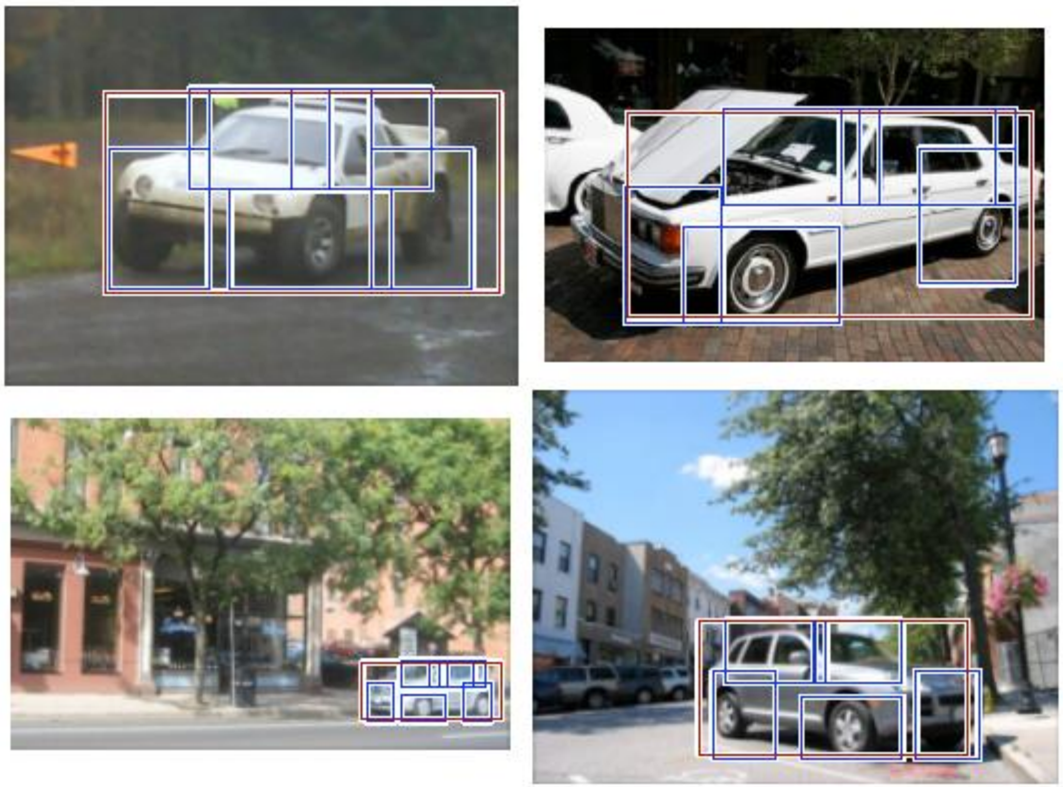


Component 2

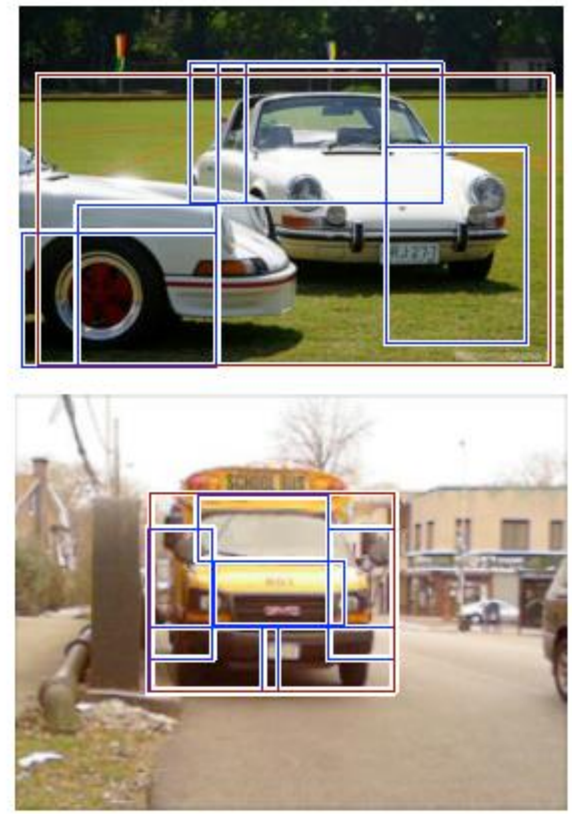


Car detections

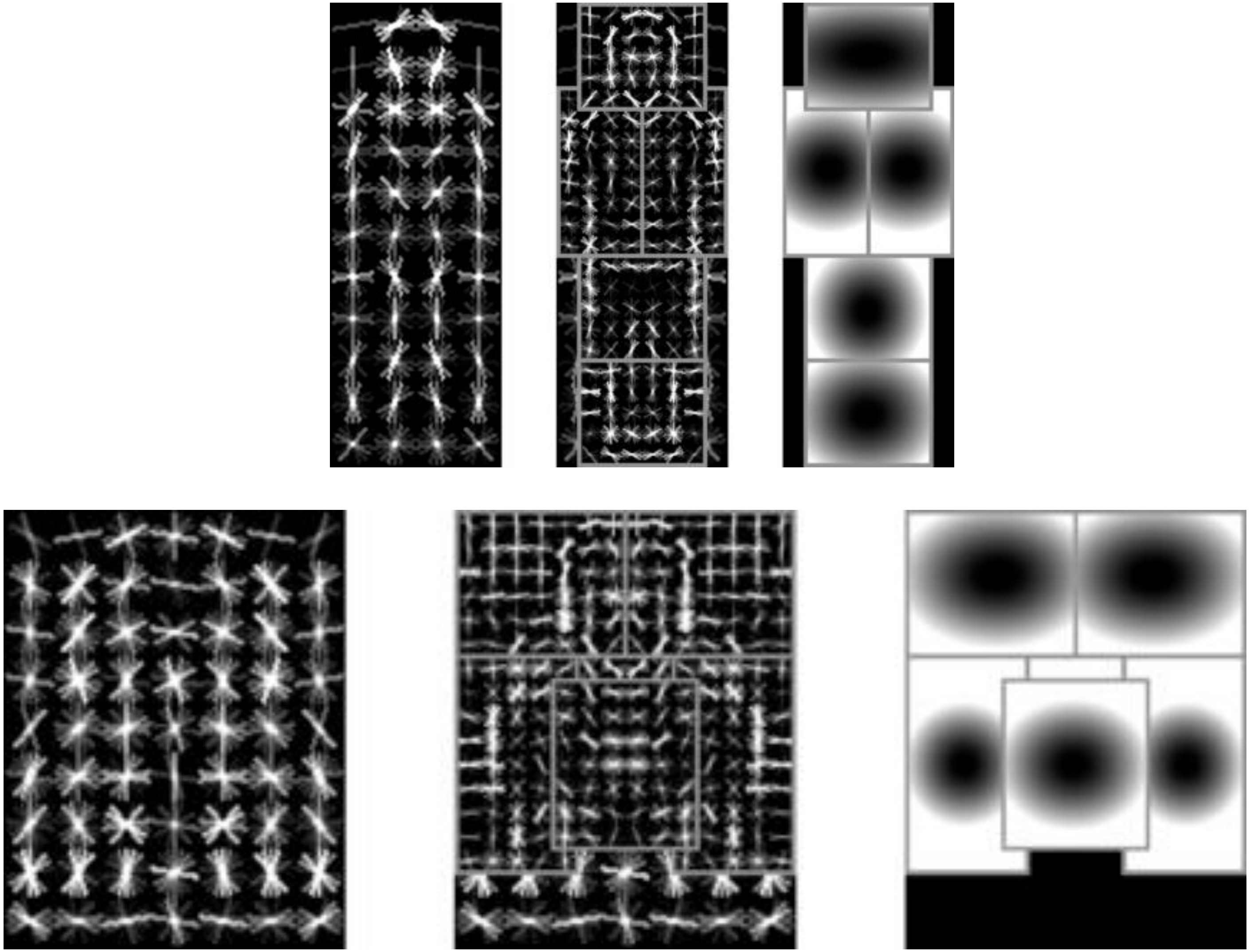
high scoring true positives



high scoring false positives



Person model

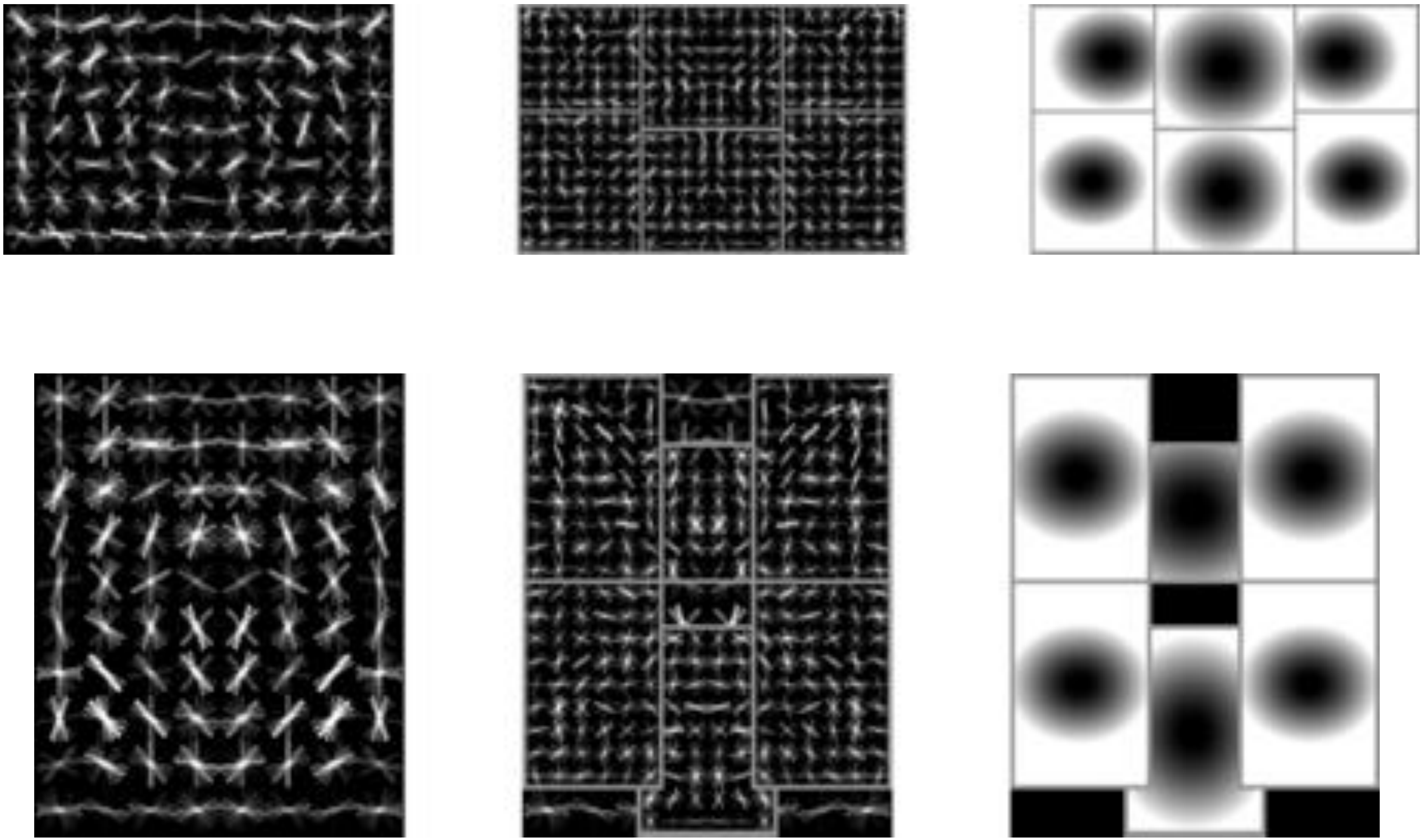


Person detections

high scoring true positives

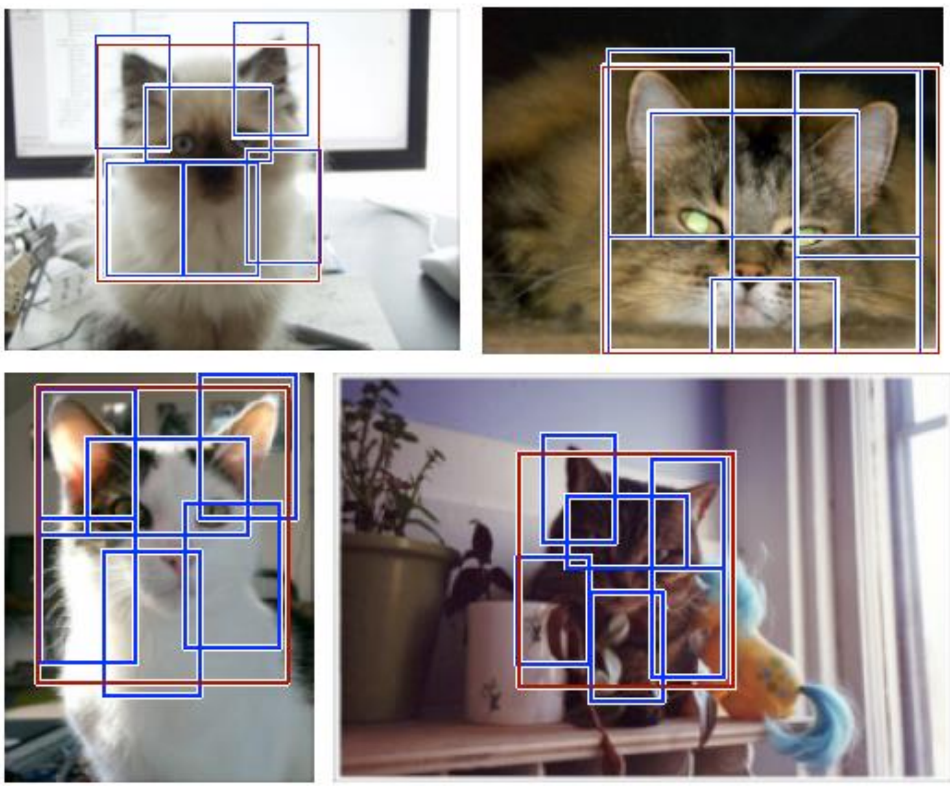
high scoring false positives
(not enough overlap)

Cat model

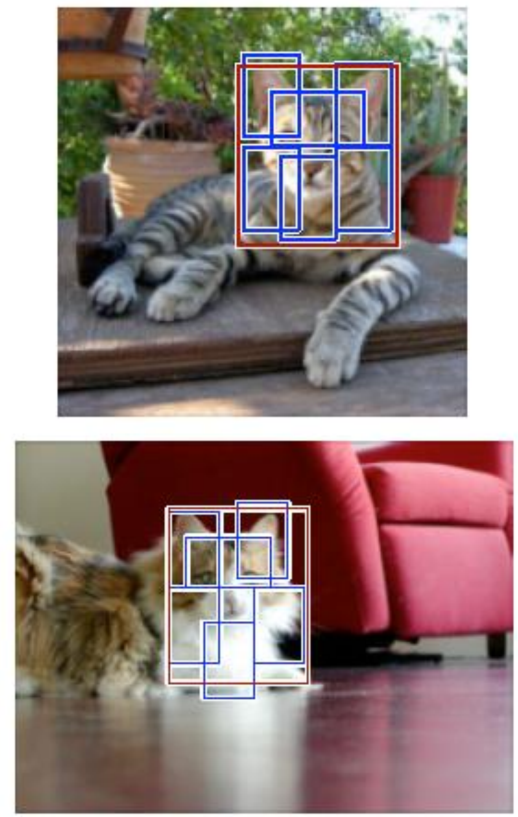


Cat detections

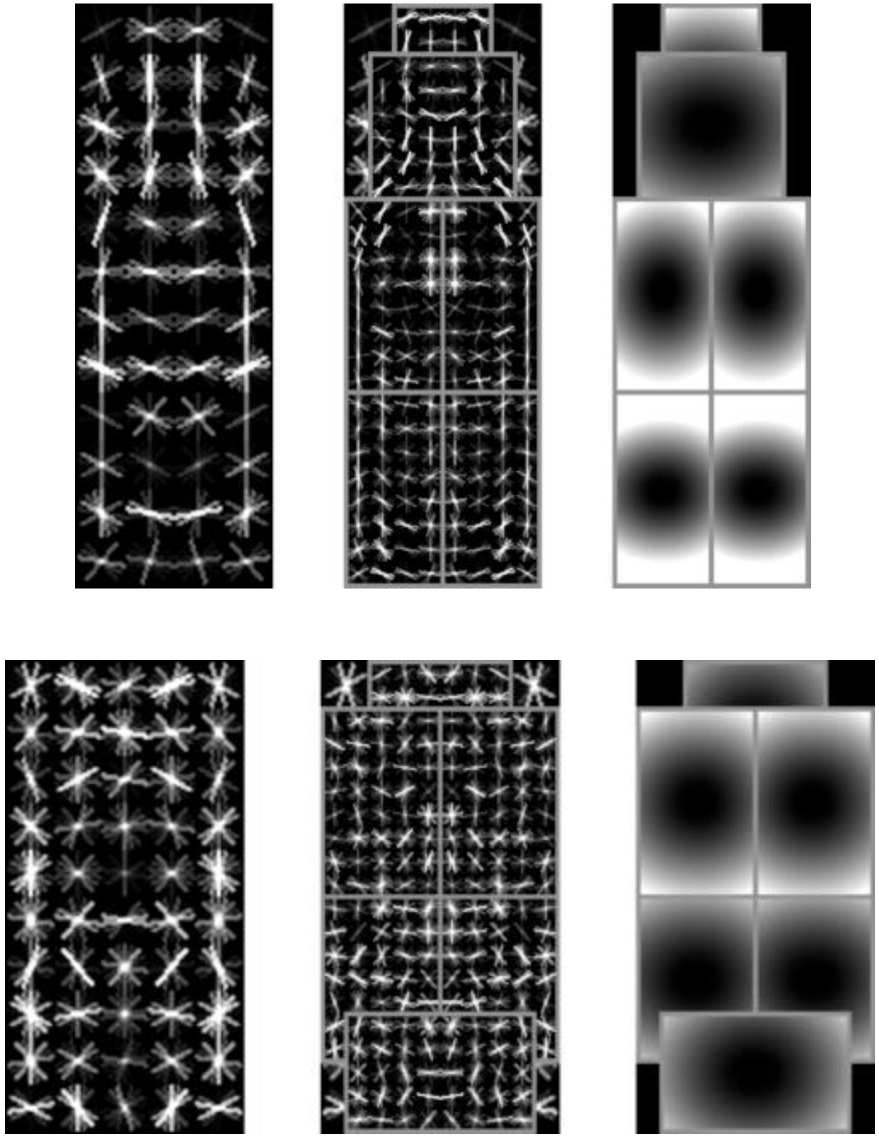
high scoring true positives



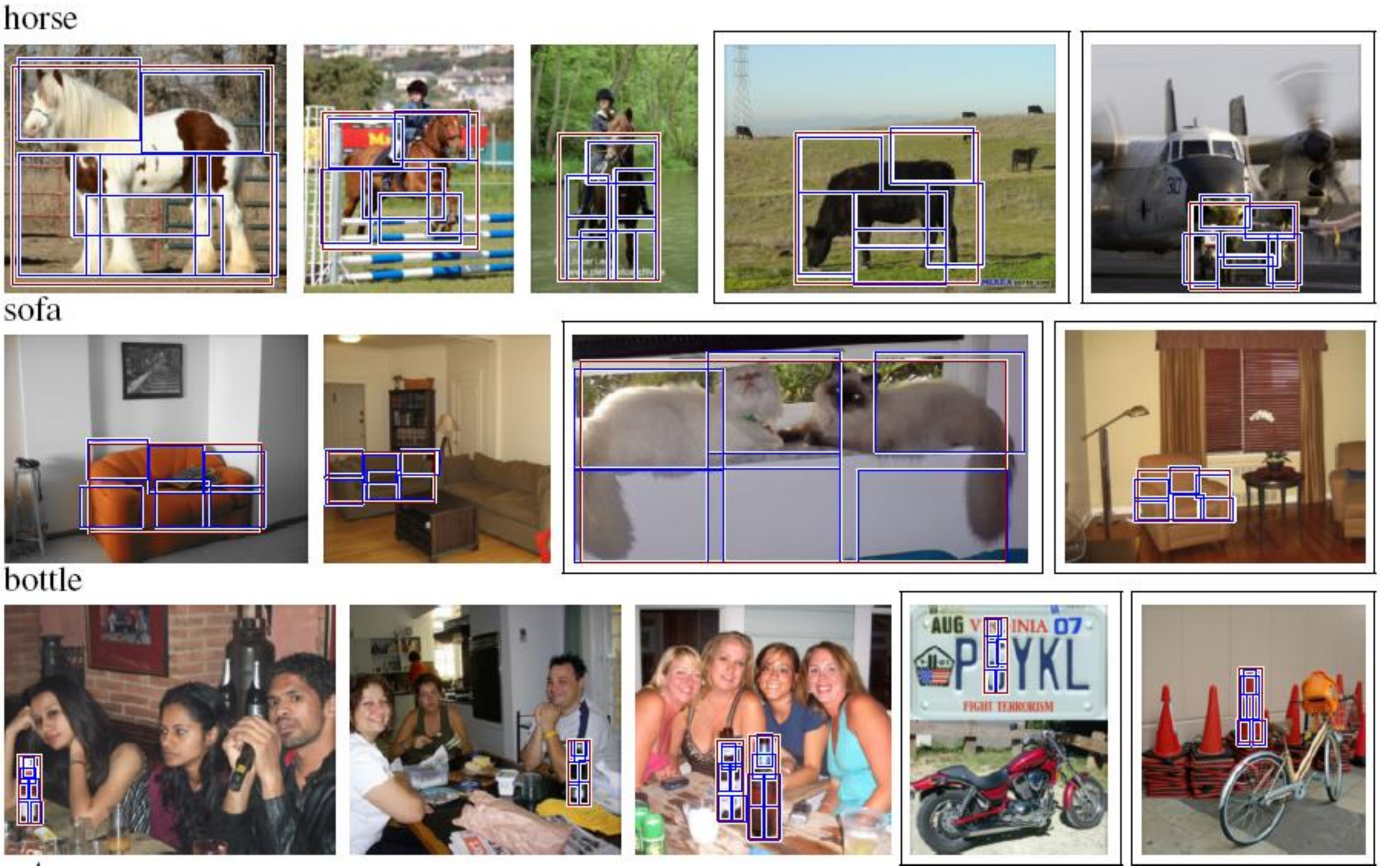
high scoring false positives
(not enough overlap)



Bottle model



More detections

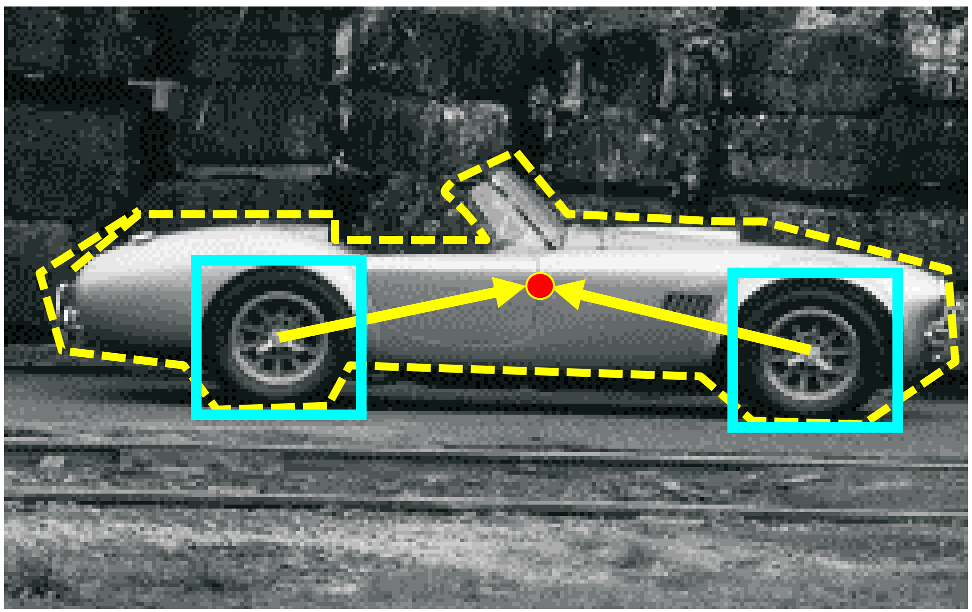


Implicit Shape Models

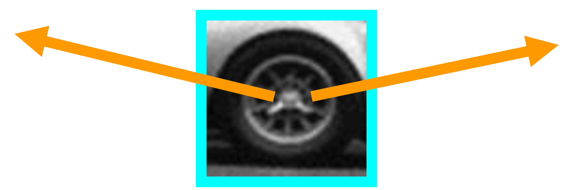
B. Leibe, A. Leonardis, and B. Schiele, Combined Object Categorization and Segmentation with an Implicit Shape Model, ECCV Workshop on Statistical Learning in Computer Vision 2004

Implicit shape models

- Visual codebook is used to index votes for object position



training image annotated with object localization info



visual codeword with displacement vectors

Implicit shape models

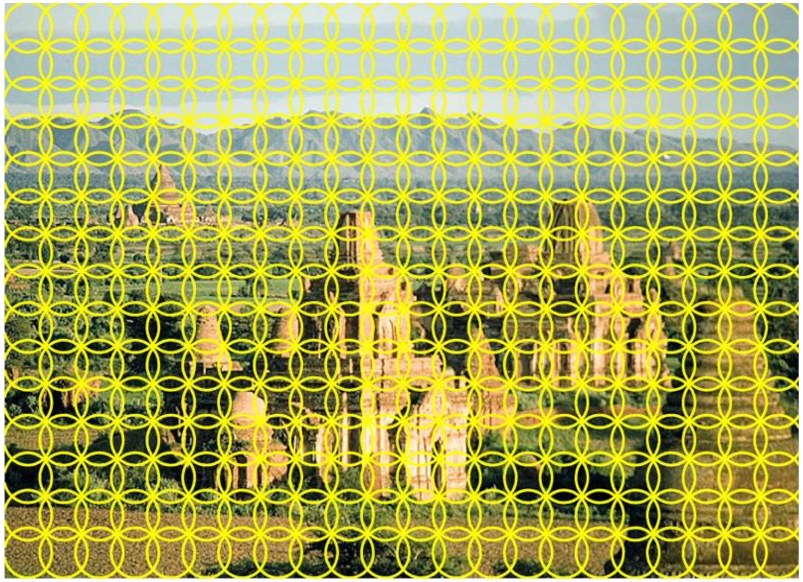
- Visual codebook is used to index votes for object position

test image

Visual codebook?

Mapping of image patches to discrete set of "visual words"

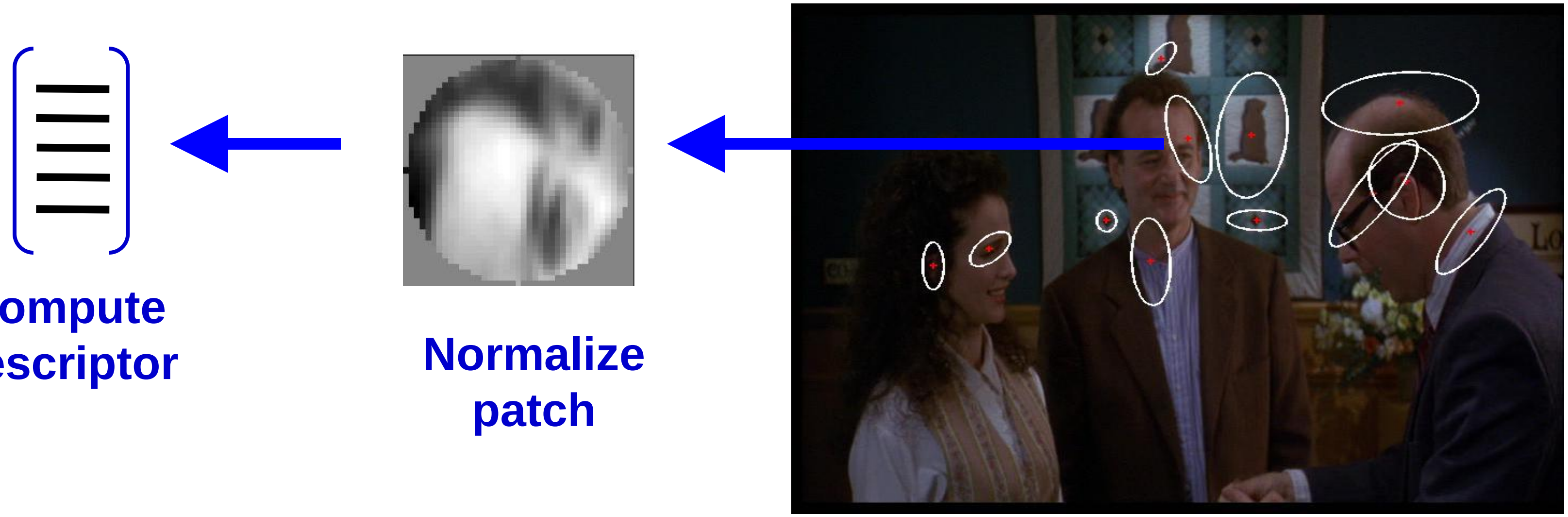
Visual codebook?



Candidate patches

Candidate patches

Visual codebook?

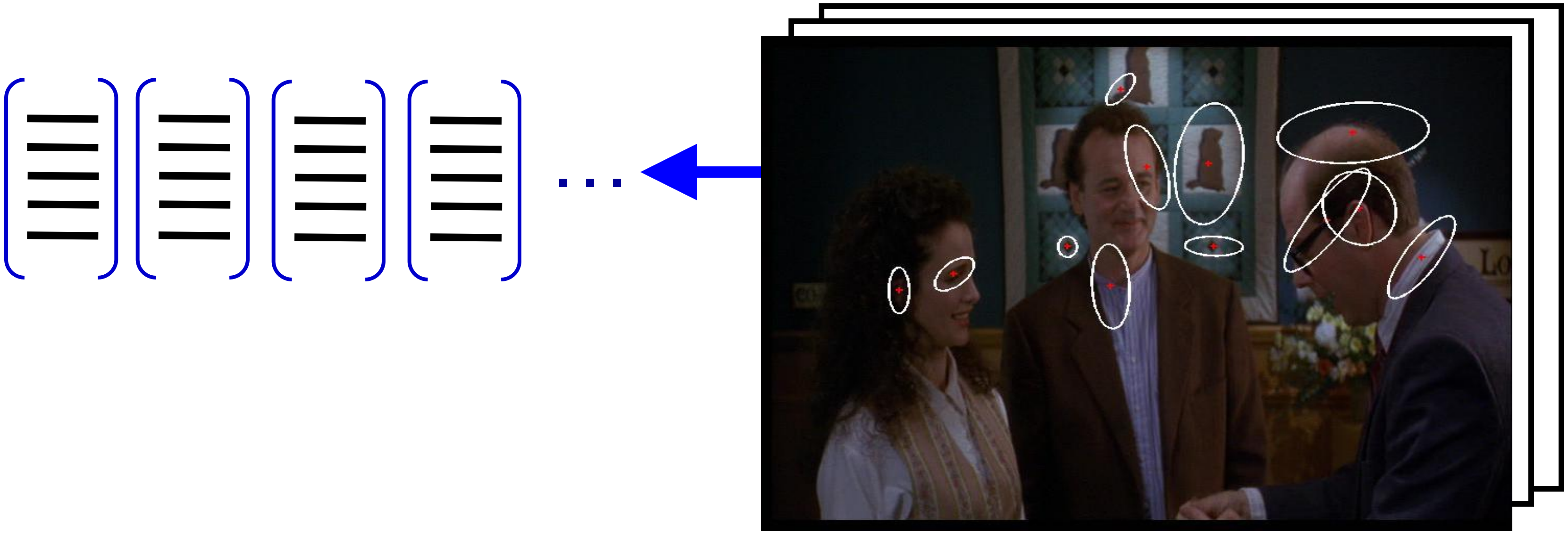


Compute descriptor

Normalize patch

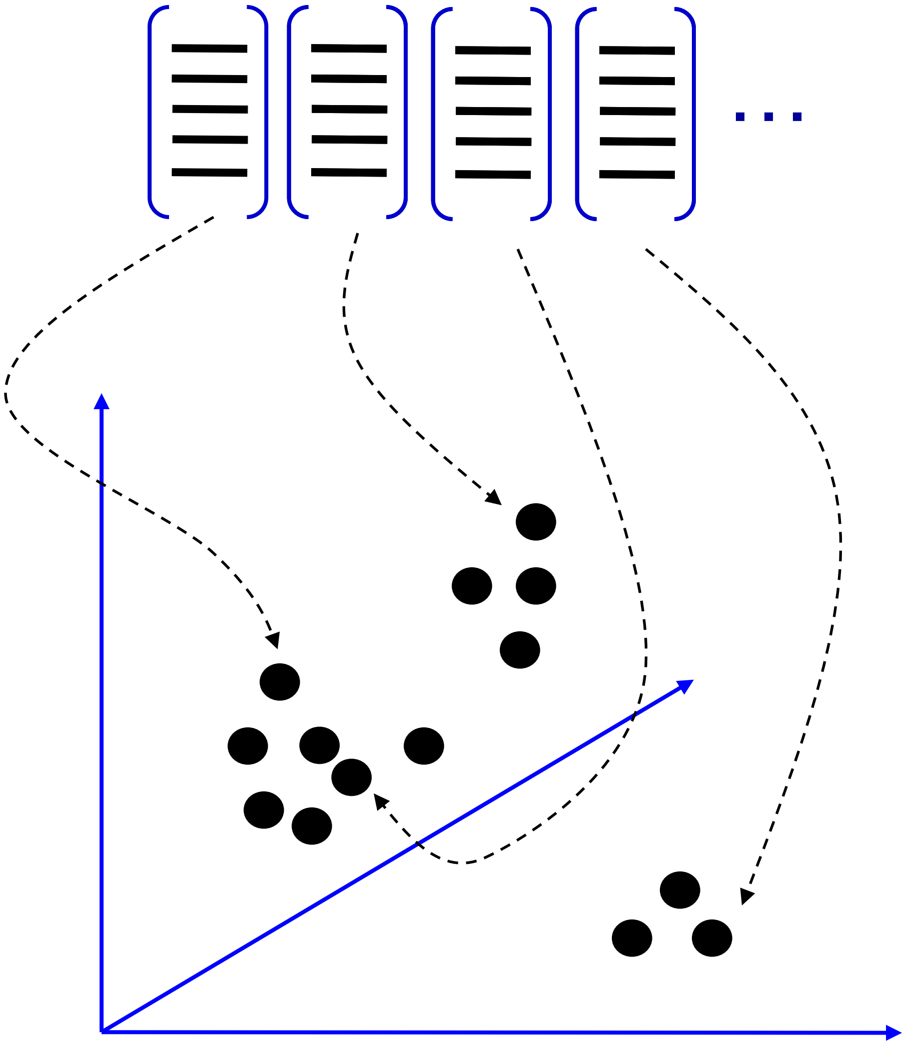
Candidate patches

Visual codebook?



Slide credit: Josef Sivic

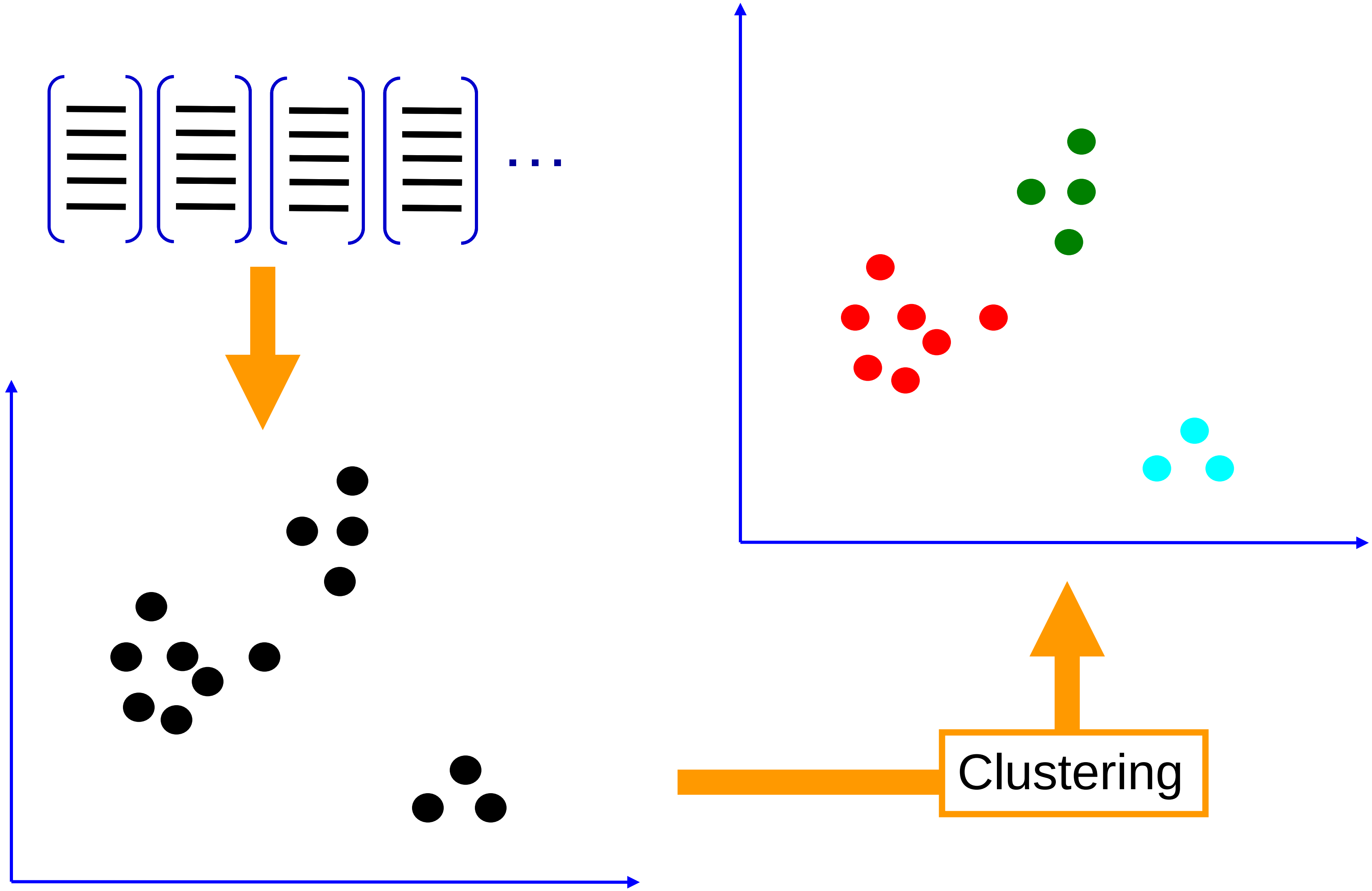
Visual codebook?



Feature space

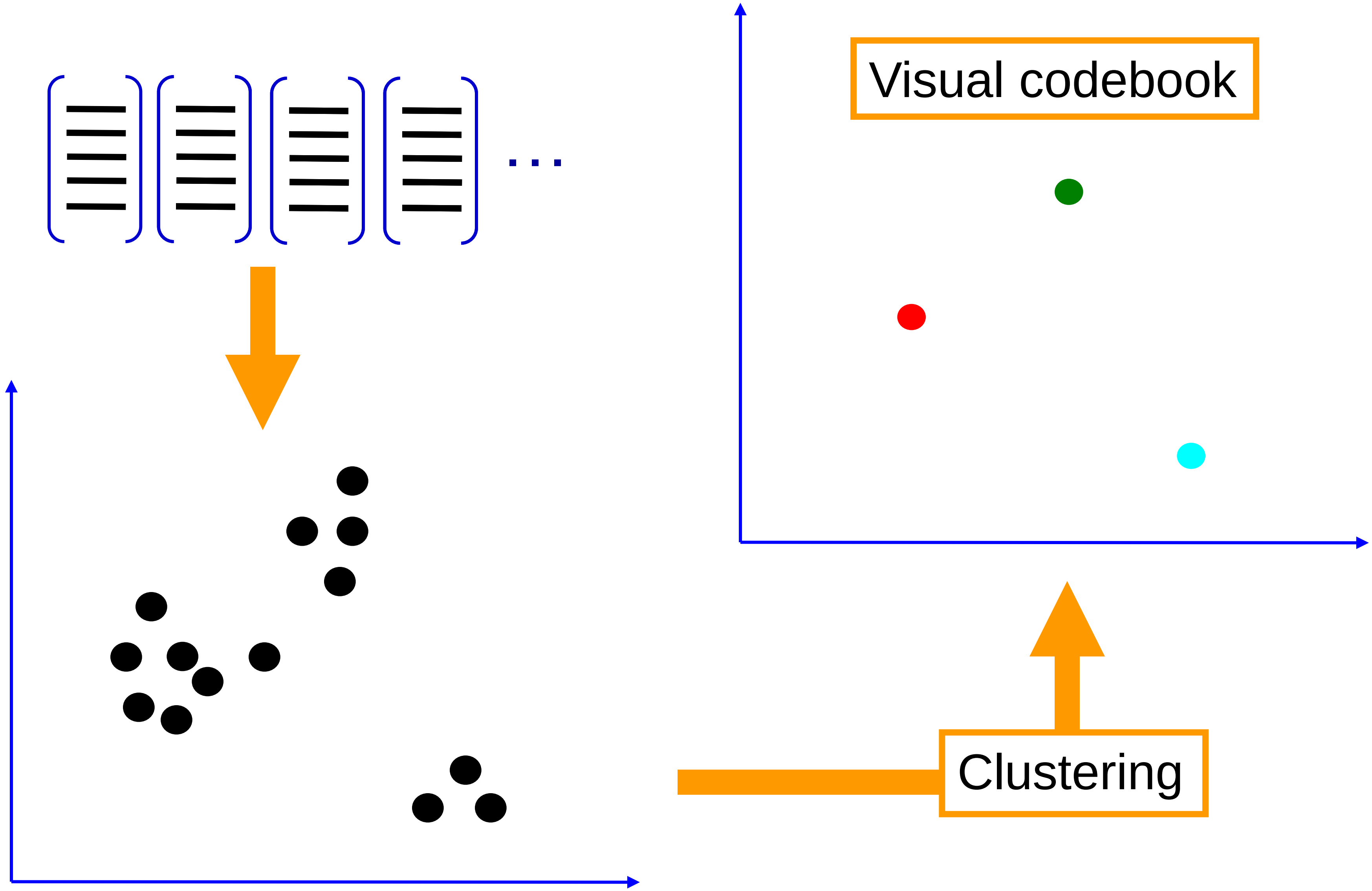
Slide credit: Josef Sivic

Visual codebook?



Slide credit: Josef Sivic

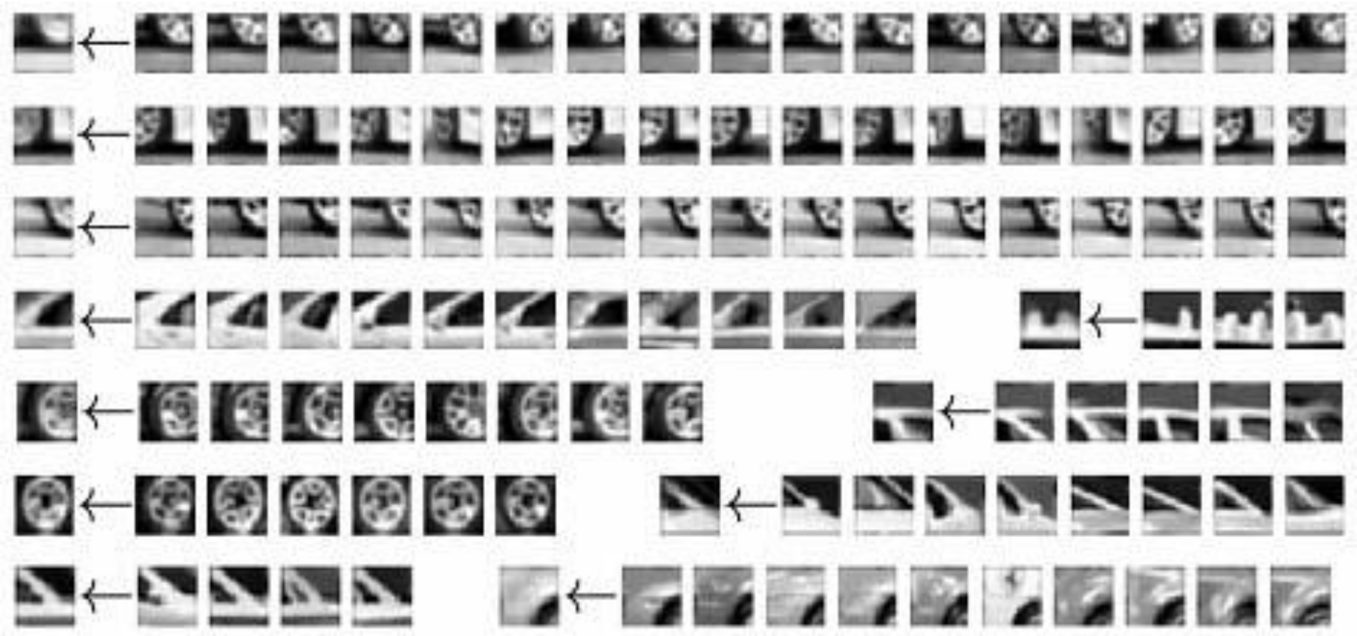
Visual codebook?



Slide credit: Josef Sivic

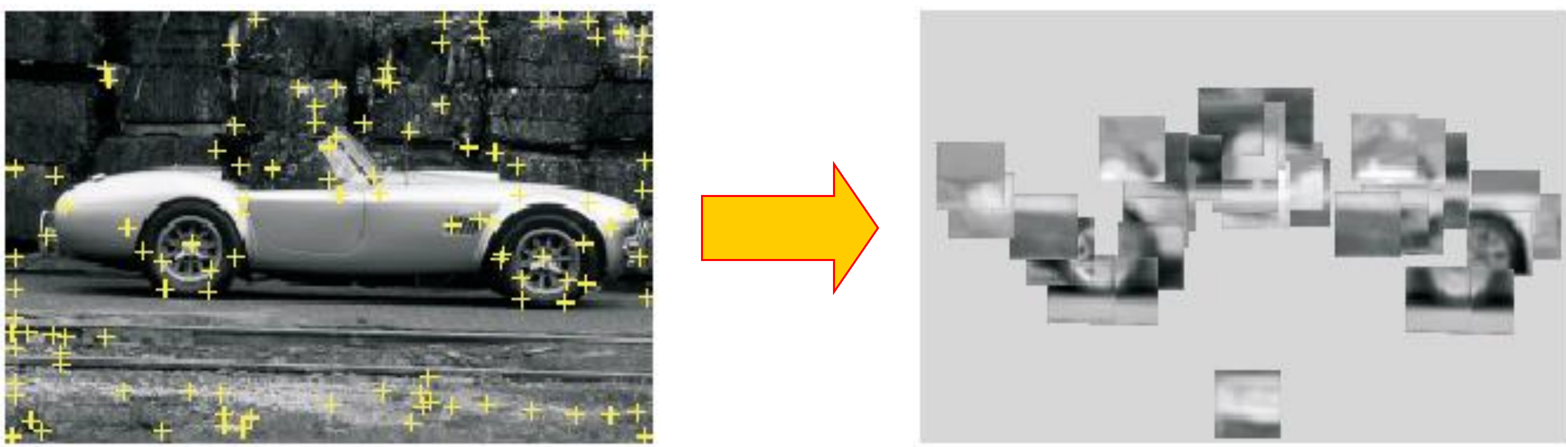
Implicit shape models: Training

1. Build codebook of patches around extracted interest points using clustering



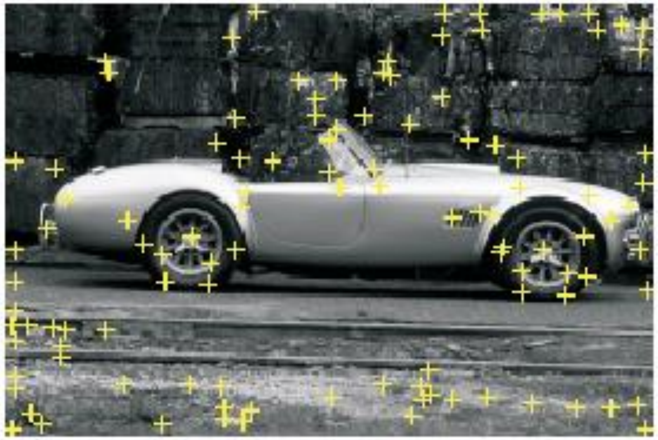
Implicit shape models: Training

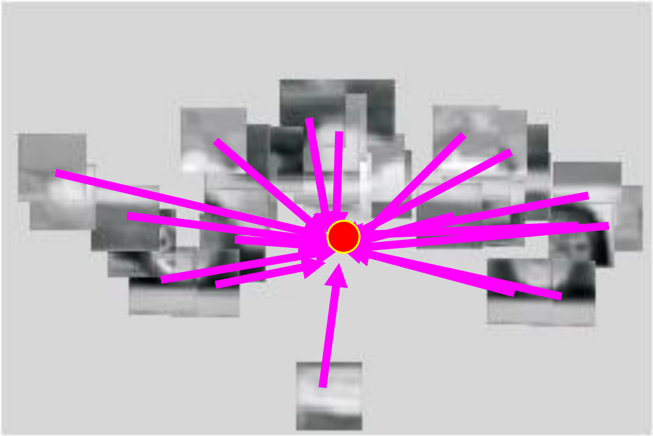
1. Build codebook of patches around extracted interest points using clustering
2. Map the patch around each interest point to closest codebook entry



Implicit shape models: Training

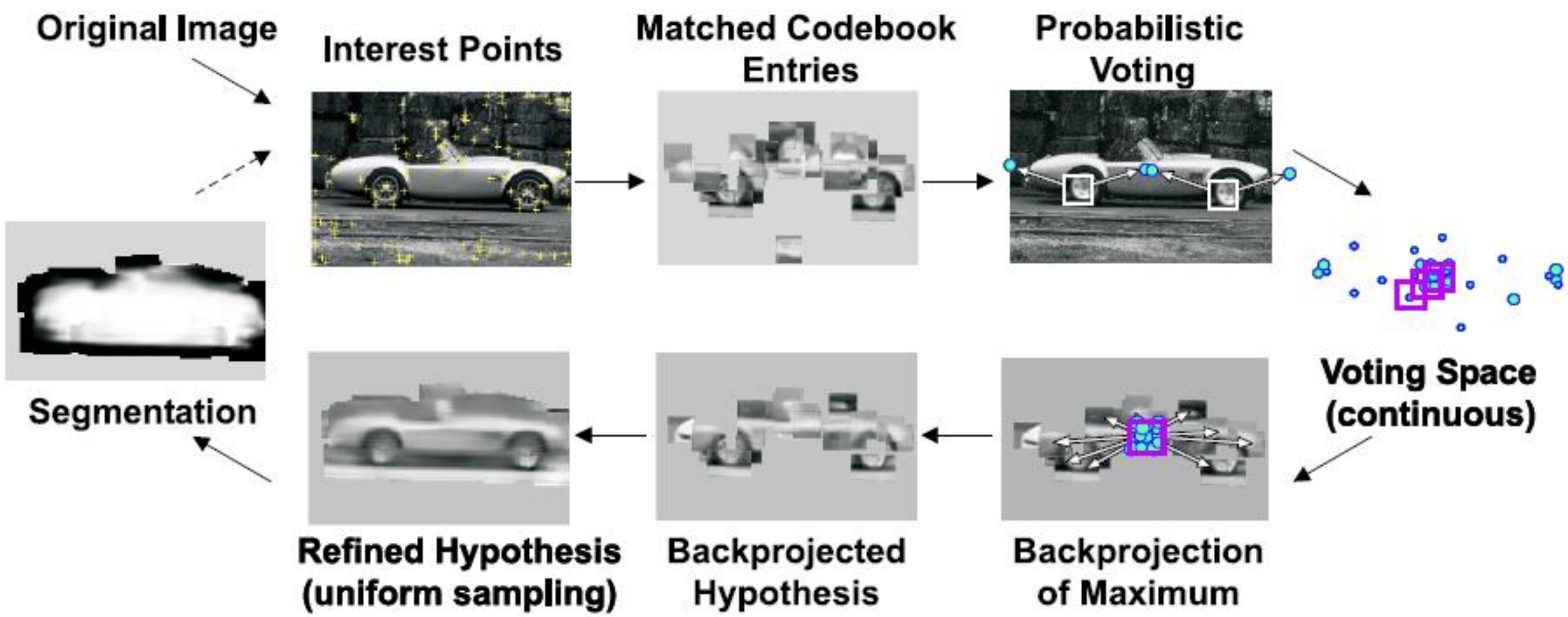
1. Build codebook of patches around extracted interest points using clustering
2. Map the patch around each interest point to closest codebook entry
3. For each codebook entry, store all positions it was found, relative to object center





Implicit shape models: Testing

1. Given test image, extract patches, match to codebook entry
2. Cast votes for possible positions of object center
3. Search for maxima in voting space
4. Extract weighted segmentation mask based on stored masks for the codebook occurrences

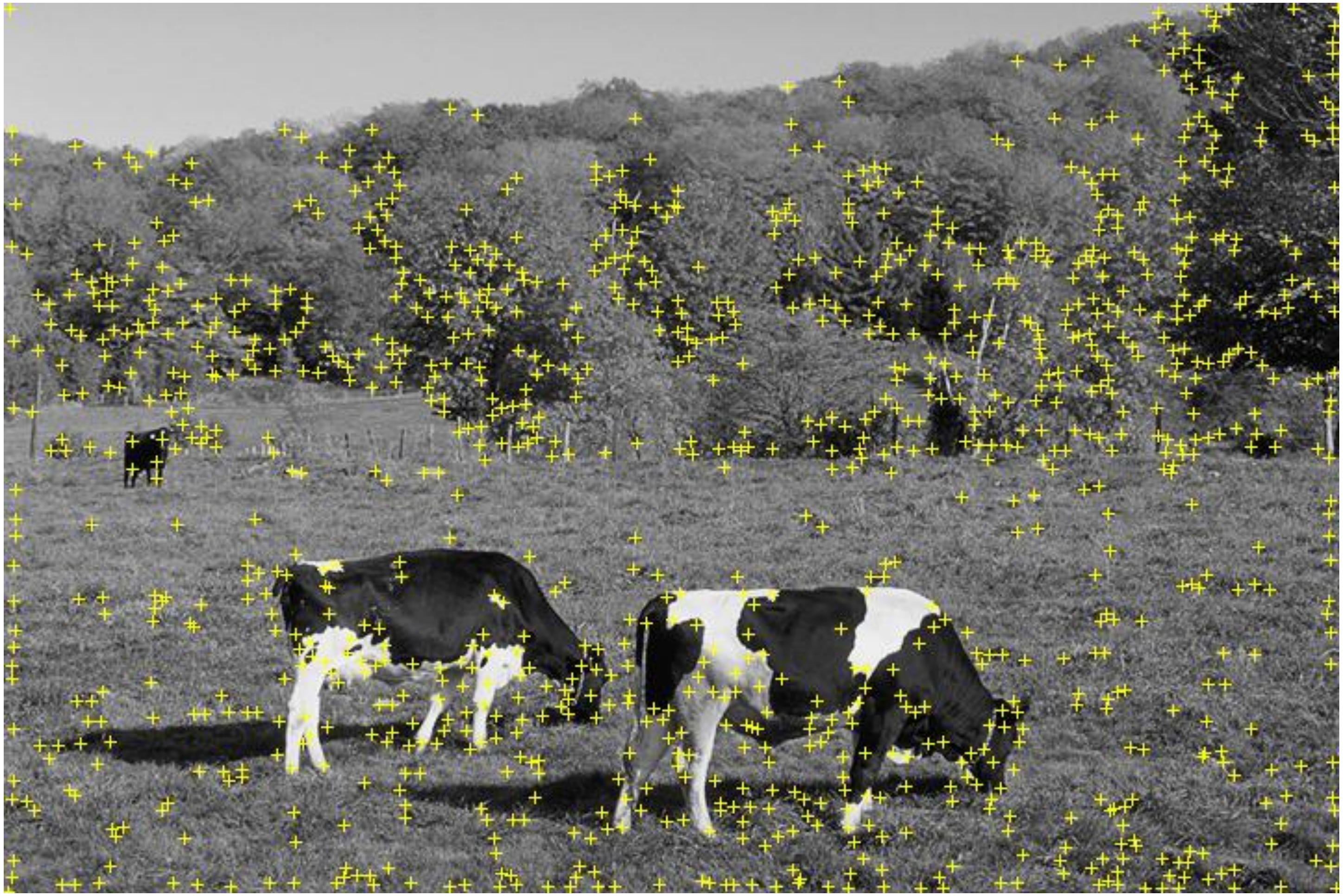


Example: Results on Cows

Original image

Source: B. Leibe

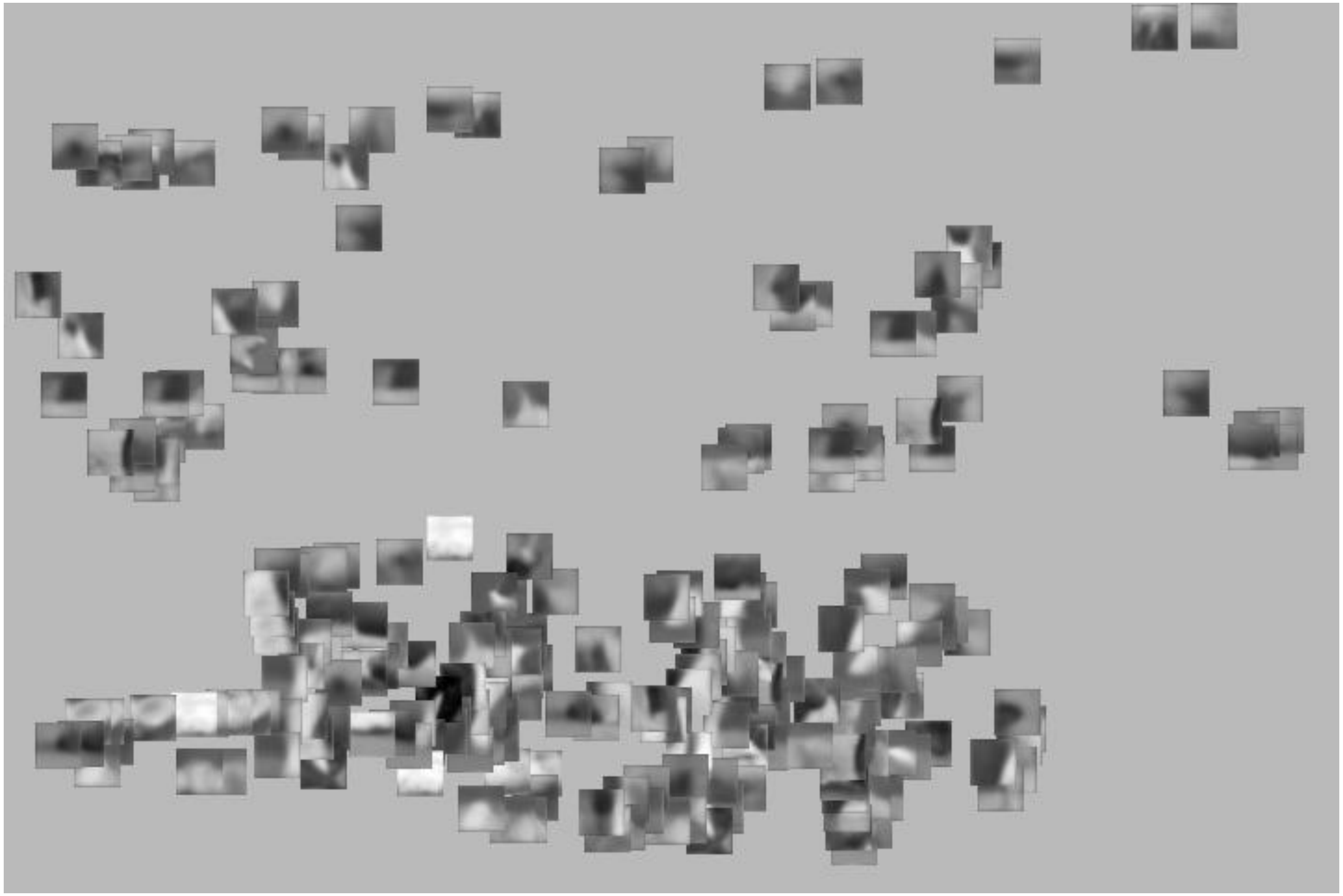
Example: Results on Cows



Interest points

Source: B. Leibe

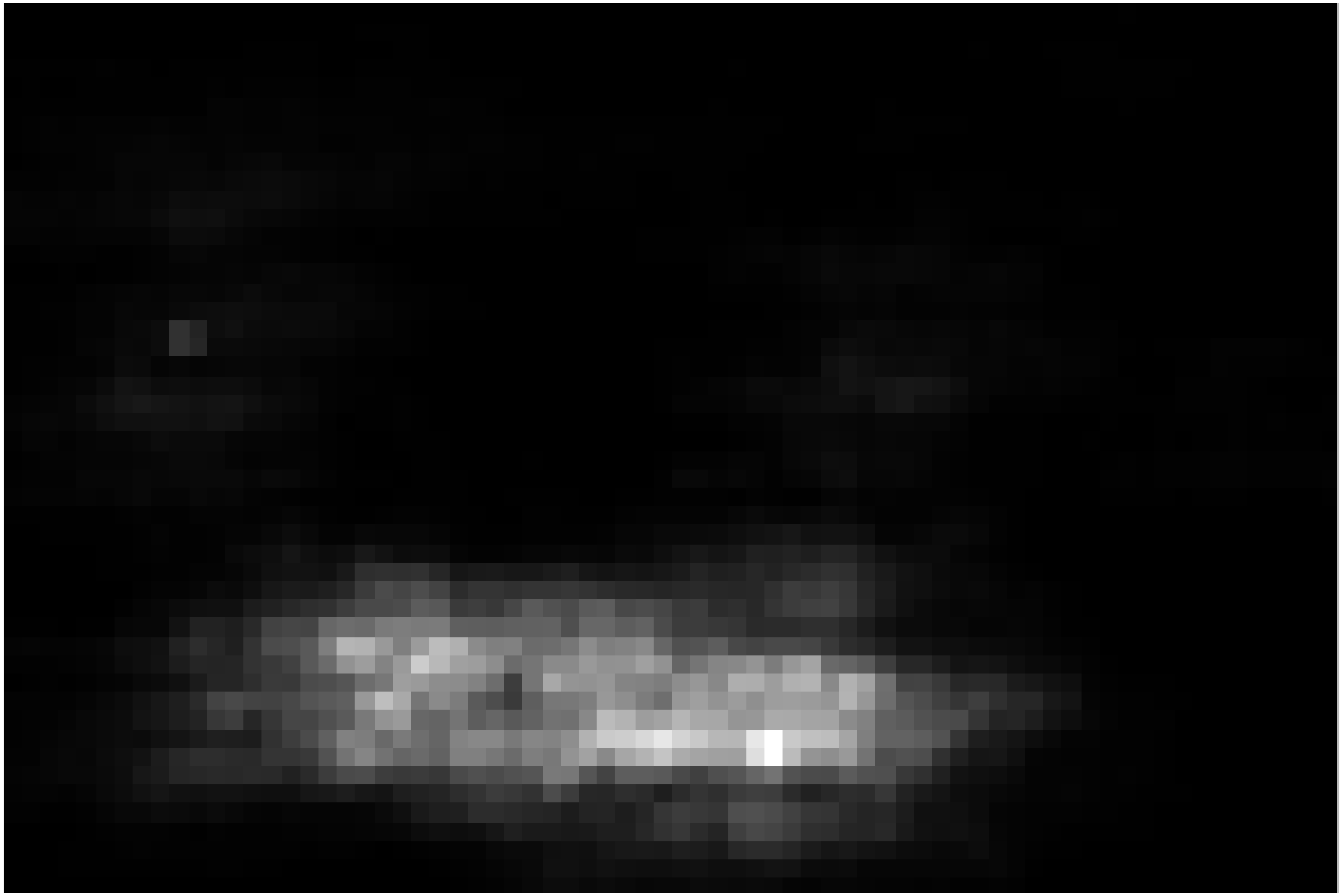
Example: Results on Cows



Matched patches

Source: B. Leibe

Example: Results on Cows



Probabilistic votes

Source: B. Leibe

Example: Results on Cows

Hypothesis 1

Source: B. Leibe

Example: Results on Cows

Hypothesis 2

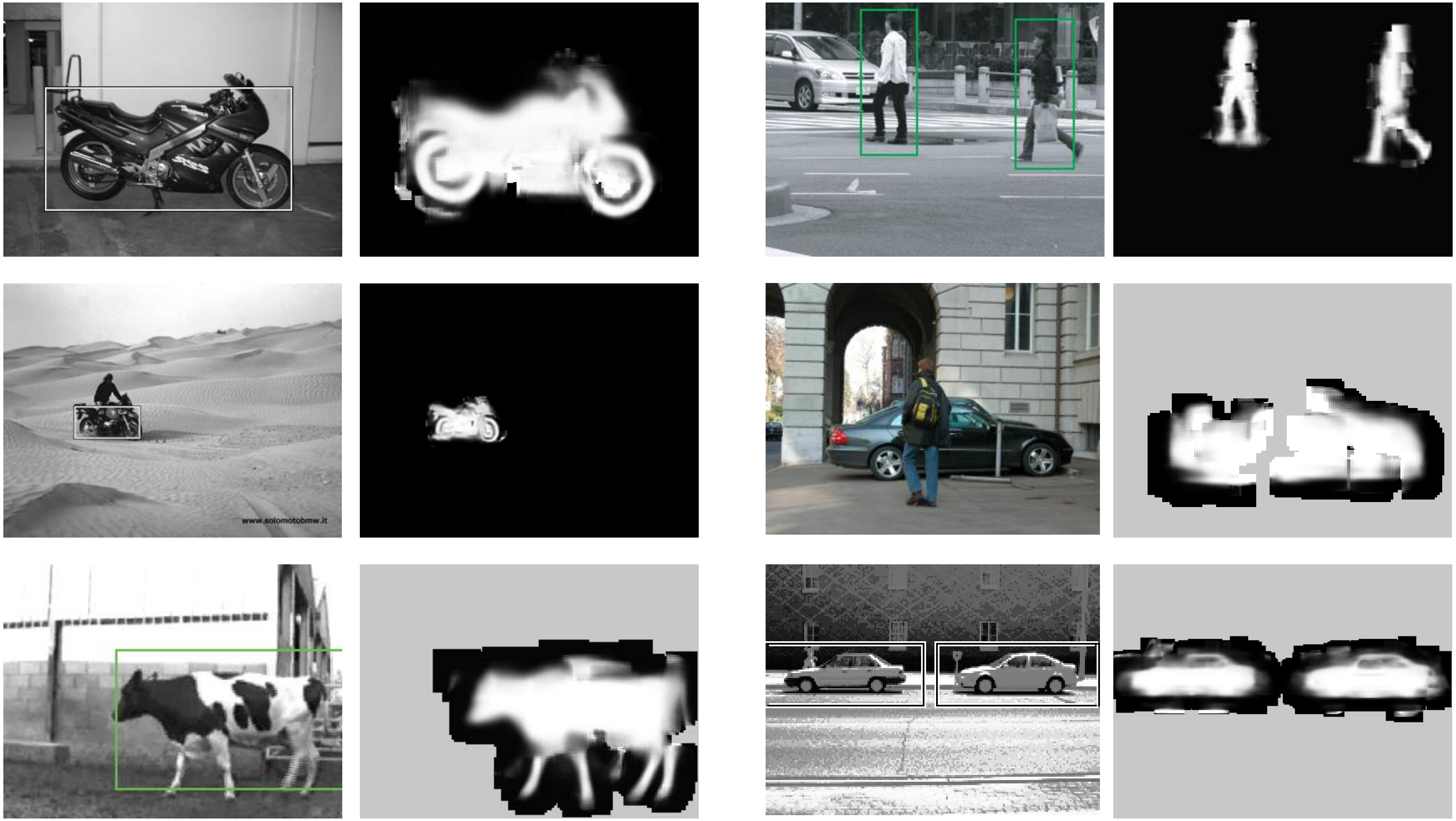
Source: B. Leibe

Example: Results on Cows

Hypothesis 3

Source: B. Leibe

Additional examples



B. Leibe, A. Leonardis, and B. Schiele, Robust Object Detection with Interleaved Categorization and Segmentation, IJCV 77 (1-3), pp. 259-289, 2008.

Example detections

[Dalal and Triggs, CVPR 2005]

Summary

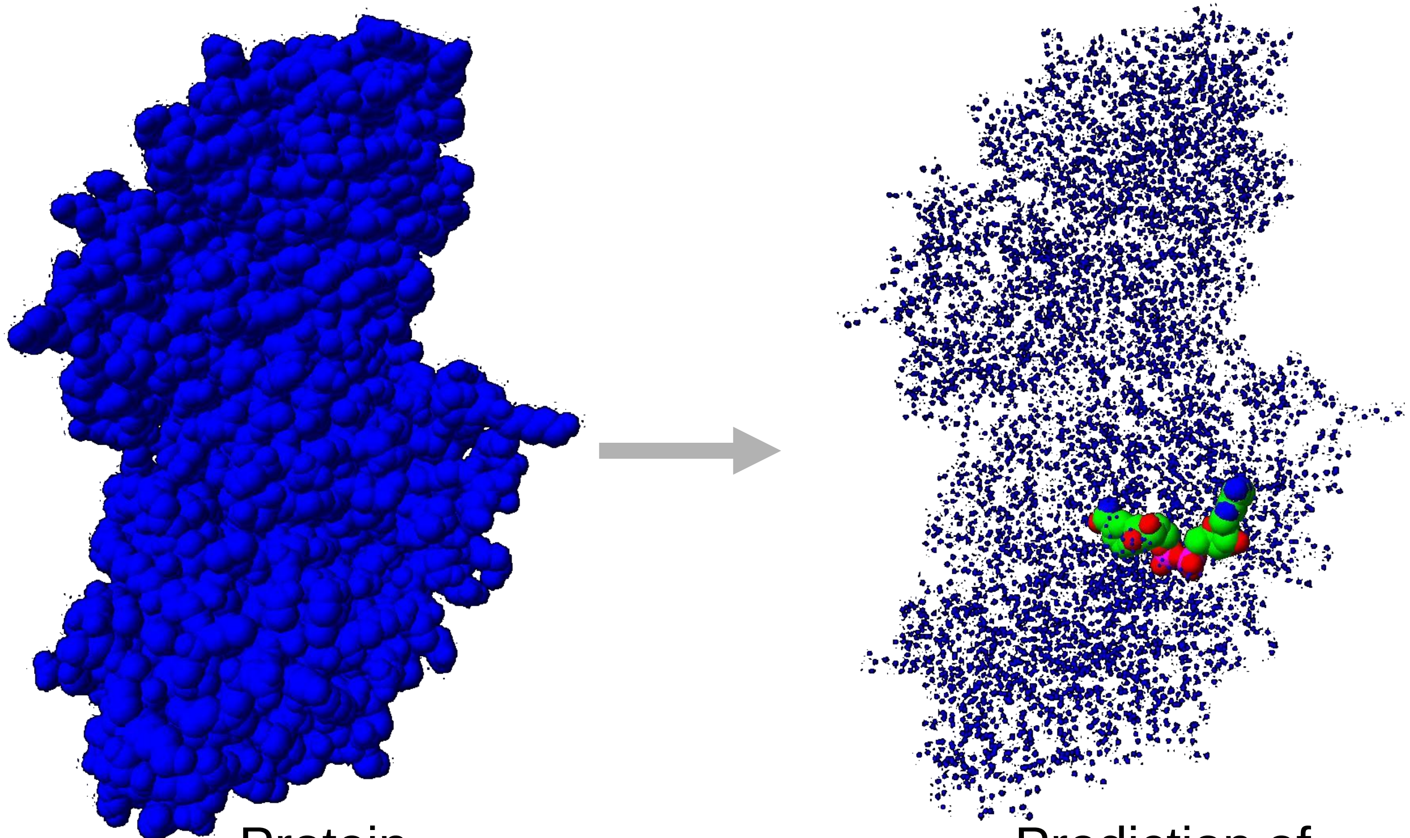
Part-based models

- Offer flexibility in comparison to rigid sliding windows
- Can be integrated with discriminative classifiers
- Provide good results for many object detection tasks

Another Application of ISM

Protein function prediction

X-site, Laskowski 1996

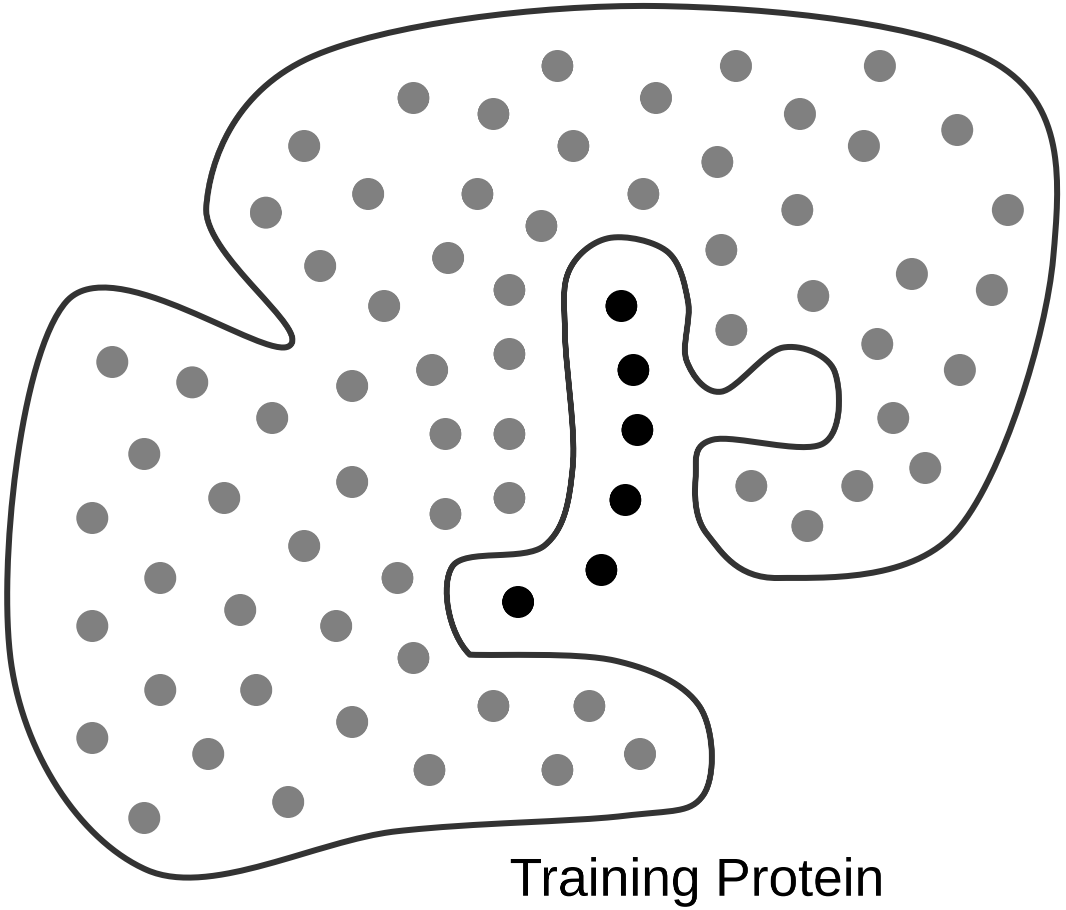


Protein Structure

Prediction of Bound Ligand

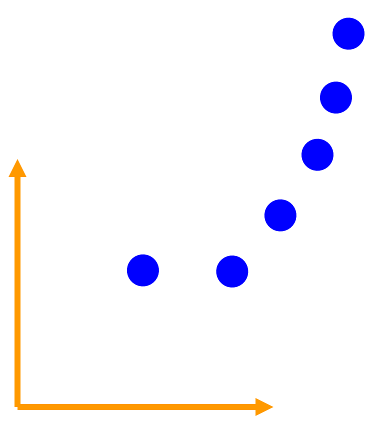
Binding Site Modeling

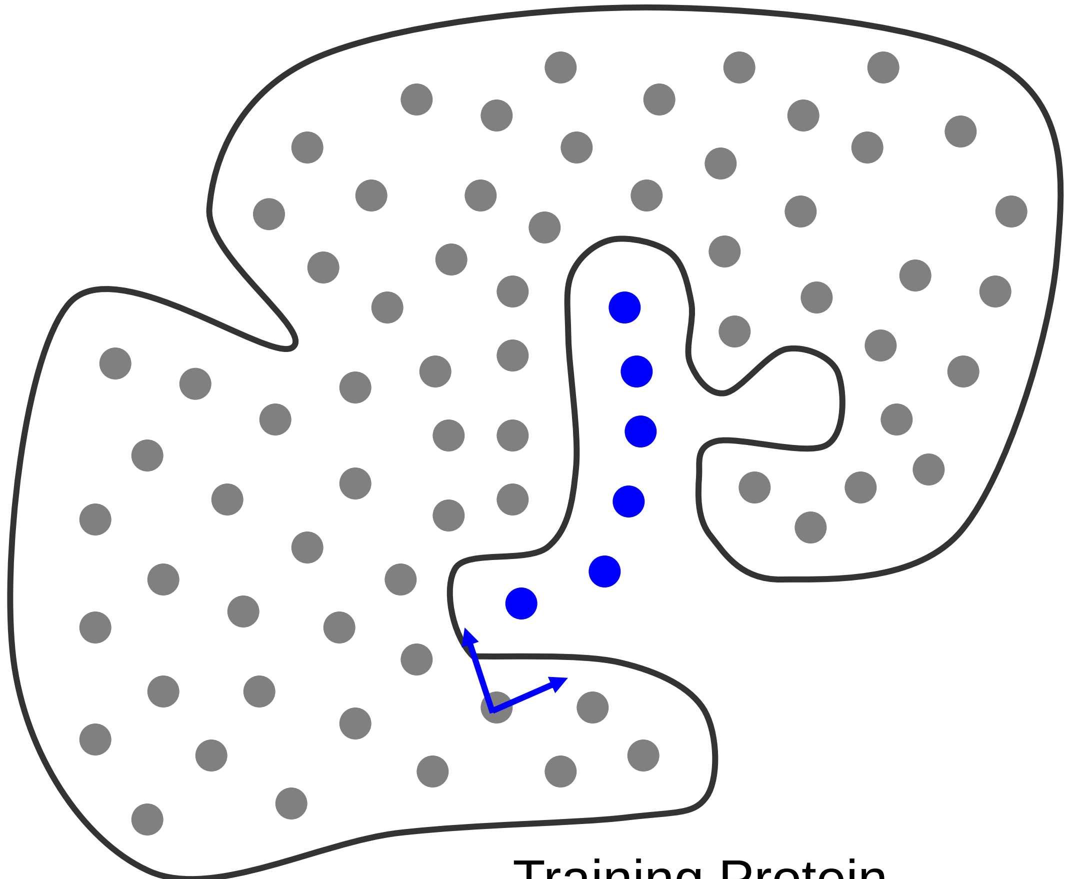
Train on distributions of ligand atoms relative to residues in bound proteins to develop predictive model for new binding sites



Binding Site Modeling

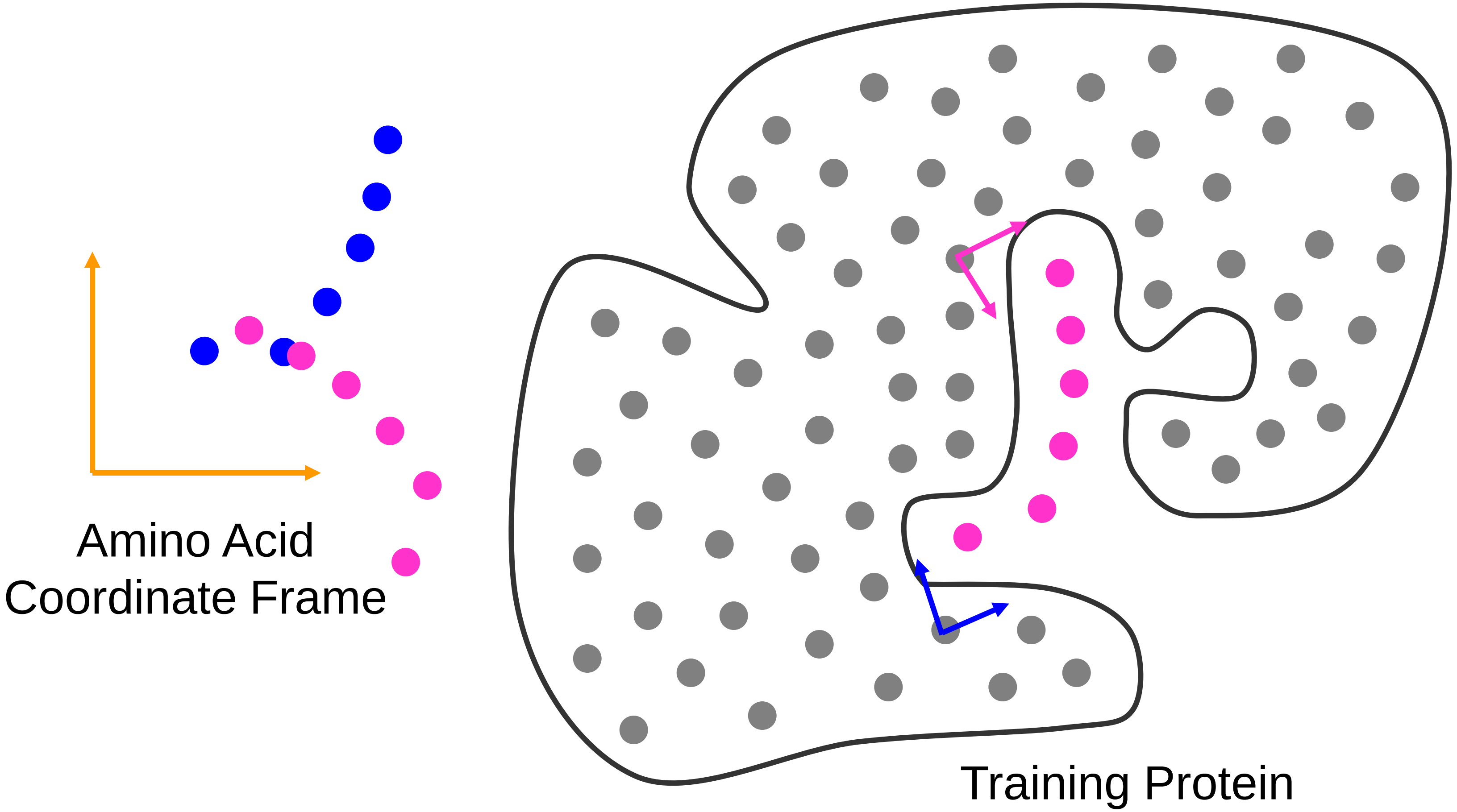
Train on distributions of ligand atoms relative to residues in bound proteins to develop predictive model for new binding sites





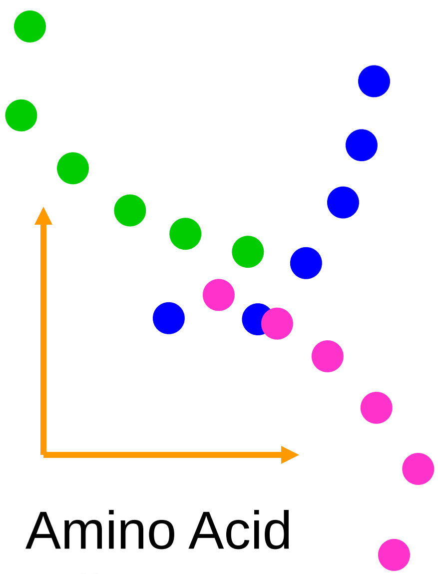
Binding Site Modeling

Train on distributions of ligand atoms relative to residues in bound proteins to develop predictive model for new binding sites

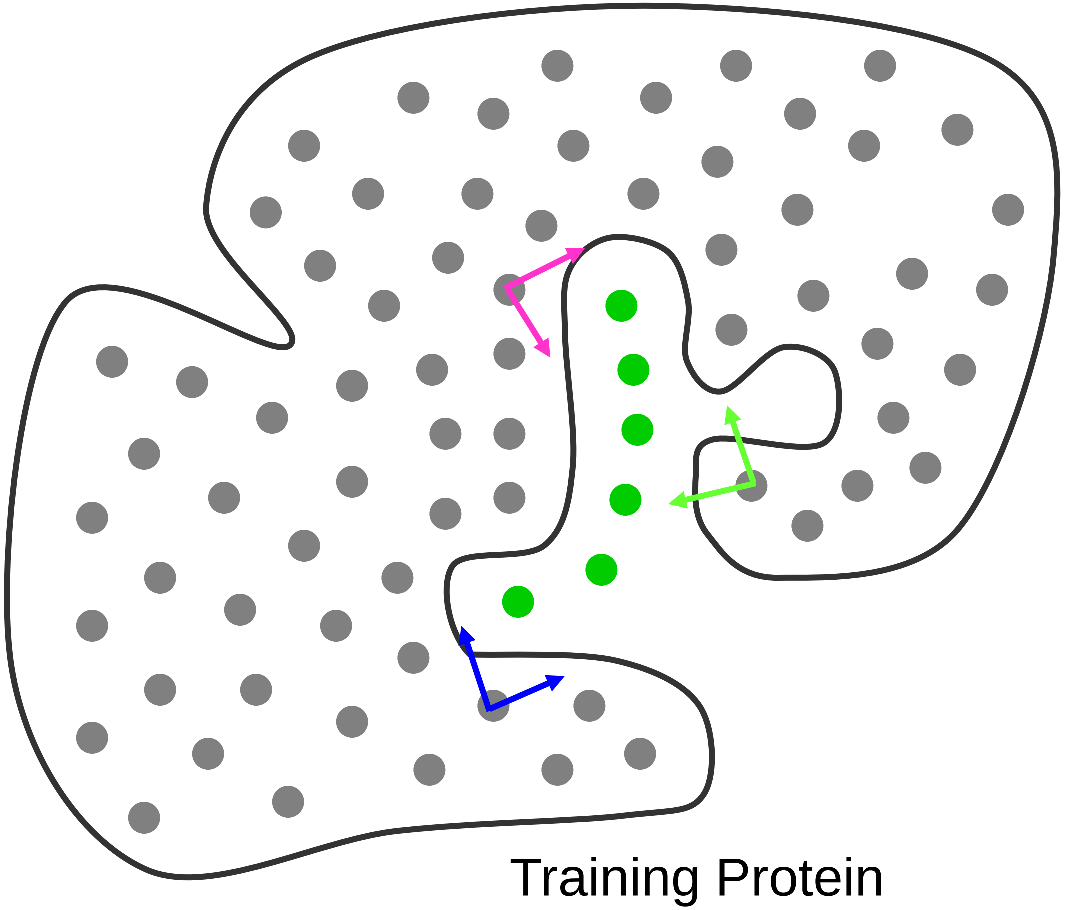


Binding Site Modeling

Train on distributions of ligand atoms relative to residues in bound proteins to develop predictive model for new binding sites



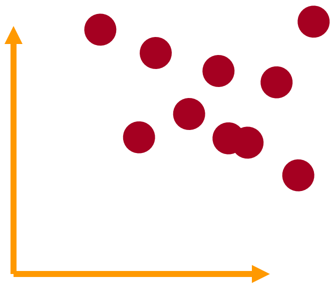
Amino Acid
Coordinate Frame



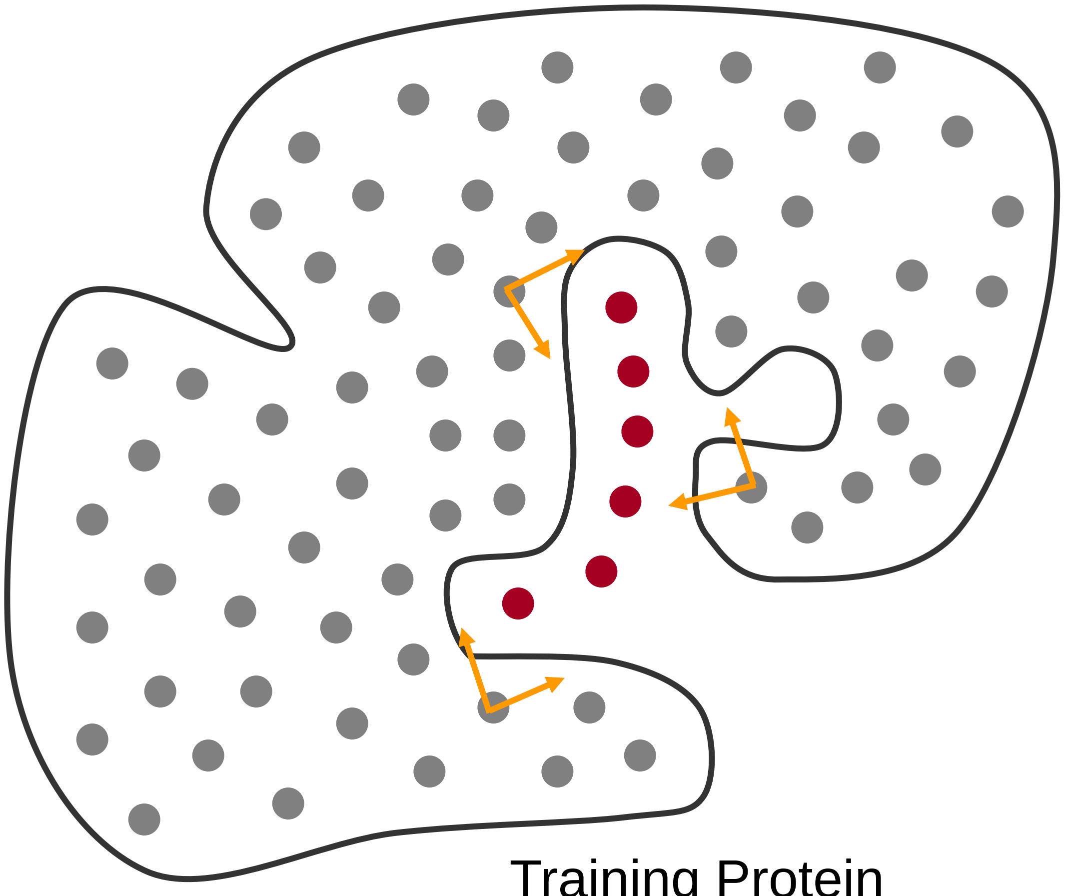
Training Protein

Binding Site Modeling

Train on distributions of ligand atoms relative to residues in bound proteins to develop predictive model for new binding sites



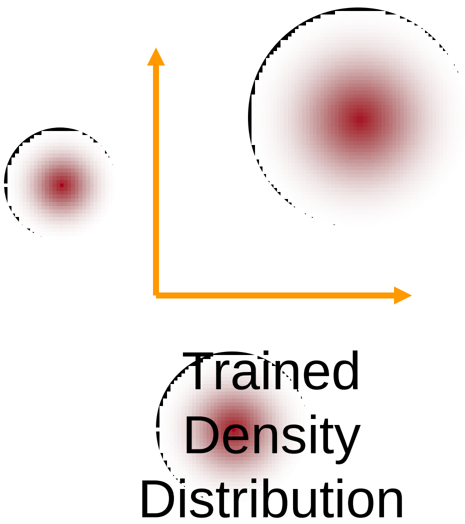
Amino Acid
Coordinate Frame

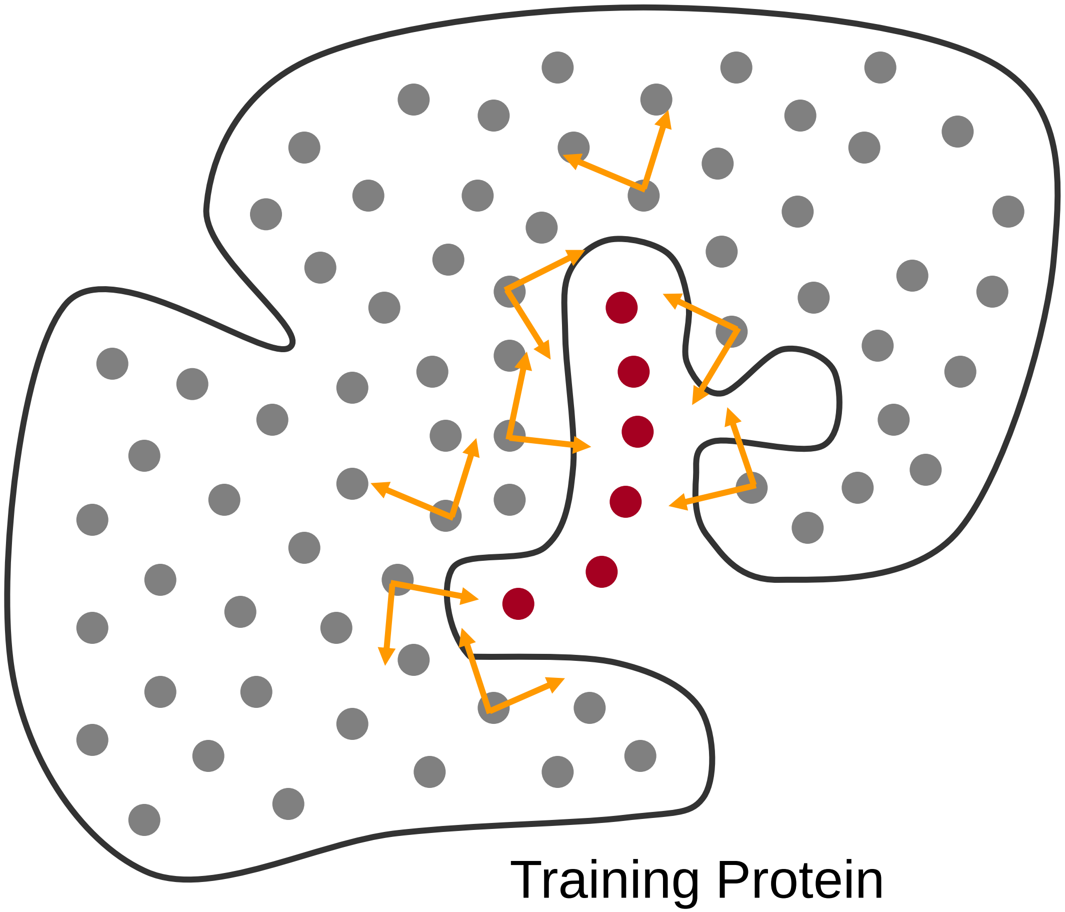


Training Protein

Binding Site Modeling

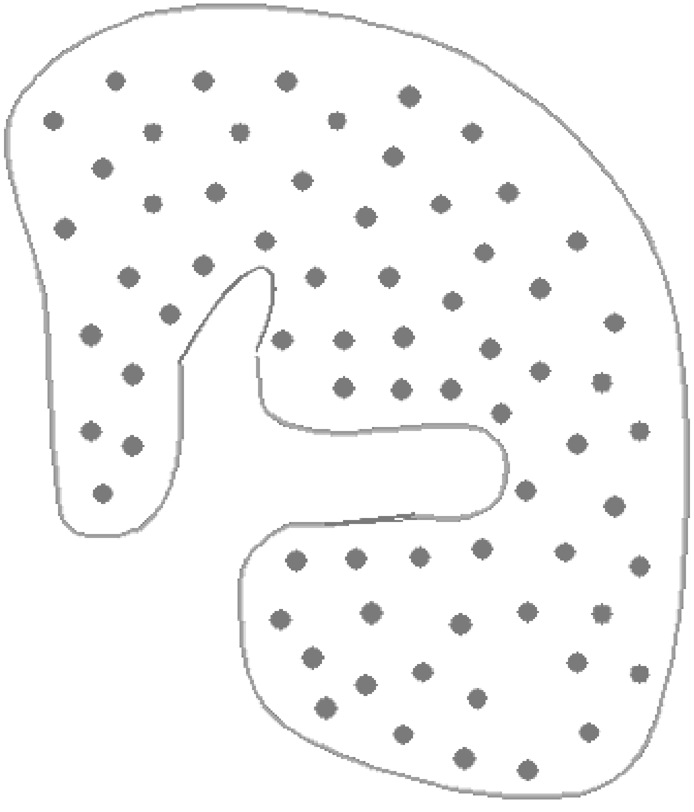
Train on distributions of ligand atoms relative to residues in bound proteins to develop predictive model for new binding sites





Binding Site Modeling

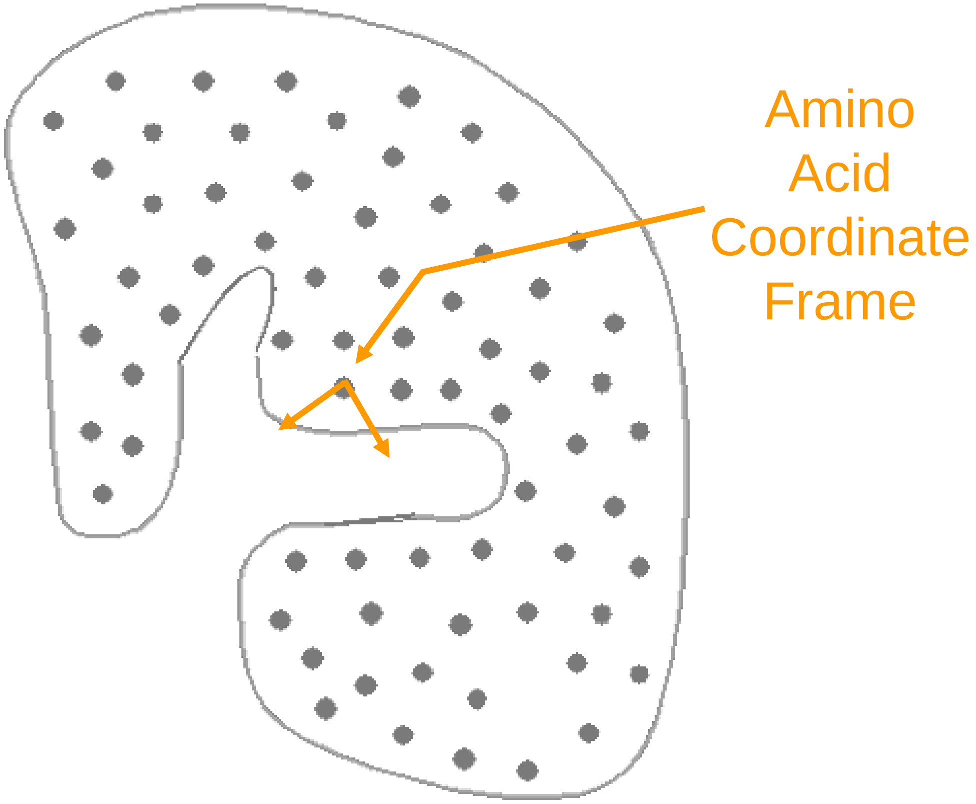
Train on distributions of ligand atoms relative to residues in bound proteins to develop predictive model for new binding sites



New Protein

Binding Site Modeling

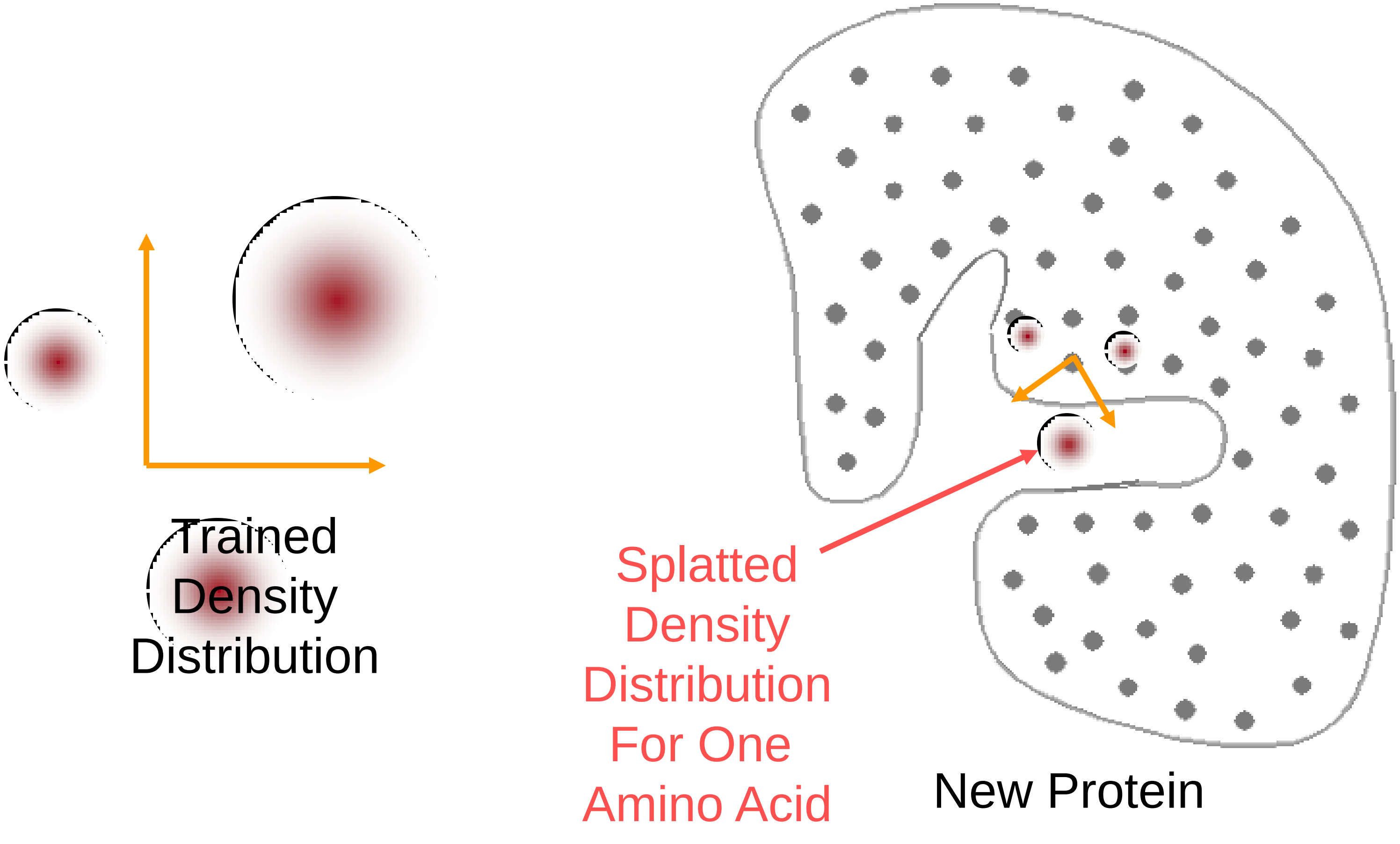
Train on distributions of ligand atoms relative to residues in bound proteins to develop predictive model for new binding sites



New Protein

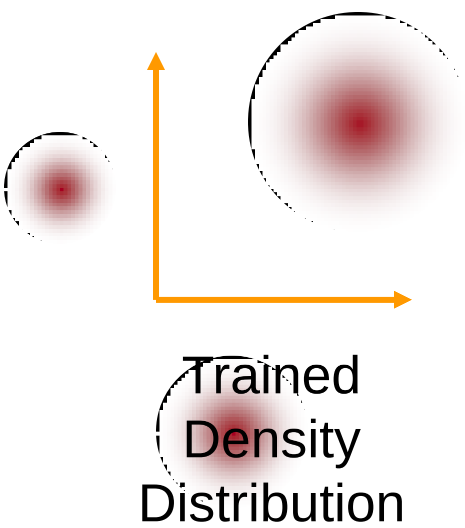
Binding Site Modeling

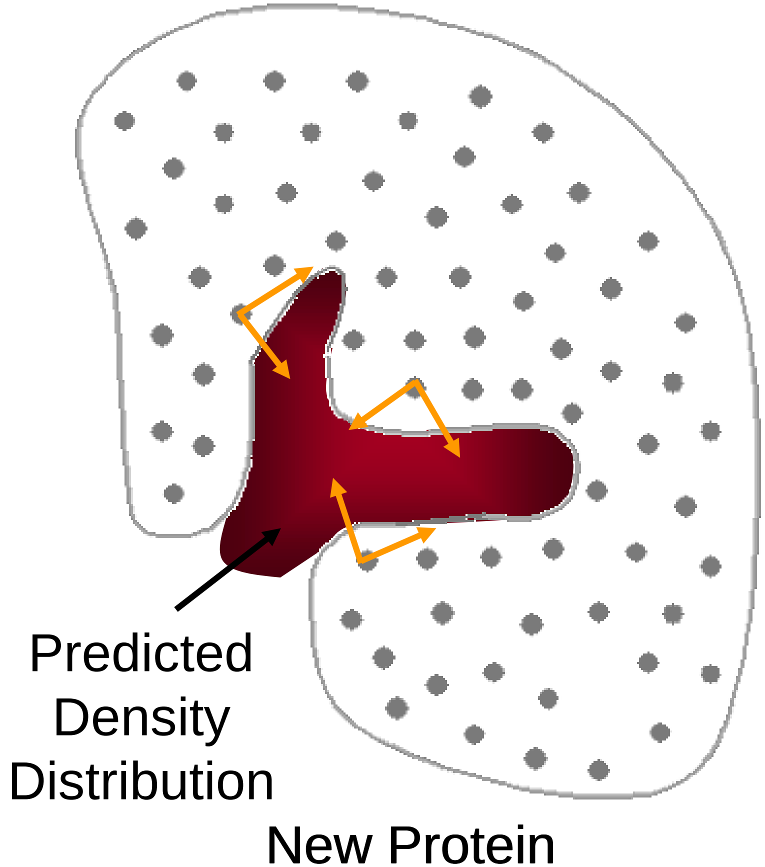
Train on distributions of ligand atoms relative to residues in bound proteins to develop predictive model for new binding sites



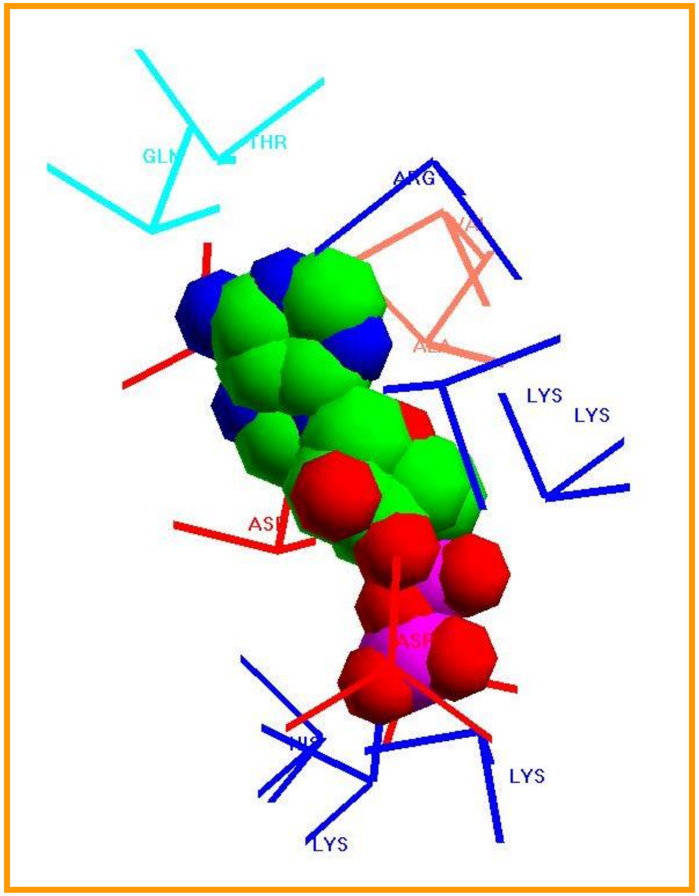
Binding Site Modeling

Train on distributions of ligand atoms relative to residues in bound proteins to develop predictive model for new binding sites

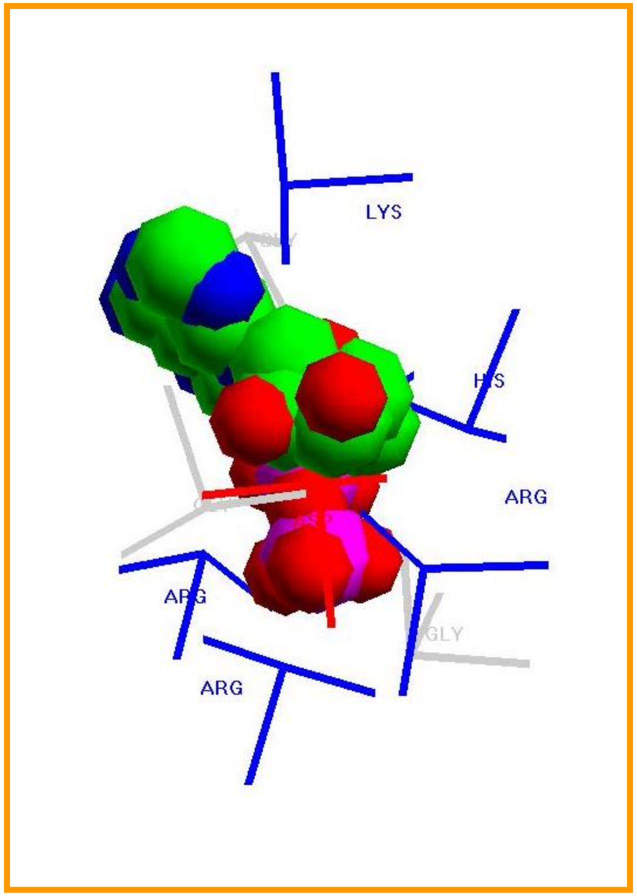




Binding Site Modeling



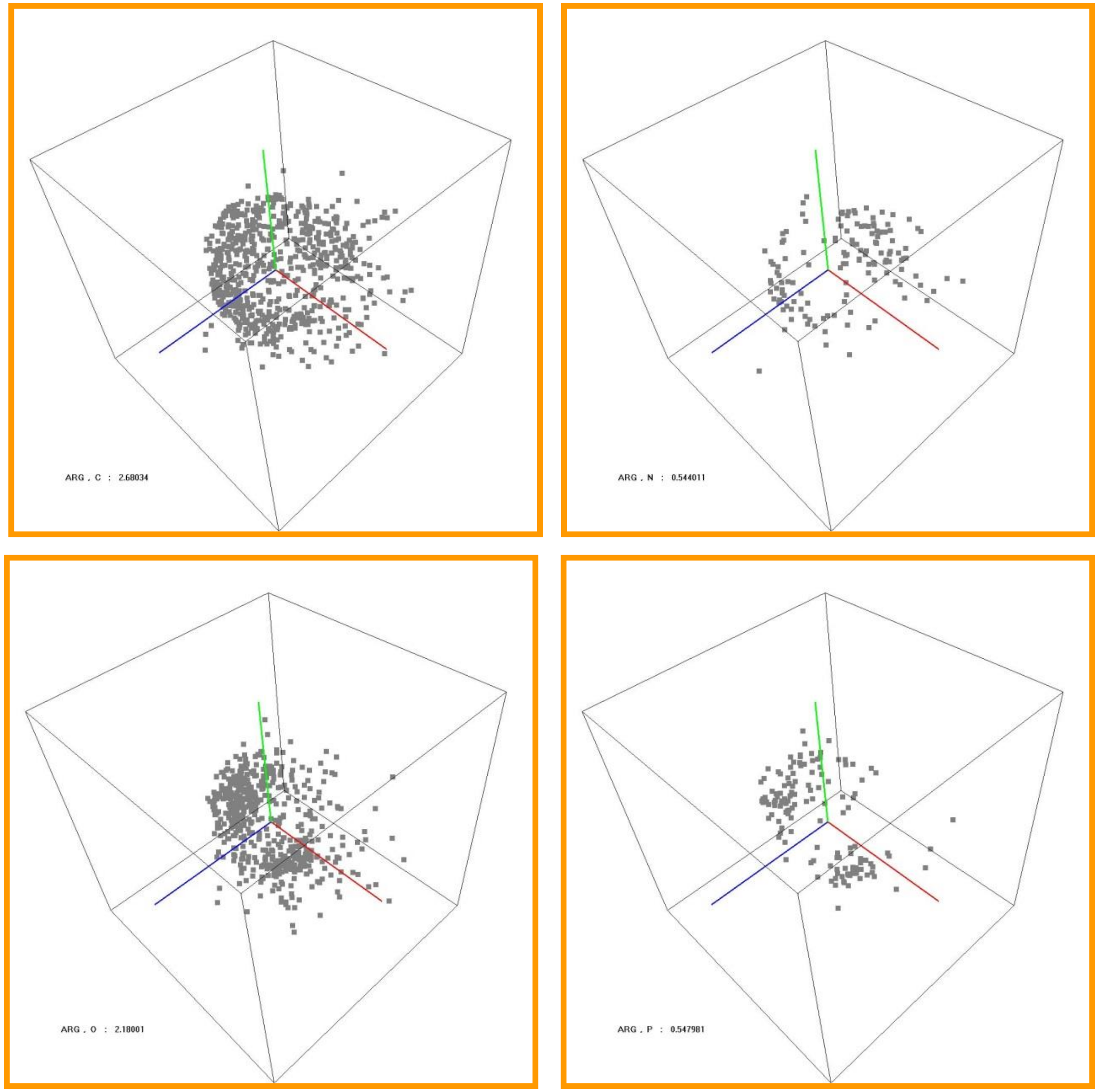
1mxb-1-A-ADP-385-_



4pfk-1-A-ADP-326-_

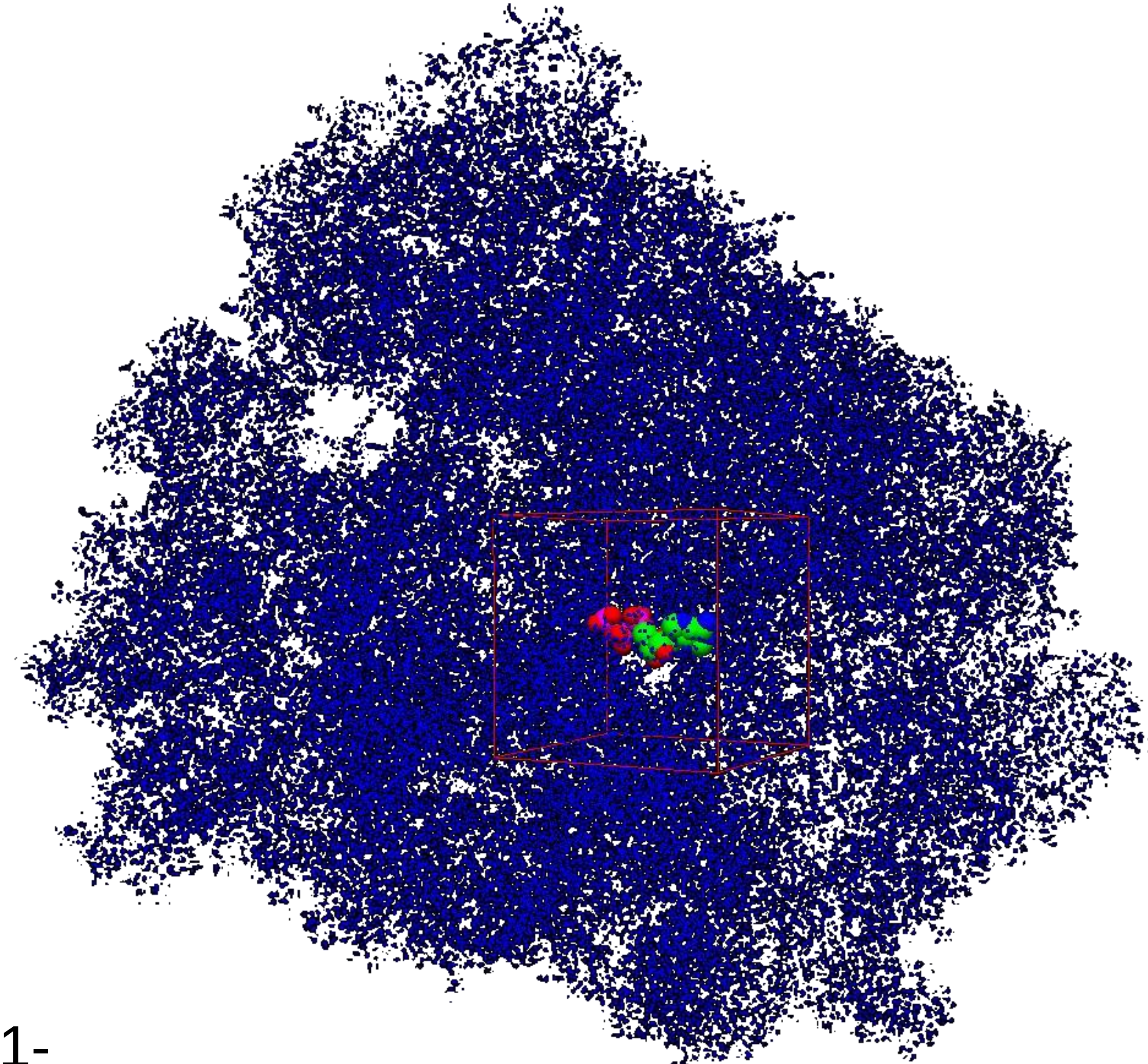
Residue Coordinate Frames

Binding Site Modeling



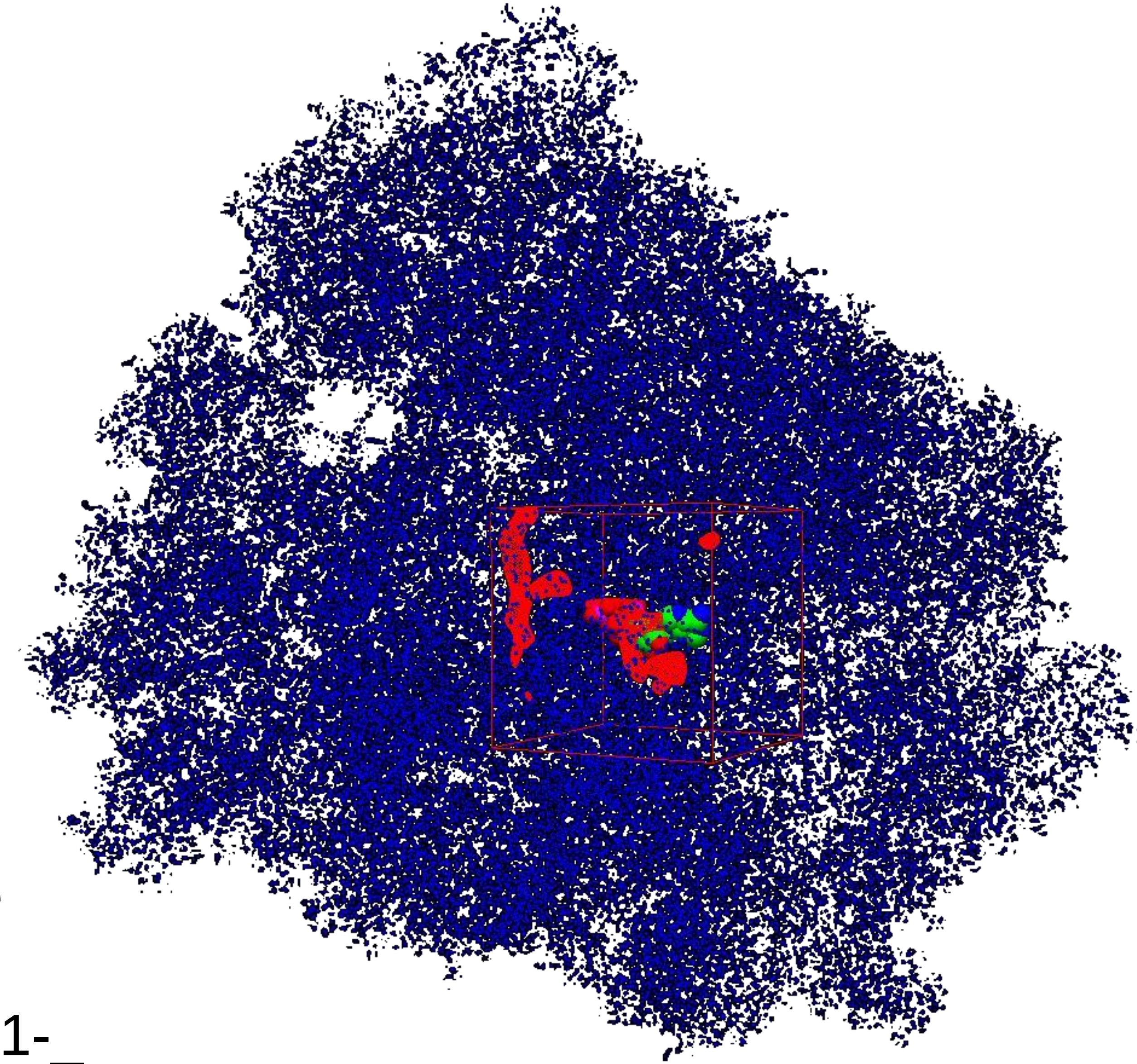
Trained
Density Distributions
for Argenine

Binding Site Modeling



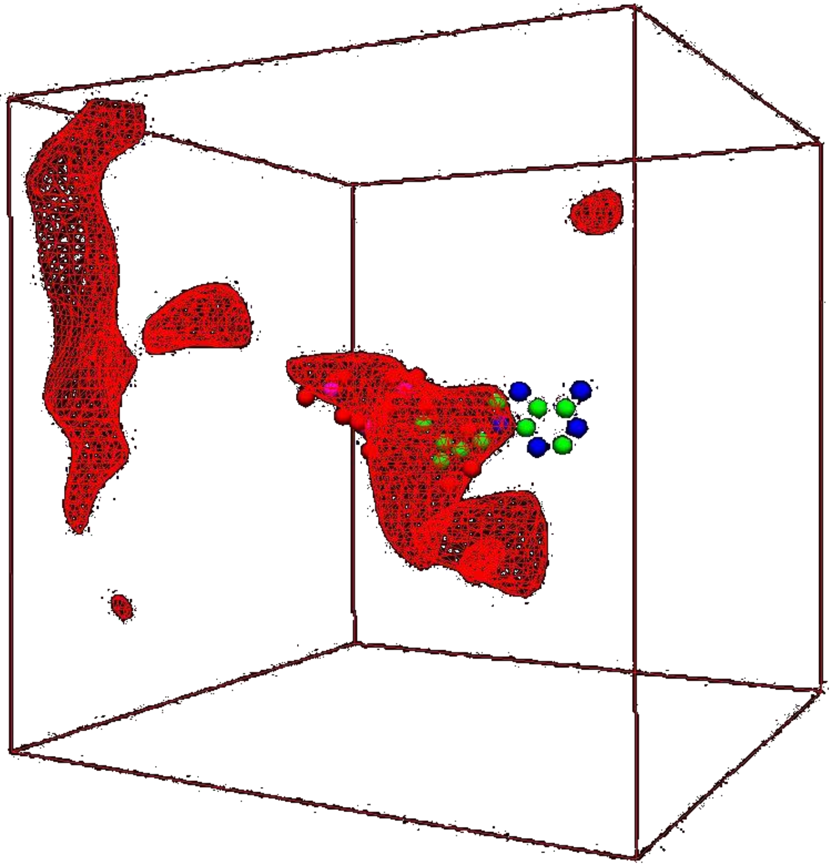
Predicted
Binding Site
Model for
1kp8-1-H-ATP-1-_

Binding Site Modeling



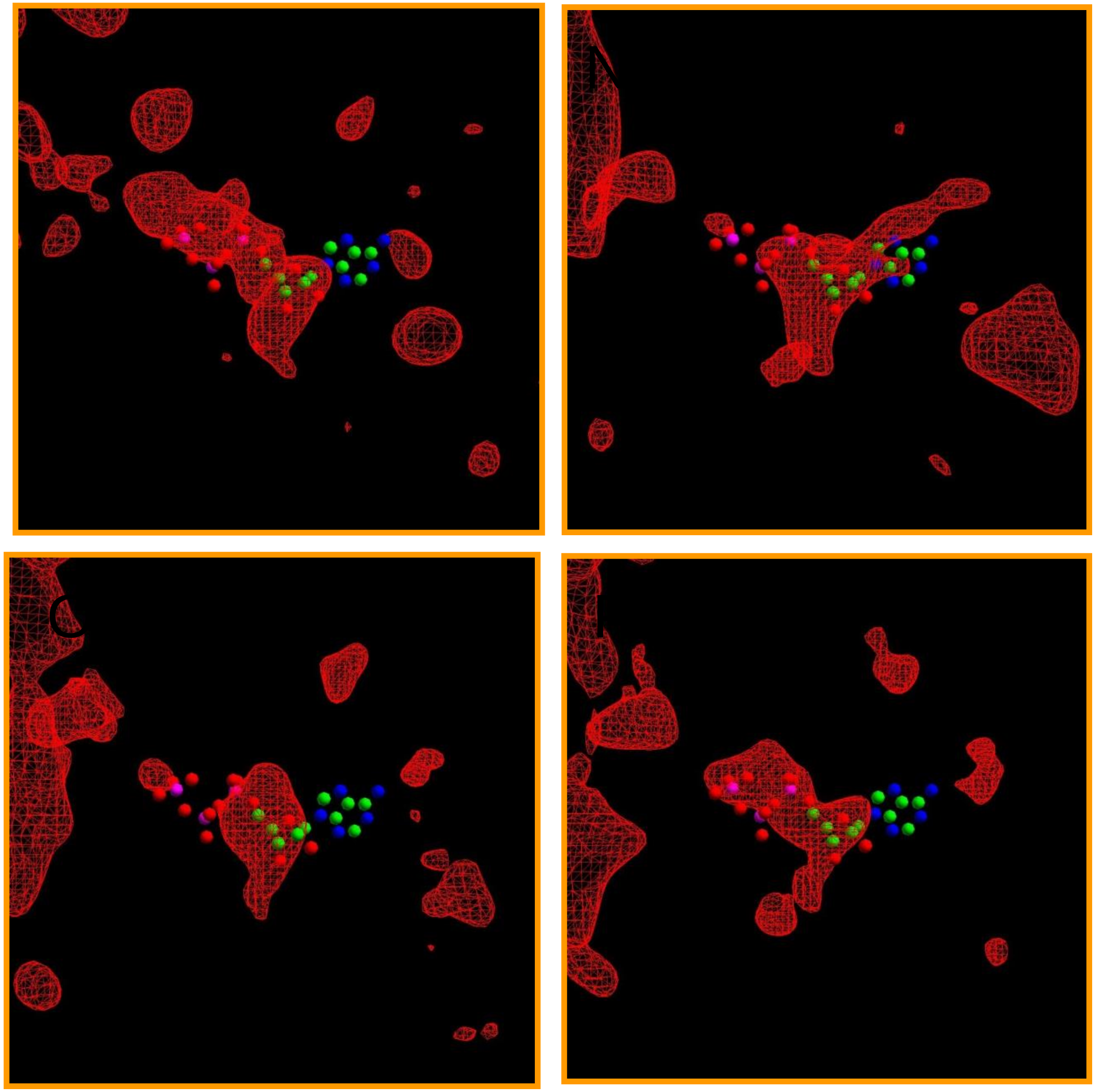
Predicted
Binding Site
Model for
1kp8-1-H-ATP-1-_

Binding Site Modeling



Predicted
Binding Site
Model for
1kp8-1-H-ATP-1-_

Binding Site Modeling



Predicted Binding Site Model for 1kp8-1-H-ATP-1-_