Comprendre les données visuelles à grande échelle

ENSIMAG
2019-2020

Karteek Alahari & Diane Larlus
19 décembre 2019

NAVER LABS EUROPE

Organisation du cours

- **17/10/19** cours 1 - Diane
- **24/10/19** cours 2 - Karteek
- **07/11/19** cours 3 - Karteek
- **14/11/19** cours 4 - Diane
- **28/11/19** cours 5 - Karteek
- **05/12/19** cours 6 - Karteek
- **12/12/19** **cours 7 - Diane**
- **19/12/19** cours 8 - Diane

Vacances d'hiver

- **09/01/20** cours 9 - Diane + présentation articles 1 & 2 + quizz
- **16/01/20** cours 10 - Diane + présentation articles 3 & 4 + quizz
- **23/01/20** cours 11 - Karteek + présentation articles 5 & 6 + quizz
- **30/01/20** cours 12 - Karteek + présentation articles 7 & 8 + quizz

Attention: la salle change régulièrement

Cours 8: Détection d'objets

Comprendre les données visuelles à grande échelle
19 novembre 2019

Première partie: méthodes de détection pré-CNNs

Comprendre les données visuelles à grande échelle
19 novembre 2019

Autres processus automatisables: la reconnaissance d’objet

Rappel cours 1 !

2) Catégorisation d’image

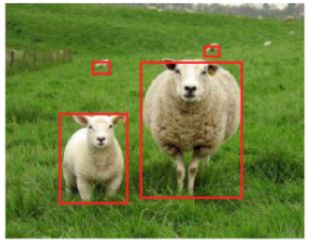
- Catégorie principale associée à l’image, ou réponse oui/non à une liste de catégories connues à l’avance

Sheep ?

yes

3) Détection d’objet

- Boite englobante pour toutes les instances d’une catégorie d’intérêt



Sheep ?

4

4) Segmentation d’objet, segmentation sémantique

- Localisation précise des objets au niveau du pixel

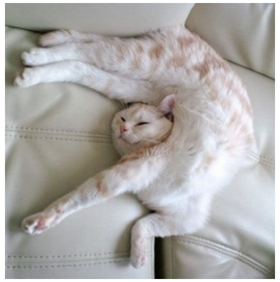
Sheep ?

NAVER LABS EUROPE

Rappel cours 1 !

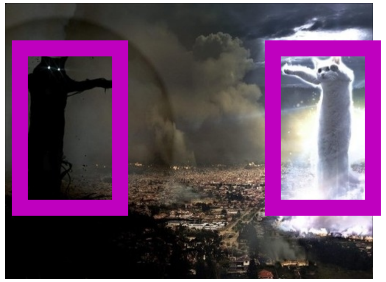
Difficultés de la modélisation des **catégories** d'objet

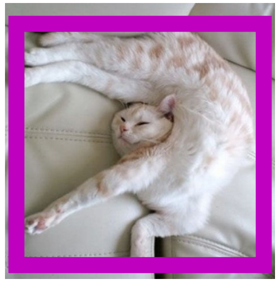
- Illumination, ombres
- Orientation et pose
- Fond texturé, distracteurs
- Occultations
- Variations intra-classe

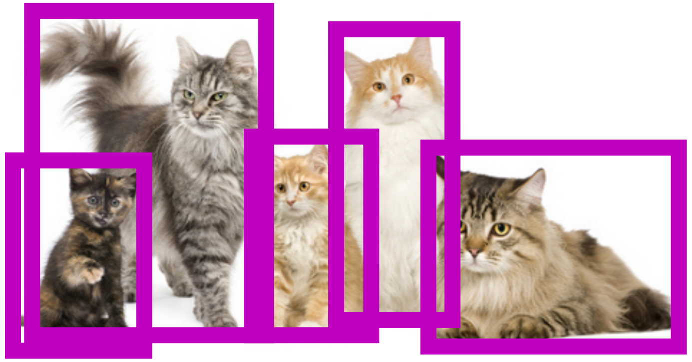


Difficultés de la **détection** des catégories d’objet

- Illumination, ombres
- Orientation et pose
- Fond texturé, distracteurs
- Occultations
- Variations intra-classe
- **Objets potentiellement très flexibles**







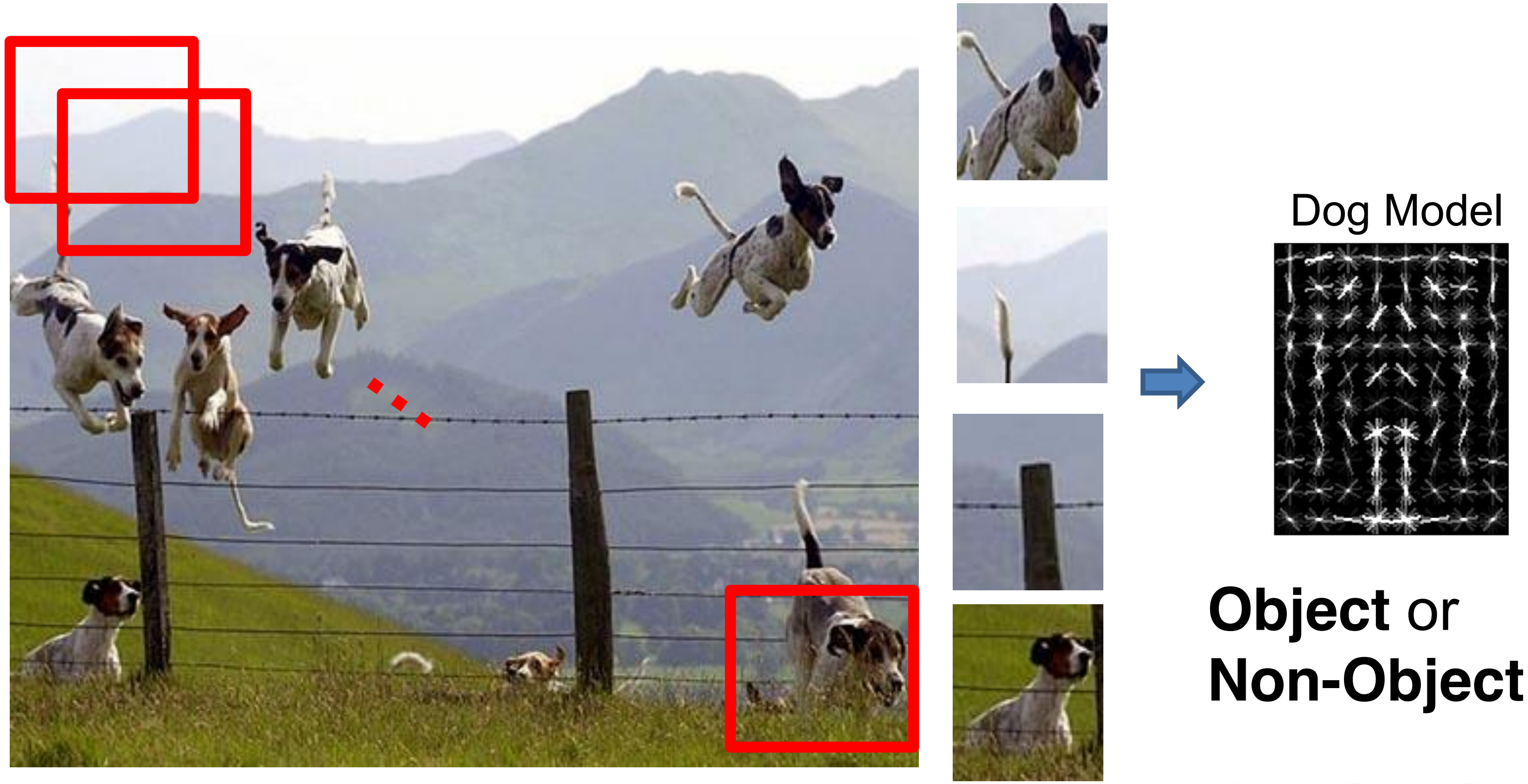
Credit

This lecture reuses slides from

- Derek Hoiem (University of Illinois at Urbana-Champaign)
  http://www.cs.illinois.edu/homes/dhoiem/
- Pedro Felzenszwalb (Brown University)
  http://www.cs.brown.edu/~pff/
- Fei-Fei Li & Justin Johnson & Serena Yeung (Sanford – cs231n_2019)
  http://cs231n.stanford.edu/

Object Category Detection

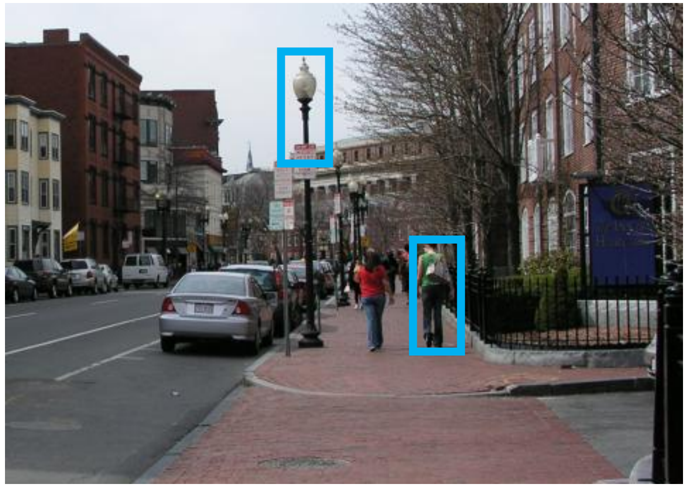
- Focus on object search: “Where is it?”
- Build templates that quickly differentiate object patch from background patch



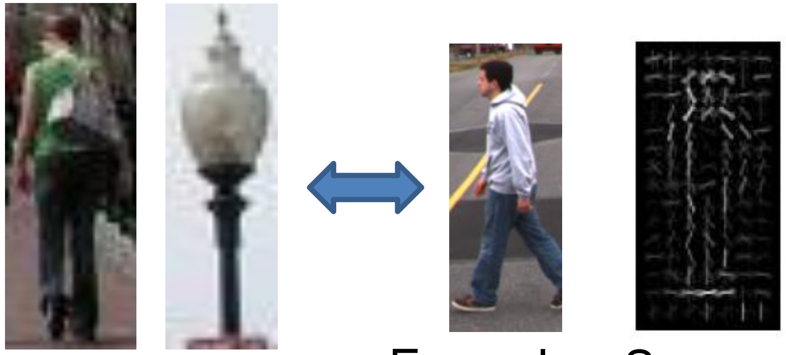
Object or **Non-Object?**

Basic Steps of Category Detection

1. Align
   - E.g., choose position, scale orientation
   - How to make this tractable?



2. Compare
   - Compute similarity to an example object or to a summary representation
   - Which differences in appearance are important?



Aligned Possible Objects

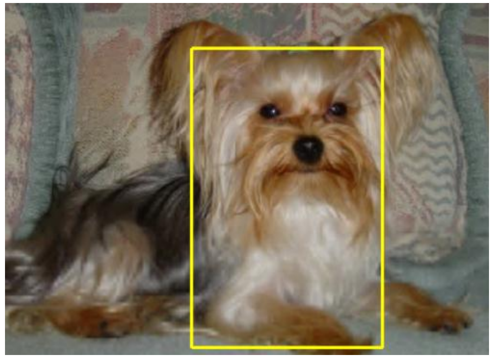
Exemplar

Summary

Challenges in modeling the non-object class

True Detections

Bad Localization



Confused with Similar Object

Misc. Background

Confused with Dissimilar Objects

General Process of Object Recognition

Specify Object Model

↓

Generate Hypotheses

↓

Score Hypotheses

↓

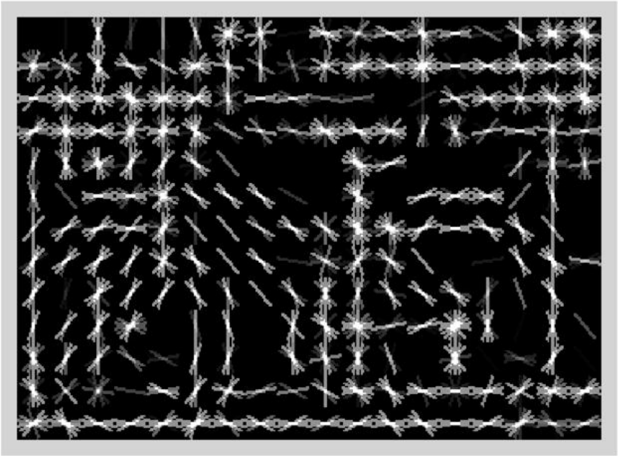
Resolve Detections

What are the object parameters?

Specifying an object model

1. Statistical Template in Bounding Box
   – Object is some (x,y,w,h) in image
   – Features defined wrt bounding box coordinates

Image

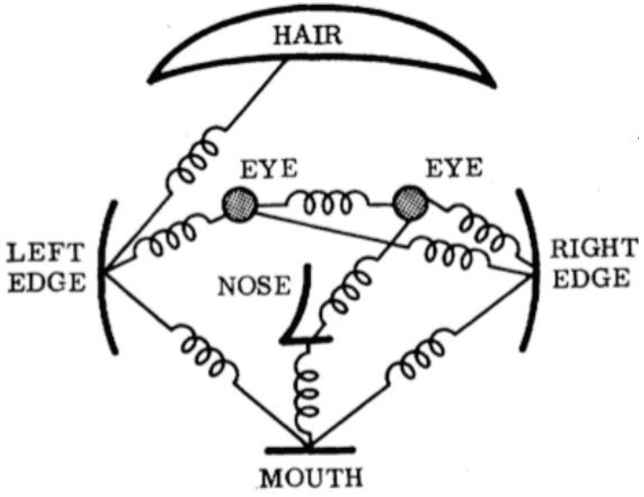


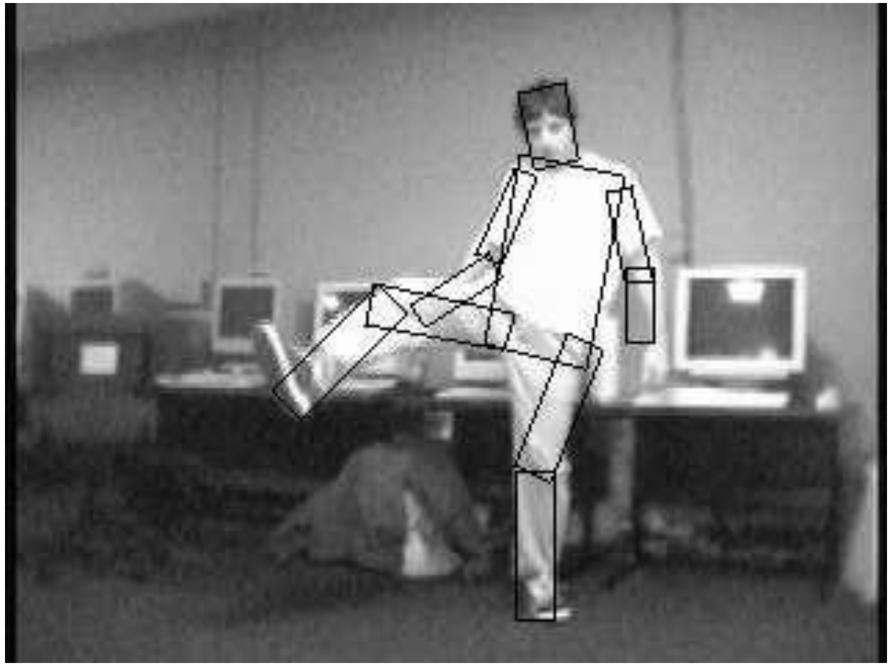
Template Visualization

Images from Felzenszwalb

Specifying an object model

2. Articulated parts model
   - Object is configuration of parts
   - Each part is detectable



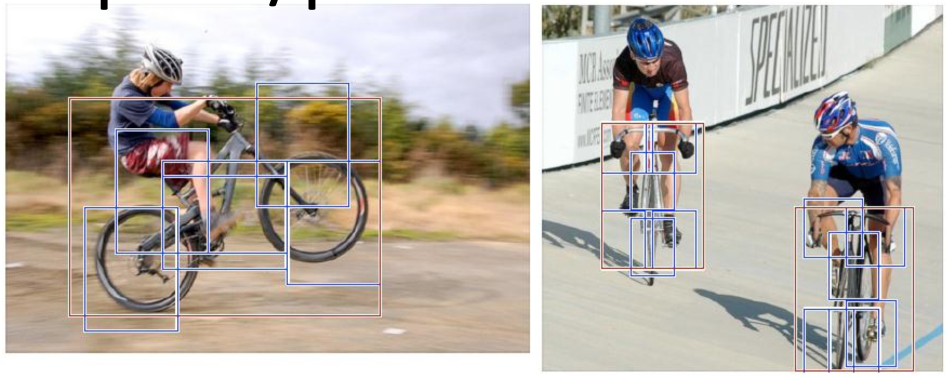


Images from Felzenszwalb

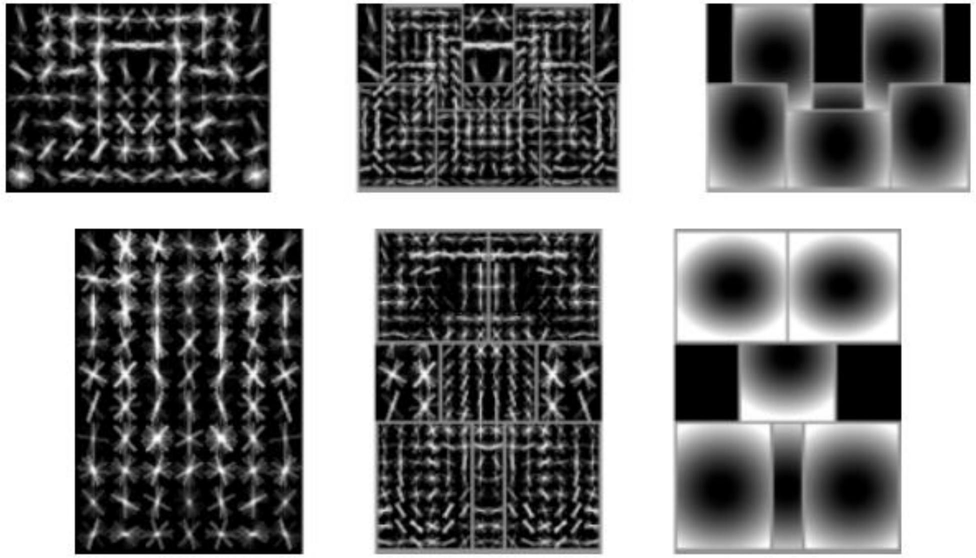
Specifying an object model

3. Hybrid template/parts model

Detections



Template Visualization



root filters
coarse resolution

part filters
finer resolution

deformation
models

Felzenszwalb et al. 2008

General Process of Object Recognition

Specify Object Model

↓

Generate Hypotheses

Propose an alignment of the model to the image

↓

Score Hypotheses

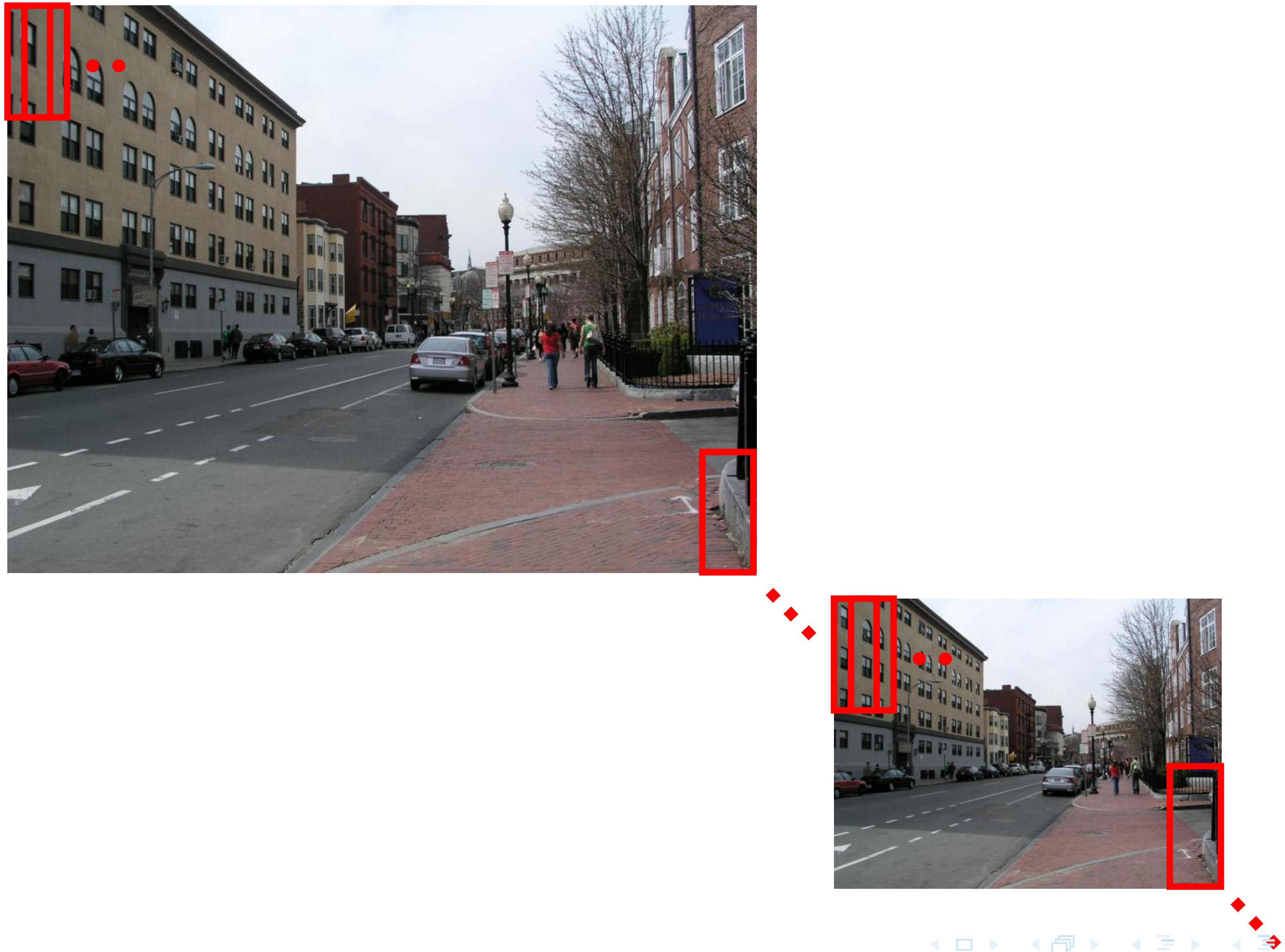
↓

Resolve Detections

Generating hypotheses

1. Sliding window
   – Test patch at each location and scale

Sliding window: a simple alignment solution

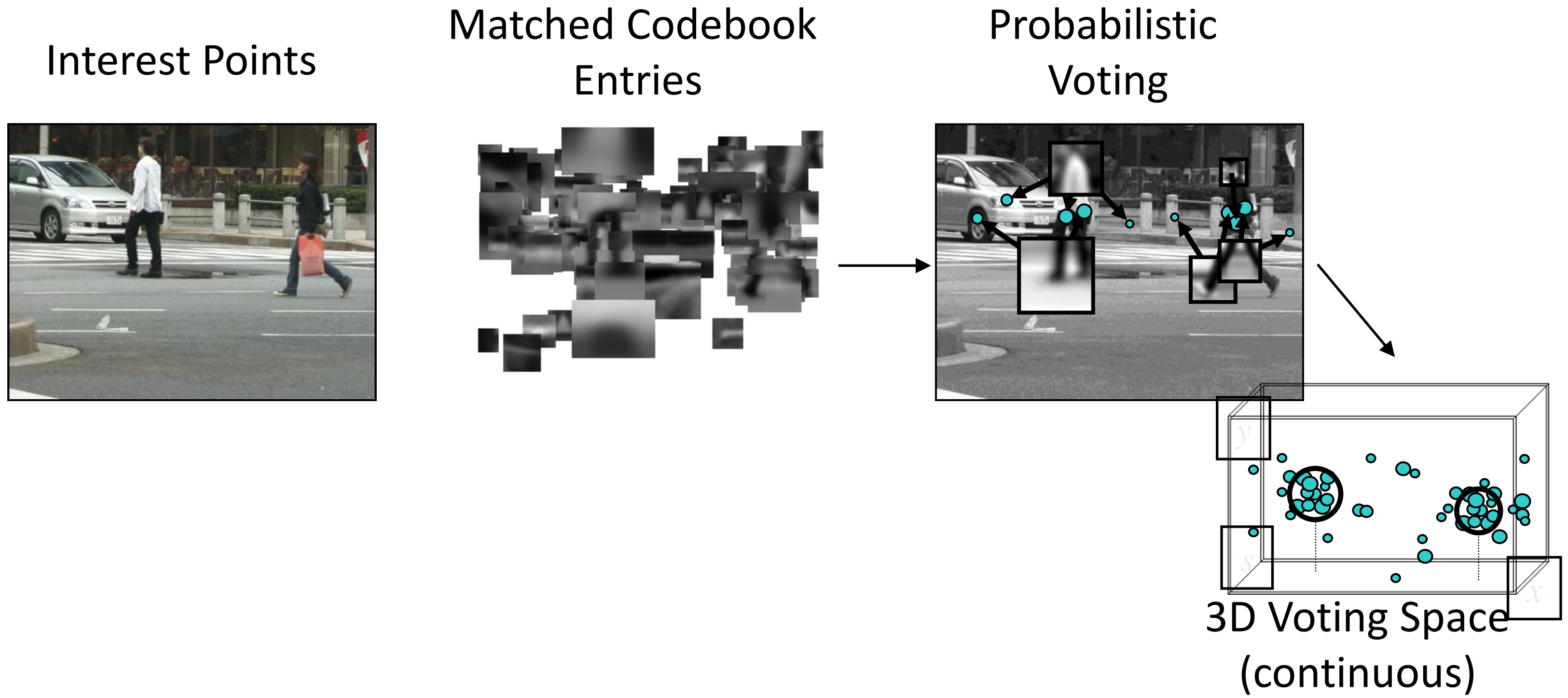


NAVER LABS EUROPE

Each window is separately classified

Generating hypotheses

2. Voting from patches/keypoints



ISM model by Leibe et al.

General Process of Object Recognition

Specify Object Model

↓

Generate Hypotheses

↓

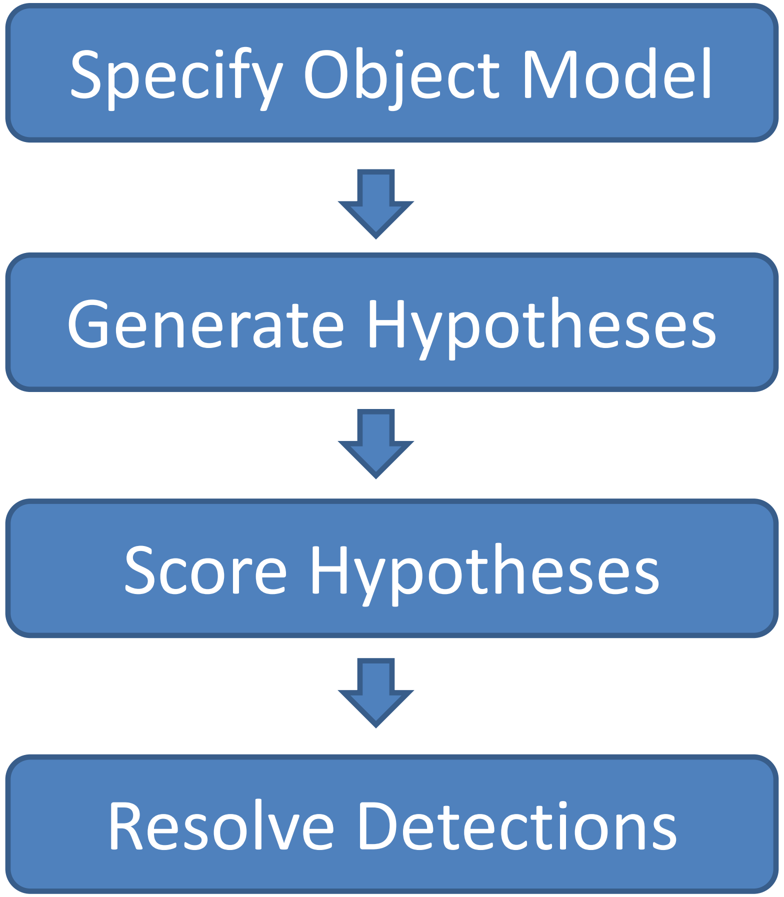
Score Hypotheses

↓

Resolve Detections

Mainly-gradient based features, usually based on summary representation, many classifiers

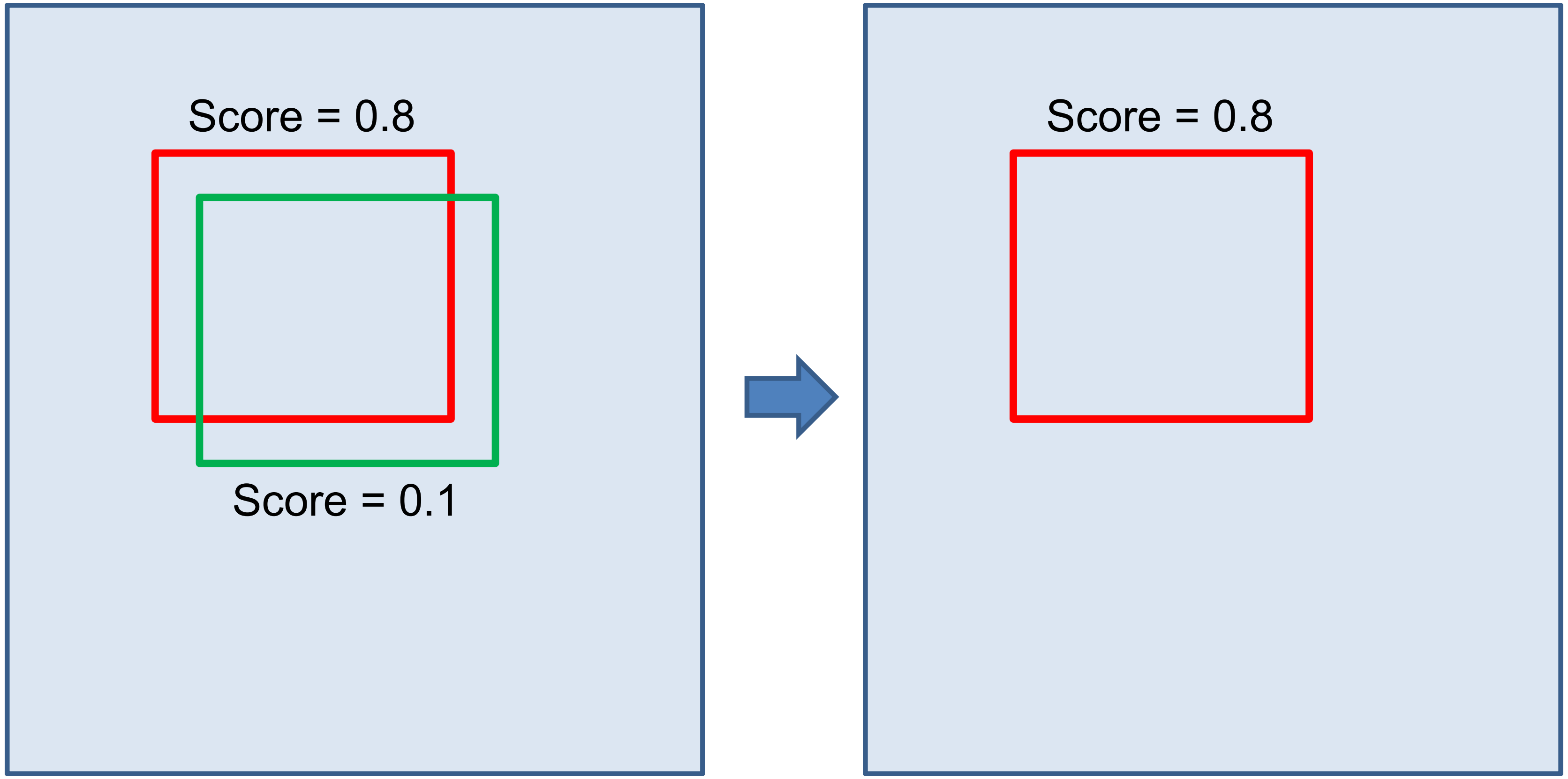
General Process of Object Recognition



Rescore each proposed object based on whole set

Resolving detection scores

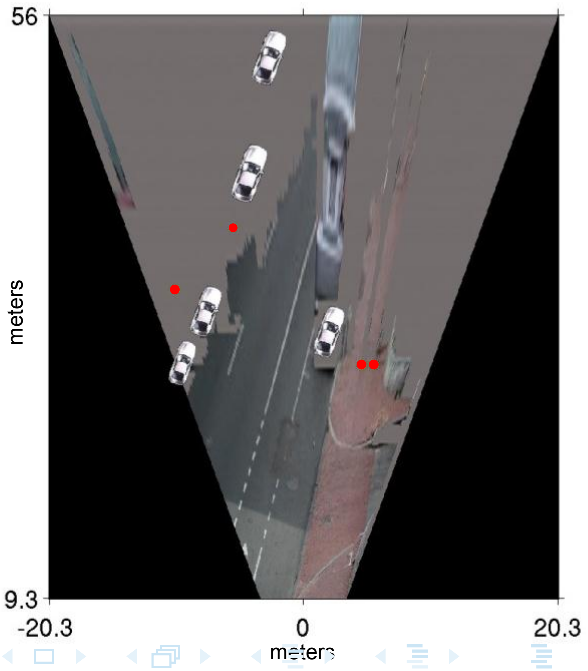
1. Non-max suppression



Resolving detection scores

2. Context/reasoning

(g) Car Detections: Local (h) Ped Detections: Local



Hoiem et al. 2006

Méthodes couvertes

Dans cette partie, nous allons couvrir les méthodes suivantes:

- Approches statistiques par *templates*:
  - Le détecteur de visages de Viola & Jones
  - Histogrammes de gradients orientés (HoG)
- Modèles déformables ou par parties
  - *Implicit Shape Model* (ISM)
  - *Pictorial Sturcture* (PS)
- Méthodes Hybrides
  - *Deformable Part Model* (DPM)

Le détecteur de visages de Viola & Jones

Comprendre les données visuelles à grande échelle

Cours 8: détection, 19 décembre 2019

NAVER LABS EUROPE

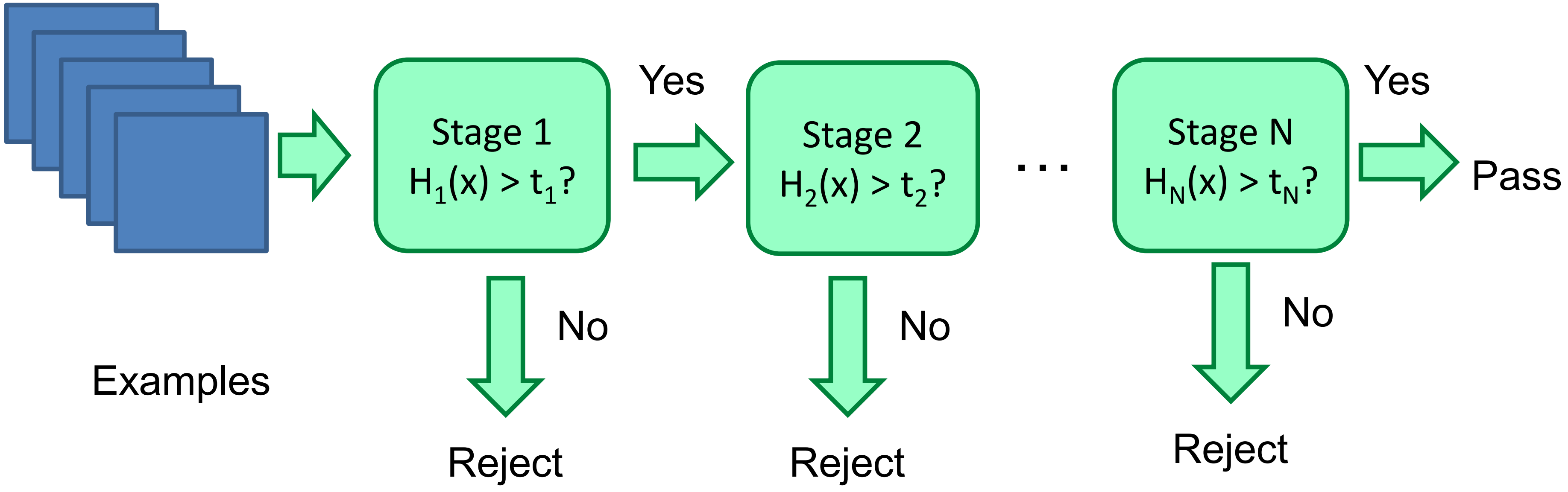
Viola-Jones sliding window detector

Fast detection through two mechanisms

- Quickly eliminate unlikely windows
- Use features that are fast to compute

Viola and Jones. Rapid Object Detection using a Boosted Cascade of Simple Features (2001).

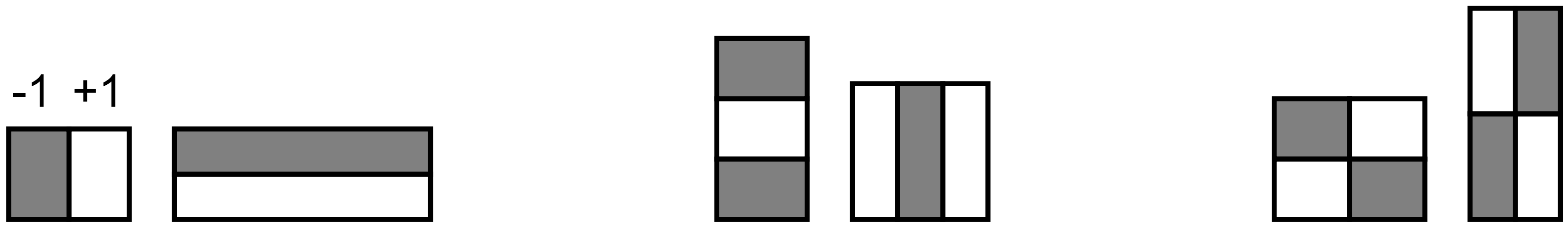
Cascade for Fast Detection



- Choose threshold for low false negative rate
- Fast classifiers early in cascade
- Slow classifiers later, but most examples don't get there

Features that are fast to compute

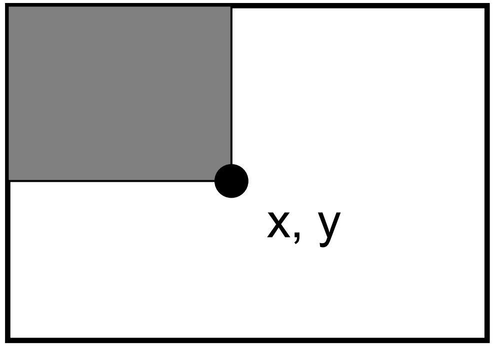
- “Haar-like features”
  - Differences of sums of intensity
  - Thousands, computed at various positions and scales within detection window



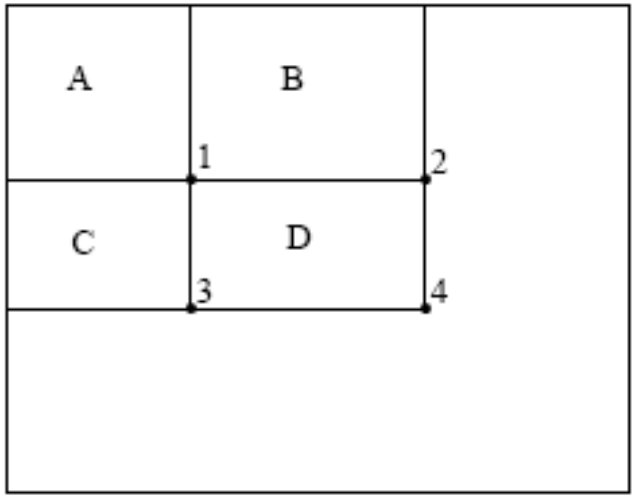
Two-rectangle features   Three-rectangle features   Etc.

Integral Images

- `ii = cumsum(cumsum(im, 1), 2)`



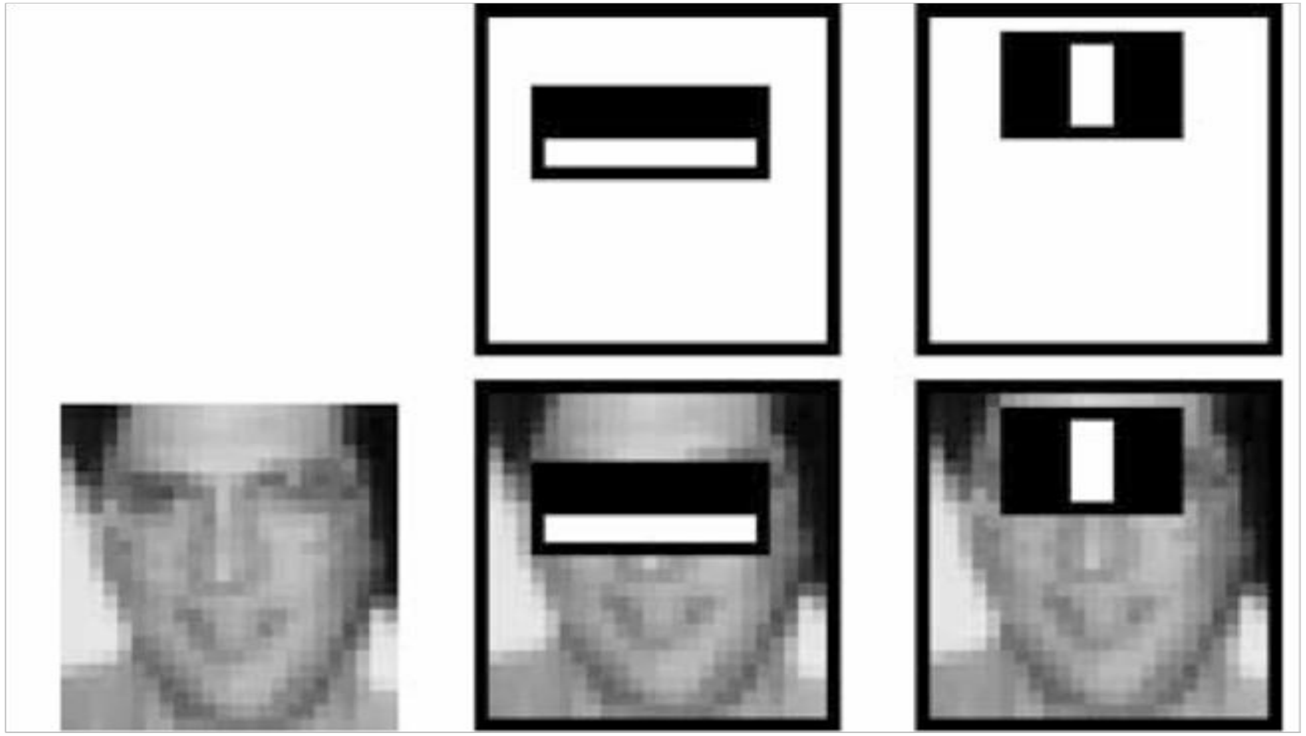
ii(x,y) = Sum of the values in the grey region



How to compute B-A?

How to compute A+D-B-C?

Top 2 selected features



Histogrammes de gradients orientés (HoG)

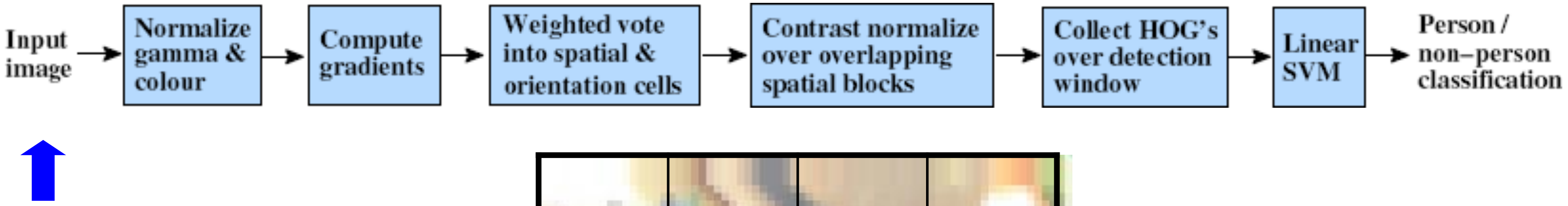
Comprendre les données visuelles à grande échelle

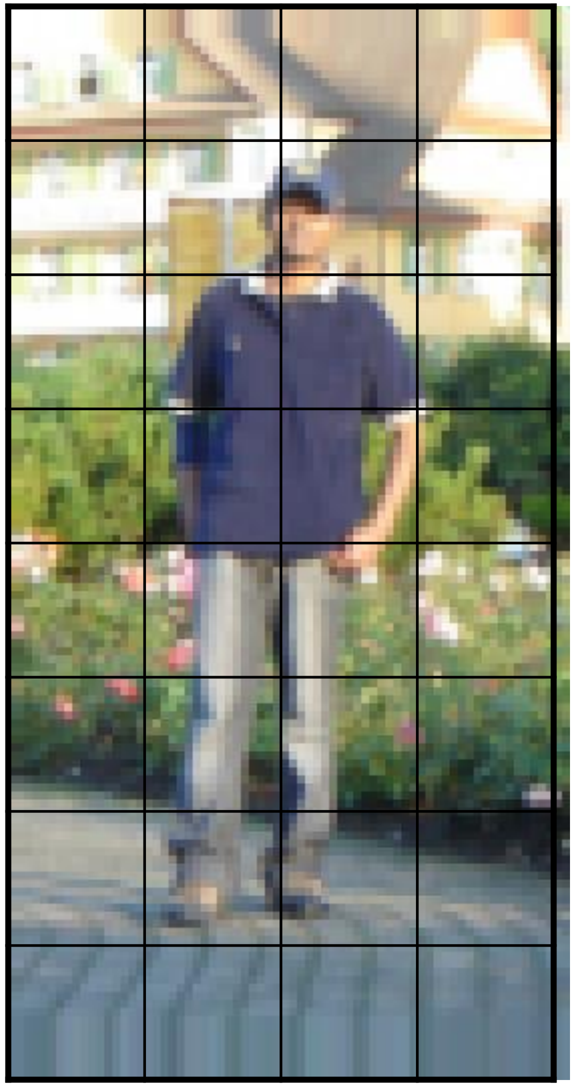
Cours 8: détection, 19 décembre 2019

Example: Dalal-Triggs pedestrian detector

1. Extract fixed-sized (64x128 pixel) window at each position and scale
2. Compute HOG (histogram of gradient) features within each window
3. Score the window with a linear SVM classifier
4. Perform non-maxima suppression to remove overlapping detections with lower scores

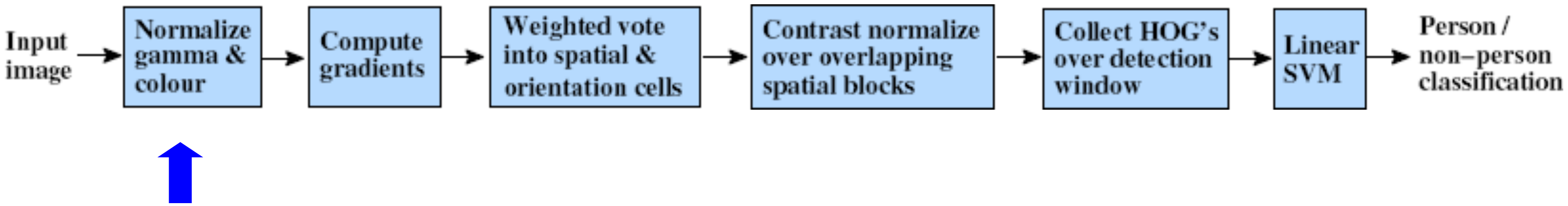
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05





Slides by Pete Barnum

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05



- Tested with
  - RGB
  - LAB
  - Grayscale

Slightly better performance vs. grayscale

- Gamma Normalization and Compression
  - Square root
  - Log

Very slightly better performance vs. no adjustment

Input image → Normalize gamma & colour → Compute gradients → Weighted vote into spatial & orientation cells → Contrast normalize over overlapping spatial blocks → Collect HOG's over detection window → Linear SVM → Person / non–person classification

Outperforms

-1	0	1

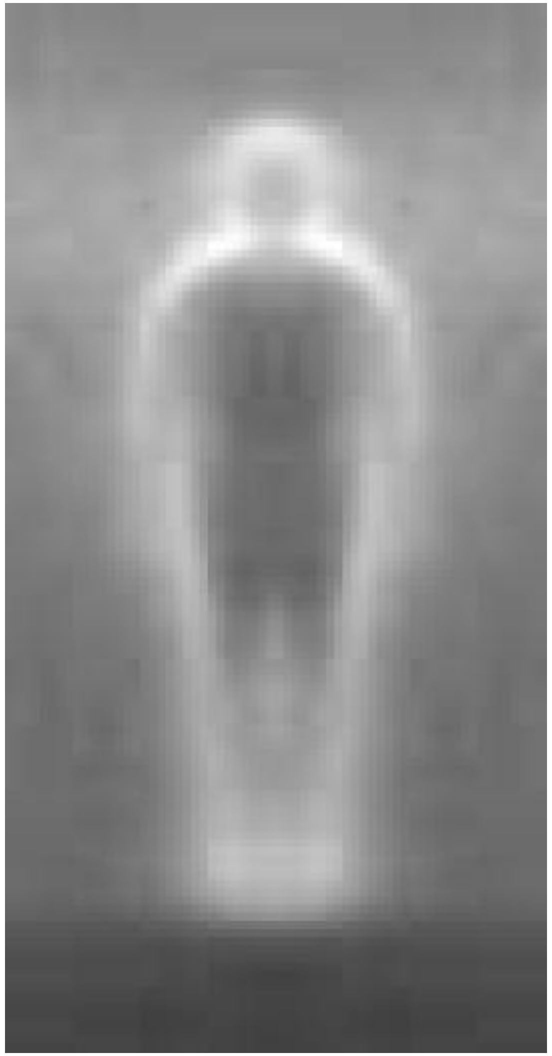
centered

-1	1

uncentered

1	-8	0	8	-1

cubic-corrected



0	1
-1	0

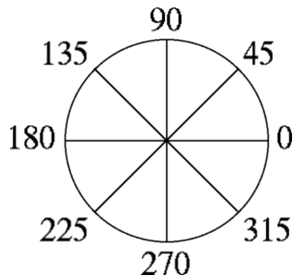
diagonal

-1	0	1
-2	0	2
-1	0	1

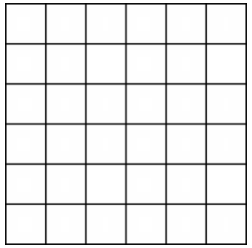
Sobel

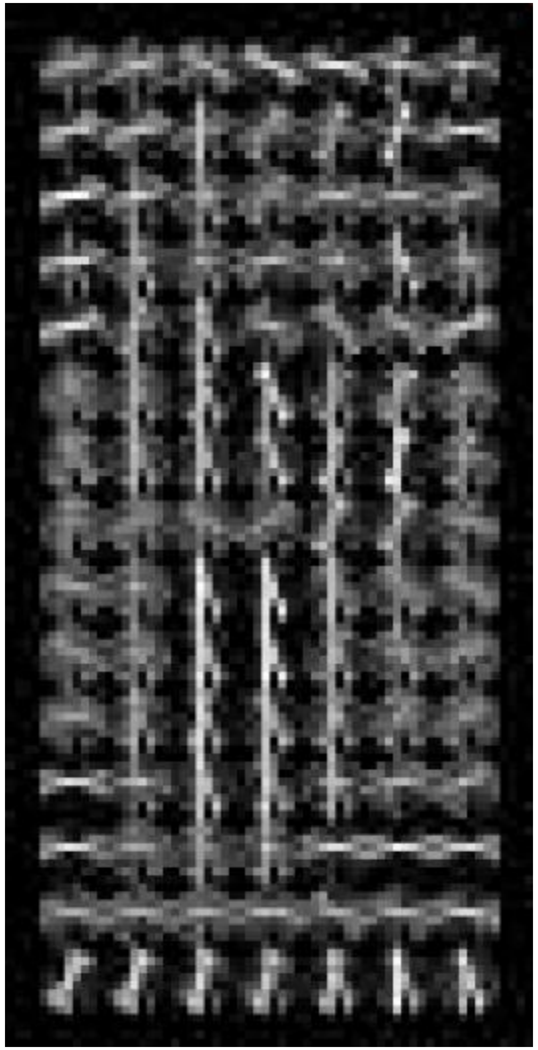
- Histogram of gradient orientations

Orientation: 9 bins (for unsigned angles)



Histograms in 8x8 pixel cells





  – Votes weighted by magnitude
  – Bilinear interpolation between cells

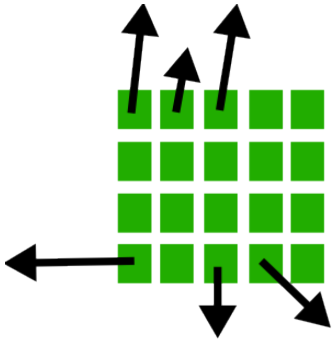
Slides by Pete Barnum

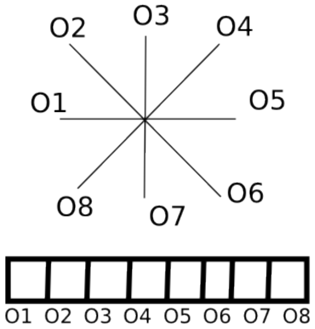
Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

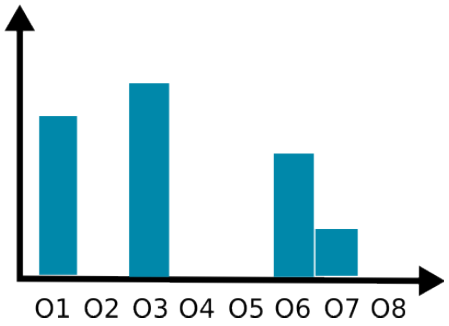
NAVER LABS EUROPE

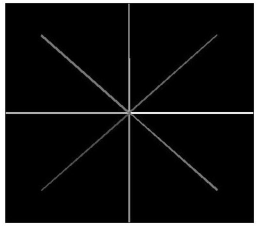
Building a cell descriptor

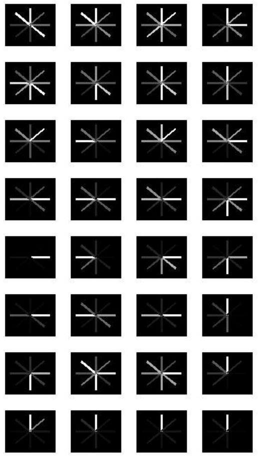
- Building a cell descriptor

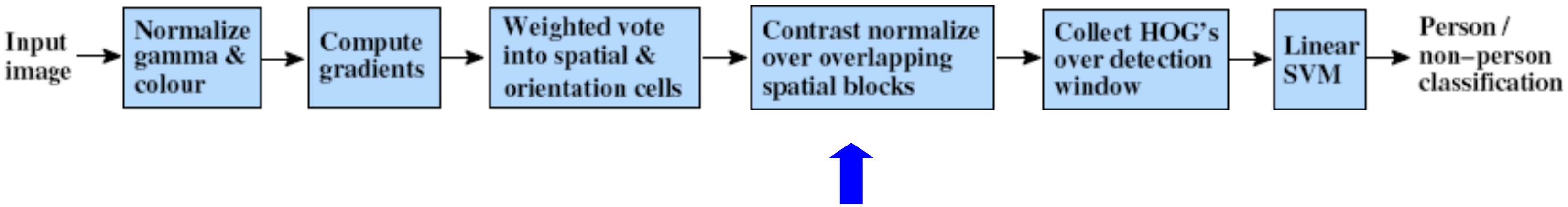




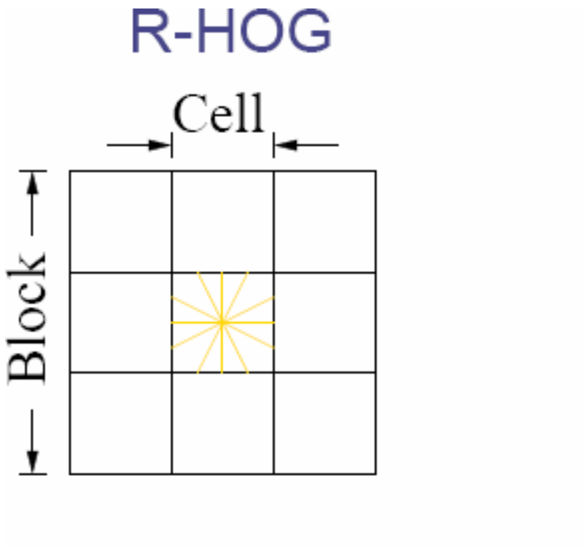






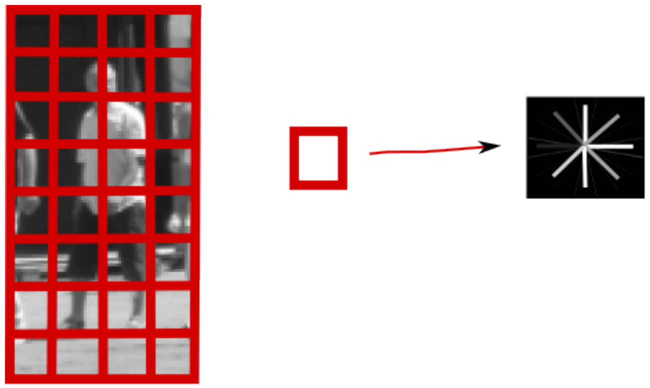


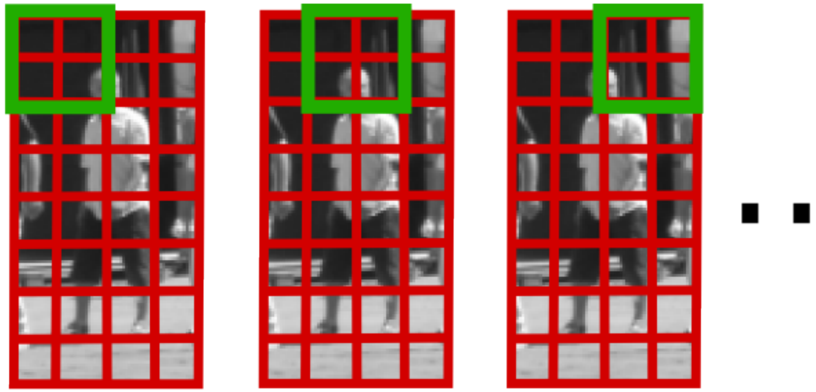
Normalize with respect to surrounding cells



$$L2 - norm : v \longrightarrow v/\sqrt{||v||_2^2 + \epsilon^2}$$

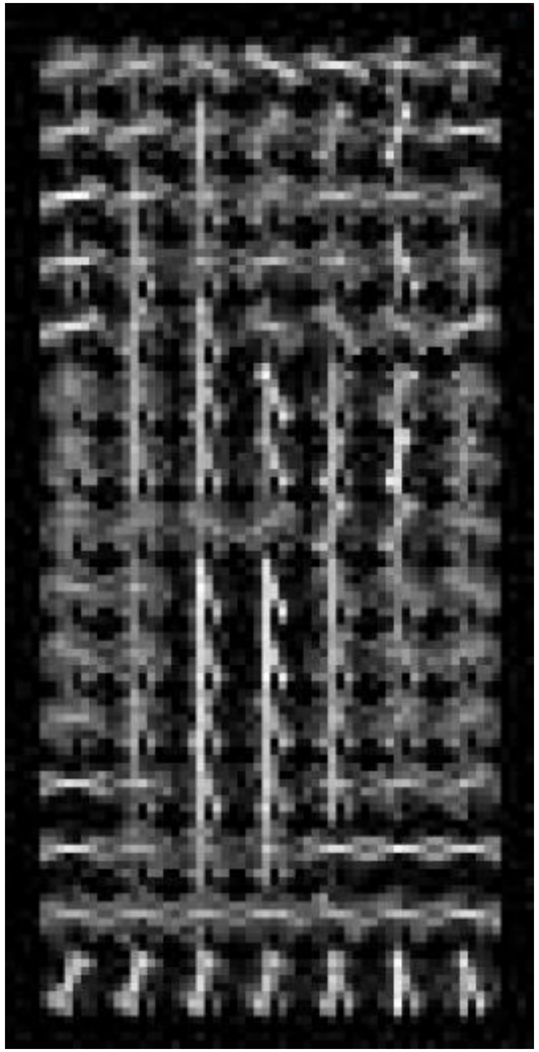
- Building a block descriptor





one cell

one block = 4 cells

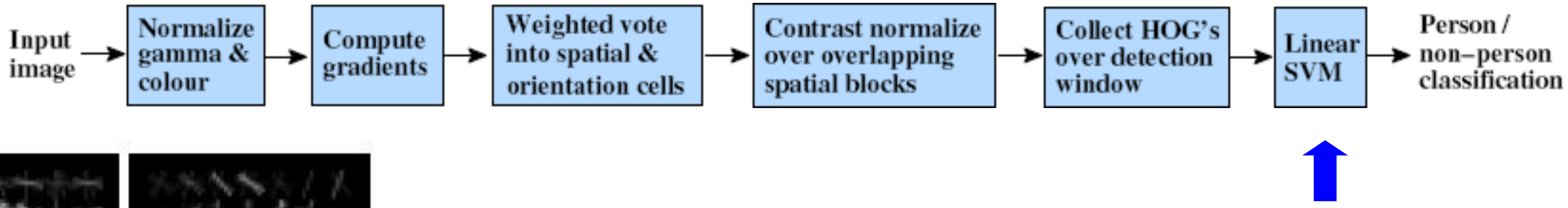
X= 

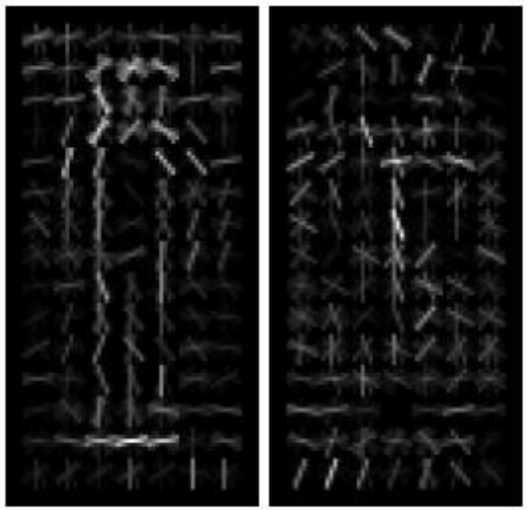
\# orientations

\# features = 15 x 7 x 9 x 4 = 3780

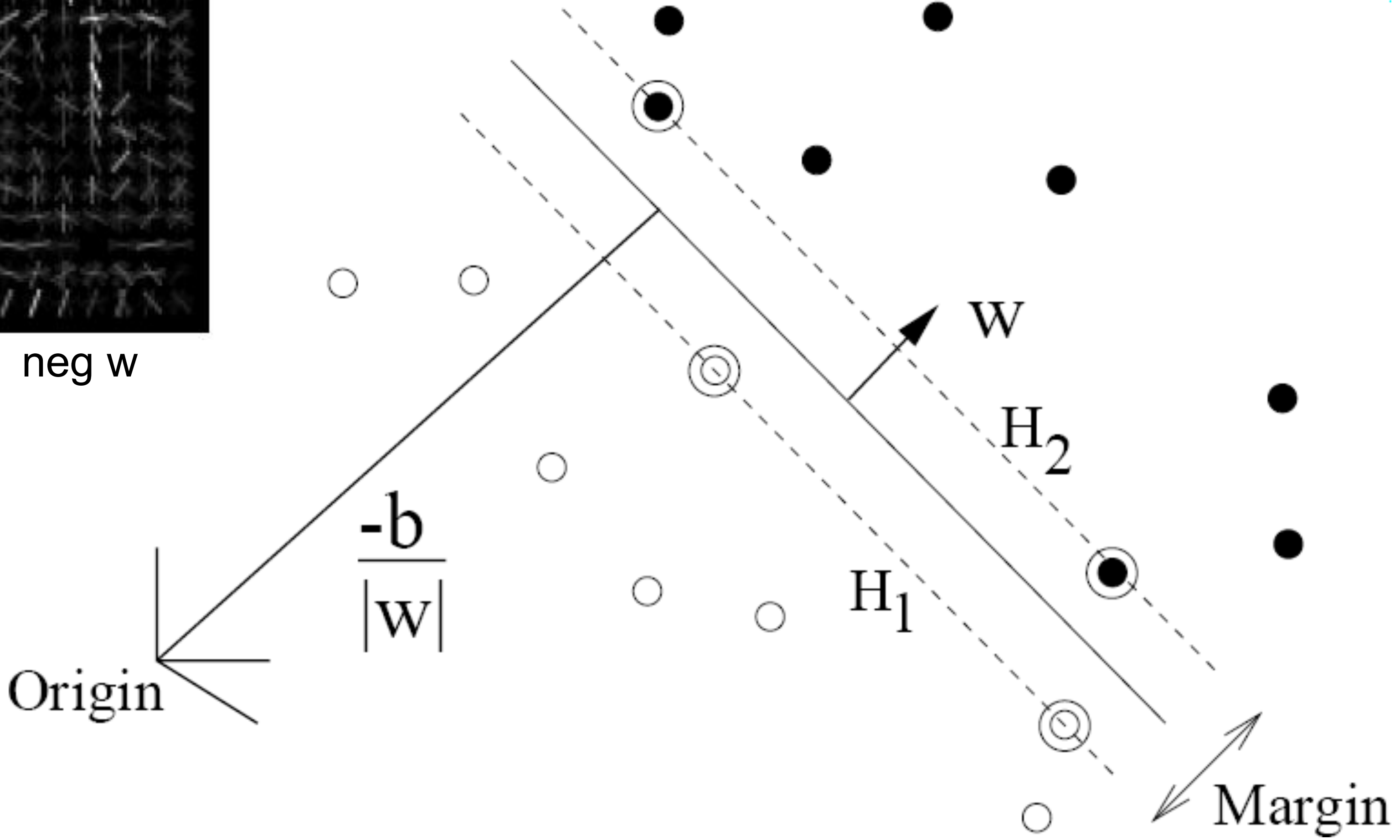
\# cells

\# normalizations by neighboring cells

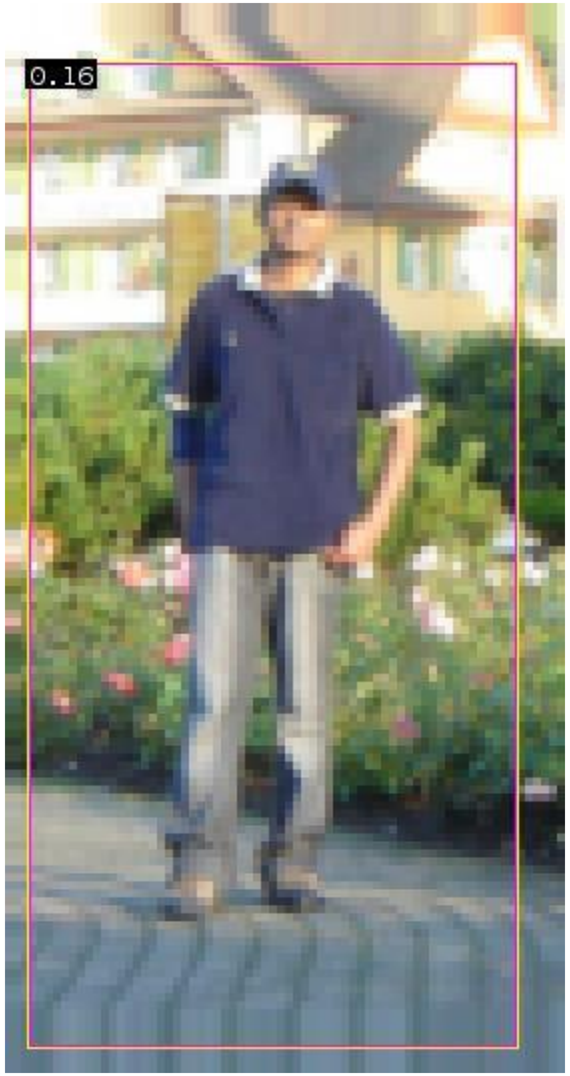




pos w   neg w



Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05



$$0.16 = w^T x - b$$

$$sign(0.16) = 1$$

$=>$ pedestrian

Approches statistiques par *templates*

Faiblesses et forces de ces méthodes

Strengths

- Works very well for non-deformable objects: faces, cars, upright pedestrians
- Fast detection

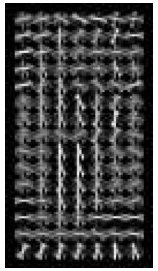
Weaknesses

- Not so well for highly deformable objects
- Not robust to occlusion
- Requires lots of training data

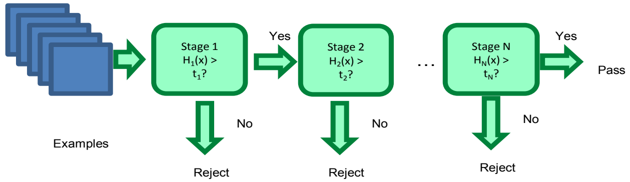
Approches statistiques par *templates*

Things to remember

- Sliding window for search
- Features based on differences of intensity (gradient, wavelet, etc.)
  - Excellent results require careful feature design



- Boosting for feature selection
- Integral images, cascade for speed



- Bootstrapping to deal with many, many negative examples

Modèles déformables ou par parties

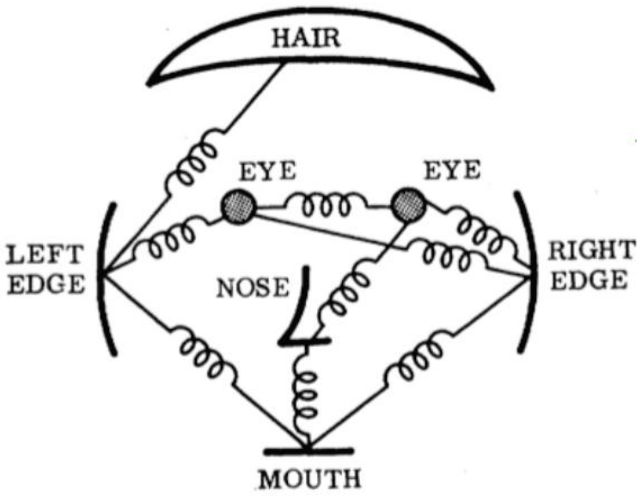
Comprendre les données visuelles à grande échelle

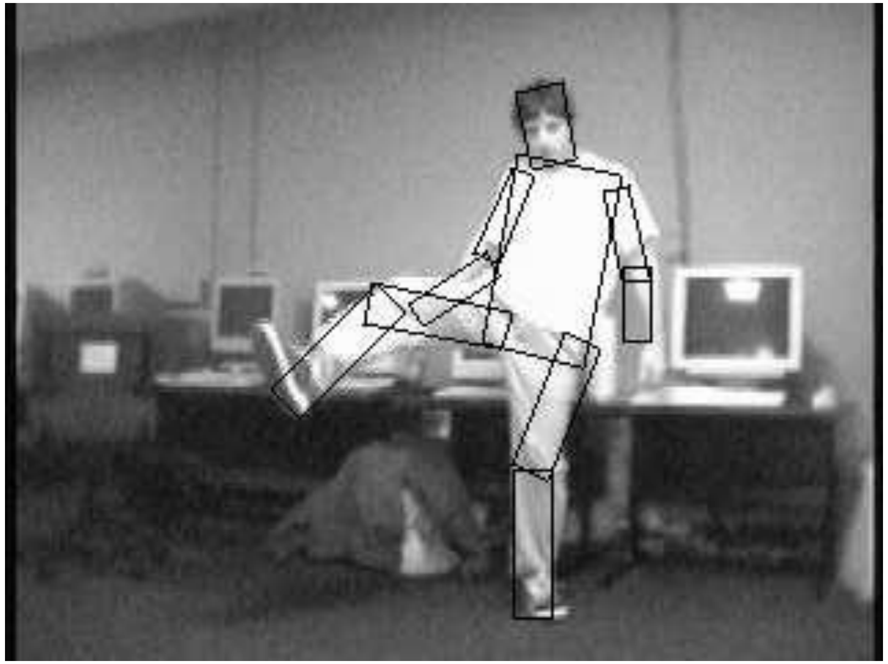
Cours 8: détection, 19 décembre 2019

NAVER LABS EUROPE

Object models: this class

- Articulated parts model
  - Object is configuration of parts
  - Each part is detectable





Images from Felzenszwalb

Parts-based Models

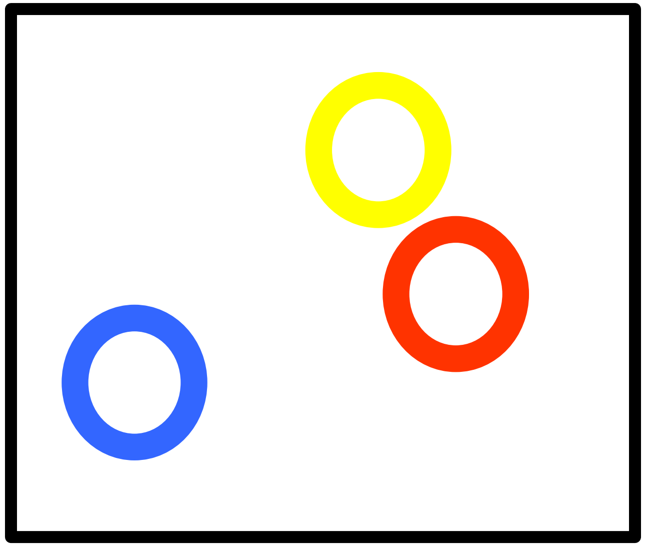
Define object by collection of parts modeled by

1. Appearance
2. Spatial configuration

Slide credit: Rob Fergus

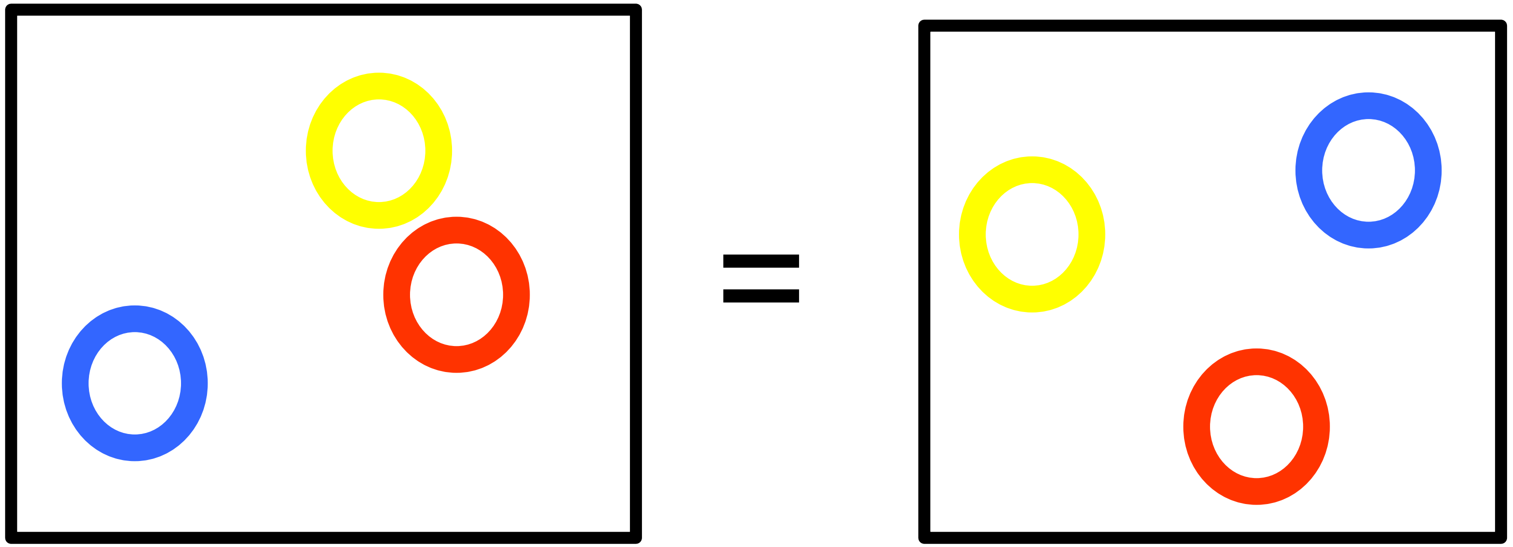
How to model spatial relations?

- One extreme: fixed template



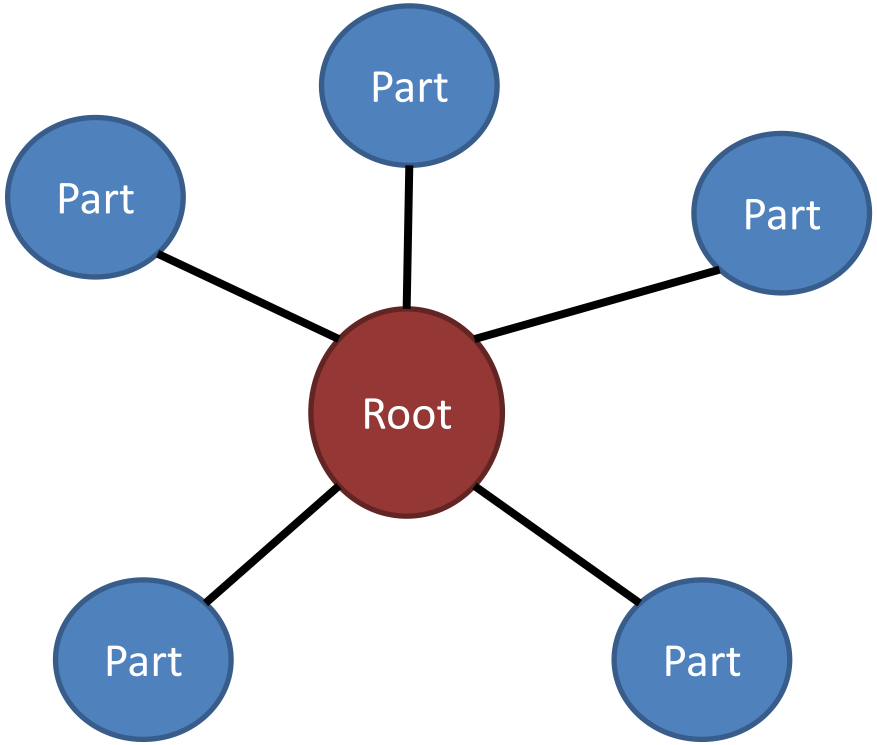
How to model spatial relations?

- Another extreme: bag of words



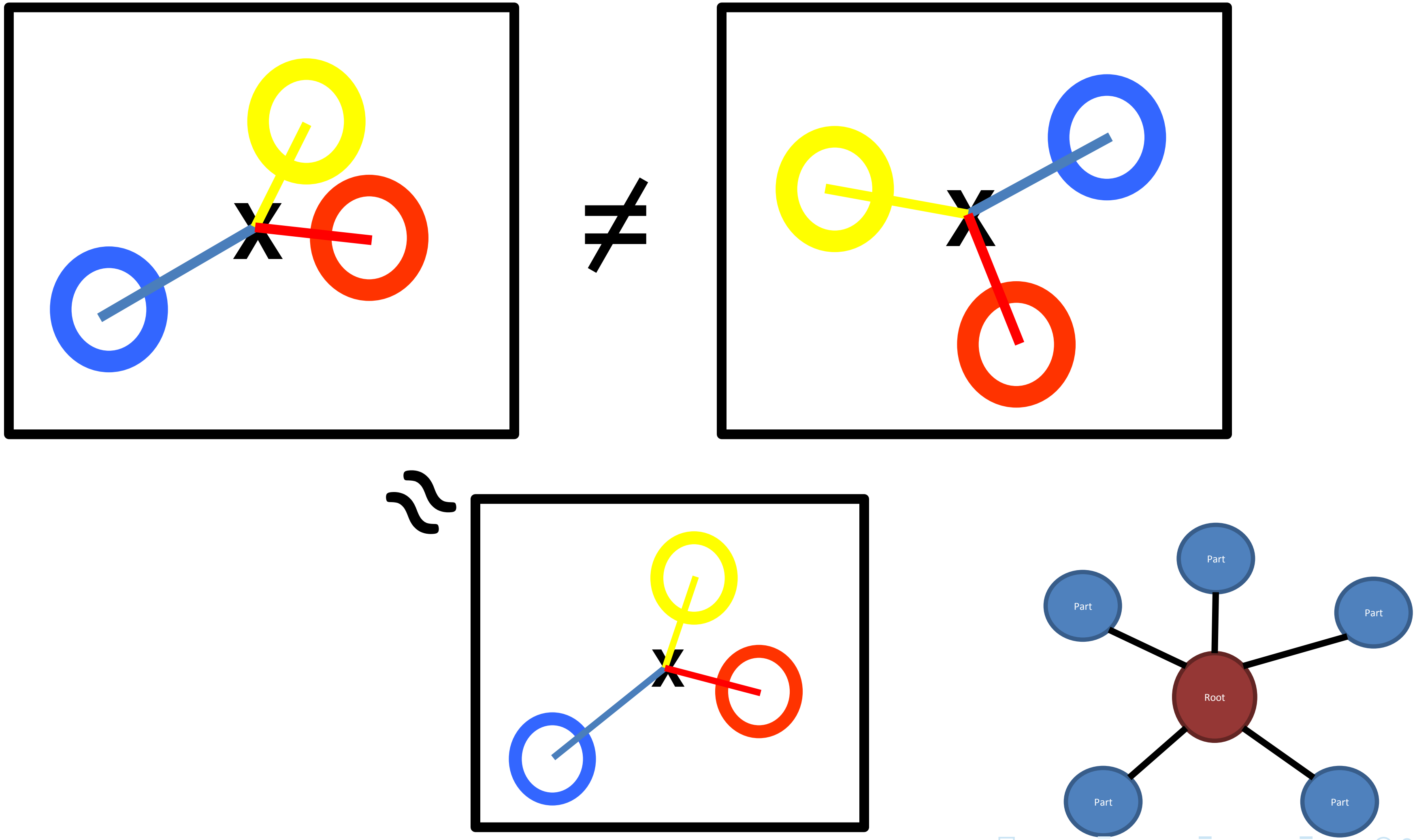
How to model spatial relations?

- Star-shaped model



How to model spatial relations?

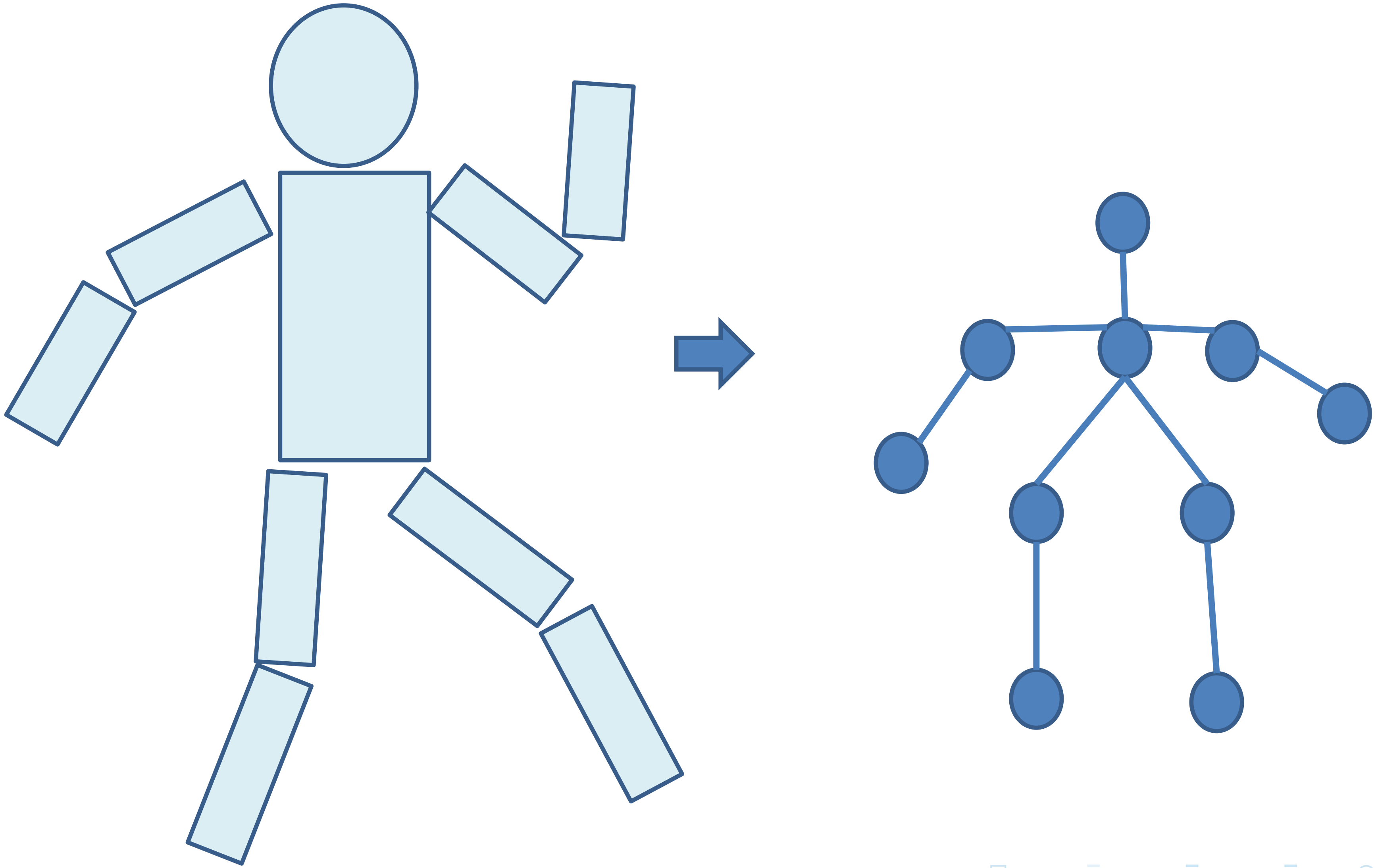
- Star-shaped model



NAVER LABS EUROPE

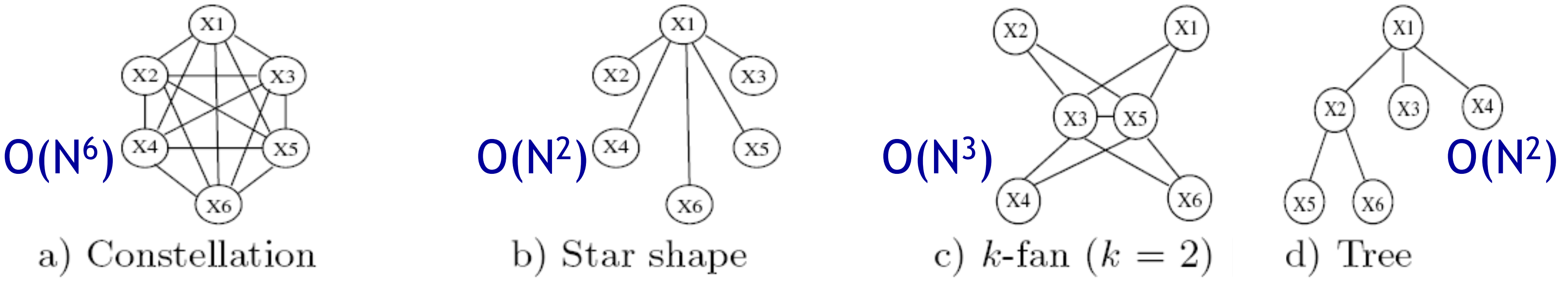
How to model spatial relations?

- Tree-shaped model



How to model spatial relations?

- Many others...

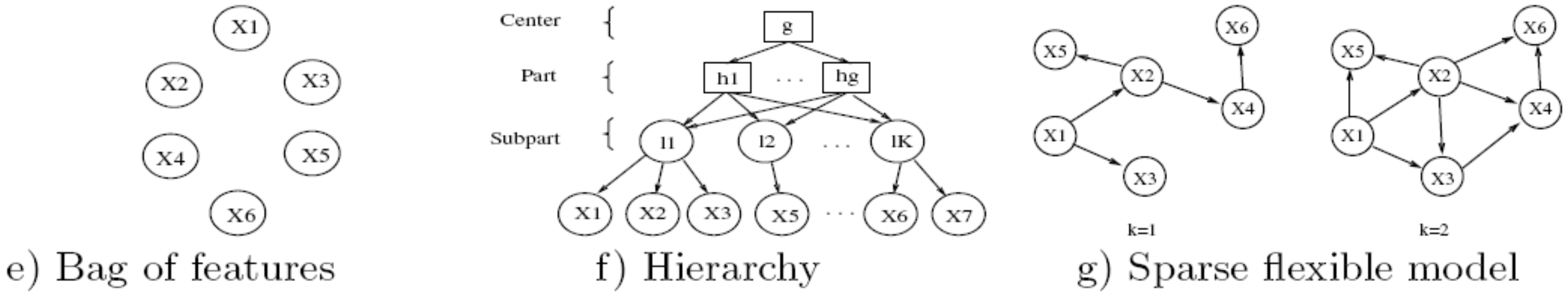


Fergus et al. '03
Fei-Fei et al. '03

Leibe et al. '04, '08
Crandall et al. '05
Fergus et al. '05

Crandall et al. '05

Felzenszwalb &
Huttenlocher '05



Csurka '04
Vasconcelos '00

Bouchard & Triggs '05

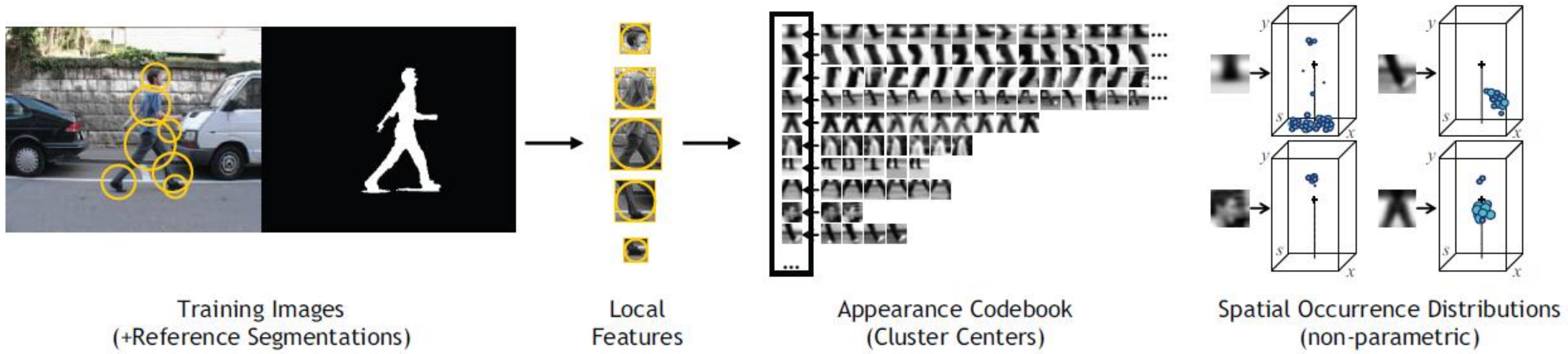
Carneiro & Lowe '06

from [Carneiro & Lowe, ECCV'06]

ISM: Implicit Shape Model

Training overview

- Start with bounding boxes and (ideally) segmentations of objects
- Extract local features (e.g., patches or SIFT) at interest points on objects
- Cluster features to create codebook
- Record relative bounding box and segmentation for each codeword

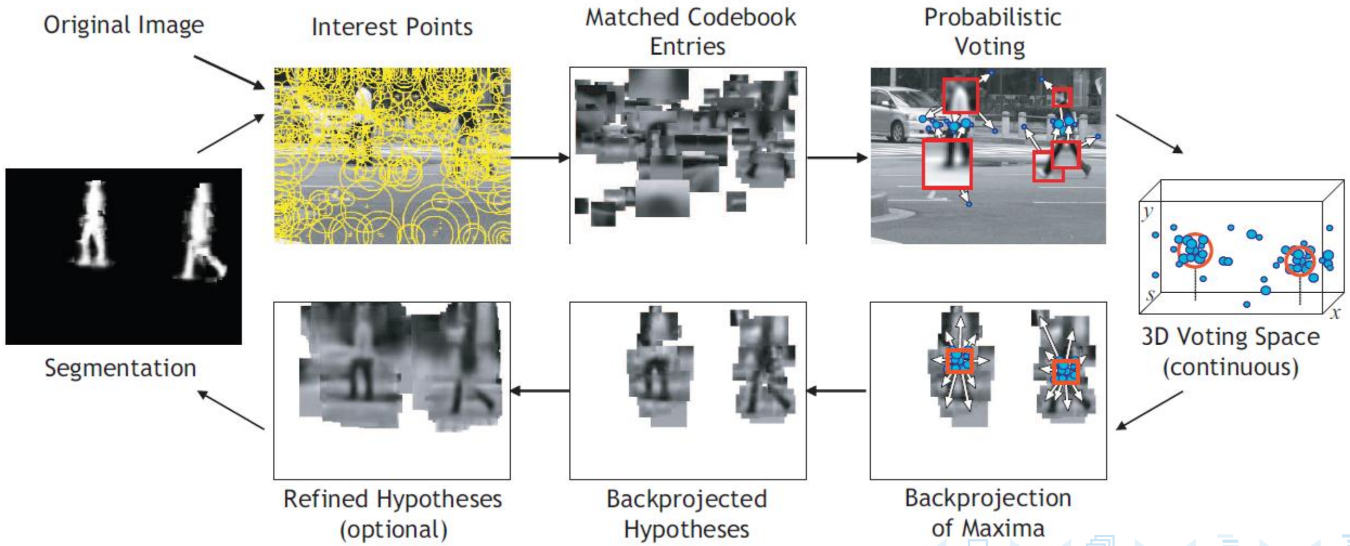


NAVER LABS EUROPE

ISM: Implicit Shape Model

Testing overview

- Extract interest points in test image
- Softly match to codebook entries
- Each matched codeword votes for object bounding box
- Compute modes of votes using mean-shift
- Check which codewords voted for modes
- Refine



NAVER LABS EUROPE

Deformable Part Model (DPM)

Comprendre les données visuelles à grande échelle

Cours 8: détection, 19 décembre 2019

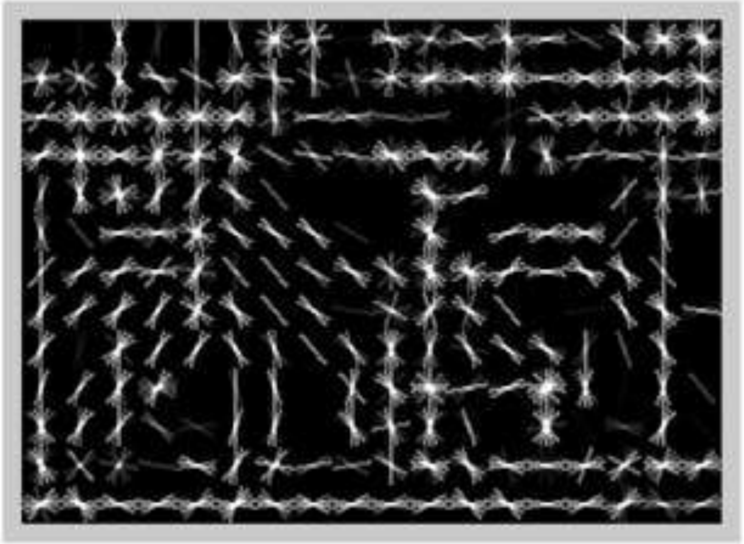
Starting point: sliding window classifiers

Feature vector
$x = [\ldots, \ldots, \ldots, \ldots]$

- Detect objects by testing each subwindow
  - Reduces object detection to binary classification
  - Dalal & Triggs: HOG features + linear SVM

NAVER LABS EUROPE

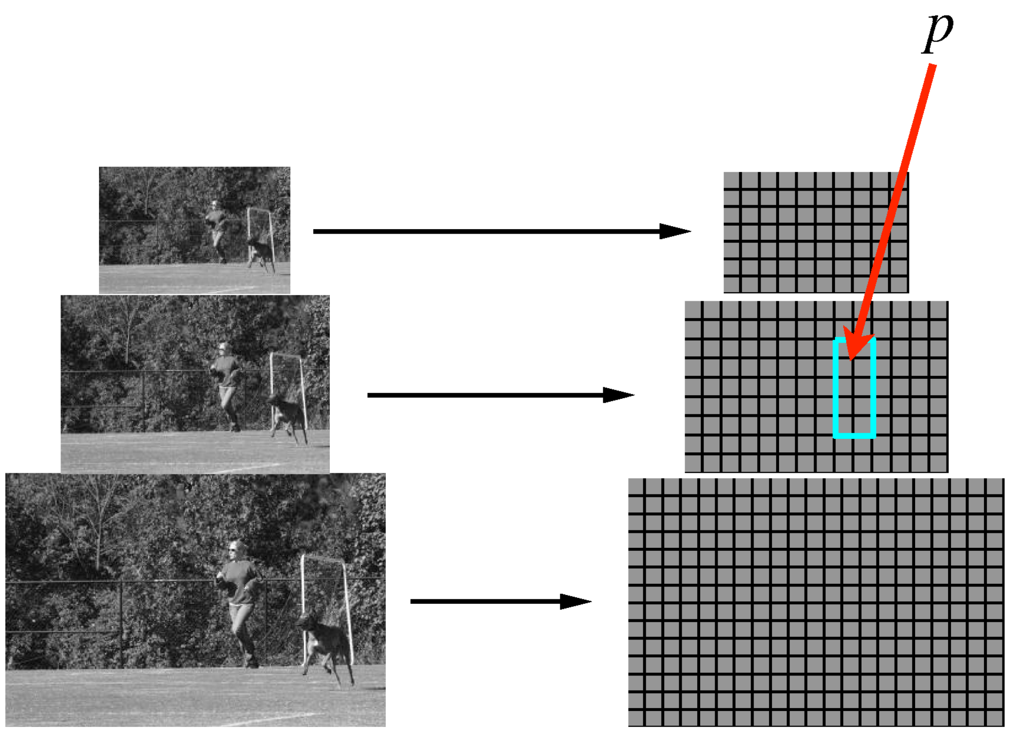
Histogram of Gradient (HOG) features

- Image is partitioned into 8x8 pixel blocks
- In each block we compute a histogram of gradient orientations
  - Invariant to changes in lighting, small deformations, etc.

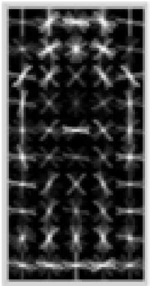
HOG Filters

- HOG filter is a template for HOG features
- Score is dot product of filter and feature vector



HOG pyramid H

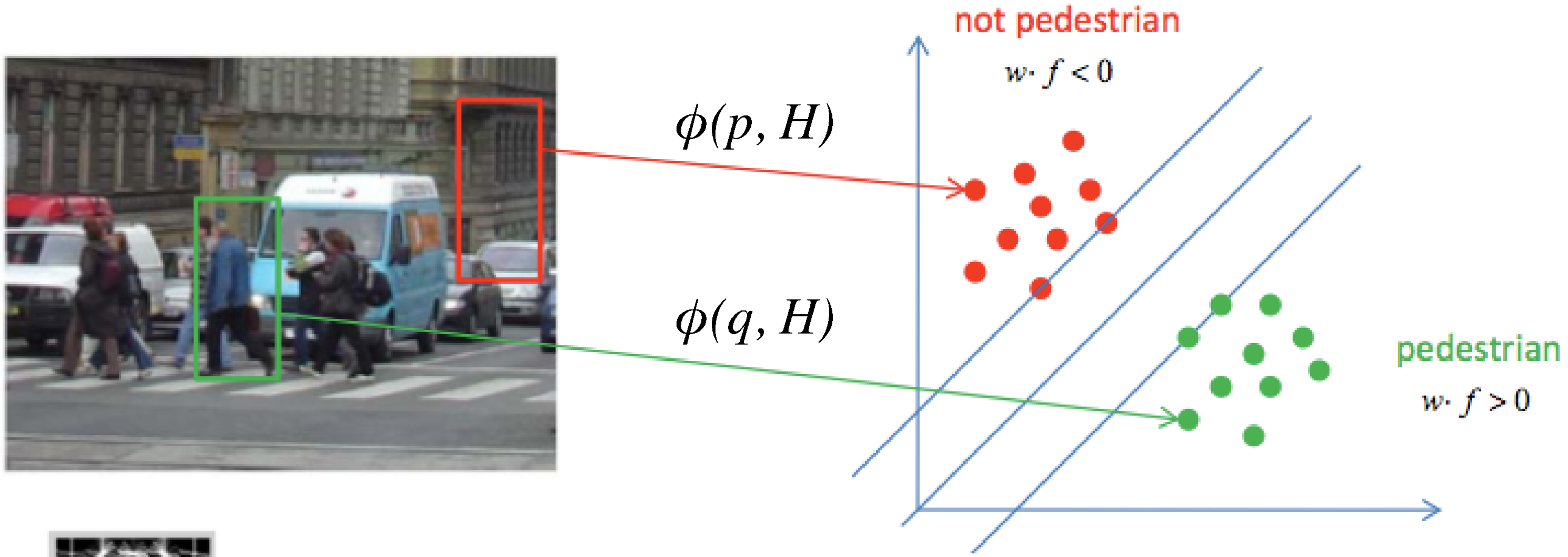
filter F

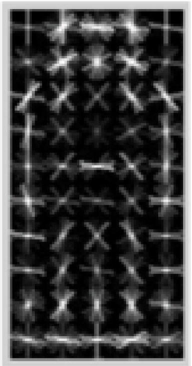


Score of F at position p is
$F \cdot \phi(p, H)$

$\phi(p, H)$ = HOG features in subwindow specified by p

Dalal & Triggs: HOG + linear SVMs





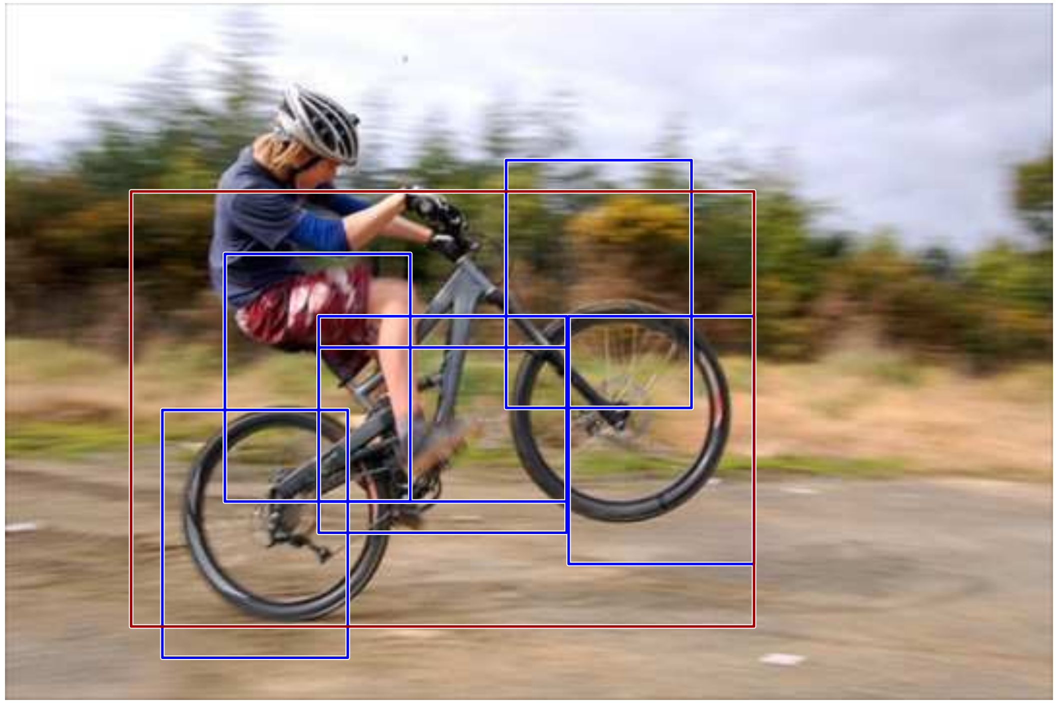
Typical form of
a model

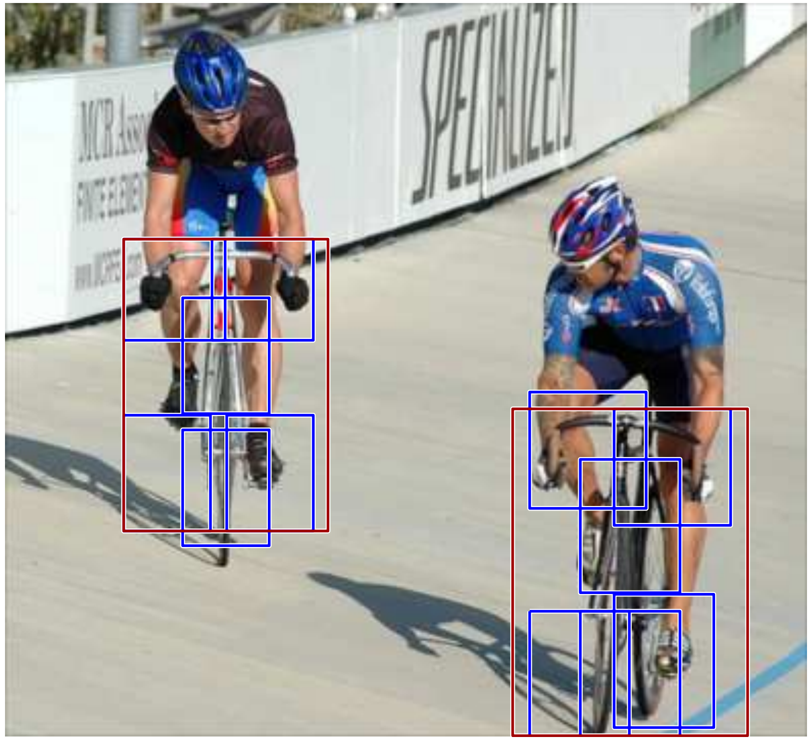
There is much more background than objects

Start with random negatives and repeat:

1) Train a model
2) Harvest false positives to define "hard negatives"

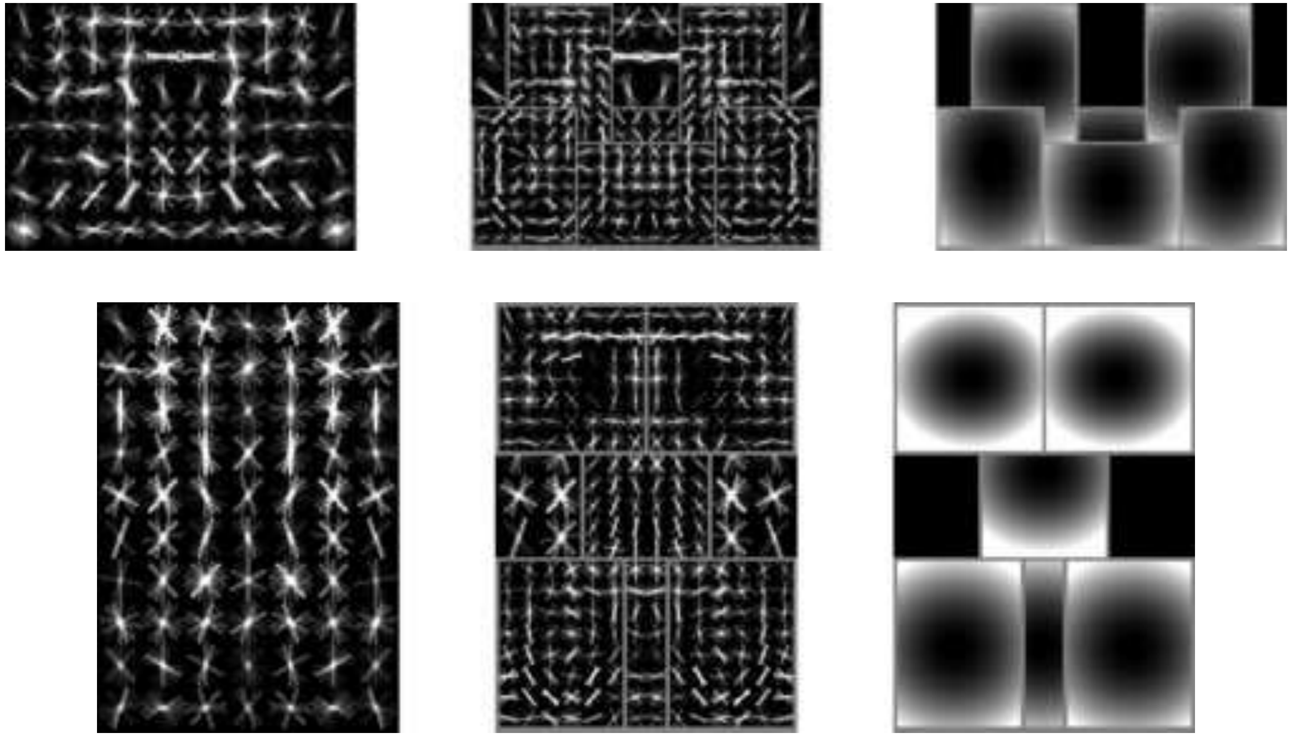
Deformable part models





- Collection of templates arranged in a deformable configuration
- Each model has global template + part templates
- Fully trained from bounding boxes alone

2 component bicycle model



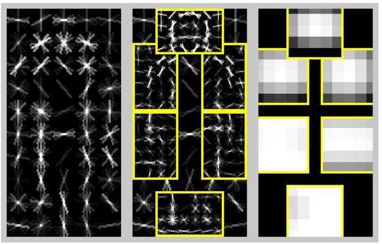
root filters
coarse resolution

part filters
finer resolution

deformation
models

Each component has a root filter F_0
and n part models (F_i, v_i, d_i)

Object hypothesis



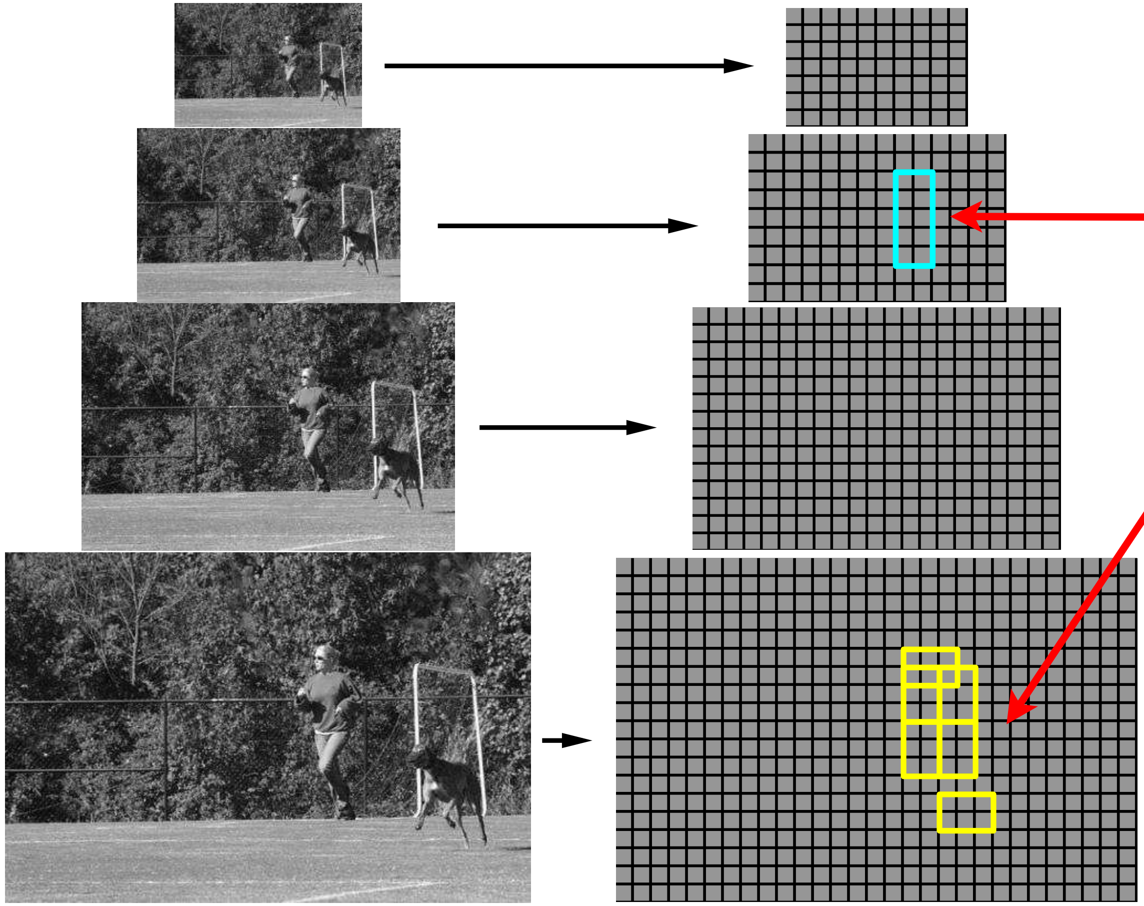


Image pyramid HOG feature pyramid

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

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

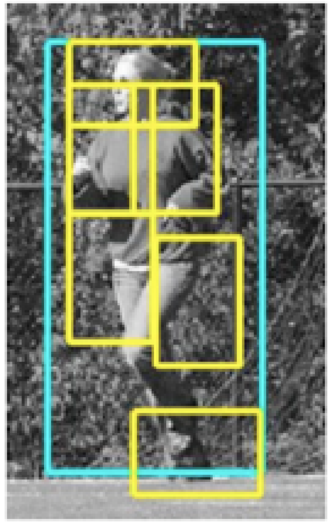
“data term” “spatial prior”

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

filters

displacements

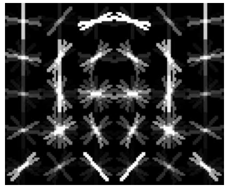
deformation parameters



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

concatenation filters and deformation parameters

concatenation of HOG features and part displacement features



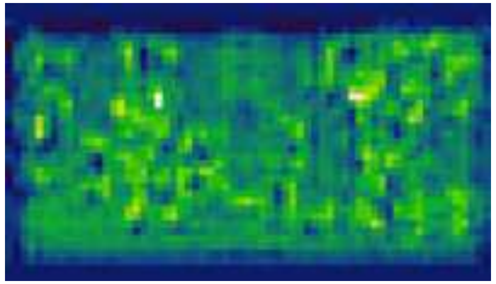
head filter

input image

Response of filter in l-th pyramid level

$$R_l(x, y) = F \cdot \phi(H, (x, y, l))$$

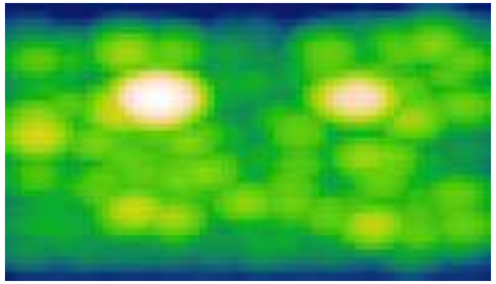
cross-correlation

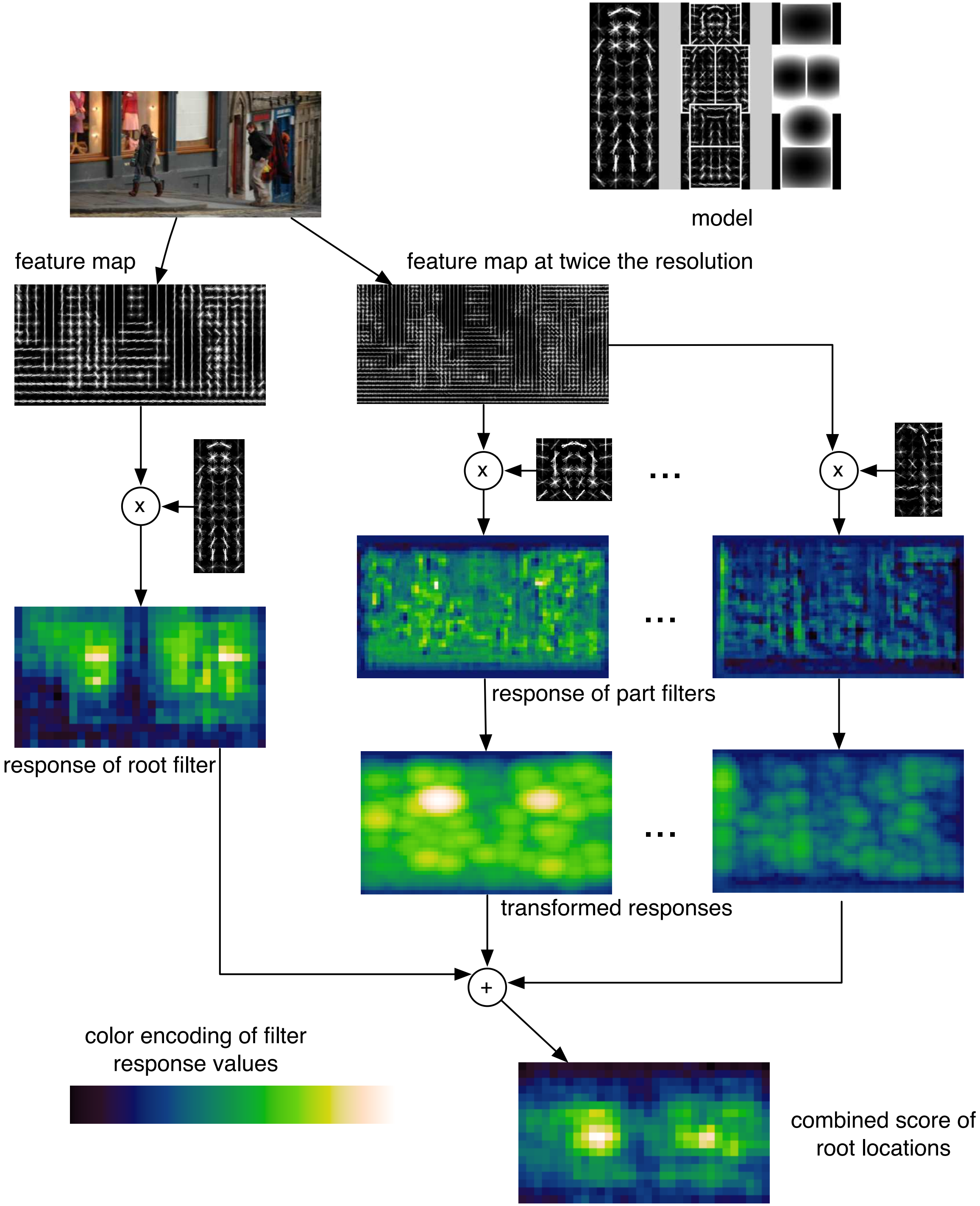


Transformed response

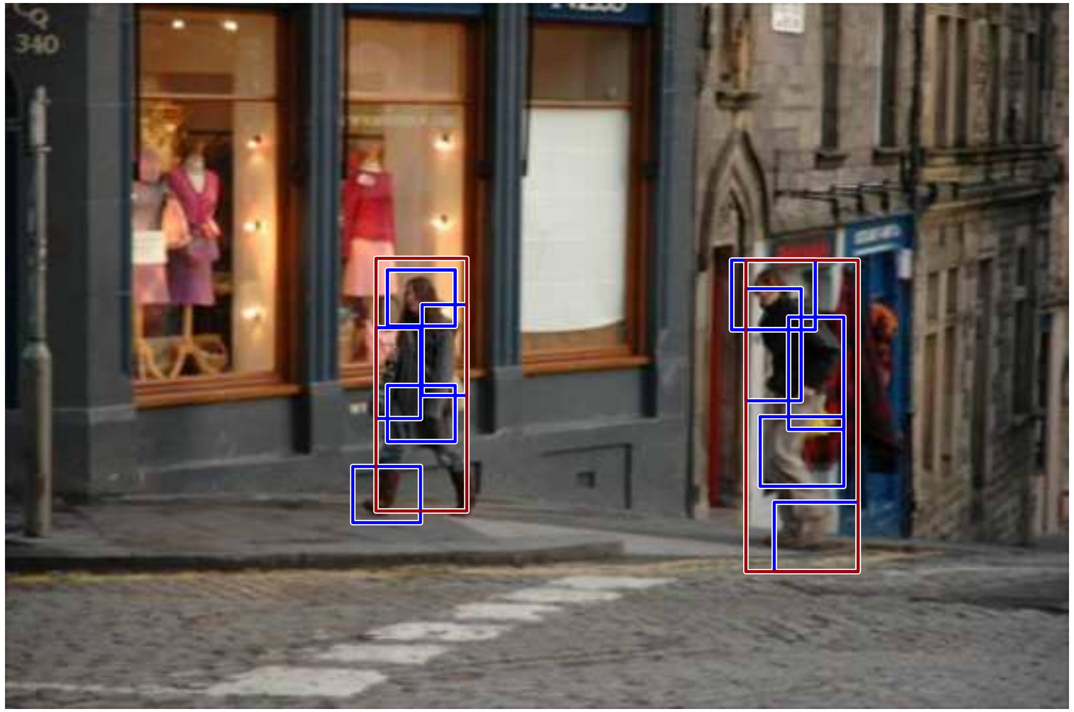
$$D_l(x, y) = \max_{dx, dy} \left(R_l(x + dx, y + dy) - d_i \cdot (dx^2, dy^2) \right)$$

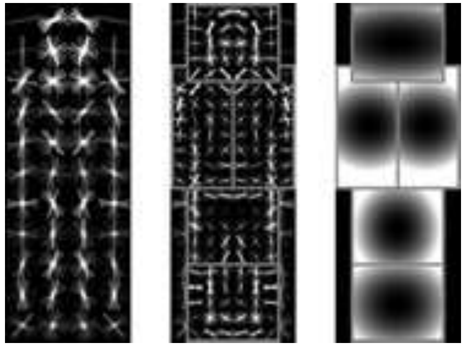
max-convolution, computed in linear time
(spreading, local max, etc)




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

Matching results



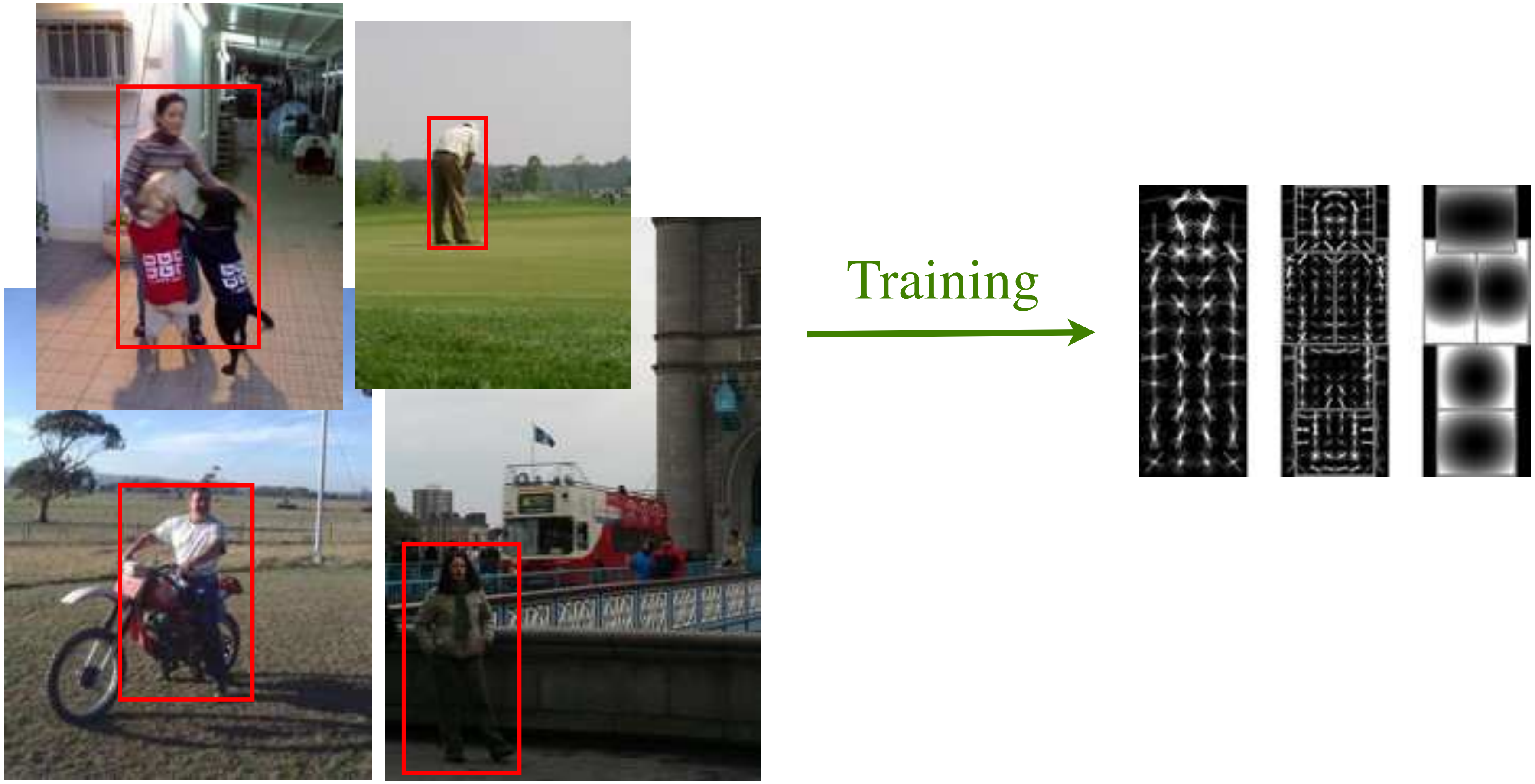


(after non-maximum suppression)

~1 second to search all scales on a multi-core computer

Learning

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



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

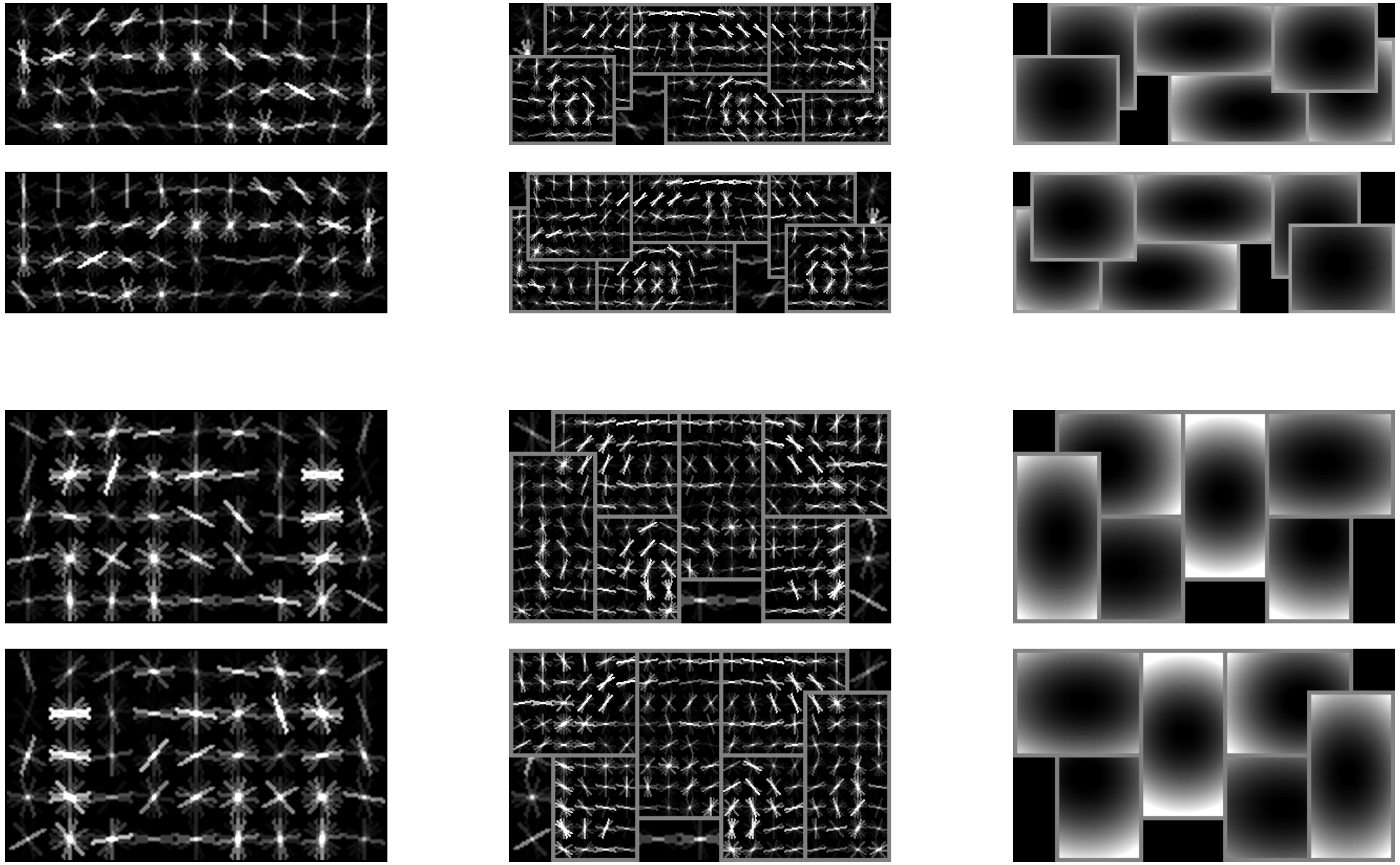
We would like to find β such that: $y_i f_\beta(x_i) > 0$

Minimize

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

6 component car model

2 of 3 symmetric pairs shown



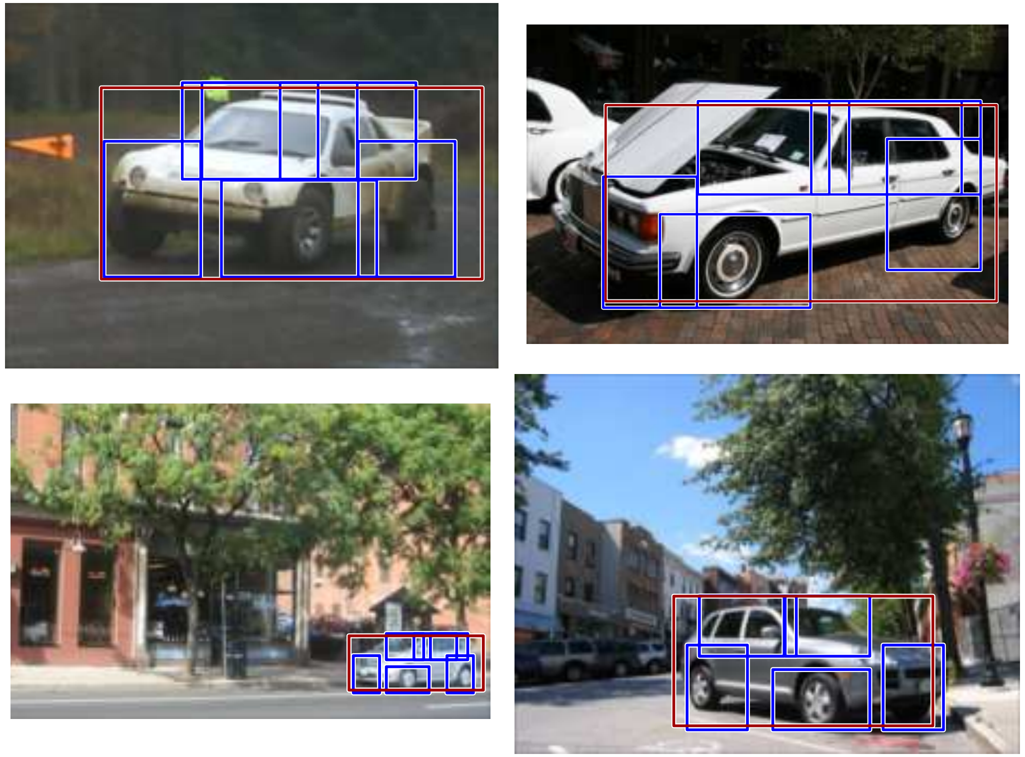
root filters
coarse resolution

part filters
finer resolution

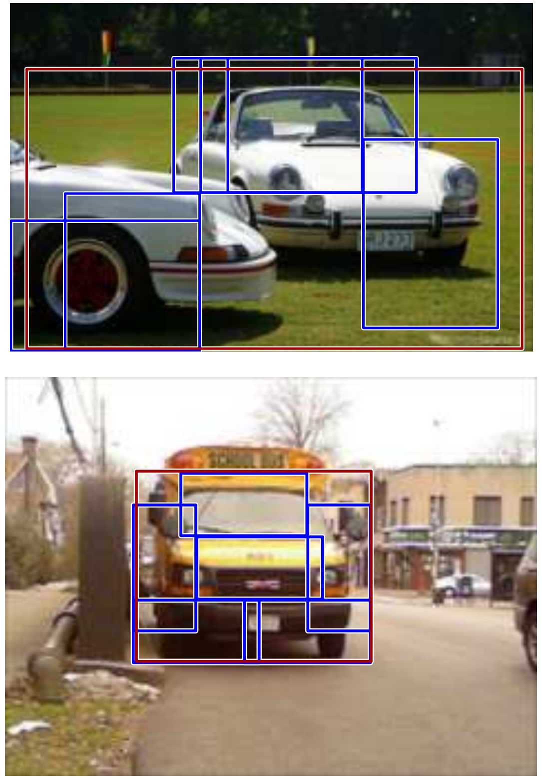
deformation
models

Car detections

high scoring true positives



high scoring false positives



Deuxième partie: la détection d’objet avec *deep learning*

Comprendre les données visuelles à grande échelle
19 novembre 2019

- (see part 2)