Computer Vision II - Recognition: *Object Detection*

Michael Yang

Self-introduction

- Since 03.2015, CVLD, TU Dresden
- 2012-2015, Postdoc, Leibniz University Hannover
- 2008-2011, Ph.D, Bonn University
- Room: 2028
- ying.yang1@tu-dresden.de

Roadmap (3 lectures)

- Object Detection (08.04)
- Image Categorization (15.04)
- Scene Understanding (22.04)

Roadmap (3 lectures)

- Object Detection (08.04)
- Image Categorization (15.04)
- Scene Understanding (22.04)
  (28.04 Intro. in Exercise 1)

Slides credits

- Bernt Schiele
- Li Fei-Fei
- Rob Fergus
- Kirsten Grauman
- Derek Hoiem
- Jianxiong Xiao
- Stefan Roth
- Jamie Shotton
- Antonio Criminisi
- Carsten Rother

Roadmap (this lecture)

- Defining the Problem
- Rigid Template
  - HOG for human detection
  - Exemplar SVM detector
- Part Based Detector
  - Deformable Part Model
  - Poselets
- New development for object detection

Recognition – What is the Goal?

- Object instance recognition (more precise: known object instance recognition)
  - We know exactly the instance

Proto-type images *Test image* *Result*

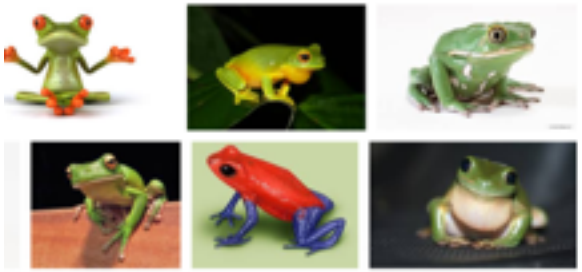
CV1

Techniques (see lecture 7):

- Robust (Ransac) matching with F, H
- Sparse Points (Harris)
- Geometric and Illumination invariant features (SIFT)

- Object class recognition (also called: Generic object recognition)
  - Different instance of the same class

Train-set



Test-set

Class versus Instance – a gray zone

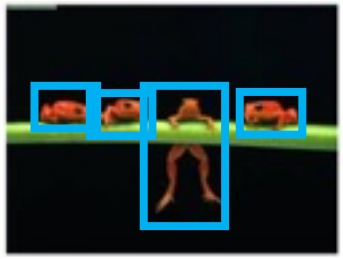
Object class: coke cans

Class-based recognition: Level of Detail

- Image Categorization
  - One or more categories per image

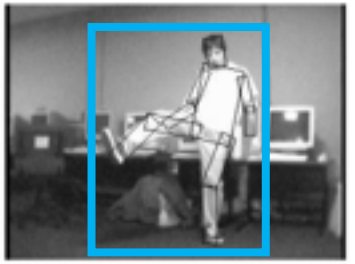
Frog, branch

- Object Class Detection
  - Also find bounding box

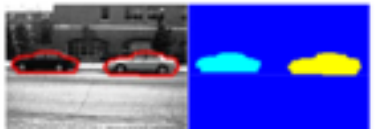


2D bounding box for each frog

- Part-based Object Detection
  - Find parts of the object
    (and in this way the full object)



- Semantic Segmentation
  (segmentation implies pixel-wise accuracy)
  - Object-class segmentation



Recognition – Many variants and extensions

- Range:

Predominantly Texture defines the class

Class is defined predominately by:
Outline (segmentation)
Individual parts
Layout of parts

- One can also add Attributes
  - Tall, flat, looks nice, “can be used for sitting on”, etc.
  - material

The Pascal VOC Challenge

Classification/Detection Competitions

1. **Classification**: For each of the twenty classes, predicting presence/absence of an example of that class in the test image.
2. **Detection**: Predicting the bounding box and label of each object from the twenty target classes in the test image.

20 classes
~27.000 labeled images

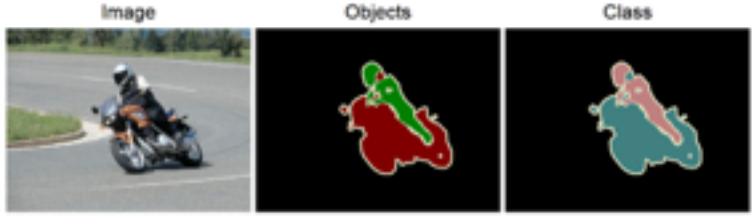
Participants may enter either (or both) of these competitions, and can choose to tackle any (or all) of the twenty object classes. The challenge allows for two approaches to each of the competitions:

1. Participants may use systems built or trained using any methods or data excluding the provided test sets.
2. Systems are to be built or trained using only the provided training/validation data.

The intention in the first case is to establish just what level of success can currently be achieved on these problems and by what method; in the second case the intention is to establish which method is most successful given a specified training set.

Segmentation Competition

- **Segmentation**: Generating pixel-wise segmentations giving the class of the object visible at each pixel, or "background" otherwise.



20 classes
~10.000 labeled images

http://host.robots.ox.ac.uk:8080/pascal/VOC/voc2012/index.html

The Pascal VOC Challenge

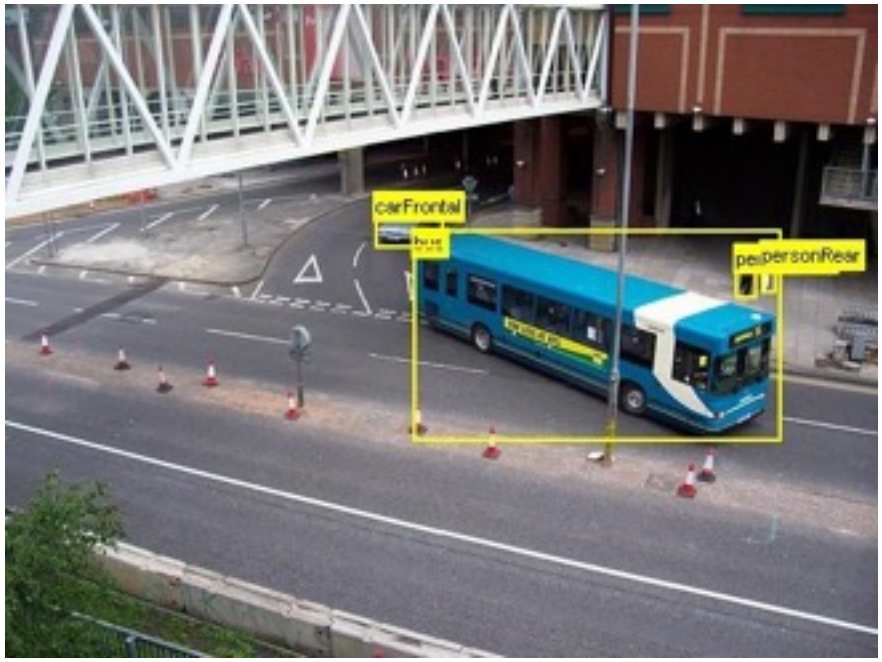
The main goal of this challenge is to recognize objects from a number of visual object classes in realistic scenes

20 classes

- *Person:* person
- *Animal:* bird, cat, cow, dog, horse, sheep
- *Vehicle:* aeroplane, bicycle, boat, bus, car, motorbike, train
- *Indoor:* bottle, chair, dining table, potted plant, sofa, tv/ monitor

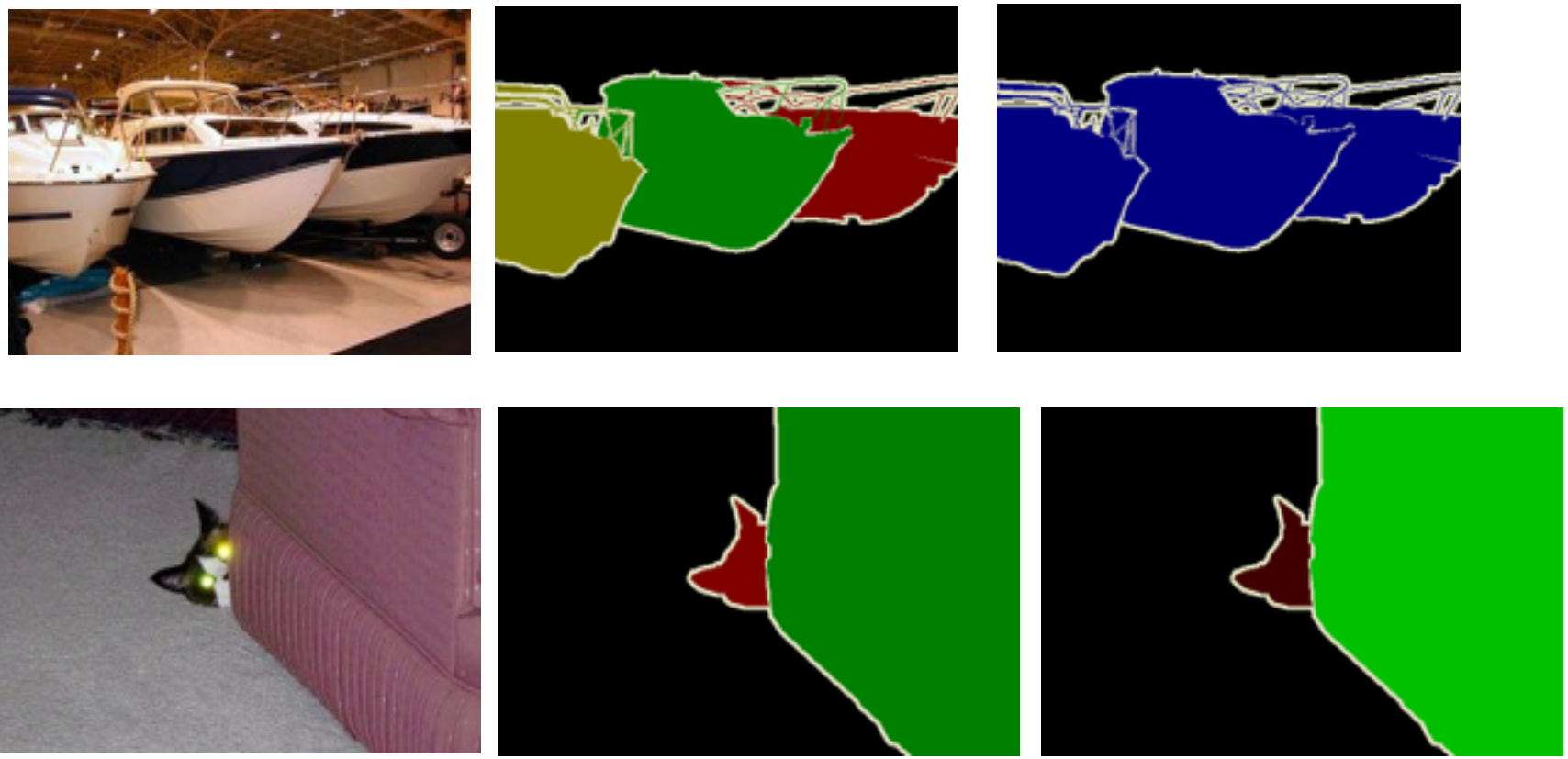
http://host.robots.ox.ac.uk:8080/pascal/VOC/voc2012/index.html

The Pascal VOC Challenge



http://host.robots.ox.ac.uk:8080/pascal/VOC/voc2012/index.html

The Pascal VOC Challenge



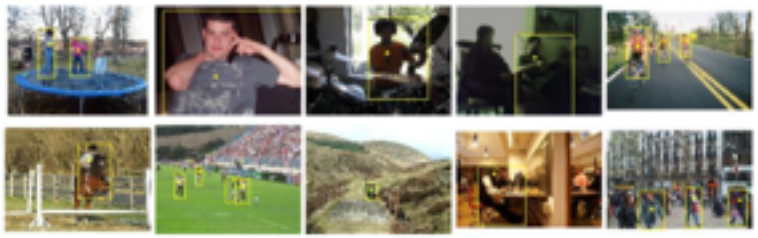
http://host.robots.ox.ac.uk:8080/pascal/VOC/voc2012/index.html

The Pascal VOC Challenge

Action Classification Competition

- **Action Classification**: Predicting the action(s) being performed by a person in a still image.

10 action classes + "other"



In 2012 there are two variations of this competition, depending on how the person whose actions are to be classified is identified in a test image: (i) by a tight bounding box around the person; (ii) by only a single point located somewhere on the body. The latter competition aims to investigate the performance of methods given only approximate localization of a person, as might be the output from a generic person detector.

ImageNet Large Scale Visual Recognition Competition

The goal of this competition is to estimate the content of photographs for the purpose of retrieval and automatic annotation using a subset of the large hand-labeled ImageNet dataset (10,000,000 labeled images depicting 10,000+ object categories) as training. Test images will be presented with no initial annotation - no segmentation or labels - and algorithms will have to produce labelings specifying what objects are present in the images. In this initial version of the challenge, the goal is only to identify the main objects present in images, not to specify the location of objects.

Further details can be found at the ImageNet website.

Person Layout Taster Competition

- **Person Layout**: Predicting the bounding box and label of each part of a person (head, hands, feet).

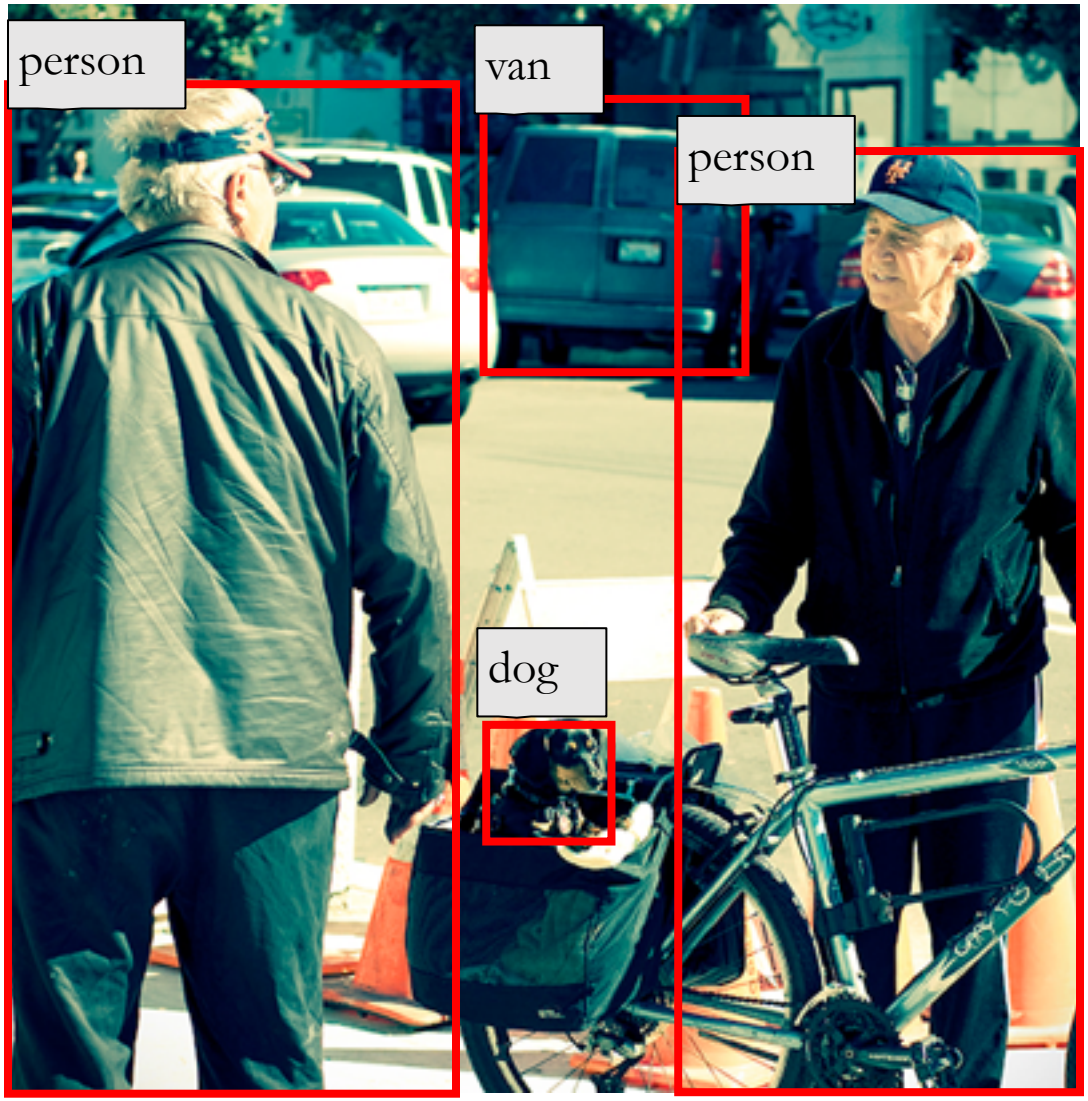
Image | Person Layout

10.000 classes
~10.000.000 labeled images

http://host.robots.ox.ac.uk:8080/pascal/VOC/voc2012/index.html

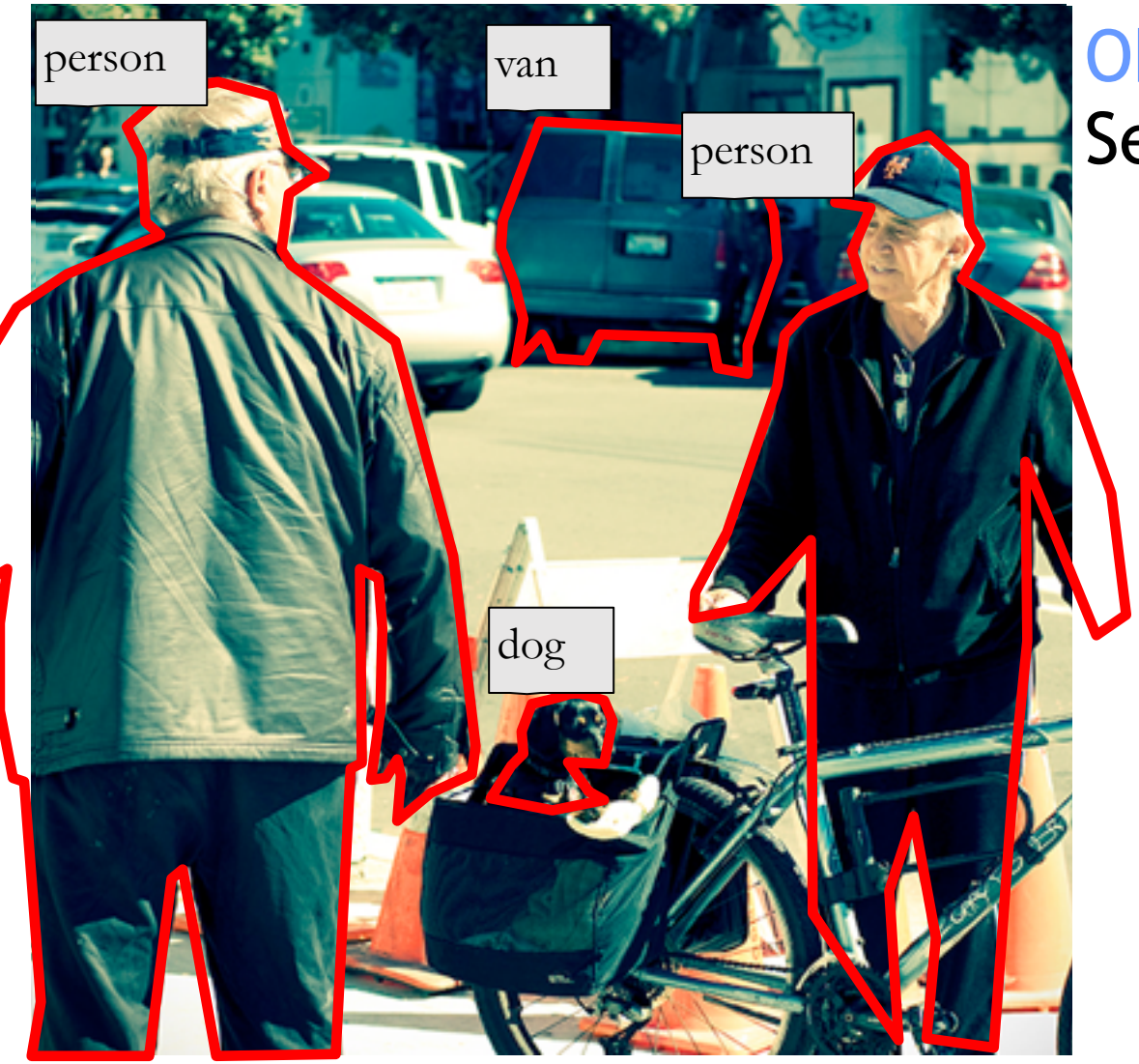
Big Picture: High-Level Computer Vision

Big Picture: High-Level Computer Vision



Object Detection

Big Picture: High-Level Computer Vision



Object Detection

Semantic Segmentation

Big Picture: High-Level Computer Vision

Object Detection
Semantic Segmentation
Pose Estimation

Big Picture: High-Level Computer Vision

Object Detection
Semantic Segmentation
Pose Estimation
Action Recognition

Big Picture: High-Level Computer Vision

Object Detection
Semantic Segmentation
Pose Estimation
Action Recognition
Attribute Classification

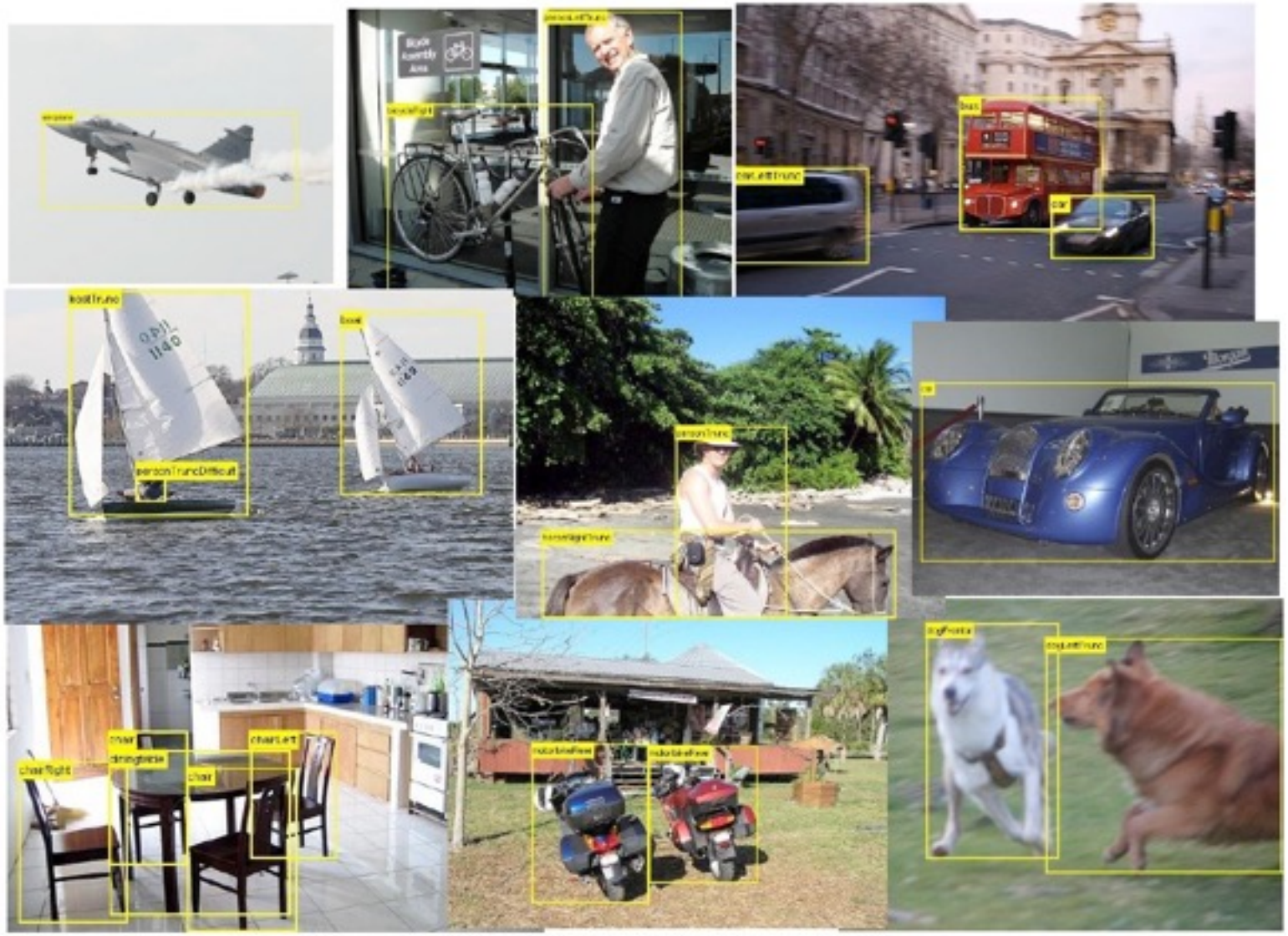
Big Picture: High-Level Computer Vision

Object Detection
Semantic Segmentation
Pose Estimation
Action Recognition
Attribute Classification

Roadmap (this lecture)

- Defining the Problem
- Rigid Template
  - HOG for human detection
  - Exemplar SVM detector
- Part Based Detector
  - Deformable Part Model
  - Poselets
- New development for object detection

Task: Generic object detection



Task: Generic object detection

In this lecture we focus on 2D image object detection but object detection can also in 3D

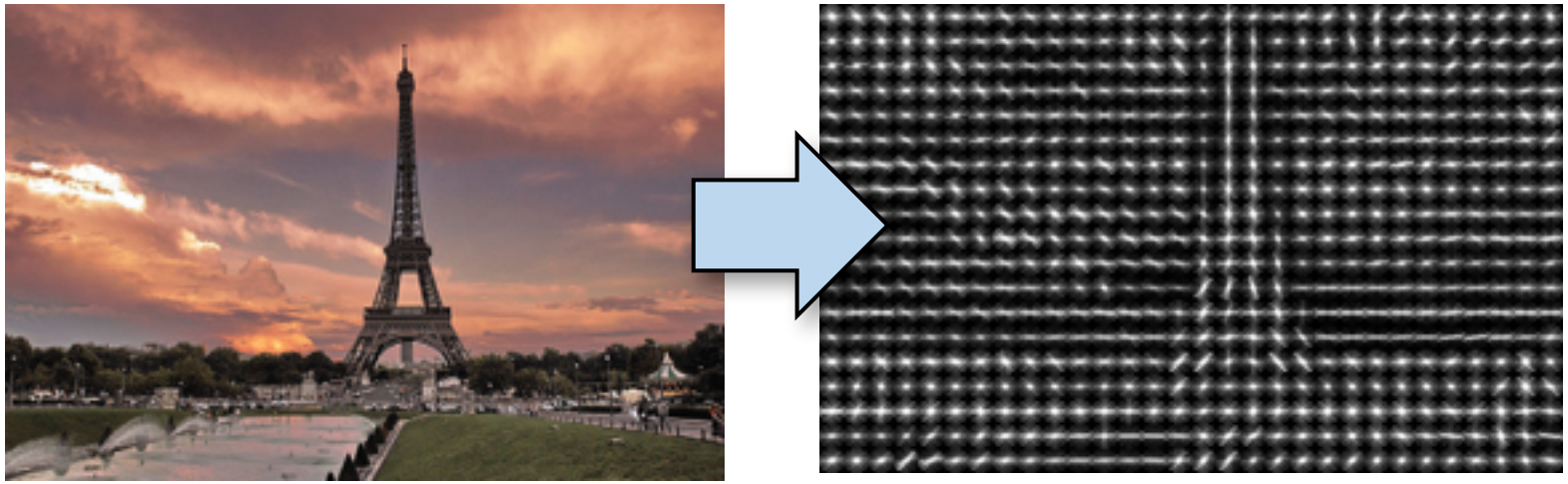
Demo: Face Detection

Histograms of oriented gradients for human detection

[Dalal&Triggs CVPR05]

Human detection with HOG: Basic Steps

1. Map image to feature Space (HOG)



Human detection with HOG: Basic Steps

1. Map image to feature Space (HOG)
2. Training with positive and negative (linear SVM)

positive training examples

negative training examples

Human detection with HOG: Basic Steps

1. Map image to feature Space (HOG)
2. Training with positive and negative (linear SVM)

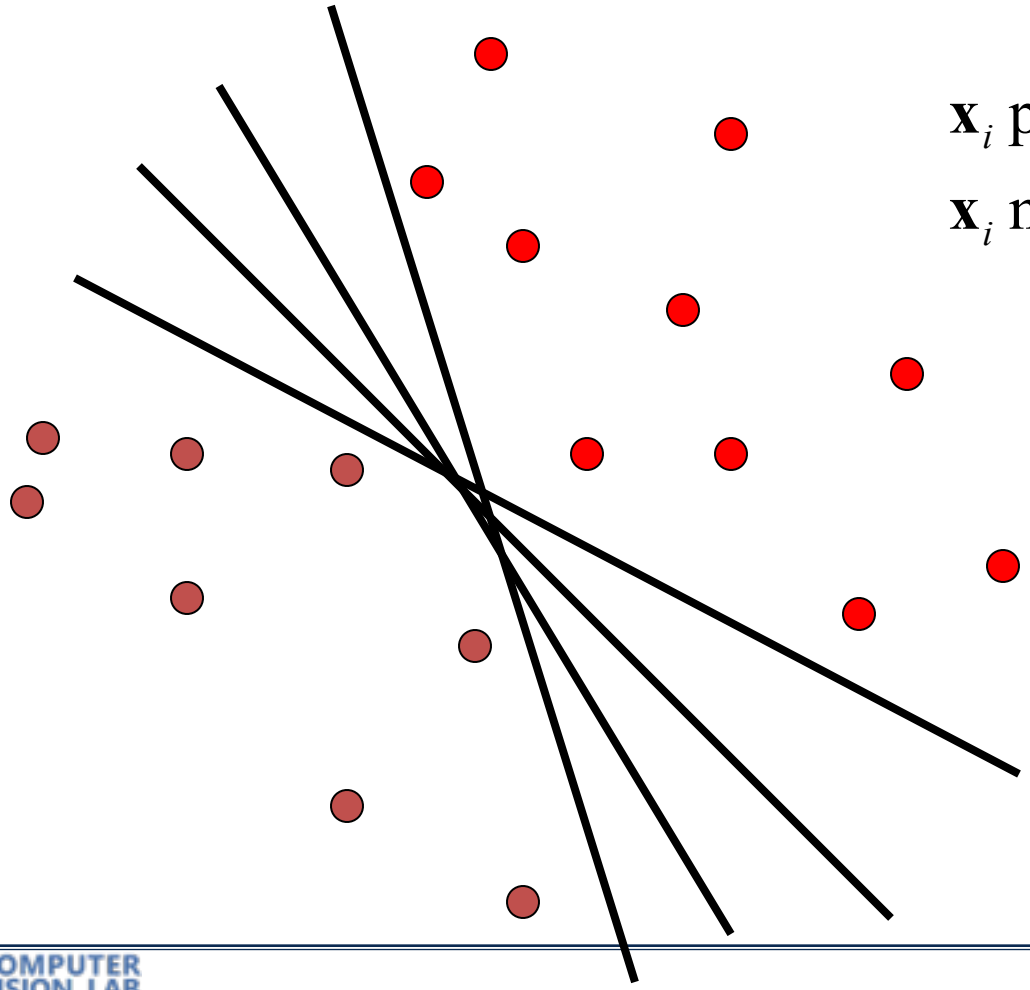
positive training examples + thousands more…

negative training examples+ millions more…

Human detection with HOG: Basic Steps

Linear classifiers

- Find linear function to separate positive and negative examples



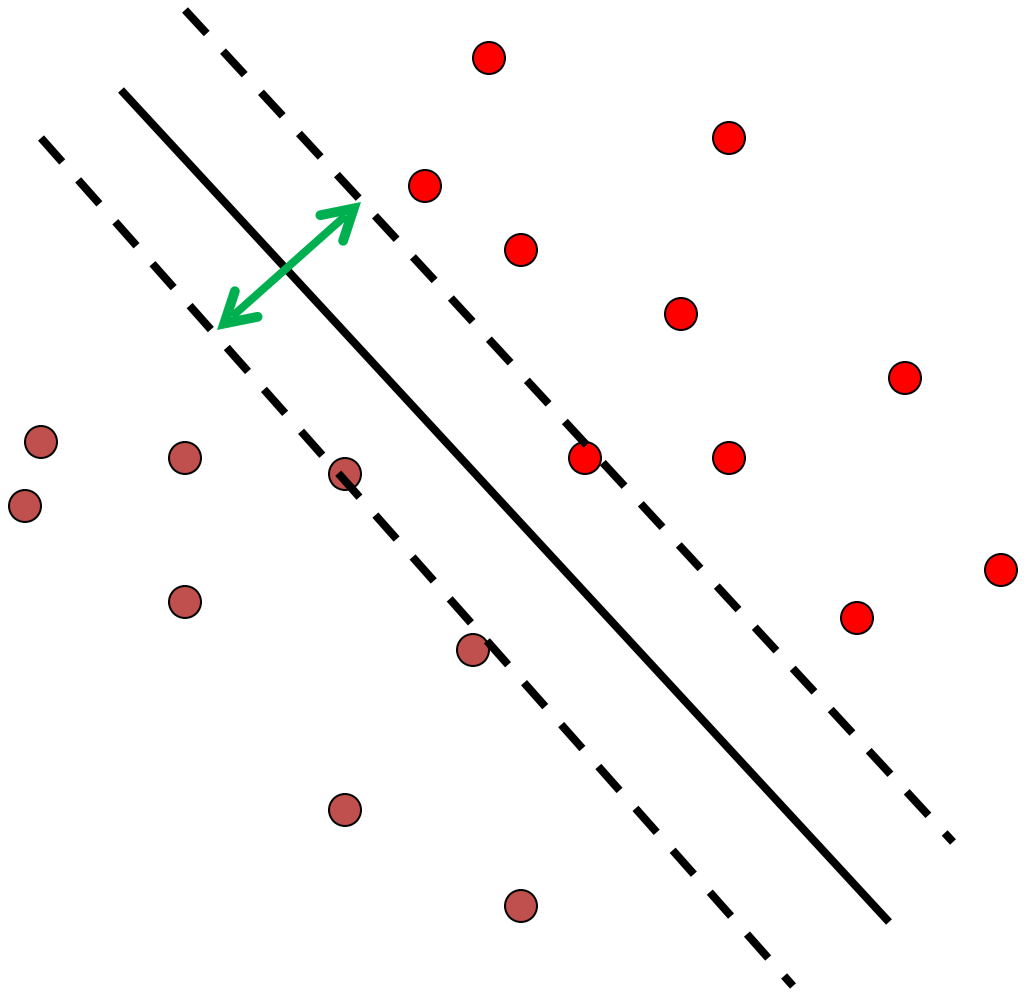
$$\mathbf{x}_i \text{ positive}: \quad \mathbf{x}_i \cdot \mathbf{w} + b \geq 0$$

$$\mathbf{x}_i \text{ negative}: \quad \mathbf{x}_i \cdot \mathbf{w} + b < 0$$

Which line is best?

Human detection with HOG: Basic Steps

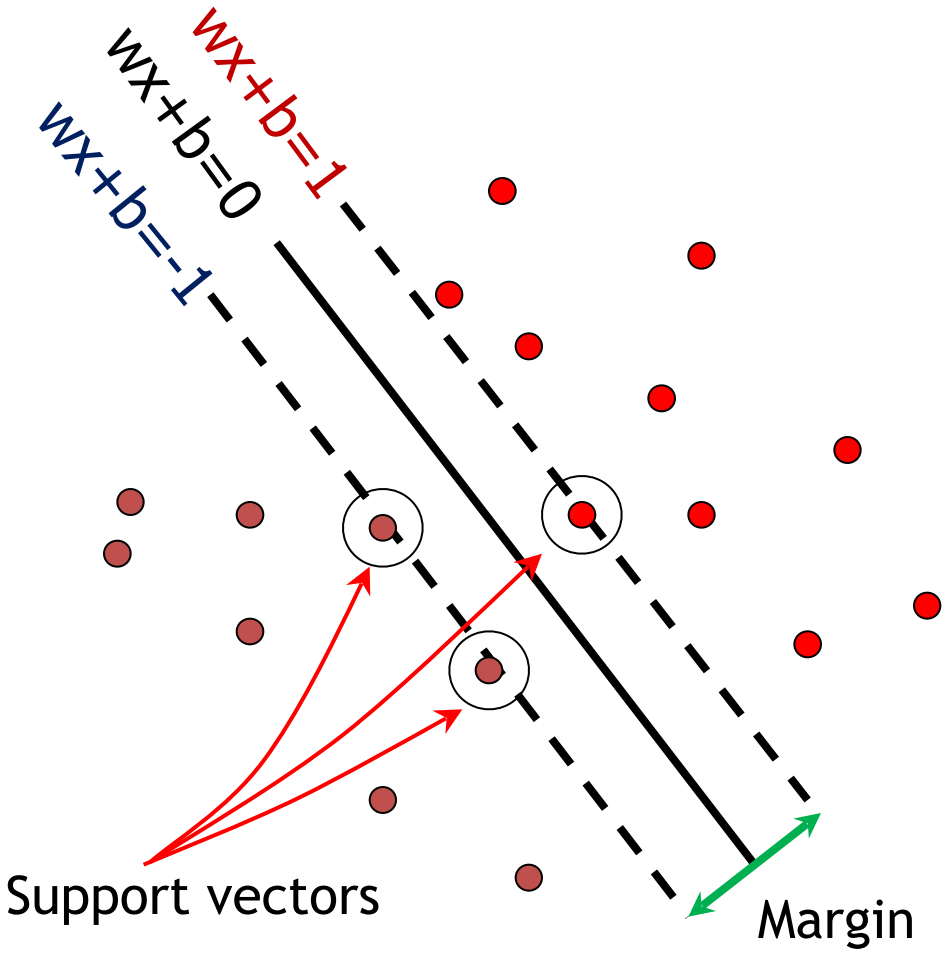
Support Vector Machines (SVMs)



- Discriminative classifier based on *optimal separating line (for 2D case)*
- Maximize the *margin* between the positive and negative training examples

Human detection with HOG: Basic Steps

Support Vector Machines (SVMs)



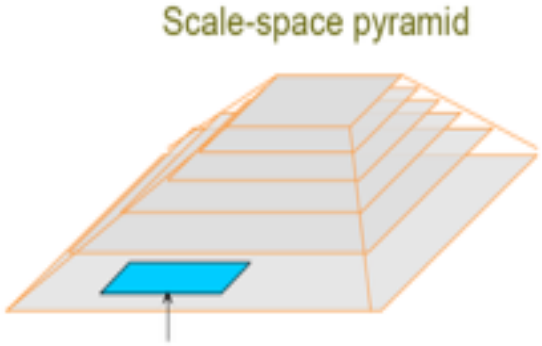
$\mathbf{x}_i$ positive $(y_i = 1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \geq 1$

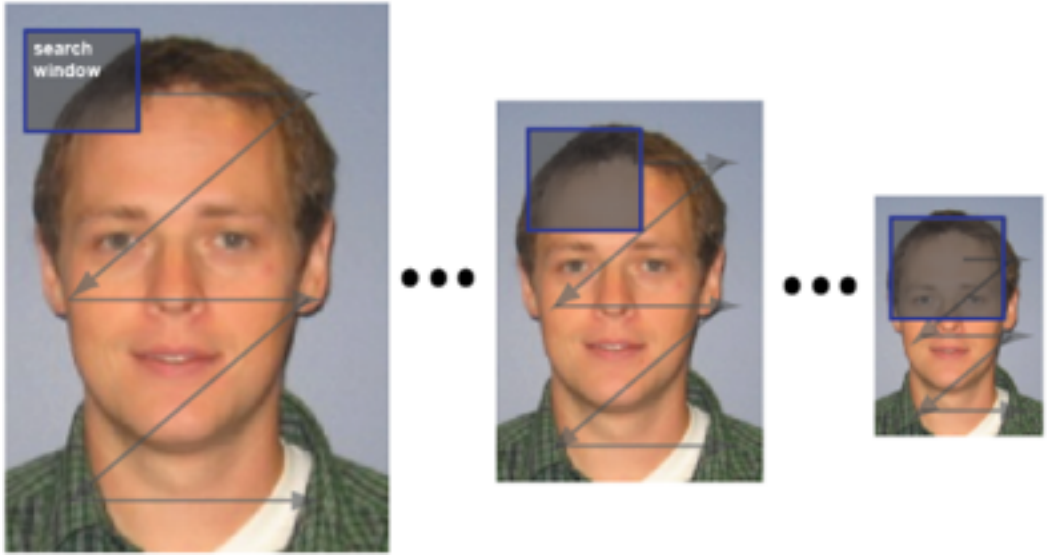
$\mathbf{x}_i$ negative $(y_i = -1)$: $\mathbf{x}_i \cdot \mathbf{w} + b \leq -1$

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

Human detection with HOG: Basic Steps

1. Map image to feature Space (HOG)
2. Training with positive and negative (linear SVM)
3. Testing : scan image in all scale and all location
   Binary classification on each location

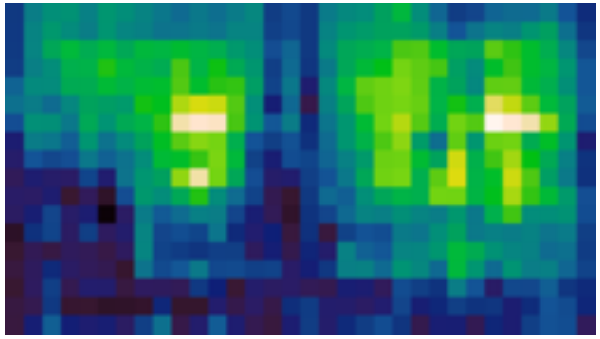


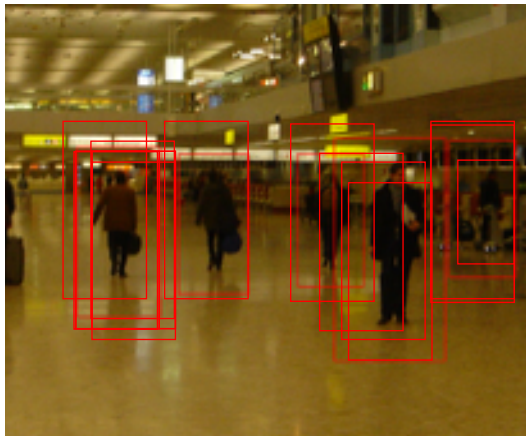


Human detection with HOG: Basic Steps

1. Map image to feature Space (HOG)
2. Training with positive and negative (linear SVM)
3. Testing : scan image in all scale and all location
4. Report box: non-maximum suppression

Detector response map





Final Boxes

After thresholding

After non-maximum suppression

Summary of Basic object detection Steps

Training:

Train a classifier describe the detection target

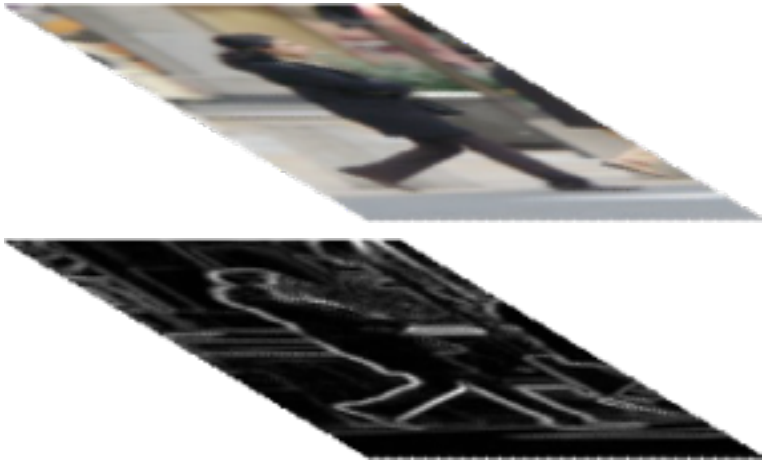
Testing :

Detection by binary classification on all location

HOG descriptor

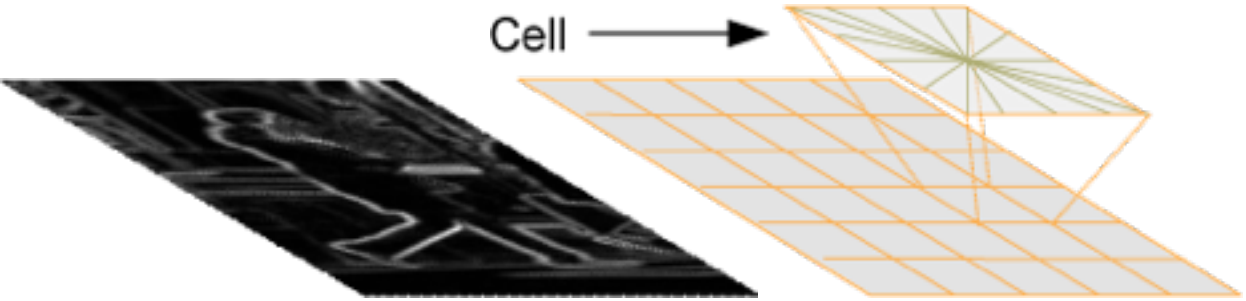
HOG: Gradients

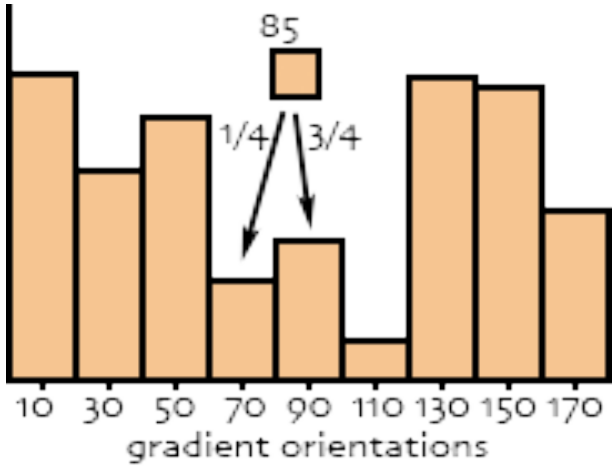
- Compress image to 64x128 pixels
- Convolution with [-1 0 1], [-1 0 1]^T filters
- Compute gradient magnitude + direction
- For each pixel: take the color channel with greatest magnitude as final gradient



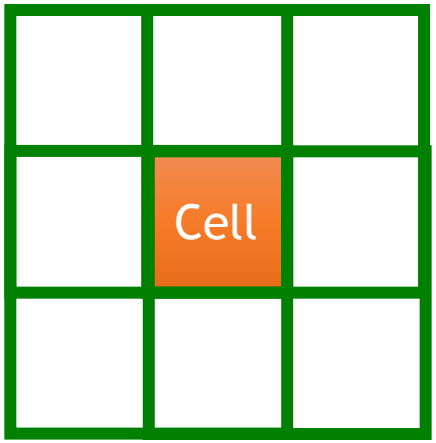
HOG: Cell histograms

- Divide the image to cells, each cell 8x8 pixels
- Snap each pixel's direction to one of 18 gradient orientations - 9 gradient orientations (unsigned)!
- Build histogram pre-cell using magnitudes

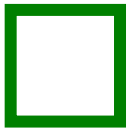




Normalization



Current cell : 1x18 histogram

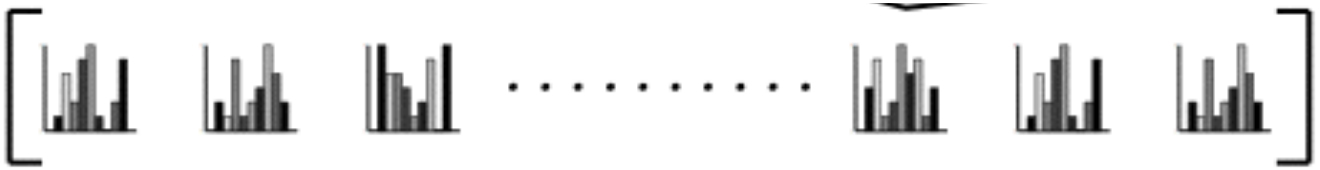


Block: 2x2 cell
overlapping with current cell

7x15 Blocks

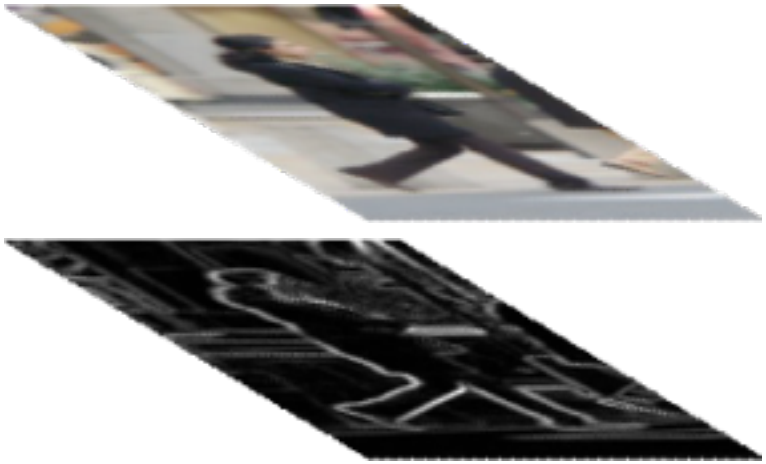
Final Descriptor

- Concatenation the normalized histogram (7*15*4*9 = 3780)



HOG Descriptor:

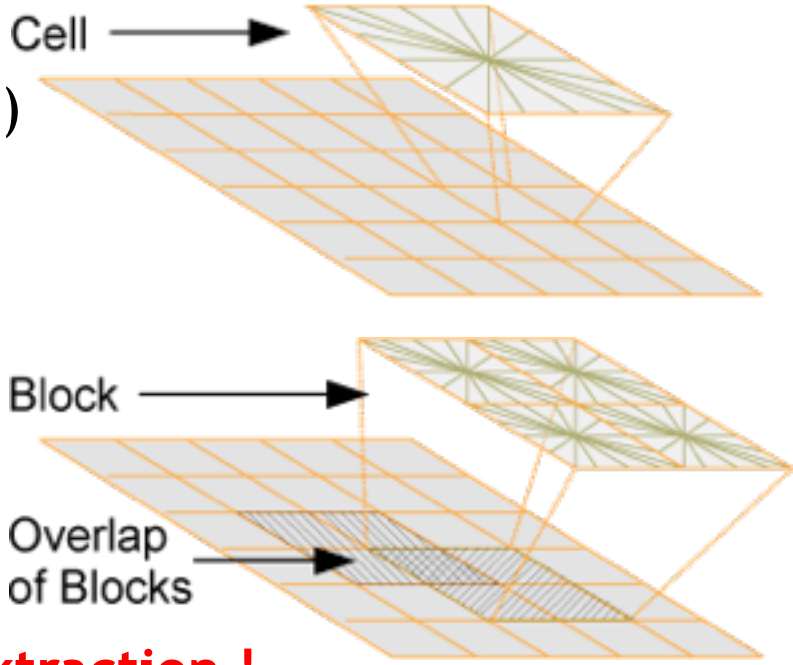
1. **Compute gradients** on an image region of 64x128 pixels



2. **Compute histograms** on ‘cells’ of typically 8x8 pixels (i.e. 8x16 cells)

3. **Normalize histograms** within overlapping blocks of cells

4. **Concatenate histograms**

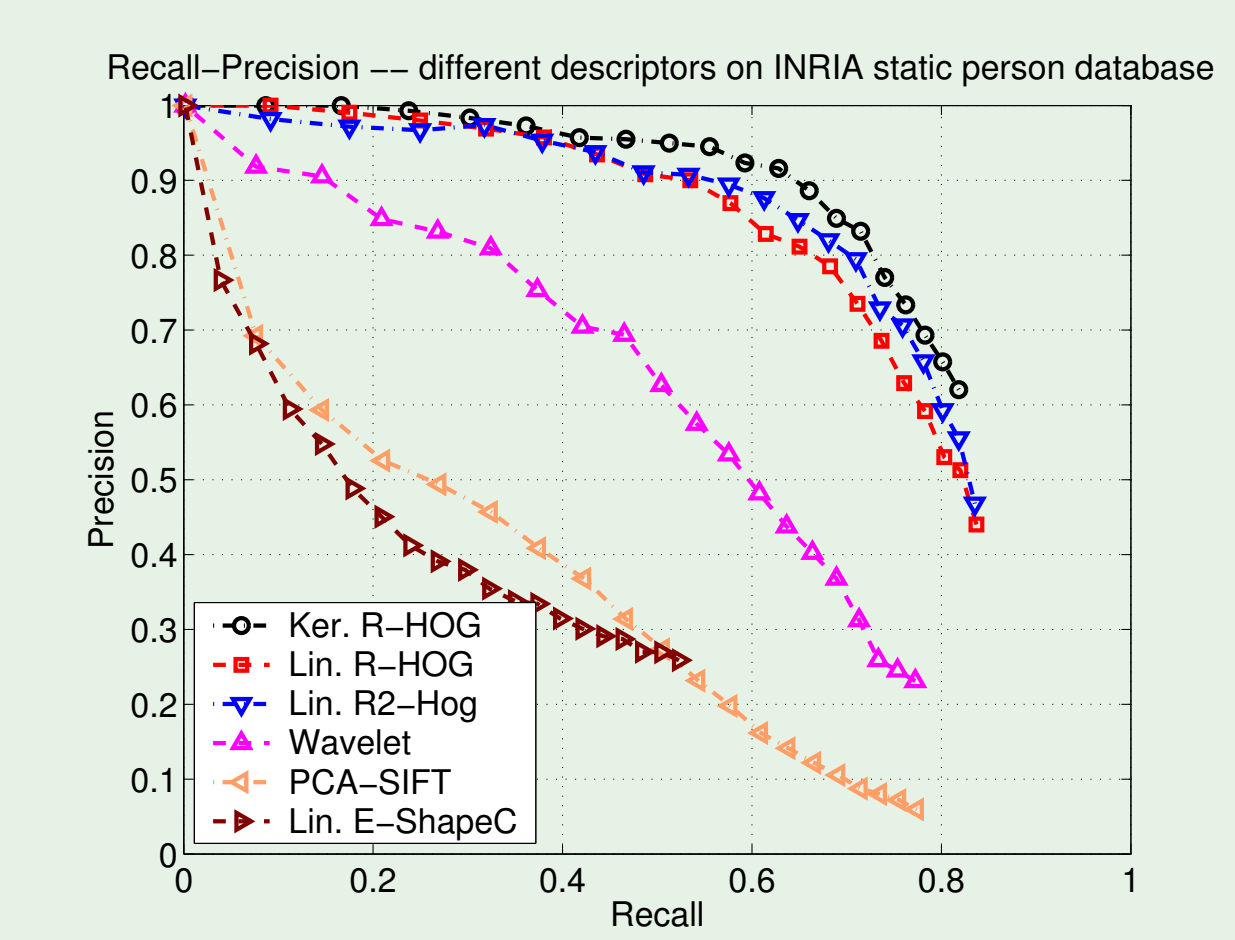


It is a typical procedure of feature extraction !

Feature Engineering

- Developing a feature descriptor requires a lot of engineering
  - Testing of parameters (e.g. size of cells, blocks, number of cells in a block, size of overlap)
  - Normalization schemes
- An extensive evaluation was performed to make these design desiccations
- It’s not only the idea, but also the engineering effort

Problem?



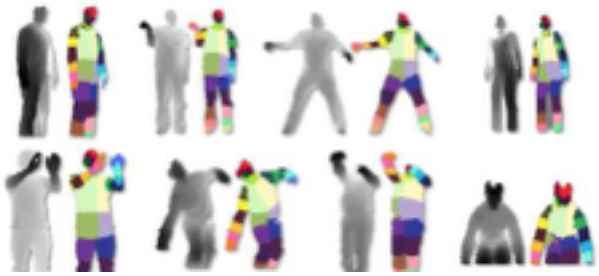
- AP=75% VERY GOOD!

Problem?

Single, rigid template usually not enough to represent a category.

- Many object categories look very different from different viewpoints, or style

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



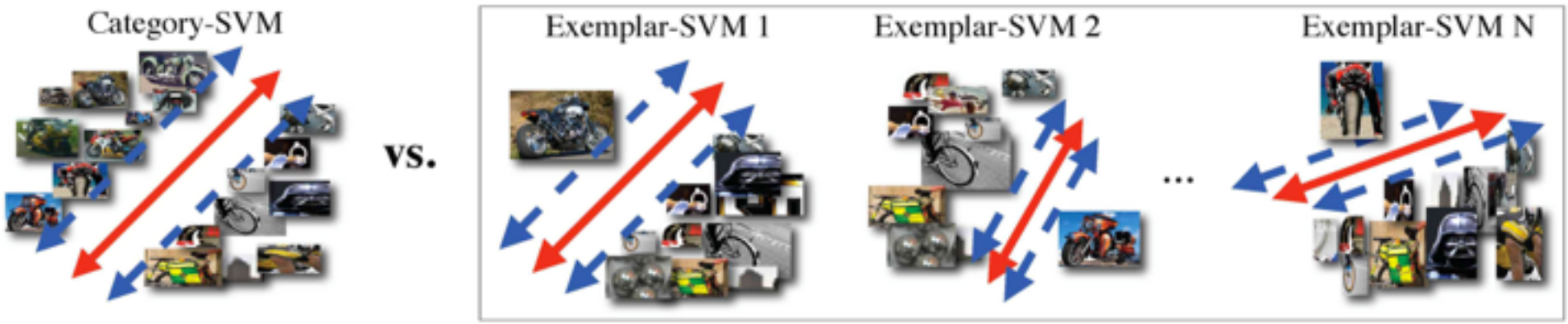
Solution :

- Exemplar SVM: Ensemble of Exemplar-SVMs for Object Detection and Beyond

- Part Based Model

Exemplar-SVM

- Still a rigid template,but train a separate SVM for each positive instance



For each category it can has exemplar with different size aspect ratio

Benefit from Exemplar-SVM ?

- Handel the intra-category variance naturally, without using complicated model.

- Compare to nearest neighbor approach: make use of negative data and train a discriminative object detector

- Explicit correspondence from detection result to training exemplar

Benefit from Exemplar-SVM ?

- Explicit correspondence from detection result to training exemplar

We not only know it is train,but also its orientation and type!

Roadmap (this lecture)

- Defining the Problem
- Rigid Template
  - HOG for human detection
  - Exemplar SVM detector
- Part Based Detector
  - Deformable Part Model
  - Poselets
- New development for object detection

Half way slide

3 Minutes break

Part Based Detector

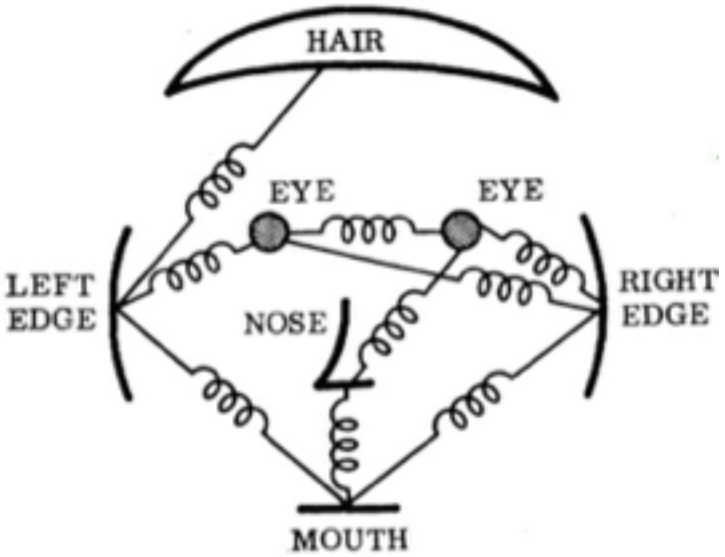
- Pictorial Structures

- Without part label
  - Deformable part model

- With part labeled
  - Poselets

Part Based Detector

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

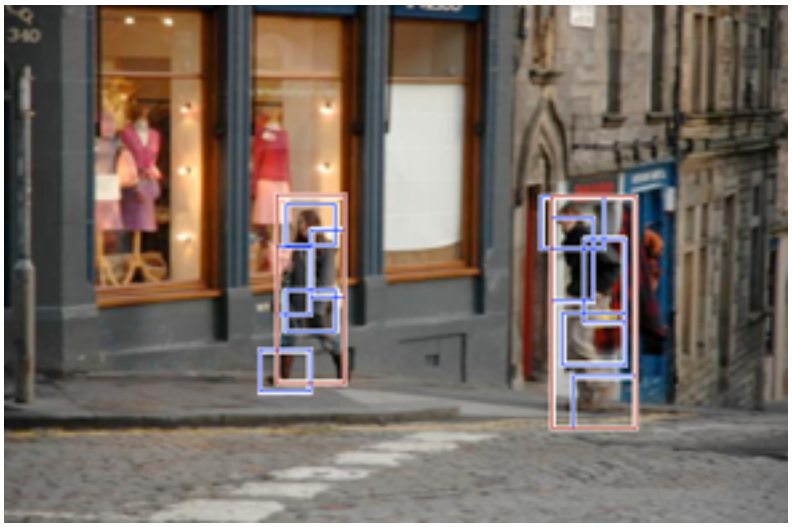


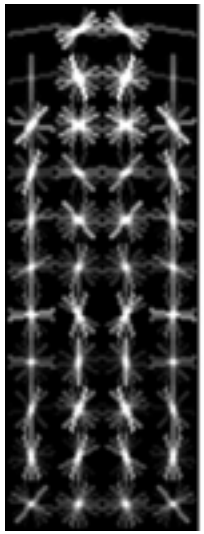
Face model by Fischler and Elschlager '73

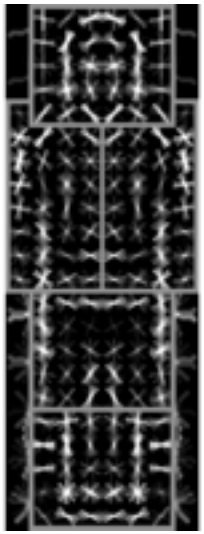
Part Based Detector

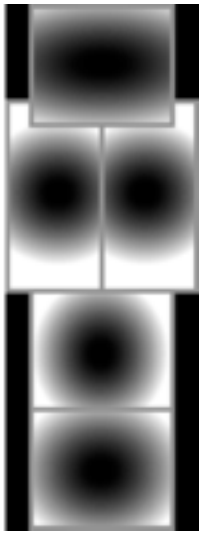
- How to defined the parts for one object category
- How to represent their spatial relation shape
- How to combine parts detection and spatial relations to obtained the final detection

DPM : Object Detection with Discriminatively Trained Part Based Models









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

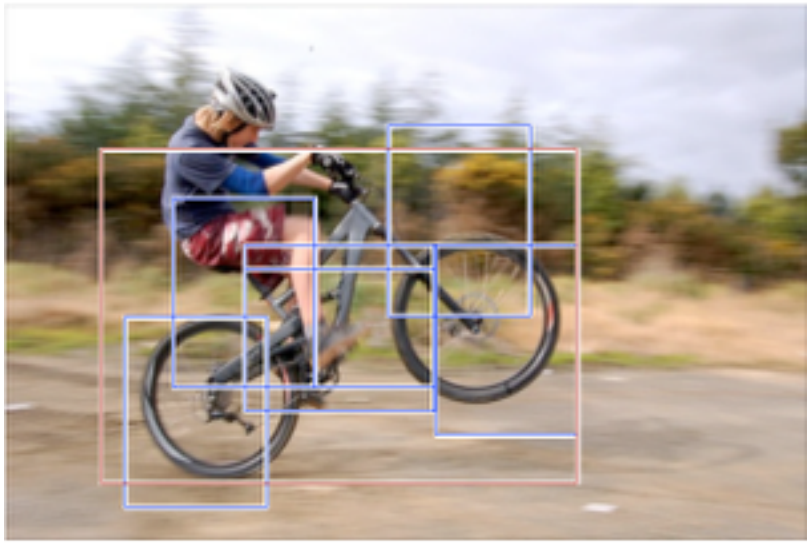
DPM: overview

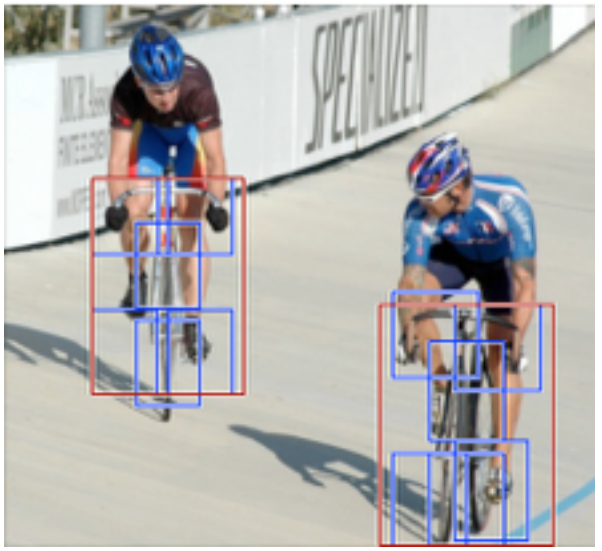
- Each category detector has mixture of deformable part models (components)

- Each component has global template + deformable parts

- Fully trained from bounding boxes alone (Latent SVM)

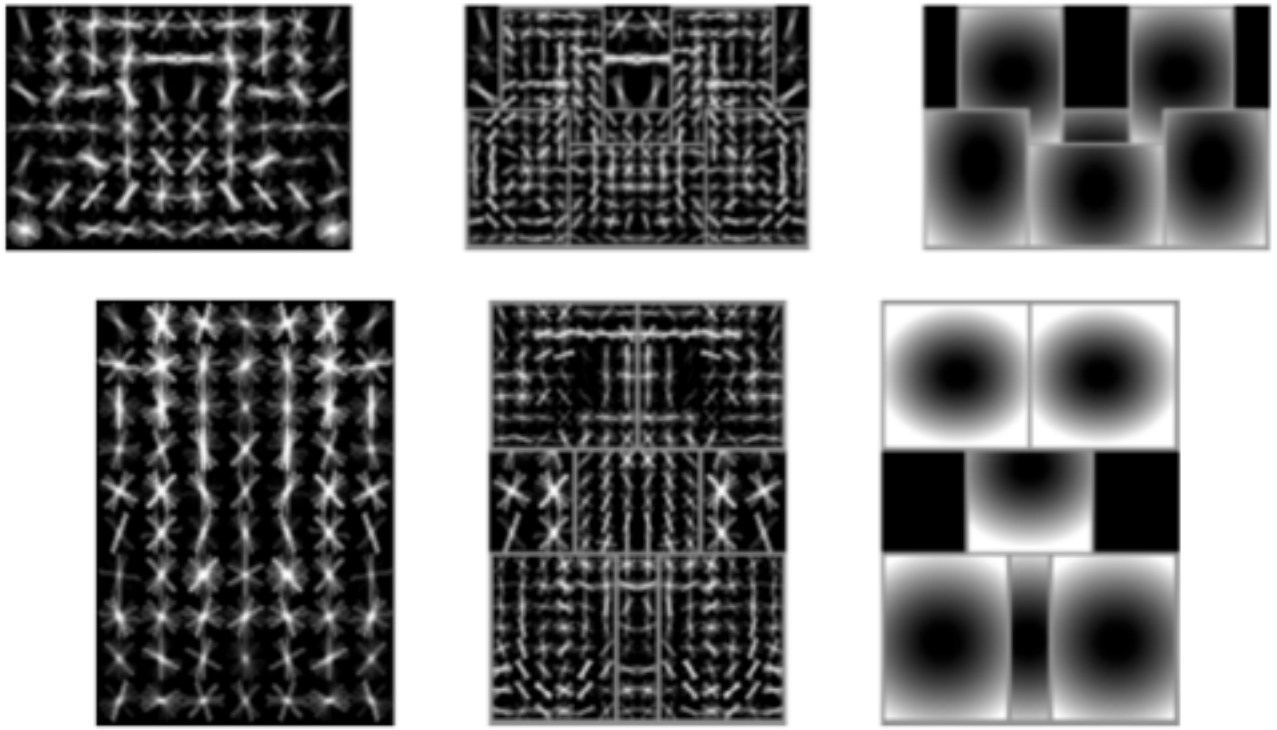
DPM: component





- Each category detector has mixture of component for different aspect ratio (handle intra-class variance)
- Each component has a it's own DPM model

DPM: component



root filters
coarse resolution

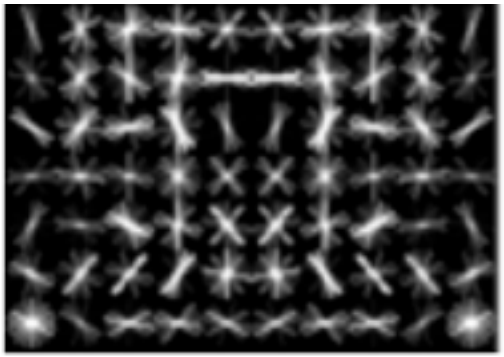
part filters
finer resolution

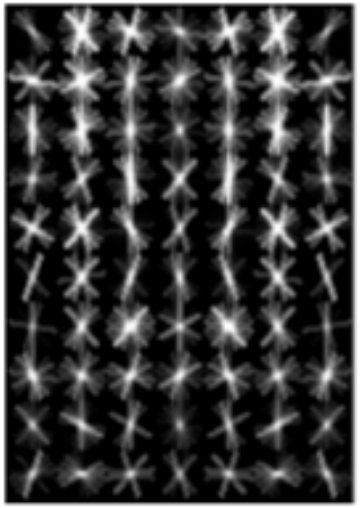
deformation
models

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

DPM: Initialization

Root filter for each component

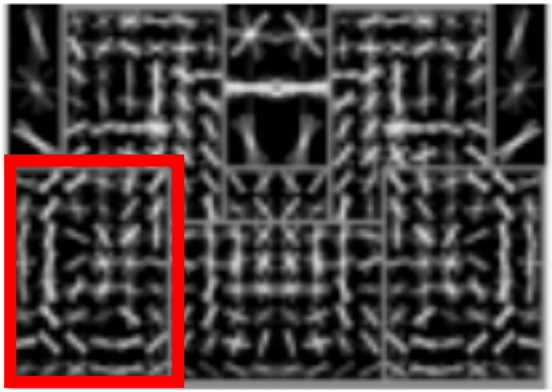


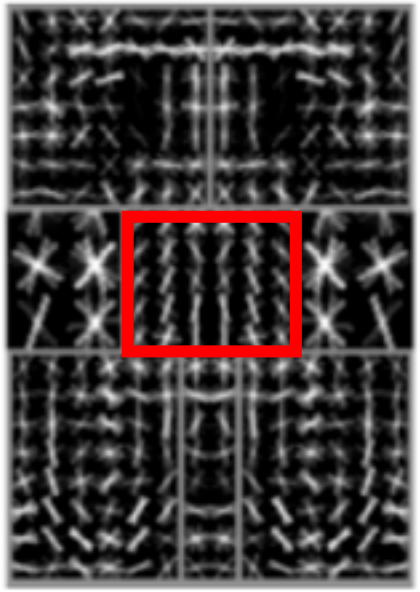


- For each component warp all positives to have same size
- Random pick negatives with same size
- Standard SVM no latent information

DPM: Initialization

Initializing Part Filter



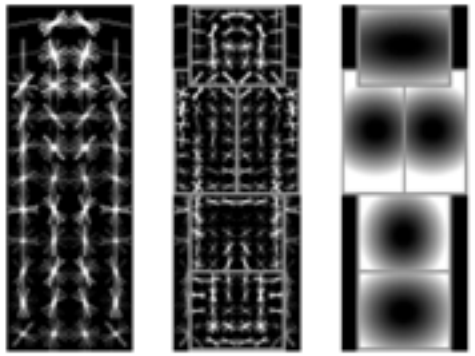


- Fixed number : 6 parts per component
- Choose the high-energy regions of the root filter
  (Energy : norm of positive weight in subwindow)
- Greedy approach: once part placed set to zero and find next high-energy part

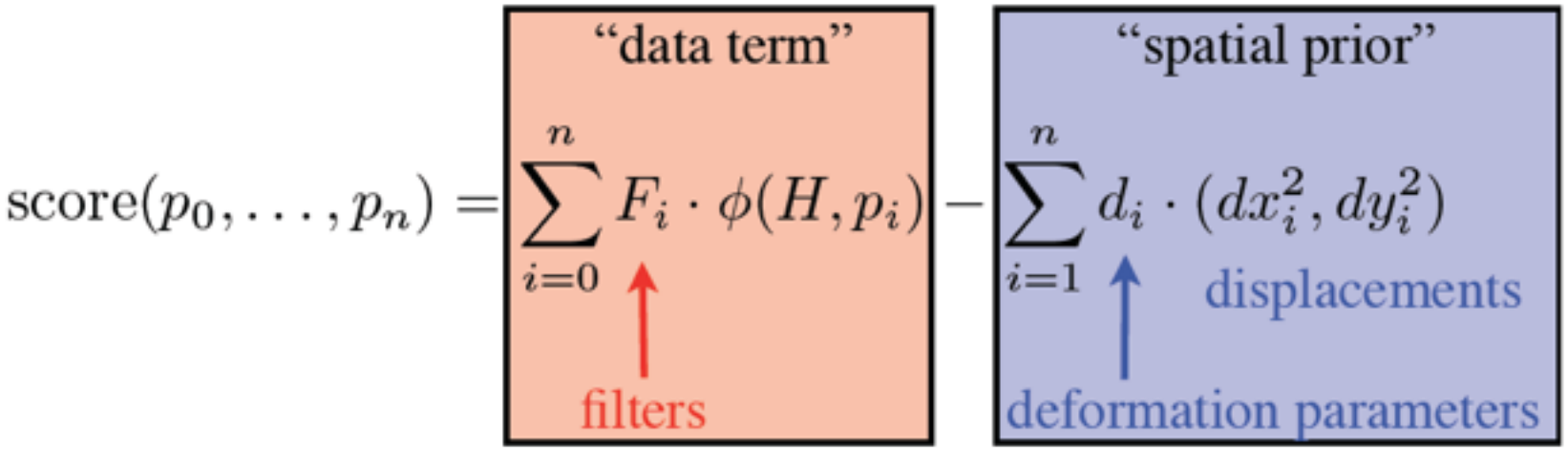
DPM: Training

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

Training



DPM: Detection



Score for one part at certain location : filter response score – deform cost relative to root

DPM: Detection

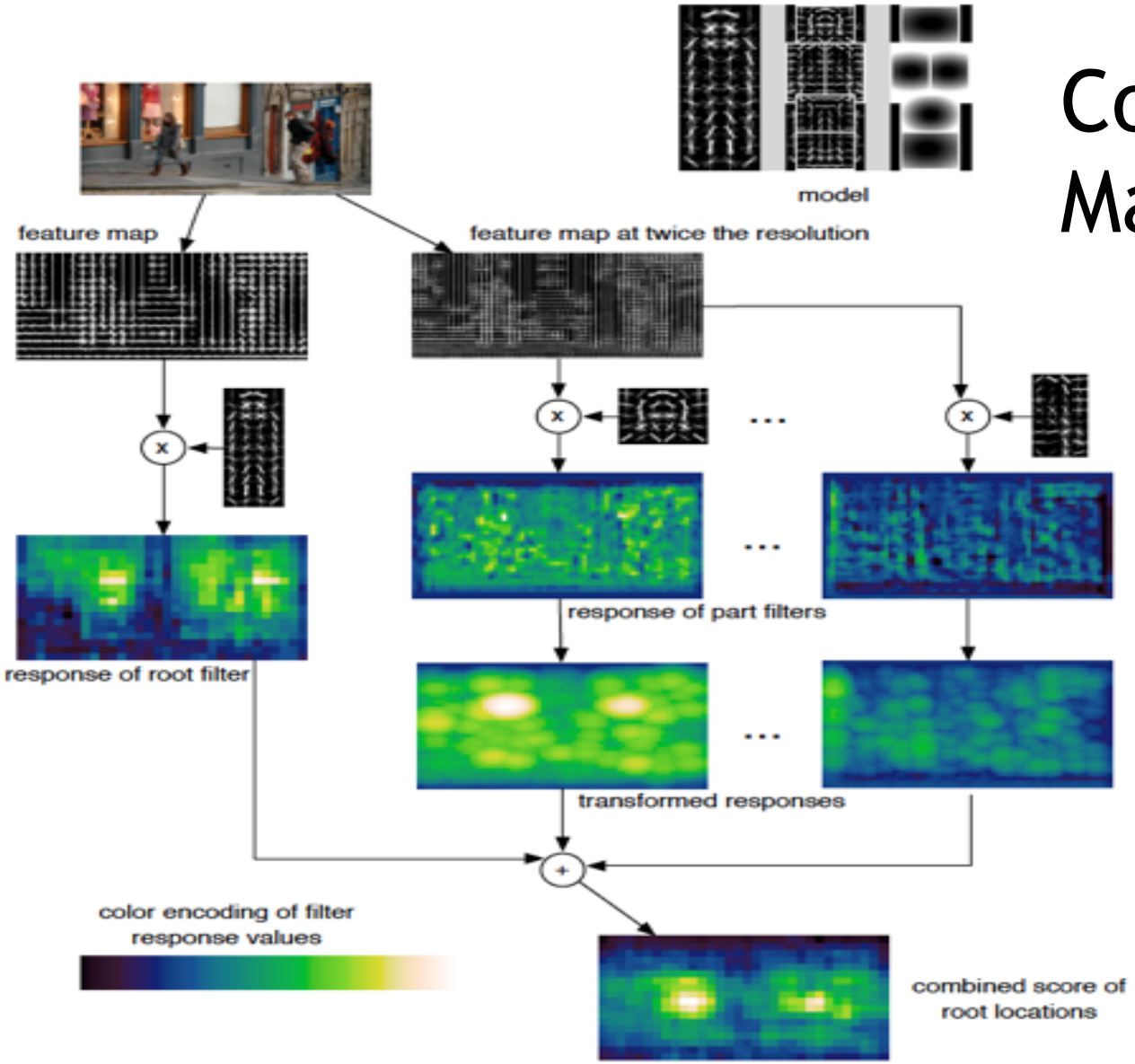
- Define an overall score for each root location
  - Based on best placement of parts

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

- High scoring root locations define detections

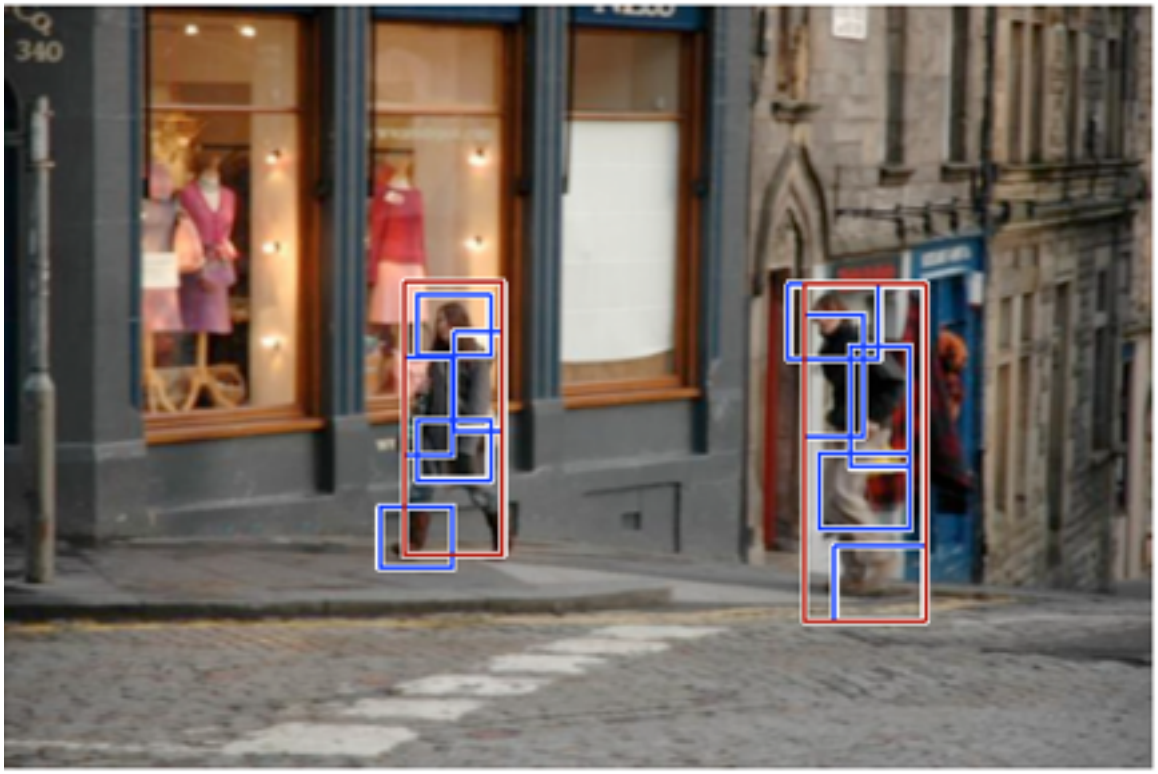
- Efficient computation: dynamic programming + generalized distance transforms

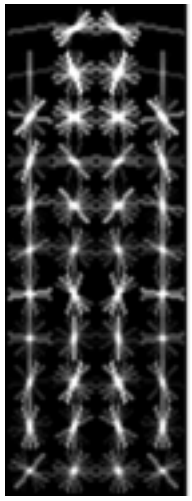
DPM: Detection

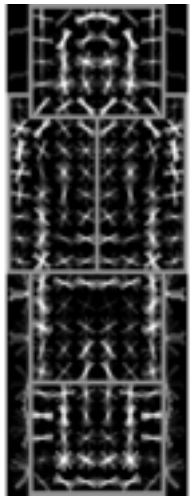


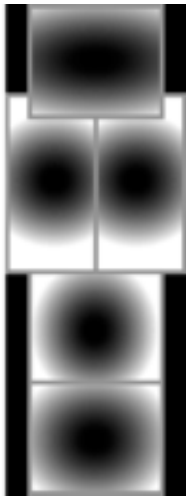
Combine Many Parts

DPM: Detection





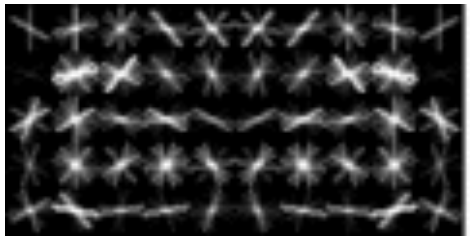
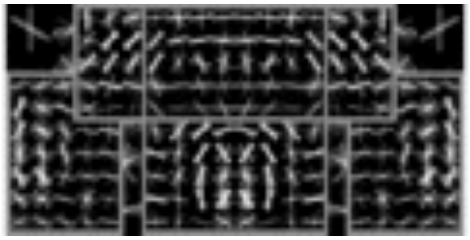
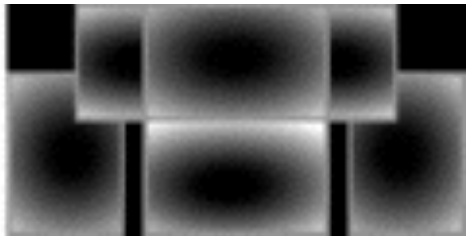




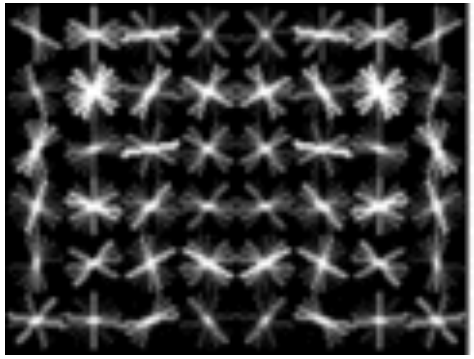
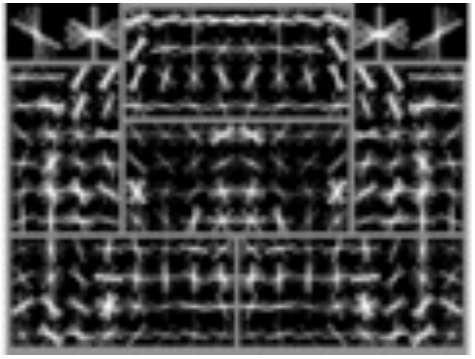
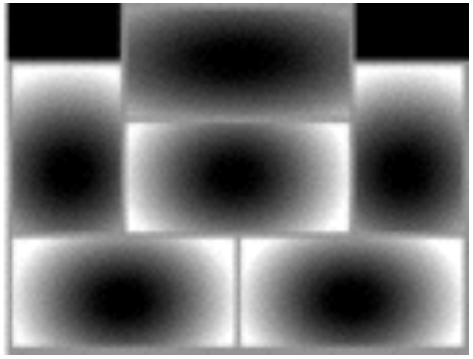
(after non-maximum suppression)
~1 second to search all scales

Car model

Component 1

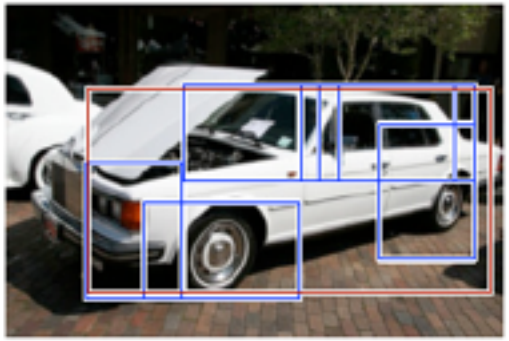




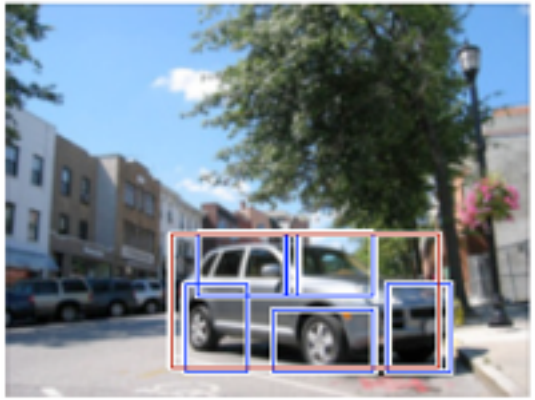
Component 2

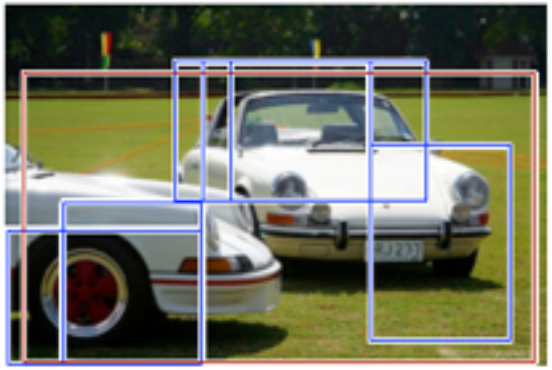
Car detections

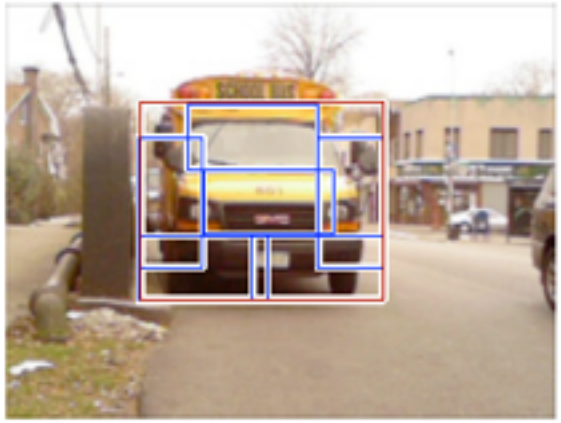
high scoring true positives





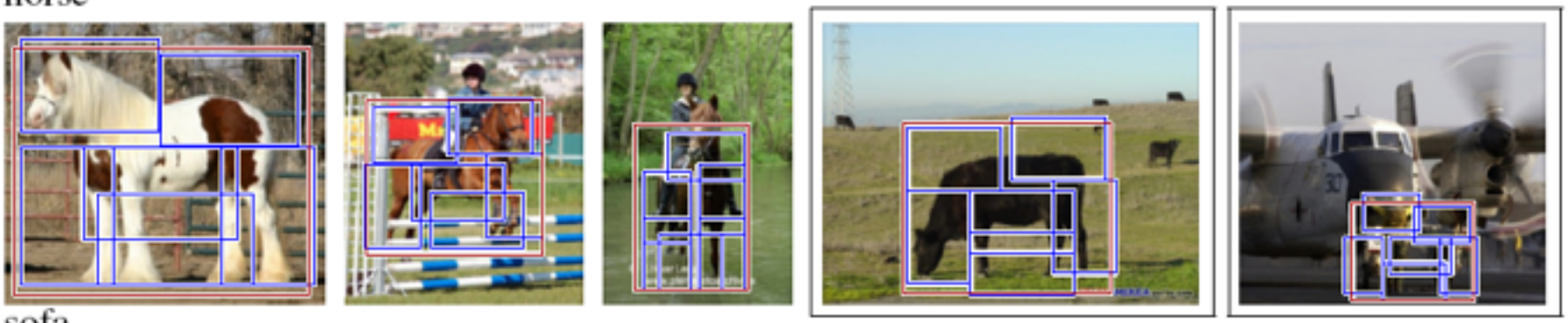
high scoring false positives



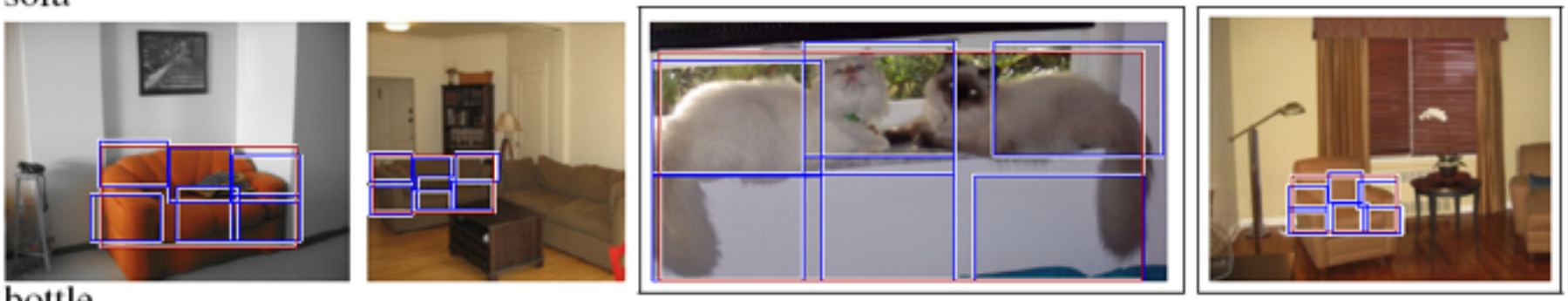


More detections

horse



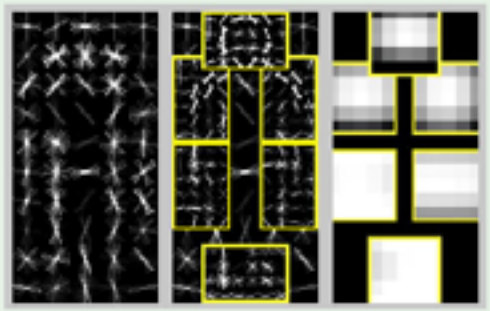
sofa



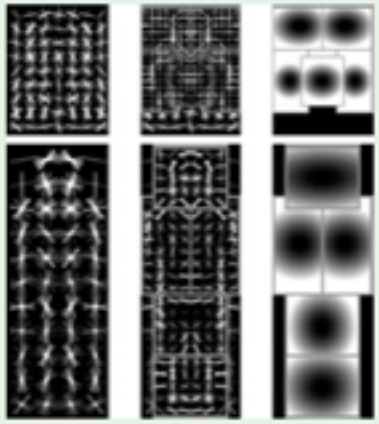
bottle

Summary of Results

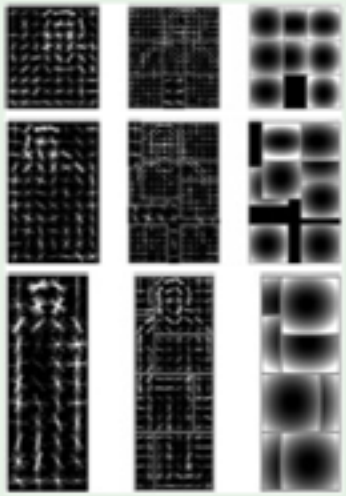
[DT'05]
AP 0.12



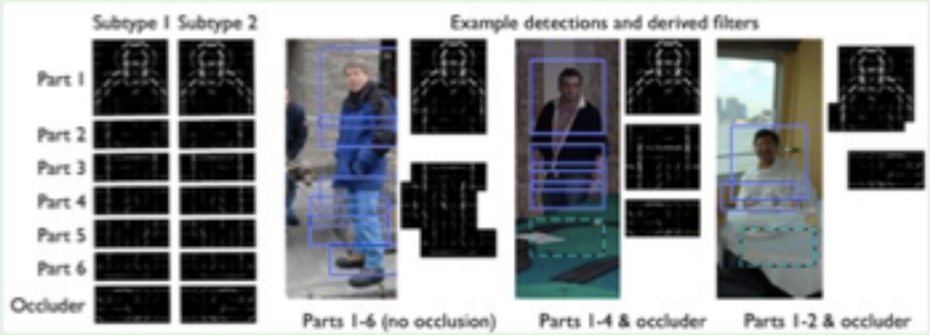
[FMR'08]
AP 0.27



[FGMR'10]
AP 0.36



[GFM voc-release5]
AP 0.45



[GFM'11]
AP 0.49

Poselet

Poselet

Poselets capture part of the pose from a given viewpoint

[Bourdev & Malik, ICCV09]

Poselet

Examples may differ visually but have common semantics

[Bourdev & Malik, ICCV09]

Poselet

One poselet one classifier

not a model for whole human body

How do we train a poselet?

How do we train a poselet?

given pose configuration

How do we train a poselet?

Finding correspondences at training time

Given part of a human pose

How do we find a similar pose configuration in the training set?

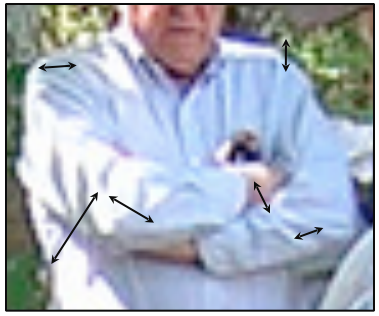
How do we train a poselet?

Finding correspondences at training time

We use key points to annotate the joints, eyes, nose, etc. of people

How do we train a poselet?

Finding correspondences at training time



Residual Error

Training poselet classifiers







Residual Error:	0.15	0.20	0.10	0.85	0.15	0.35

1. Given a seed patch
2. Find the closest patch for every other person
3. Sort them by residual error
4. Threshold them

Training poselet classifiers

1. Given a seed patch
2. Find the closest patch for every other person
3. Sort them by residual error
4. Threshold them
5. Use them as positive training examples to train a linear SVM with HOG features

Training poselet classifiers

One poselet one classifier not a model for whole human body

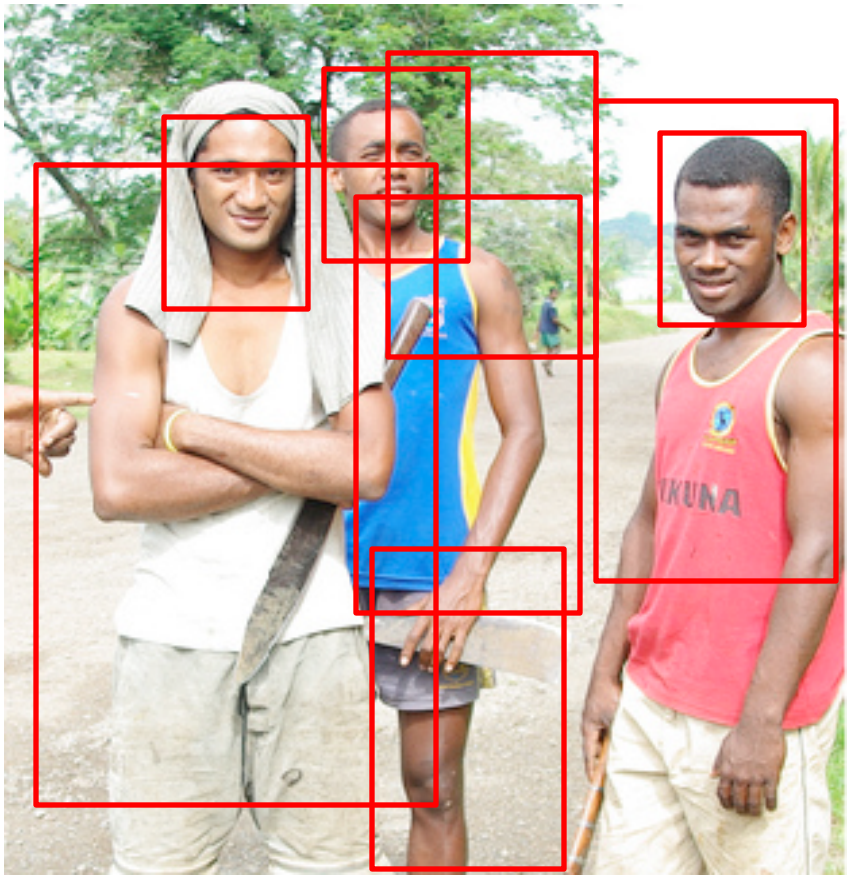
1. Given a seed patch
2. Find the closest patch for every other person
3. Sort them by residual error
4. Threshold them
5. Use them as positive training examples to train a linear SVM with HOG features

Testing

Goal

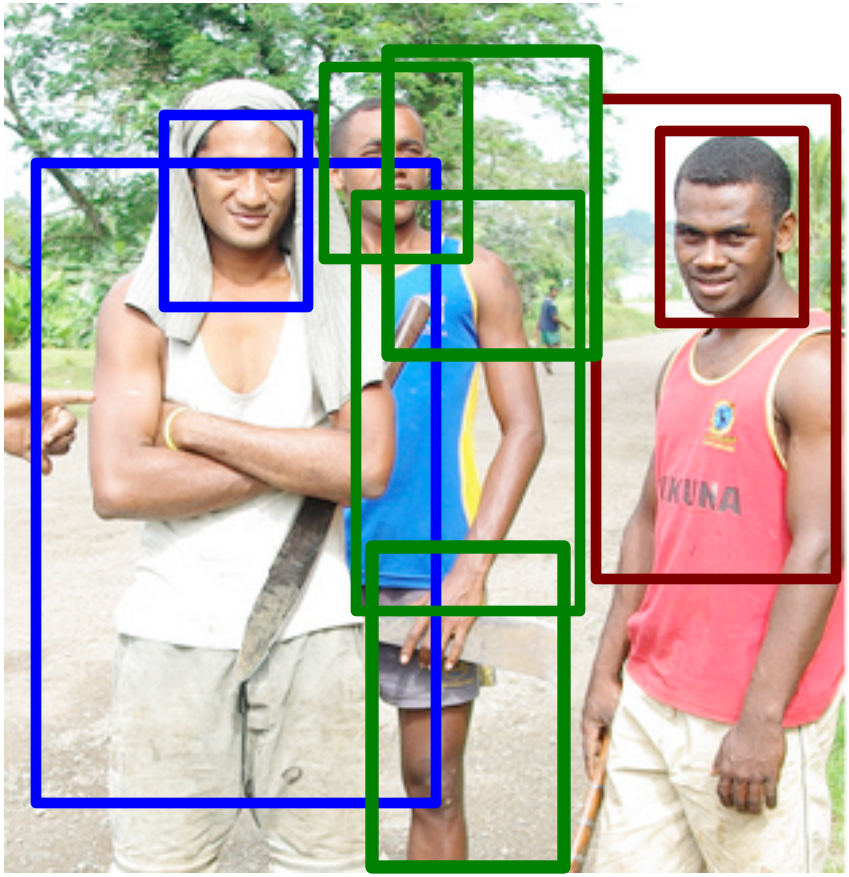
Testing

Step 1: Detect poselet activations



Testing

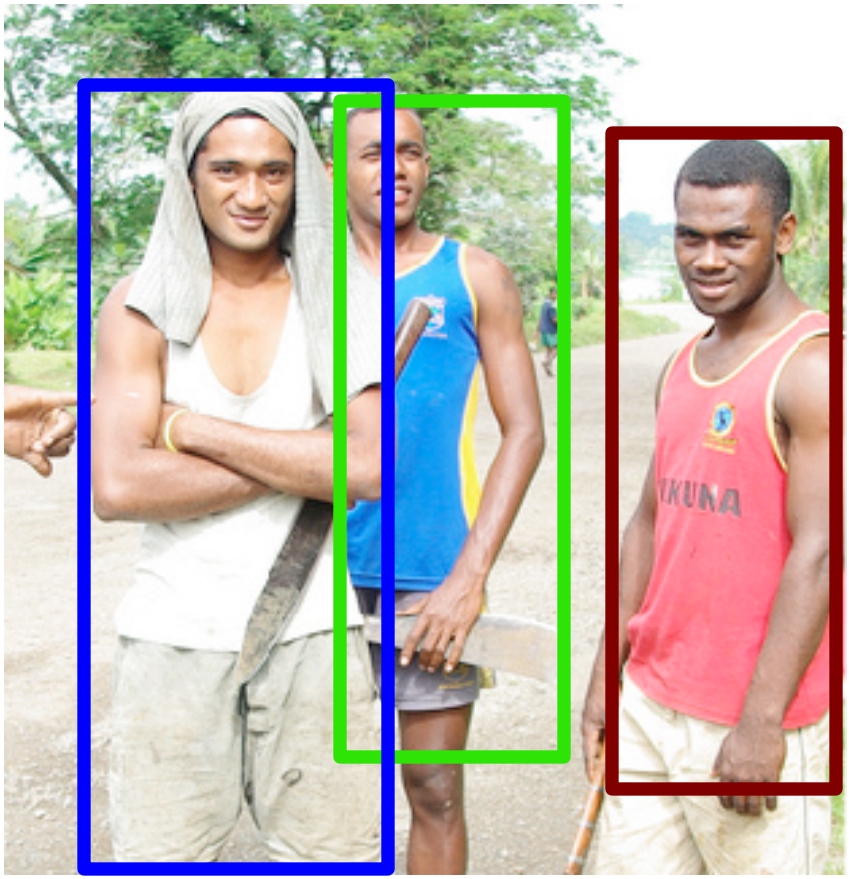
Step 2: Cluster the activations



Because we know the joint for each poselet

Testing

Step 3: Predict person bounds



Testing

Step 4: Identify the correct cluster

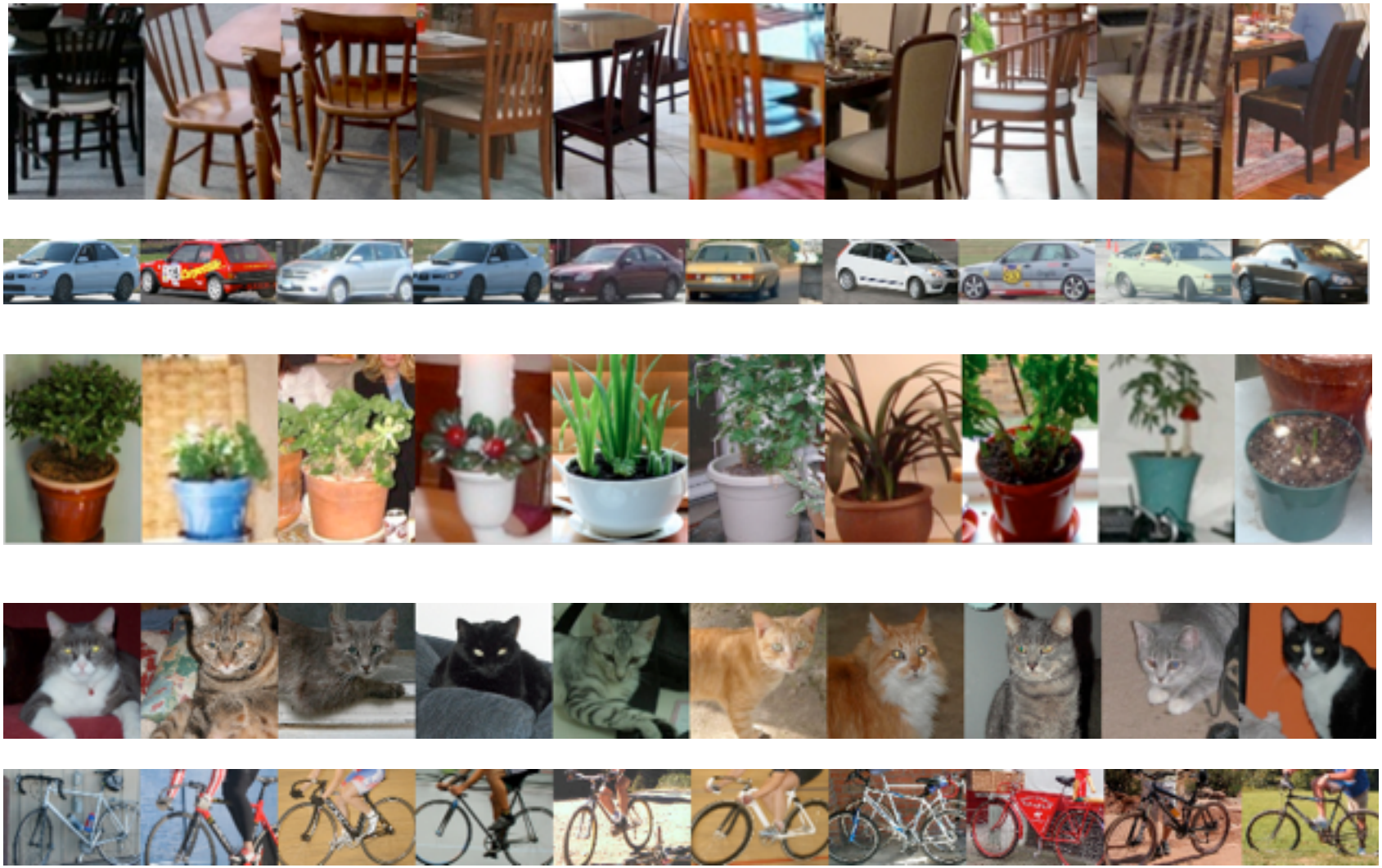
Max-flow in bipartite graph

Detection results on PASCAL

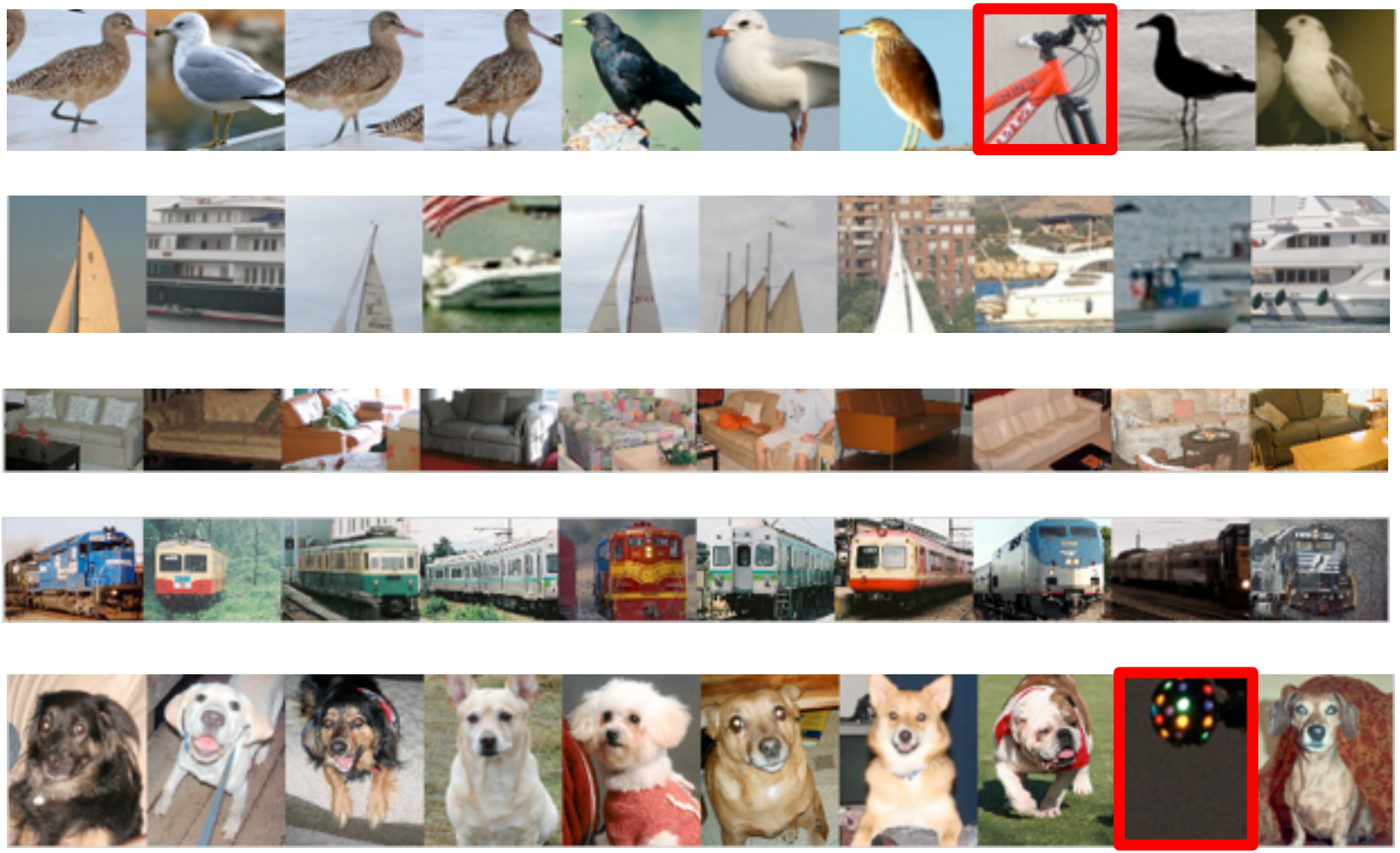
- **Person recognition:**

	Poselets	DPMs
2010	48.5%	47.7%
2009	48.3%	47.4%
2008	54.1%	43.1%
2007	46.9%	43.2%

Highest scoring hits on PASCAL test set



Highest scoring hits on PASCAL test set



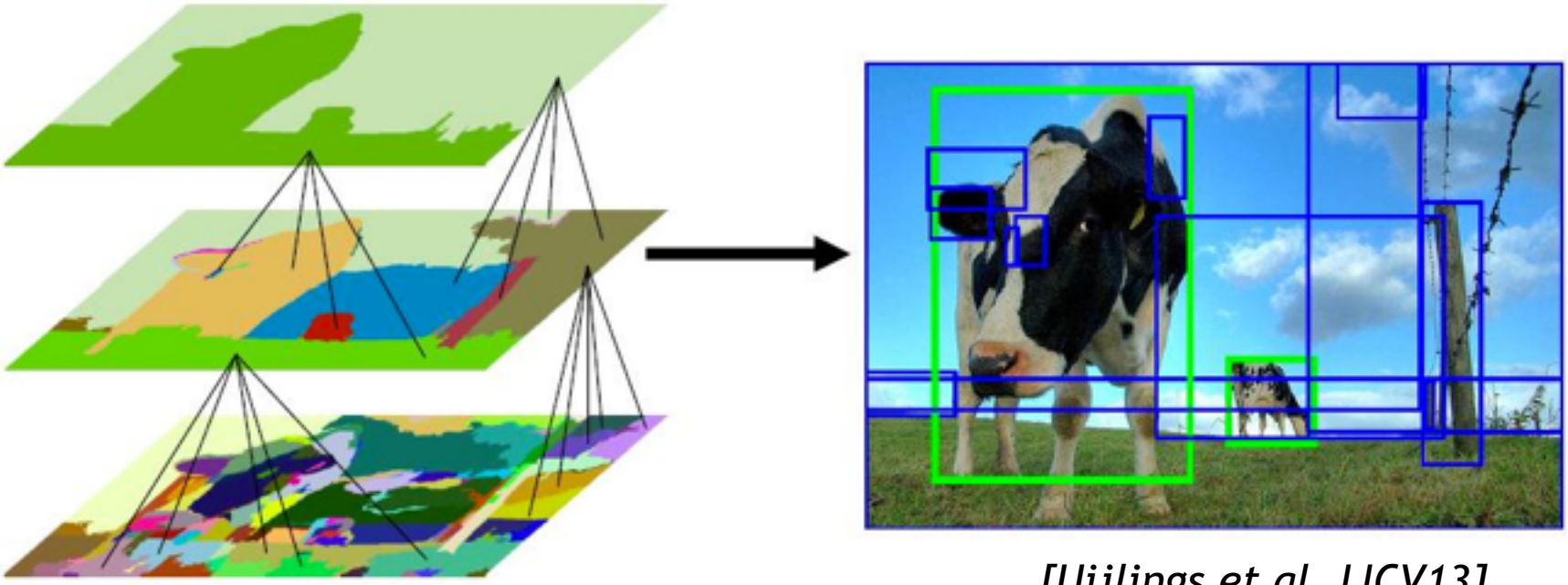
Roadmap (this lecture)

- Defining the Problem
- Rigid Template
  - HOG for human detection
  - Exemplar SVM detector
- Part Based Detector
  - Deformable Part Model
  - Poselets
- New development for object detection

New development for object detection

- Selective search
- RCNN

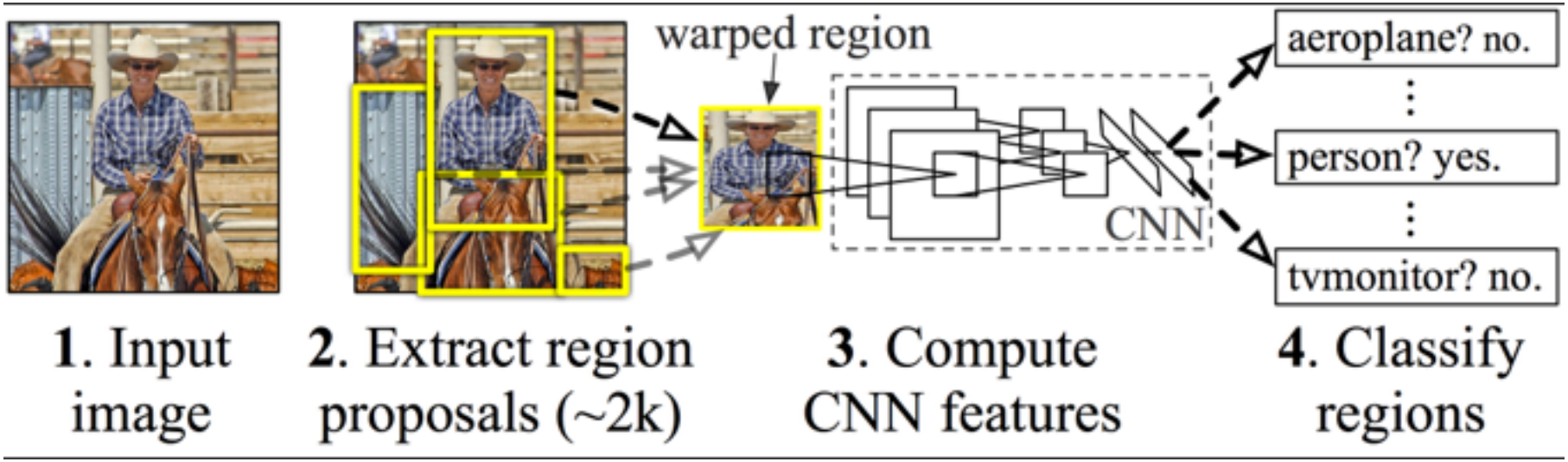
Selective Search for Object Recognition

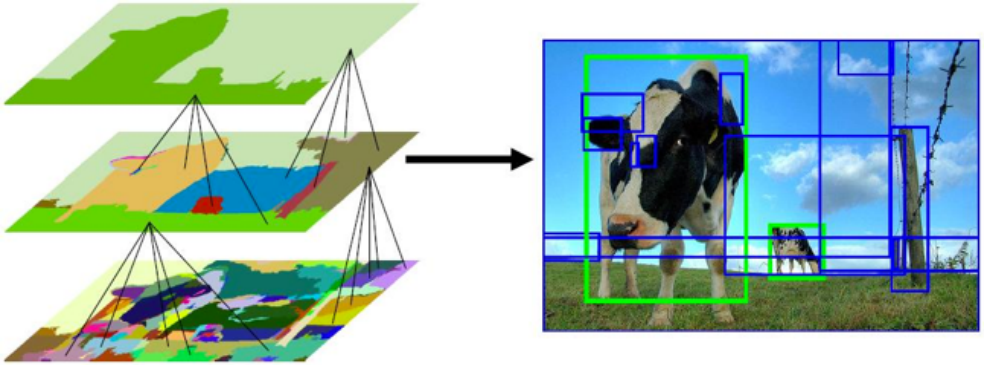


[Uijlings et al. IJCV13]

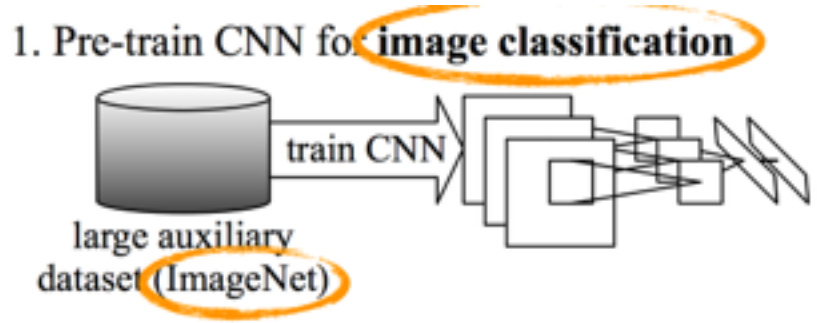
Like segmentation, we use the image structure to guide our sampling process.
Like exhaustive search, we aim to capture all possible object locations.

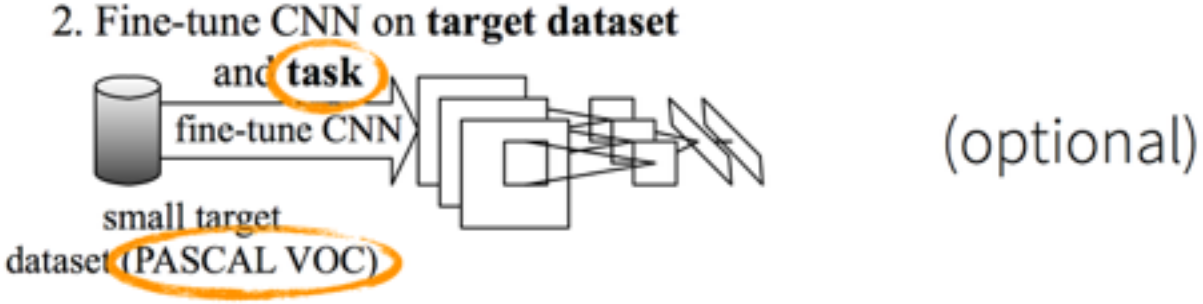
Deep learning in object detection



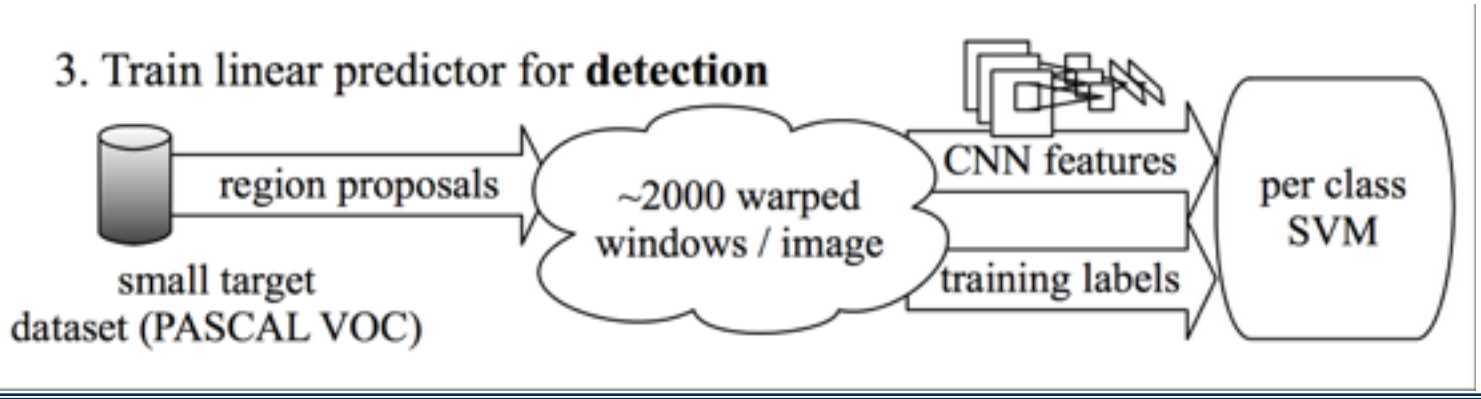


Training RCNN





(optional)



Results

	VOC 2007	VOC 2010
DPM v5 (Girshick et al. 2011)	33.7%	29.6%
UVA sel. search (Uijlings et al. 2012)		35.1%
Regionlets (Wang et al. 2013)	41.7%	39.7%
R-CNN $pool_5$	40.1%	
R-CNN fc_6	43.4%	
R-CNN fc_7	42.6%	
R-CNN FT $pool_5$	42.1%	
R-CNN FT fc_6	47.2%	
R-CNN FT fc_7	**48%**	**43.5%**

Fine-tuned - pre-trained

Roadmap (this lecture)

- Defining the Problem
- Rigid Template
  - HOG for human detection
  - Exemplar SVM detector
- Part Based Detector
  - Deformable Part Model
  - Poselets
- New development for object detection