# Computer Vision

Bogdan Alexe

bogdan.alexe@fmi.unibuc.ro

University of Bucharest, 2nd semester, 2023-2024

# Course structure

1. **Features and filters: low-level vision**

   Linear filters, color, texture, edge detection, template matching

2. **Grouping and fitting: mid-level vision**

   Fitting curves and lines, robust fitting, RANSAC, Hough transform, segmentation

3. **Multiple views**

   Local invariant feature and description, epipolar geometry and stereo, object instance recognition

4. **Object Recognition: high – level vision**

   Object classification, object detection, part based models, bovw models

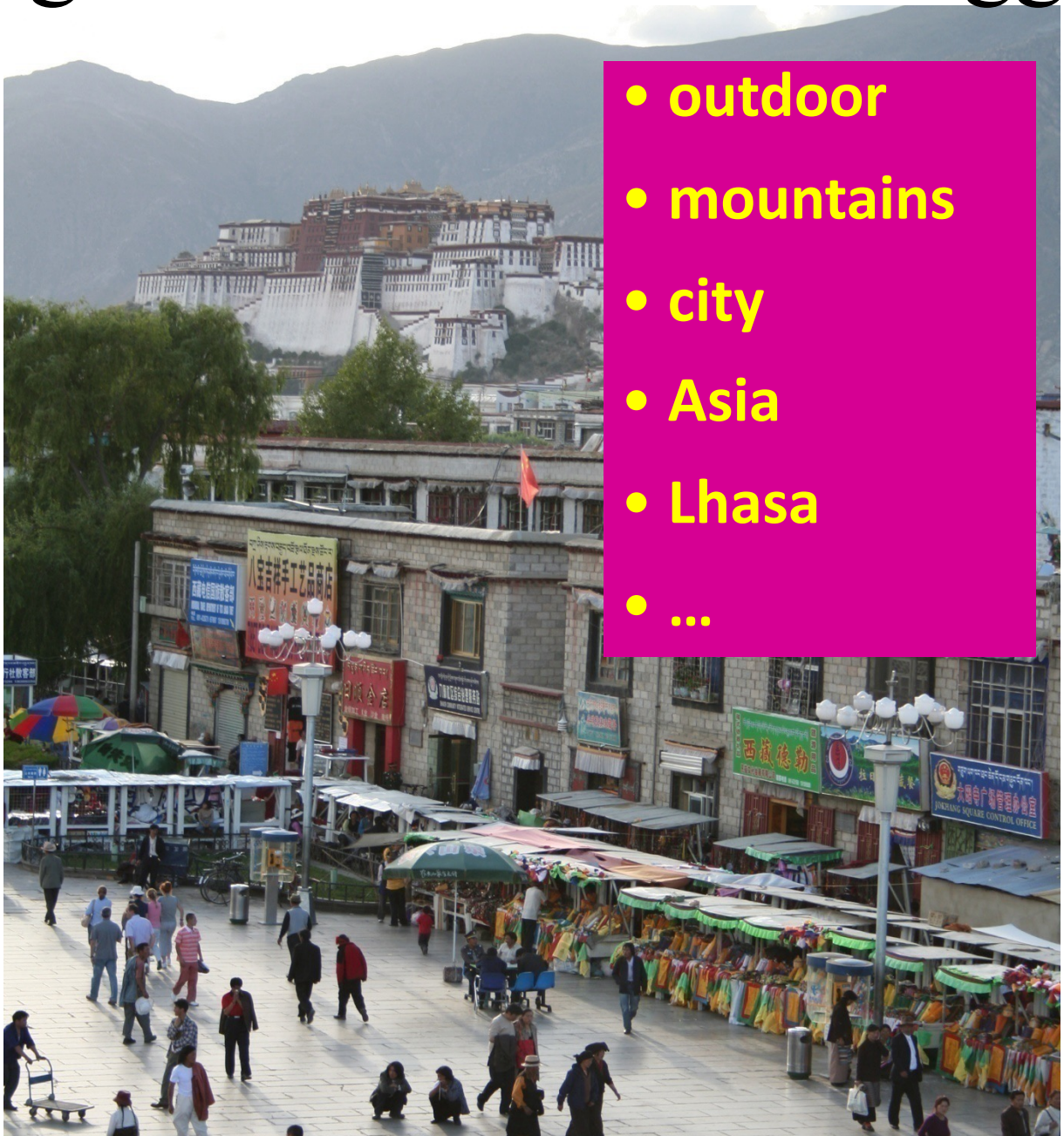
5. **Video understanding**

   Object tracking, background subtraction, motion descriptors, optical flow

# Common recognition tasks

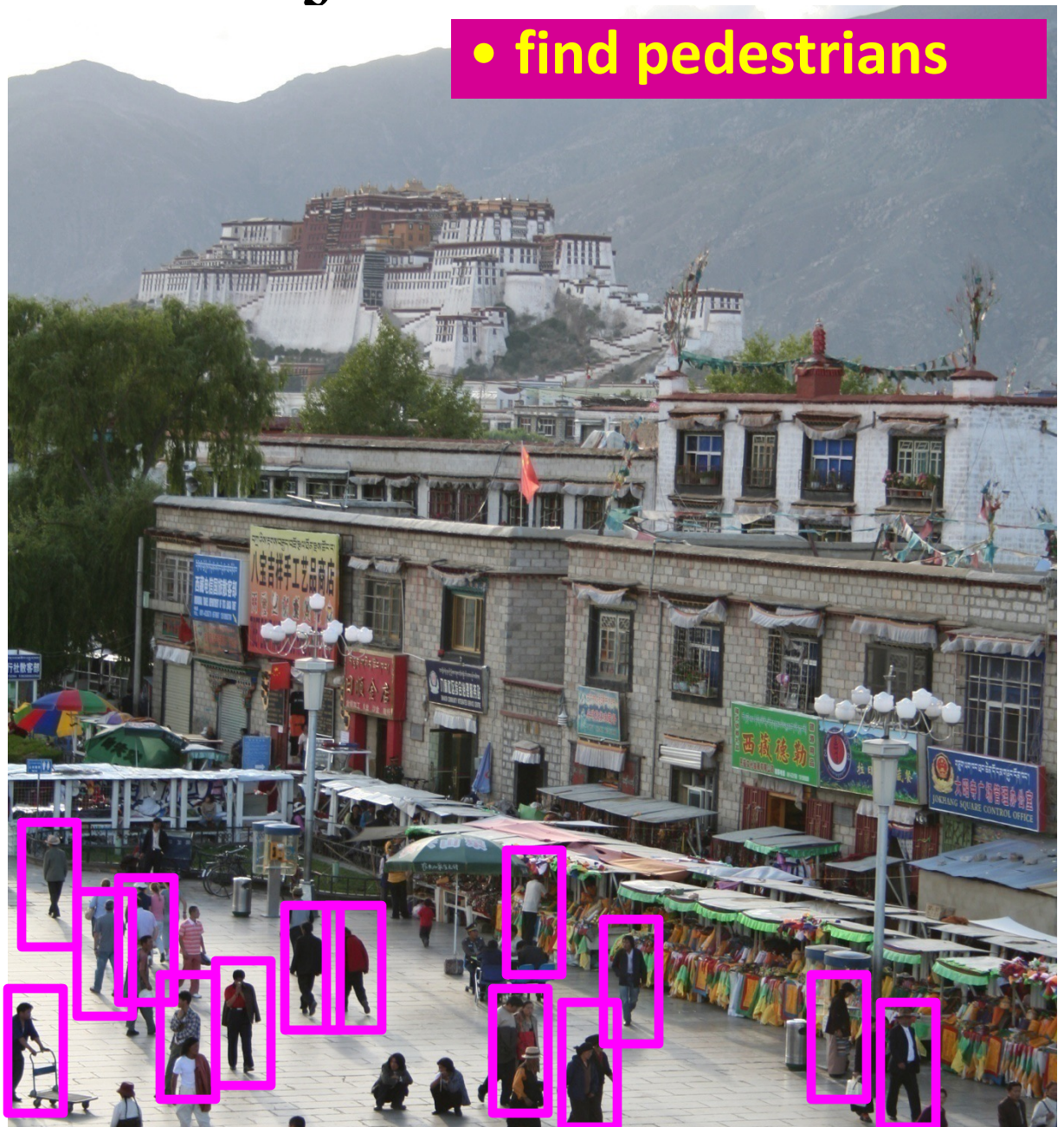
Slide credit
Fei-Fei Li

# Image classification and tagging



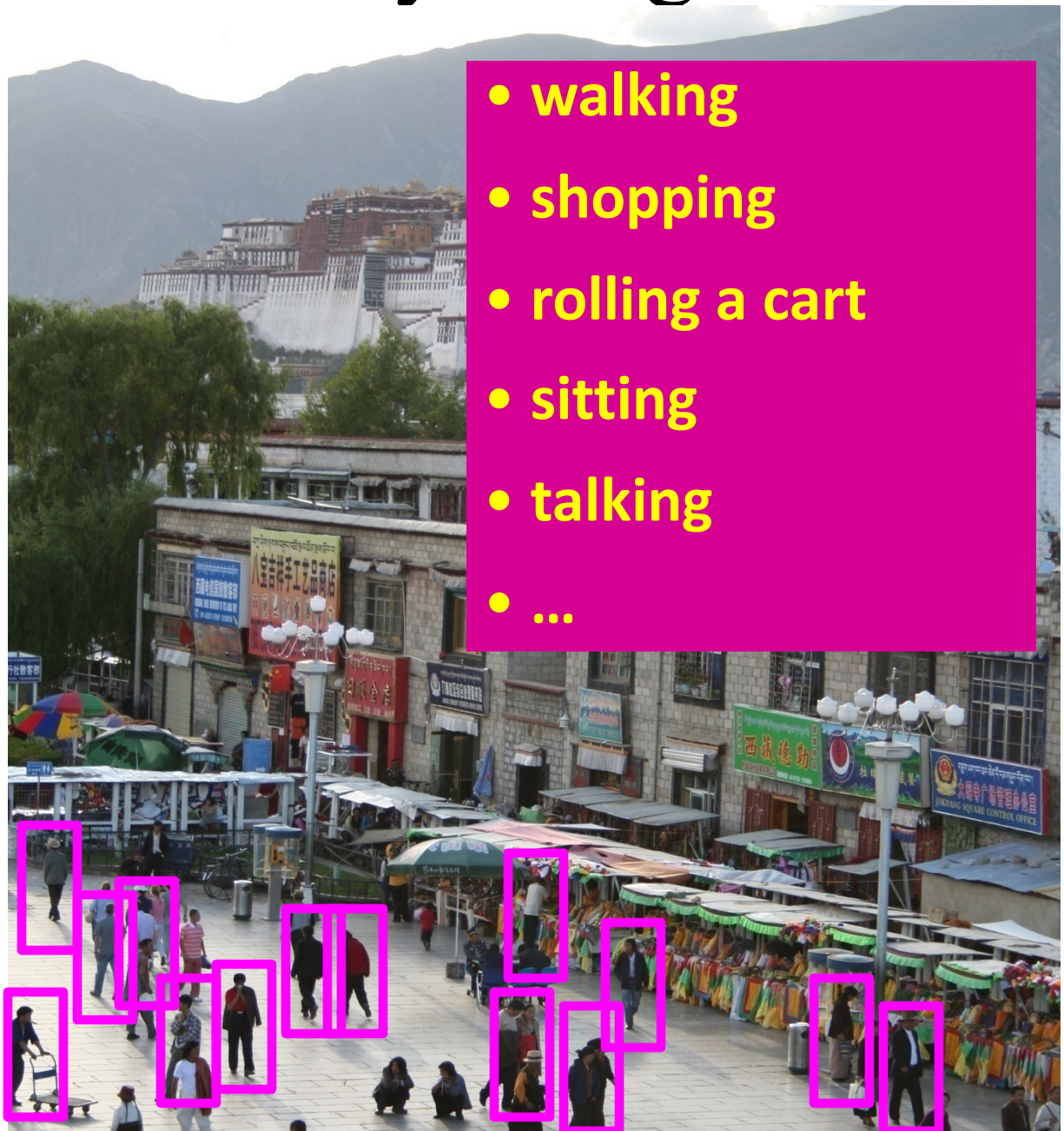

Slide credit
Fei-Fei Li

# Object detection




Slide credit
Fei-Fei Li

# Activity recognition




Slide credit
Fei-Fei Li

# Semantic segmentation

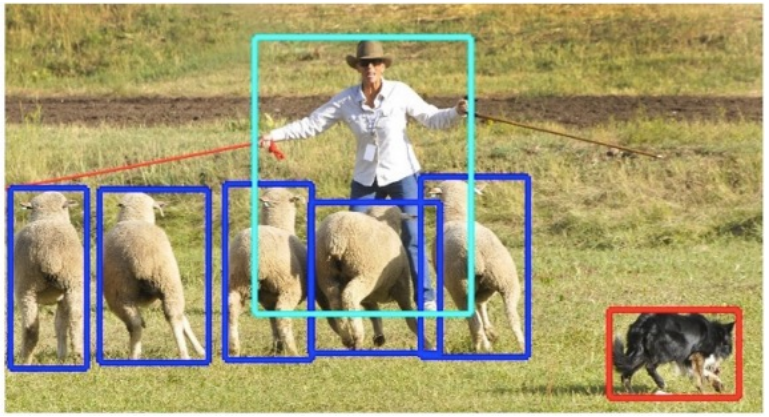
Slide credit
Fei-Fei Li

# Semantic segmentation

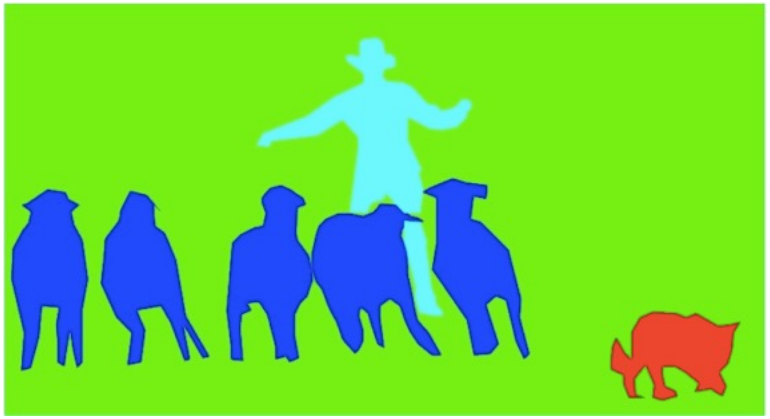

Slide credit
Fei-Fei Li

# Classification, detection, semantic segmentation, instance segmentation


image classification



object detection



semantic segmentation

instance segmentation

# Image classification

# The statistical learning framework

- Apply a prediction function to a feature representation of the image to get the desired output:

f(  ) = “apple”

f(  ) = “tomato”

f(  ) = “cow”

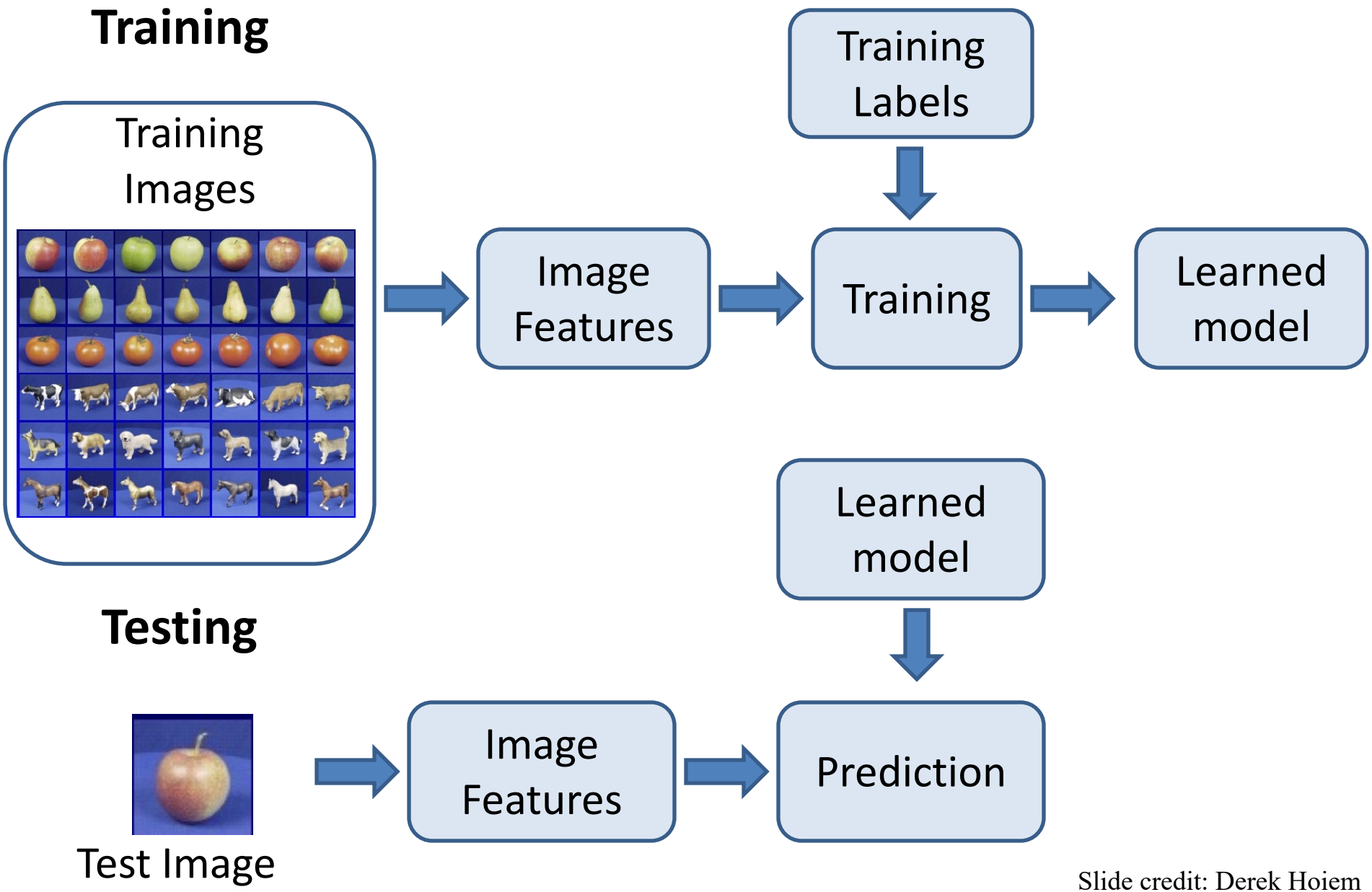
# The statistical learning framework

$$y = f(\mathbf{x})$$

output prediction function feature representation

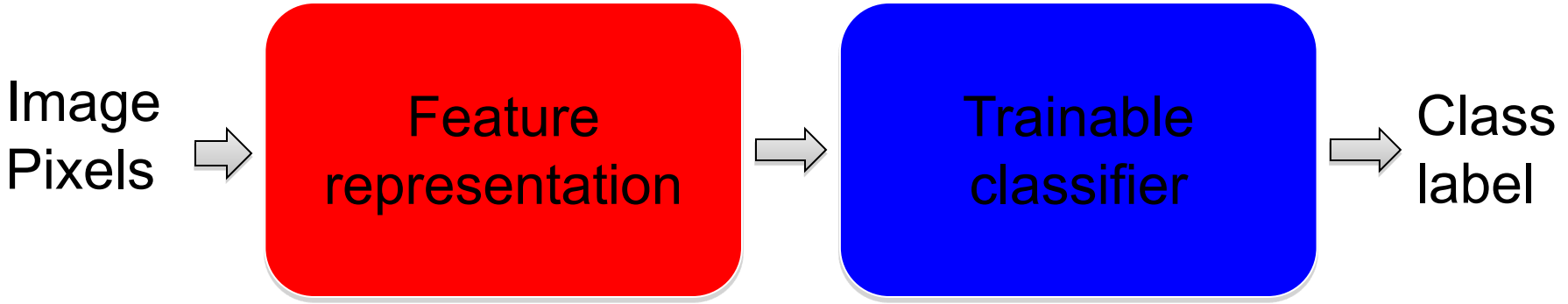
- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function $f$ by minimizing the prediction error on the training set
- **Testing:** apply $f$ to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

# Steps




Slide credit: Derek Hoiem

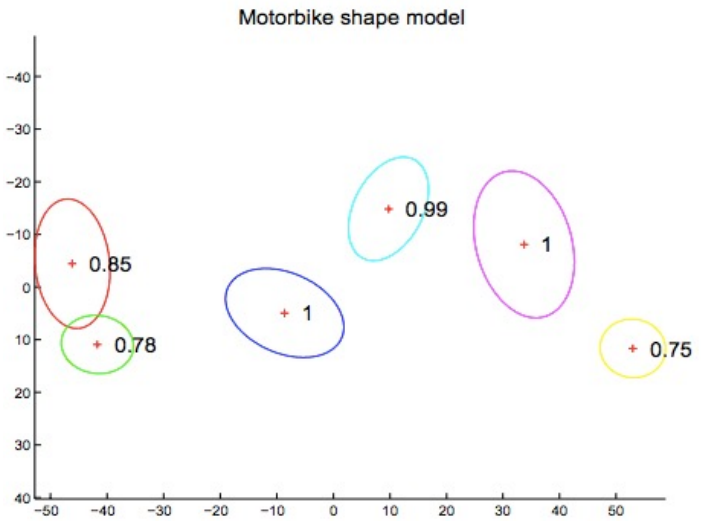
# “Classic” recognition pipeline



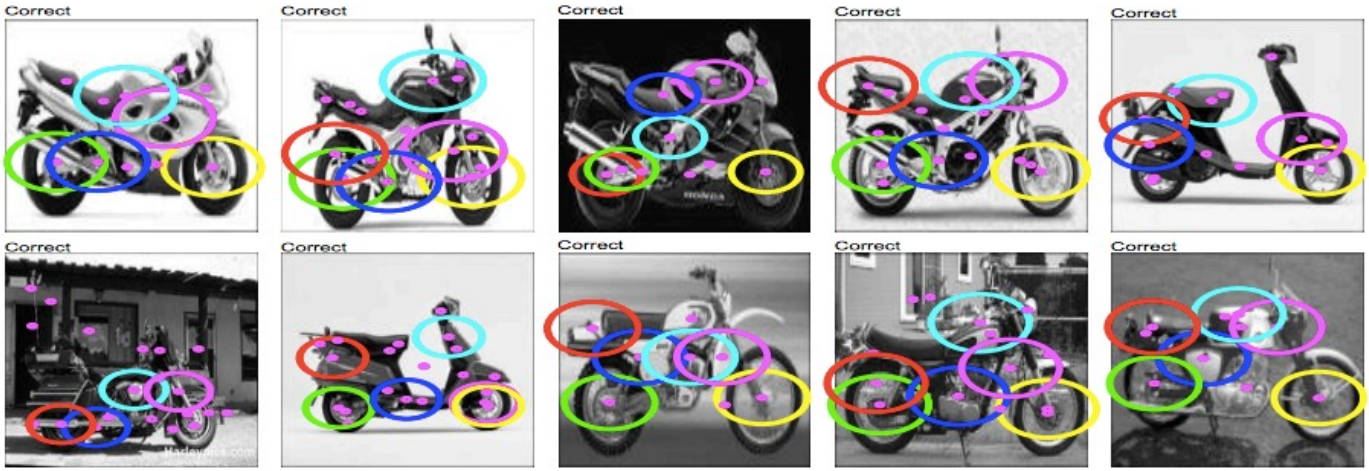

- Hand-crafted feature representation (classic computer vision)
- Off-the-shelf trainable classifier (SVM, k-NN, neural network)

# “Classic” representation: Bag of (visual) features

# Motivation 1: Part-based models

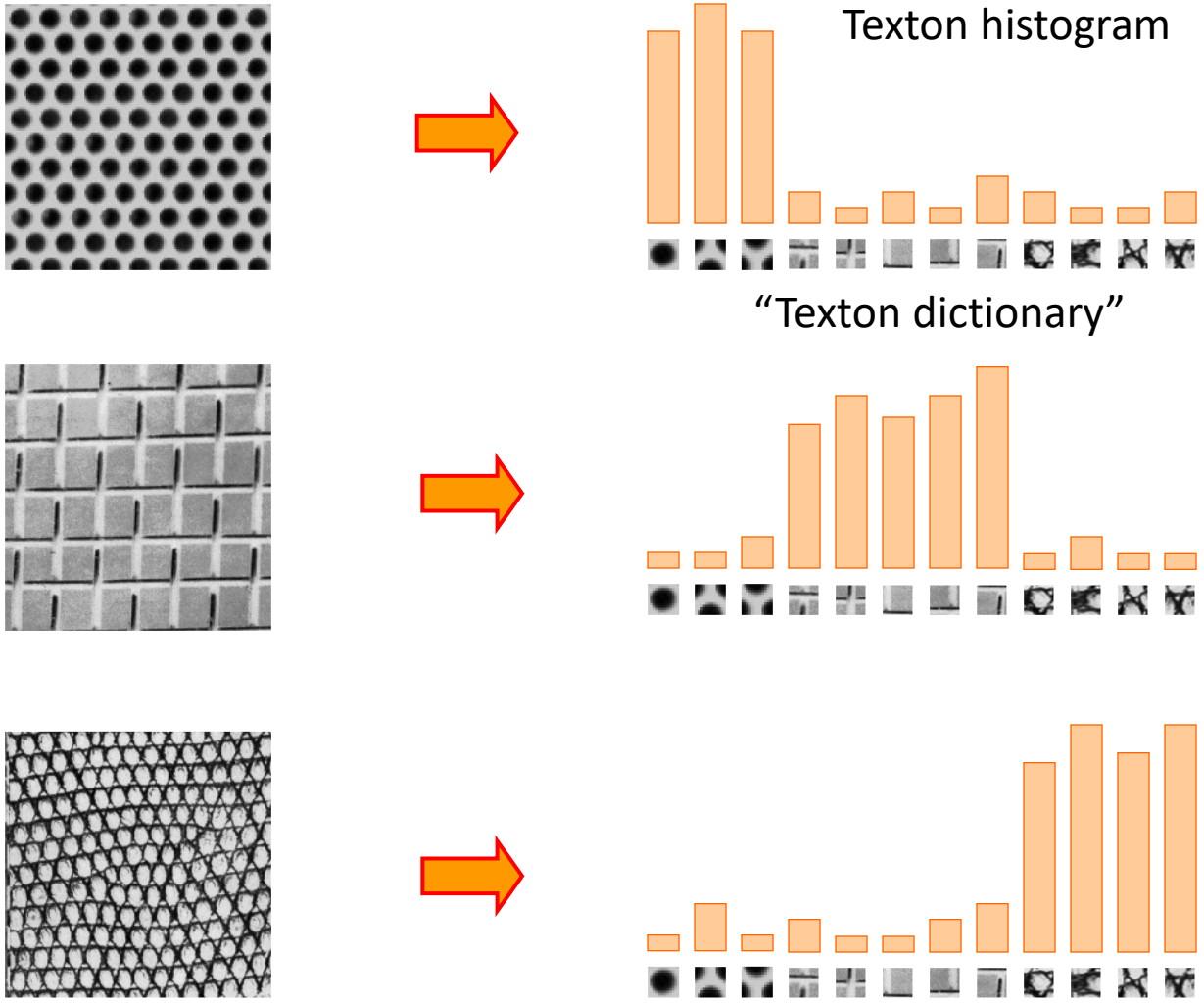







Weber, Welling & Perona (2000), Fergus, Perona & Zisserman (2003)

# Motivation 2: Texture models




Julesz, 1981; Cula & Dana, 2001; Leung & Malik 2001; Mori, Belongie & Malik, 2001; Schmid 2001; Varma & Zisserman, 2002, 2003; Lazebnik, Schmid & Ponce, 2003

# Motivation 3: Bags of words

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

# Motivation 3: Bags of words

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

# Motivation 3: Bags of words

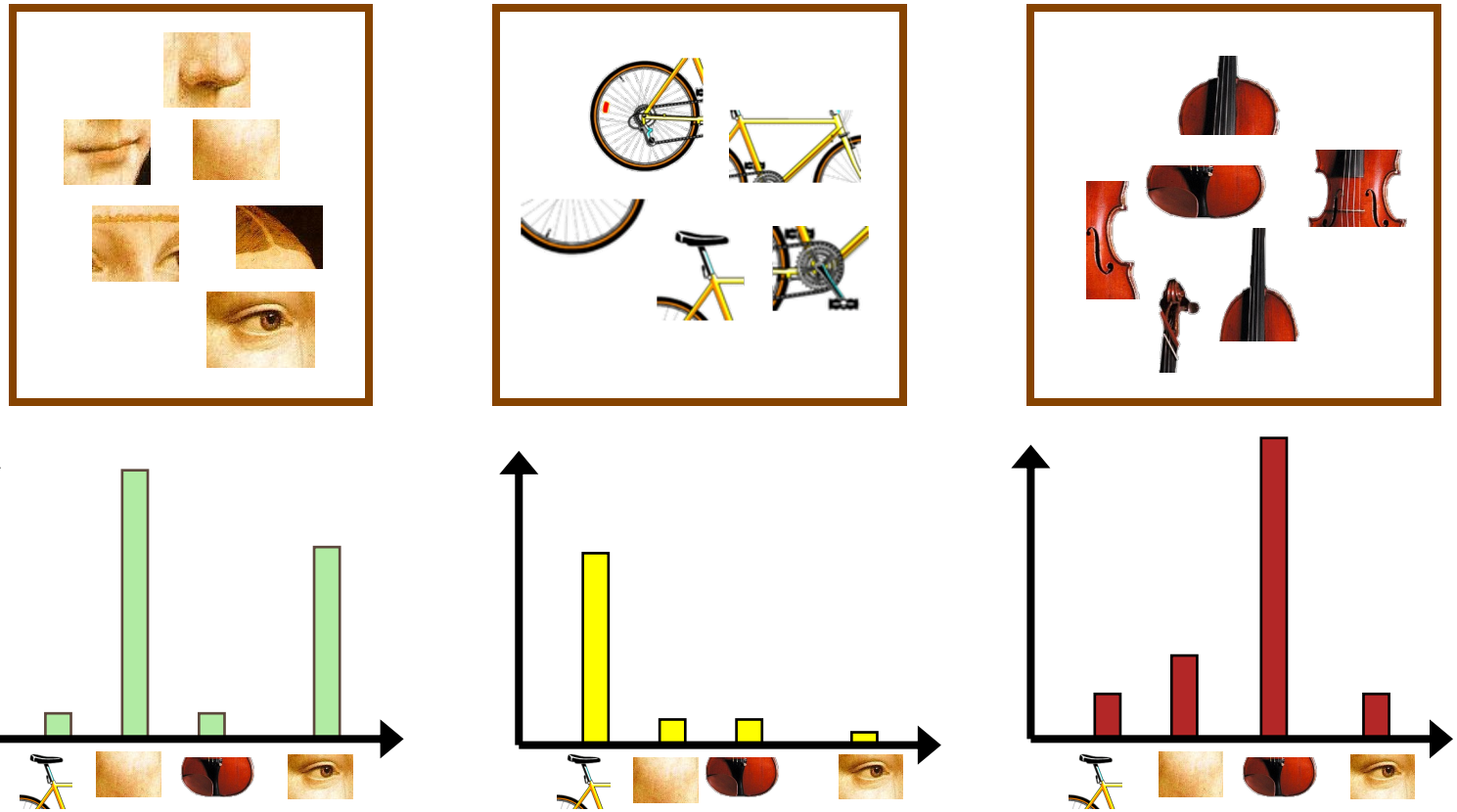
- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

# Motivation 3: Bags of words

- Orderless document representation: frequencies of words from a dictionary Salton & McGill (1983)

# Bag of features: Outline

1. Extract local features
2. Learn “visual vocabulary”
3. Quantize local features using visual vocabulary
4. Represent images by frequencies of “visual words”



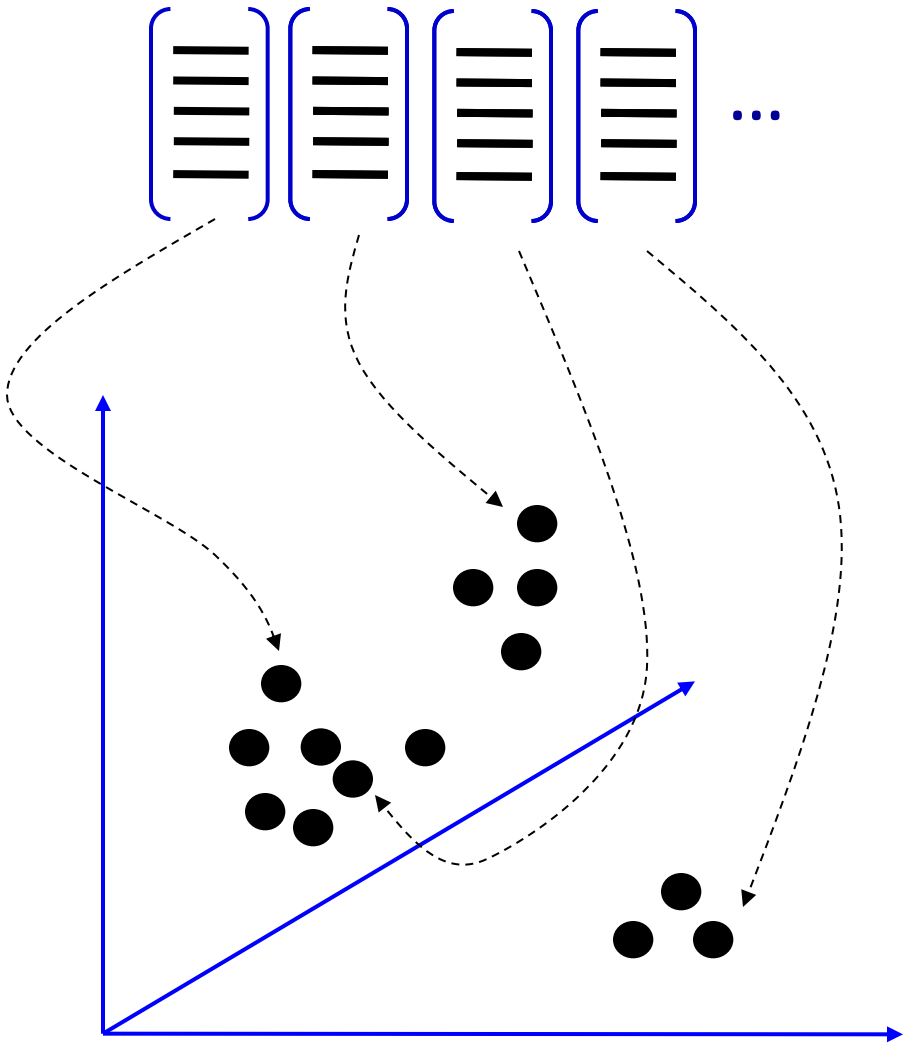
# 1. Local feature extraction

- Sample patches and extract descriptors

Use interest points

Use a dense grid

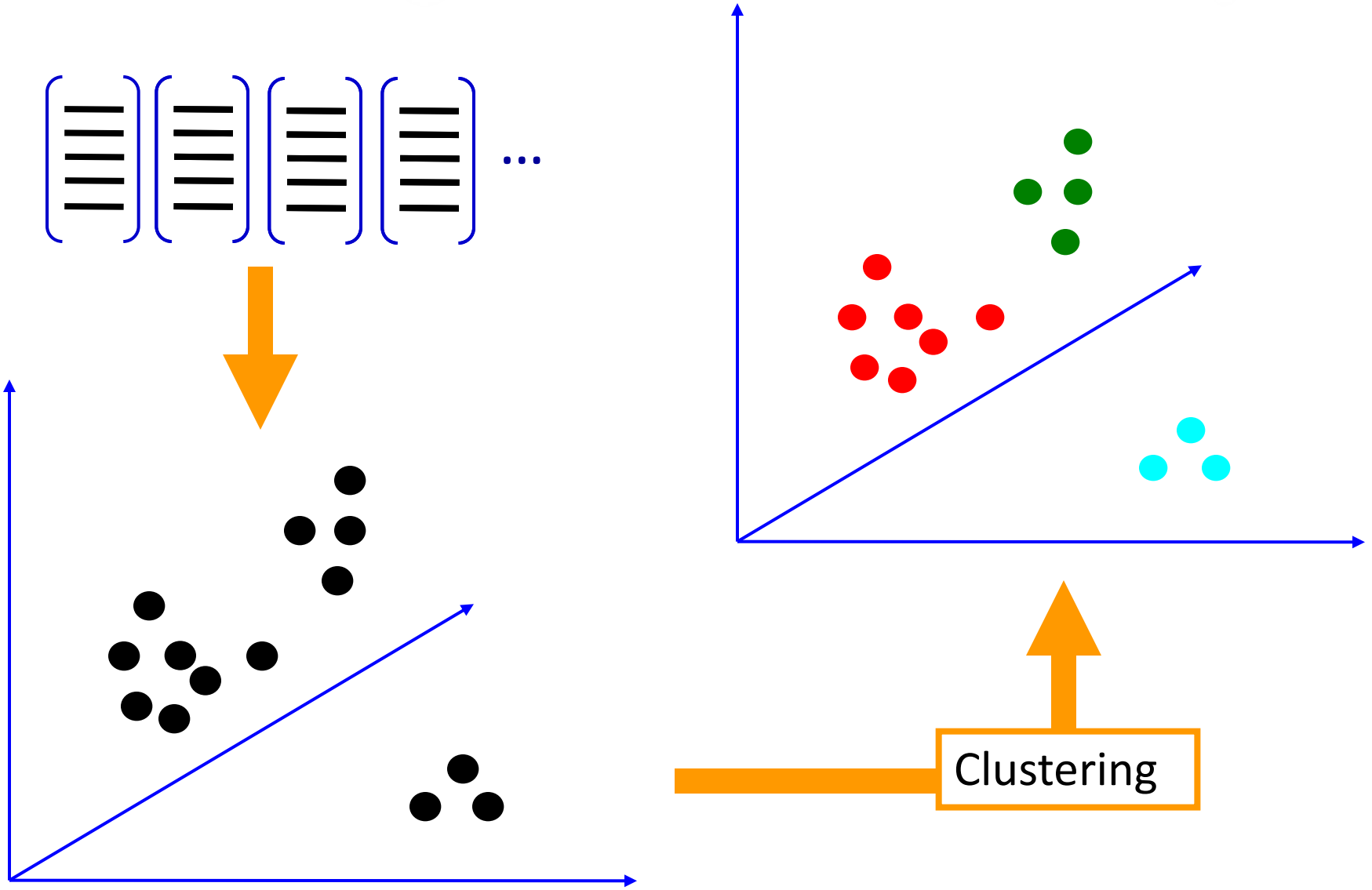
# 2. Learning the visual vocabulary



Extracted descriptors from the training set

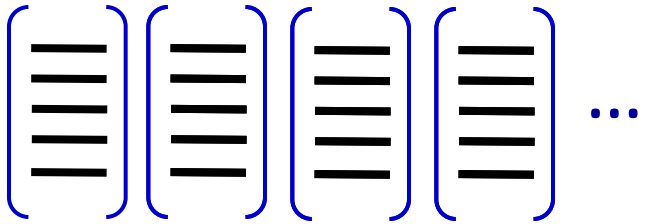
Slide credit: Josef Sivic

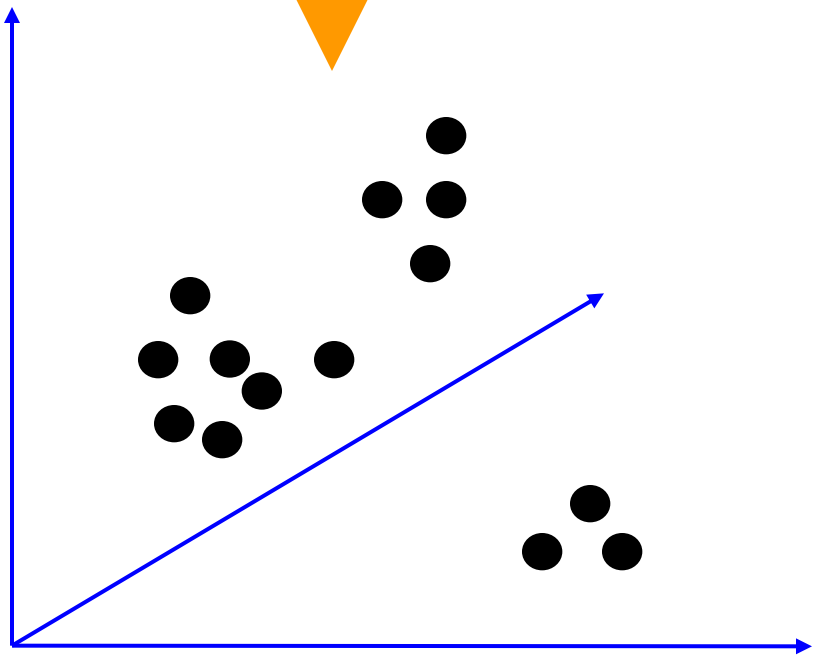
# 2. Learning the visual vocabulary

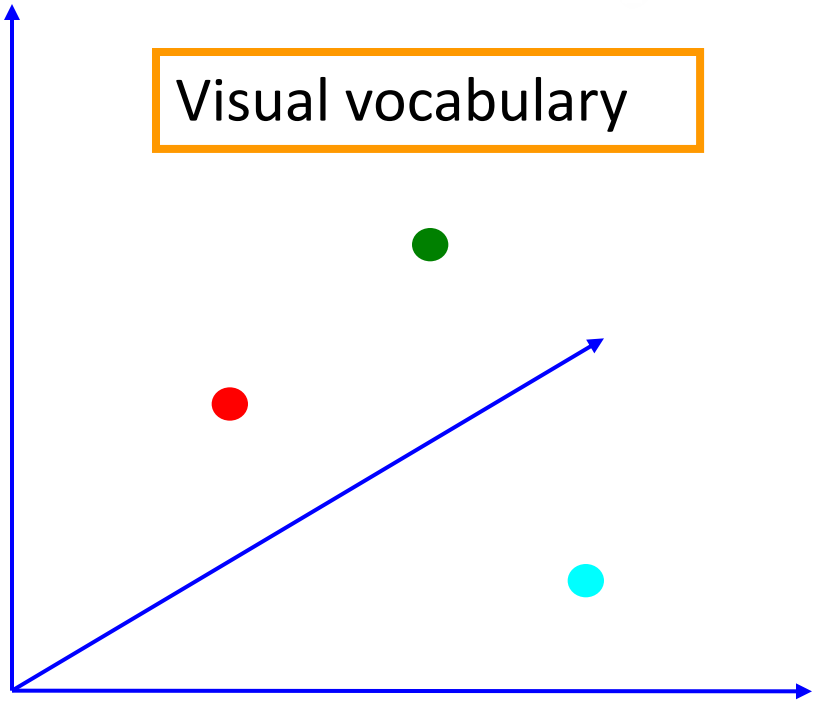



Slide credit: Josef Sivic

# 2. Learning the visual vocabulary









Slide credit: Josef Sivic

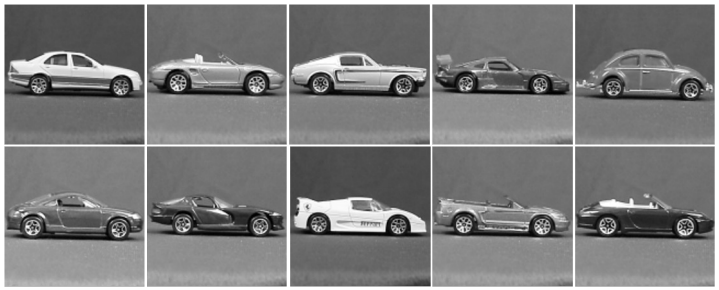
# Recall: K-means clustering

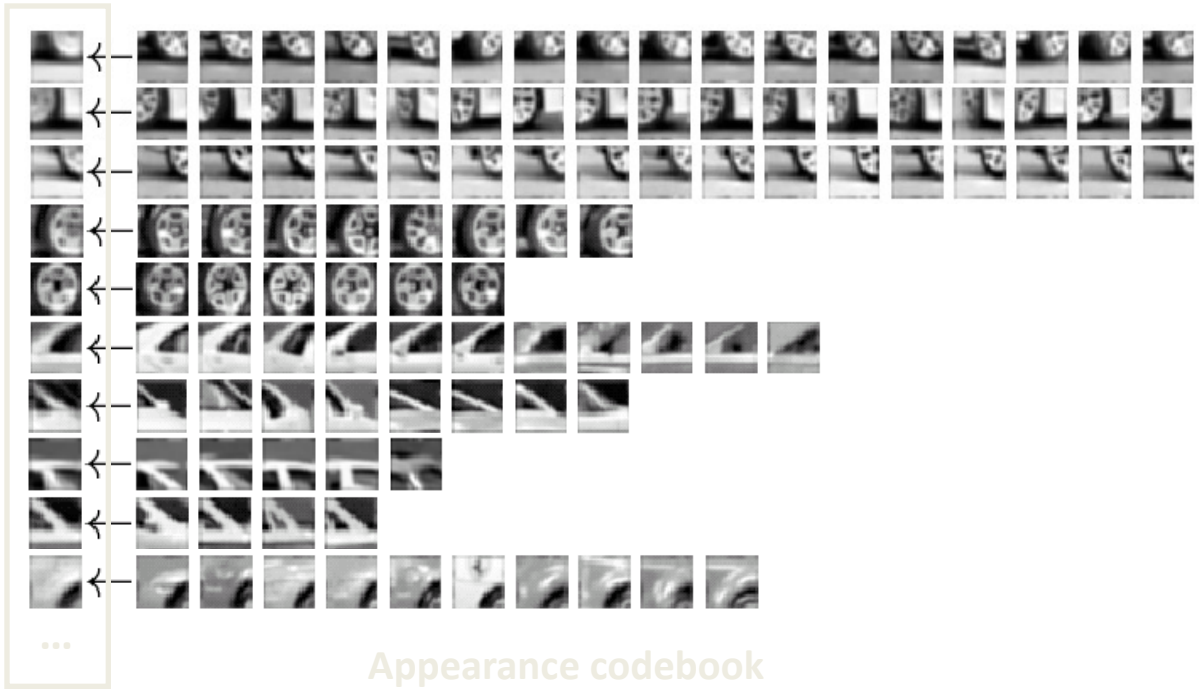
- Want to minimize sum of squared Euclidean distances between features $\mathbf{x}_i$ and their nearest cluster centers $\mathbf{m}_k$

$$D(X, M) = \sum_{\text{cluster } k} \sum_{\substack{\text{point } i \text{ in} \\ \text{cluster } k}} (\mathbf{x}_i - \mathbf{m}_k)^2$$

- Algorithm:
- Randomly initialize K cluster centers
- Iterate until convergence:
  - Assign each feature to the nearest center
  - Recompute each cluster center as the mean of all features assigned to it

# Visual vocabularies

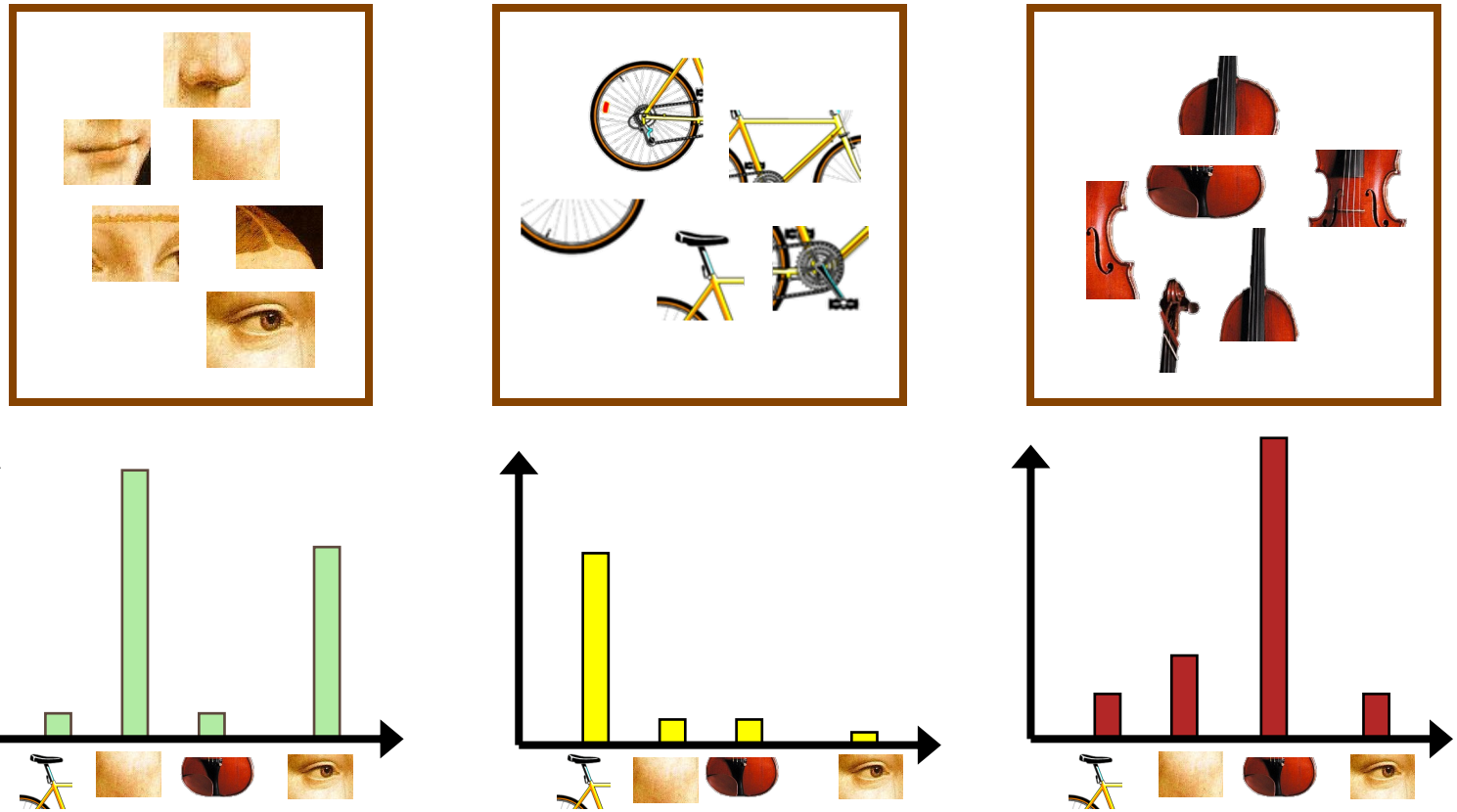




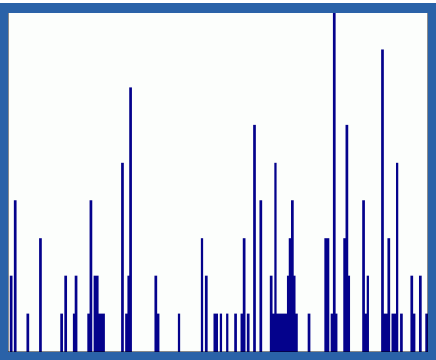

Slide credit: Bastian Leibe

# Bag of features: Outline

1. Extract local features
2. Learn “visual vocabulary”
3. **Quantize local features using visual vocabulary**
4. **Represent images by frequencies of “visual words”**



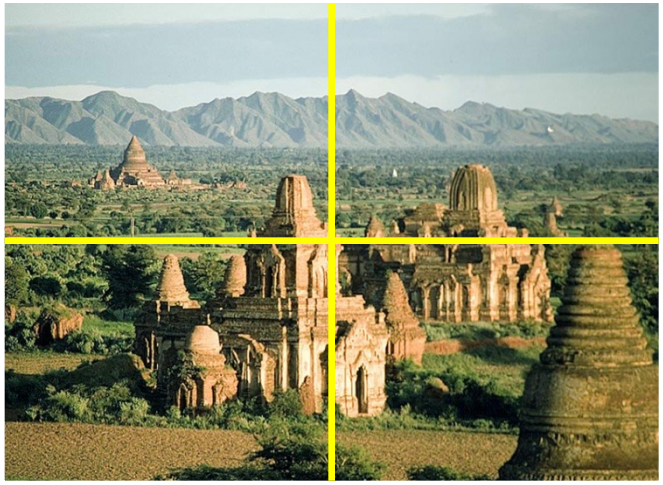
# Spatial pyramids

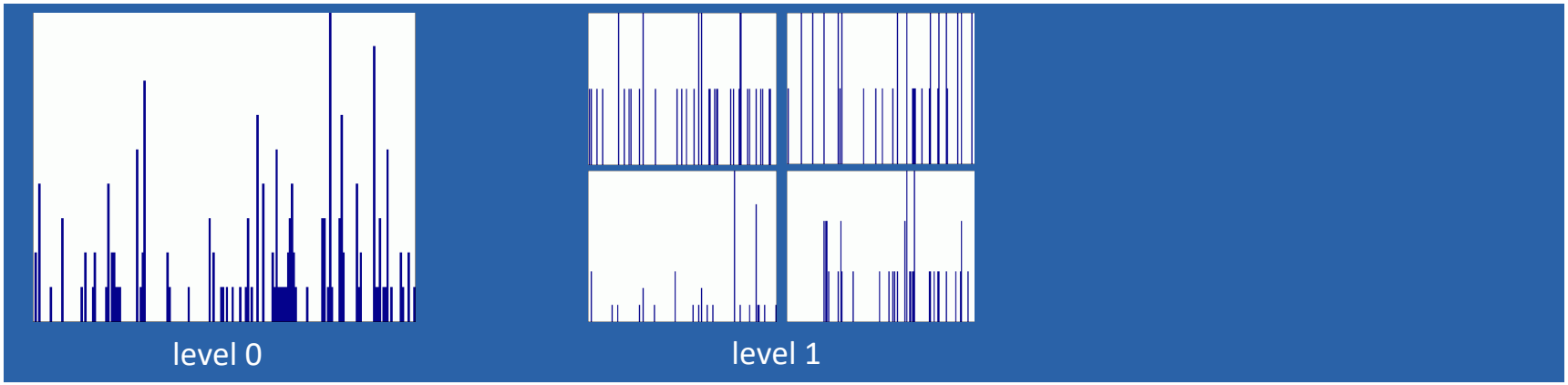


level 0

Lazebnik, Schmid & Ponce (CVPR 2006)

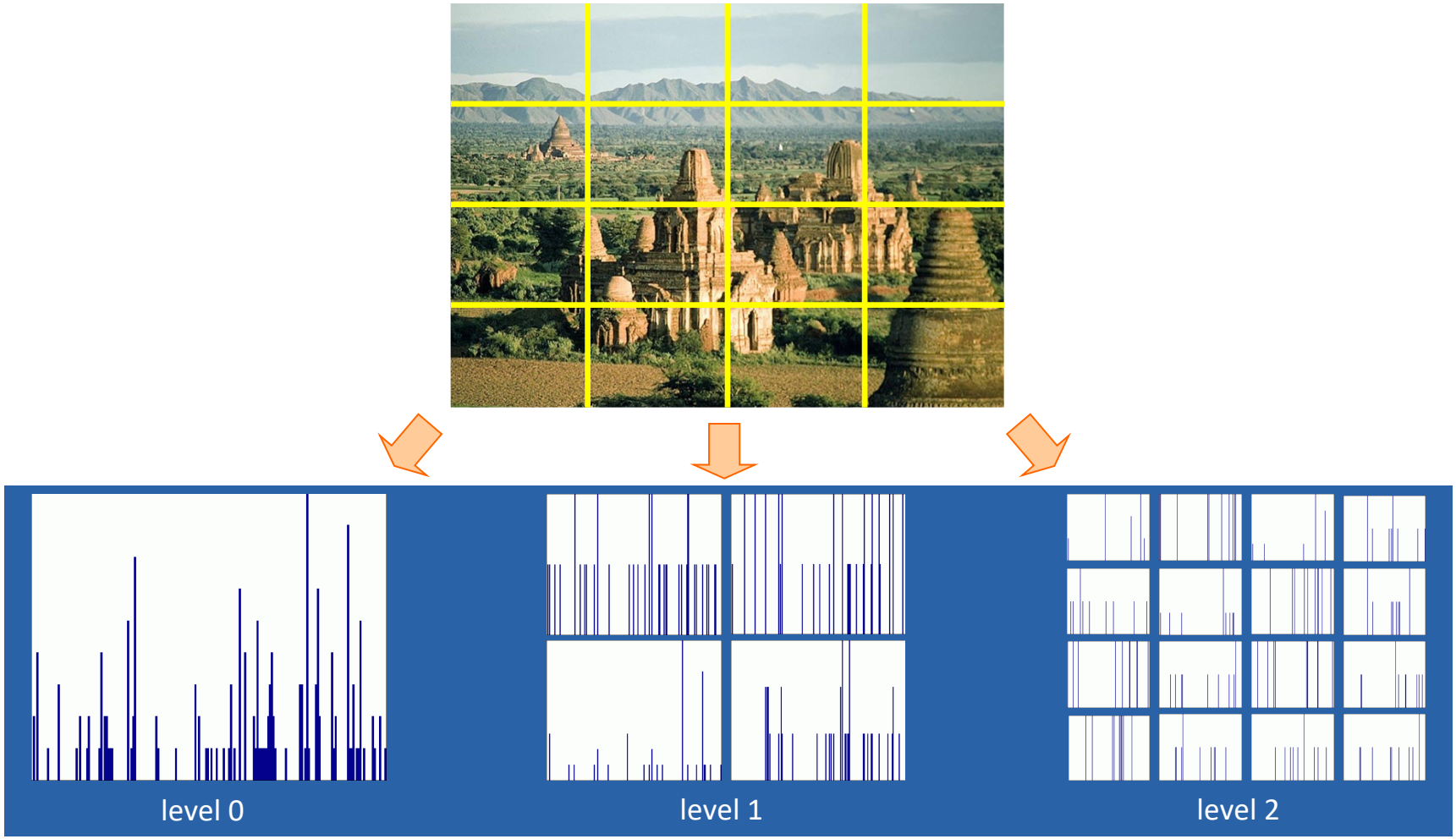
# Spatial pyramids





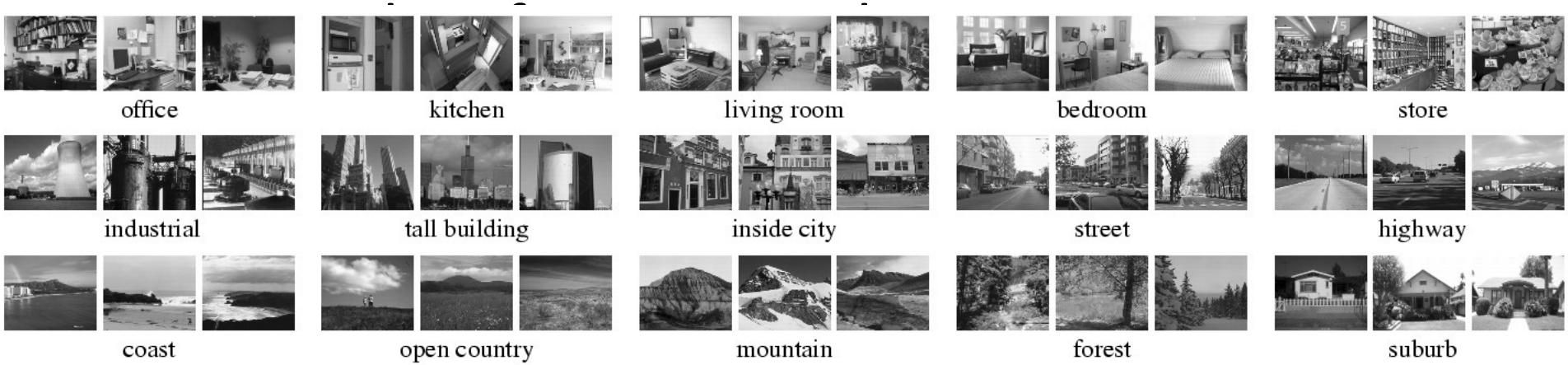

Lazebnik, Schmid & Ponce (CVPR 2006)

# Spatial pyramids




Lazebnik, Schmid & Ponce (CVPR 2006)

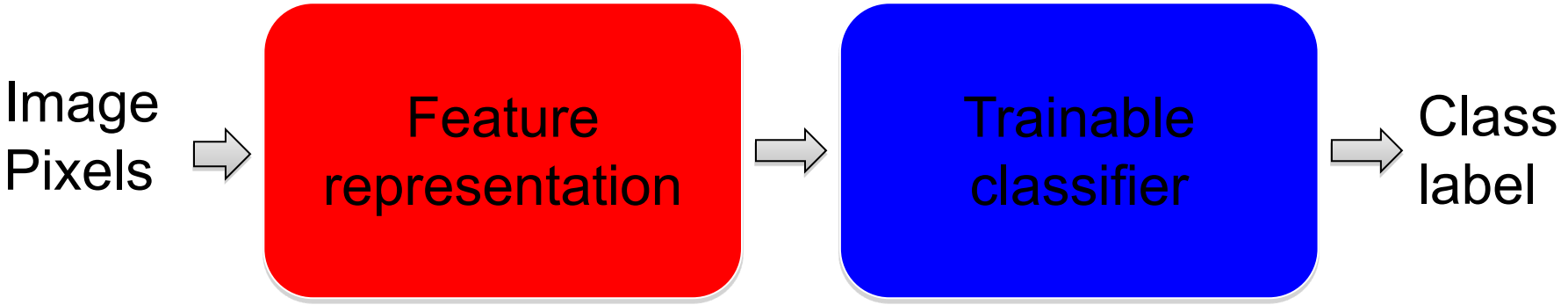
# Spatial pyramids



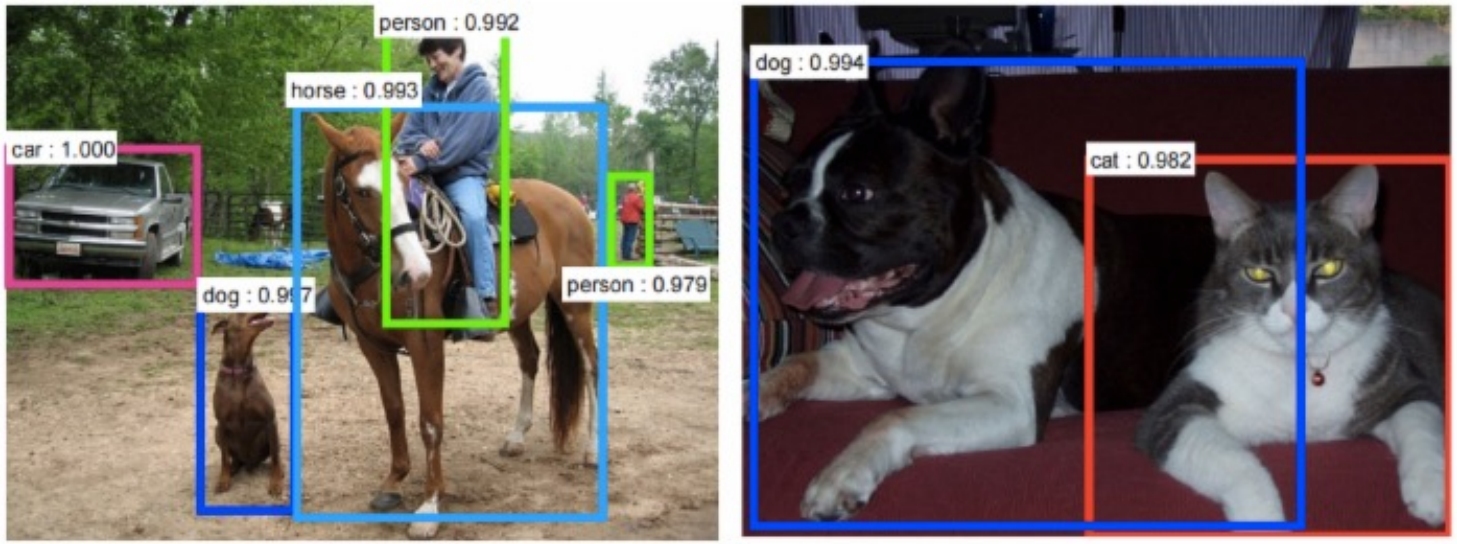

| | Weak features (vocabulary size: 16) | | Strong features (vocabulary size: 200) | |
|---|---|---|---|---|
| Level | Single-level | Pyramid | Single-level | Pyramid |
| 0 ($1 \times 1$) | 45.3 ±0.5 | | 72.2 ±0.6 | |
| 1 ($2 \times 2$) | 53.6 ±0.3 | 56.2 ±0.6 | 77.9 ±0.6 | 79.0 ±0.5 |
| 2 ($4 \times 4$) | 61.7 ±0.6 | 64.7 ±0.7 | 79.4 ±0.3 | **81.1** ±0.3 |
| 3 ($8 \times 8$) | 63.3 ±0.8 | **66.8** ±0.6 | 77.2 ±0.4 | 80.7 ±0.3 |

S. Lazebnik, C. Schmid, J. Ponce: "*Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scenes Categories*", CVPR 2006

# From image classification to object detection

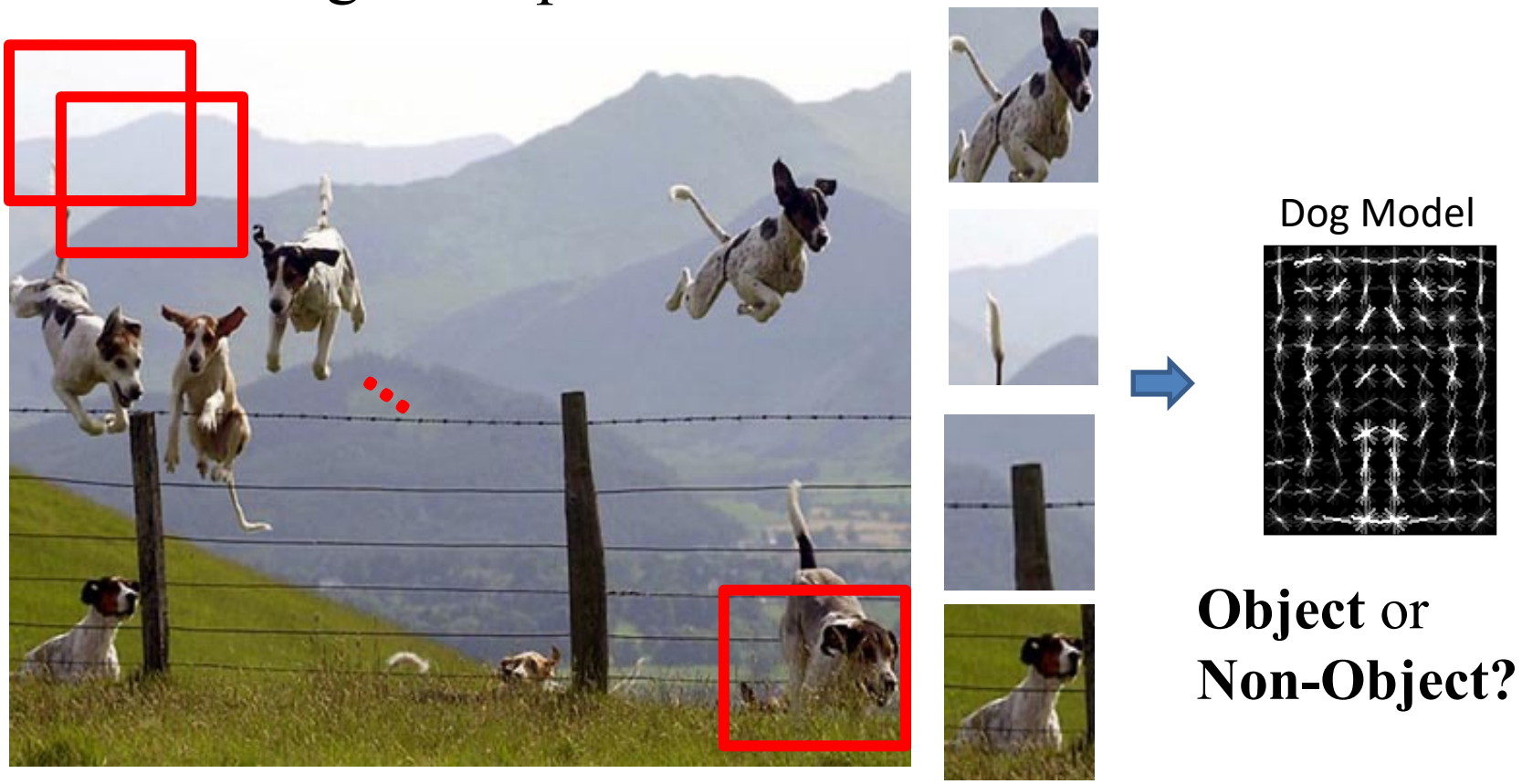



Object detection



# Object Category Detection

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



# Challenges in modeling the object class

Illumination

Object pose

Clutter

Occlusions

Intra-class appearance

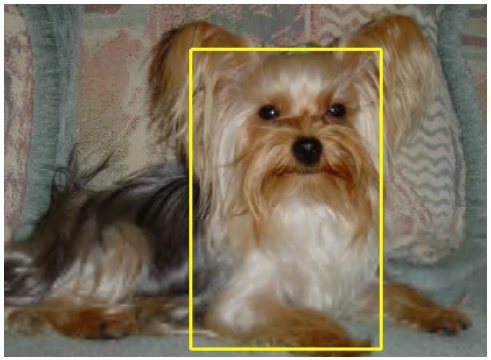
Viewpoint

Slide credit K. Grauman, B. Leibe

# Challenges in modeling the object class

True Detections

Bad Localization



Confused with Similar Object

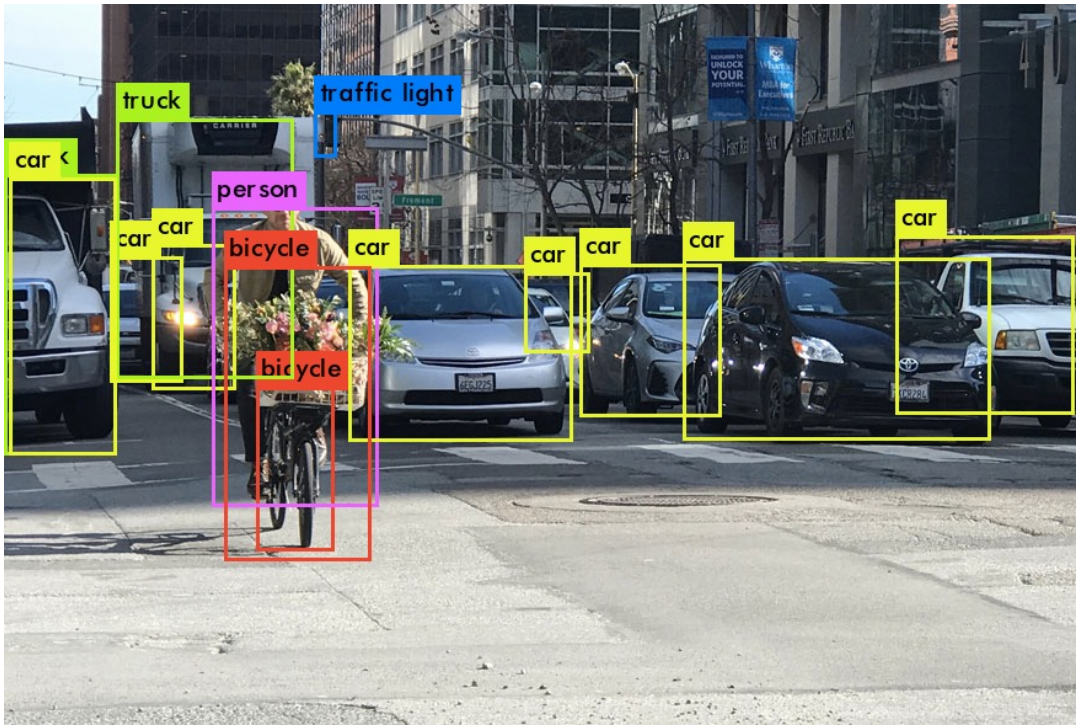
Misc. Background

Confused with Dissimilar Objects

Slide credit K. Grauman, B. Leibe

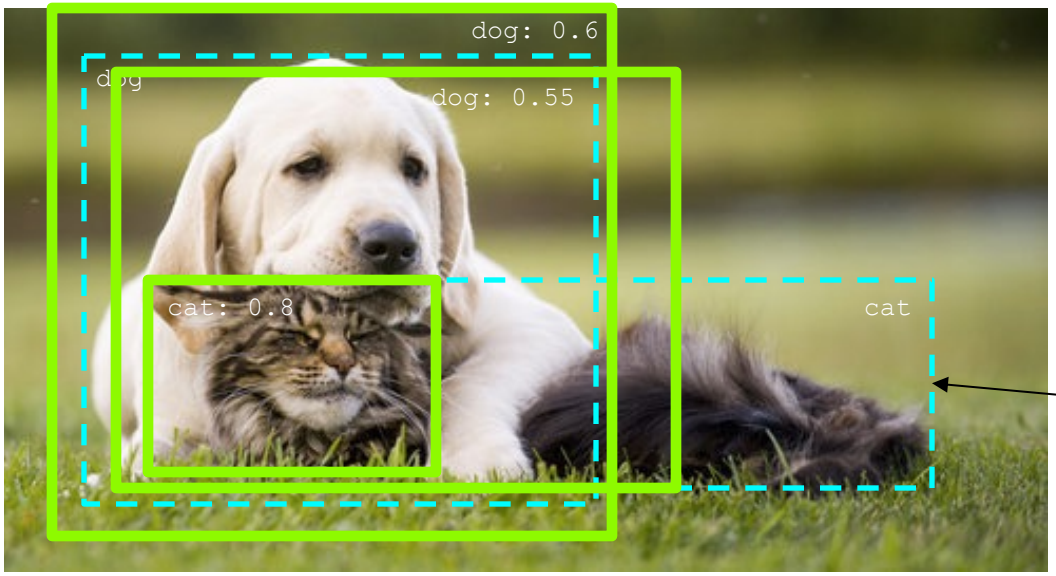
# Other challenges of object detection

- Images may contain more than one class, multiple instances from the same class
- Bounding box localization
- Evaluation



# Object detection evaluation

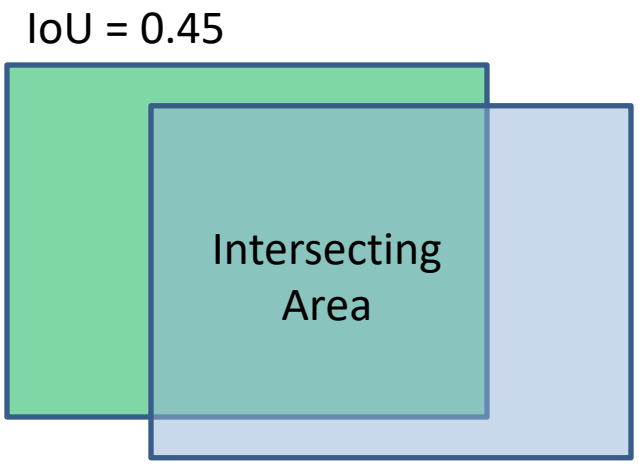
- At test time, predict bounding boxes, class labels, and confidence scores
- For each detection, determine whether it is a true or false positive
  - PASCAL criterion: Area(GT ∩ Det) / Area(GT ∪ Det) > 0.5 (threshold)
  - For multiple detections of the same ground truth box, only one considered a true positive

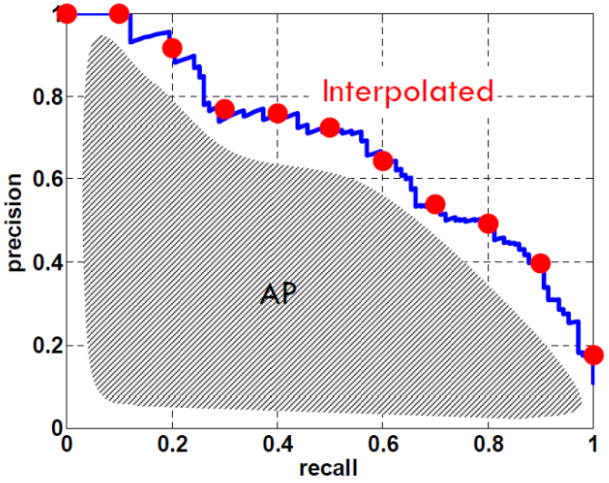


# Object detection evaluation

- Datasets
  - PASCAL VOC (2005-2012): 20 classes, ~20,000 images
  - MS COCO (2014-): 80 classes, ~300,000 images

- Evaluation
  - Output: for each class, predict bounding boxes (xmin, ymin, xmax, ymax) with confidences
  - Metric:
    - True detection: >= 0.5 Intersection over Union (IoU), not a duplicate
    - Precision, Recall
    - AP: area under the interpolated curve



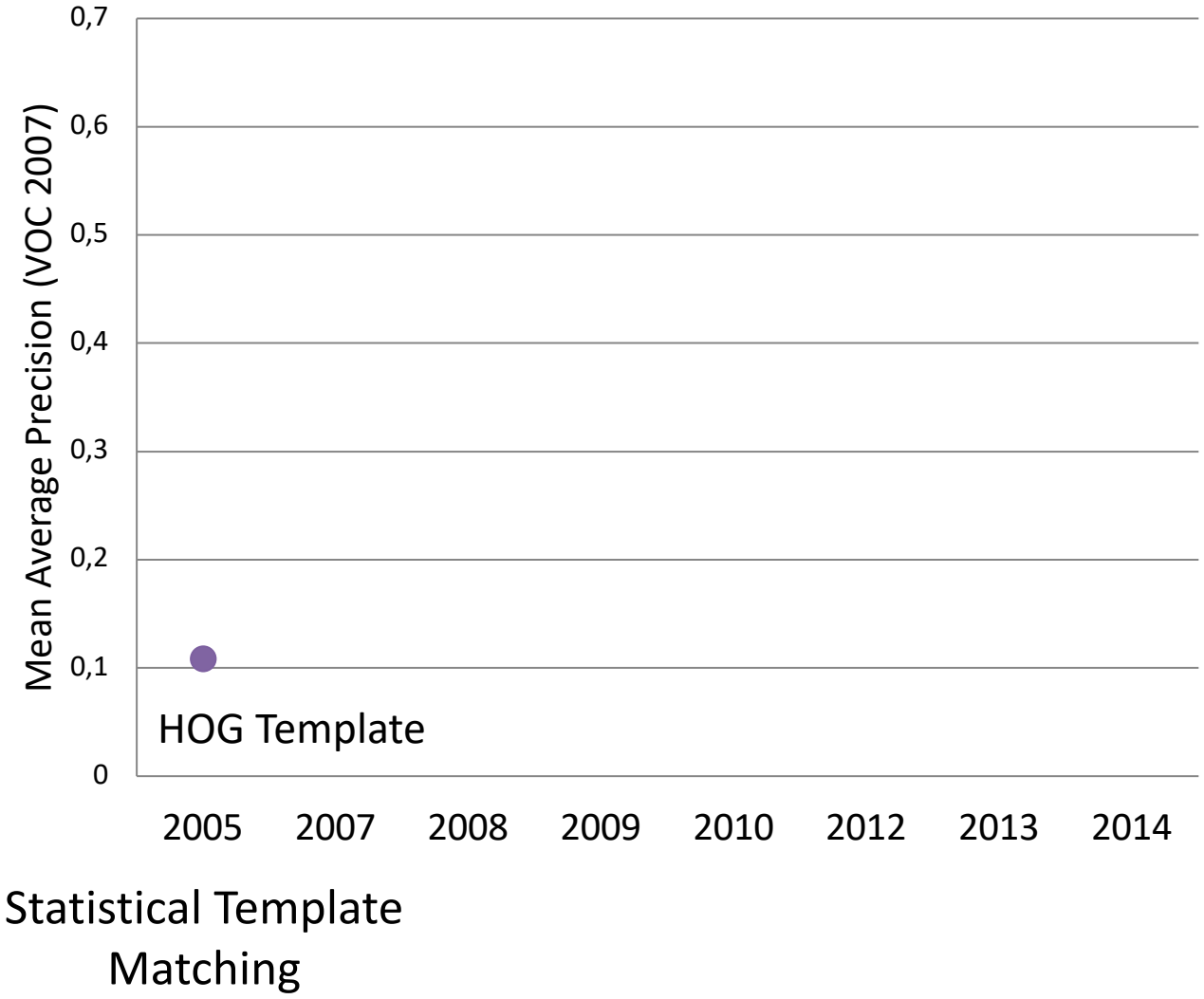
# PASCAL VOC Challenge (2005-2012)

- 20 challenge classes:
  - *Person*
  - *Animals:* bird, cat, cow, dog, horse, sheep
  - *Vehicles:* aeroplane, bicycle, boat, bus, car, motorbike, train
  - *Indoor:* bottle, chair, dining table, potted plant, sofa, tv/monitor

- Dataset size (by 2012): 11.5K training/validation images, 27K bounding boxes, 7K segmentations

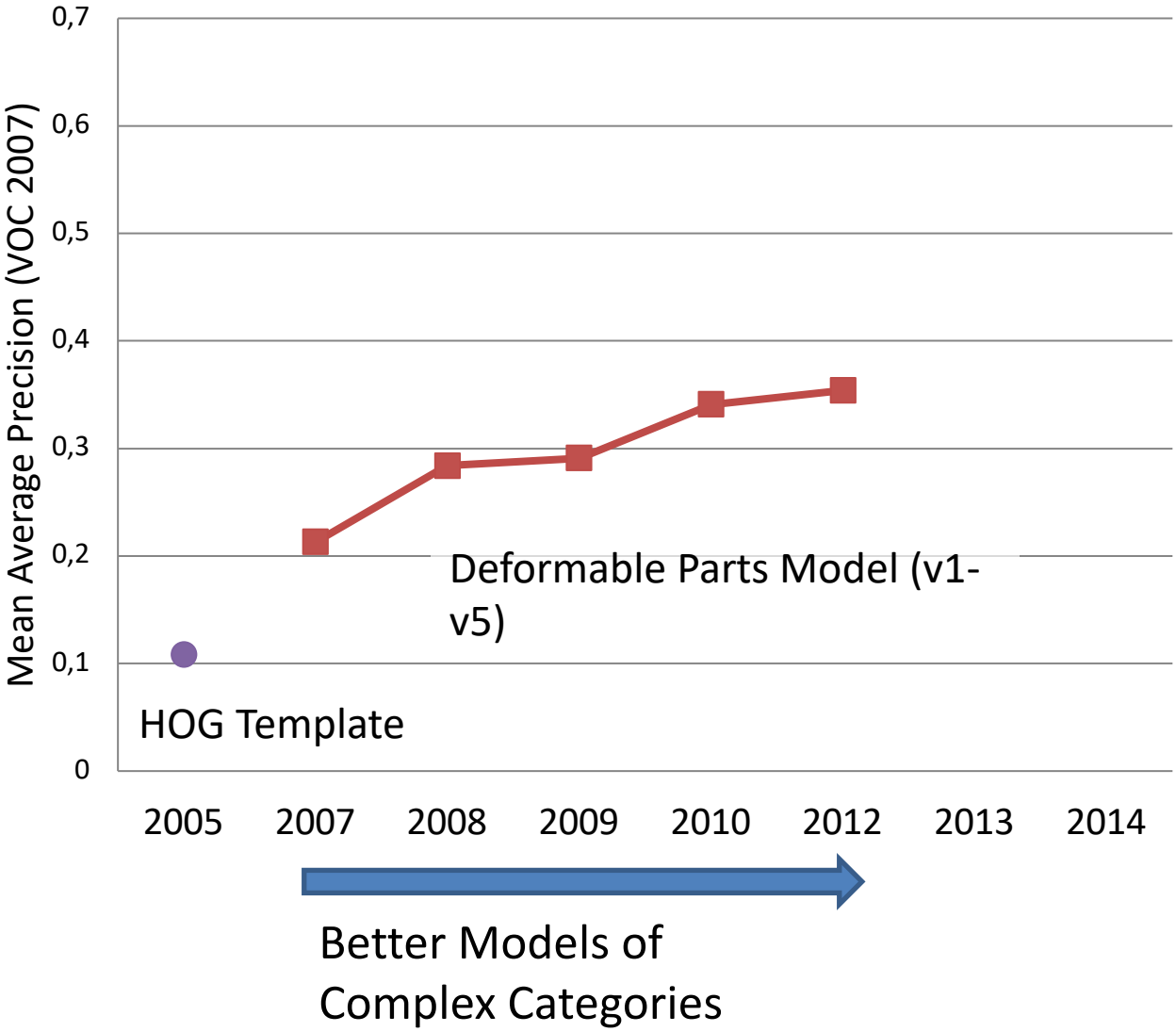
http://host.robots.ox.ac.uk/pascal/VOC/

# Improvements in object detection



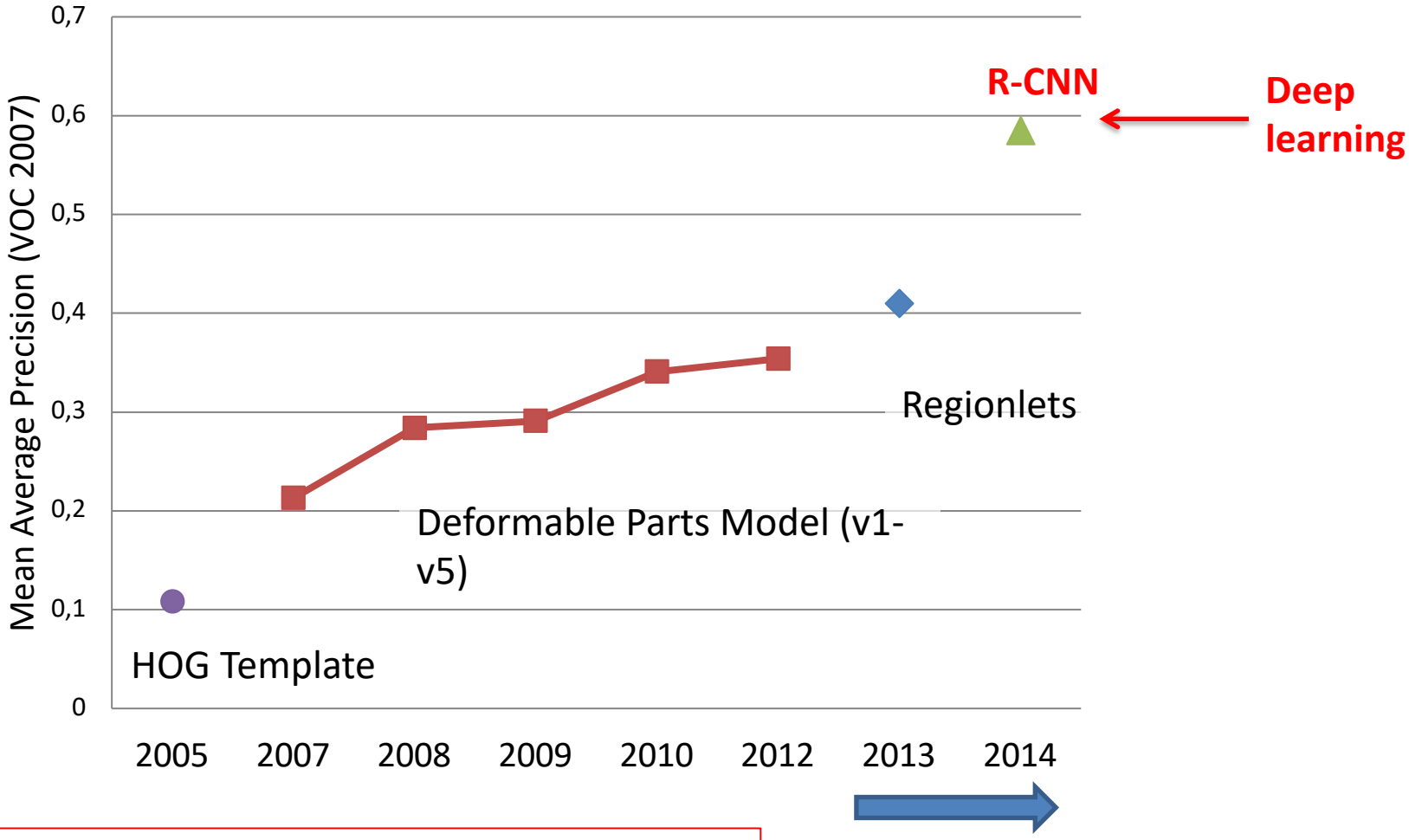

HOG: Dalal-Triggs 2005

# Improvements in object detection




HOG: Dalal-Triggs 2005 DPM: Felzenszwalb et al. 2008-2012

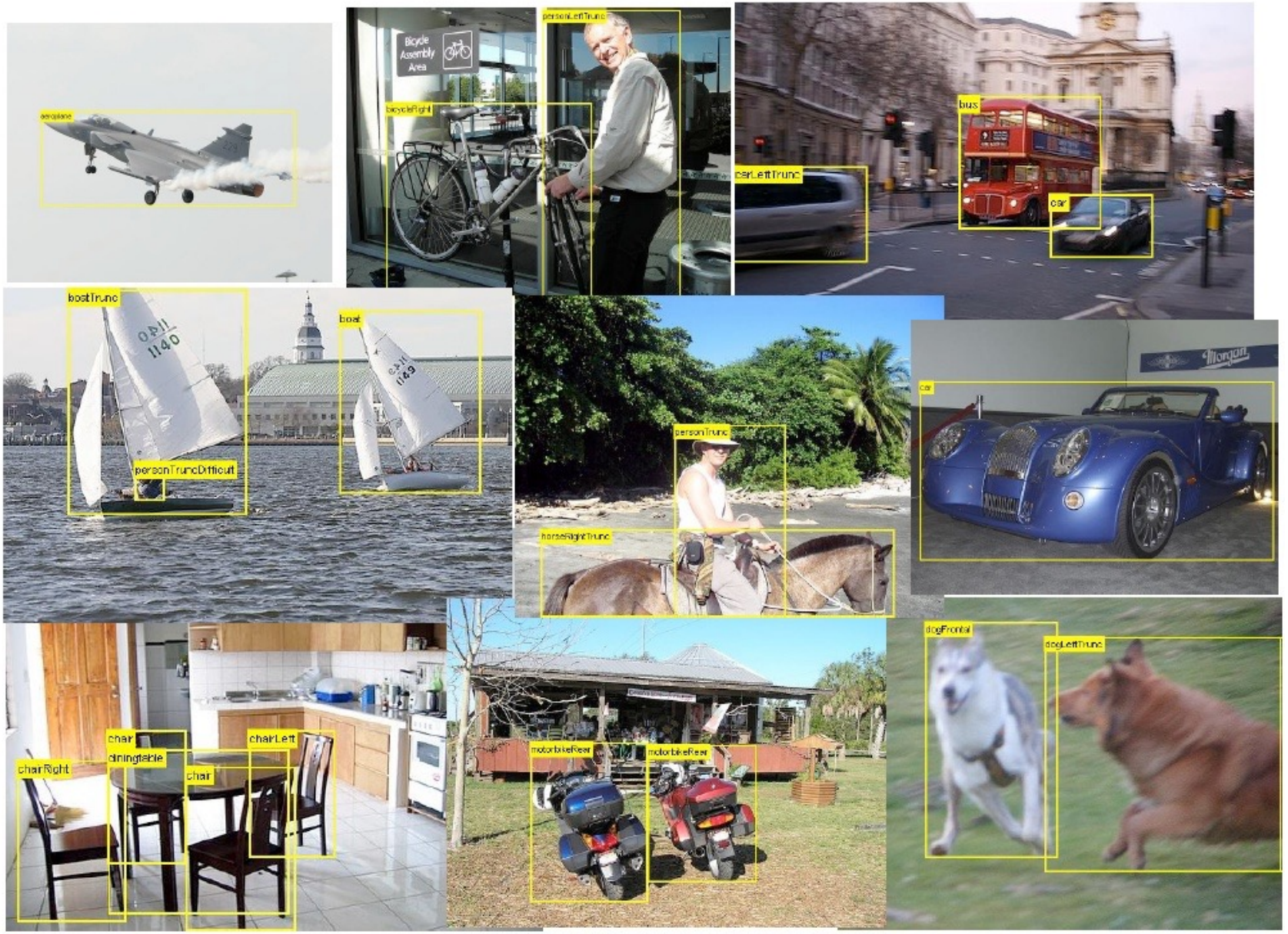
# Improvements in object detection



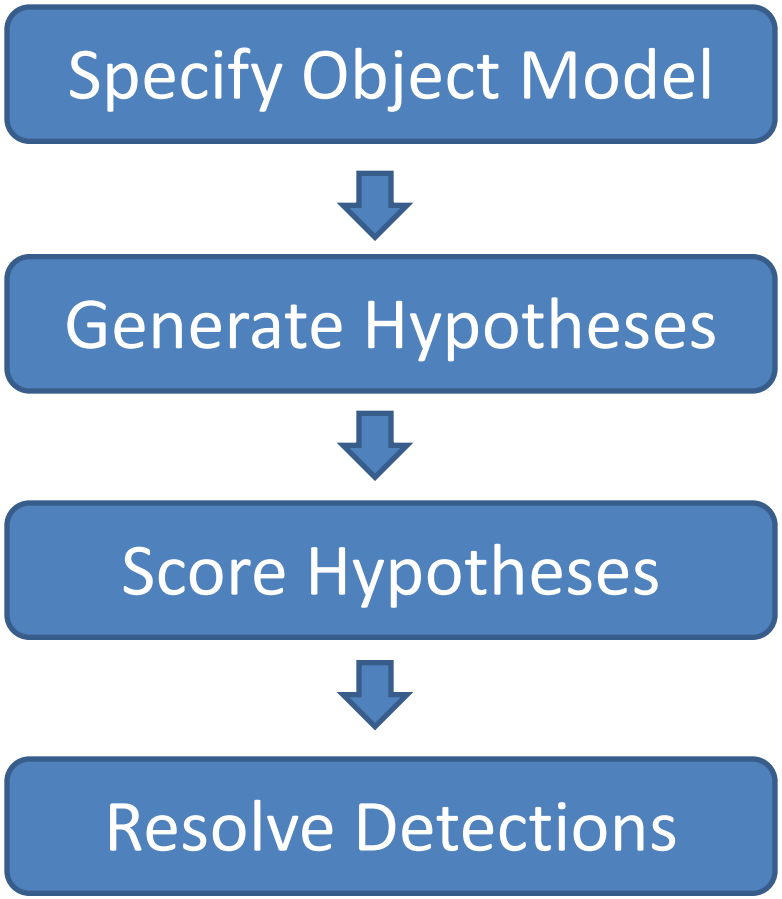

Key Advance: Learn effective features from massive amounts of labeled data *and* adapt to new tasks with less data

HOG: Dalal-Triggs 2005 DPM: Felzenszwalb et al. 2008-2012 Regionlets: Wang et al. 2013 R-CNN: Girshick et al. 2014

# Detection before deep learning



# General Process of Object Detection

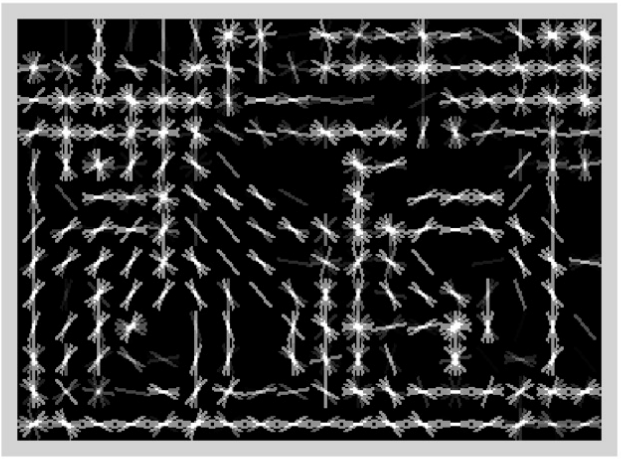



What are the object parameters?

# 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

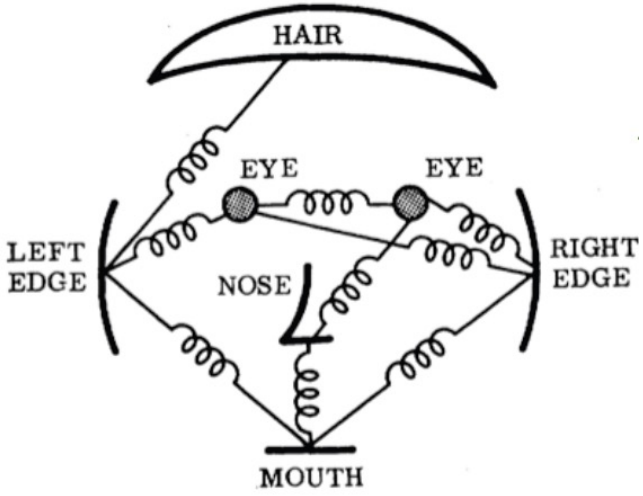


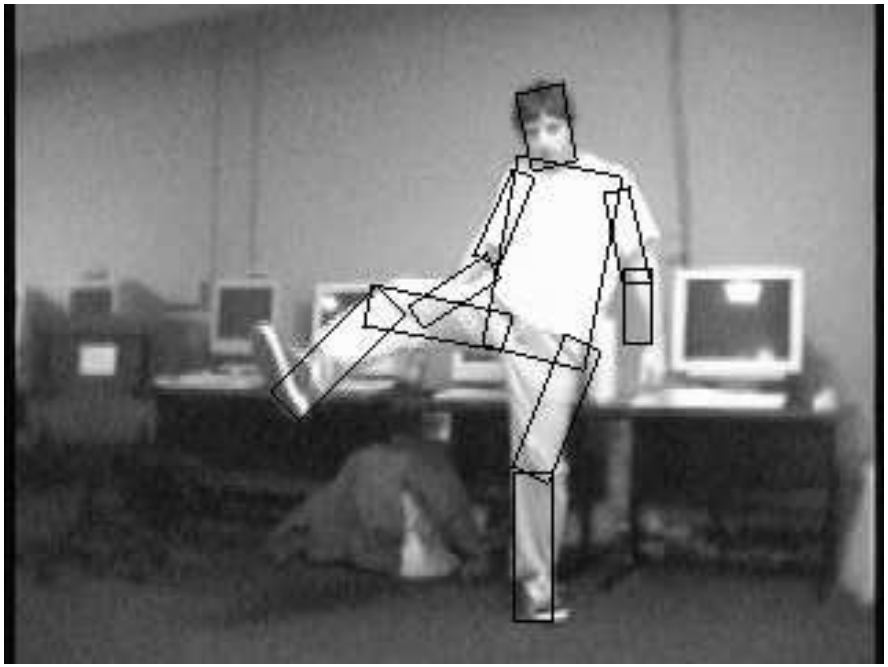
Template Visualization

Images from Felzenszwalb

# Specifying an object model

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



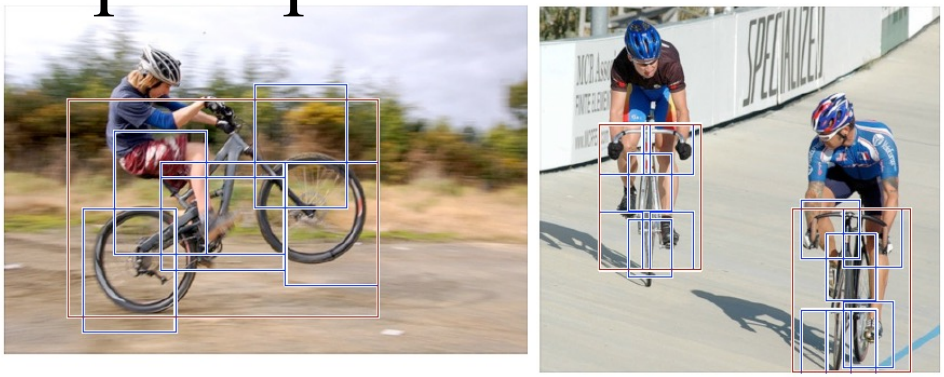



Images from Felzenszwalb

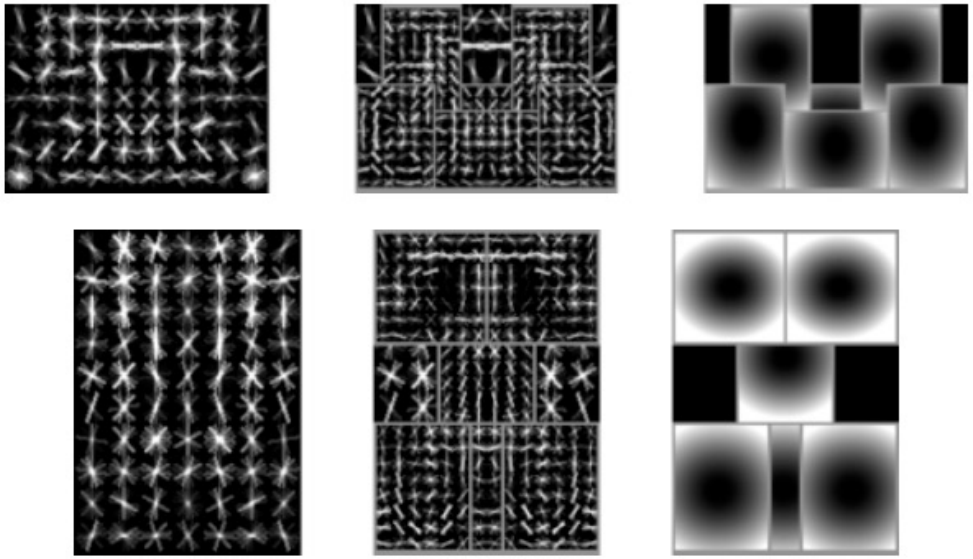
# Specifying an object model

## 3. Hybrid template/parts model

Detections



Template Visualization



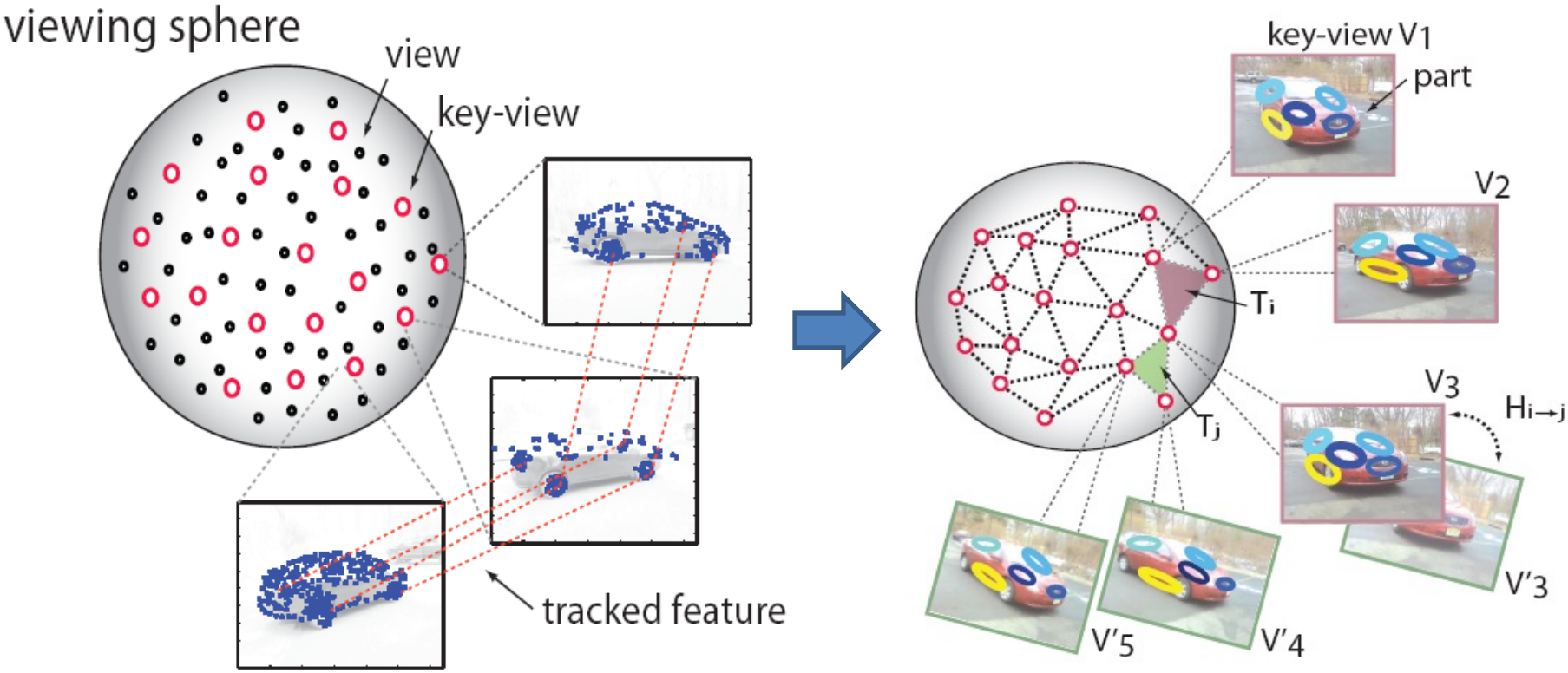
root filters coarse resolution | part filters finer resolution | deformation models

Felzenszwalb et al. 2008

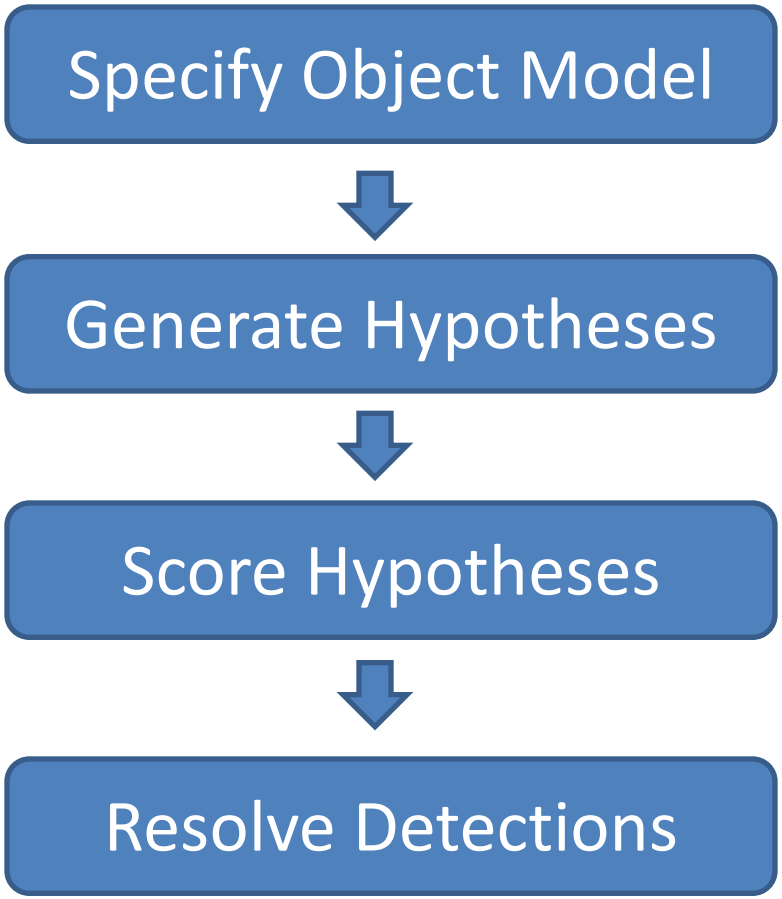
# Specifying an object model

4. 3D-ish model

- Object is collection of 3D planar patches under affine transformation



# General Process of Object Detection




Propose an alignment of the model to the image

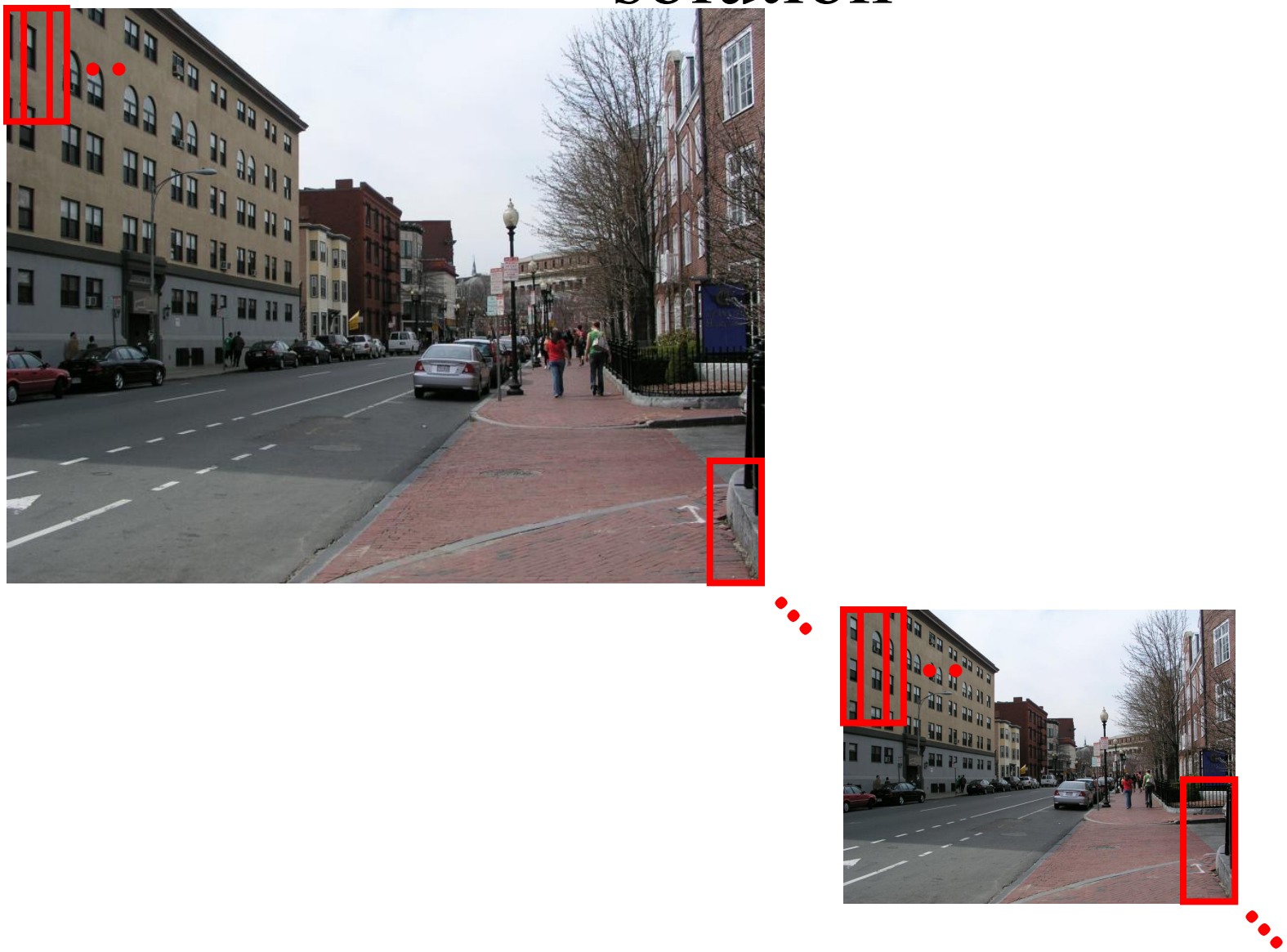
# Generating hypotheses

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

# Generating hypotheses

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

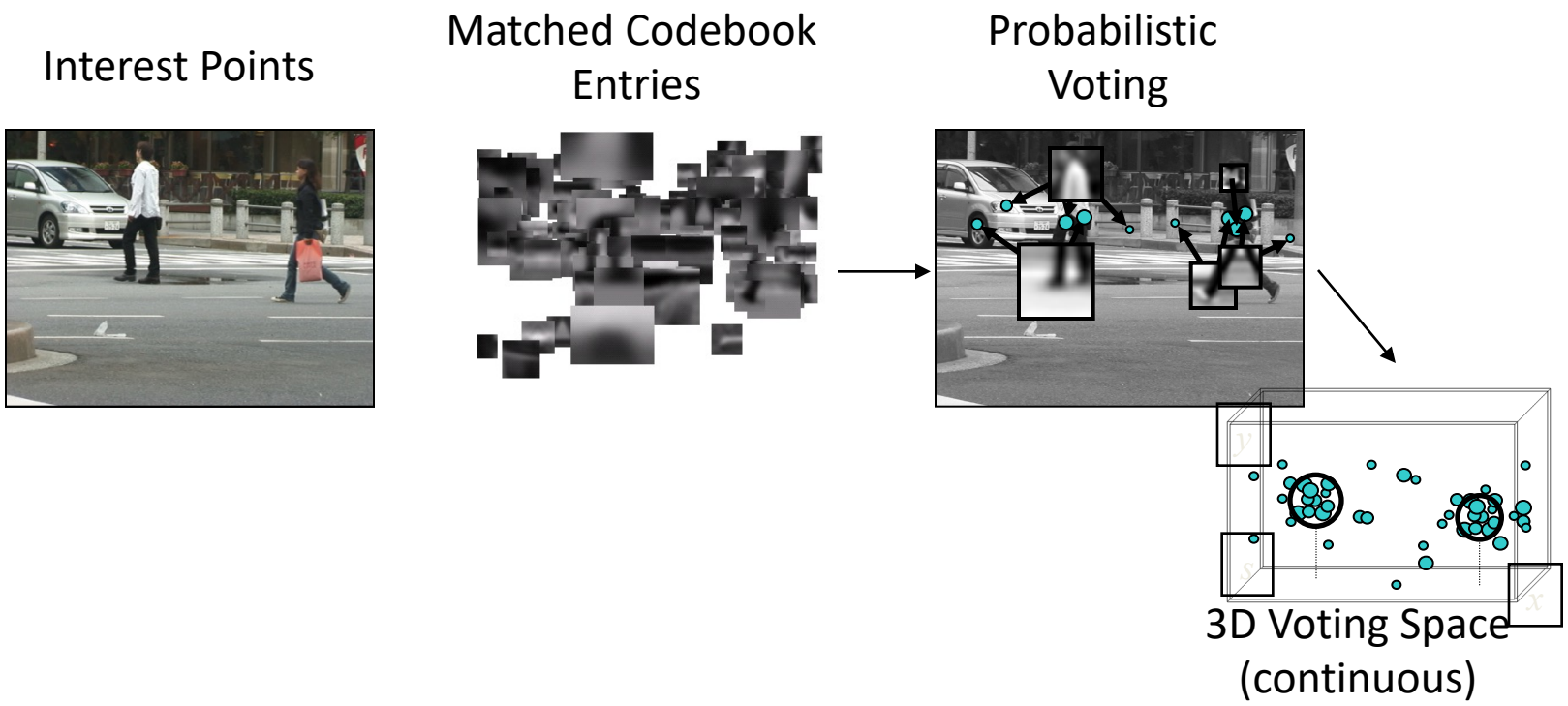
# Sliding window: a simple alignment solution



# Each window is separately classified

# Generating hypotheses

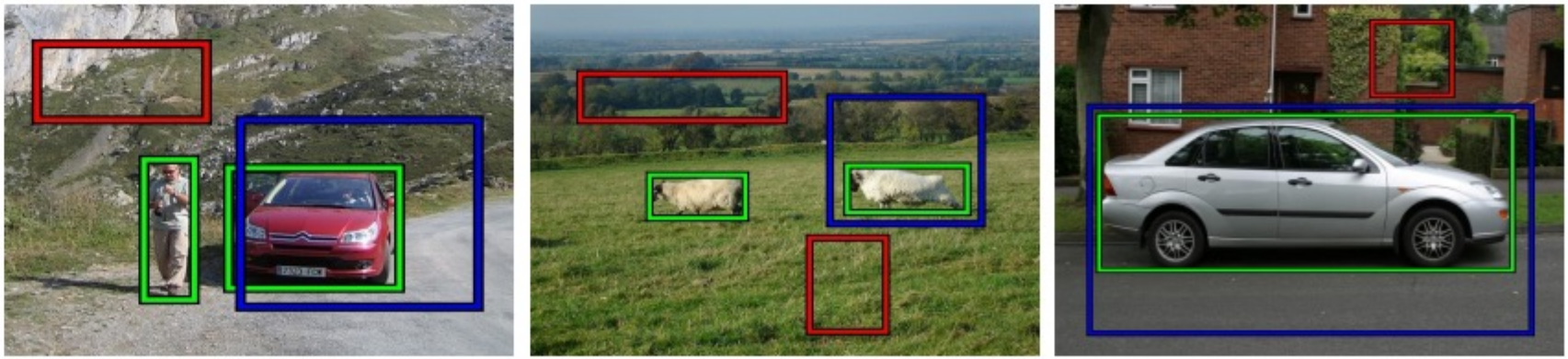
2. Voting from patches/keypoints




ISM model by Leibe et al.

# Generating hypotheses

3. Region-based proposal
   - Learn to generate category-independent regions/boxes that have object-like properties.
   - Let object detector search over “proposals”, not exhaustive sliding windows



Alexe et al. *What is an object?*, CVPR 2010
Alexe et al. *Measuring the objectness of image windows*, PAMI 2012

# Generating hypotheses


Alexe et al. *What is an object?*, CVPR 2010
Alexe et al. *Measuring the objectness of image windows*, PAMI 2012

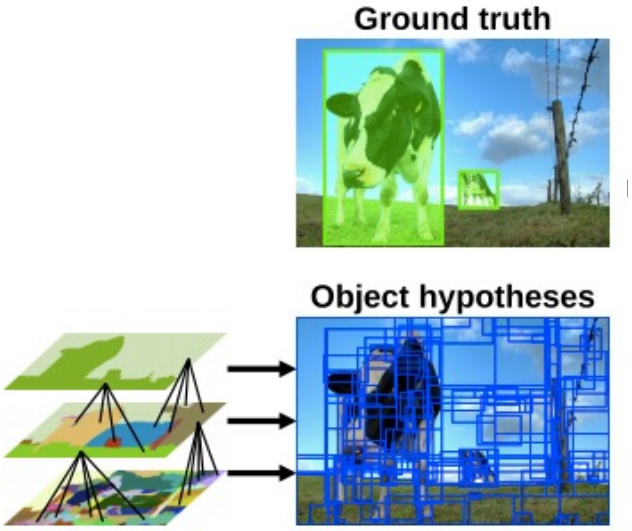
# Generating hypotheses

3. Region-based proposal

Endres Hoiem 2010

# Generating hypotheses

## 3. Region-based proposal

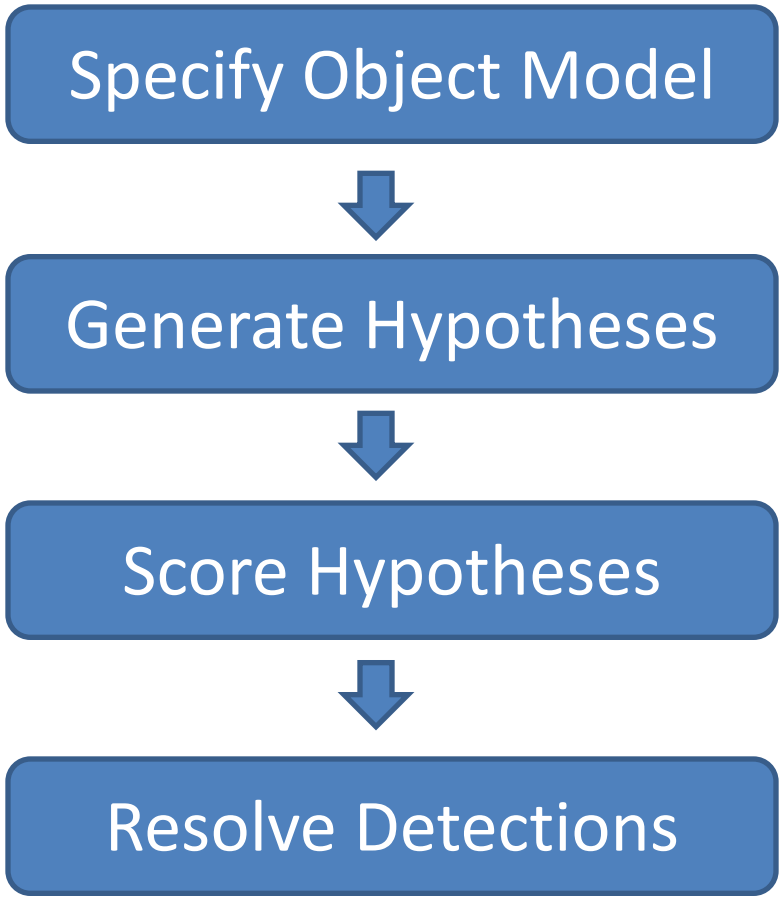




Use hierarchical segmentation: start with small *superpixels* and merge based on diverse cues

Used by R-CNN, Fast R-CNN.

J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, Selective Search for Object Recognition, IJCV 2013

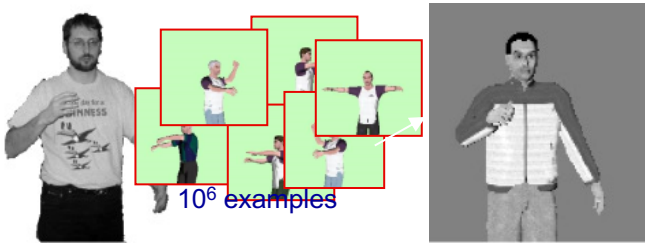
# General Process of Object Detection



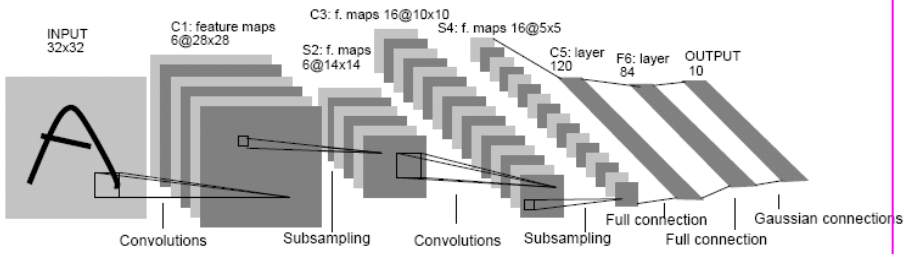

Different classifiers

# Discriminative classifier construction

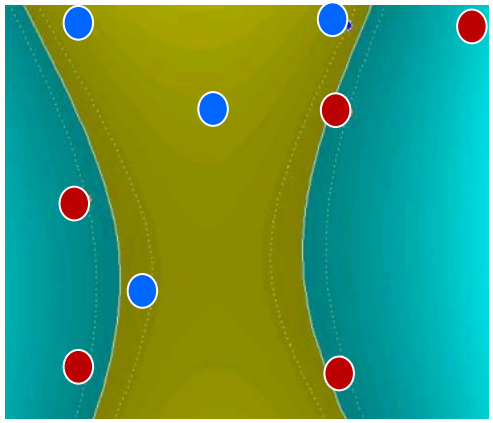
## Nearest neighbor



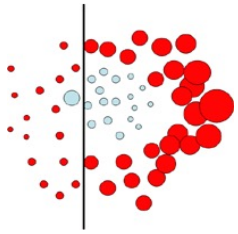

## Neural networks



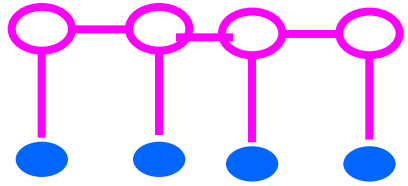

## Support Vector Machines



## Boosting

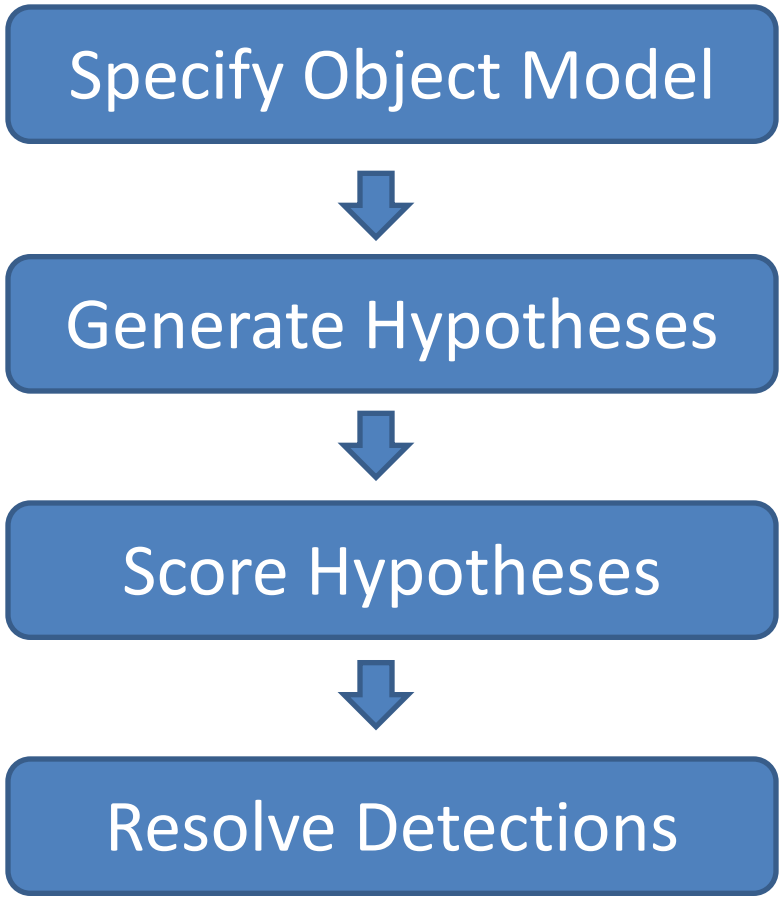


## Conditional Random Fields



Slide credit Antonio Torralba

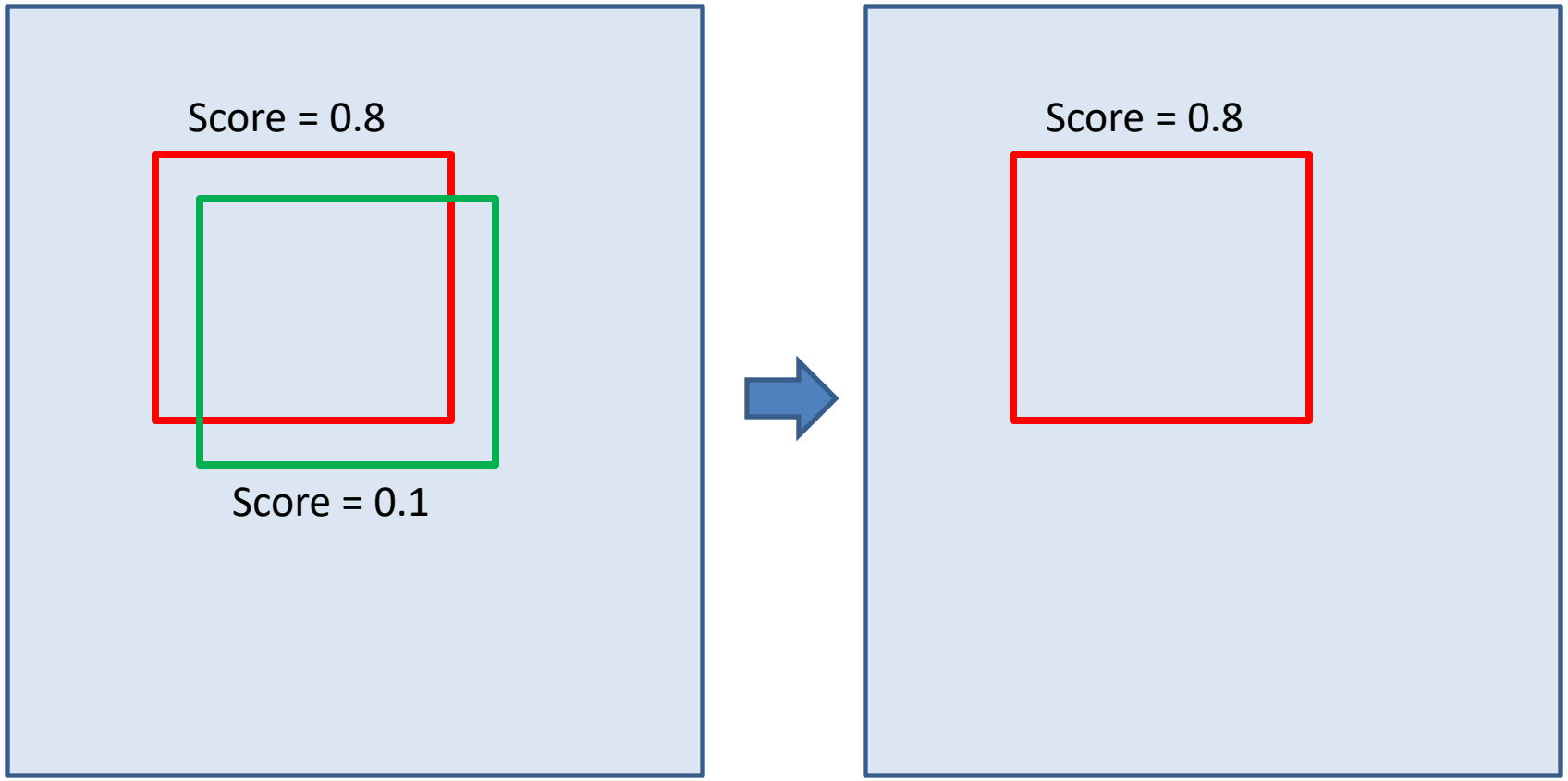
# General Process of Object Detection




Optionally, rescore each proposed object based on whole set

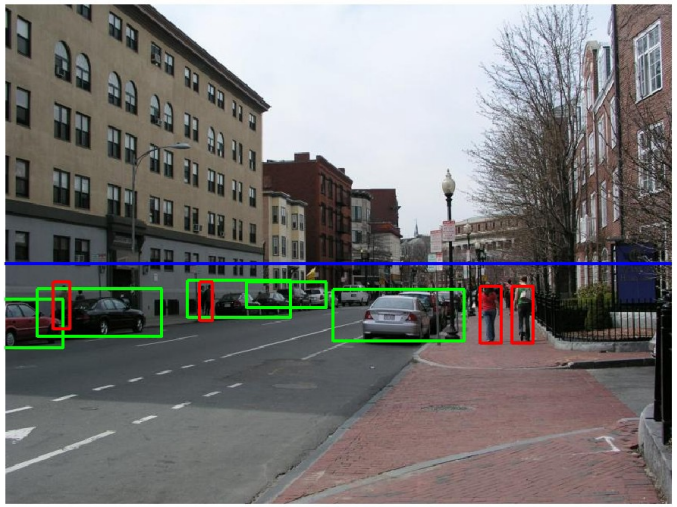
# Resolving detection scores

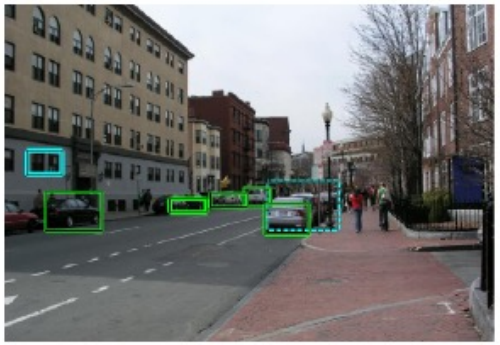
1. Non-max suppression



# Resolving detection scores

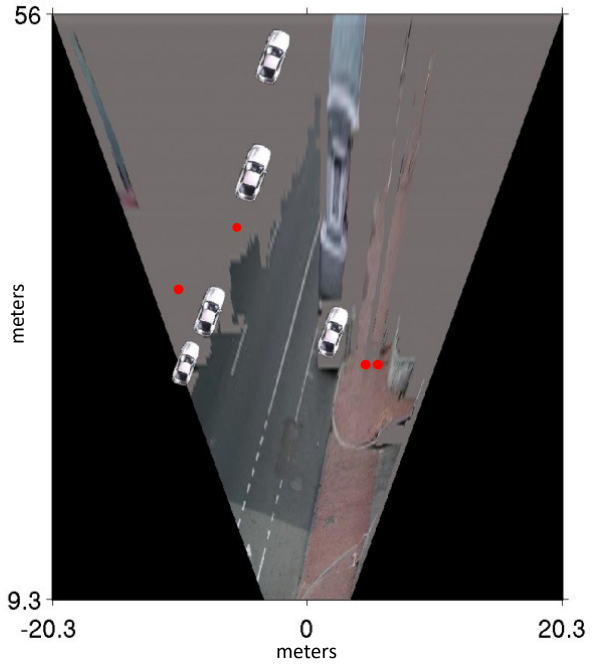
2. Context/reasoning





(g) Car Detections: Local

(h) Ped Detections: Local




Hoiem et al. 2006

# Design challenges

- How to efficiently search for likely objects
  - Sliding windows require searching hundreds of thousands of positions and scales
- Feature design and scoring
  - How should appearance be modeled? What features correspond to the object?
- How to deal with different viewpoints?
  - Often train different models for a few different viewpoints
- Implementation details
  - Window size
  - Aspect ratio
  - Translation/scale step size
  - Non-maxima suppression

# Histograms of oriented gradients (HOG)

- Partition image into blocks and compute histogram of gradient orientations in each block

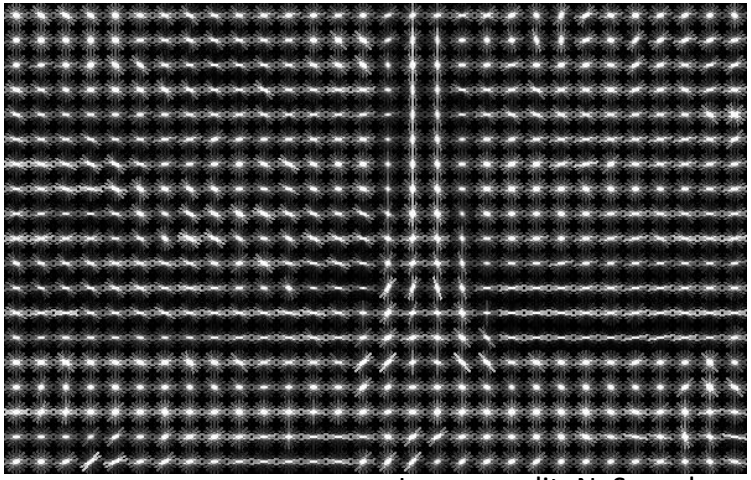


Image credit: N. Snavely

N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

# Pedestrian detection with HOG

- Train a pedestrian template using a linear support vector machine

positive training examples

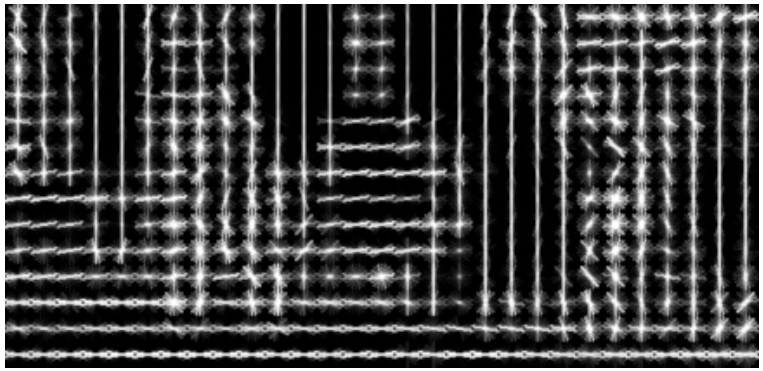
negative training examples

N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

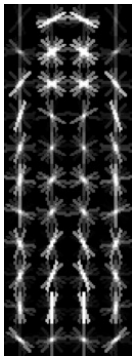
# Pedestrian detection with HOG

- Train a pedestrian template using a linear support vector machine
- At test time, convolve feature map with template
- Find local maxima of response
- For multi-scale detection, repeat over multiple levels of a HOG *pyramid*

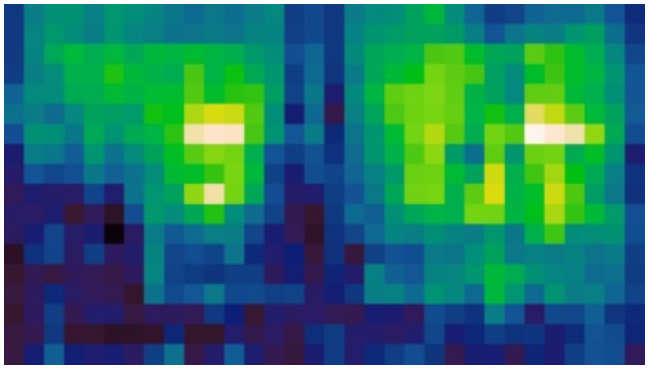
HOG feature map



Template

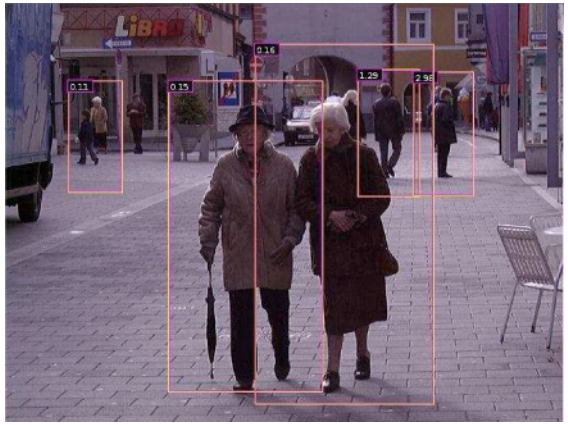


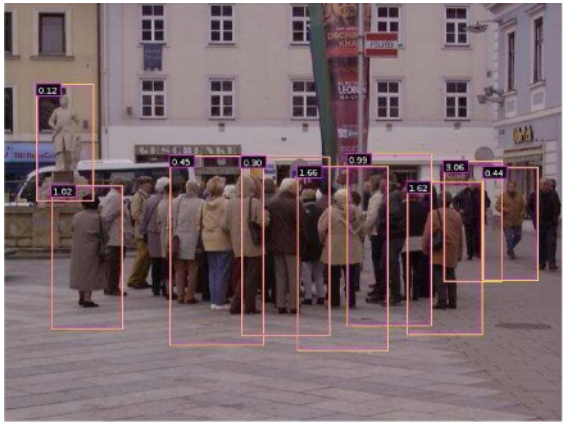
Detector response map

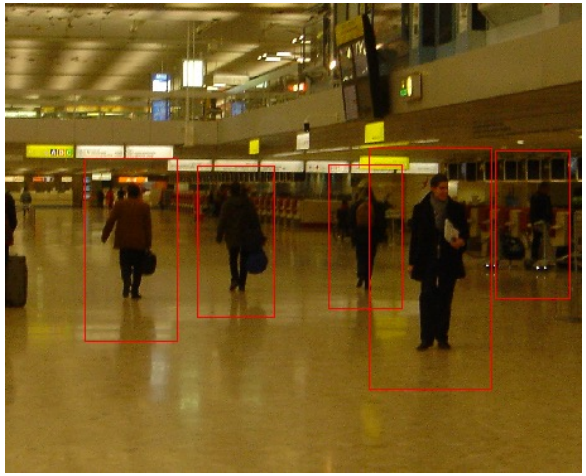


N. Dalal and B. Triggs, Histograms of Oriented Gradients for Human Detection, CVPR 2005

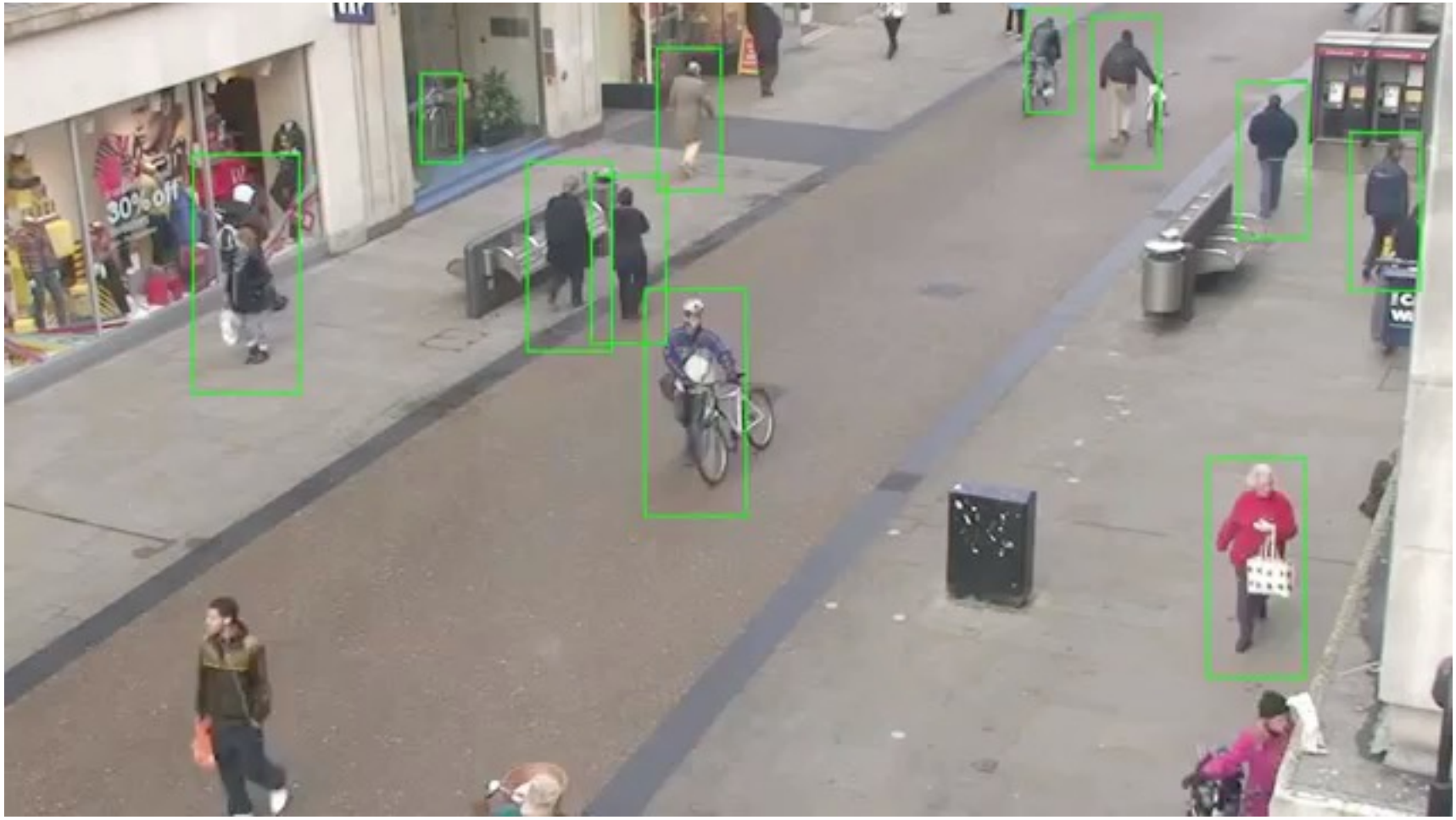
# Example detections








# Example detections in video



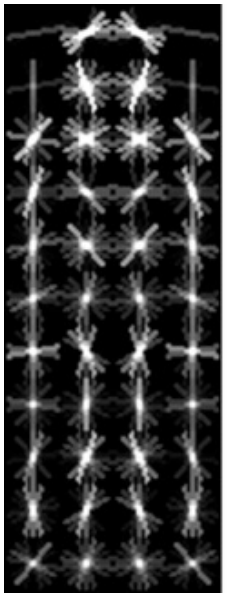
# Discriminative part-based models

- 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

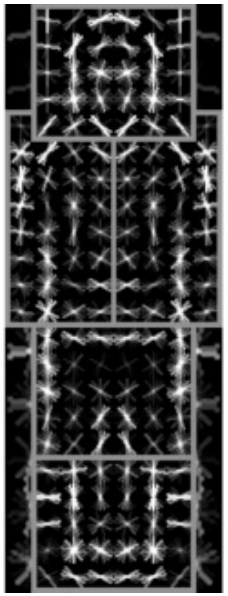
  - Many object categories look very different from different viewpoints, or from instance to instance


# Discriminative part-based models

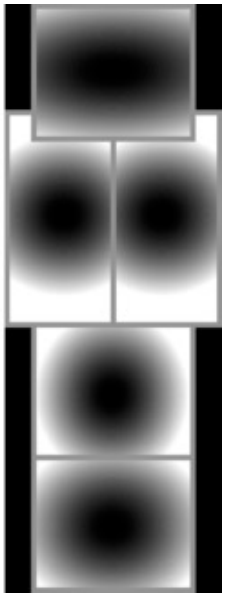
Root filter

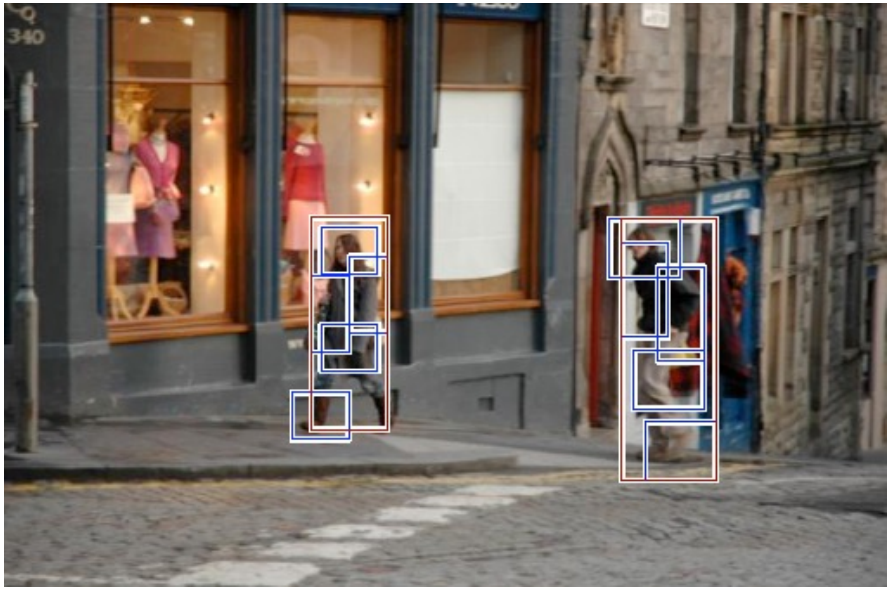


Part filters



Deformation weights

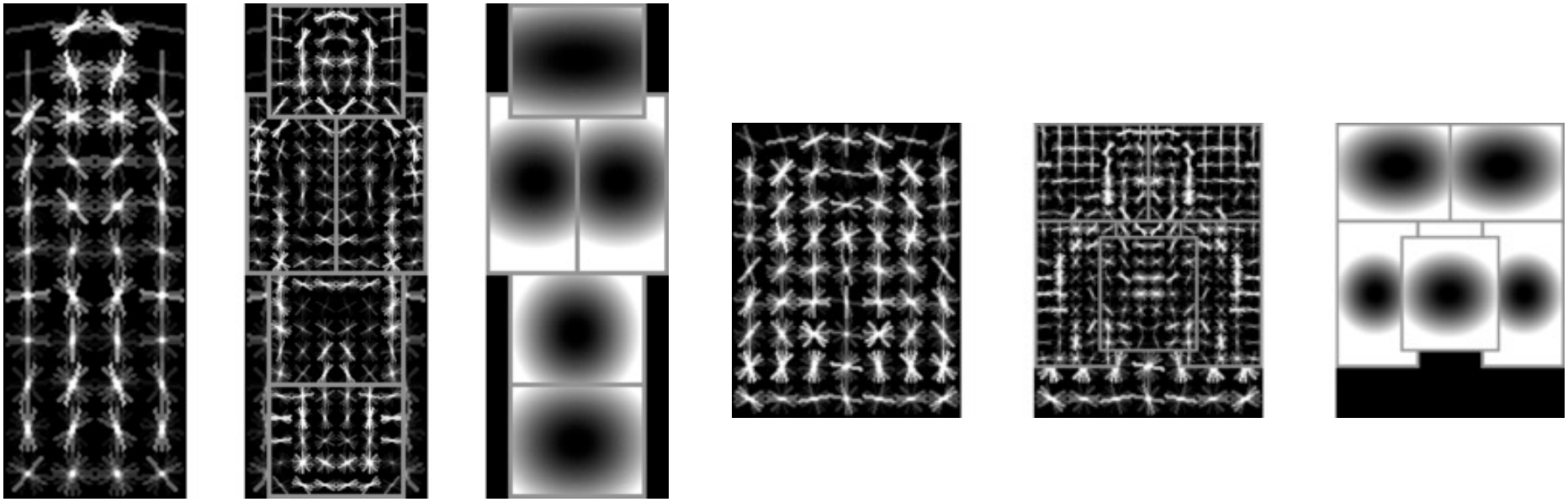




P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part Based Models, PAMI 32(9), 2010

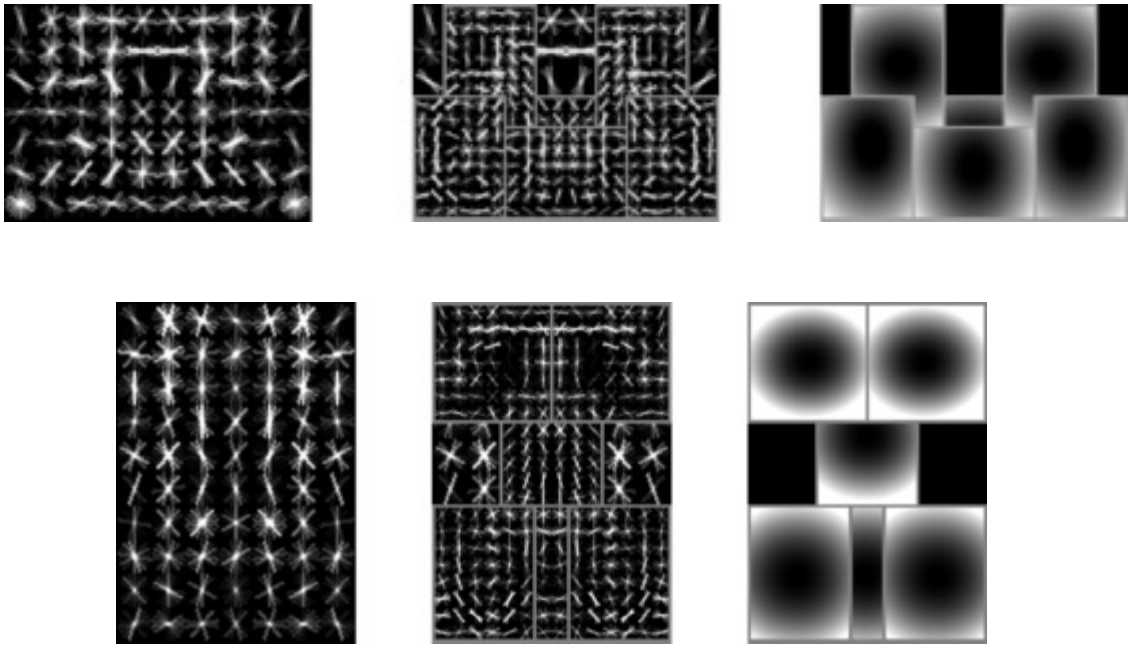
# Discriminative part-based models

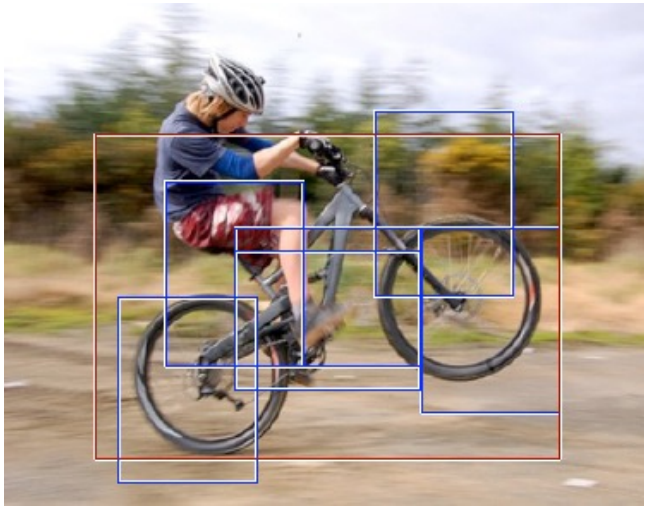
## Multiple components



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

# Discriminative part-based models



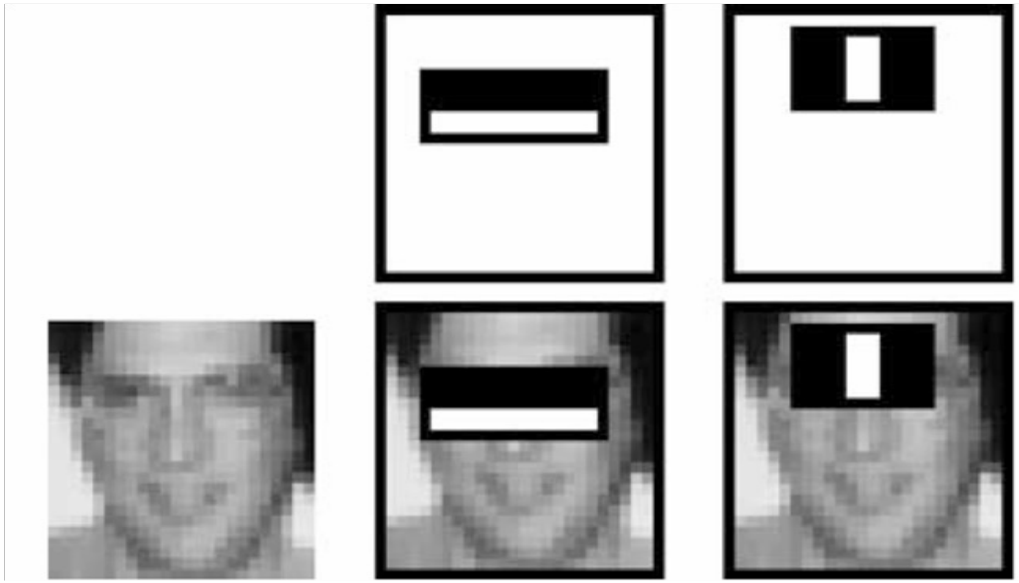


P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part Based Models, PAMI 32(9), 2010

# Viola-Jones sliding window detector

**Fast** detection through two mechanisms

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

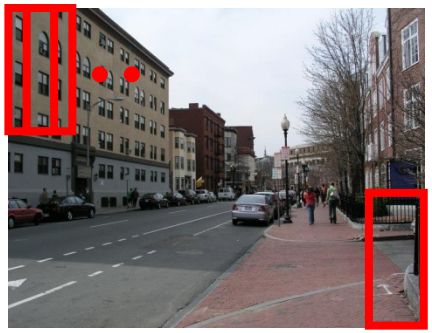


- Haar features
- Integral images
- AdaBoost

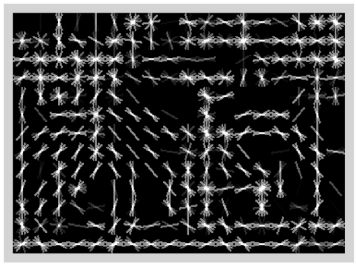
Viola and Jones. Rapid Object Detection using a Boosted Cascade of Simple Features (2001).

# Summary: statistical templates

Propose Window → Extract Features → Classify → Post-process



Sliding window: scan image pyramid



HOG

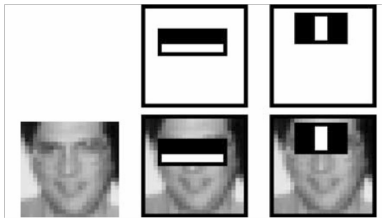
SVM

Boosted stubs

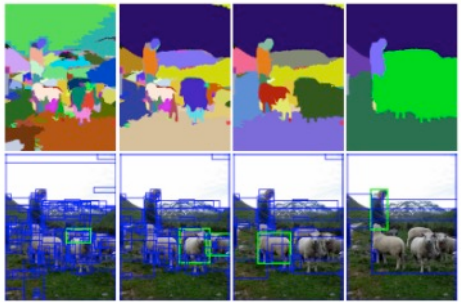
Neural network

Non-max suppression

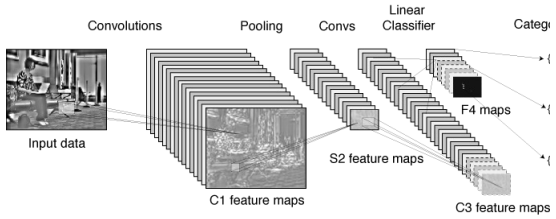
Segment or refine localization



Fast randomized features

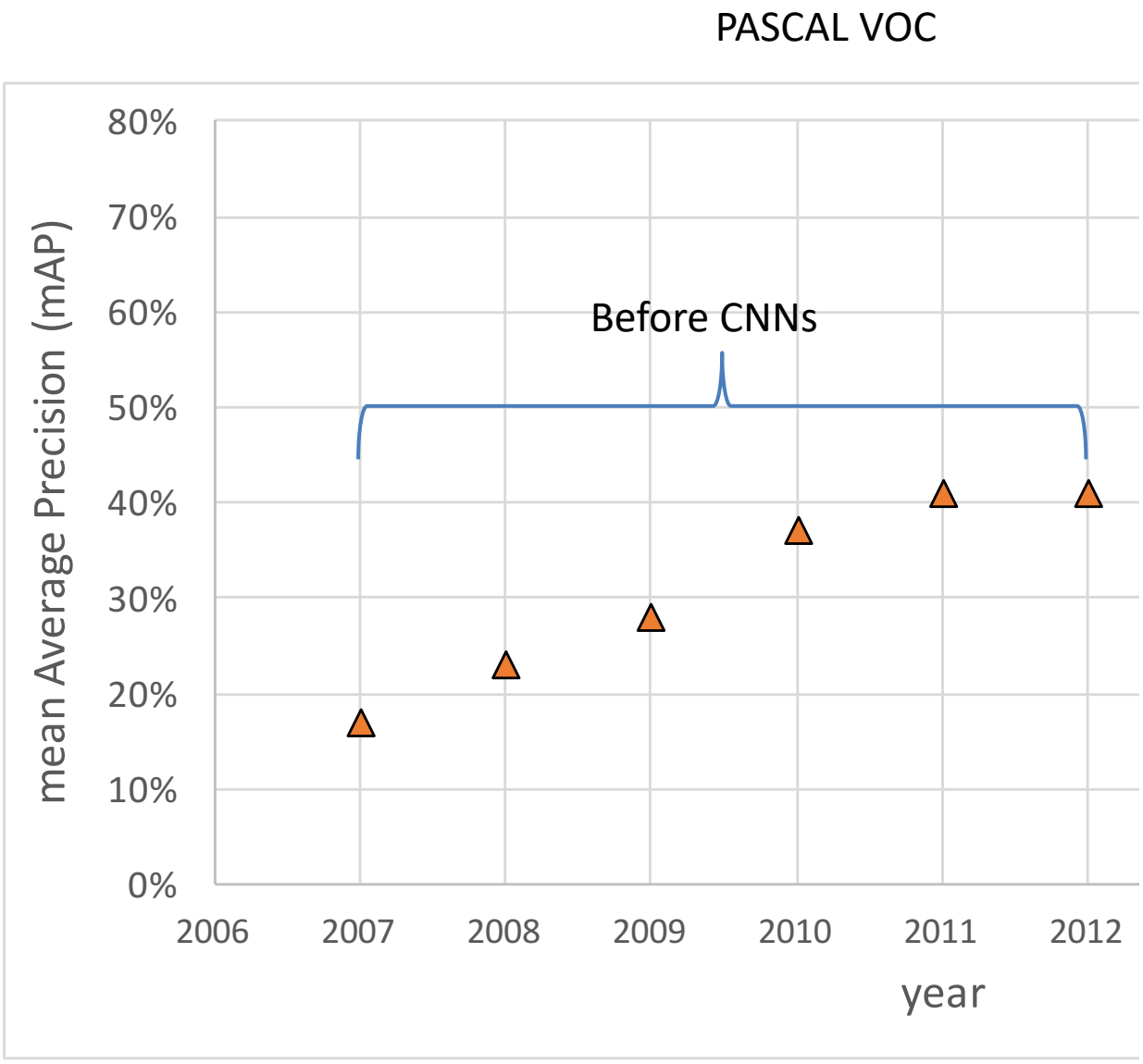


Region proposals: edge/region-based, resize to fixed window

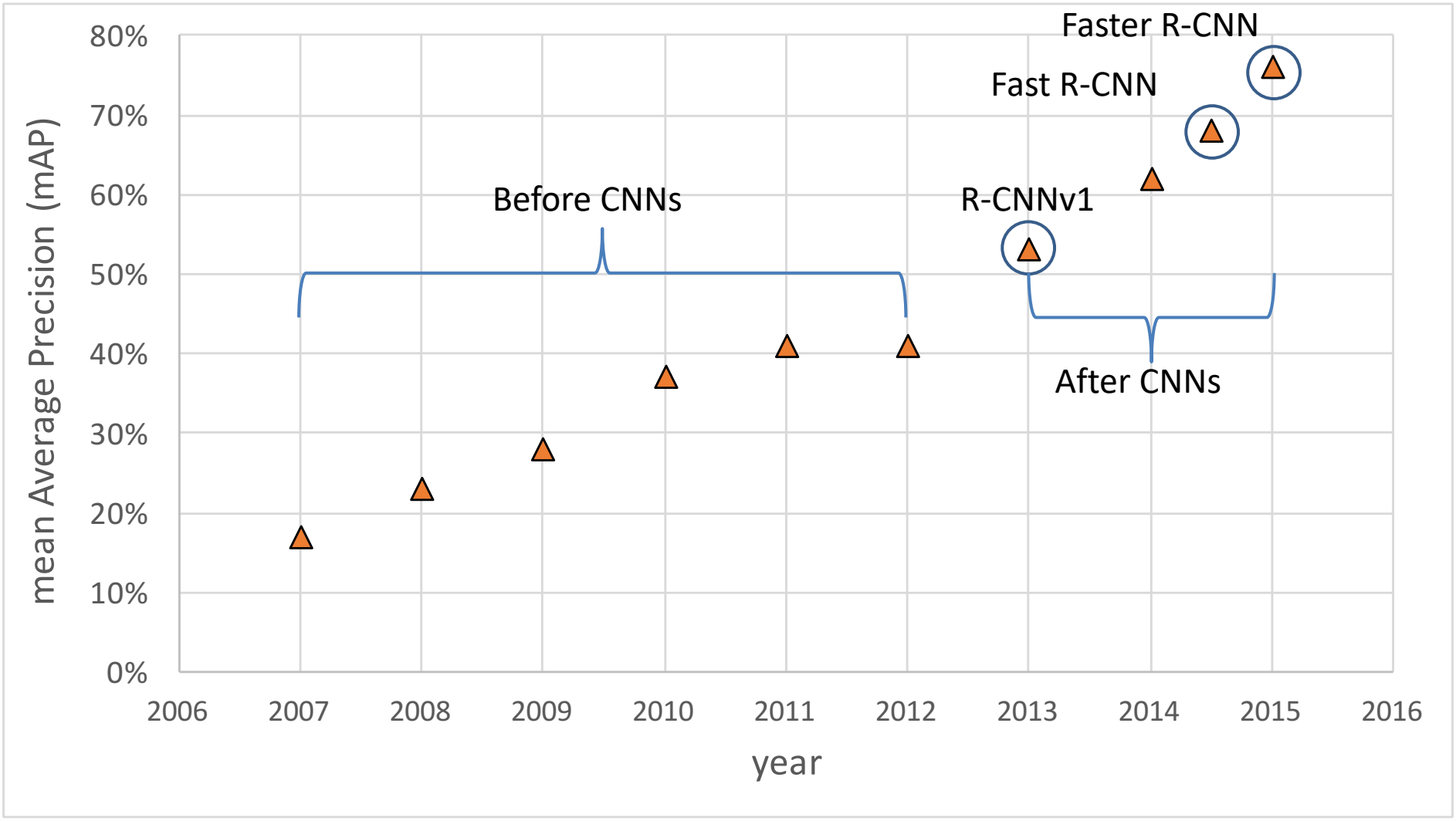



CNN features

# Progress on PASCAL detection



# Progress on PASCAL detection



# Project 2 – presentation

# Visual surveillance of on-street parking places

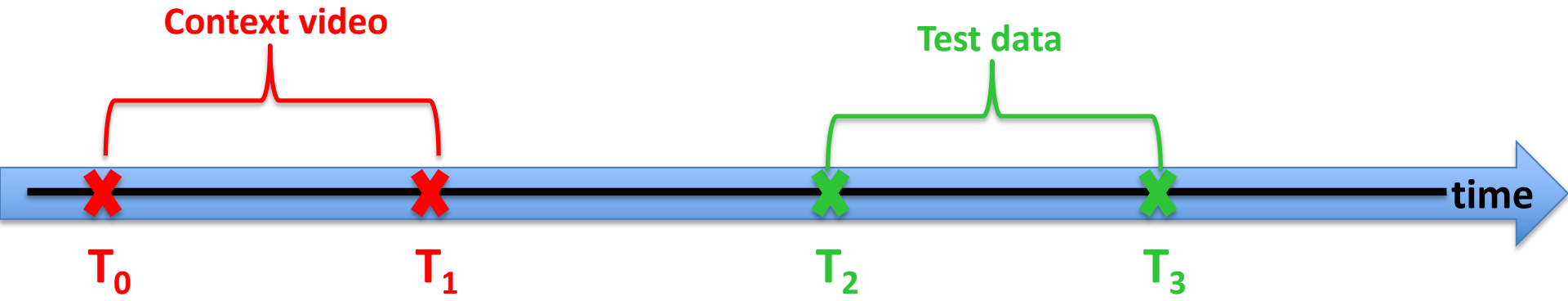
**tinyurl.com/CV-2024-Project2**

# Setup

- Static camera monitoring the on-street parking places in a scene
- Videos collected at different times during day/week
- Changes in illumination

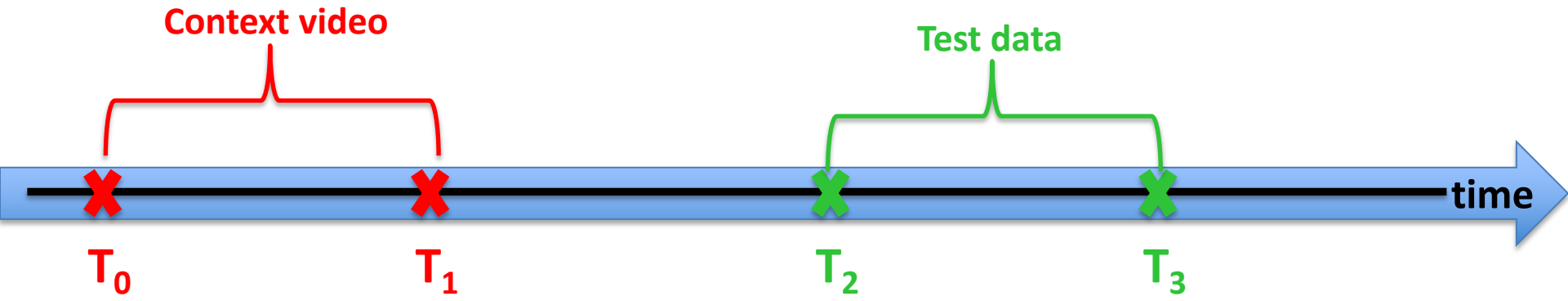
# Tasks

For each task you are given a “context” video that is taken with a few minutes before the test data (images or video) that you have to analyze is acquired.



# Tasks

For each task you are given a “context” video that is taken with a few minutes before the test data (images or video) that you have to analyze is acquired.



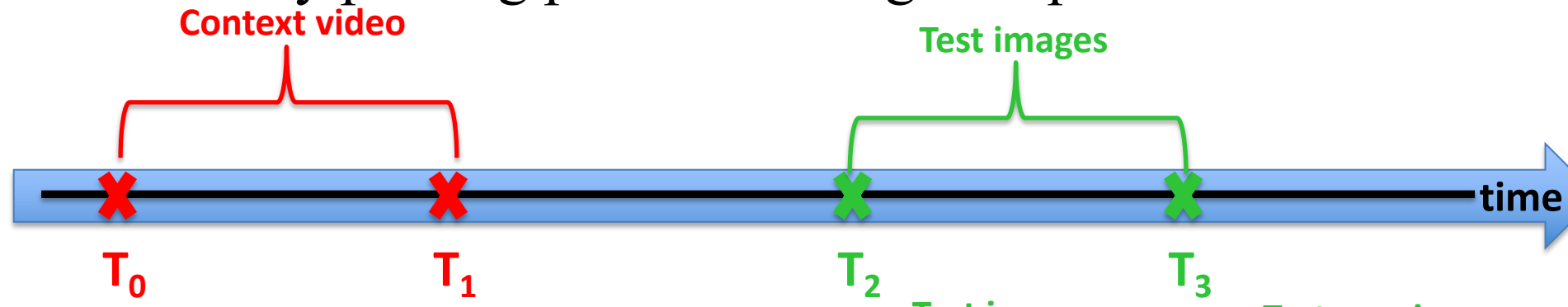

1. Classify parking places as being occupied or not
2. List the configuration of chosen 10 parking places
3. Track a specific vehicle in a video
4. Count the vehicles queueing at a traffic light in a video

# Task1 – on-street parking place numbering

1. Classify parking places as being occupied or not

# Task1

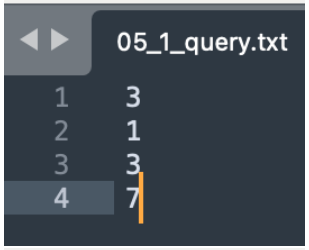
1. Classify parking places as being occupied or not

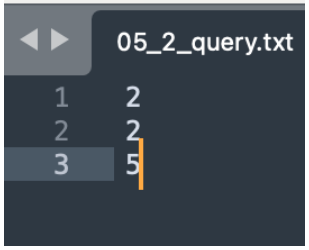


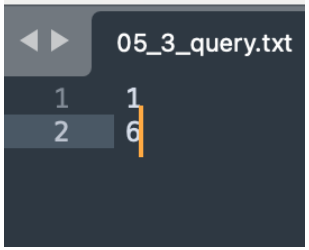

Context video

Test images

Test queries



# Task1

1. Classify parking places as being occupied or not

**Test images**

**Test queries**

05_1_query.txt

```
3
1
3
7
```

05_2_query.txt

```
2
2
5
```

05_3_query.txt

```
1
6
```

**Ground truth**

05_1_gt.txt

```
3
1 1
3 0
7 0
```

05_2_gt.txt

```
2
2 1
5 0
```

05_3_gt.txt

```
1
6 0
```

# Task 1 – classify parking places

- the task is to classify certain parking places based on a query as being occupied or not

- you are given 15 training examples (context videos + 3 or 4 images per video) for training (with annotated ground-truth)

- at test time you have to do the prediction on 50 test images (15 test videos, for the first 10 videos you get 3 test images, for the last 5 videos you get 4 test images)

- 50 images * 0.03 points/image = 1.5 points

# Task 2 – configuration of the 10 parking places

**input video**


**input initial configuration**

01.txt

| | |
|---|---|
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| 5 | 0 |
| 6 | 0 |
| 7 | 1 |
| 8 | 1 |
| 9 | 1 |
| 10 | 1 |

01_gt.txt

| | |
|---|---|
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| 5 | 0 |
| 6 | 1 |
| 7 | 1 |
| 8 | 1 |
| 9 | 1 |
| 10 | 1 |

**output final configuration**

# Task 2 – configuration of the 10 parking places

- the task is to list the configuration of the 10 on-street parking places given the initial configuration and a video to process

- you are given 15 videos for training (with annotated ground-truth: initial + final configuration)

- at test time you have to do the prediction on 15 videos (you are given only the initial configuration, you should output the final configuration)

- 15 videos * 0.05 points/video = 0.75 points

# Task 3 – track a specific vehicle

# Task 3 – track a specific vehicle

## Training example 1

01.txt

```
1771 -1 -1 -1 -1
0 468 302 550 376
```

01_gt.txt

```
1771 -1 -1 -1 -1
0 468 302 550 376
1 466 302 548 376
2 469 300 551 374
3 466 303 548 377
4 471 309 553 383
5 468 305 550 379
6 473 305 555 379
7 472 306 554 380
8 470 306 552 380
9 474 306 556 380
10 473 307 555 381
11 471 307 553 381
12 471 307 556 380
13 470 311 558 385
14 472 309 560 383
15 470 308 554 384
16 473 309 557 385
17 475 311 559 387
18 477 309 561 385
19 475 307 561 387
20 473 312 561 385
21 470 311 559 389
22 474 309 563 387
23 474 311 559 386
24 476 311 561 386
25 478 314 564 392
26 476 312 565 395
```

# Task 3 – track a specific vehicle

In each video we will consider that your algorithm *correctly tracks the vehicle* if in more (greater or equal) than 80% of the video frames your algorithm *correctly localizes the vehicle to be tracked*. We consider that your algorithm *correctly localizes the vehicle to be tracked* in a specific frame if the value of the IOU (intersection over union) between the window provided by your algorithm and the ground-truth window is more than 30%. The format that you need to follow is the one used in the ground-truth files (located in the subdirectory *ground-truth*) with the first line containing the number of frames N of the video, and each line having the format [frame_index xmin ymin xmax ymax]. The first frame for which we provide the bounding box initialization has frame index 0, the last frame of a video with $N$ frames has frame index $N - 1$. Please note that the first line of the annotation file has the format [N -1 -1 -1 -1] as it is easy to load the entire matrix $(N + 1) \times 5$ to assess the correctness of your algorithm.

# Task 3 – track a specific vehicle

- the task is to track a specific vehicle in a given video (you are given the initial bounding box in the first frame of the video)

- you are given 15 videos for training (with annotated ground-truth)

- at test time you have to do the prediction on 15 test videos

- 15*0.1 points/video = 1.5 points

# Task 4 - Vehicle numbering

4. Count the vehicles queueing at the traffic light

**input video**

**output**

02_gt.txt

1 4

# Task 4 – count the number of vehicles

- the task is to count the number of vehicles which queue to the traffic light located in the upper part of the scene

- you are given 15 videos for training (with annotated ground-truth)

- at test time you have to make predictions on 15 test videos

- 15*0.05 points/video = 0.75 points

# Summary - Visual surveillance of on-street parking places

**Task 1:**

- 50 images * 0.03 points/image = 1.5 points

**Task 2:**

- 15 videos * 0.05 points/video = 0.75 points

**Task 3:**

- 15 videos * 0.1 points/video = 1.5 points

**Task 4:**

- 15 videos * 0.05 points/video = 0.75 points

**Oral Presentation + pdf**

- 0.5 points

**Ex-officio (right format + right paths):**

- 0.5 points

**Total: 5.5 points**

# Project 2 details

- data is already released here: **tinyurl.com/CV-2024-Project2**

- read carefully the pdf (format, etc)

- Deadlines:
  - Tuesday 25th of June (code submission)
  - Wednesday 26th of June (results submission on released test set + presentation f2f or online)