中山大學
SUN YAT-SEN UNIVERSITY

Lecture 11.
Detecting objects by parts

Pattern Recognition and Computer Vision

Guanbin Li,
School of Computer Science and Engineering, Sun Yat-Sen University

扫码签到

What we will learn today?

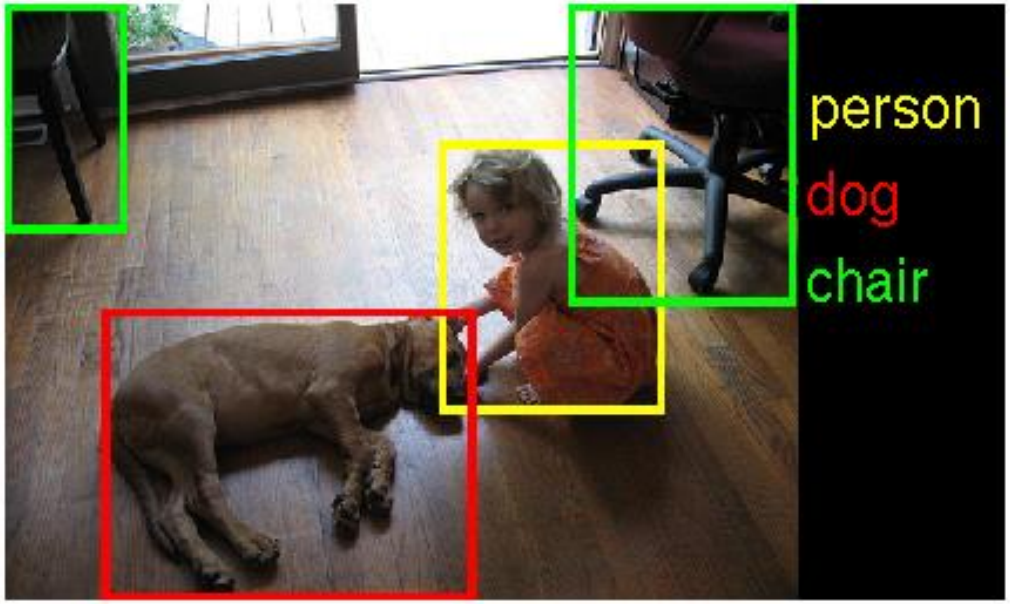
- Object detection – Task and evaluation
- A simple detector
- Deformable parts model

Object Detection

- What do you see in the image?

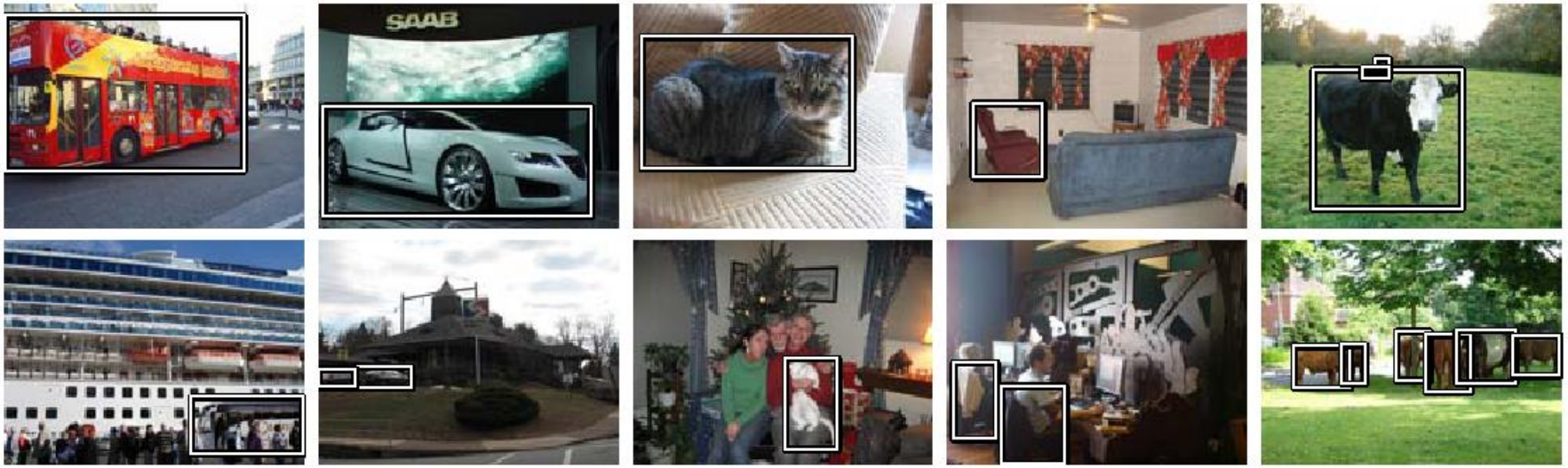
Object Detection

- Problem: Detecting and localizing generic objects from various categories, such as cars, people, etc.
- Challenges: Illumination, viewpoint, deformations, Intra-class variability.



Object Detection Benchmarks

- PASCAL VOC Challenge
  - 20 categories
  - Annual classification, detection, segmentation, … challenges



Object Detection Benchmarks

- PASCAL VOC Challenge
- ImageNet Large Scale Visual Recognition Challenge (ILSVR)
  - 200 Categories for detection

Object Detection Benchmarks

- PASCAL VOC Challenge
- ImageNet Large Scale Visual Recognition Challenge (ILSVR)
- Common Objects in Context (COCO)
  - 80 Object categories



How do we evaluate object detection?

predictions

ground truth

How do we evaluate object detection?

predictions

ground truth

True positive:

- The overlap of the prediction with the ground truth is MORE than 0.5

How do we evaluate object detection?

predictions

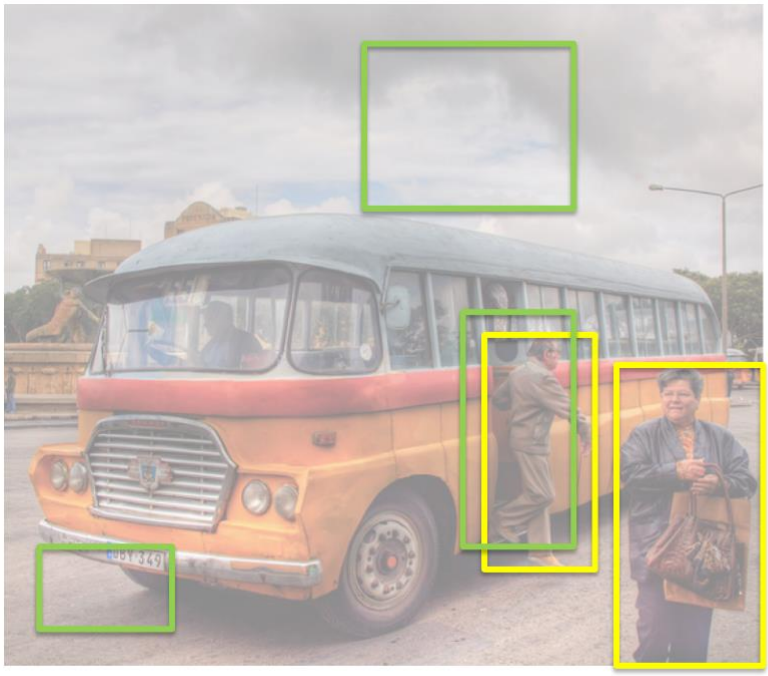
ground truth

True positive:

False positive:

- The overlap of the prediction with the ground truth is LESS than 0.5

How do we evaluate object detection?



predictions

ground truth

True positive:

False positive:

False negative:

- The objects that our model doesn't find

What is a True Negative?

How do we evaluate object detection?

	Predicted 1	Predicted 0
True 1	true positive	false negative
True 0	false positive	true negative

	Predicted 1	Predicted 0
True 1	TP	FN
True 0	FP	TN

	Predicted 1	Predicted 0
True 1	hits	misses
True 0	false alarms	correct rejections

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

How do we evaluate object detection?

predictions

ground truth

True positive: 1

False positive: 2

False negative: 1

So what is the
- precision?
- recall?

Precision versus recall

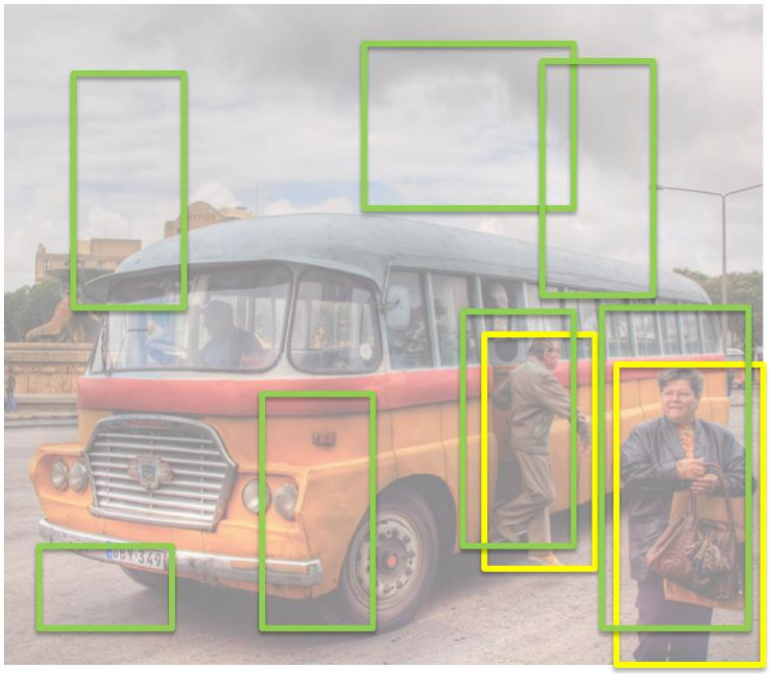
- Precision:

  –how many of the object detections are correct?

- Recall:

  –how many of the ground truth objects can the model detect?

How do we evaluate object detection?



predictions

ground truth

Here are all the boxes that are predicted with score > 0.

This means that our
- Recall is perfect!
- But our precision is BAD!

How do we evaluate object detection?

predictions

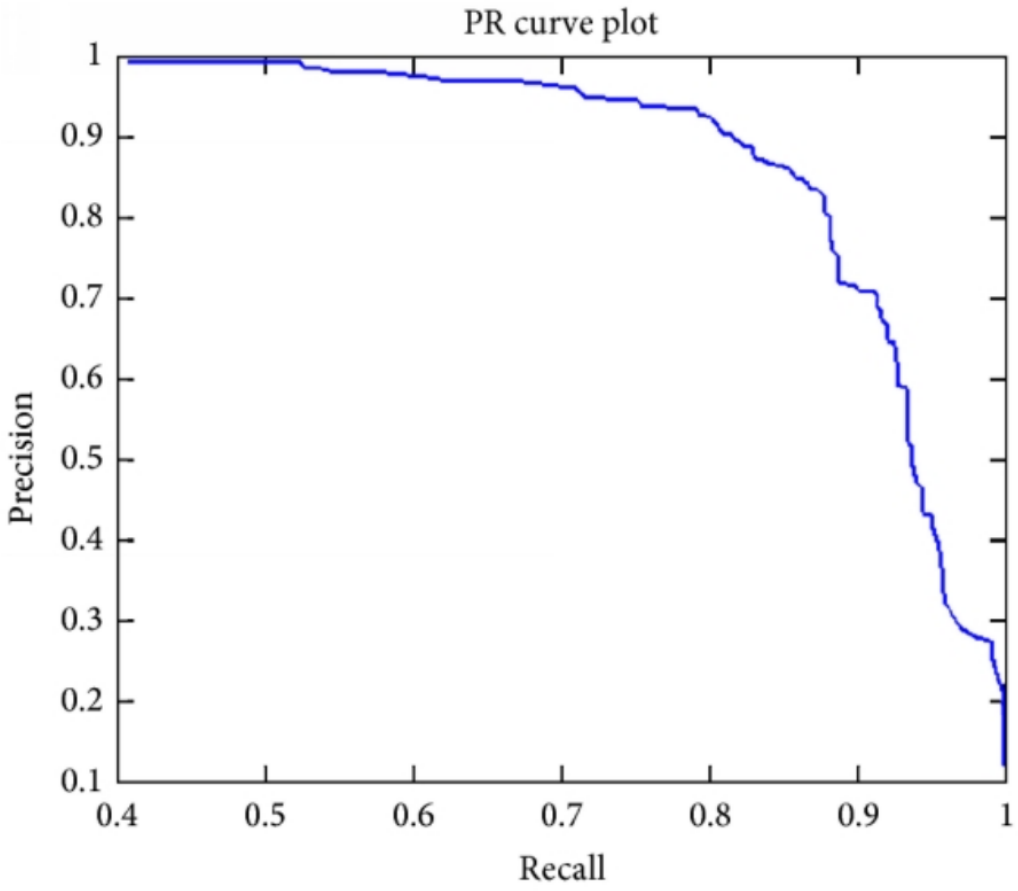
ground truth

Here are all the boxes that are predicted with score > 0.5

We are setting a threshold of 0.5

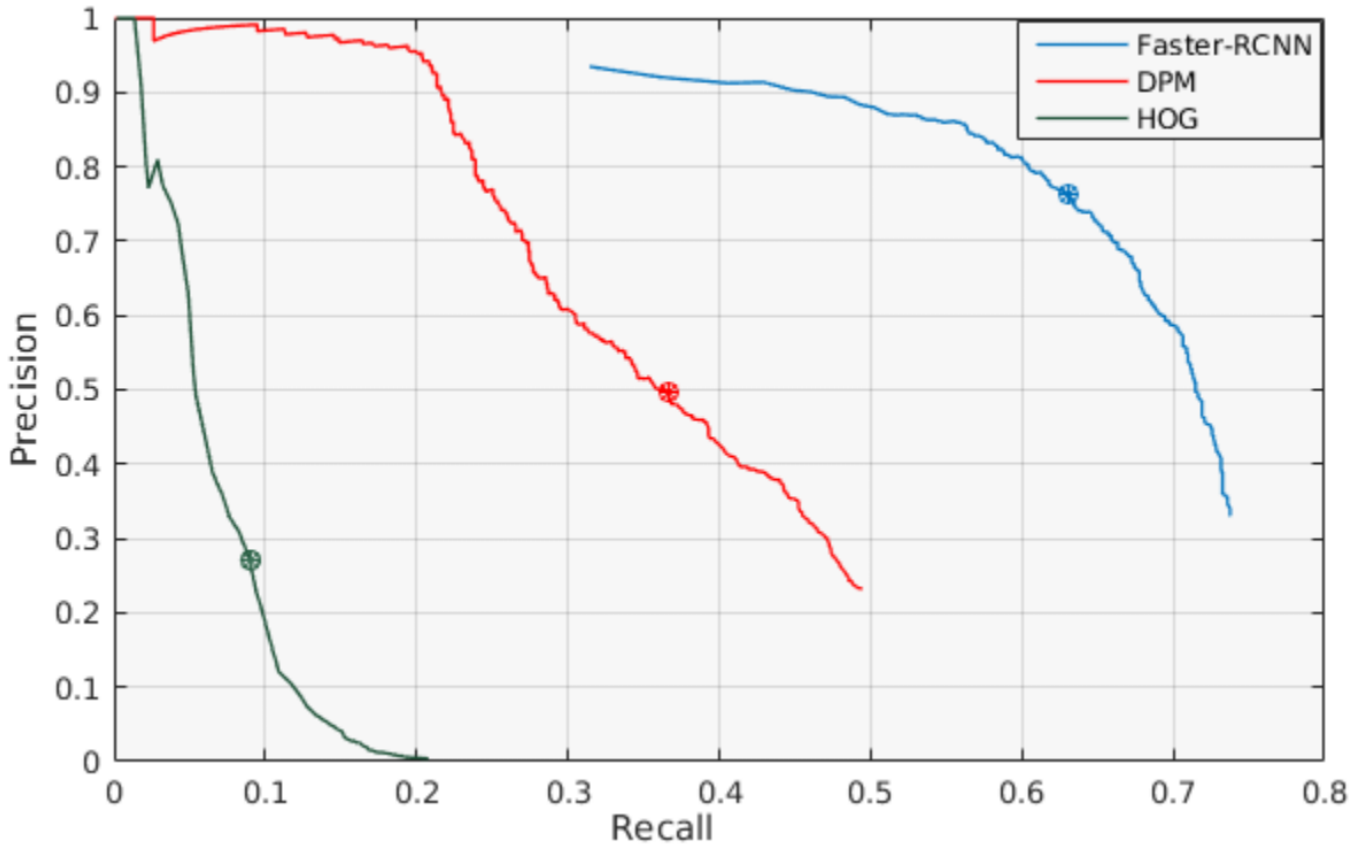
How do we evaluate object detection?

Precision – recall curve (PR curve)



How do we evaluate object detection?

Which model is the best?



Some examples

True Positives - Person

UoCTTI_LSVM-MDPM

MIZZOU_DEF-HOG-LBP

NECUIUC_CLS-DTCT

Some examples

False Positives - Person

UoCTTI_LSVM-MDPM

MIZZOU_DEF-HOG-LBP

NECUIUC_CLS-DTCT

Some examples

"Near Misses" - Person

UoCTTI_LSVM-MDPM

MIZZOU_DEF-HOG-LBP

NECUIUC_CLS-DTCT

Some examples

True Positives - Bicycle

UoCTTI_LSVM-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT

Some examples

False Positives - Bicycle

UoCTTI_LSVM-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT

What we will learn today?

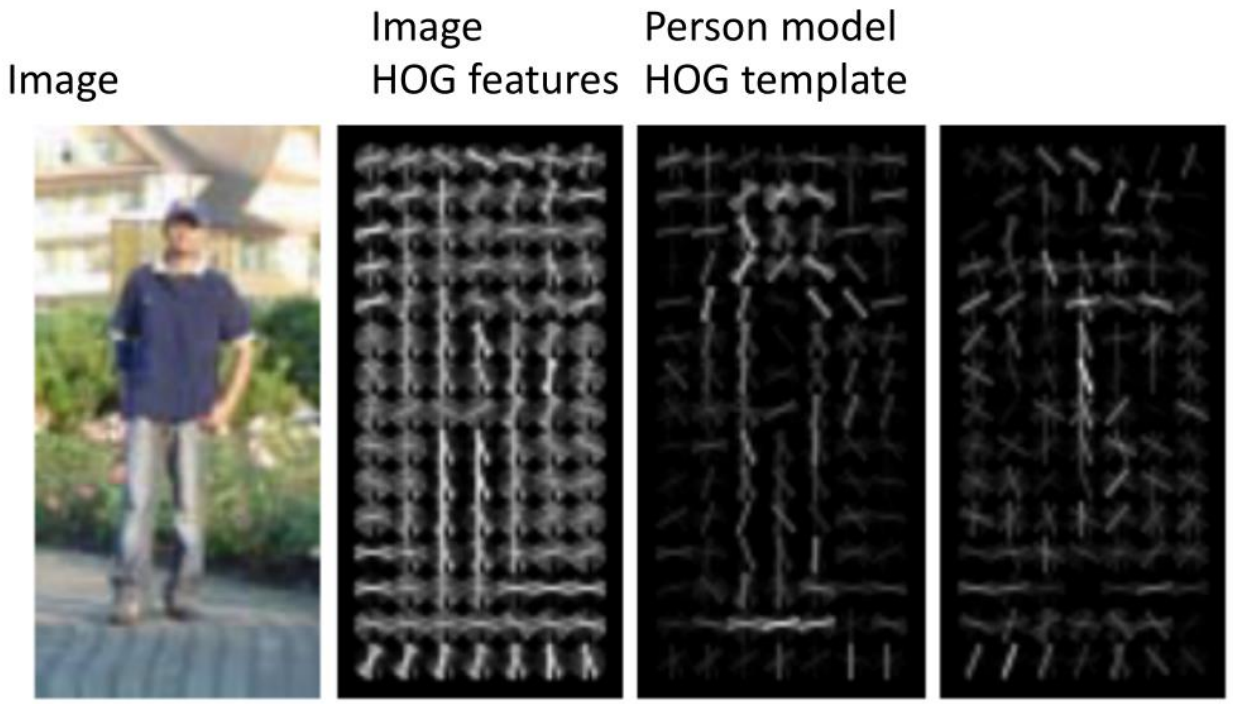
- Object detection – Task and evaluation
- A simple detector
- Deformable parts model

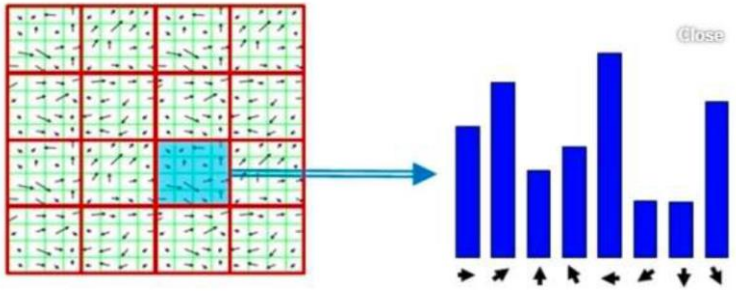
Dalal-Triggs method

sliding window

Recap – HOG features

- Find a HOG template and use as filter

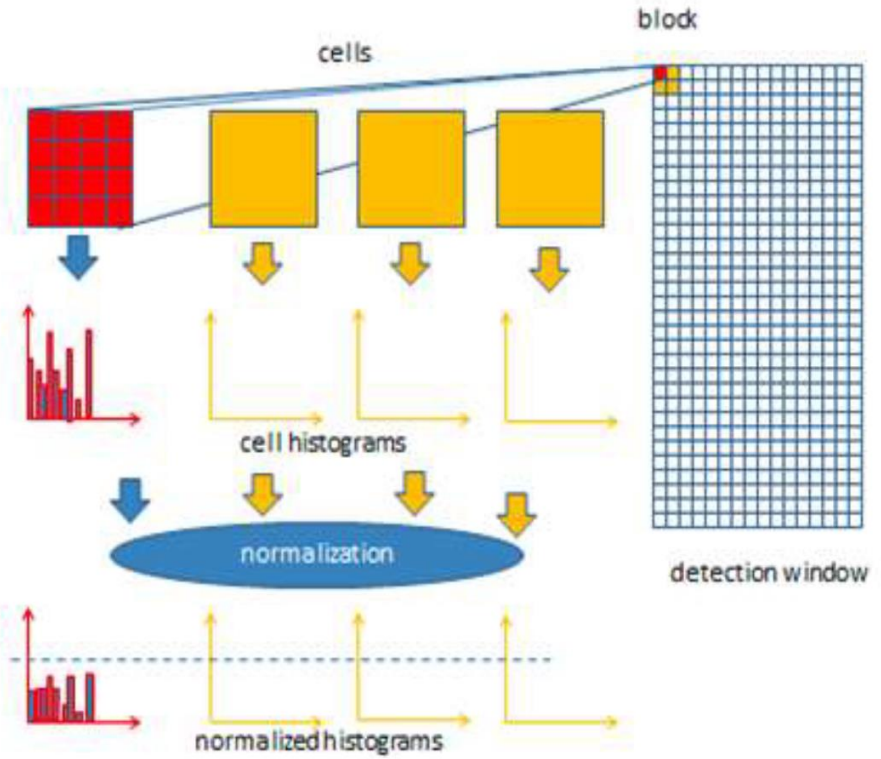




Recap – HOG features

Steps to calculate HOG

1. Preprocessing(resizing)
2. Calculate Gradient Images
3. Calculate Histogram of Gradients in 8×8 cells
4. Block Normalization
5. Form HOG feature vector

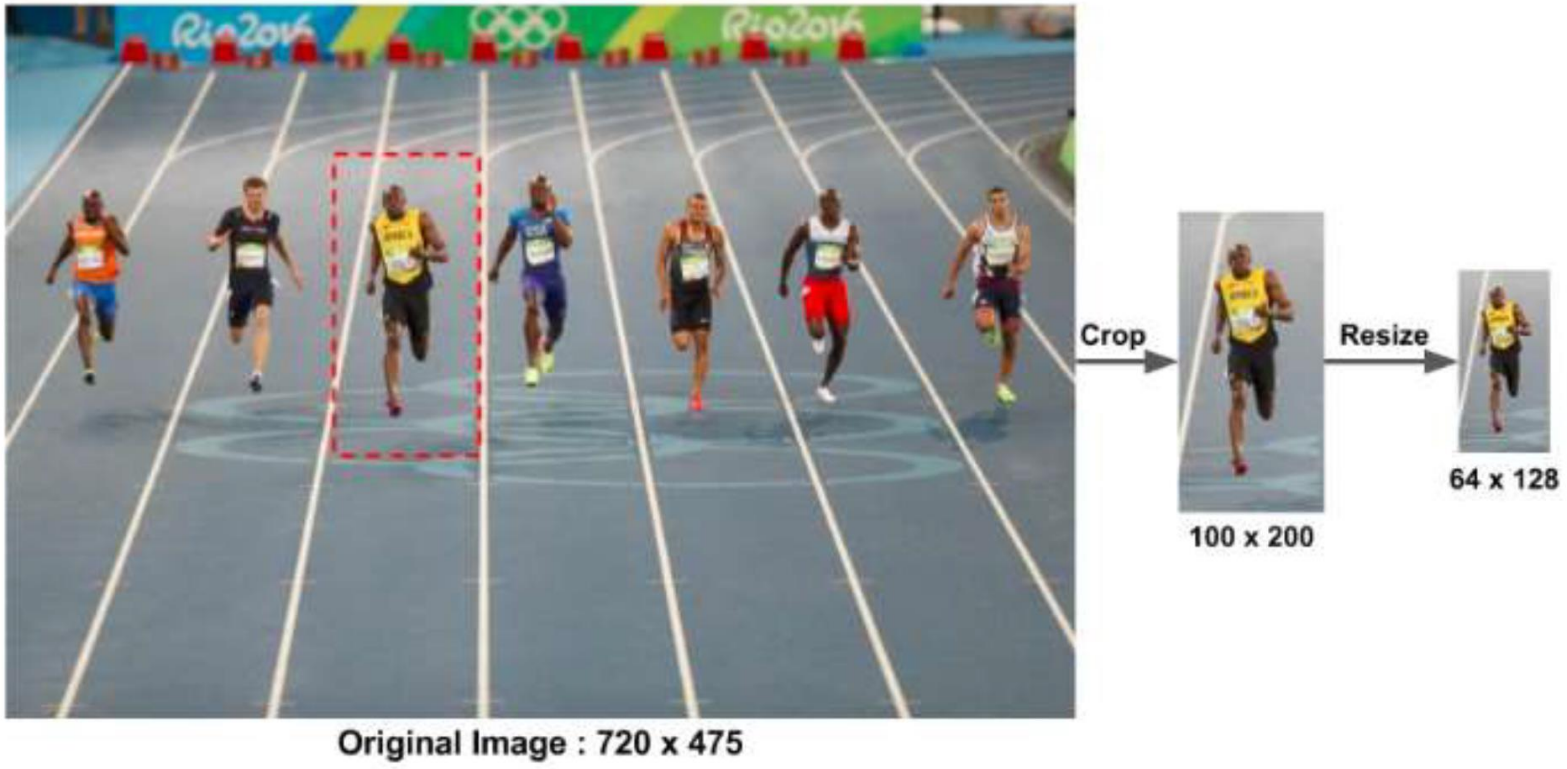


Recap – HOG features

Pre-Processing

- Patches being analyzed should have a fixed aspect ratio.

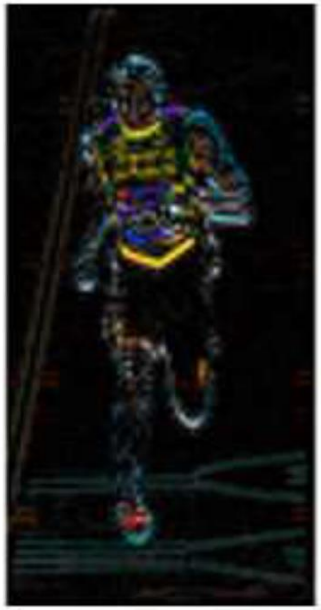
For example, they can be 100×200, 128×256, or 1000×2000 but not 101×205.



Recap – HOG features

Calculate Gradient Images

- Get information on the movement of energy from left to right and up to down
- Apply 1D filter kernel



Left : Absolute value of x-gradient. Center : Absolute value of y-gradient.
Right : Magnitude of gradient.

-1	0	1

-1
0
1

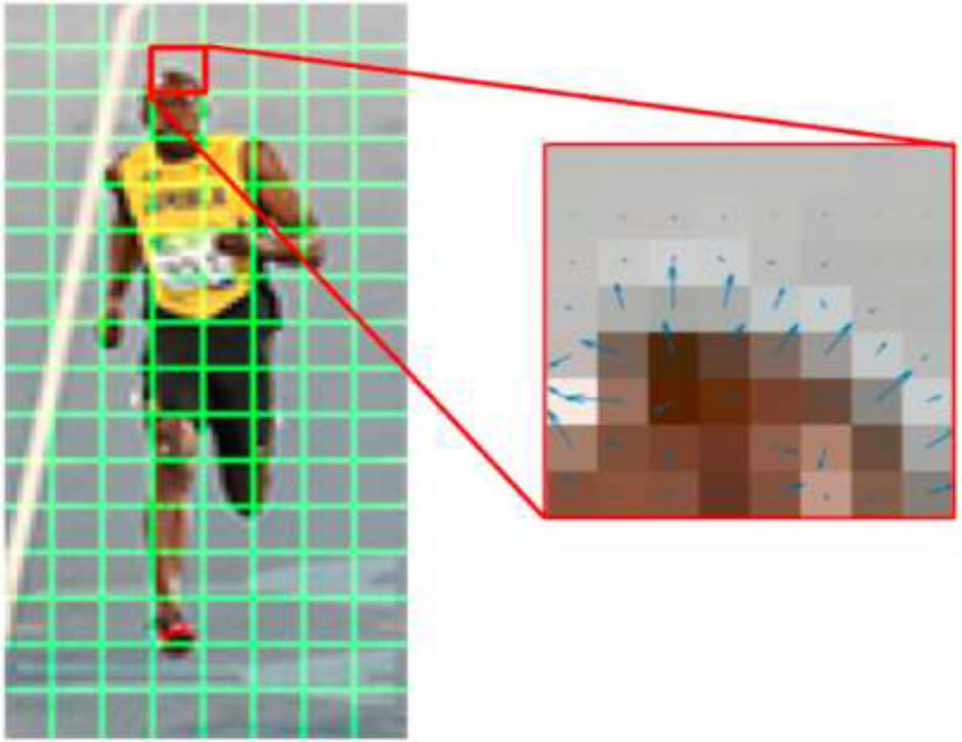
$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \arctan \frac{g_y}{g_x}$$

Recap – HOG features

Calculate Histogram of Gradients in 8*8 cells

- The gradient of an 8×8 patch contains 2 values (magnitude and direction) per pixel which adds up to 8x8x2 = 128 numbers



2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

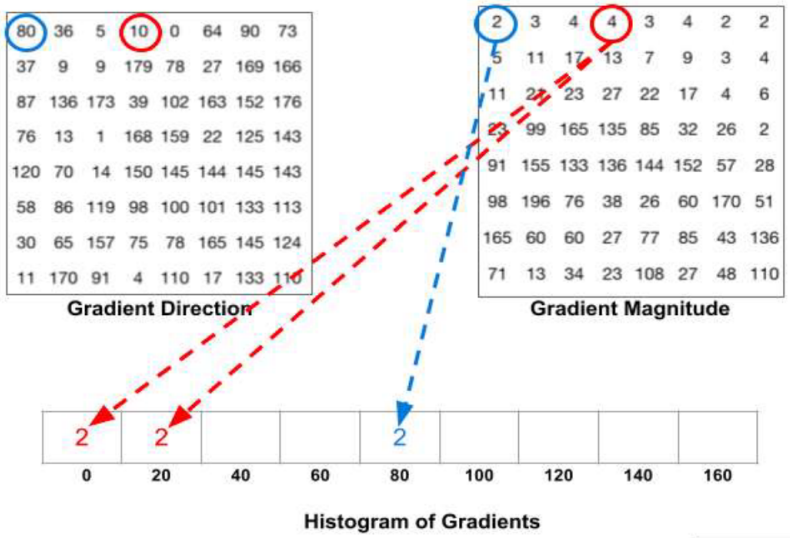
Gradient Magnitude

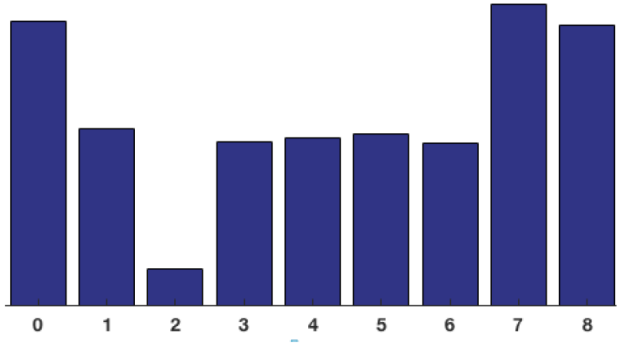
80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

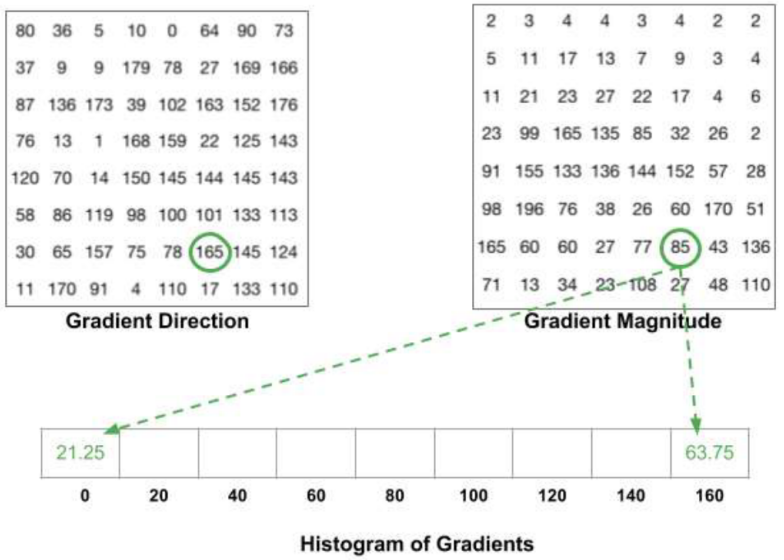
Center : The RGB patch and gradients represented using arrows. Right : The gradients in the same patch represented as numbers

Recap – HOG features





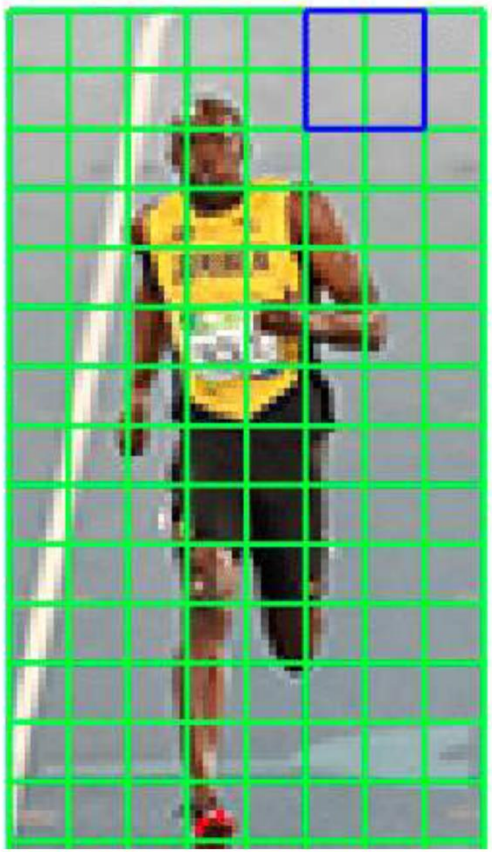
For color images, the gradients of the three channels are evaluated. The magnitude of gradient at a pixel is the maximum of the magnitude of gradients of the three channels, and the angle is the angle corresponding to the maximum gradient.



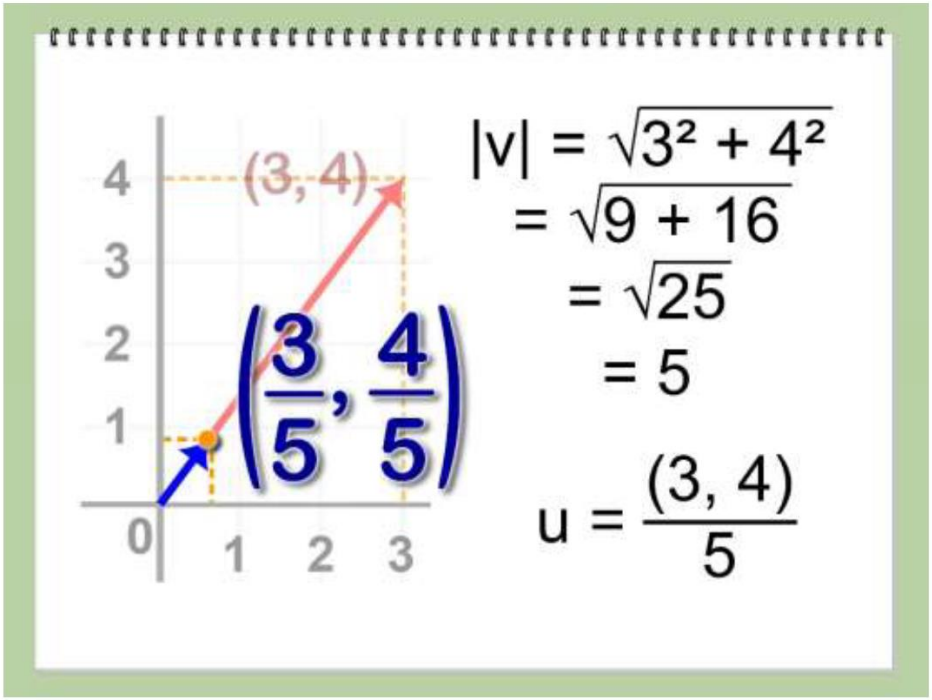
Recap – HOG features

Block Normalization

- Normalize the histogram so they are not affected by lighting variations.
- Normalizing a vector removes the scale.



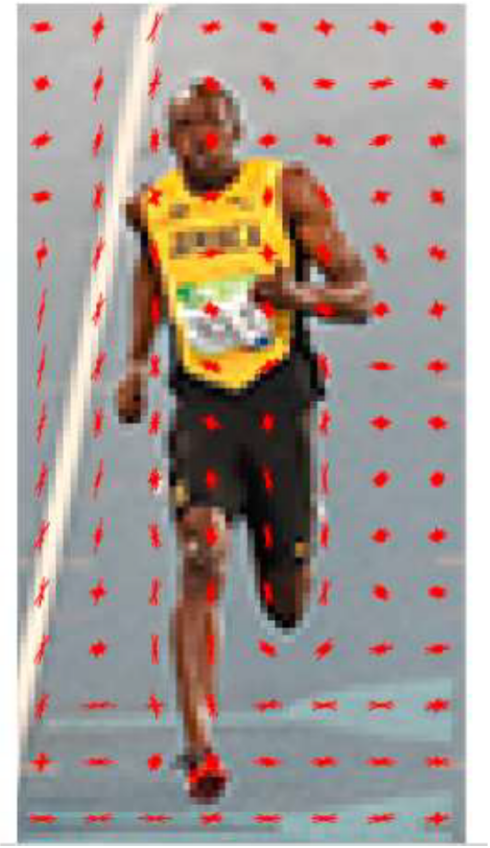
• A 16×16 block has 4 histograms which can be concatenated to form a 36 x 1 element vector



Recap – HOG features

Calculate the HOG feature vector

To calculate the final feature vector for the entire image patch, the 36×1 vectors are concatenated into one giant vector



1. How many positions of the 16×16 blocks do we have ? There are 7 horizontal and 15 vertical positions making a total of 7 x 15 = 105 positions.
2. Each 16×16 block is represented by a 36×1 vector. So when we concatenate them all into one gaint vector we obtain a 36×105 = **3780**dimensional vector.

Sliding window + hog features

No person here

- Slide through the image and check if there is an object at every location

Sliding window + hog features

YES!! Person match found

- Slide through the image and check if there is an object at every location

Sliding window + hog features

- But what if we were looking for buses?

Sliding window + hog features

No bus found

- But what if we were looking for buses?

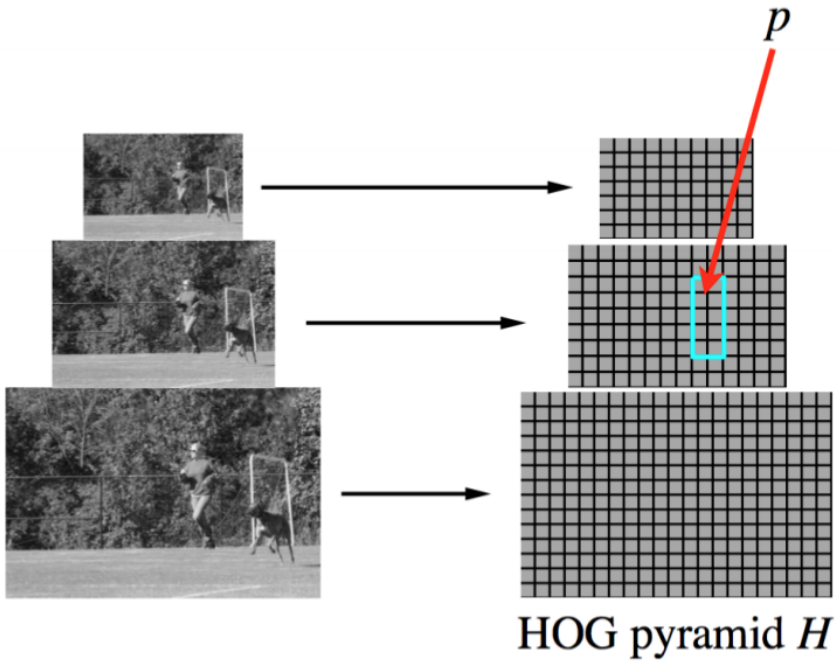
Sliding window + hog features

- We will never find the object if we don’t choose our window size wisely!

Sliding window + hog features

- We need to do multi scale sliding window

Create a feature pyramid



Filter F

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

$\phi(p, H)$ = concatenation of HOG features from subwindow specified by p

What we will learn today?

- Object detection – Task and evaluation
- A simple detector
- Deformable parts model

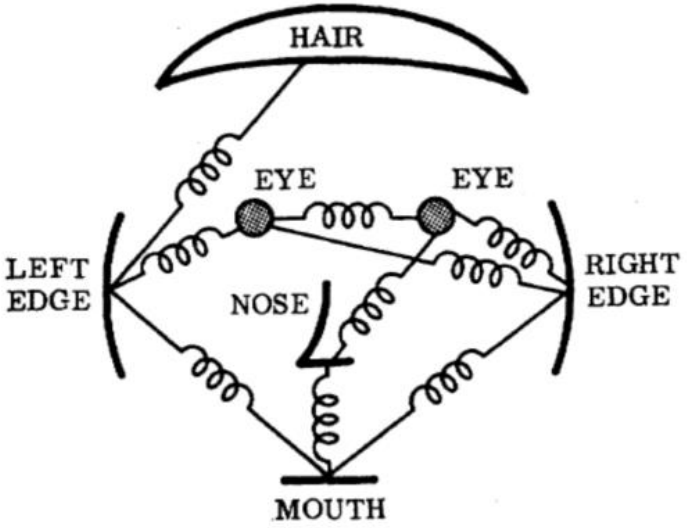
Recap – bag of visual words

- We can present images as a set of words
  –Where each word represents a part of the image.

- Can we do the same for objects within those images?

Deformable Parts Model

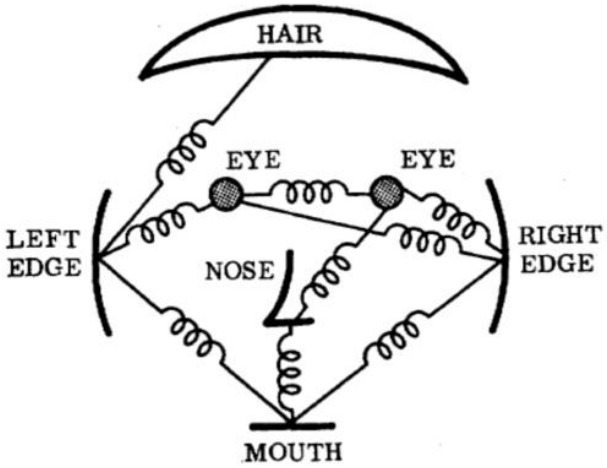
- Represents an object as a collection of parts arranged in a deformable configuration
- Each part represents local appearances
- Spring-like connections between certain pairs of parts



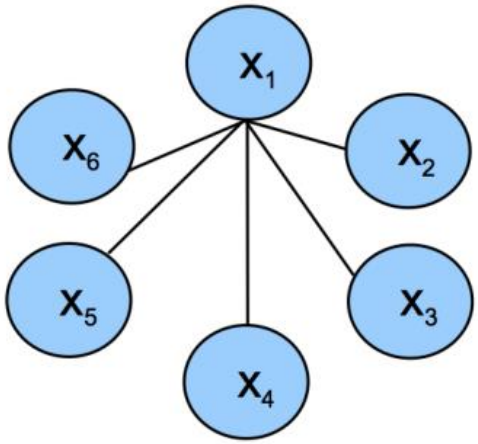
M. Fischler and R. Elschlager, "The Representation and Matching of Pictorial Structures," 1973.

Deformable Parts Model

- The parts of an object form pairwise relationships.

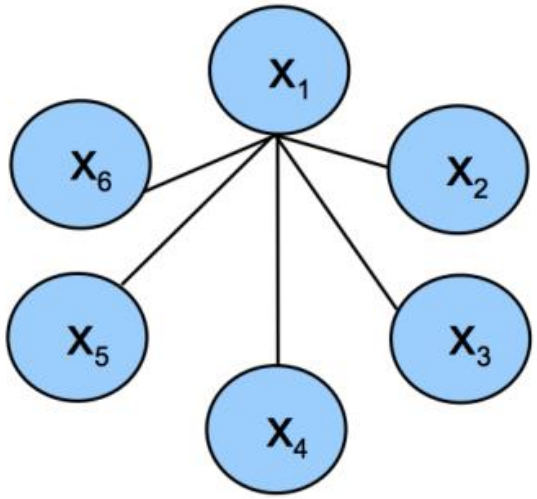


- We can model this using a "star model", where every part is defined relative to a root.



Deformable Parts Model

- For example, a person can be modelled as having a head, left arm, right arm, etc.
- All parts can be modelled relative to the global person detector, which acts as the root.

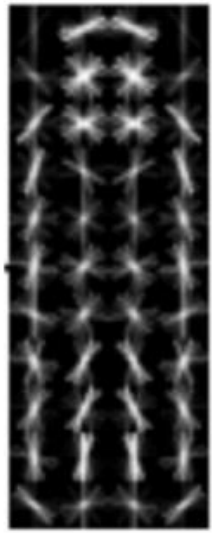


Deformable Parts Model

DPM includes：

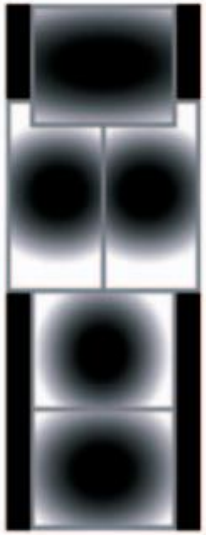
- A global filter and a set of part filters which have finer details.
- A spatial model for the location of each part relative to the root.

Here is an example of a person model：



Global/root filter

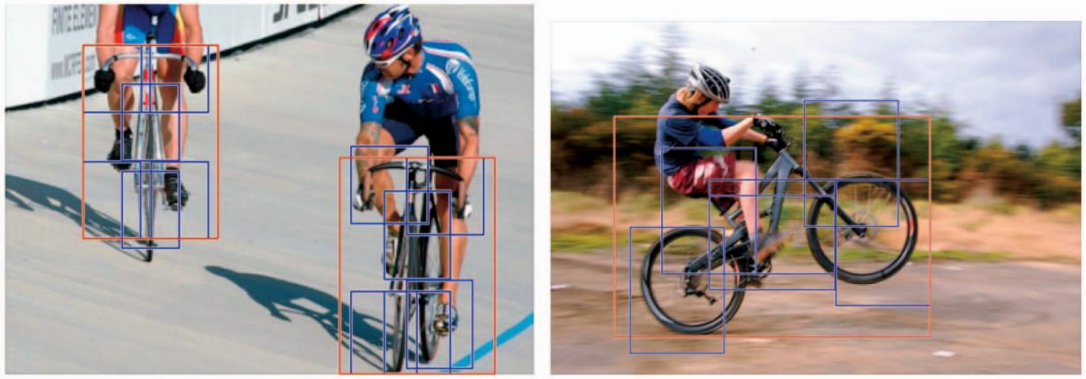
Several part filter



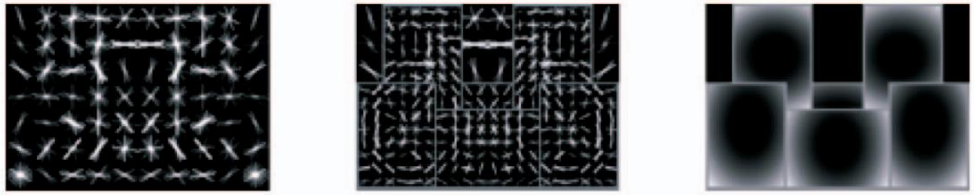
Spatial model

Deformable Parts Model

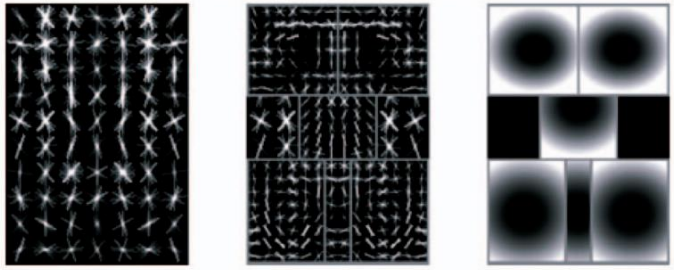
- Two-component bicycle model



sideways views



frontal and
near frontal views



Deformable Parts Model

- A model for an object with n parts is a $(n+2)$ tuple:

$$(F_0, P_1, \cdots, P_n, b)$$

Root filter — Model for 1st part — Bias term

- Each part-based model defined as:

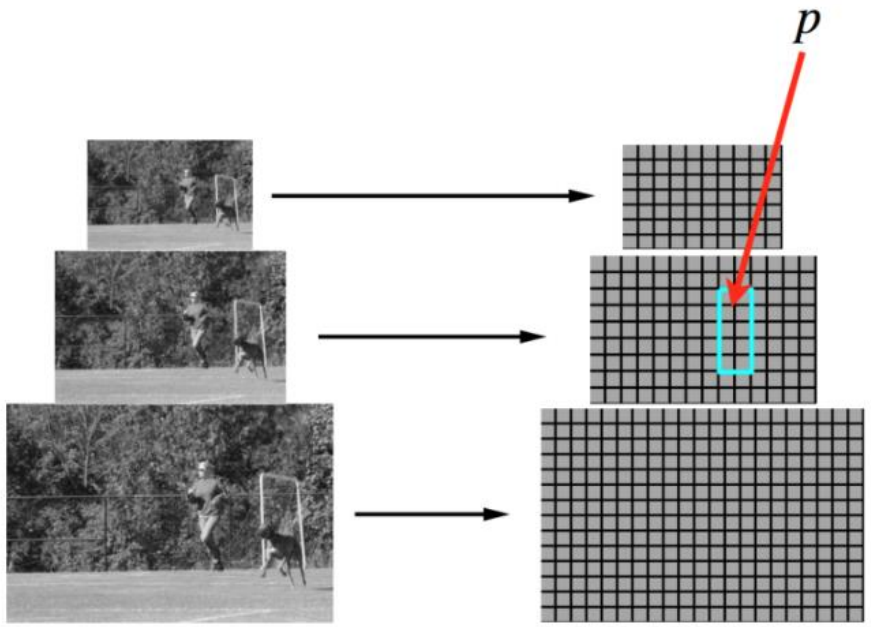
$$(F_i, v_i, d_i)$$

F_i filter for the *i*-th part

v_i "anchor" position for part *i* relative to the root position

d_i defines a deformation cost for each possible placement of the part relative to the anchor position

Remember from Dalal and Triggs



Filter F

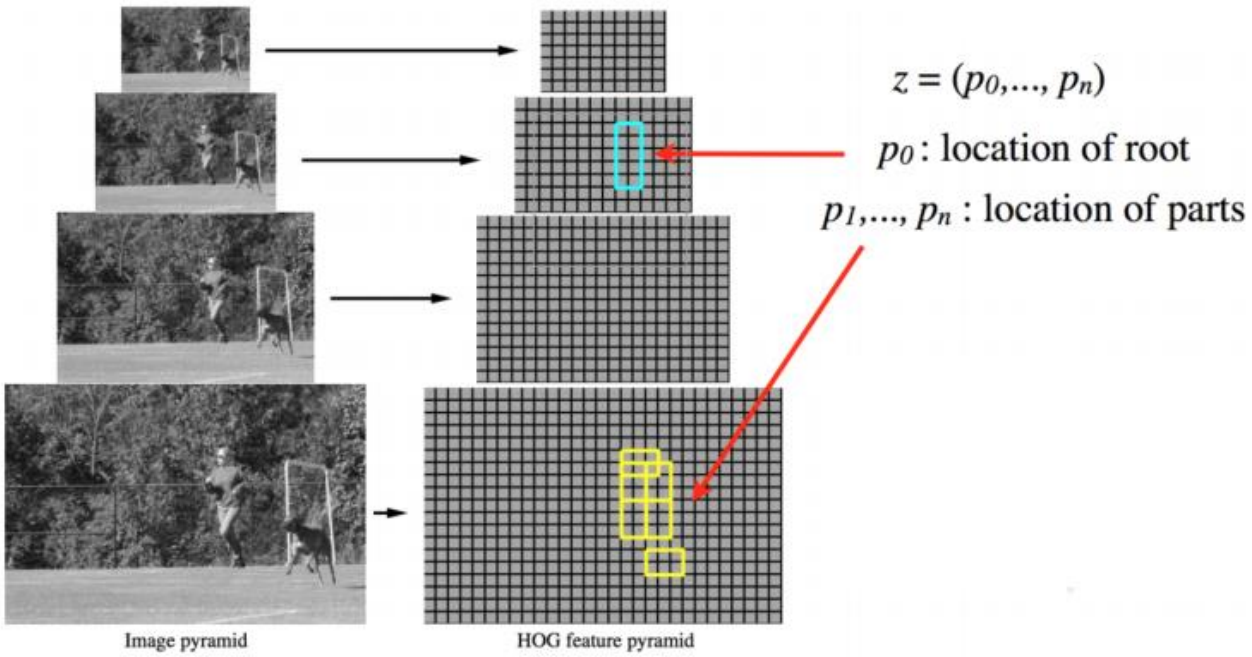
Score of F at position p is
$F \cdot \phi(p, H)$

$\phi(p, H)$ = concatenation of HOG features from subwindow specified by p

Calculating the score for a detection

- Deformable parts calculates a score for each part along with a global score

$p_i = (x_i, y_i, l_i)$ specifies the level and position of the *i*-th filter



Calculating the score for a detection

- The score for a detection is defined as the sum of scores for the global and part detectors minus the sum of deformation costs for each part.
- This means that if a detection's parts are really far away from where they should be, it's probably a false positive

$$detection\ score = \sum_{i=0}^{n} F_i \, \phi(p_i, H) - \sum_{i=1}^{n} d_i\left(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2\right)$$

Calculating the score for a detection

$$detection\ score = \sum_{i=0}^{n} F_i\, \phi(p_i, H) - \sum_{i=1}^{n} d_i\left(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2\right)$$

Scores for each part filter + global filter (similar to Dalal and Triggs).

Calculating the score for a detection

$$detection\ score = \sum_{i=0}^{n} F_i\, \phi(p_i, H) - \sum_{i=1}^{n} d_i\left(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2\right)$$

The deformation costs for each part.

Δx_i measures the distance in the x-direction from where part i should be.

Δy_i measures the same in the y-axis direction.

d_i is the weight associated for part i that penalizes the part for being away.

In general the deformation cost is an arbitrary separable quadratic function of the displacements.

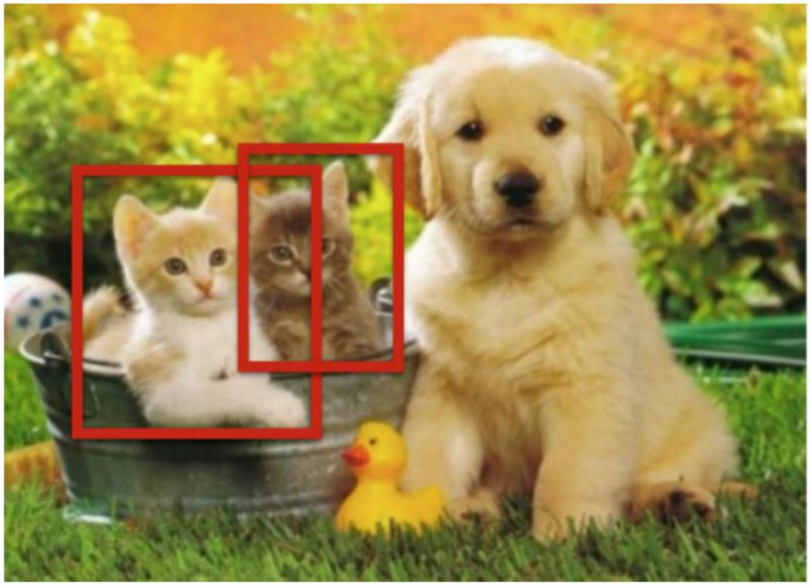
Calculating the score for a detection

$$detection\ score = \sum_{i=0}^{n} F_i\, \phi(p_i, H) - \sum_{i=1}^{n} d_i\left(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2\right)$$

If d_i = (0, 0, 1, 0). What does this mean?

Detection pipeline

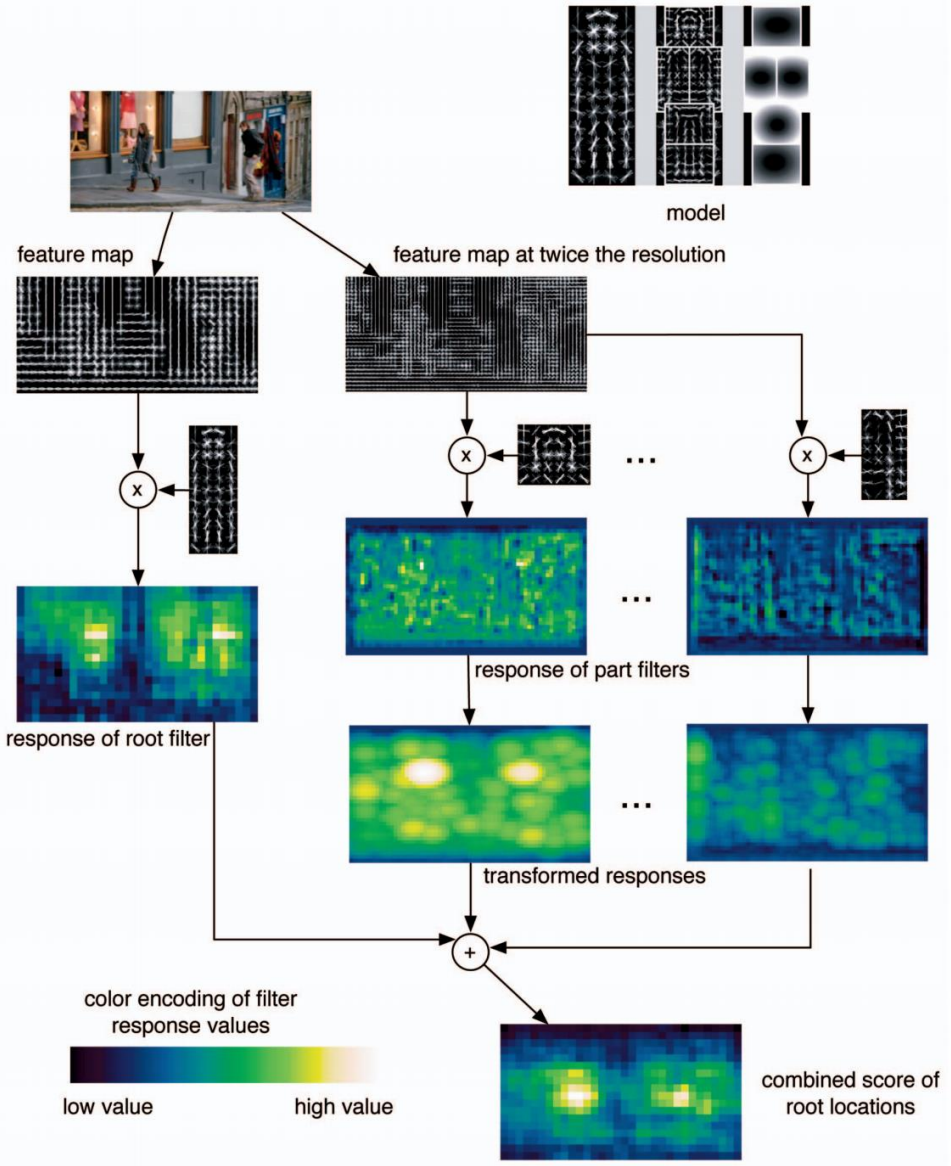
- So, to make a detection, we use the sliding window technique and with the global and part filters.
- To score a detection, we accumulate the global and part scores and penalize the deformation of the parts.



Detection pipeline

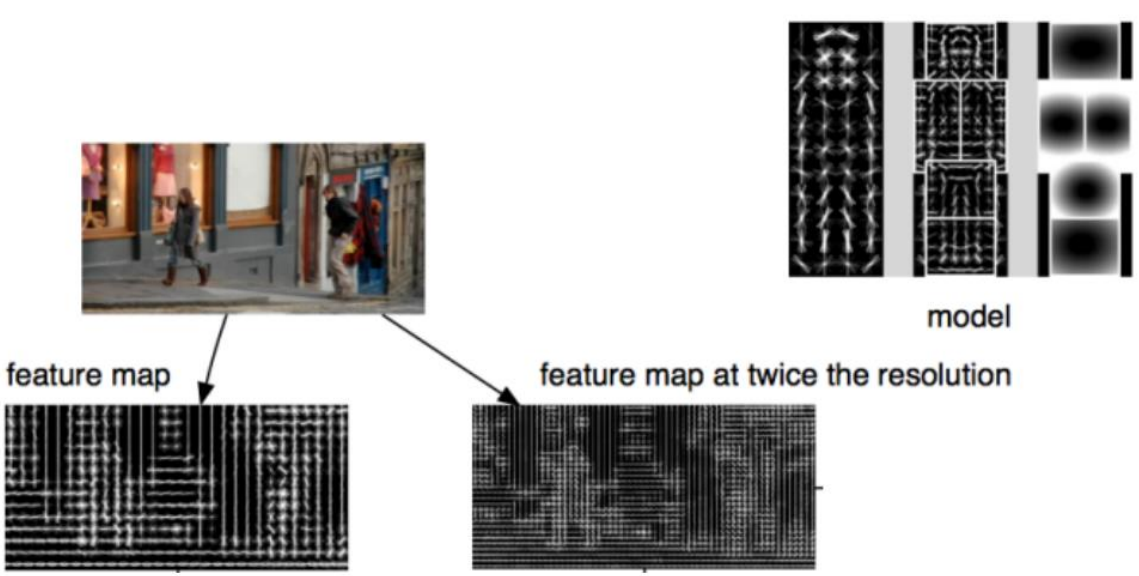
Overall detection pipeline

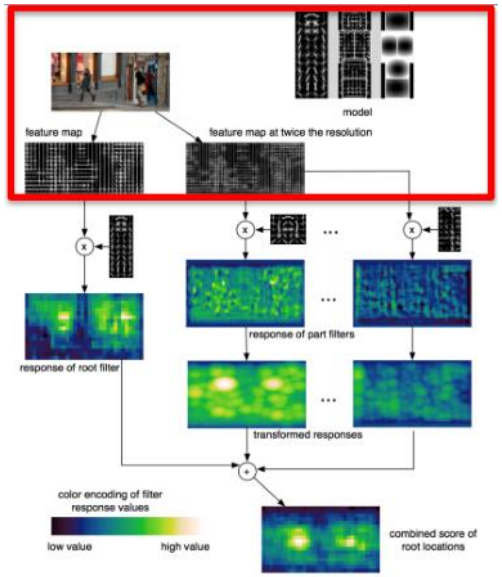
Let's break this down



Detection pipeline

1. Make sure you have filters for the global and the parts: F_i
2. Compute HOG feature maps from the input image

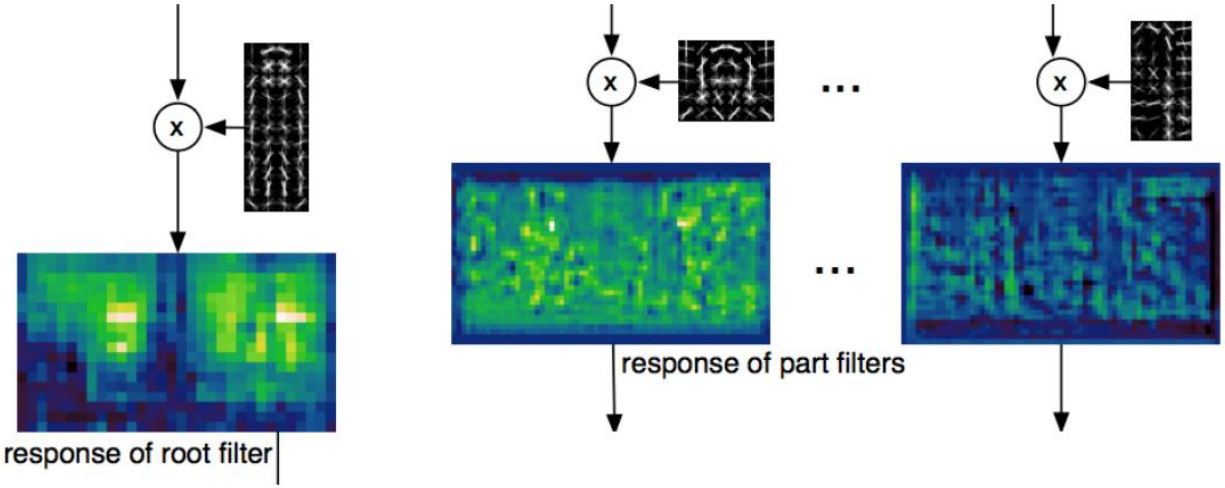


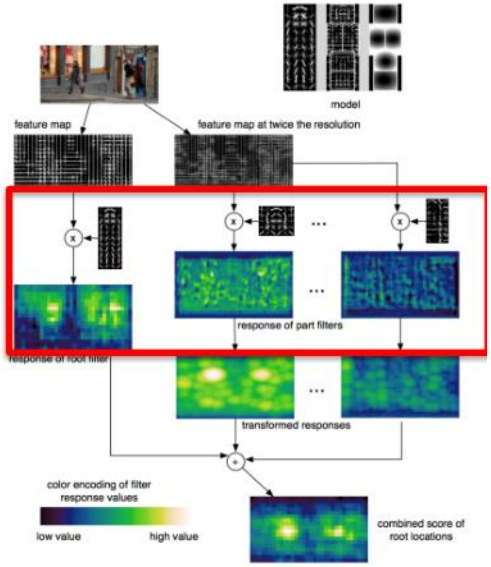


Detection pipeline

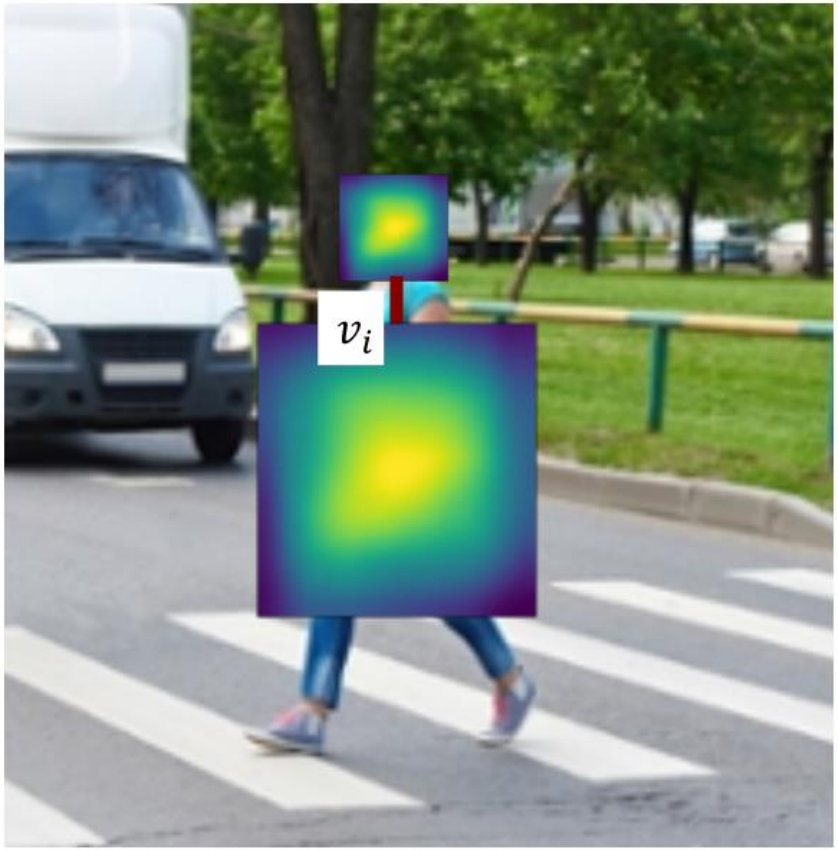
Apply the filters:

$$F_i \, \phi(p_i, H) \;, \; i = 1, \ldots, n$$





Accounting for Spatial cost with a Transformation

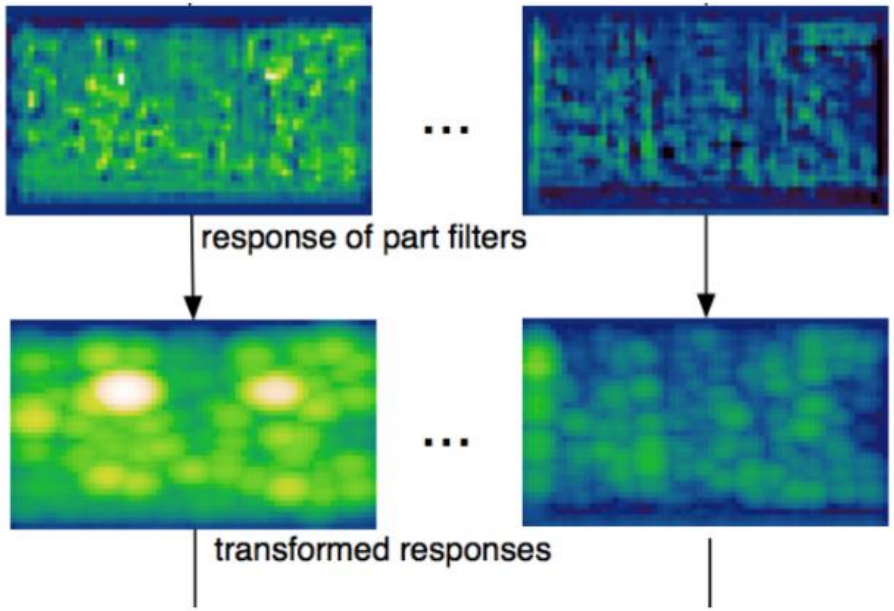


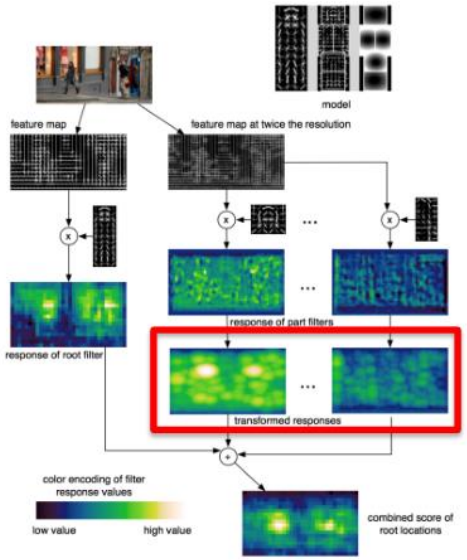
- Given the location for the detected head, we can guess where the body should be.
- The body should be in the direction calculated from the root person filter:v_i
- But we allow for some deformation or spatial shift on the location of the head with respect to the body: d_i
- We should ‘spread’ the head detection when calculating potential locations of the root!

Detection pipeline

Now apply the spatial costs for each part:

$$detection\ score = Fi\ \phi(p_i, H) - d_i\left(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2\right)$$

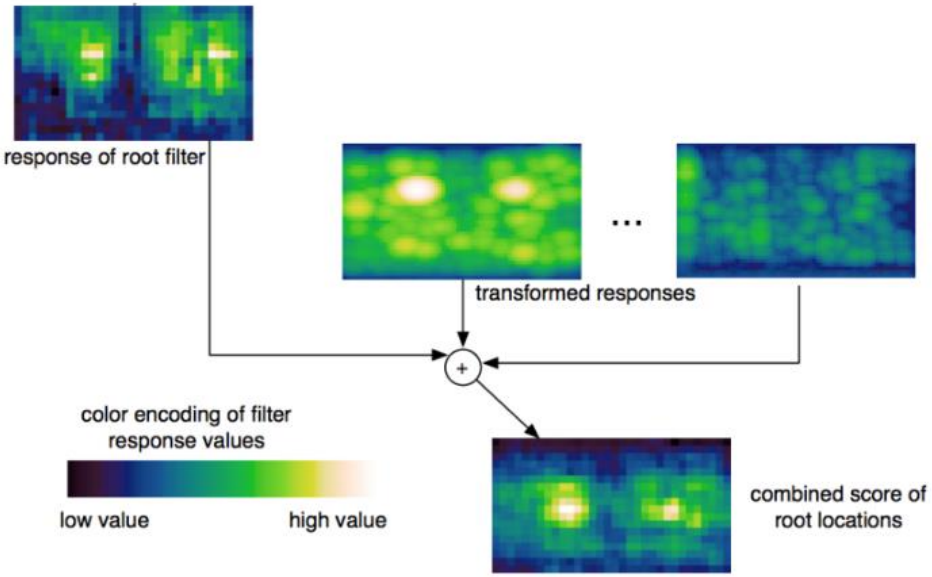


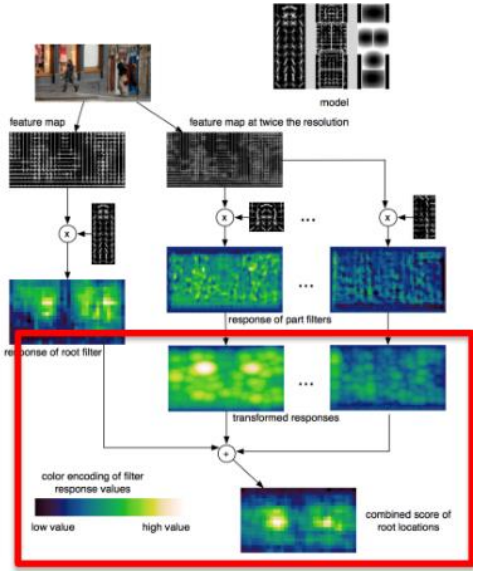


Detection pipeline

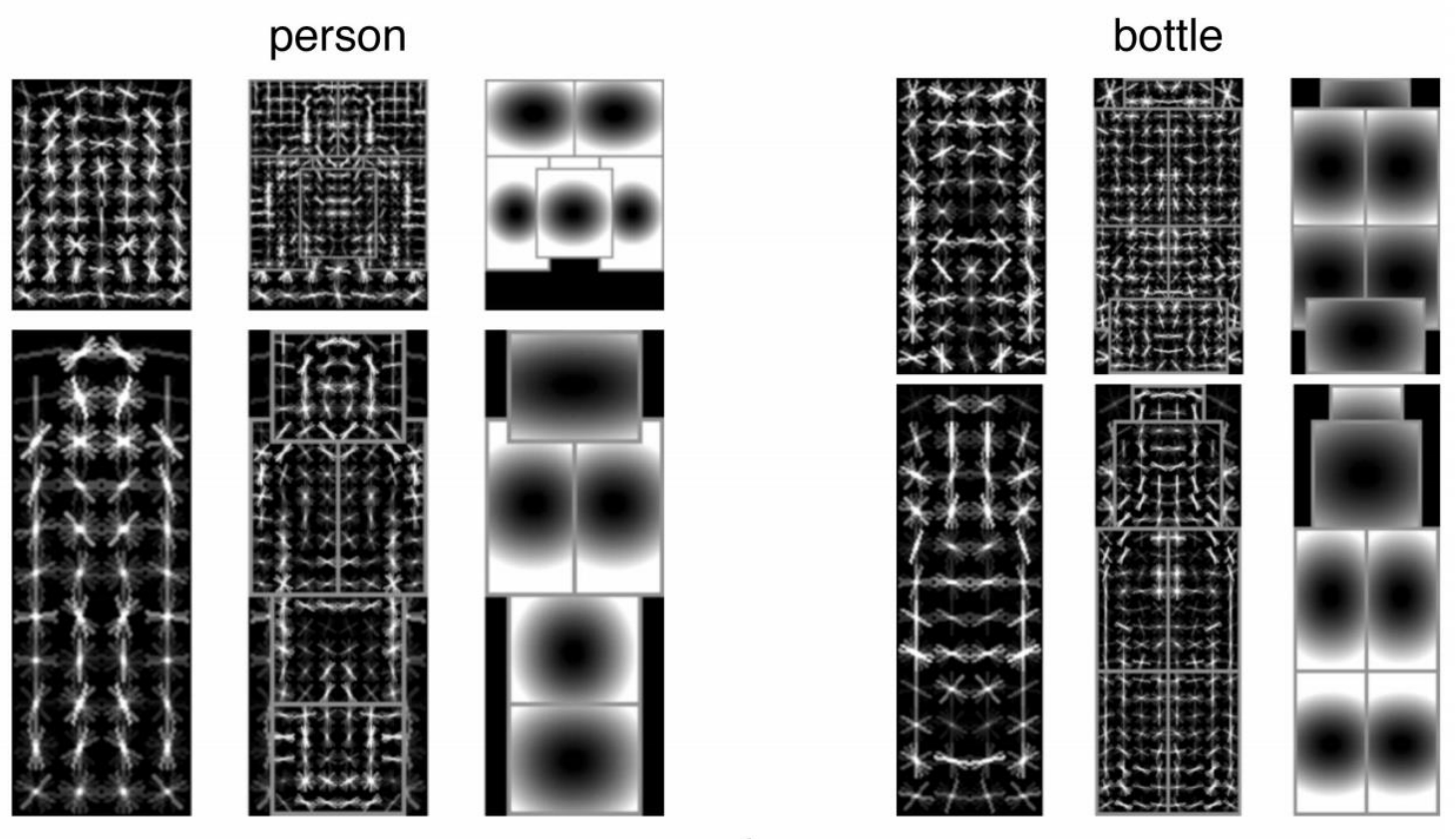
Now add the global filter:

$$detection\ score = F_0\phi(p_i, H) + \sum_{i=1}^{n} F_i\,\phi(p_i, H) - \sum_{i=1}^{n} d_i\left(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2\right)$$

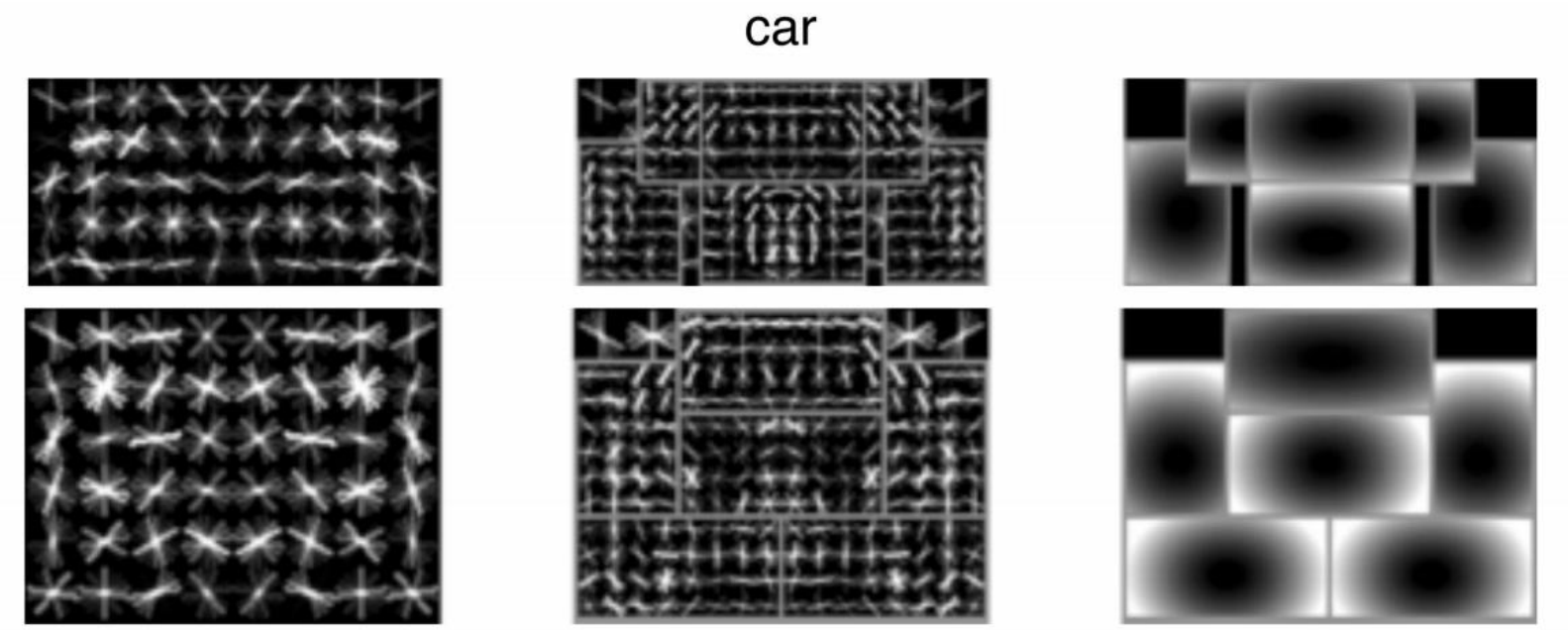




Some of the models

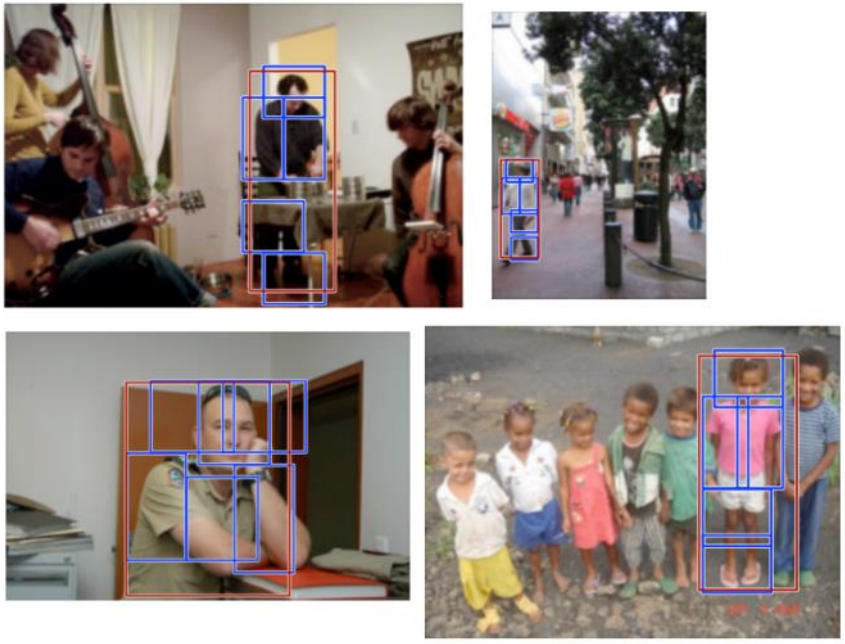


Some of the models

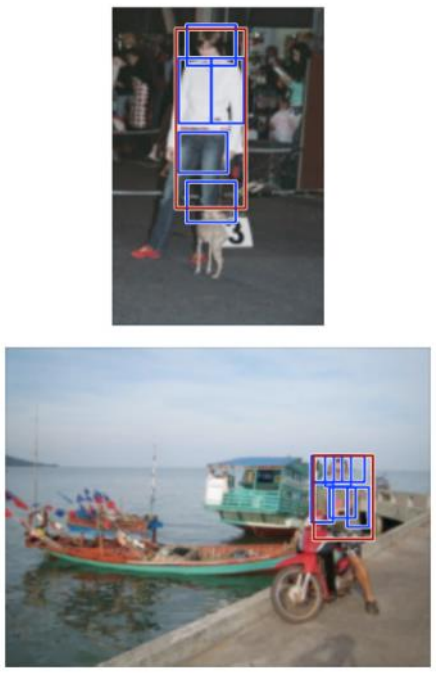


Results – person detection

high scoring true positives

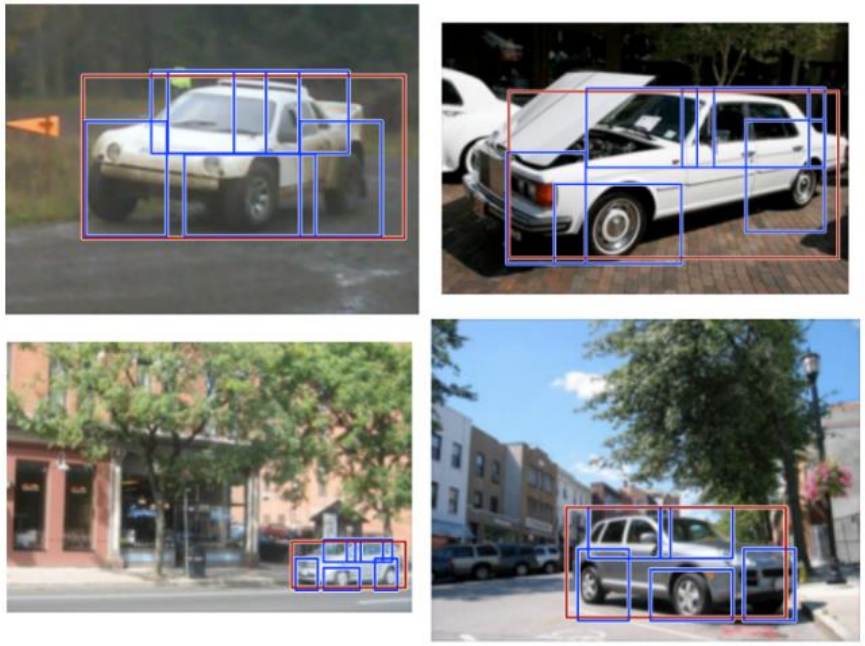


high scoring false positives
(not enough overlap)

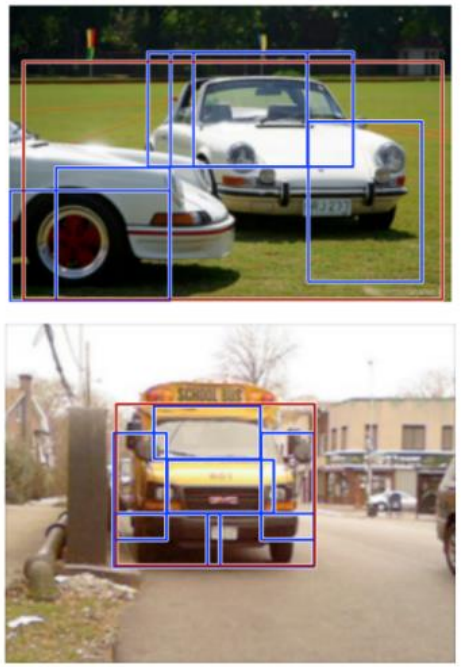


Results – car detection

high scoring true positives

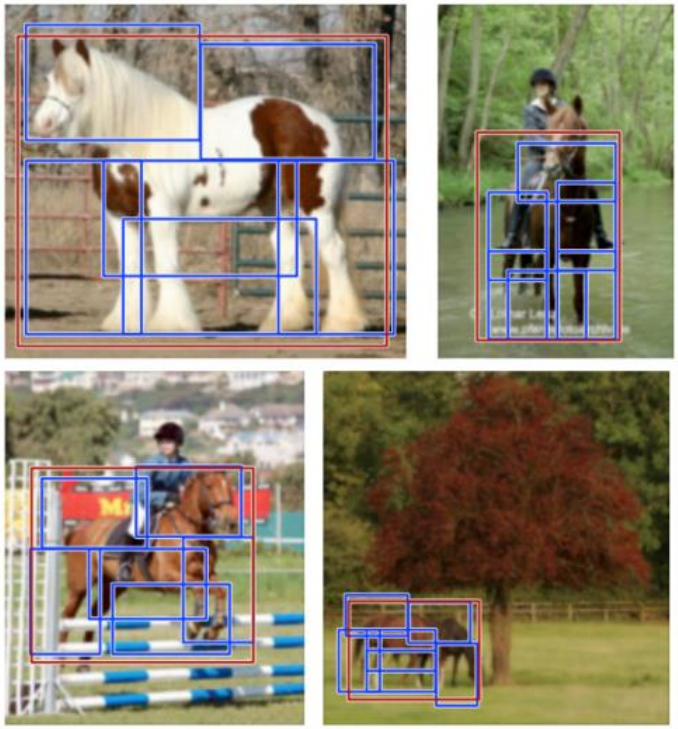


high scoring false positives

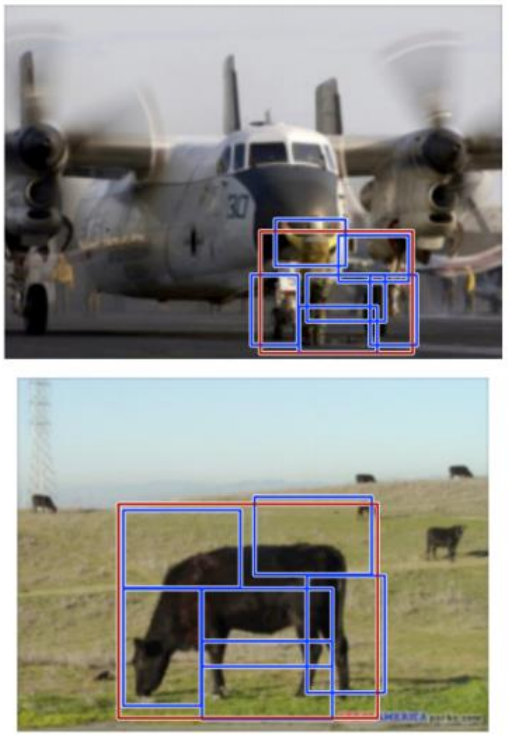


Results – horse detection

high scoring true positives



high scoring false positives



DPM - discussion

- Approach
  - Manually selected set of parts
  - Specific detector trained for each part
  - Spatial model trained on part activations
  - Evaluate joint likelihood of part activations
- Advantages
  - Parts have intuitive meaning
  - Standard detection approaches can be used for each part
  - Works well for specific categories
- Disadvantages
  - Parts need to be selected manually
  - Semantically motivated parts sometimes don't have a simple appearance distribution
  - No guarantee that some important part hasn't been missed
  - When switching to another category, the model has to be rebuilt from scratch

What we have learned today

- Object detection – Task and evaluation
- A simple detector
- Deformable parts model