Lecture 17

Object Detection

Administrative

A4 is due May 30

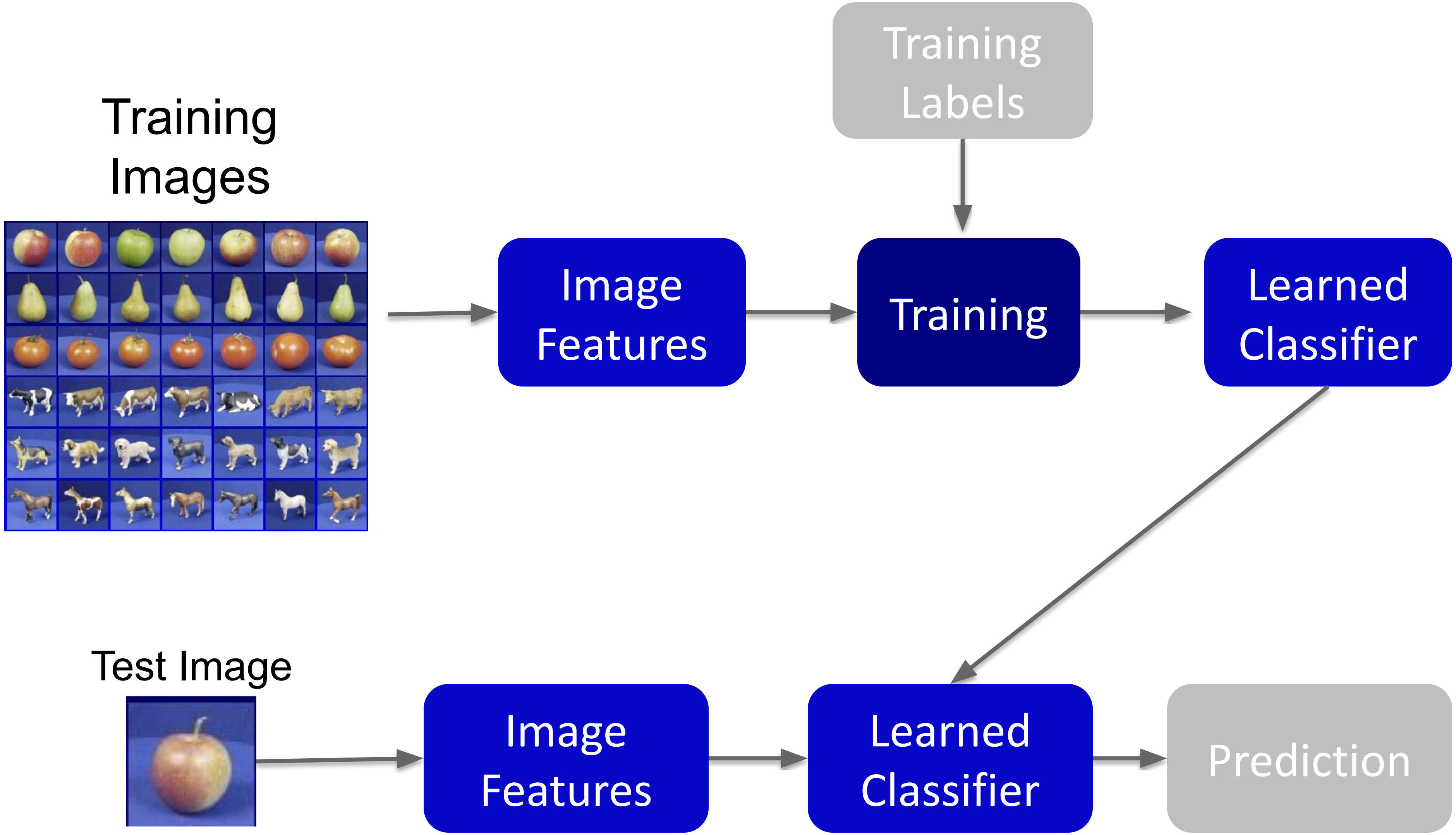
A5 (bonus A6) out next week
- Due Jun 10

Administrative

- Final Exam on 6/9 at 2:30 pm

- Makeup exam on 6/6
  - See EdStem for details

So far: A simple recognition pipeline

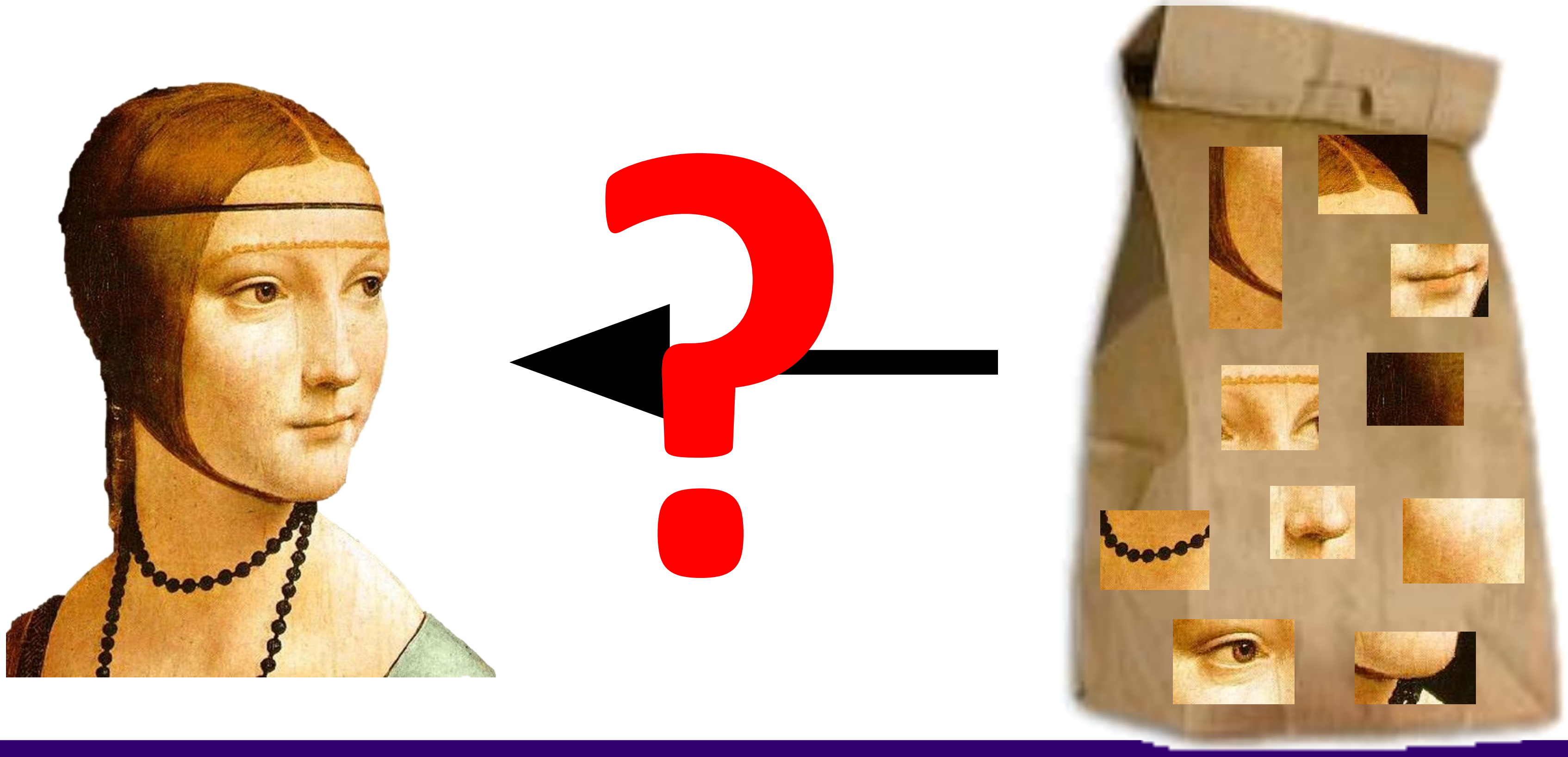


Today’s agenda

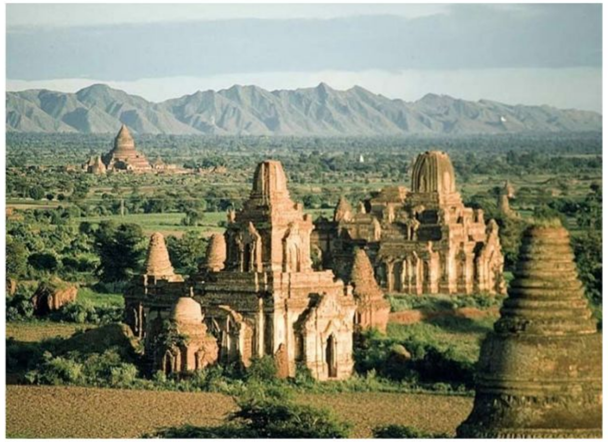
- Object detection
  - Task and evaluation
- A simple detector
- Deformable parts model

How do we choose the size of the patches?

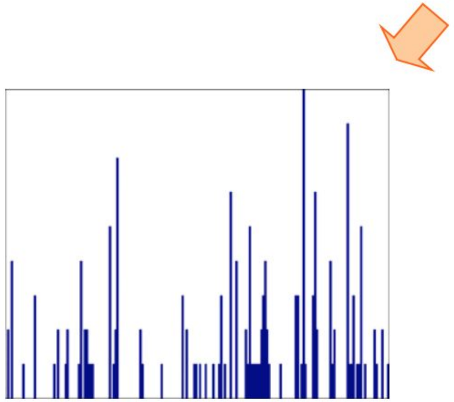
- If the object is close to the camera, larger patches are better
- If the object is really far away, smaller patches are better for finding it.



Bag of words + pyramids

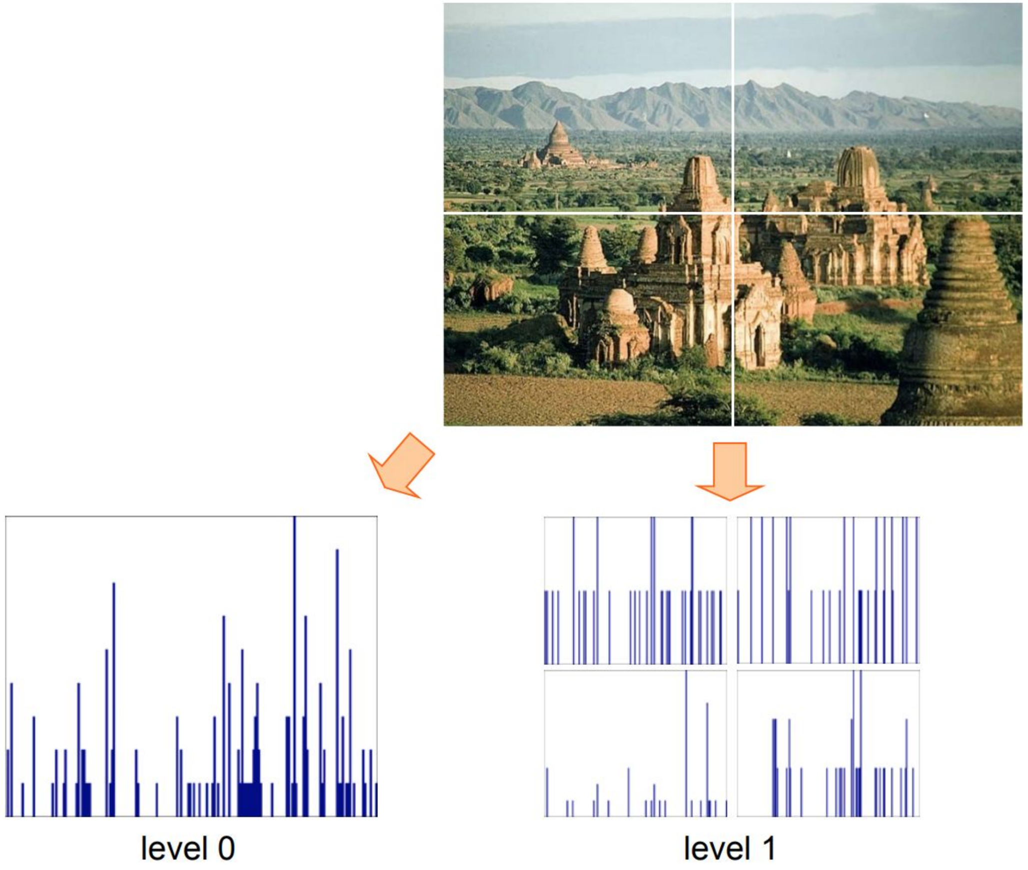


Locally orderless representation at several levels of spatial resolution



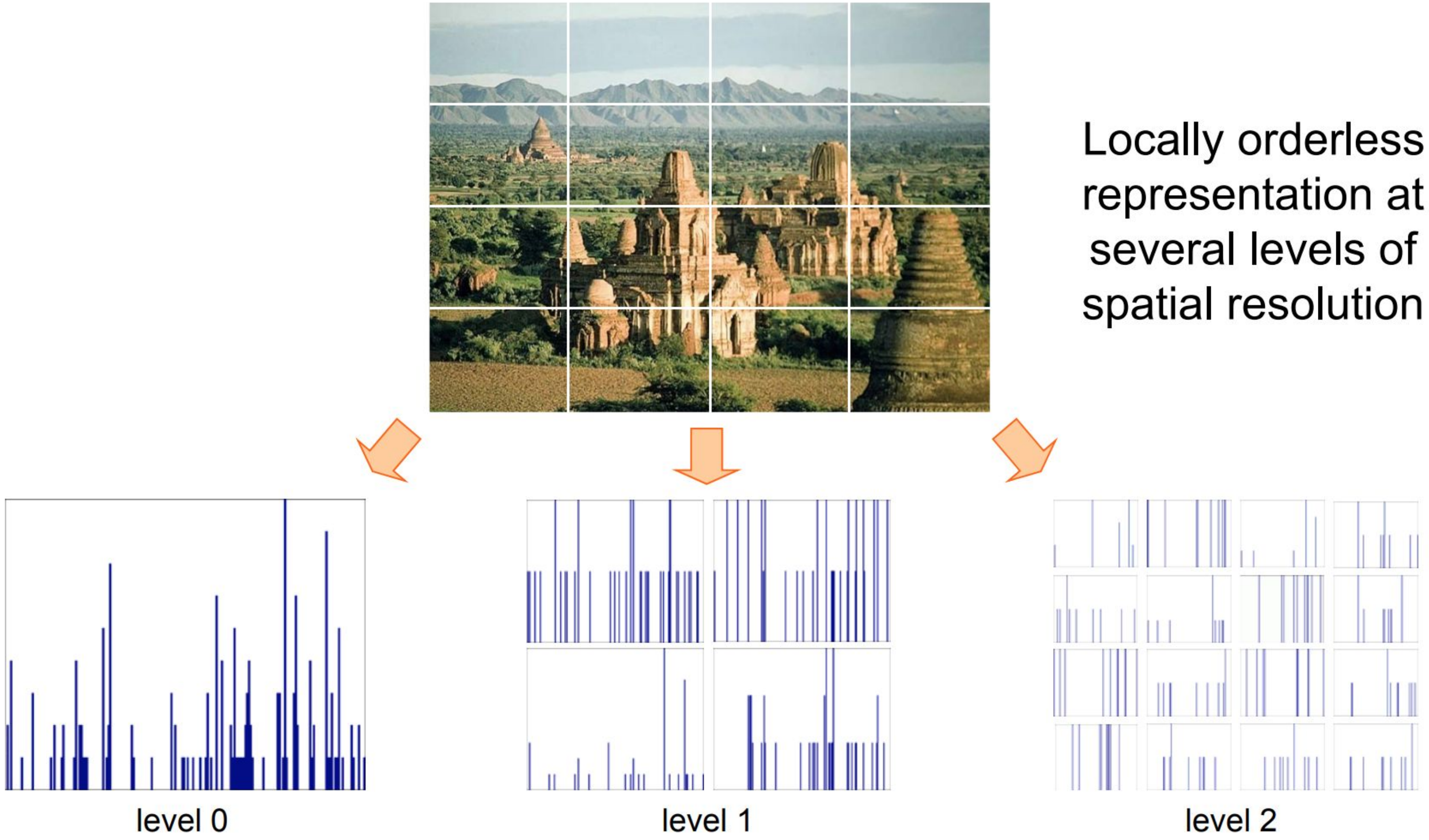
level 0

Bag of words + pyramids



Locally orderless representation at several levels of spatial resolution

Bag of words + pyramids



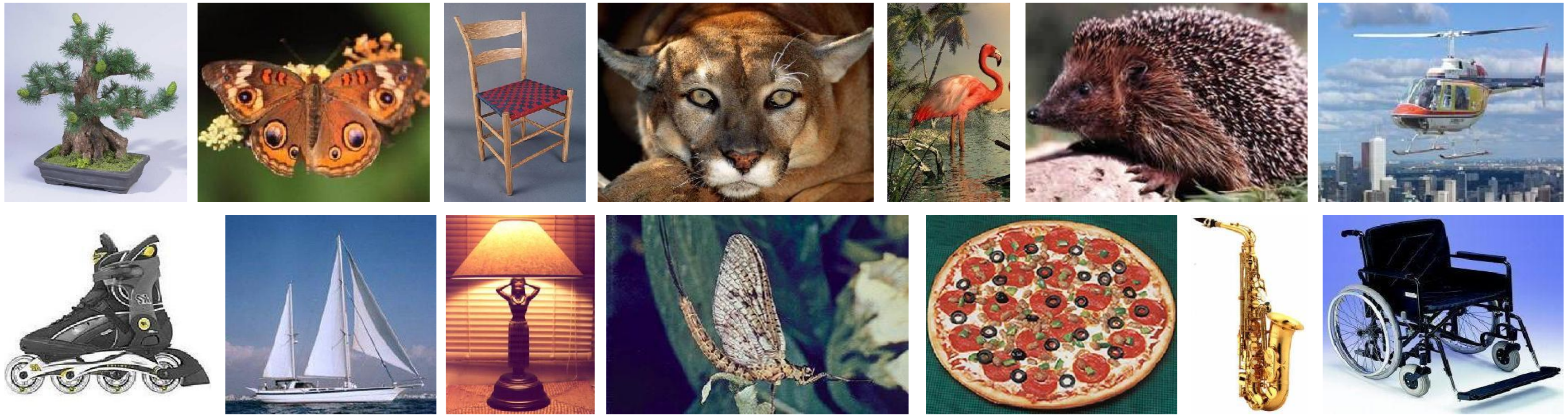
Locally orderless representation at several levels of spatial resolution

Pyramids are a general idea that is used in all vision models today (including swin transformers)

- Very useful for representing images.
- Pyramid is built by using multiple copies of image.
- Each level in the pyramid is 1/4 of the size of previous level.

Caltech101 dataset

Multi-class classification results (30 training images per class)



Level	Single-level	Pyramid	Single-level	Pyramid
0	15.5 ±0.9		41.2 ±1.2	
1	31.4 ±1.2	32.8 ±1.3	55.9 ±0.9	57.0 ±0.8
2	47.2 ±1.1	49.3 ±1.4	63.6 ±0.9	**64.6** ±0.8
3	52.2 ±0.8	**54.0** ±1.1	60.3 ±0.9	64.6 ±0.7

Today's agenda

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

Object Detection

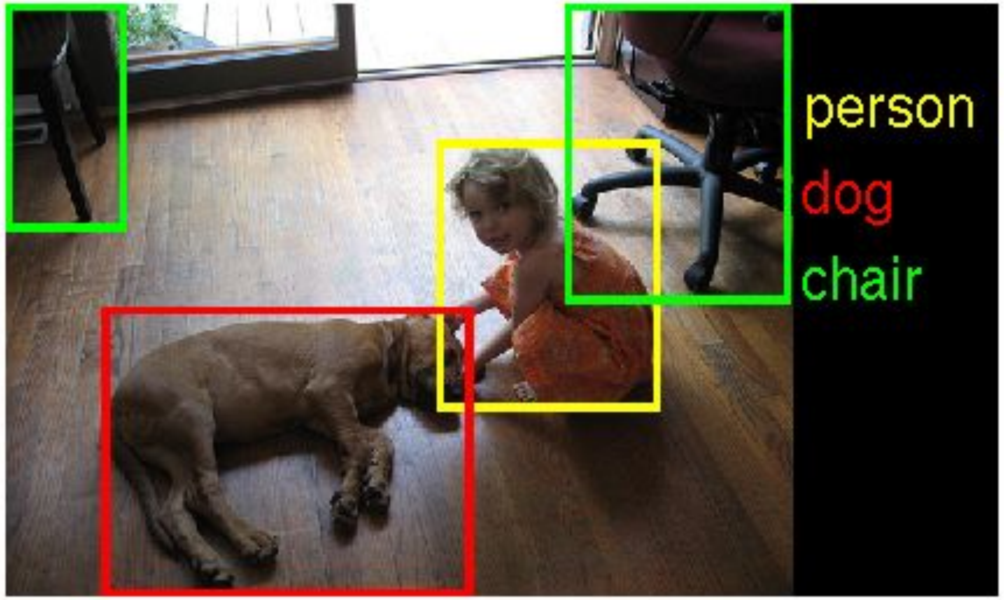
Credit: Flickr user neilalderney123

- What do you see in the image?

Object Detection

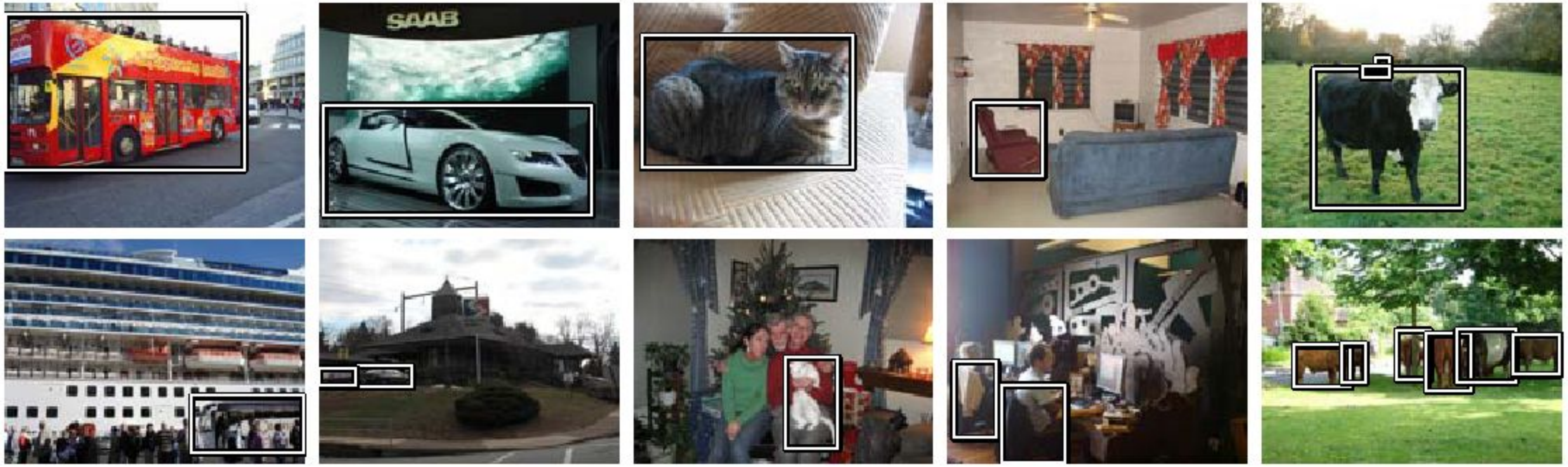
- **Problem**: Detecting and localizing objects from various categories, such as cars, people, etc.

- Challenges:
  - Illumination,
  - viewpoint,
  - deformations,
  - Intra-class variability



Object Detection Benchmarks

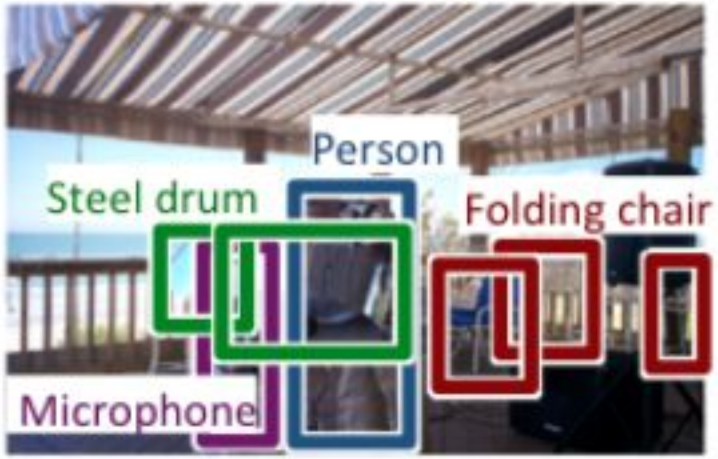
- PASCAL VOC Challenge



- 20 categories
- Annual classification, detection, segmentation, … challenges

Object Detection Benchmarks

- PASCAL VOC Challenge
- ImageNet Large Scale Visual Recognition Challenge (ILSVRC)
  - 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

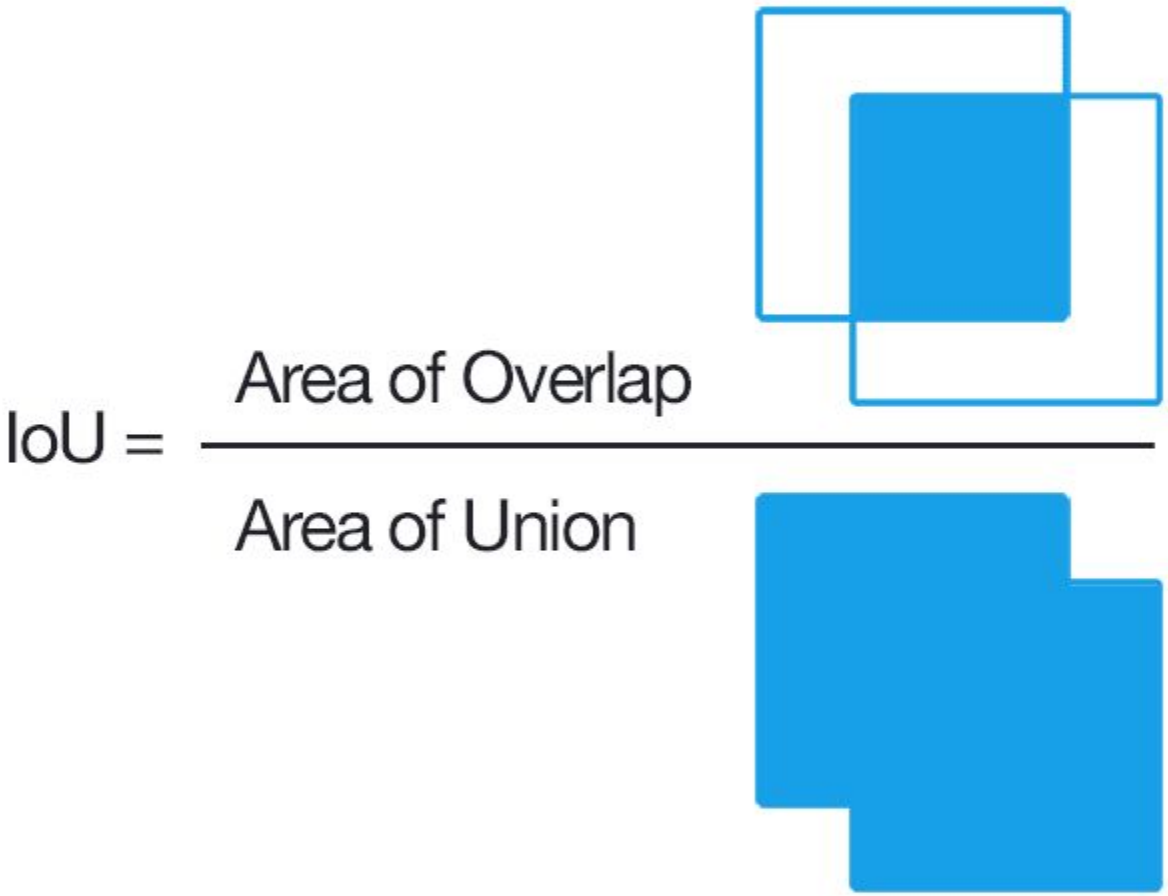
ground truth

Defining what is a good versus bad detection

IoU is a metric used to decide good from bad predictions.

Given a predicted box and and ground truth box:

IoU = intersection between the two boxes over (divided by) the union of the two



Defining what is a good versus bad detection

We say a prediction was good if it has IoU > 0.5 with any of the ground truth boxes

0.5 is a threshold that is generally accepted as a good heuristic.

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?

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

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?

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

- Precision:
  - how many of the predicted detections are correct?

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

- Recall:
  - how many of the ground truth objects are detected?

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

How do we evaluate object detection?

predictions
ground truth

True positive: 1
False positive: 2
False negative: 1

Q. What is the precision?

How do we evaluate object detection?

predictions

ground truth

True positive: 1
False positive: 2
False negative: 1

Q. What is the precision? 1/3

How do we evaluate object detection?

predictions

ground truth

True positive: 1
False positive: 2
False negative: 1

Q. What is the precision? 1/3

Q. What is the recall?

How do we evaluate object detection?

predictions

ground truth

True positive: 1
False positive: 2
False negative: 1

Q. What is the precision? 1/3

Q. What is the recall? 1/2

In reality, our model makes a lot of predictions with varying scores between 0 and 1

predictions

ground truth

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

From this, we see that:
- 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 using a threshold of 0.5

Q. Is precision high or low if threshold is high?

How do we evaluate object detection?

predictions

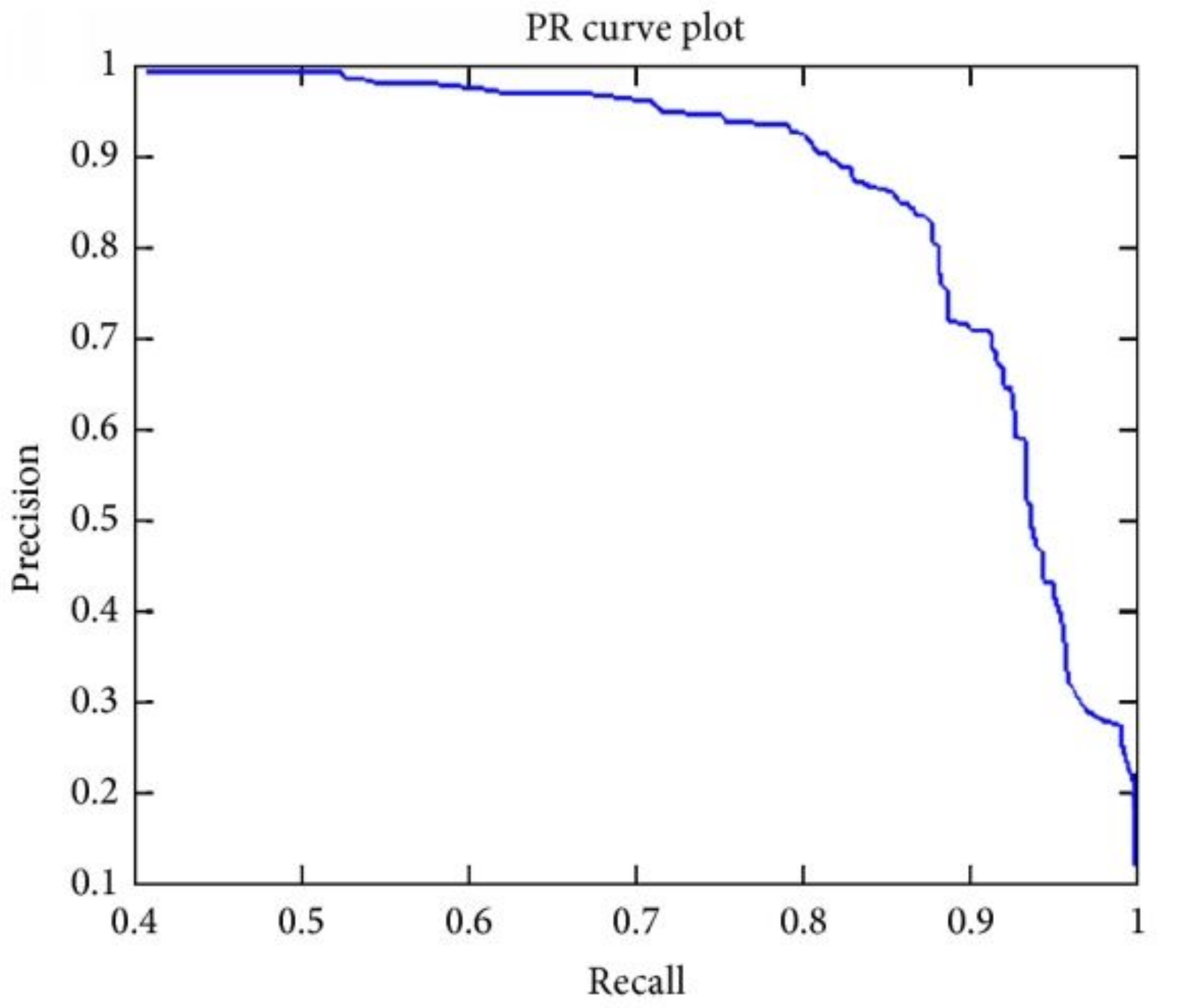
ground truth

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

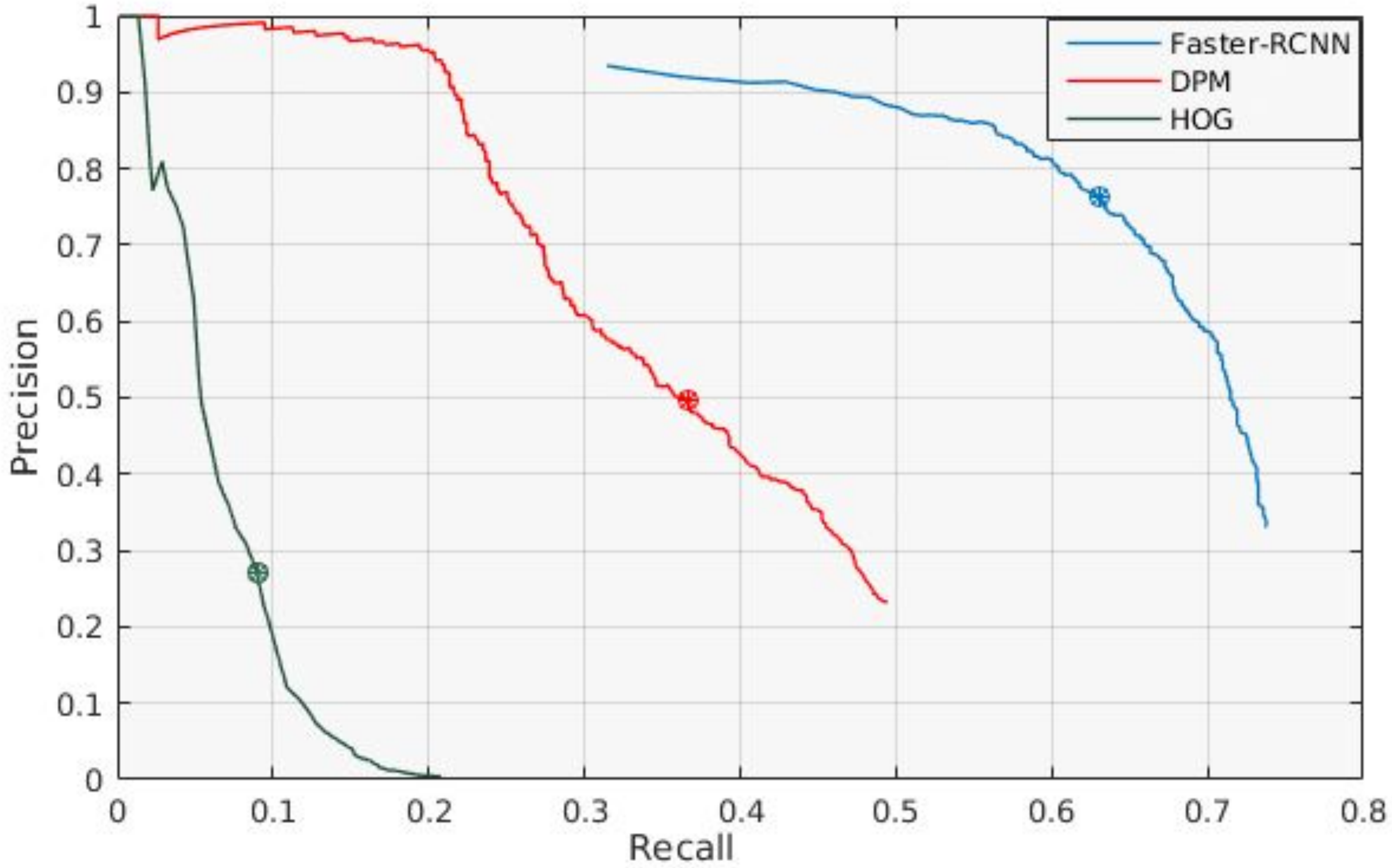
We are using a threshold of 0.5

Q. What happens to recall if threshold is high?

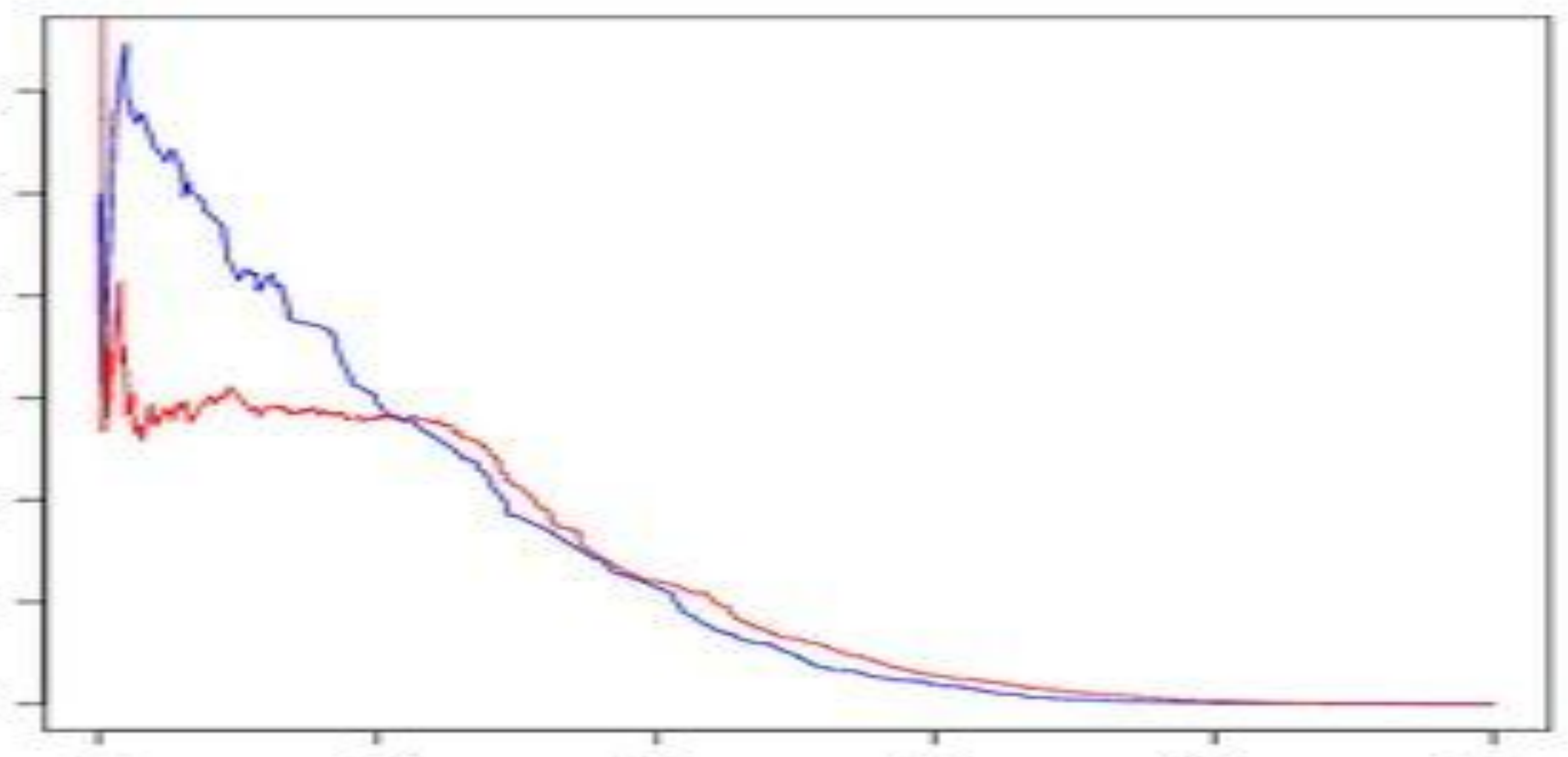
Precision – recall curve (PR curve)



Which model is the best?



Which model is the best?



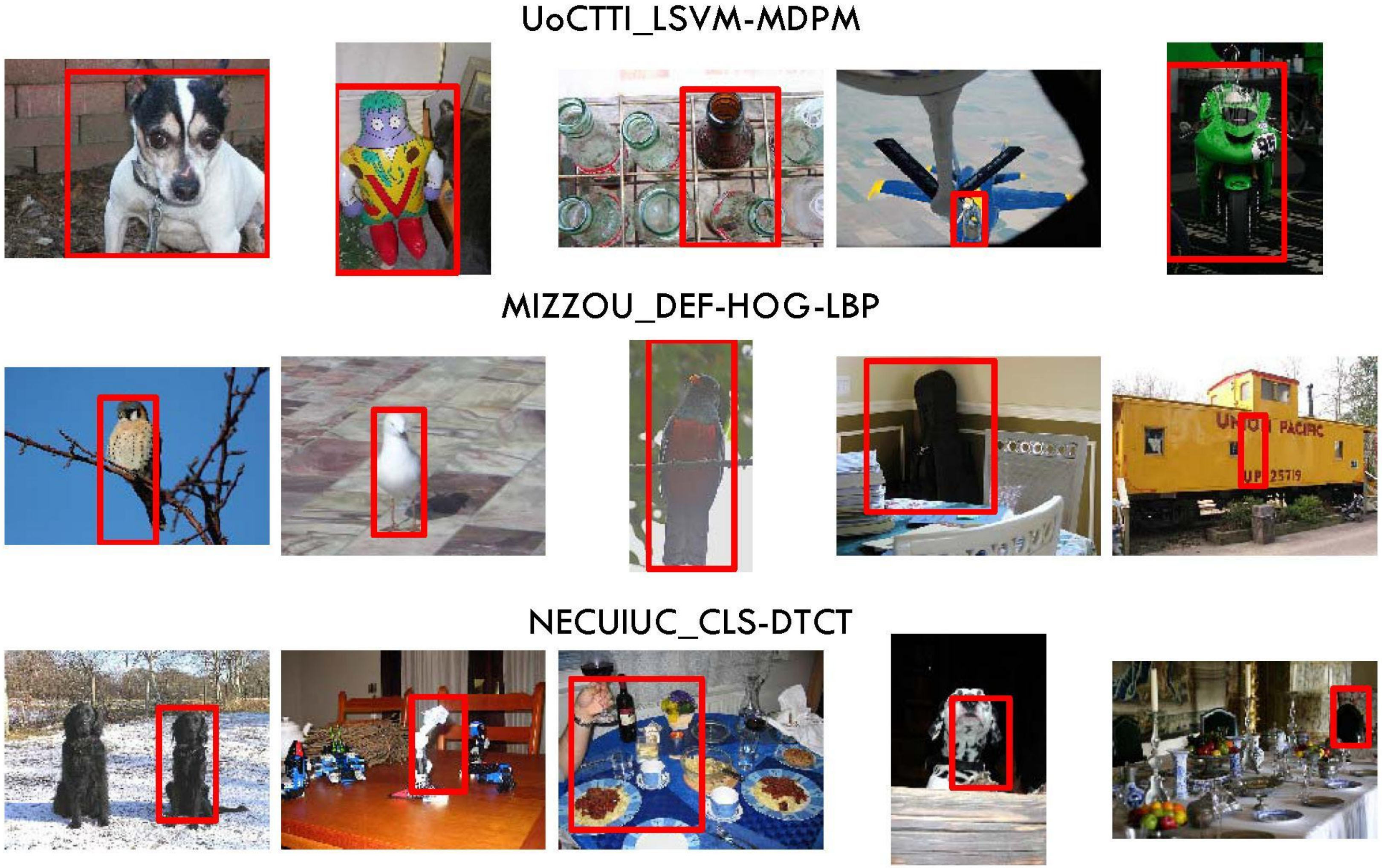
True positives - detecting person

UoCTTI_LSVM-MDPM

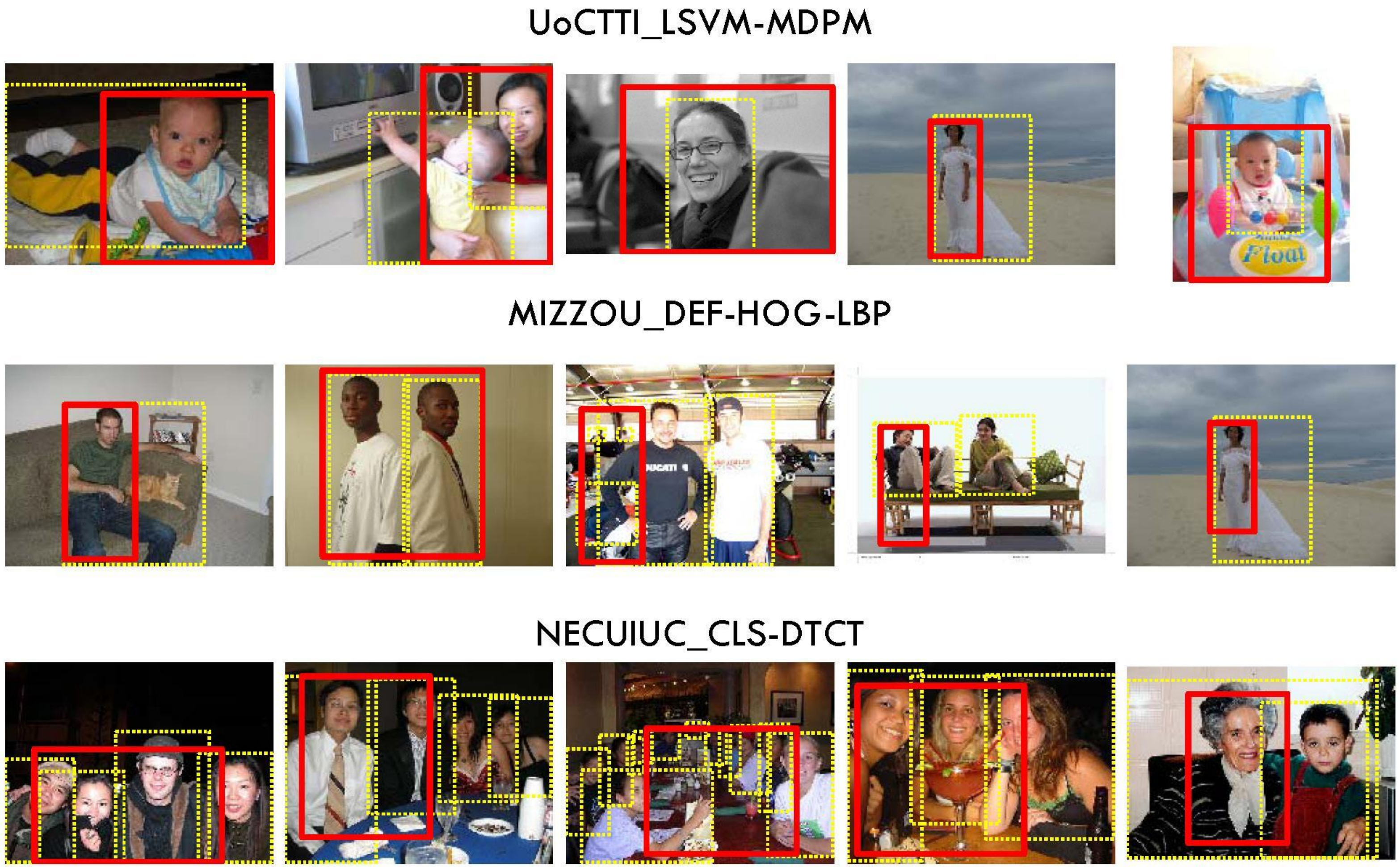
MIZZOU_DEF-HOG-LBP

NECUIUC_CLS-DTCT

False positives - detecting person



Near misses: IoU falls short of 0.5



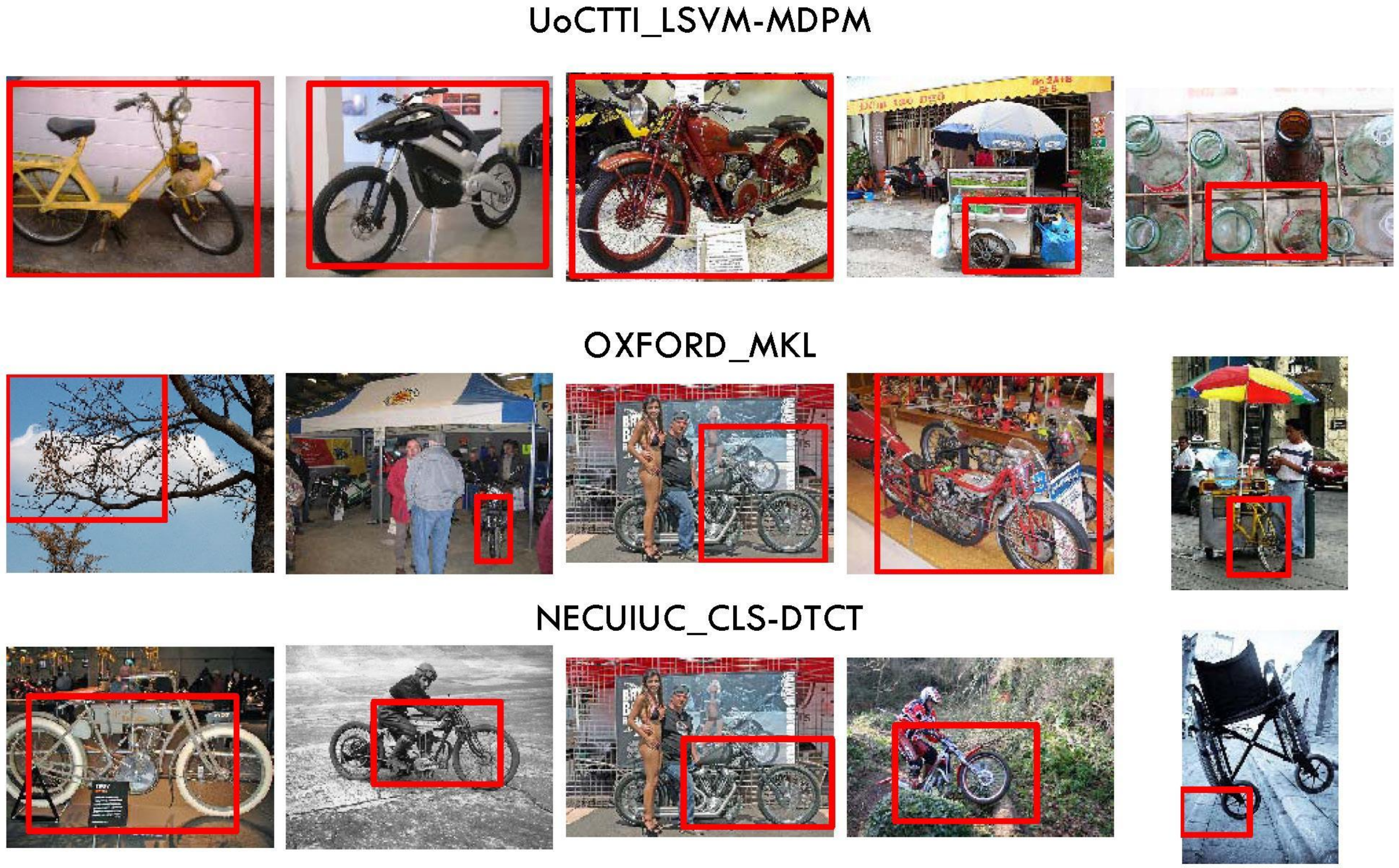
True positives - detecting bicycle

UoCTTI_LSVM-MDPM

OXFORD_MKL

NECUIUC_CLS-DTCT

False positives - detecting bicycle



Today's agenda

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

Dalal-Triggs method

Sliding window (Convolution)

At every patch as the window slides

1. Convert the image patch into your favorite feature representation
   a. For example:
      i. HoG,
      ii. HoG with PCA,
      iii. Bag of words on RGB
      iv. etc.
2. Use a trained classifier to determine if it is a specific class
   a. e.g. kNN classifier
3. Accumulate the predictions over all the patches

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

No bus found

- But what if we were looking for buses?

Sliding window + hog features

No bus found

- 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 windows with pyramids

Computationally, we first resize the image to different sizes and then extract features at each size.

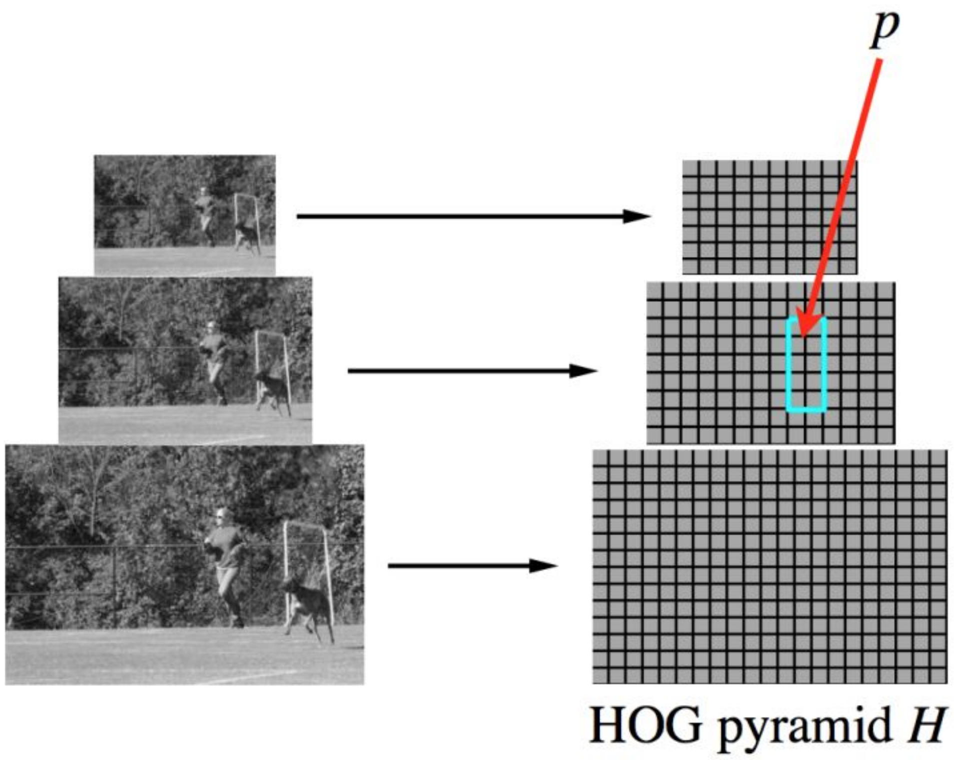


Image Pyramid:
An important idea even as of today!!

Today’s agenda

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

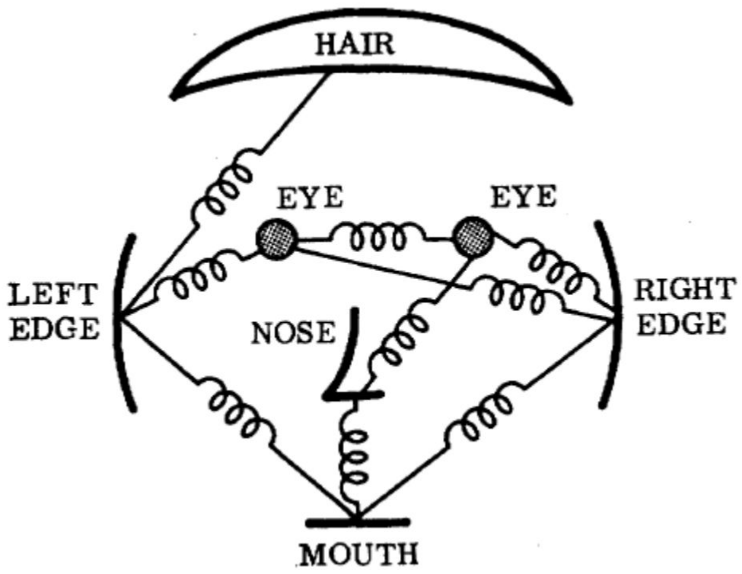
Recap – bag of words

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

- Can we use the location of these patches to find objects within those images?

Deformable Parts Model

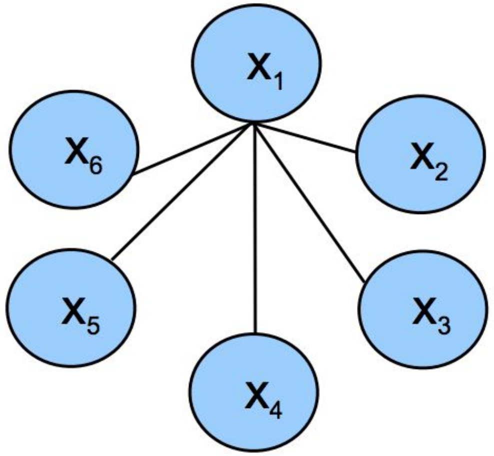
- Represents an object as a "collection of parts"
- Each part represents local appearances
- Make prediction jointly



Fischler and Elschlager, Pictoral Structures, 1973

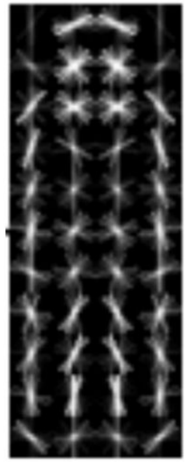
Detecting a person with their parts

- Star model: every part is defined relative to a root.
- 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

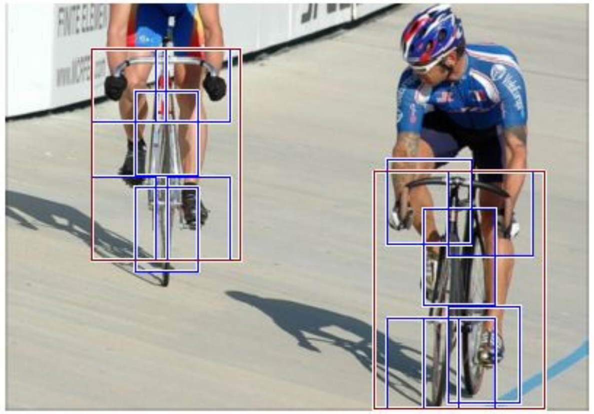
- Each model will have a global model. And a set of part models. Here is an example of a global person HoG filter with it's 'head' part filter:

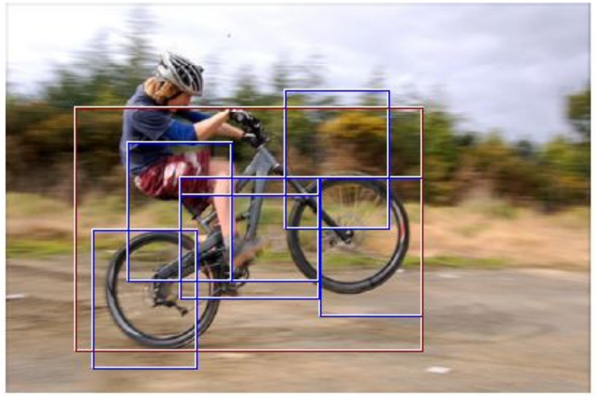


Global/root filter

Part filter

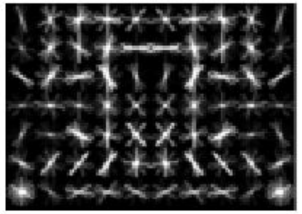
5-part bicycle model



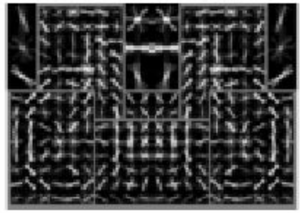


"side view" bike
model component

Root filter

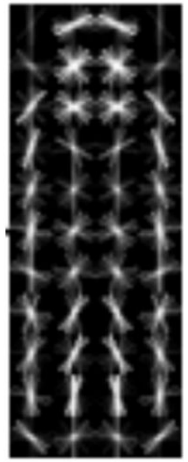


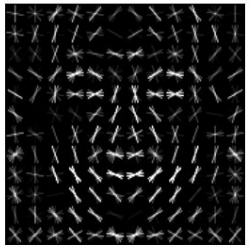
Part filters



Deformable parts model

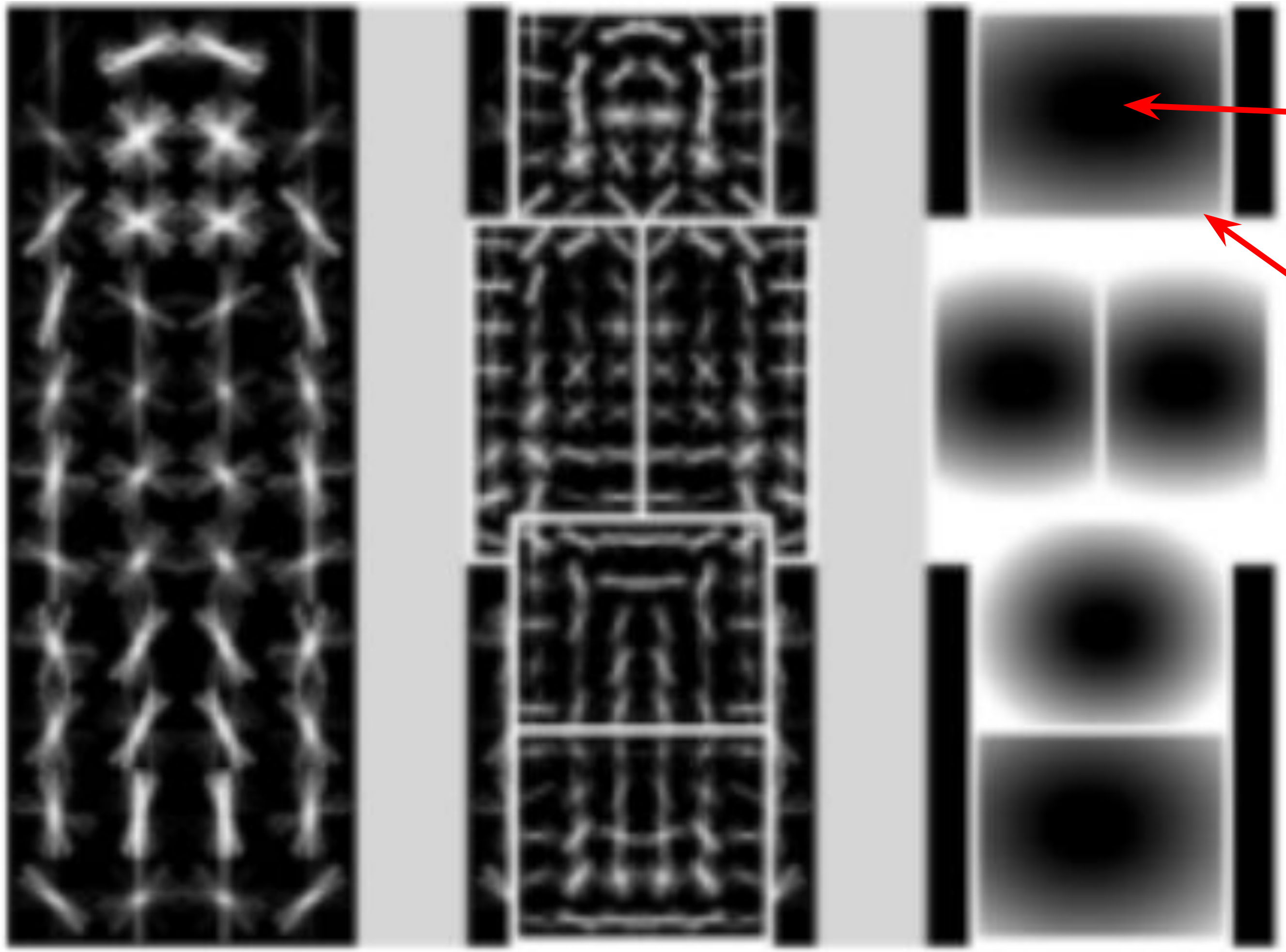
- Mixture of deformable part models





- Each component has global component + deformable parts
- Part filters have finer details

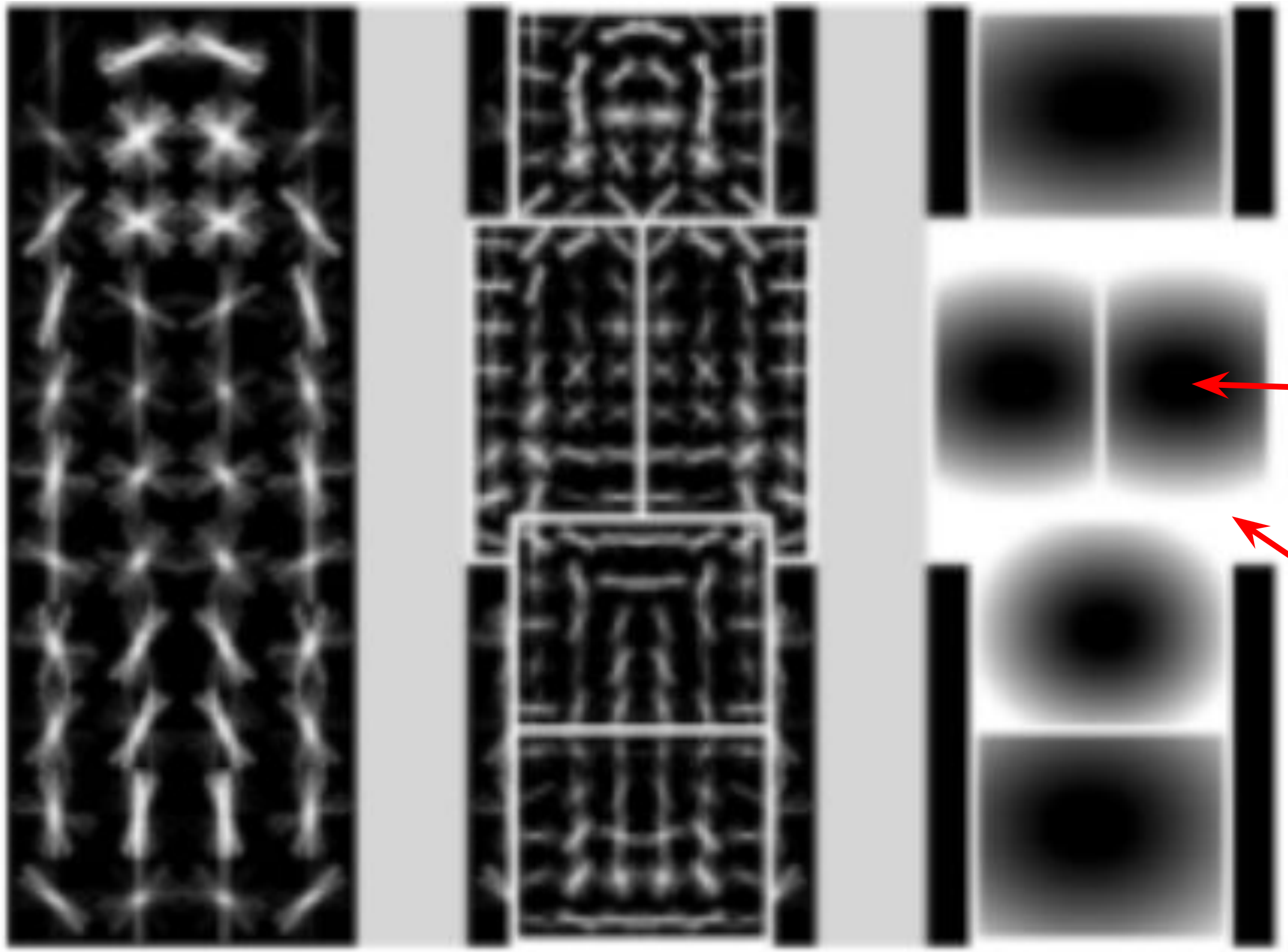
DPM for person model with 5 parts



If the head is here,
the score is high

If the head is here,
the score is low

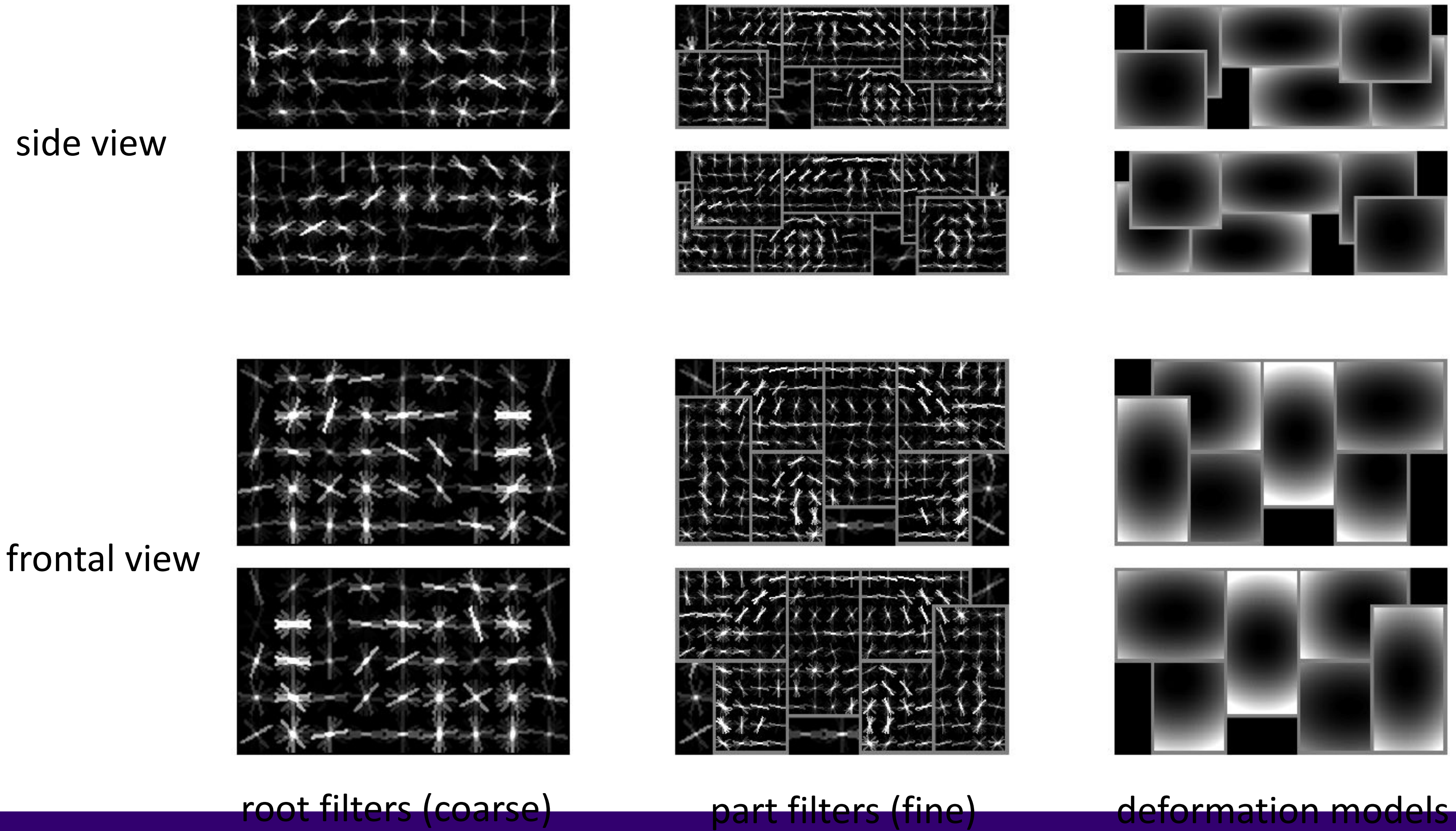
DPM for person model with 5 parts



If the arm is here, the score is high

If the arm is here, the score is low

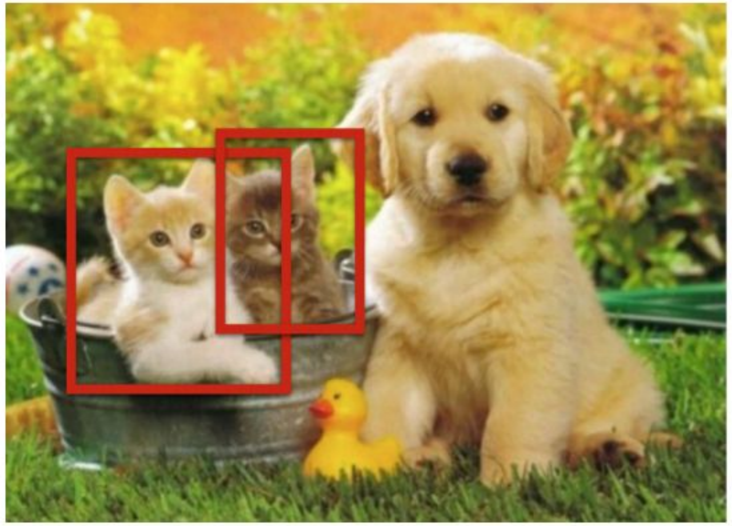
DPM for car with 6 parts



How do we use the parts to make a detection?

Intuition:

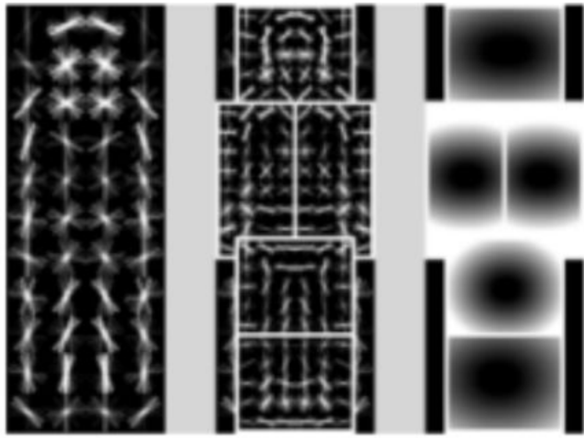
1. First, use the sliding windows at different pyramid scales to detect each part (and the root).
2. Each part gives you a score for where the person might be
3. Accumulate the global and part scores (and penalize the deformation of the parts.)



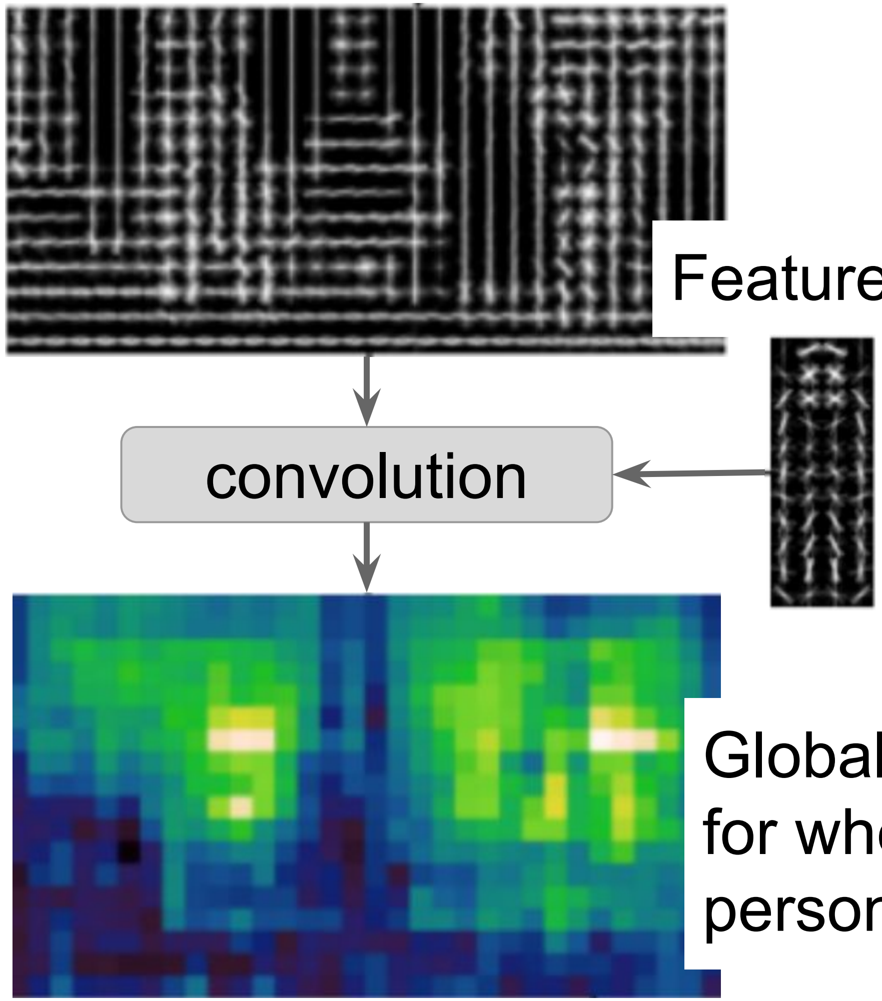
Example for detecting people

Image input

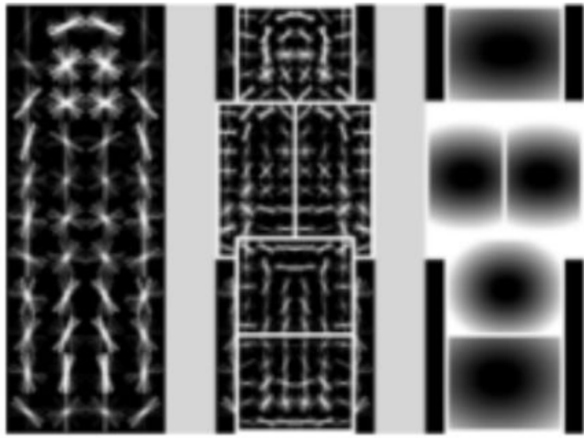
A feature template for person



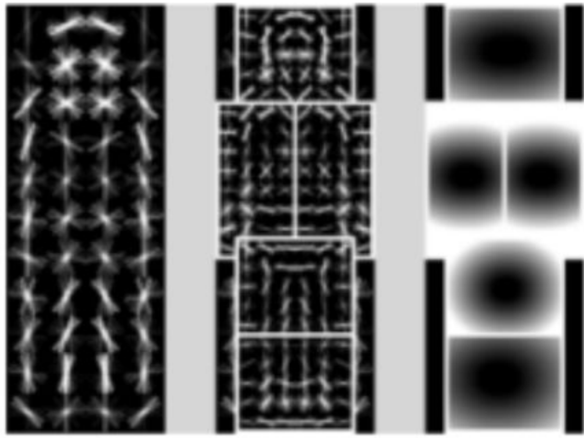
Calculate scores for global template

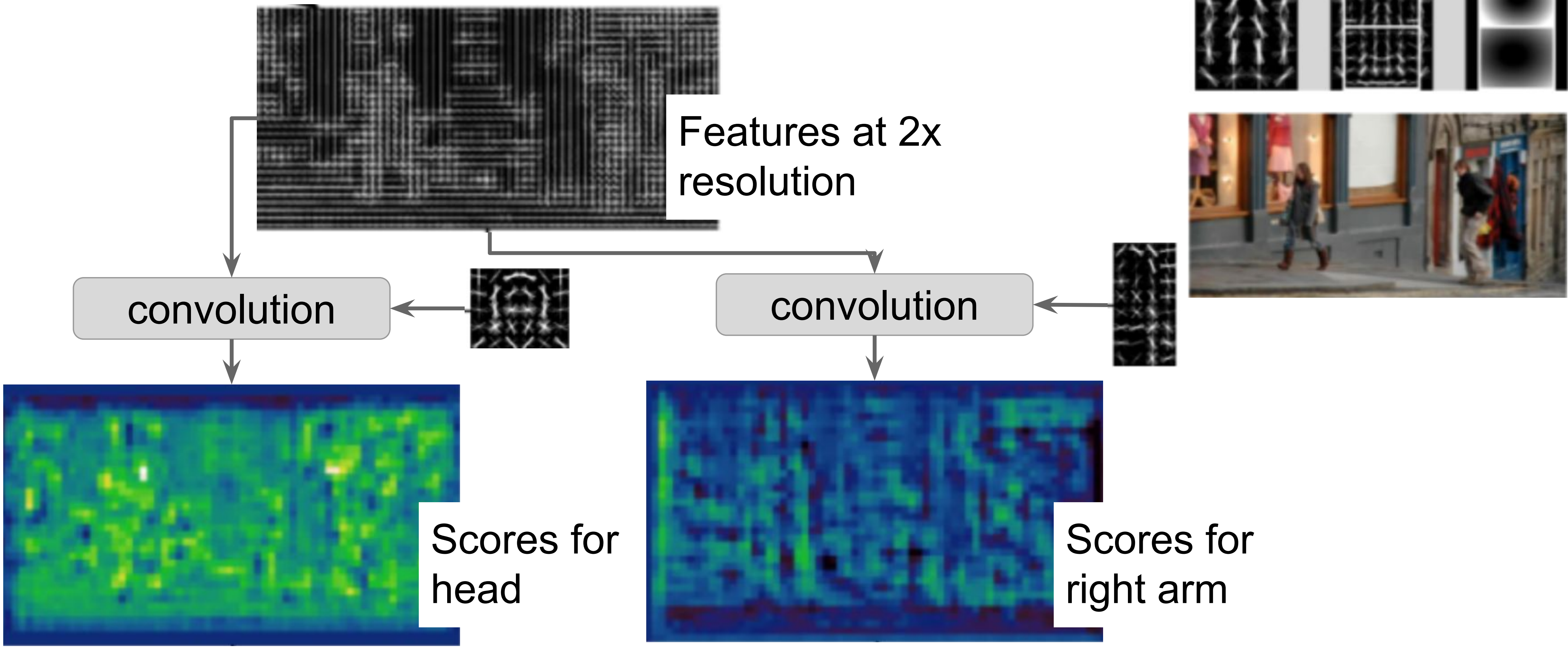


Global scores for where a person might be



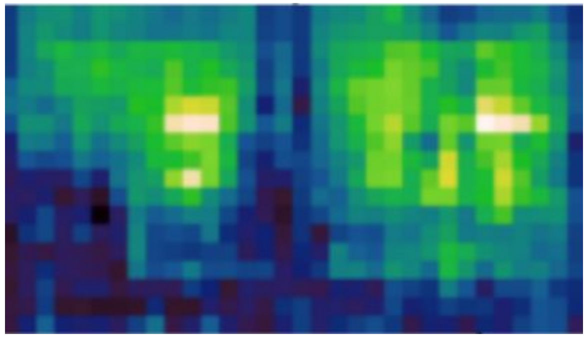
Calculate scores for part templates



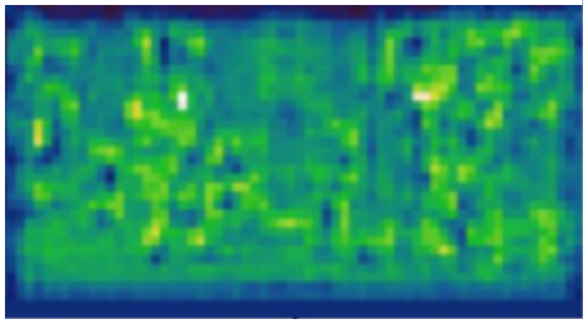


After step 1, we have scores for all parts and global template

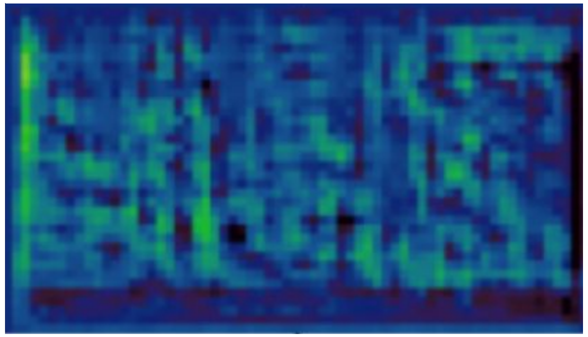
Global scores

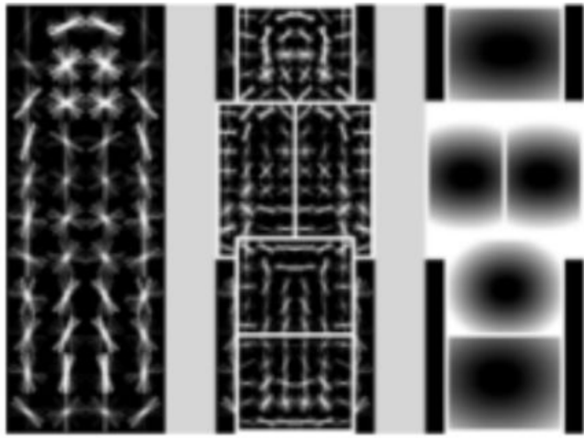


Head scores

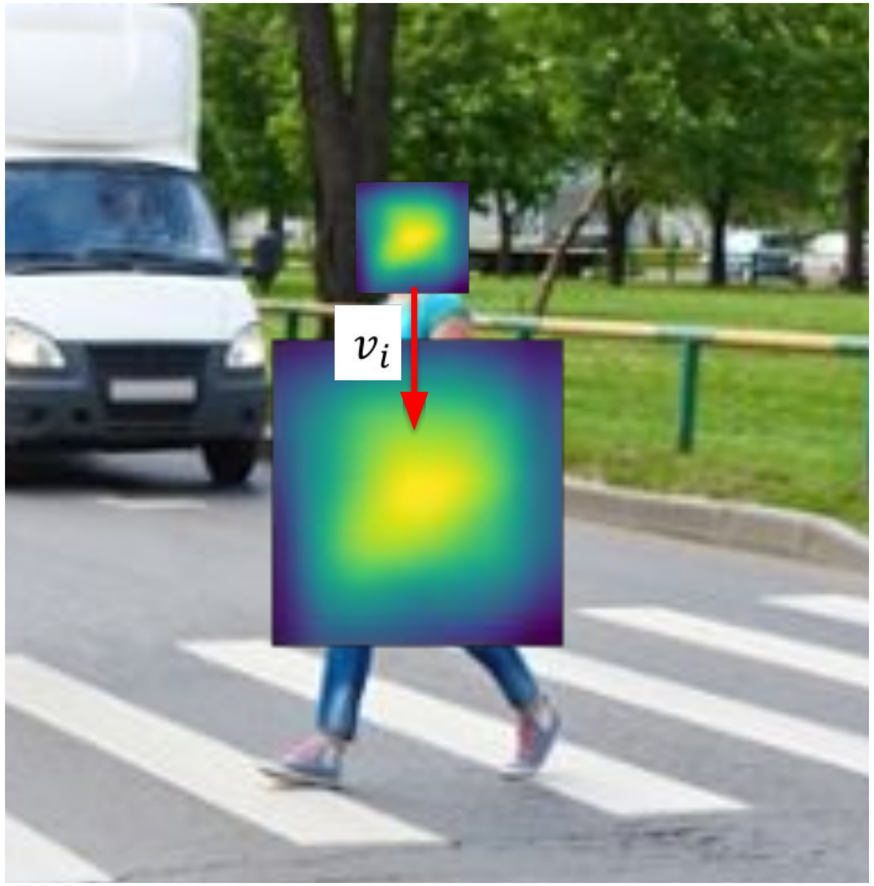


Right arm scores



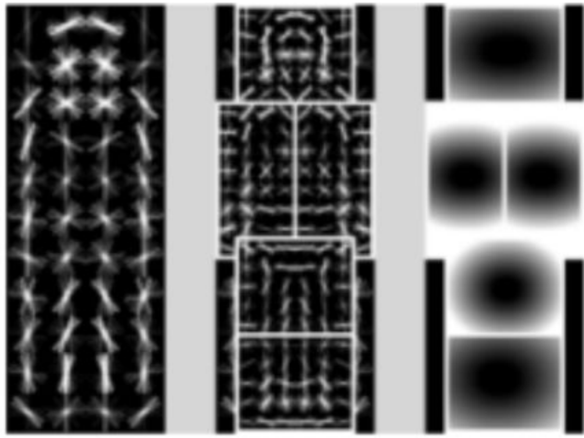


Allowing each part to deform and guess where the entire body is.

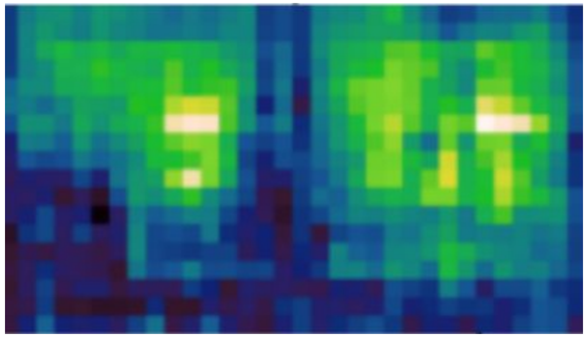


- Given the location for the detected head, we can guess where the body should be.
- The body should be in the direction (v_i) predefined in the model
- Bodies can be of different sizes and shapes. So we allow it to deform by some variable d_i
- This deformation spreads the scores to potential locations of the body

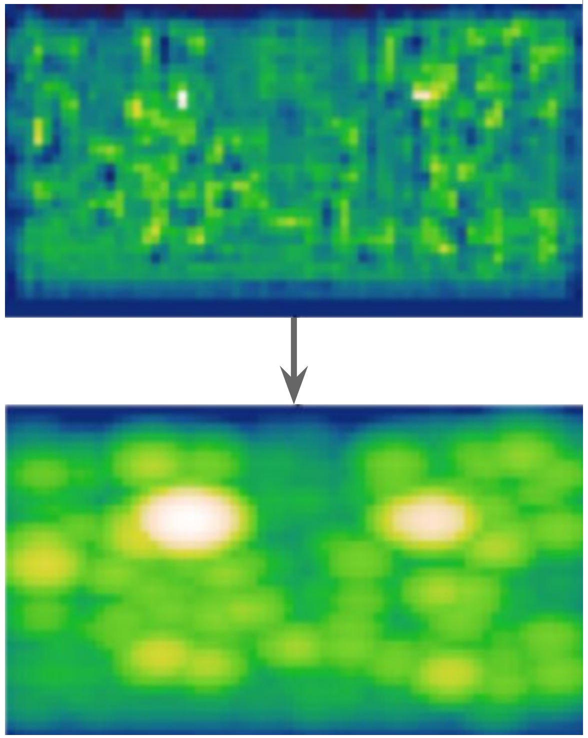
Step 2: each part gives you a score for where the person might be



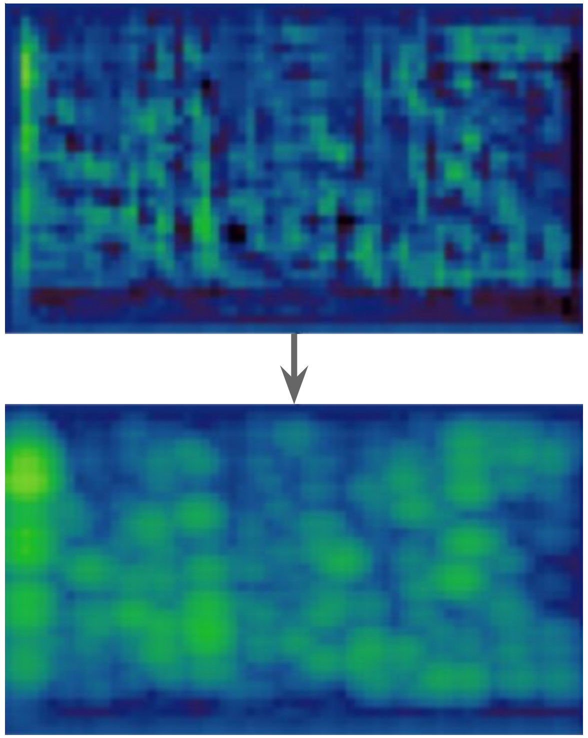
Global scores



Scores for head



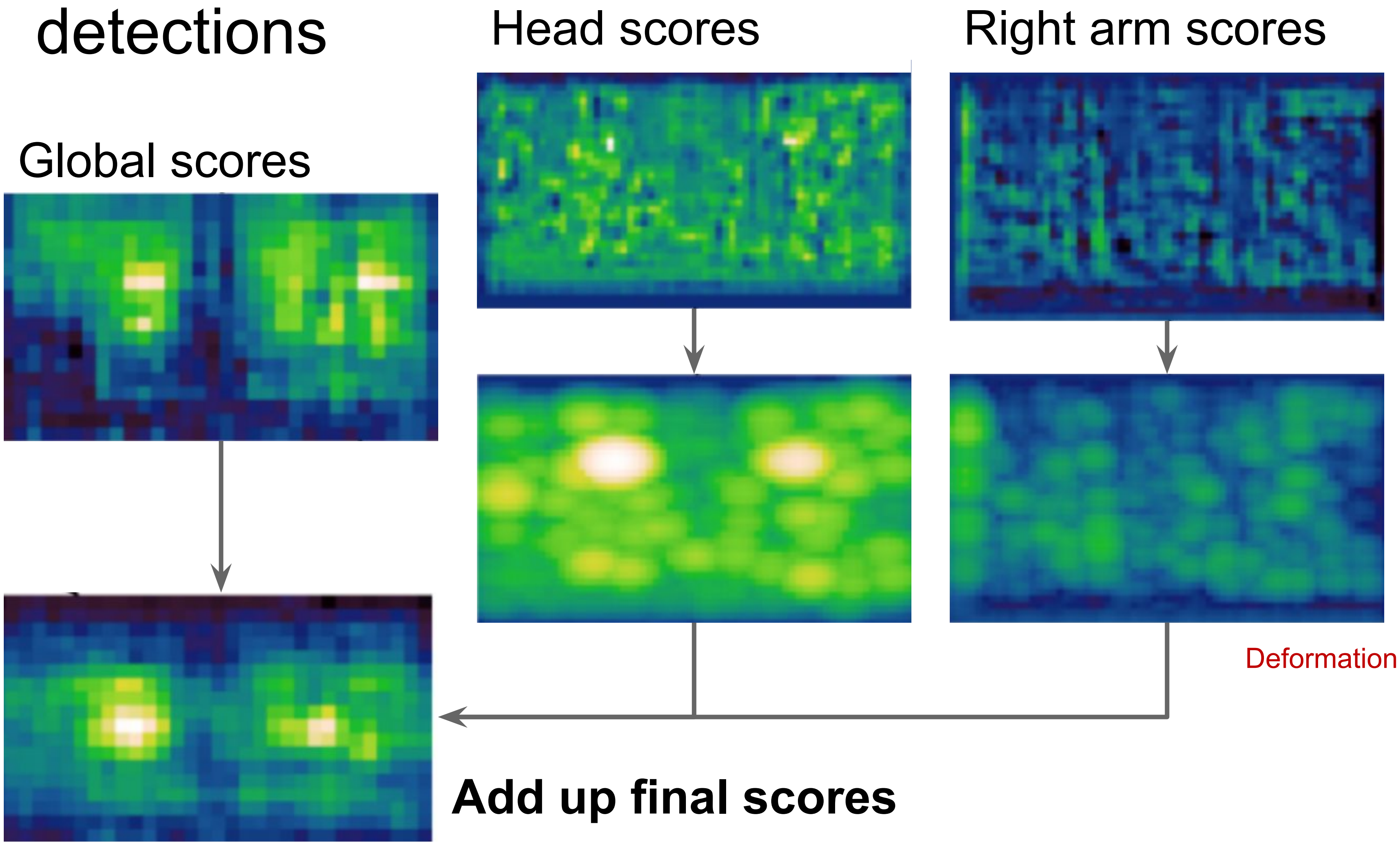
Scores for right arm

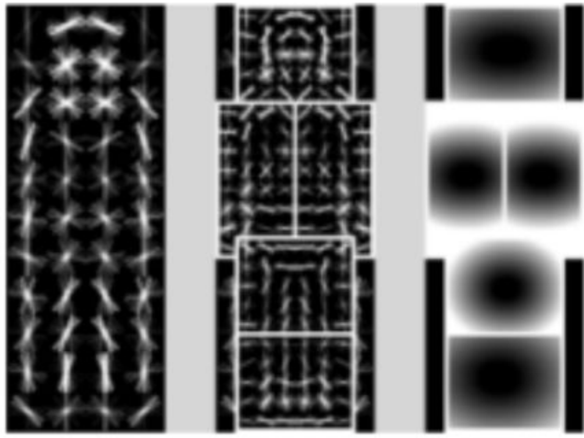


Each part is allowed to deform. So it deforms to where the person might be.

Intuition: If the head is here, where is the whole person likely to be?

Step 3: Add up the scores for the final detections





Deformation: score propagation (Your HW4)

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.

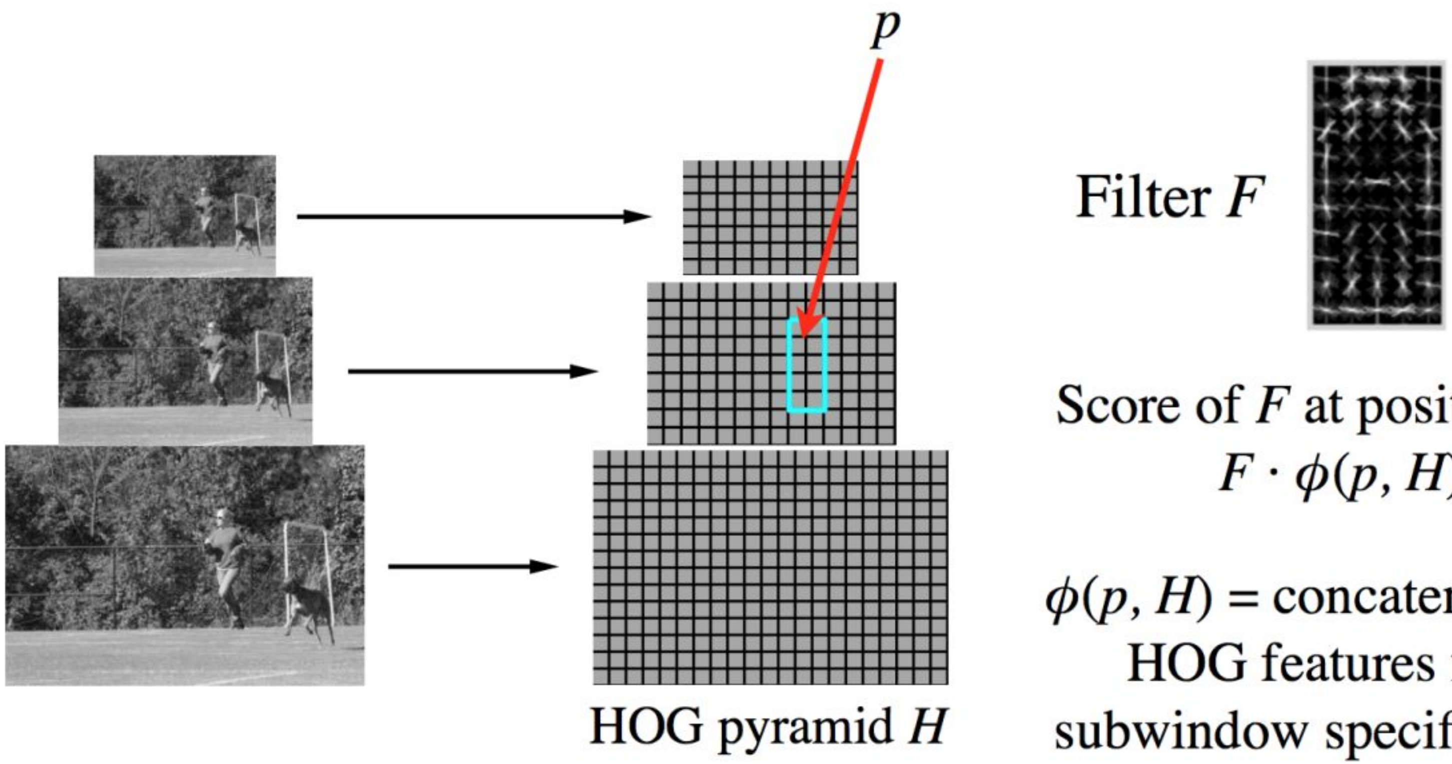
$$detection\ score = \sum_{i=0}^{n} F_i\, \phi(p_i, H) - \sum_{i=1}^{n} d_i(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2)$$

Calculating the score for a detection

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

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

Remember from Dalal and Triggs

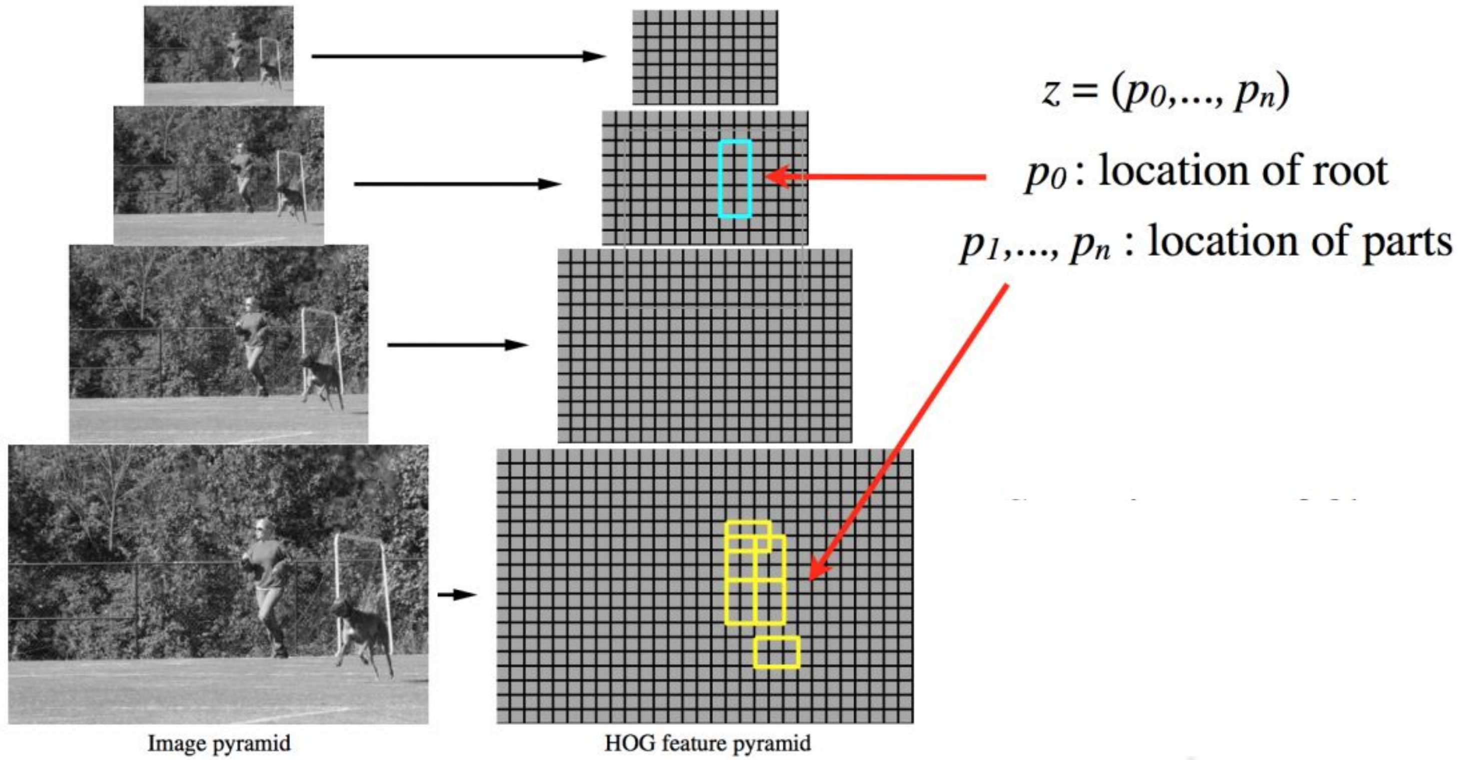


Score of F at position p is
$F \cdot \phi(p, H)$

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

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

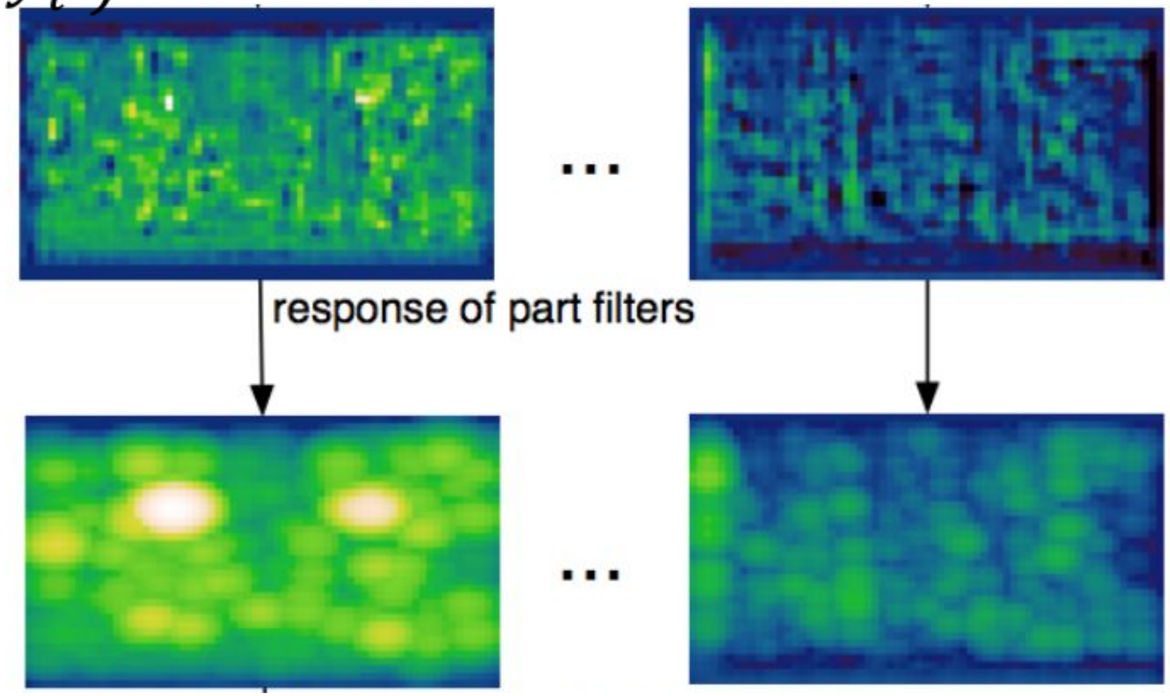
$p_i = (x_i, y_i, l_i)$ specifies the level and position of the *i*-th filter



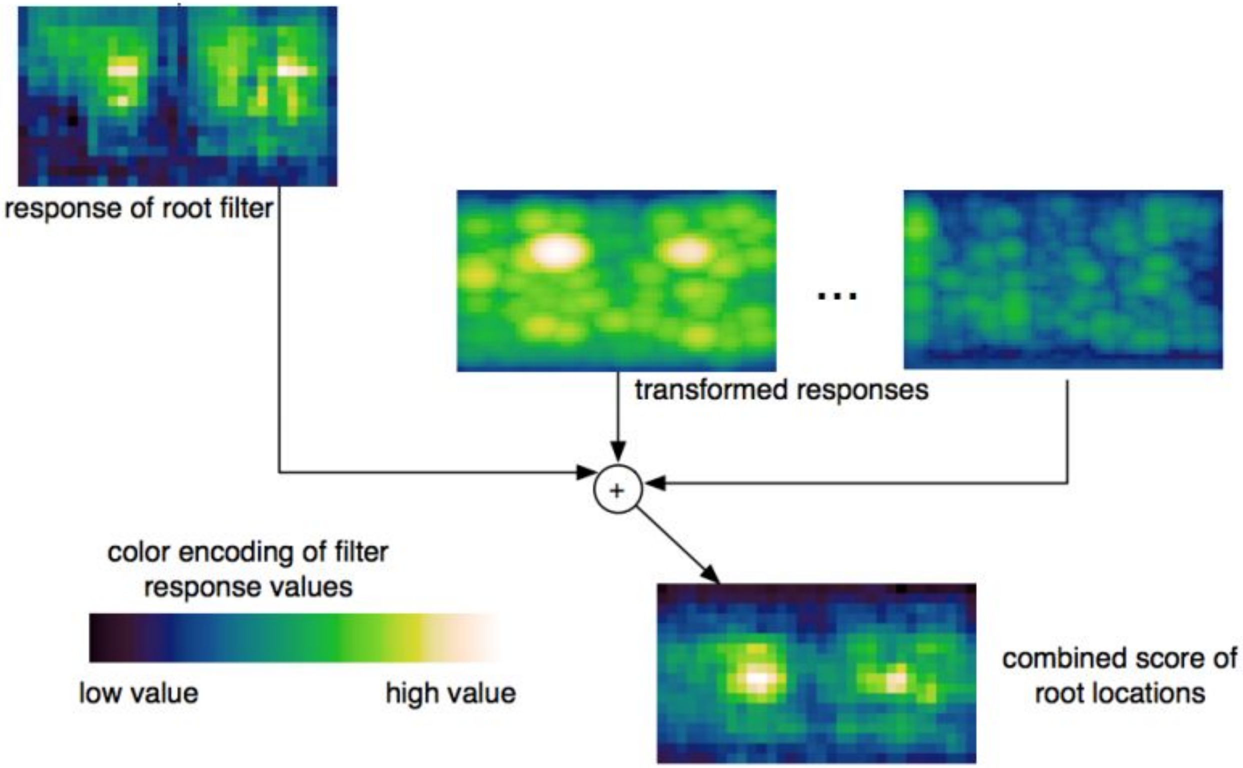
Detection pipeline

Now apply the spatial costs for each part:

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



Detection pipeline



Now add the global filter:

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

Calculating the score for a detection

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

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.

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?