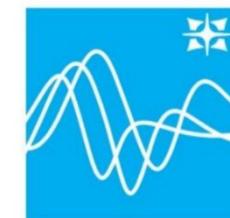


Компьютерное Зрение  
Лекция № 8, осень 2022

# Распознавание образов



Кафедра  
технологий  
проектирования  
сложных  
технических  
систем

# Что будем изучать сегодня

- Object detection
  - Задача и постановка
  - Простой детектор
  - Модели на частичном распознавании (Deformable parts model)

# Object Detection



Credit: Flickr user [neilalderney123](#)

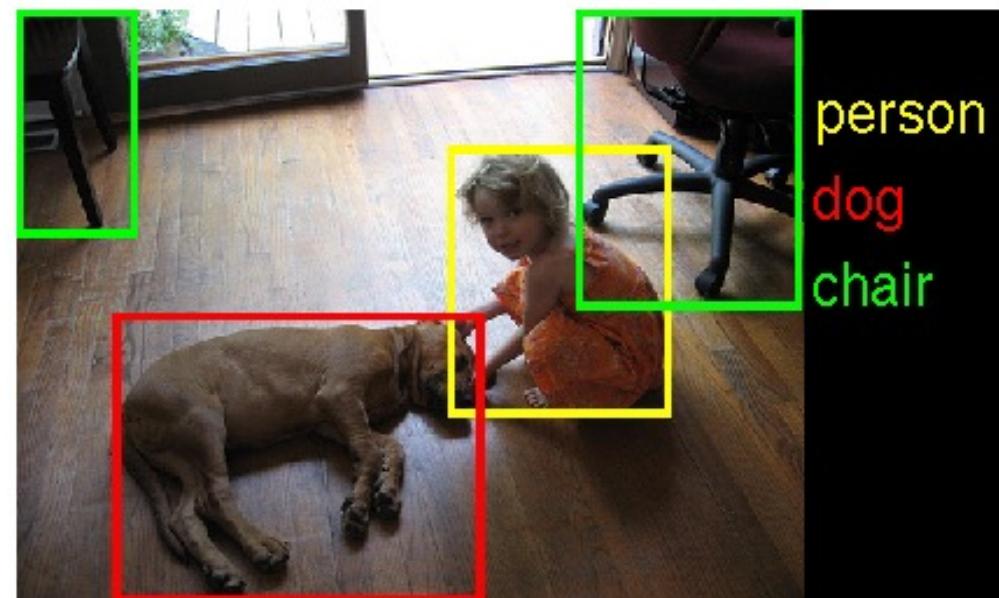
- Что видим на изображении?

# Object Detection

- **Задача:** Обнаружить и классифицировать объекты различных категорий

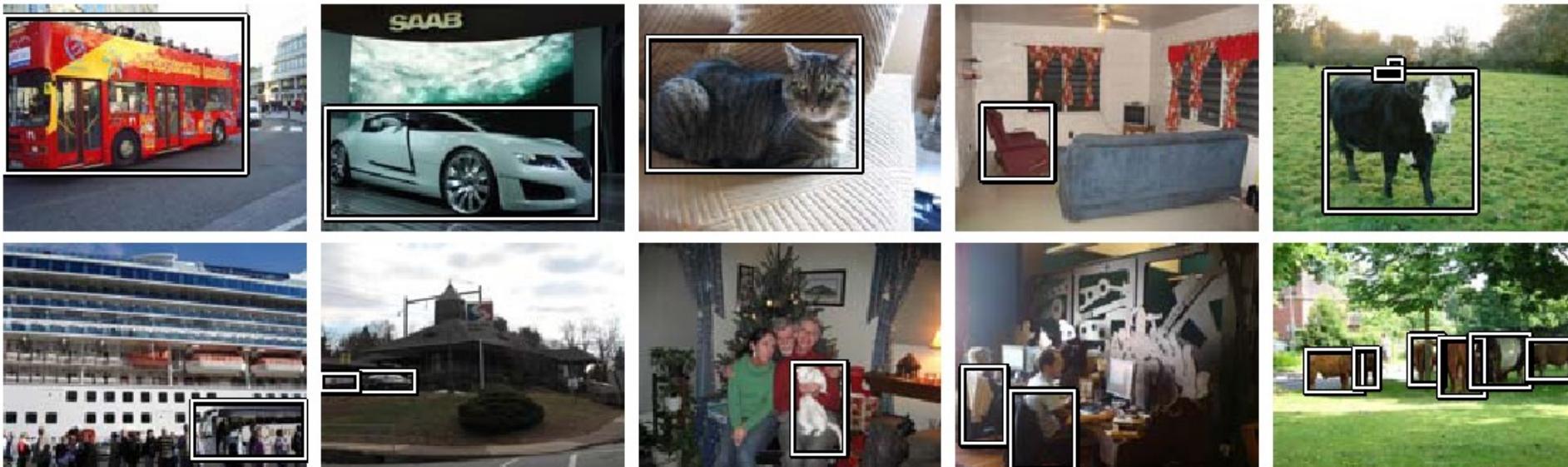
Проблемы:

- Освещение,
- Точка наблюдения
- Деформации
- Искажения
- Внутриклассовая изменчивость



# Object Detection Benchmarks

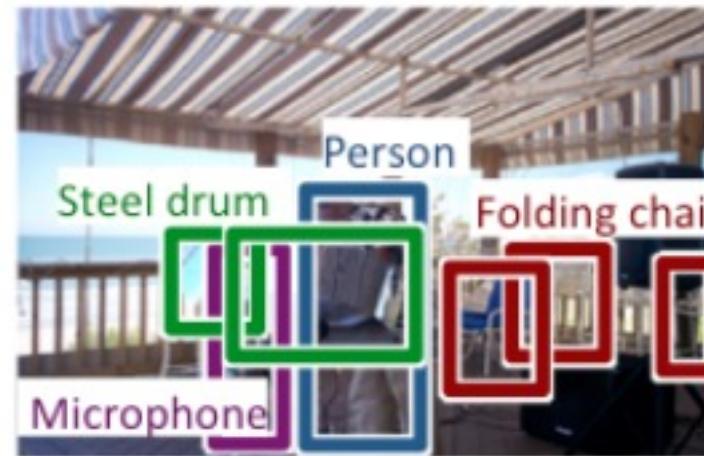
- PASCAL VOC Challenge



- 20 категории
- Классификация, обнаружение, сегментация, ...

# Object Detection Benchmarks

- PASCAL VOC Challenge
- ImageNet Large Scale Visual Recognition Challenge (ILSVR)
  - 200 категорий



# Object Detection Benchmarks

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



# Как будем измерять качество?



— predictions  
— ground truth

# Как будем измерять качество?



— predictions

— ground truth

## True positive:

- Перекрытие prediction с ground truth **больше**, чем 0.5

# Как будем измерять качество?



— predictions

— ground truth

**True positive:**

**False positive:**

- Пересечение  
prediction с ground  
truth **меньше**, чем 0.5

# Как будем измерять качество?



— predictions

— ground truth

**True positive:**

**False positive:**

**False negative:**

- Объекты, которые  
не были найдены

# Как будем измерять качество?



— predictions  
— ground truth

**True positive:**

**False positive:**

**False negative:**

- Объекты, которые не были найдены

Тогда что такое **True Negative**?

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	true positive	false negative
<u>True 0</u>	false positive	true negative

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	true positive	false negative
<u>True 0</u>	false positive	true negative

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	TP	FN
<u>True 0</u>	FP	TN

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	true positive	false negative
<u>True 0</u>	false positive	true negative

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	TP	FN
<u>True 0</u>	FP	TN

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	hits	misses
<u>True 0</u>	false alarms	correct rejections

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	true positive	false negative
<u>True 0</u>	false positive	true negative

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	TP	FN
<u>True 0</u>	FP	TN

	<u>Predicted 1</u>	<u>Predicted 0</u>
<u>True 1</u>	hits	misses
<u>True 0</u>	false alarms	correct rejections

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

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

# Как будем измерять качество?



— predictions  
— ground truth

**True positive: 1**

**False positive: 2**

**False negative: 1**

Тогда как считать

- precision?
- recall?

# Precision versus recall

- Precision:
  - Как много объектов было обнаружено верно?
- Recall:
  - Как много истинных объектов (ground truth) может обнаружить модель?

В реальности наша модель делает много прогнозов с различными оценками от 0 до 1



Отобразим все  
bboxes, у которых  
 $score > 0$

Видно, что  
- Recall идеален!  
- Но Precision очень  
плох!

# Как будем измерять качество?

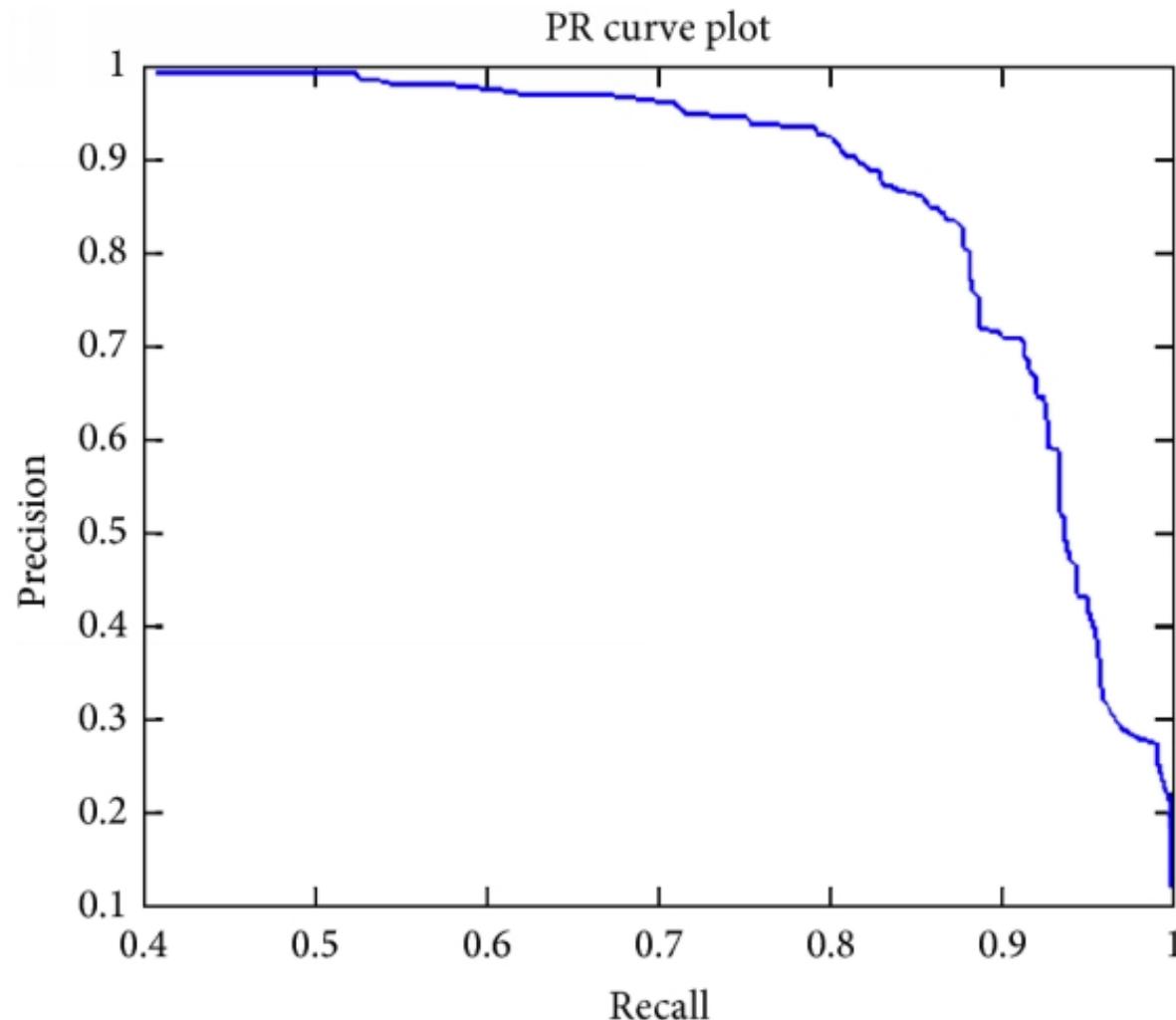


— predictions  
— ground truth

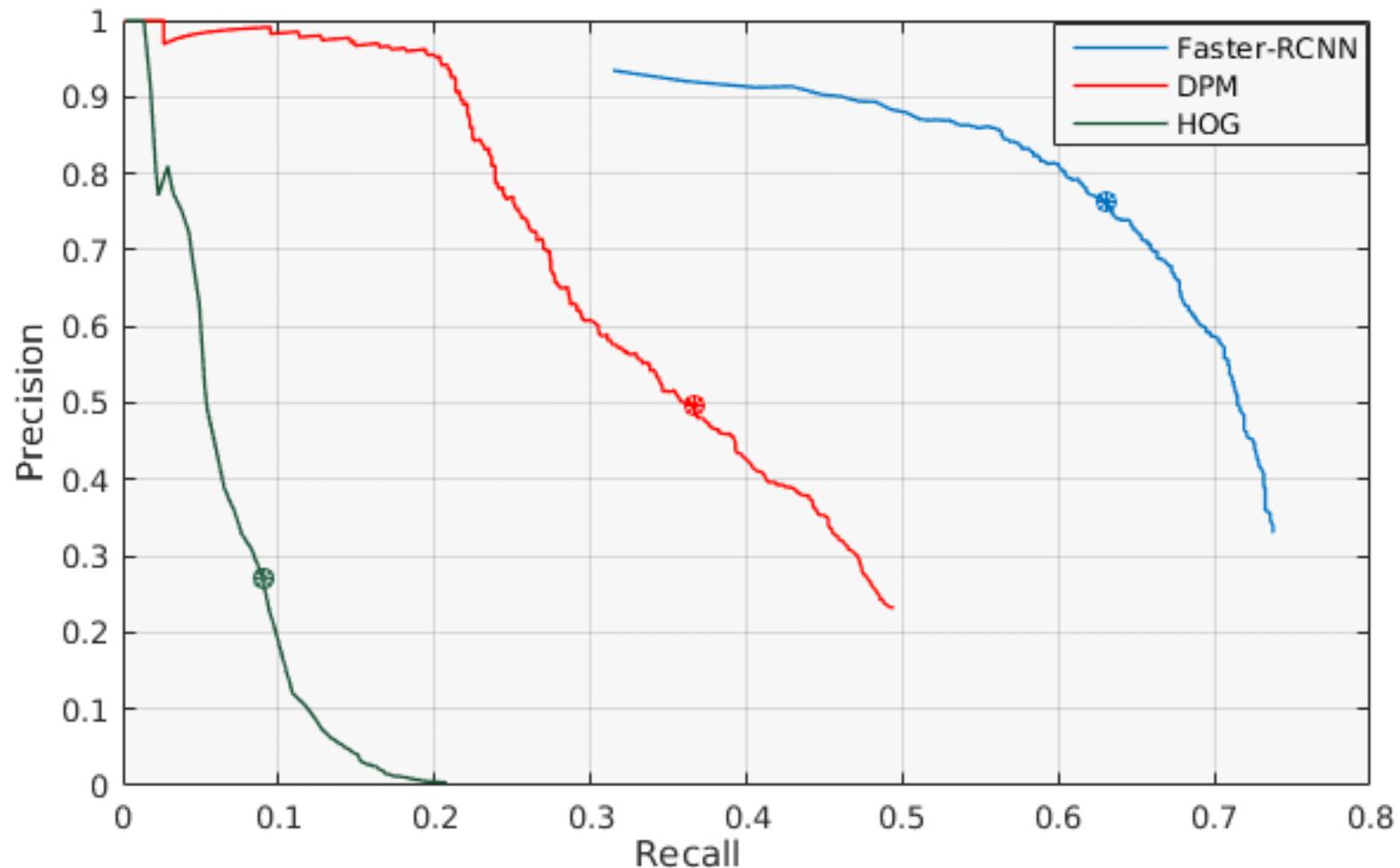
Отобразим только те  
bboxes, где  $\text{score} > 0.5$

Установим порог  
равный 0.5

# Precision – recall curve (PR curve)



# Какая модель лучше?



# True Positives - Person

---

UoCTTI\_LSVM-MDPM



MIZZOU\_DEF-HOG-LBP



NECUIUC\_CLS-DTCT



# False Positives - Person

UoCTTI\_LSVM-MDPM



MIZZOU\_DEF-HOG-LBP



NECUIUC\_CLS-DTCT



# “Near Misses” - Person

UoCTTI\_LSVM-MDPM



MIZZOU\_DEF-HOG-LBP



NECUIUC\_CLS-DTCT



# True Positives - Bicycle

UoCTTI\_LSVM-MDPM



OXFORD\_MKL



NECUIUC\_CLS-DTCT



# False Positives - Bicycle

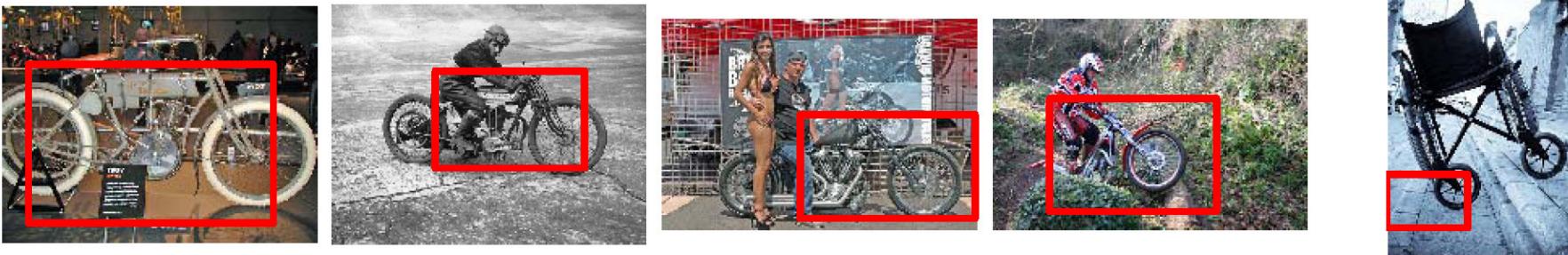
UoCTTI\_LSVM-MDPM



OXFORD\_MKL



NECUIUC\_CLS-DTCT



# Что будем изучать сегодня

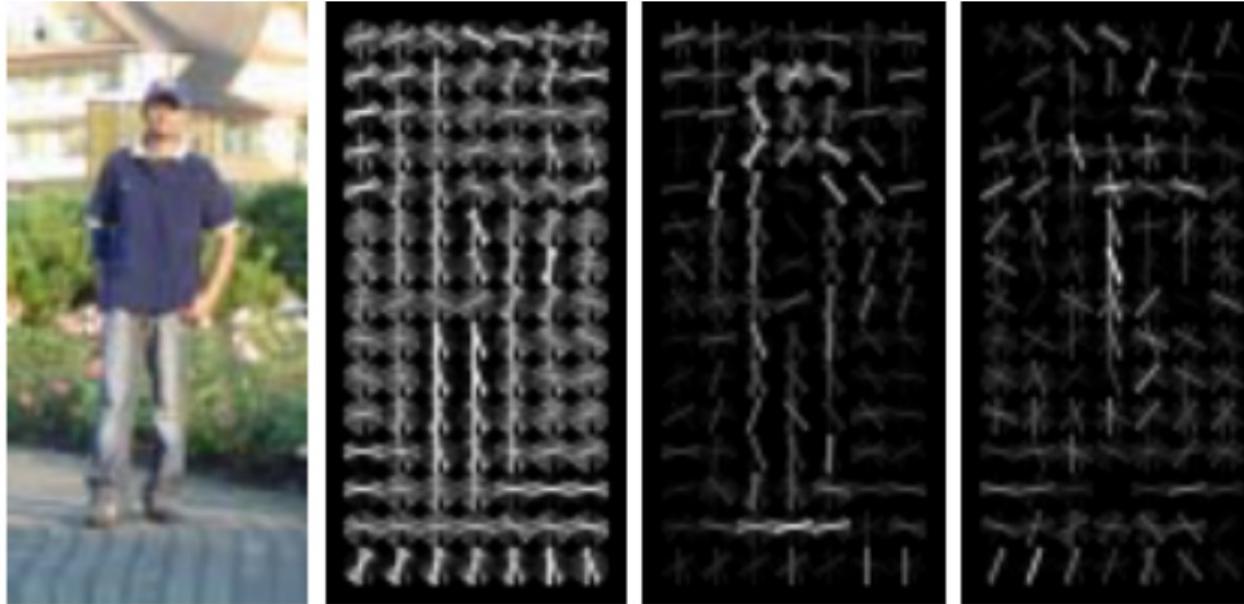
- Object detection
  - Задача и постановка
- Простой детектор
- Модели на частичном распознавании (Deformable parts model)

# Dalal-Triggs метод



sliding window

# Повторение. HOG фичи



Находим шаблон HOG и используем в качестве шаблона

# Sliding window + hog features



- Просмотрим окно изображения и проверим, есть ли там объект

No person here

# Sliding window + hog features



- Просмотрим окно изображения и проверим, есть ли там объект

YES!! Person match found

# Sliding window + hog features



- Но что если мы ищем  
что-то другое?  
Например, автобус

No bus found

# Sliding window + hog features



- Но что если мы ищем  
что-то другое?  
Например, автобус

No bus found

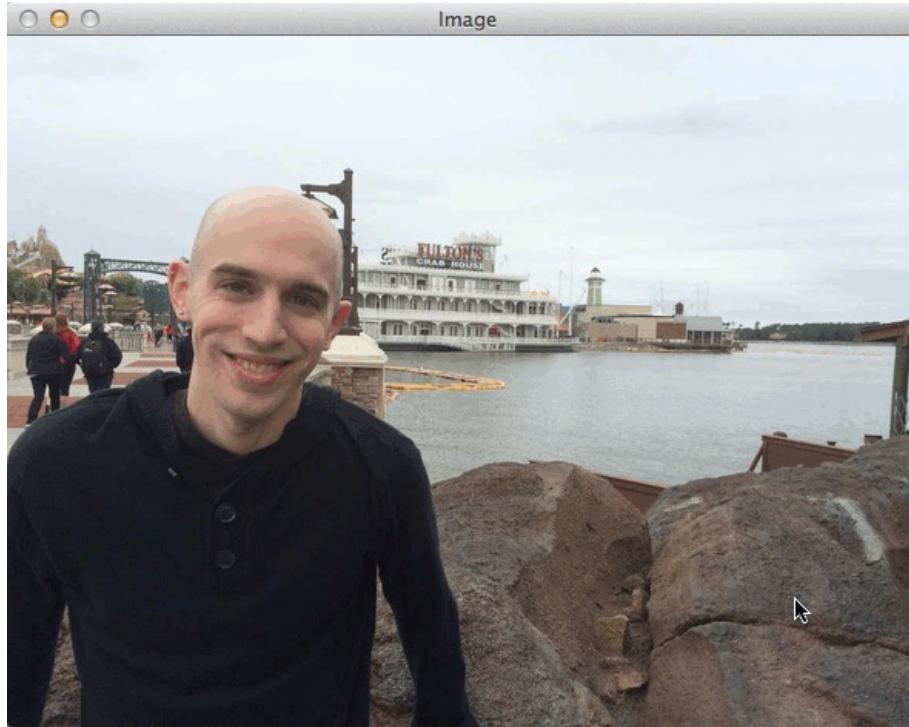
# Sliding window + hog features



- Мы никогда не найдем объект, если не подберем верный размер окна изображения

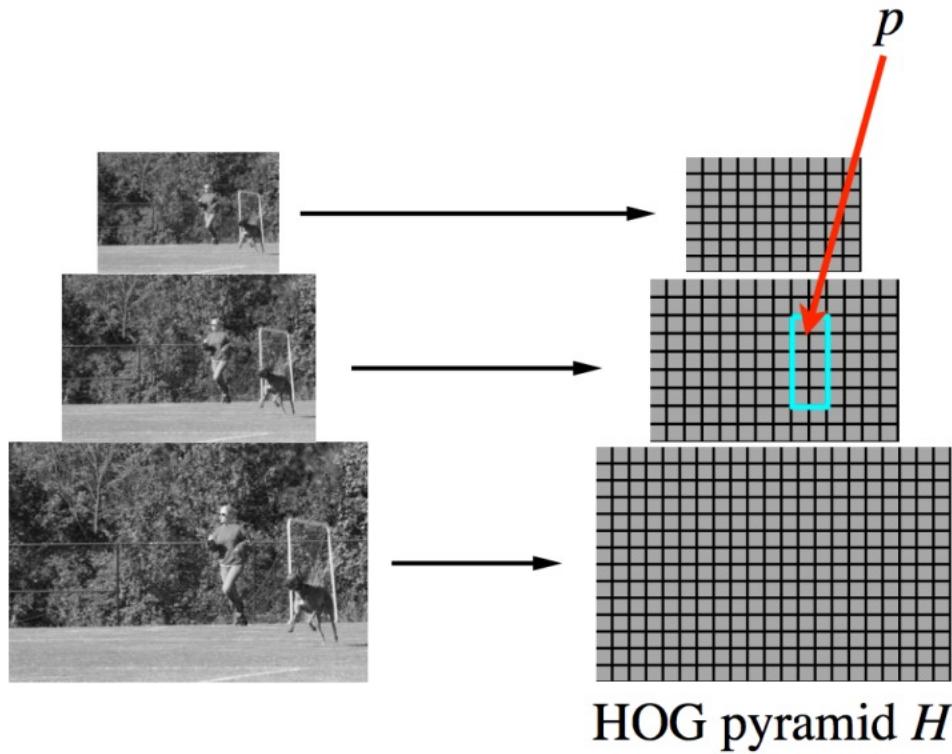
No bus found

# Sliding window + hog features



Нужно использовать **multi scale** sliding window

# Создадим 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  $p$

# Что будем изучать сегодня

- Object detection
  - Задача и постановка
- Простой детектор
- Модели на частичном распознавании (Deformable parts model)

# Повторение. Bag of visual words

- Можем представить изображение (образ) в виде «набора слов», где слово – обозначает часть изображения (образа)

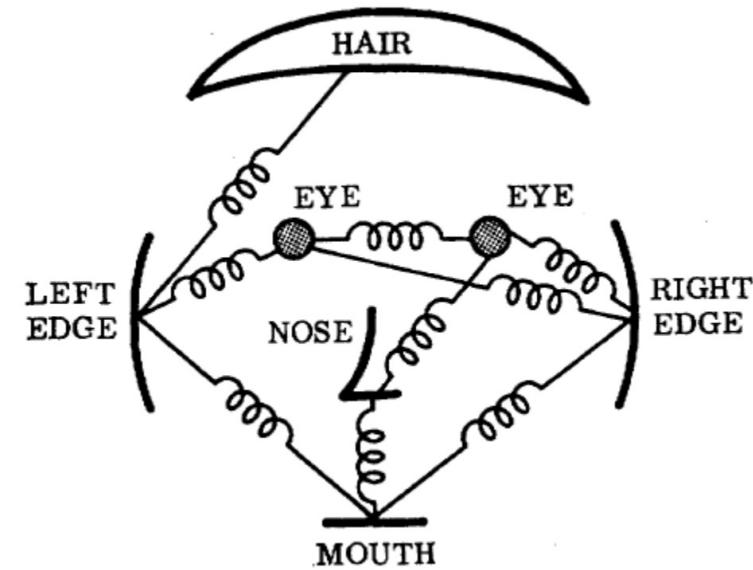
Bag of ‘words’



- Можем ли мы сделать то же самое для объектов внутри этих изображений?

# Deformable parts model

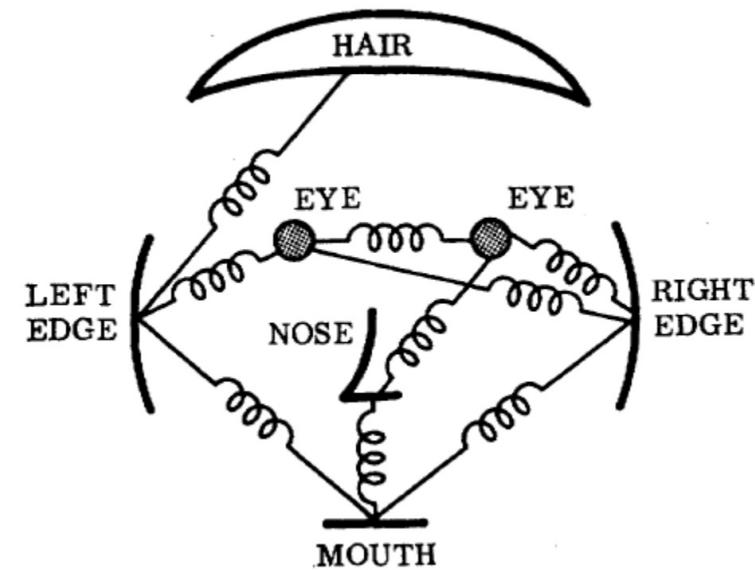
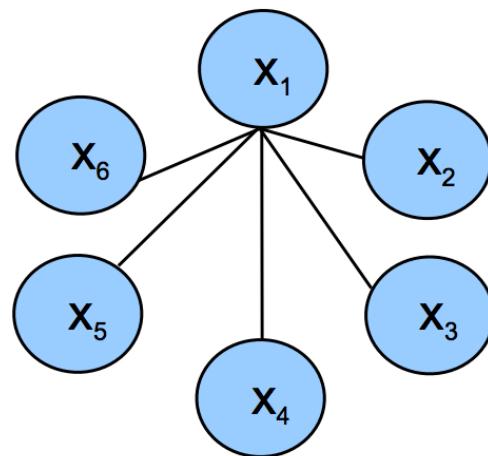
- Представим объект в виде коллекции деталей, расположенных в деформируемой конфигурации
- Каждая часть представляет собой локальные выступления
- Детали соединены «пружиноподобными связями»



Fischler and Elschlager,  
Pictoral Structures,  
1973

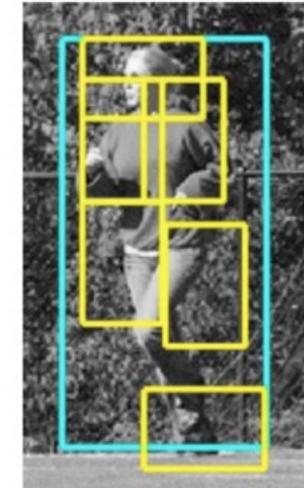
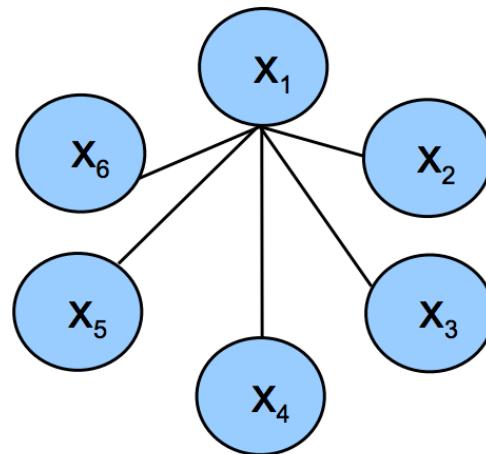
# Deformable parts model

- Части объекта образуют парные отношения.
- Мы можем смоделировать это, используя «графовую модель», где каждая часть определяется относительно корня



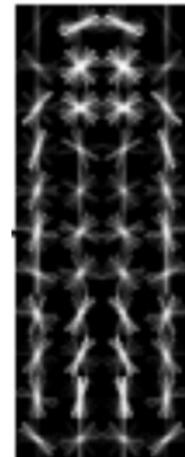
# Обнаружение человека по его частям

- Например, человек может быть смоделирован так: имеет голову, левую руку, правую руку и т.д.
- Все части могут быть смоделированы относительно глобального детектора человека, который выглядит, как граф (дерево)

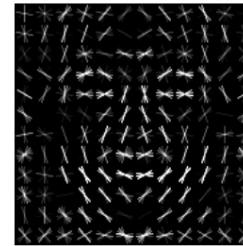


# Deformable parts model

- Каждая модель будет иметь **глобальный фильтр** и набор фильтров деталей. Приведем пример глобального фильтра для людей с фильтром частей "голова":

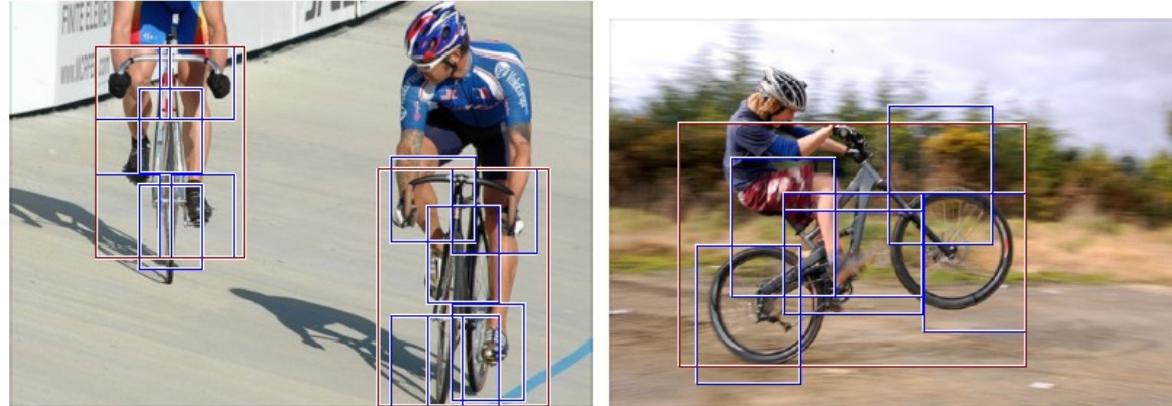


Global/root  
filter



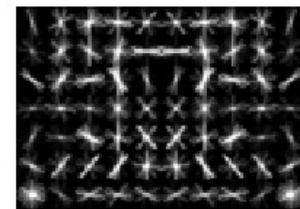
Part  
filter

# Двухкомпонентная модель велосипеда

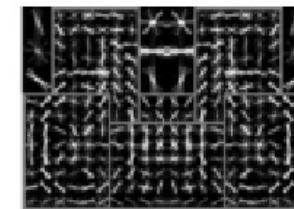


"вид сбоку"  
компонент модели  
велосипеда

Root filter

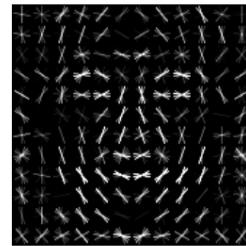
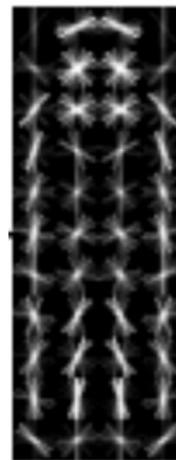


Part filters

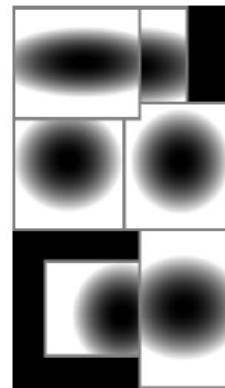
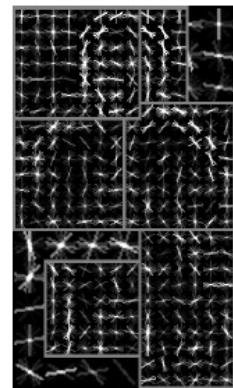
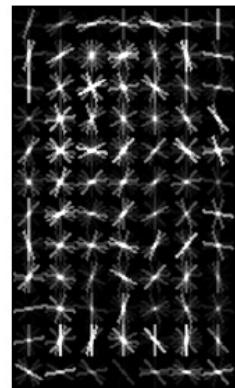
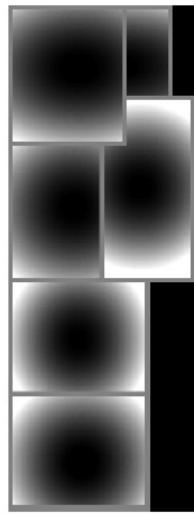
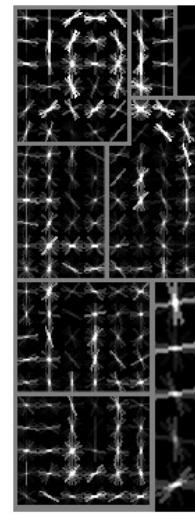
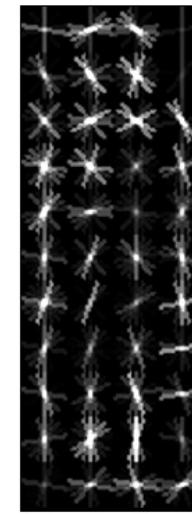
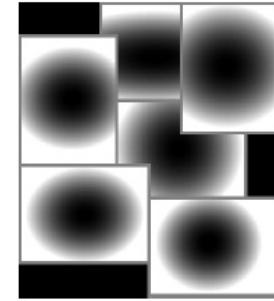
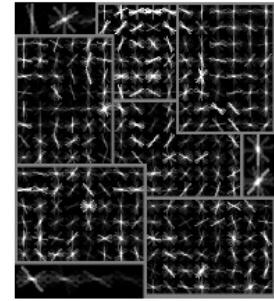


# Deformable parts model

- Ансамбль деформируемых моделей деталей
- Каждый компонент имеет глобальный компонент + деформируемые части
- Фильтры деталей имеют более тонкие особенности образа

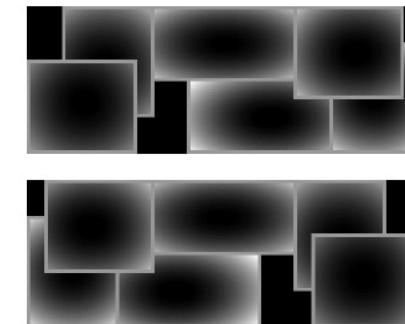
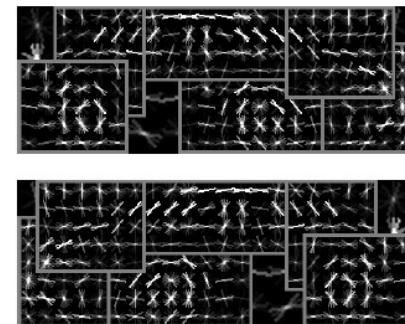
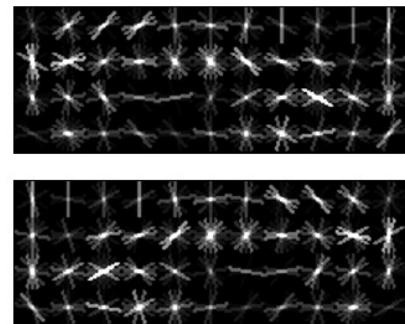


# Шестикомпонентная модель человека

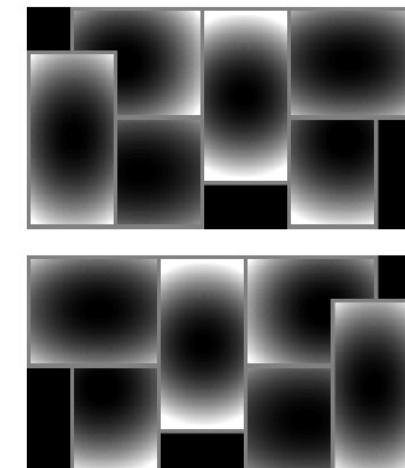
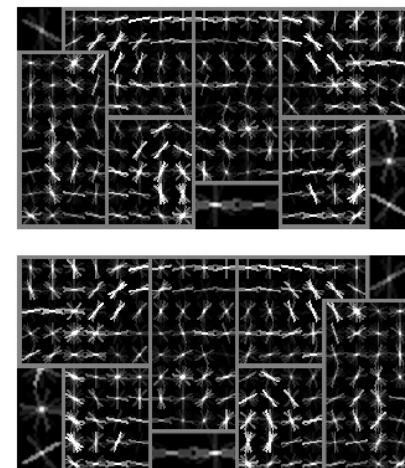
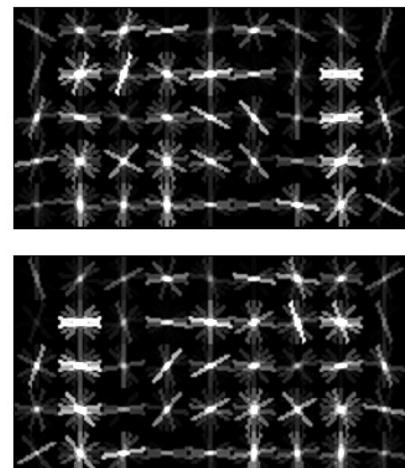


# Шестикомпонентная модель машины

side view



frontal view

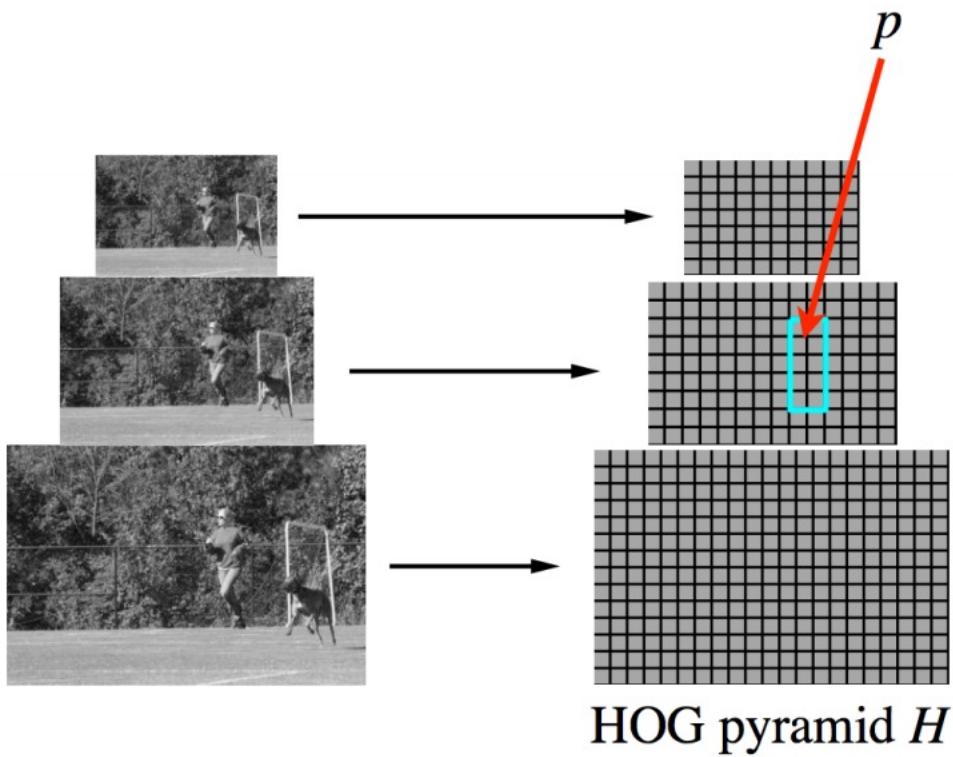


root filters (coarse)

part filters (fine)

deformation models

# Вспомним метод Dalal-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  $p$

# Deformable parts model

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

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

Root filter      Model for 1st part      Bias term

- Each part-based model defined as:

$$(F_i, v_i, d_i)$$

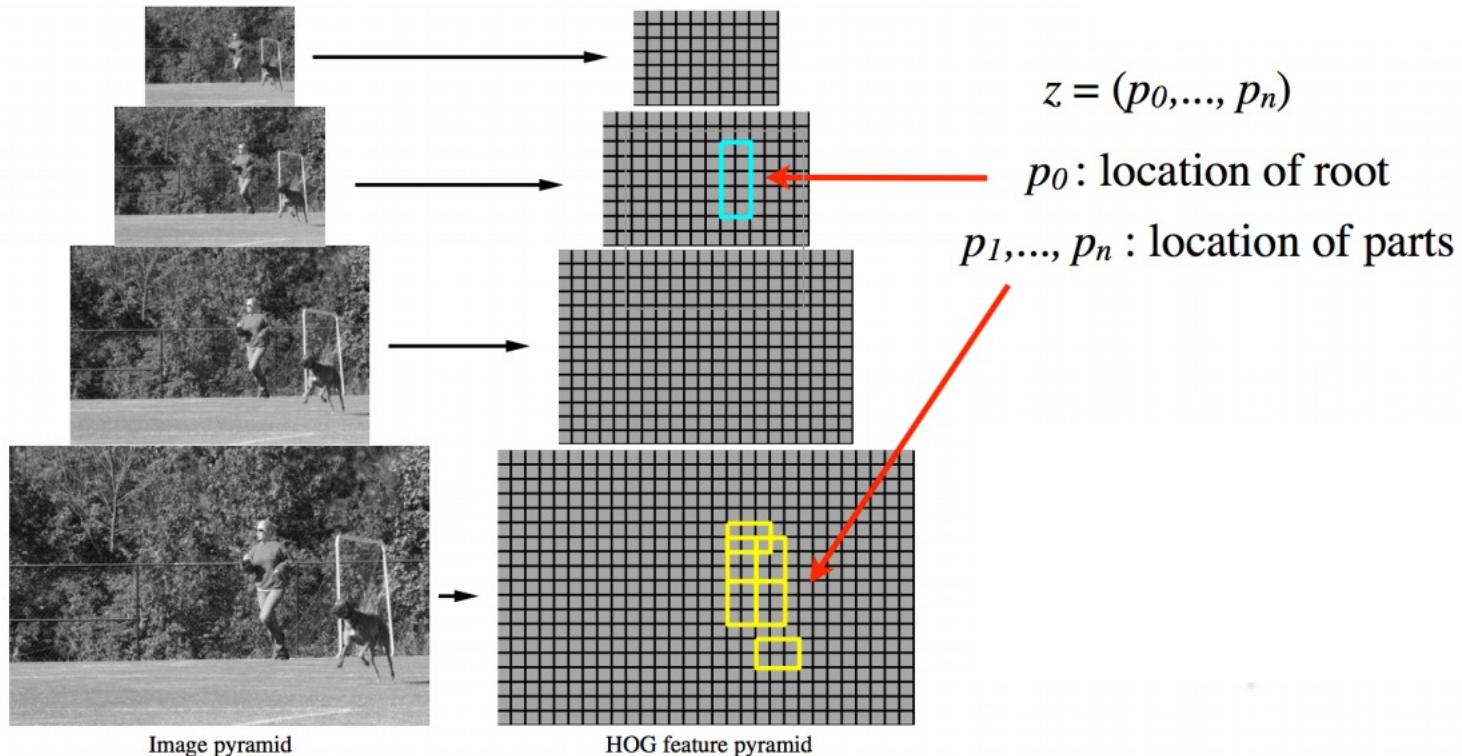
$F_i$  filter for the  $i$ -th part

$v_i$  “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

Деформируемые части вычисляют локальную оценку в каждом масштабе для расчета глобальной оценки

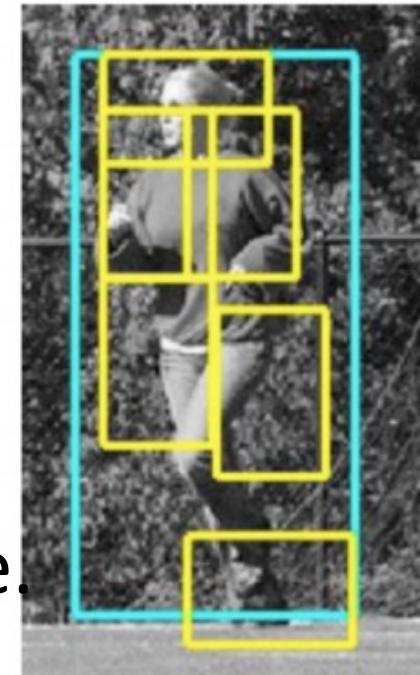
$p_i = (x_i, y_i, l_i)$  specifies the level and position of the  $i$ -th filter



# Расчет оценки качества обнаружения

Сумма баллов за обнаружение определяется как сумма баллов для глобальных детекторов и детекторов деталей за вычетом суммы затрат на деформацию для каждой детали.

Это означает, что если детектор части очень далеко от того, где они должны быть, возможно, это ложное срабатывание.



# Calculating the score for a detection

Сумма баллов за обнаружение определяется как сумма баллов для глобальных детекторов и детекторов деталей за вычетом суммы затрат на деформацию для каждой детали.

*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

*detection score*

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

Score для каждого фильтра  
детали + глобальный фильтр  
(аналогично по Dalal-Triggs).

# Calculating the score for a detection

*detection score*

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

Затраты на деформацию каждой детали:

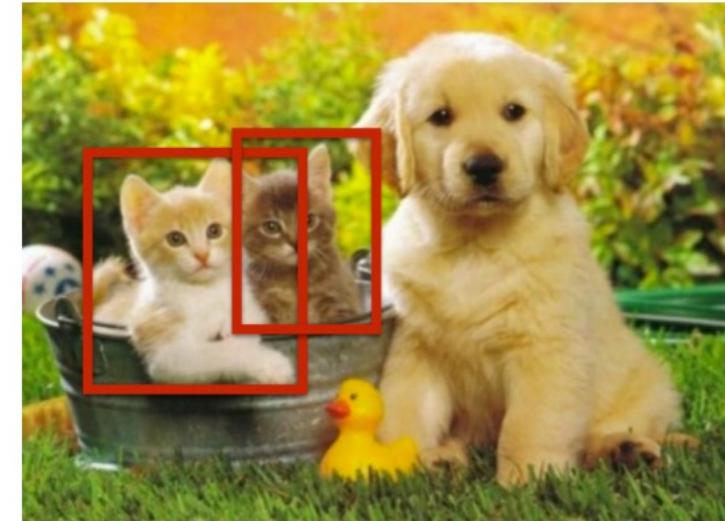
$\Delta x_i$  измеряет расстояние в x-направлении от того места, где должна находиться часть i.

$\Delta y_i$  измеряет аналогичное расстояние в направлении оси y.

$d_i$  - это вес, связанный с частью i, который наказывает часть за то, что она уехала.

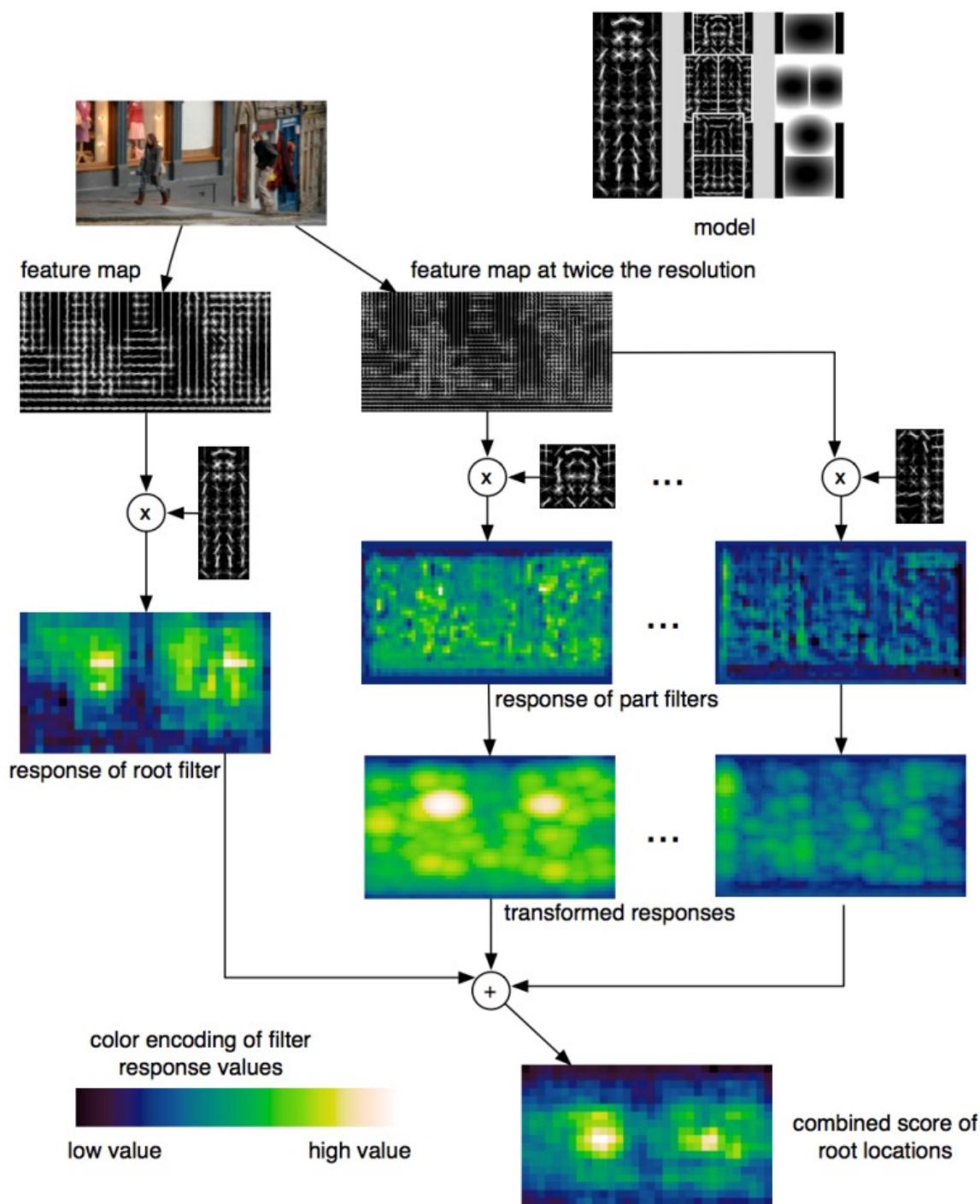
# Detection pipeline

- Итак, для обнаружения мы используем технику скользящего окна, а также глобальные фильтры и фильтры деталей.
- Чтобы добиться обнаружения, мы накапливаем общий балл и баллы детали и наказываем за деформацию деталей.



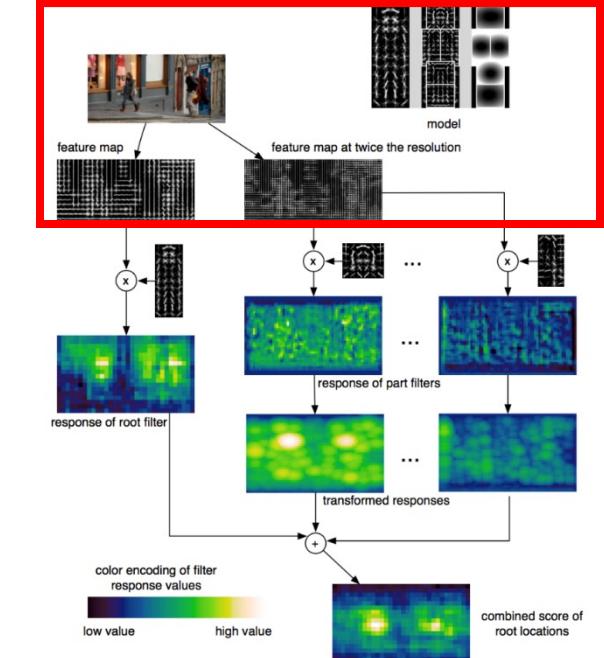
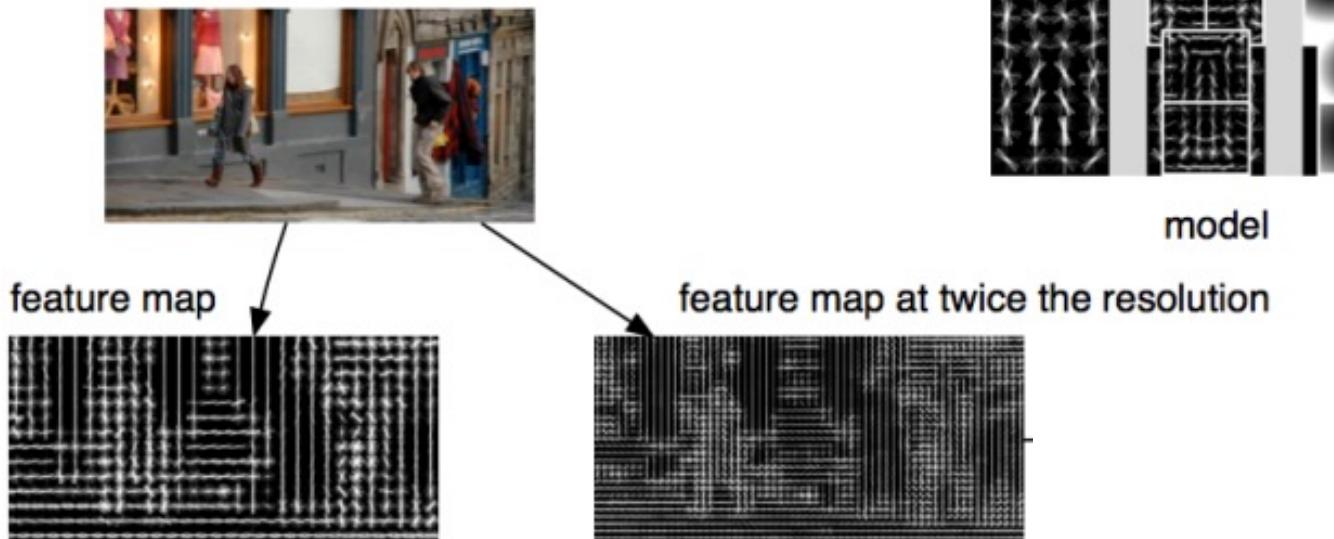
# Общий пайплайн детектора

Разберем по  
частям



# Detection pipeline

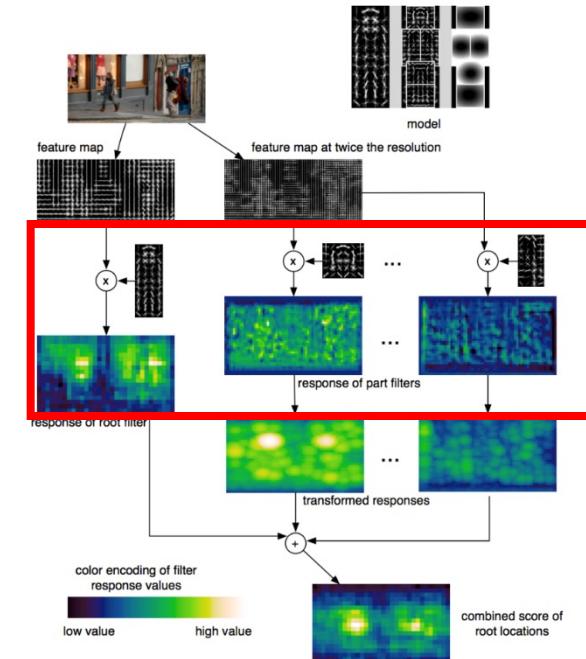
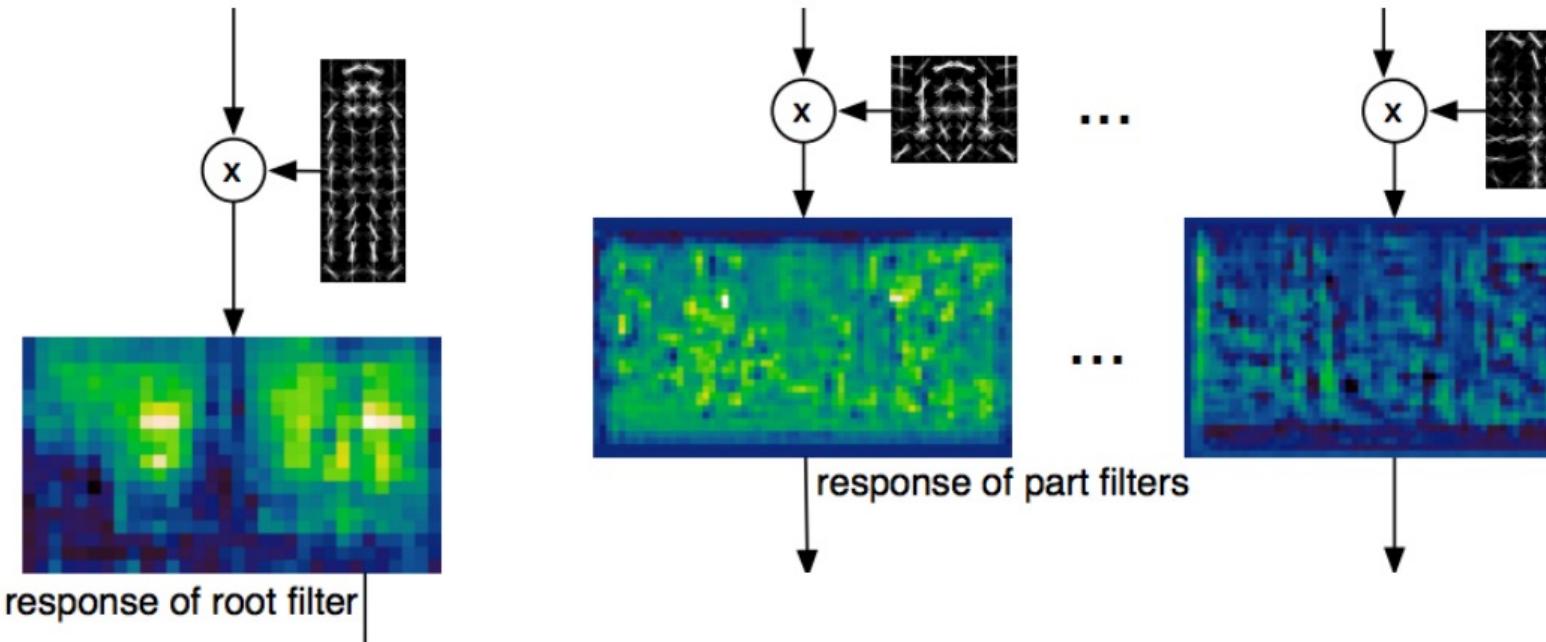
1. Убедитесь, что у вас есть фильтры (шаблоны) для глобальных и составных частей изображений (образов):  $F_i$
2. Вычислить карты функций HOG из входного изображения



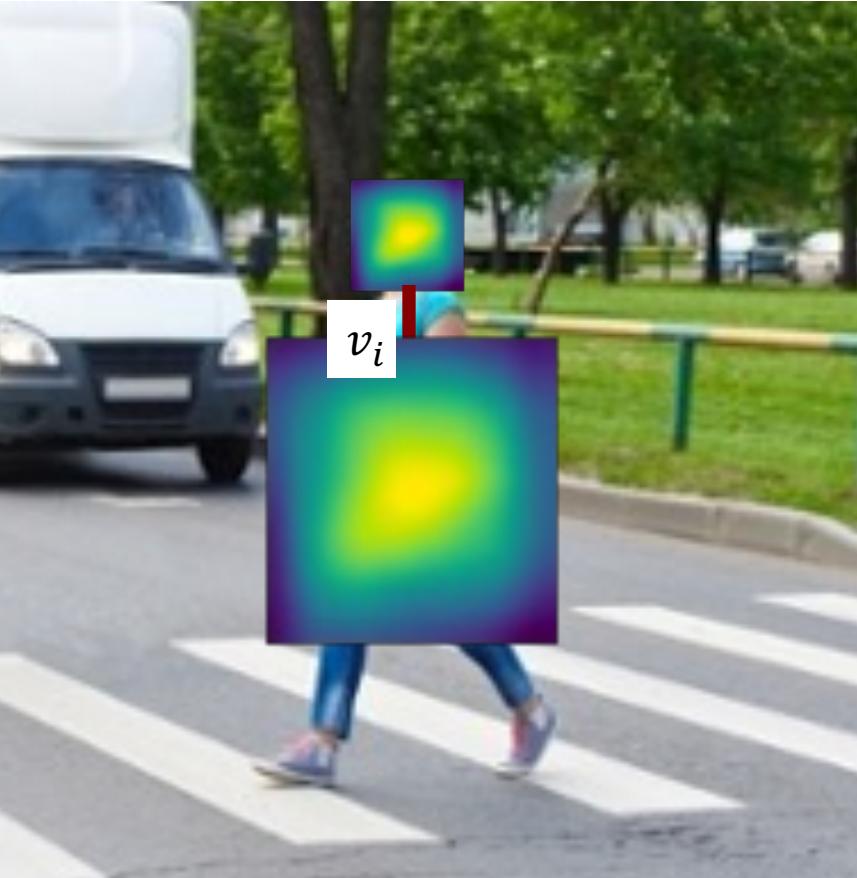
# Detection pipeline

Применить фильтры

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



# Учет пространственного расположения

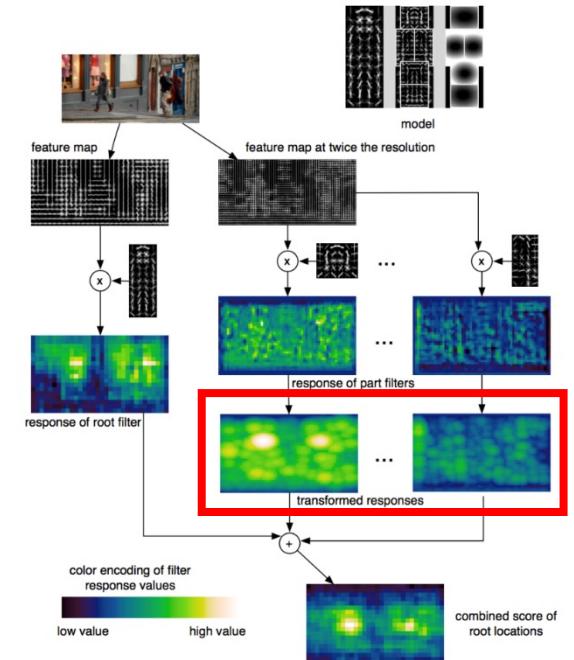
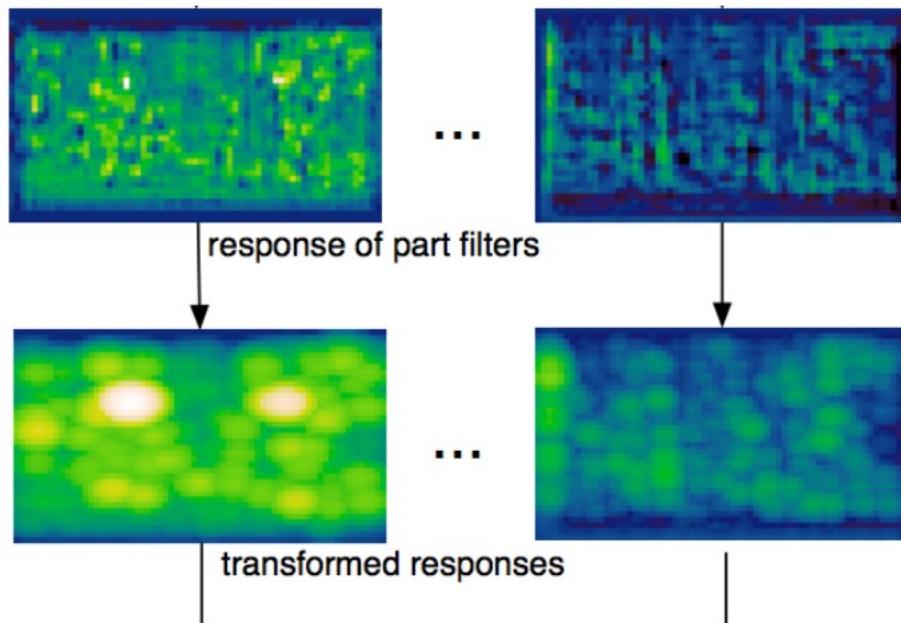


- Учитывая местоположение обнаруженной головы, мы можем угадать, где должно быть тело.
- Тело должно быть в направлении, вычисленном из фильтра корневых персон:  $v_i$
- Но мы допускаем некоторую деформацию или пространственный сдвиг в расположении головы по отношению к телу:  $d_i$
- Мы должны "распространить" обнаружение головы при вычислении потенциальных местоположений корня

# Detection pipeline

Рассчитаем score для каждой части:

$$\text{detection score} = F_i \phi(p_i, H) - d_i(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2)$$

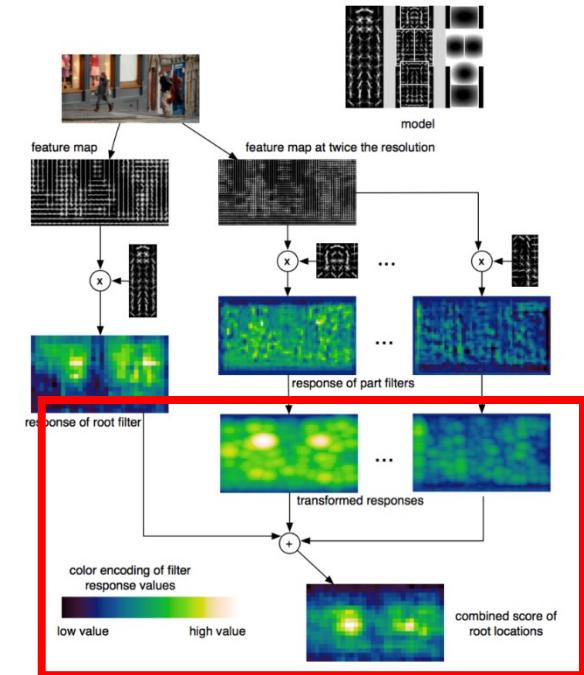
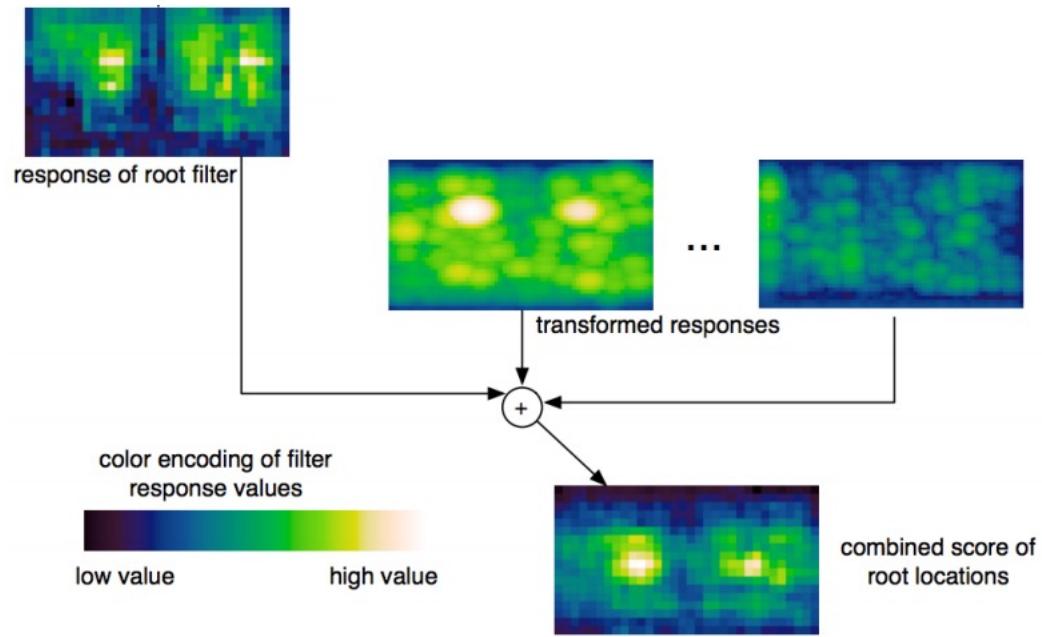


# Detection pipeline

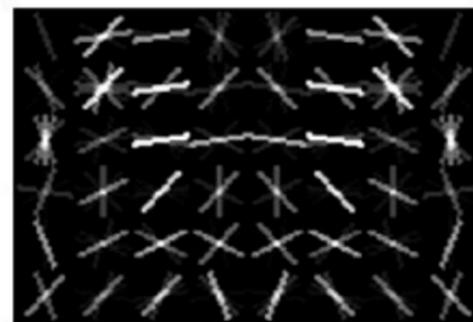
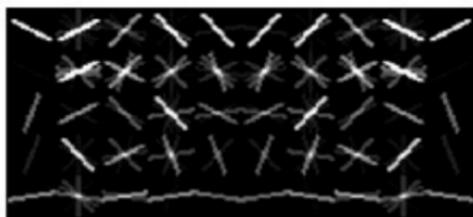
Теперь добавим результат от корня:

*detection score*

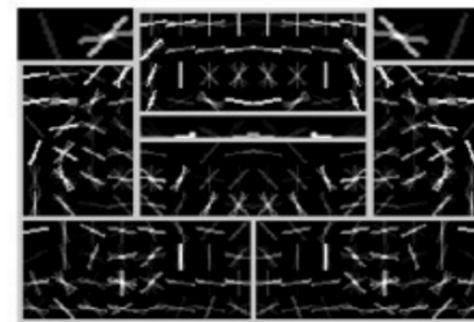
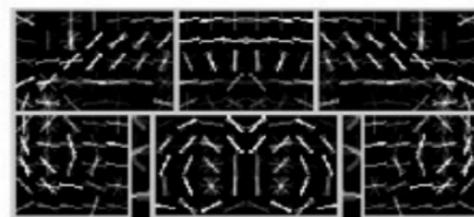
$$= F_0 + \sum_{i=1}^n F_i \phi(p_i, H) - \sum_{i=1}^n d_i(\Delta x_i, \Delta y_i, \Delta x_i^2, \Delta y_i^2)$$



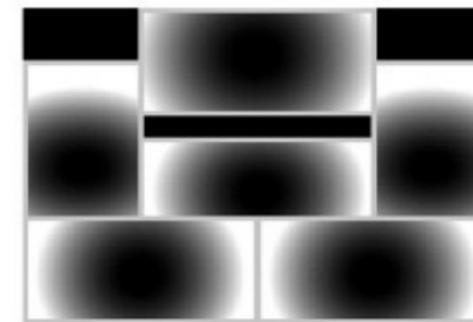
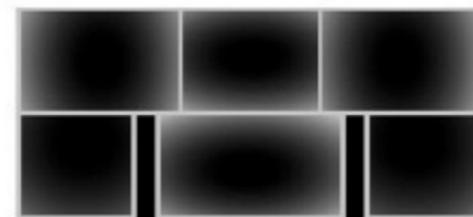
# DPM - bicycle



root filters  
coarse resolution

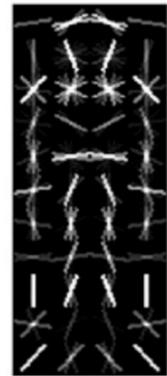


part filters  
finer resolution



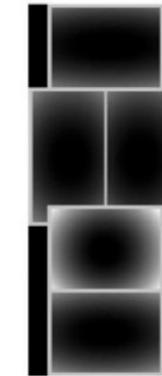
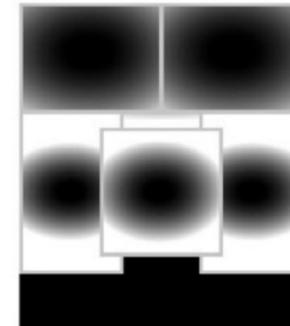
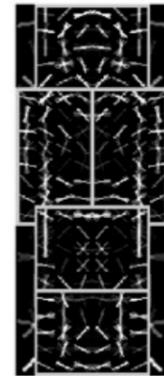
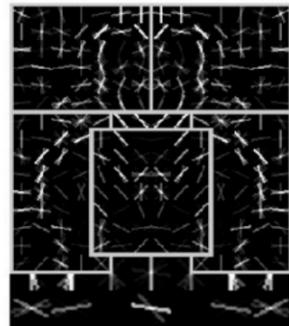
deformation  
models

# DPM - person



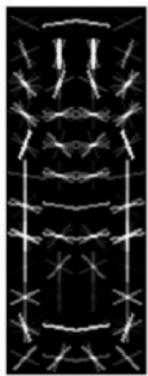
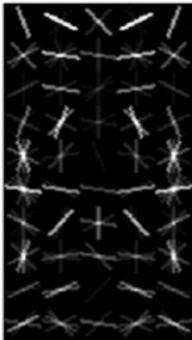
root filters  
coarse resolution

part filters  
finer resolution

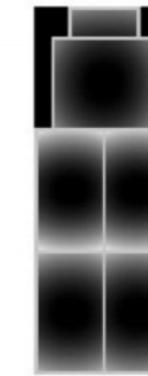
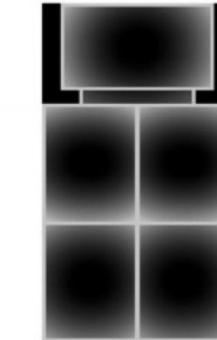
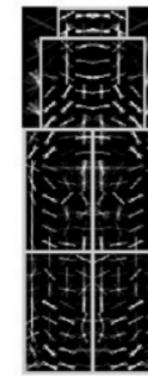
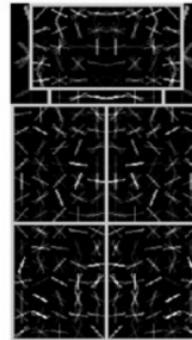


deformation  
models

# DPM - bottle



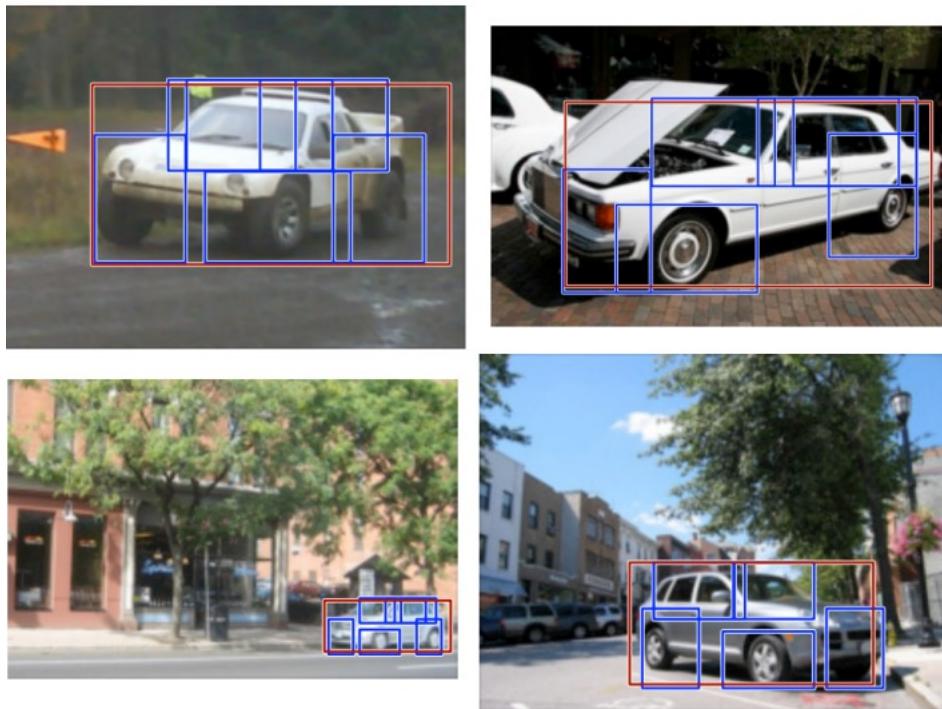
root filters  
coarse resolution      part filters  
finer resolution



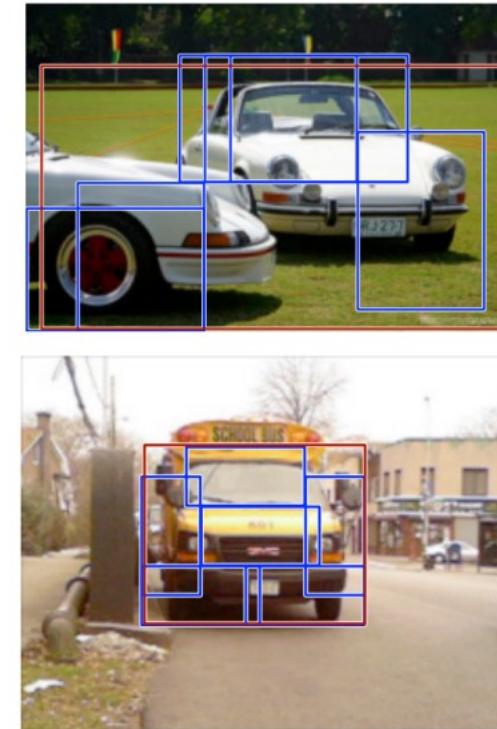
deformation  
models

# Results – Car

high scoring true positives

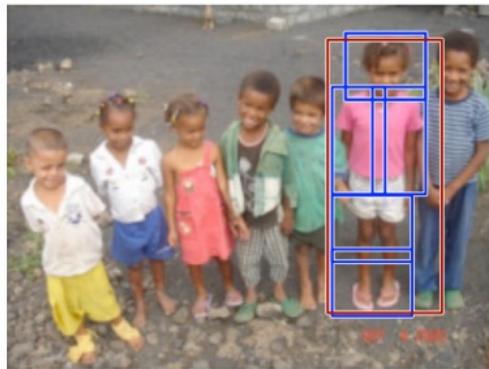
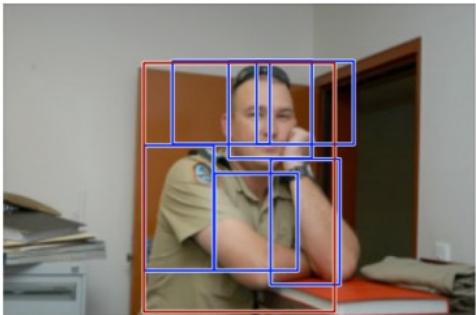
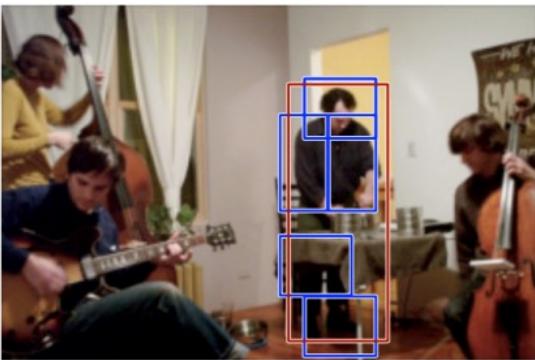


high scoring false positives

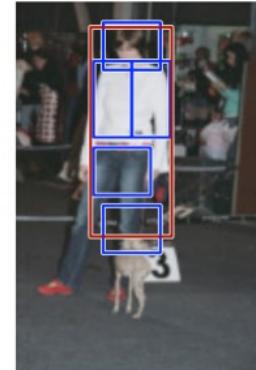


# Results – Person

high scoring true positives

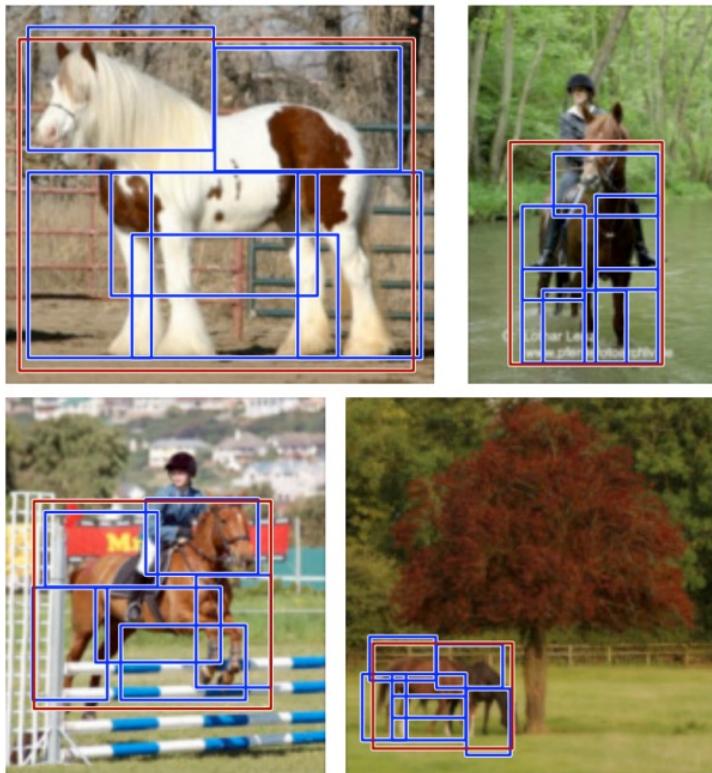


high scoring false positives  
(not enough overlap)

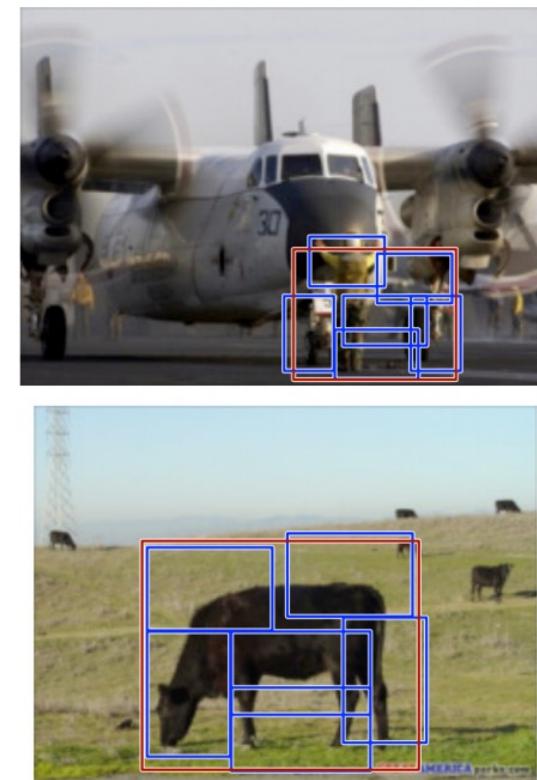


# Results – Horse

high scoring true positives



high scoring false positives

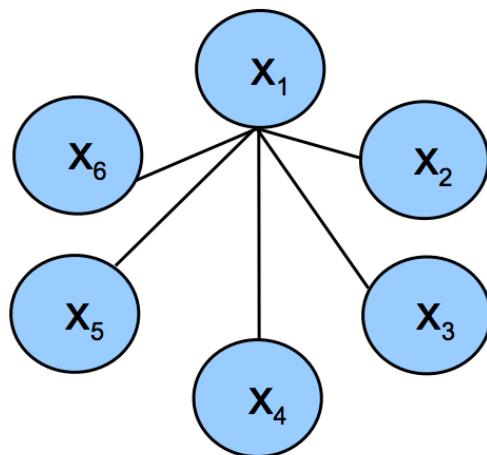


# DPM

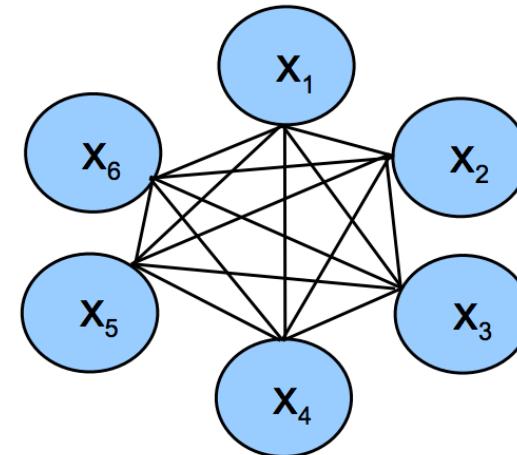
- Подход
  - Вручную подобранный набор деталей - Специальные детекторы, обученные для каждой детали
  - Пространственная модель, тренированная на частичных активациях
  - Оценка совместной вероятности частичной активации
- Преимущества
  - Детали имеют интуитивную интерпретацию
  - Для каждой детали могут быть использованы стандартные подходы к обнаружению
  - Хорошо работает для определенных категорий
- Недостатки
  - Детали должны быть выбраны вручную
  - Семантически детали иногда не имеют простого распределения по внешнему виду
  - Нет гарантии, что какая-то важная часть не была пропущена
  - При переходе в другую категорию модель необходимо перестраивать с нуля

# Дополнительно

“Star” shape model



Fully connected shape model



- ▶ e.g. ISM (Implicit Shape Model)
- ▶ Parts mutually independent
- ▶ Recognition complexity:  $O(NP)$
- ▶ Method: Generalized Hough Transform

- ▶ e.g. Constellation Model
- ▶ Parts fully connected
- ▶ Recognition complexity:  $O(N^P)$
- ▶ Method: Exhaustive search

# Заключение

- Object detection
  - Задача и постановка
- Простой детектор
- Модели на частичном распознавании (Deformable parts model)