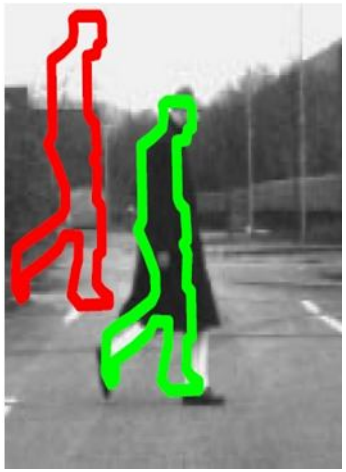
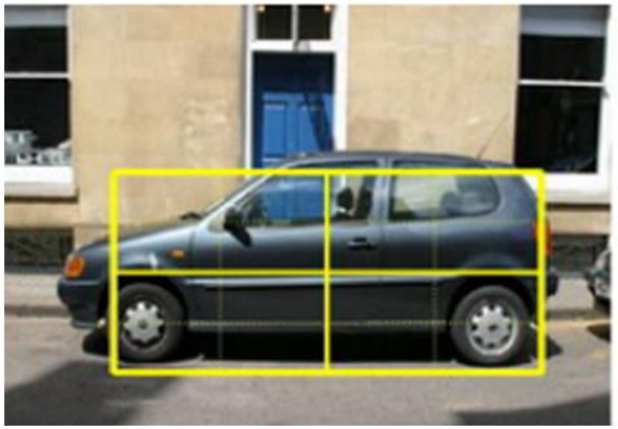
Microsoft Research

Поиск и локализация объектов





Many slides adopted from Svetlana Lazebnik, Ondra Chum, Alyosha Efros, Mark Everingham, Pedro Felzenszwalb, Rob Fergus, Kristen Grauman, Bastian Leibe, Ivan Laptev, Fei-Fei Li, Marcin Marszalek, Pietro Perona, Deva Ramanan, Bernt Schiele, Jamie Shotton, Josef Sivic and Andrea Vedaldi

Общая информация

Microsoft Research

Этот курс подготовлен и читается при поддержке Microsoft Research

Microsoft Research

- Страница курса http://courses.graphicon.ru/main/vision

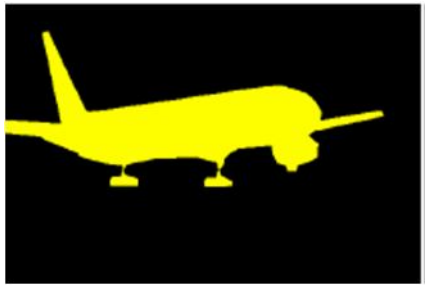
Microsoft Research

Задачи распознавания

- Классификация изображений
  - Image classification
  - Содержит ли изображение самолёт?

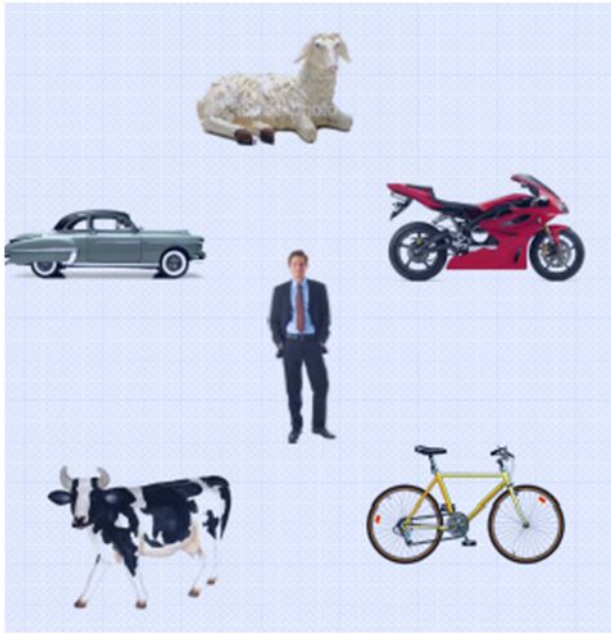
- Поиск и локализация объектов
  - Object class detection and localization
  - Где на изображении (и есть ли) самолет?

- Семантическая сегментация
  - Object class segmentation
  - Какие пиксели изображения принадлежат самолёту?



Объекты и материалы

- Объекты (Things)
  - Имеют определенные размер и формы



Ted Adelson, Forsyth et al. 1996.

- Материалы (Stuff)
  - Однородный или повторяющийся шаблон мелких деталей
  - Нет определенного размера и формы

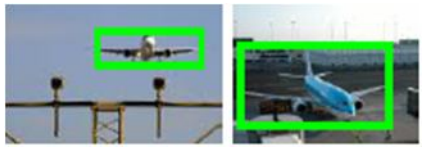
Slide: Geremy Heitz

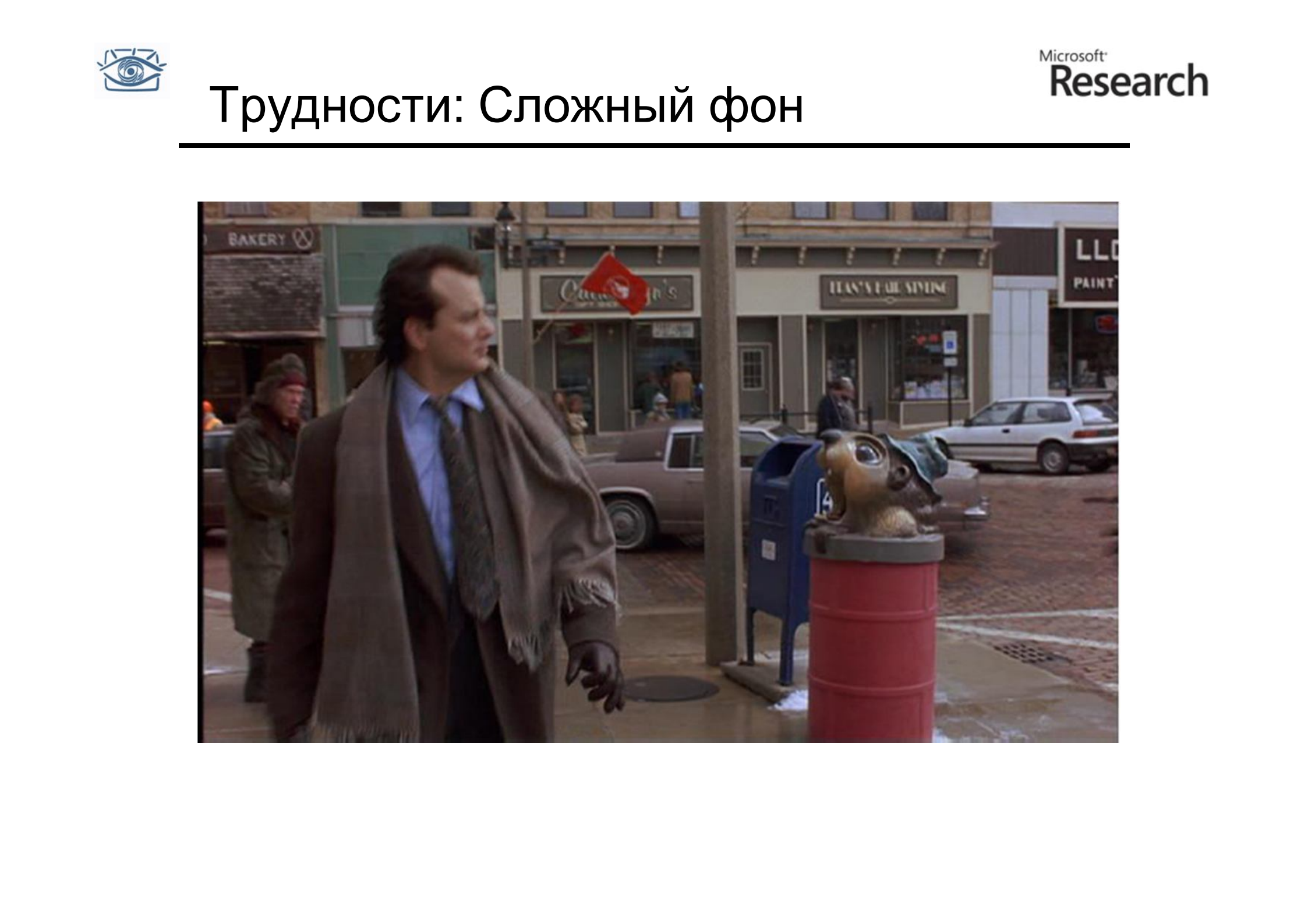
Задача поиска и локализации

- Поиск и локализация объектов
  - Где на изображении (и есть ли) самолет?

- Сложности
  - Внутриклассовая изменчивость
  - Изменение свойств съёмки (освещение, точка обзора, размер)

- Сравнение с классификацией
  - Структурный выход (где объект, ограничивающий прямоугольник)
  - Положение обычно задаётся при тренировке





Трудности: Сложный фон

Microsoft Research

Перекрытие и обрезание

Внутриклассовая изменчивость

Машинное обучение

- Построить модель класса объекта крайне сложно, поэтому будем обучать модель / алгоритм по набору примеров изображений объекта

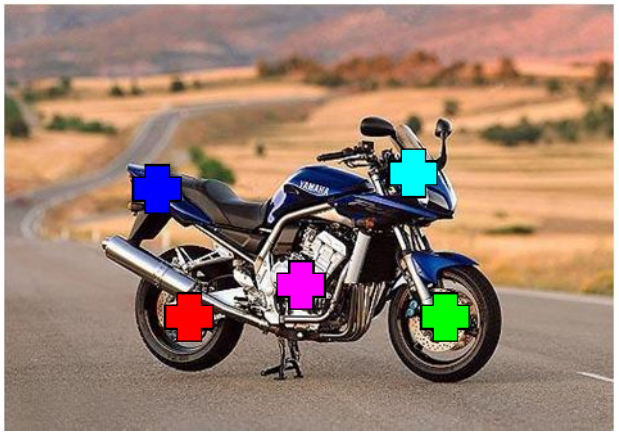
Уровень надзора

Метка изображения

Ограничивающая рамка

Попиксельная разметка

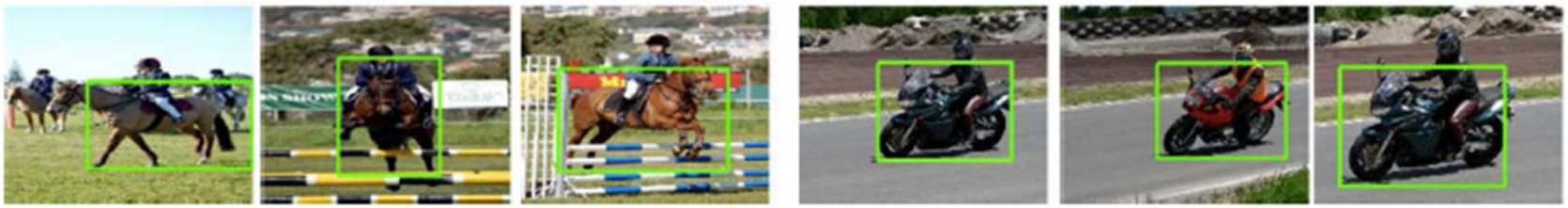
Части объекта



Типичные результаты

aeroplane bicycle

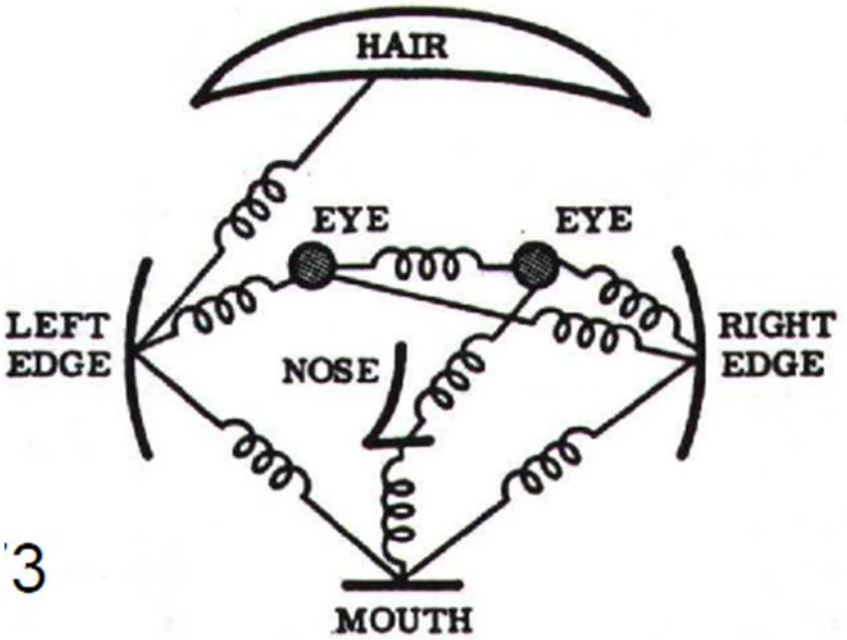
car cow



horse motorbike

Структура из пиктограмм

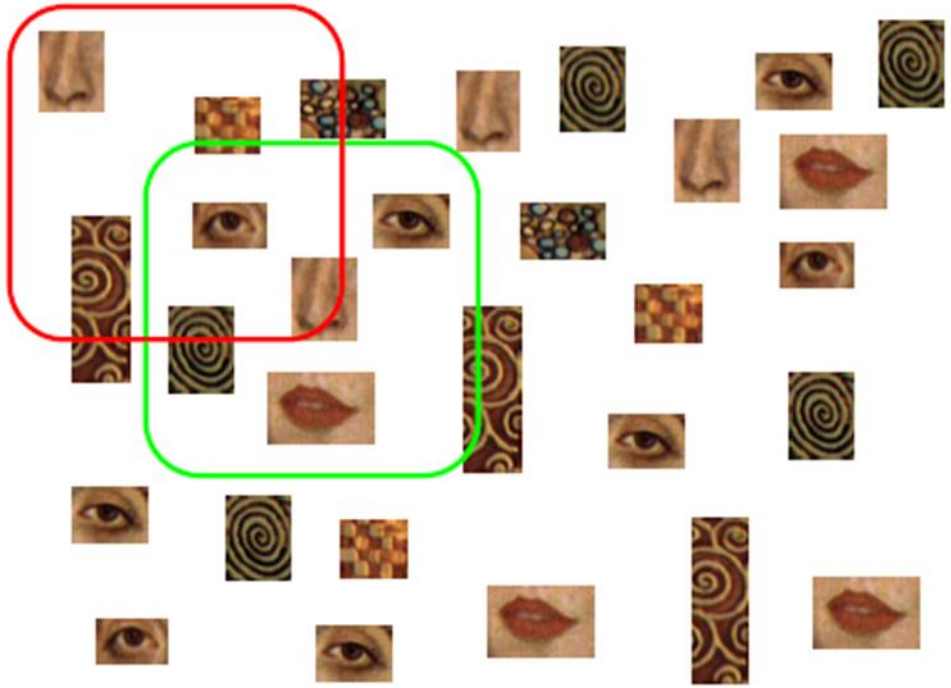
- Интуитивная модель объекта
- Состоит из 2х компонентов:
  - Частей (2D фрагменты изображения)
  - Структуры (конфигурация частей)
- Впервые была предложение Fischler & Elschlager 1973




Ограничения на конфигурацию
Microsoft Research

Решение для сложного фона

- Используем окно
  - Если положение окна точное, тогда посторонние объекты не попадают в окно
  - Сканируем окном всё изображение для поиска объектов
  - Меняем размер окна для поиска объектов разного масштаба

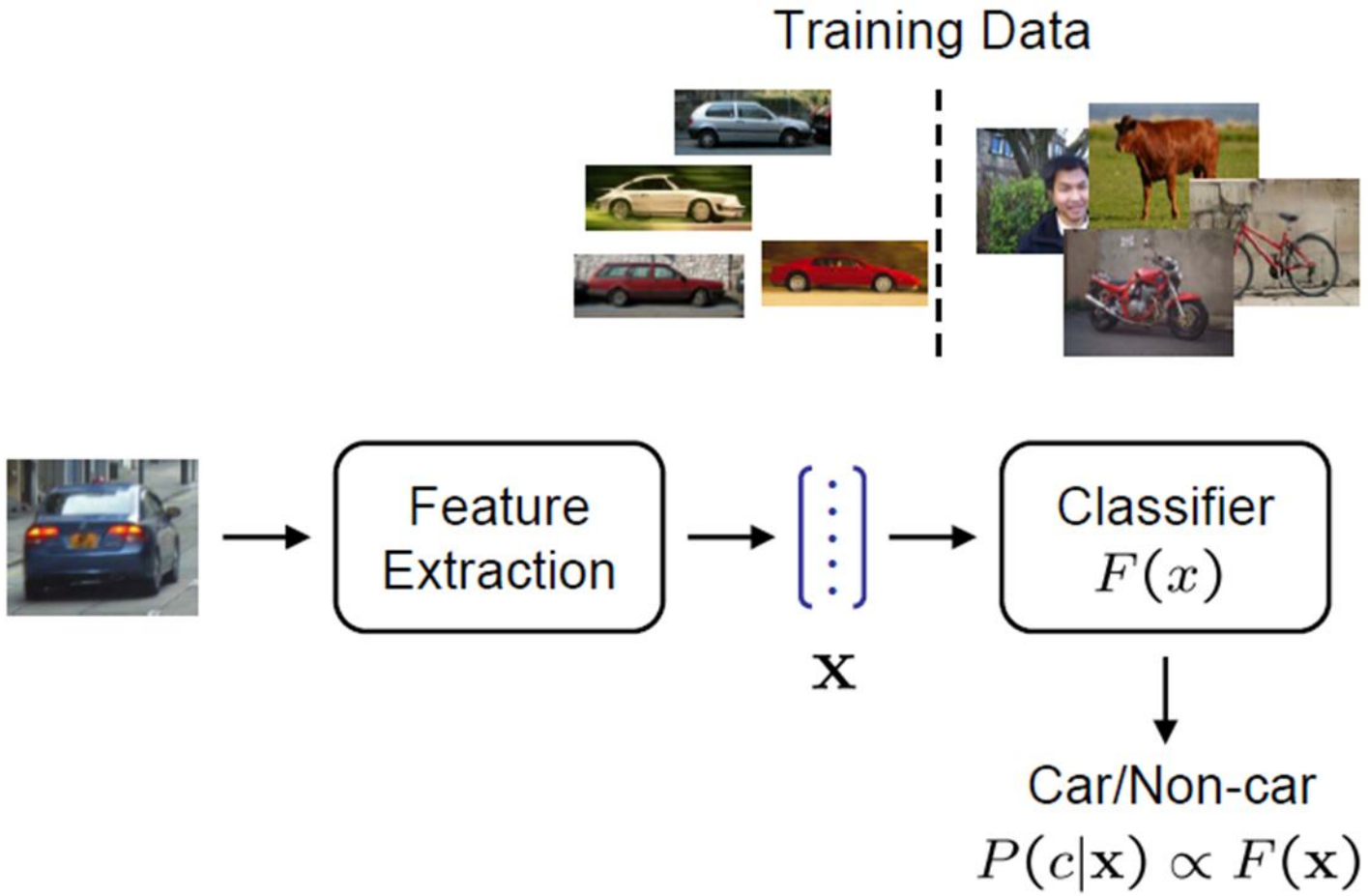


Microsoft Research

Пример: поиск лиц

- Подход на основе скользящего окна (sliding window).
- Сканируем изображение окном поиска и применяем классификатор к каждому положению
- Можем использовать любой классификатор изображений, но к окну

Схема метода

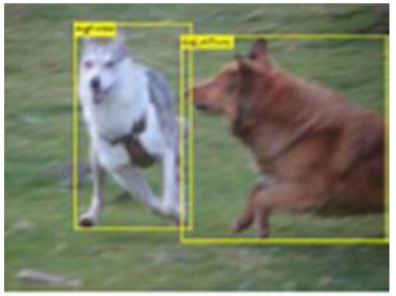


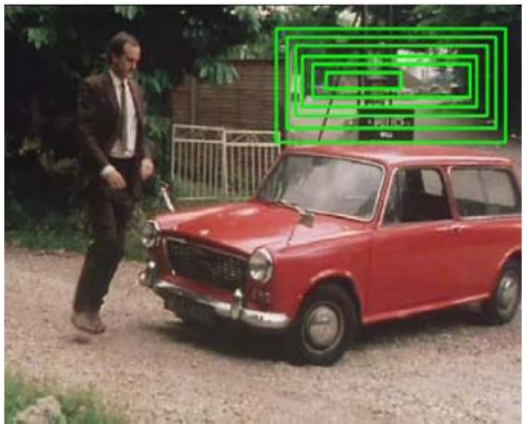
- Признаки обычно придумывают
- Классификатор обучается по данным

Microsoft Research

Недостатки скользящего окна

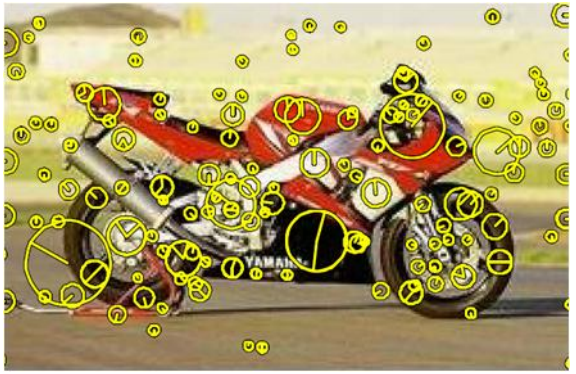
- Соотношение размеров рамки
- Дискретность сдвига
- Частичное перекрытие
- Множественные отклики

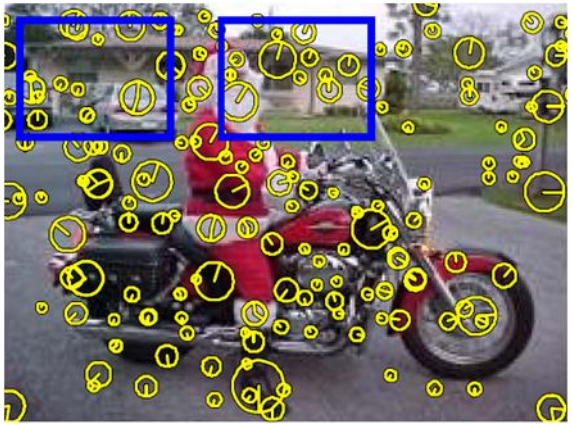




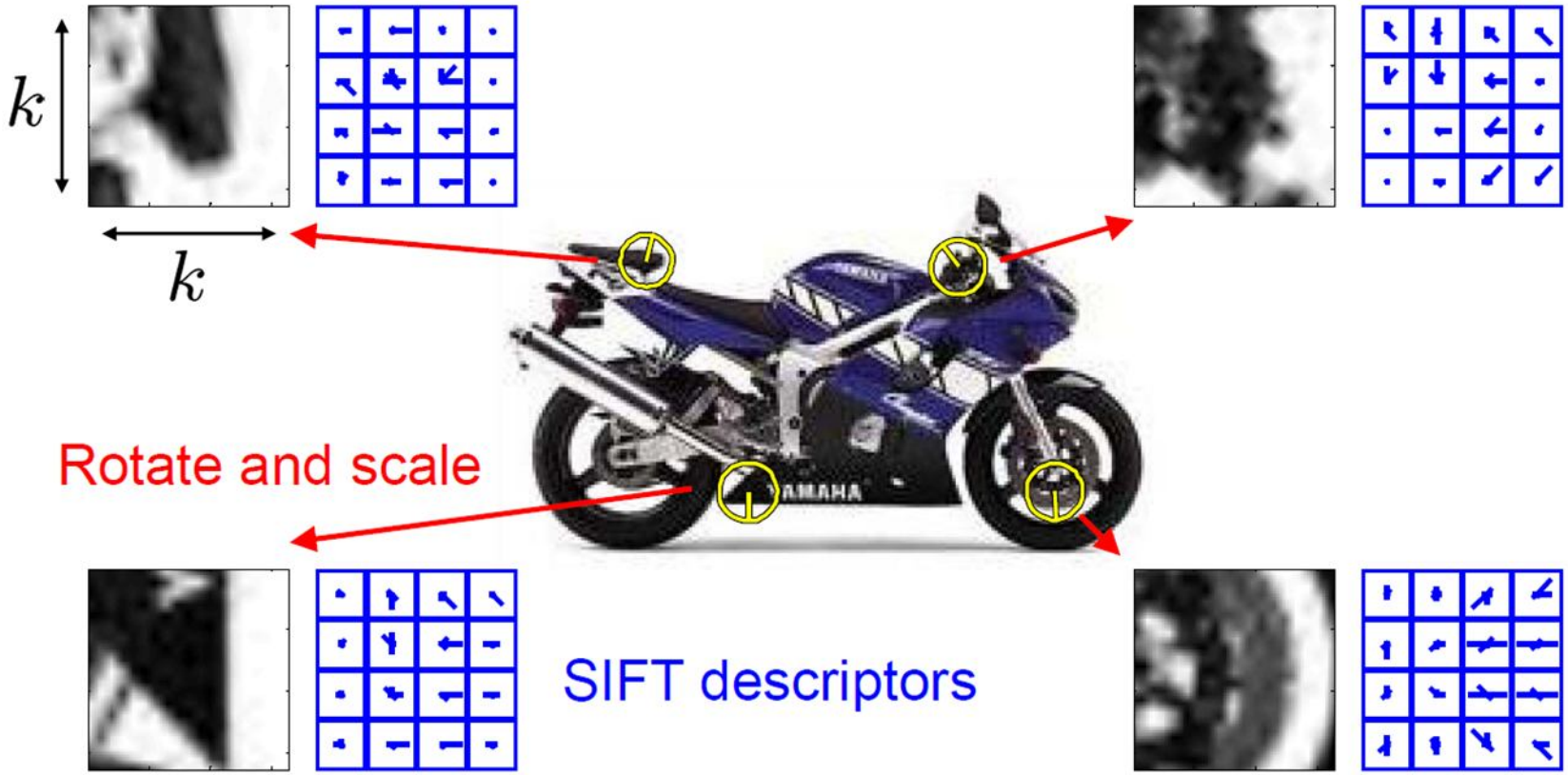
Мешок слов

- Поиск локальных особенностей (пр.: DoG, LoG, Harris-Laplace)
- Опишем каждую точку хорошим дискриптором (пр. SIFT, размерность 128)
- Необходимо описать содержимое скользящего окна вектором фиксированной длины для классификации:
  - Сопоставим дескрипторы общему словарю визуальных слов
  - Опишем содержание гистограммой по визуальным словам – «мешком слов»



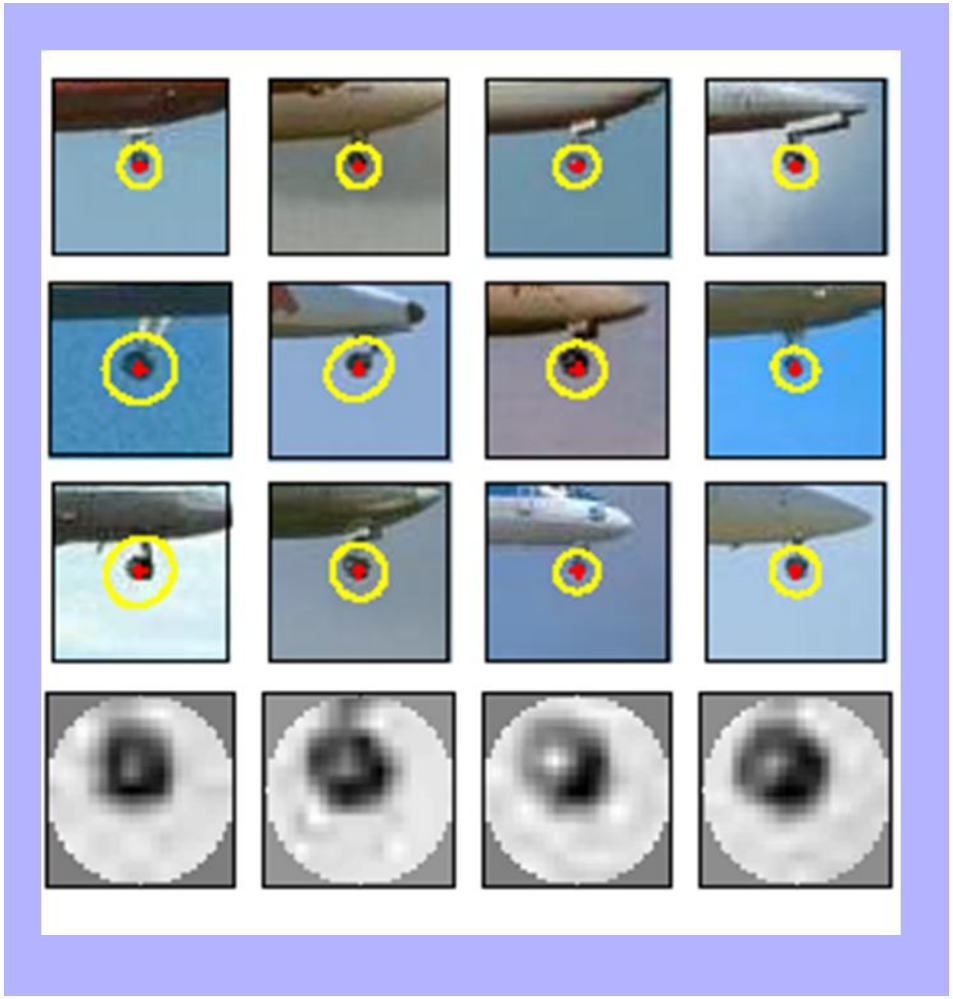


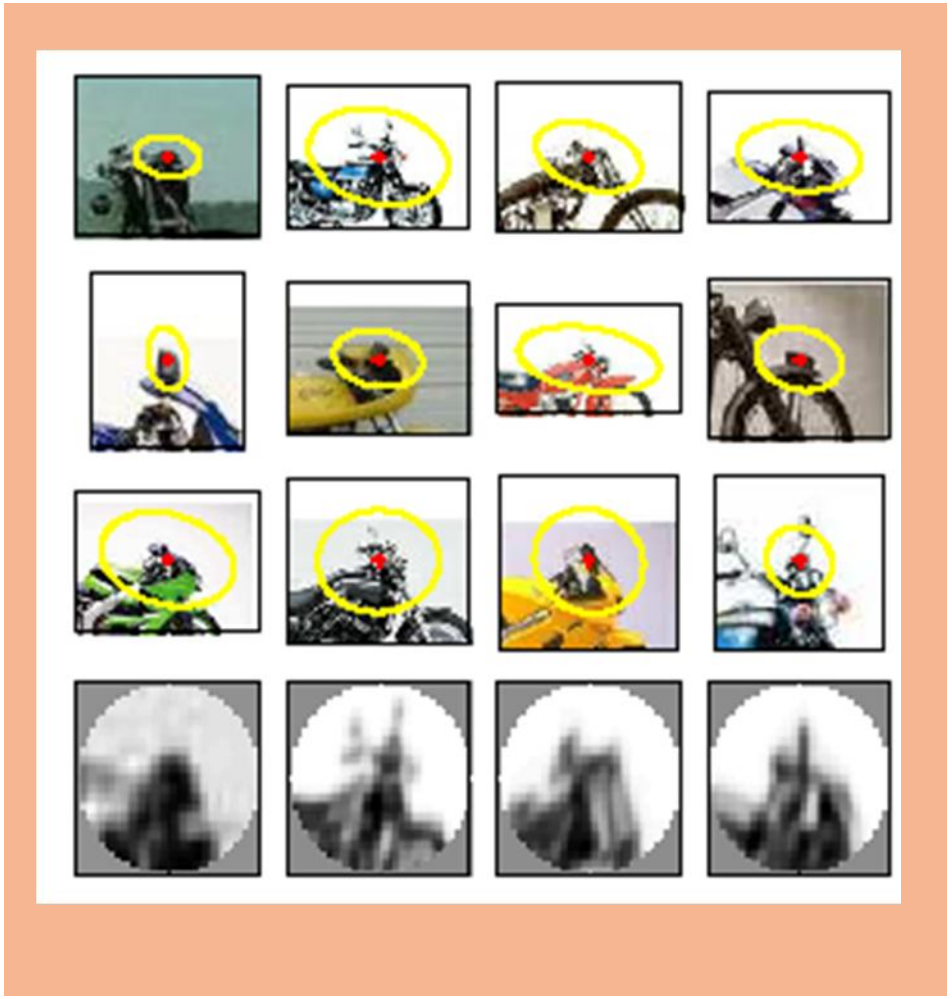
Дескриптор области



- Нормализуем области к фиксированному размеру и форме
- Опишем каждую область дескриптором SIFT
- Квантуем вектора по визуальным словам (пр. K-средних)

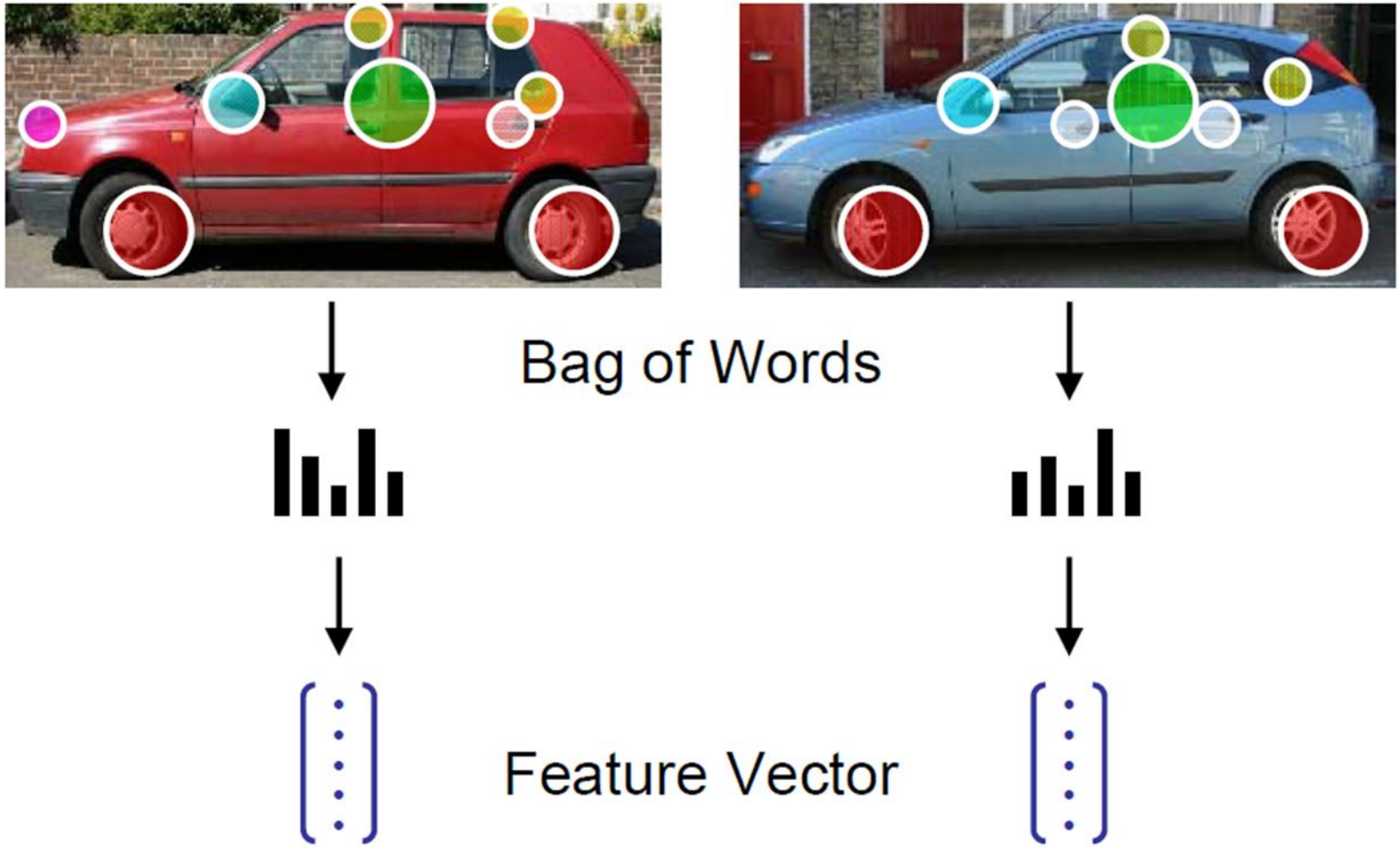
Примеры визуальных слов





Sivic et al. 2005

«Мешок слов»



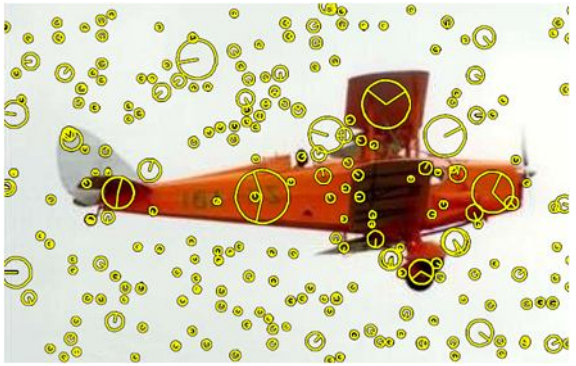
- Опишем содержание области гистограммой частот визуальных слов (вектор фиксированной длины)

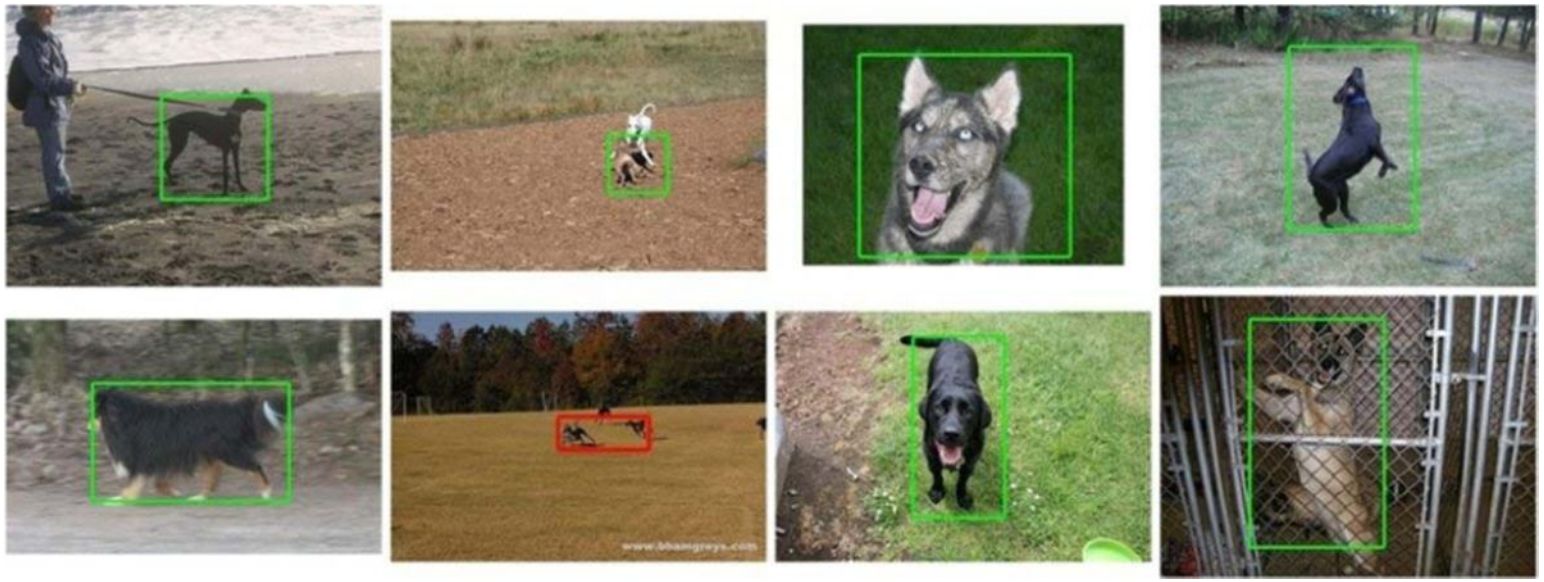
Свойства «мешка слов»

- Визуальные слова – «характерные» (“iconic”) фрагменты изображения
- Детектор и дескриптор SIFT обеспечивает инвариантность к локальным поворотам и масштабированию
- Переход к частотам отбрасывает пространственную информацию и обеспечивает инвариантность к относительному расположение фрагментов
  - Инвариантность к конфигурации

Простой детектор

- Скользящее окно
- «Мешок слов»
- Классификатор – MOB(SVM)





Оценка «мешка слов»

- Плюсы
  - Нет явного моделирования пространственной информации ⇒ инвариантность к положению и ориентации в изображении
  - Вектор фиксированной длины ⇒ применение стандартных методов машинного обучения

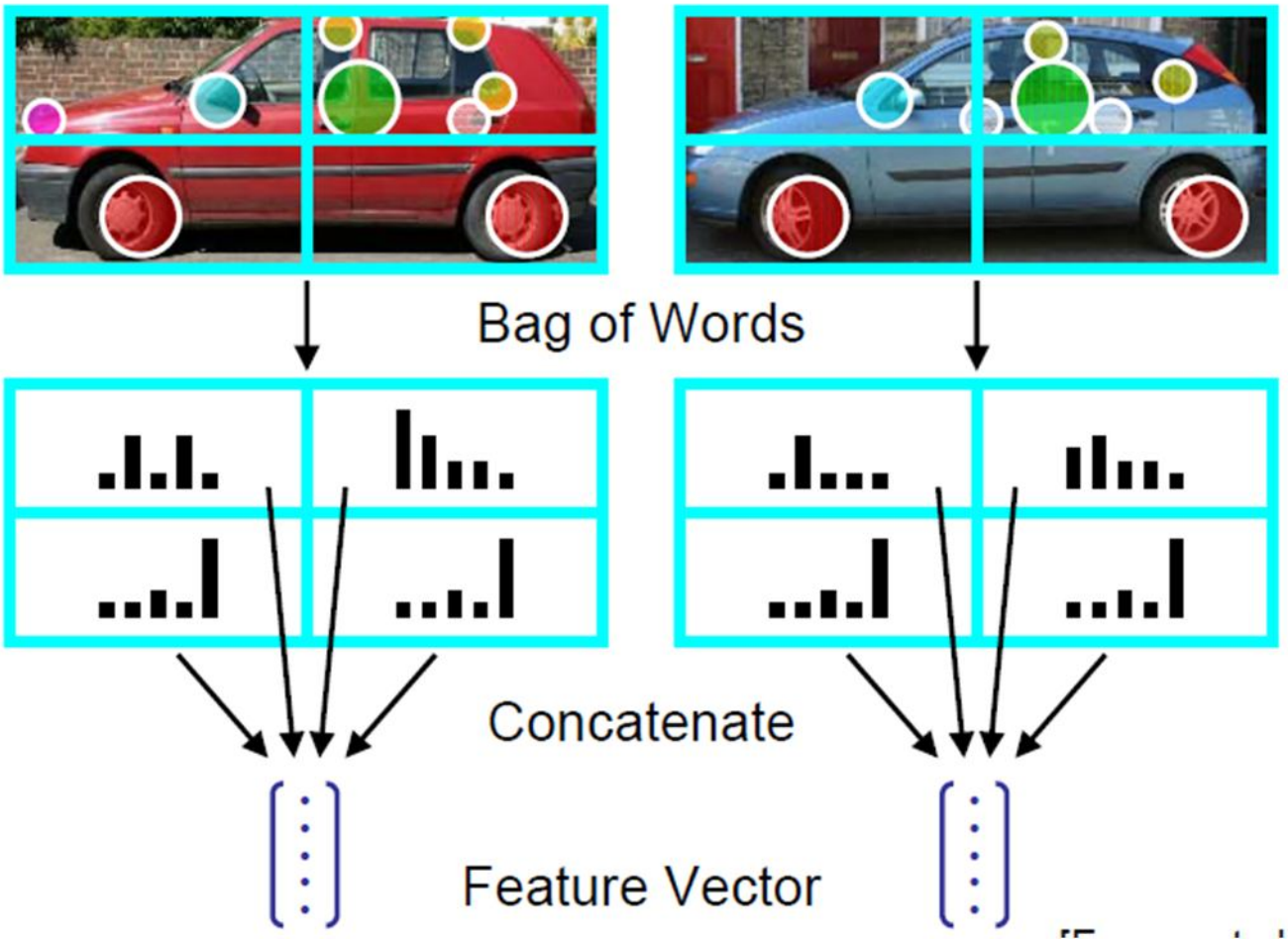
- Минусы
  - Нет явного моделирования пространственной информации ⇒ хуже различающая способность (discriminative power)

Microsoft Research

Развитие «мешка слов»

- Отталкиваются от «мешка слов» для области
  - Не учитывается пространственная информация
  - Скользящее окно

Учет расположения

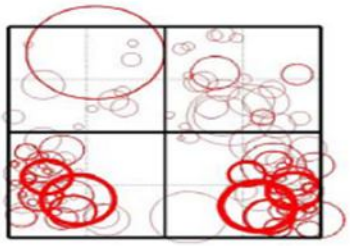


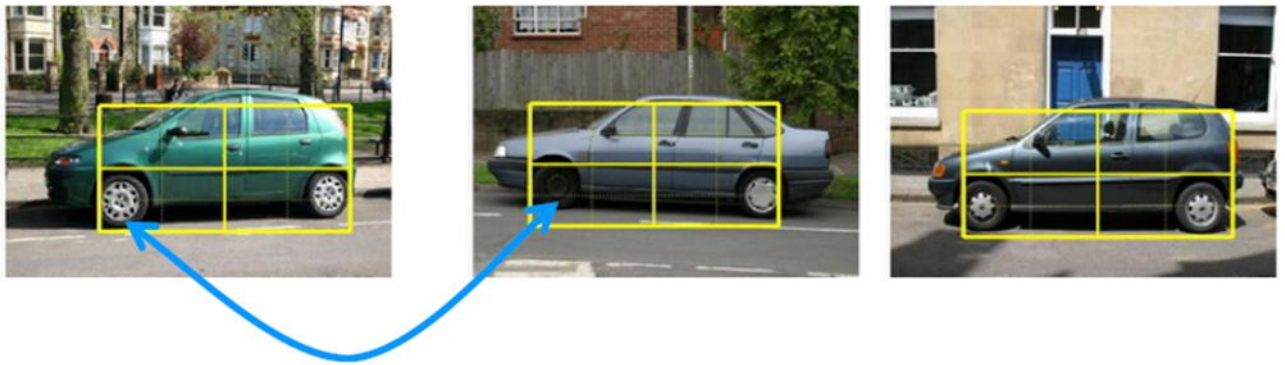
- Сохраняем вектор фиксированной длины

Lazebnik et al. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. CVPR 2006.

Учет расположения

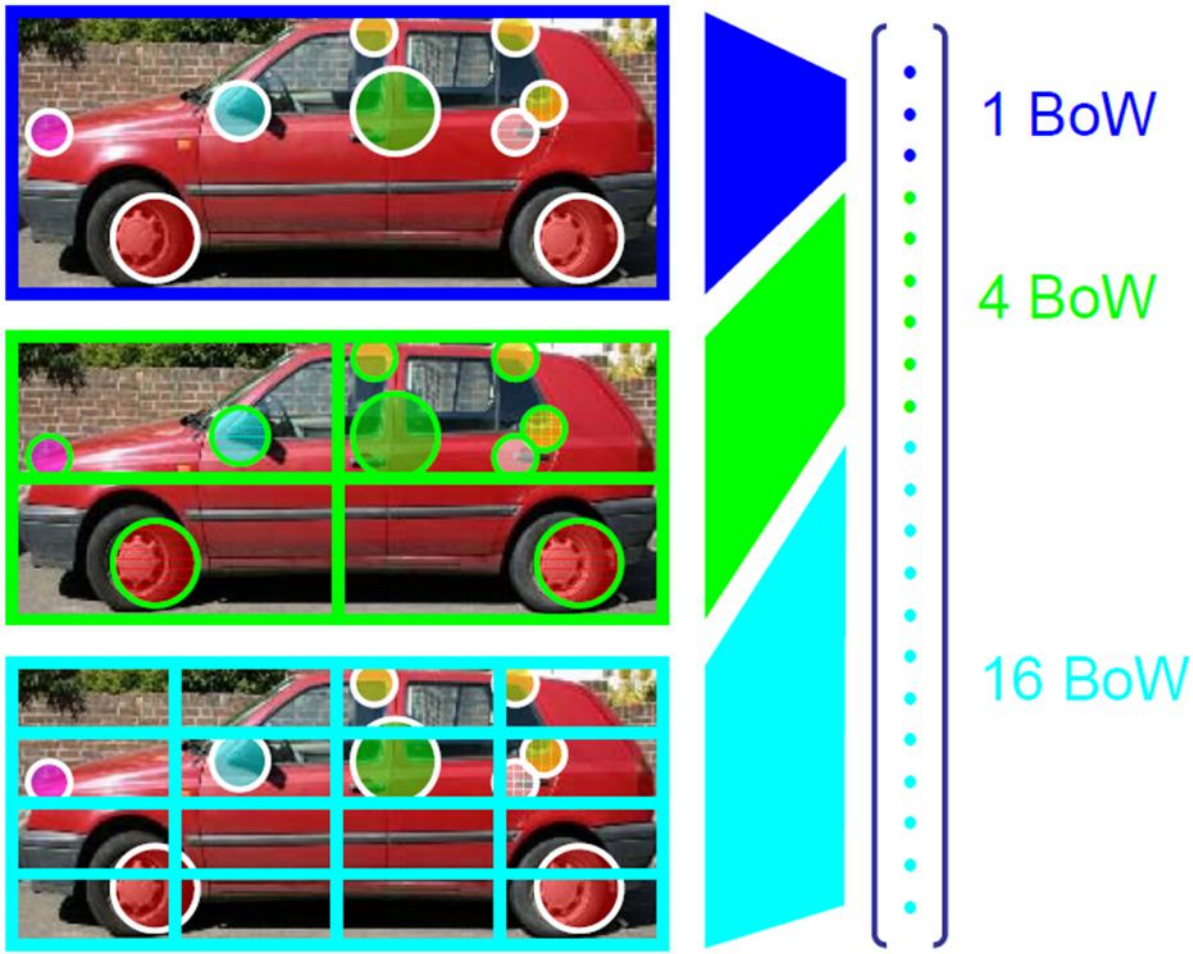
- parameter: number of tiles





- Если в словаре V слов, то в представлении 4V

Пирамиды



- Можно использовать пирамиду

Плотные визуальные слова

- Зачем извлекать только отдельные фрагменты?
  - Хорошо для достижения высокой инвариантности
  - Не очень актуально для скользящего окна
  - Можно извлекать фрагменты плотно, с перекрытием

• Больше деталей, жертвуем инвариантностью

• Пирамидальная гистограмма визуальных слова (PHOW)

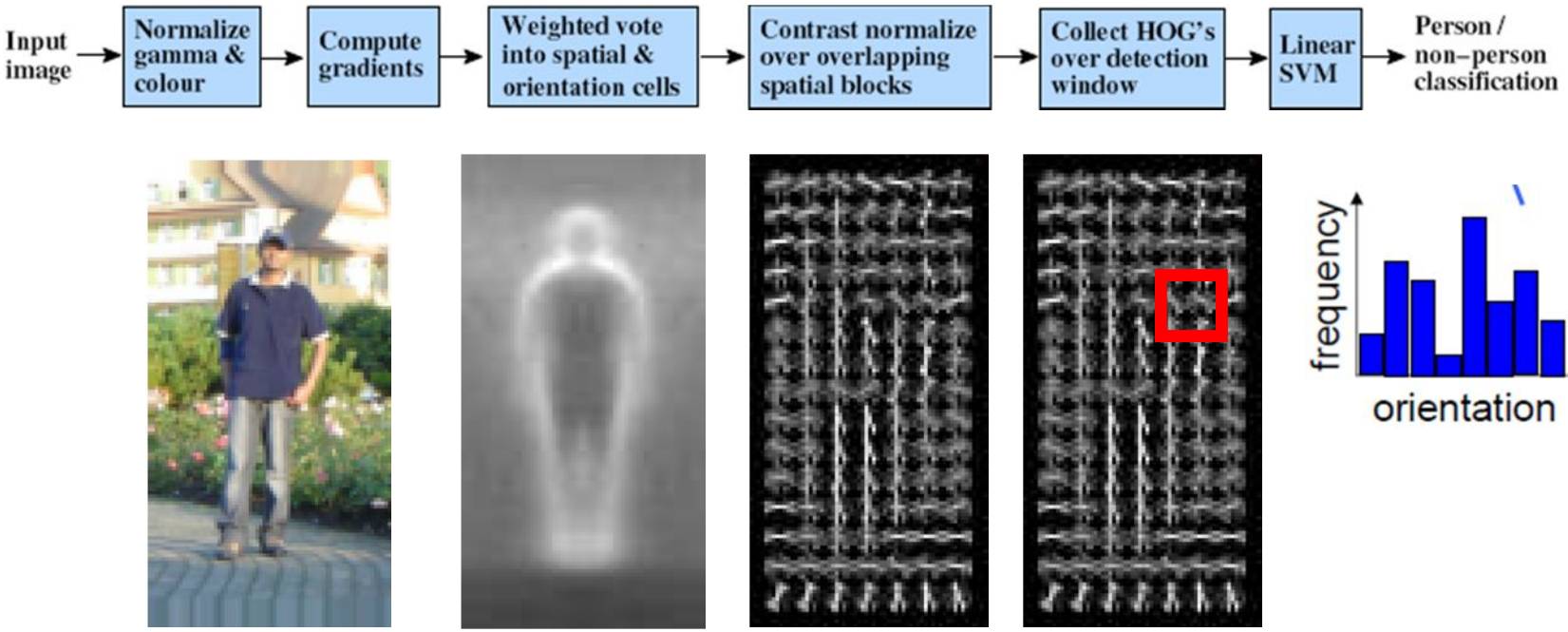
Microsoft Research

Поиск пешеходов

- Цель: обнаружить (локализовать) стоящих людей в изображении
  - Скользящее окно
  - Обучаем бинарный классификатор – содержит ли окно фигуру стоящего человека
  - Гистограмма ориентированных градиентов (HOG)
- Хотя схема HOG + SVM изначально была предложена для пешеходов, она успешно применялась в дальнейшем к разным категориями объектов

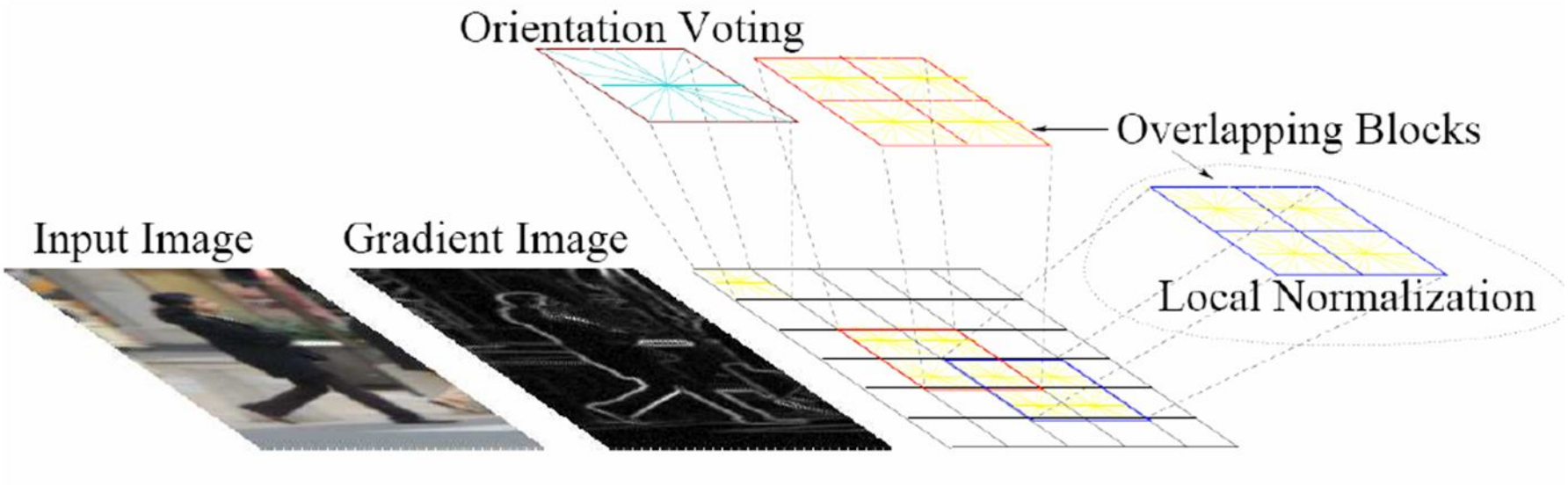
Navneet Dalal, Bill Triggs,Histograms of Oriented Gradients for Human Detection, CVPR 2005

Схема



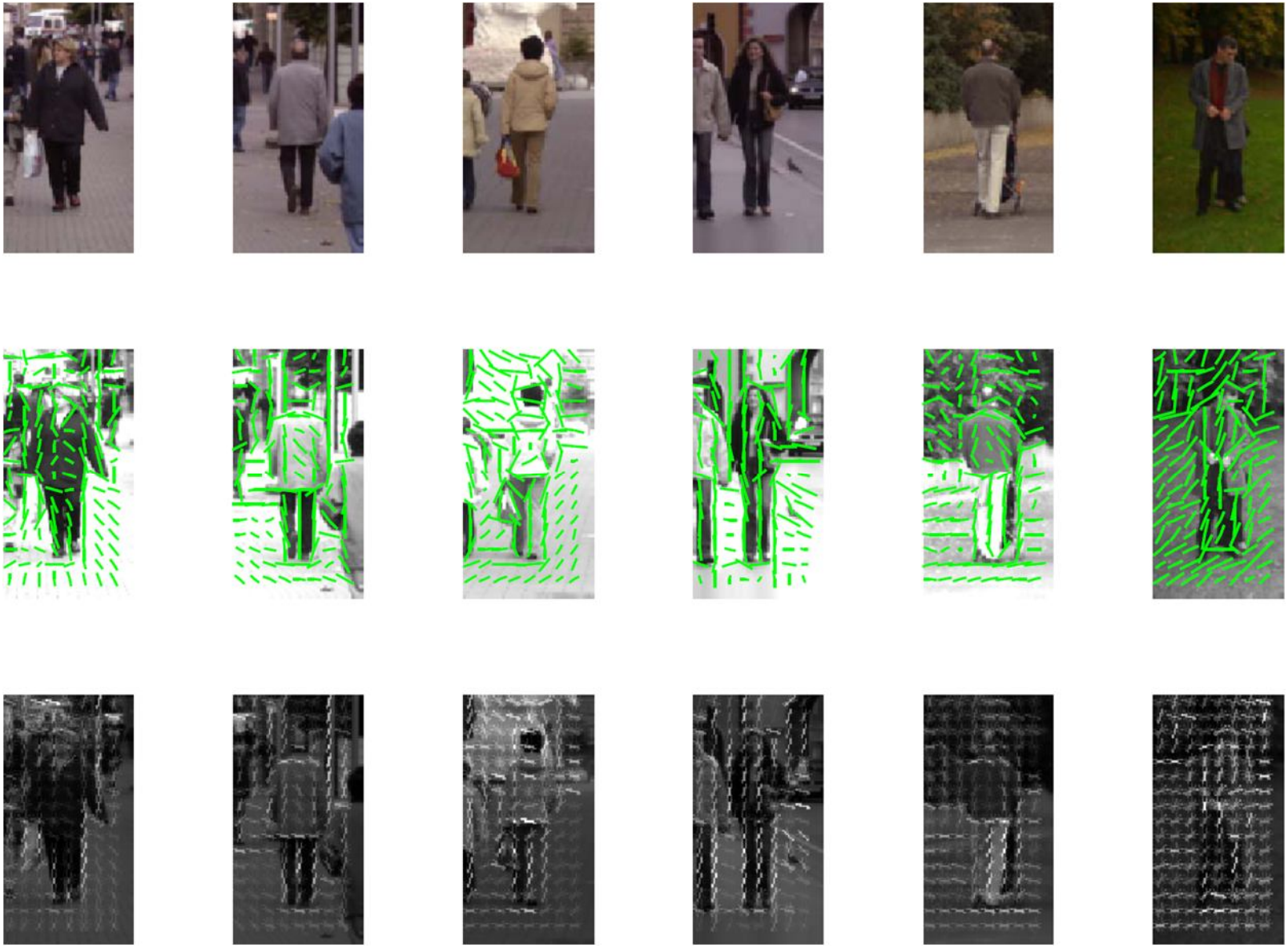
- Раскладываем окно 64 x 128 пикселей на ячейки 8 x 8 пикселей
- Каждую ячейку описываем гистограммой по 8 направлениям (пр. углы в интервале 0-180 градусов)

Схема



- Добавляем второй уровень:
  - Выделяем блоки 2x2 с перекрытием
  - Нормализуем гистограммы в каждом блоке
  - Это дает большую пространственную поддержку
- Размер вектора = 16 x 8 (тайлинг) x 8 (ориентаций) x 4 (блоки с перекрытием) = 4096

Пример



SVM

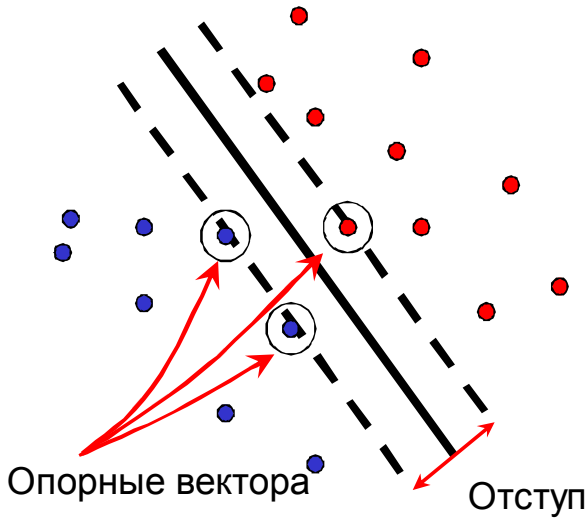
• Обучаем линейную SVM:

$$f(x) = w^T x + b$$

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

• Опорные вектора с положительными и отрицательными весами

• Чем фактически в нашем случае являются ***x*** ?



SVM

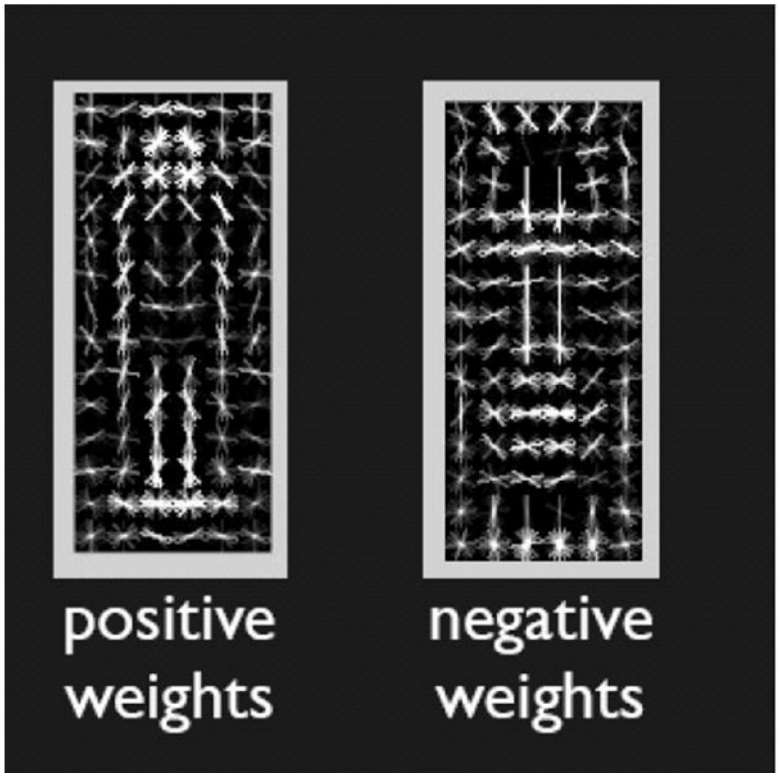
• Обучаем линейную МОВ:

$$f(x) = w^T x + b$$

$$\mathbf{w} = \sum_i \alpha_i y_i \mathbf{x}_i$$

• Опорные вектора с положительными и отрицательными весами

• Линейную МОВ можно рассматривать как линейный фильтр изображения



Microsoft Research

What do negative weights mean?

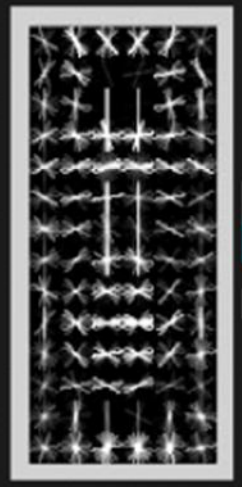
$$wx > 0$$

$$(w_+ - w_-)x > 0$$

$$w_+ > w_-x$$

pedestrian model

>



pedestrian background model

Complete system should compete pedestrian/pillar/doorway models

Discriminative models come equipped with own bg

(avoid firing on doorways by penalizing vertical edges)

Slide from Deva Ramanan

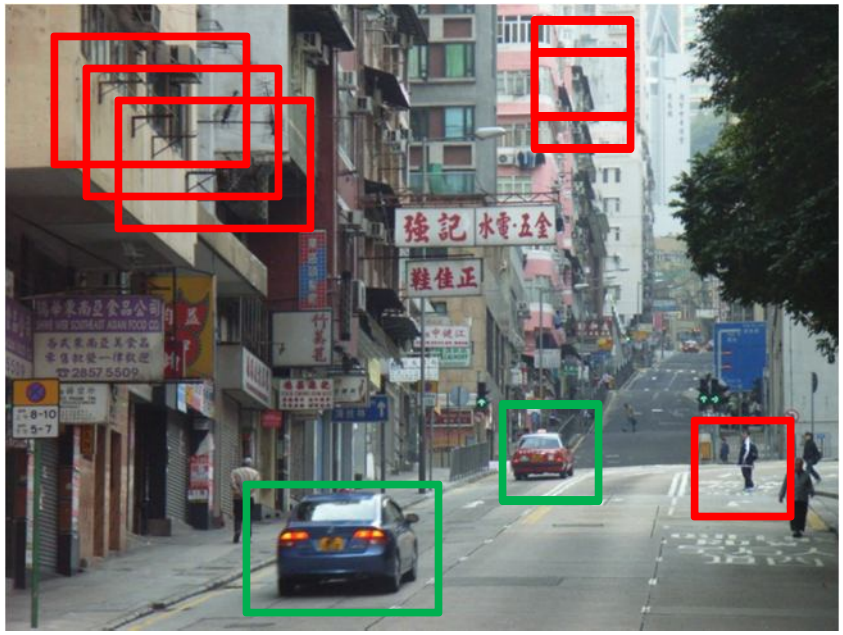
Обсуждение HOG + SVM

- Аналогично SIFT,учитывает свойства восприятия
- По сравнению с обучением только по краям:
  - Можно моделировать сложные соединения
  - Избегаем проблемы «раннего» обрезания по порогу
  - Учитываем слабые внутренние градиенты
- Методы с учетом только краёв устарели

ГОГ также дает вектор фиксированной длины, удобно для классификации

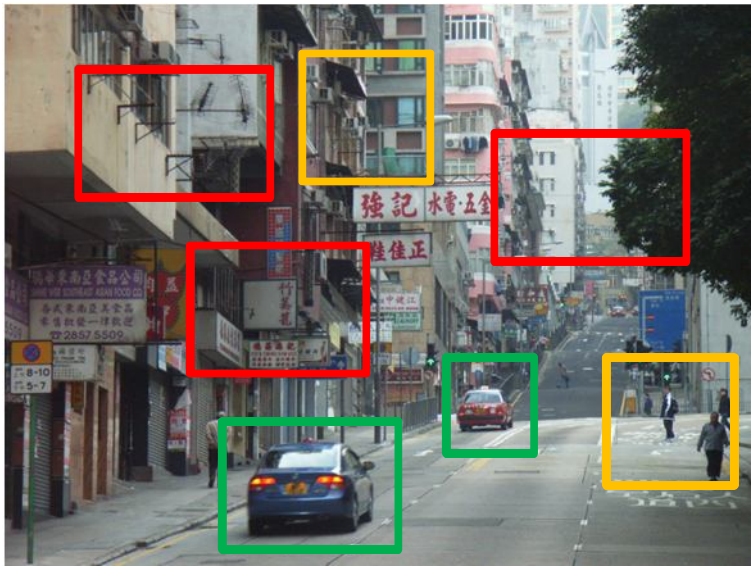
Обучение детектора

- Поиск объектов ассиметричная задача: объектов гораздо меньше, чем «не-объектов»



- Классификатору нужна очень низкая ошибка 2 рода (мало ложных обнаружений)
- Класс «не объект» очень сложный – нужно много данных для обучения

Бутстраппинг (Bootstrapping)



- Выбираем отрицательные примеры случайным образом
- Обучаем классификатор
- Применяем к данным
- Добавляем ложные обнаружение к выборке
- Повторяем

- Смысл:
  - Соберем ограниченную, но представительную выборку «не-обьектов»
  - Заставим классификатор сконцентрироваться на сложных отрицательных (**hard negative**) примерах

I. Laptev "Improvements of Object Detection Using Boosted Histograms« BMVC 2006

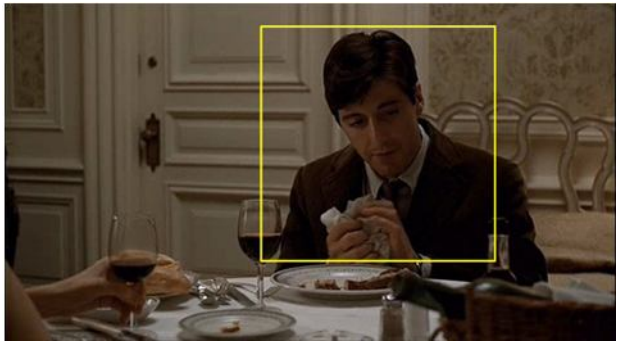
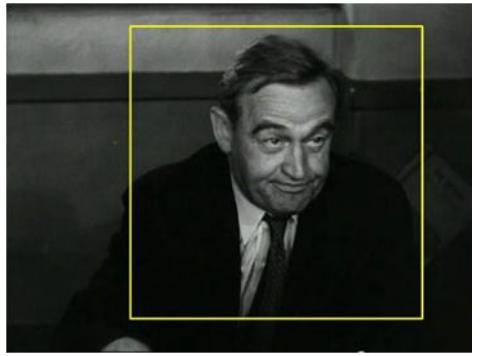
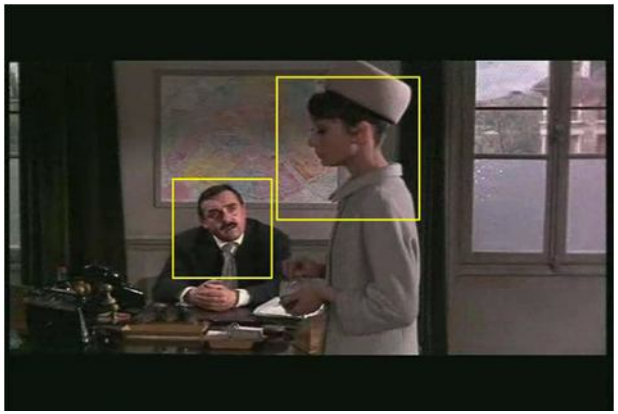
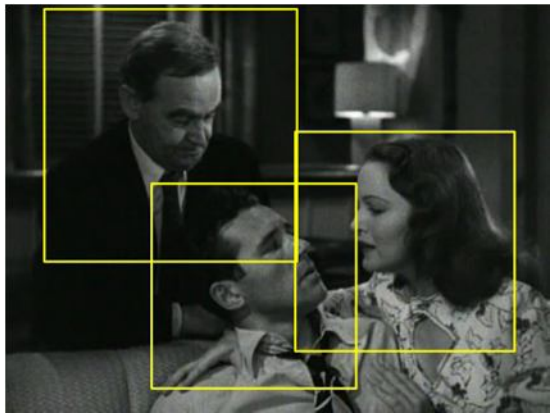
Пример

- Детектор «верхней части тела и головы»
- Обучающая выборка
  - 33 Hollywood2 фильмов
  - 1122 кадров с размеченным объектами
- Первая стадия (бутстреппинг)
  - 1607 размеченных объектов искажается (jittered) для получения 32k примеров
  - 55k отрицательных примеров выбирается из видео
- Вторая стадия (перетренировка)
  - 50k сильных отрицательных примеров выбираются

Положительные примеры



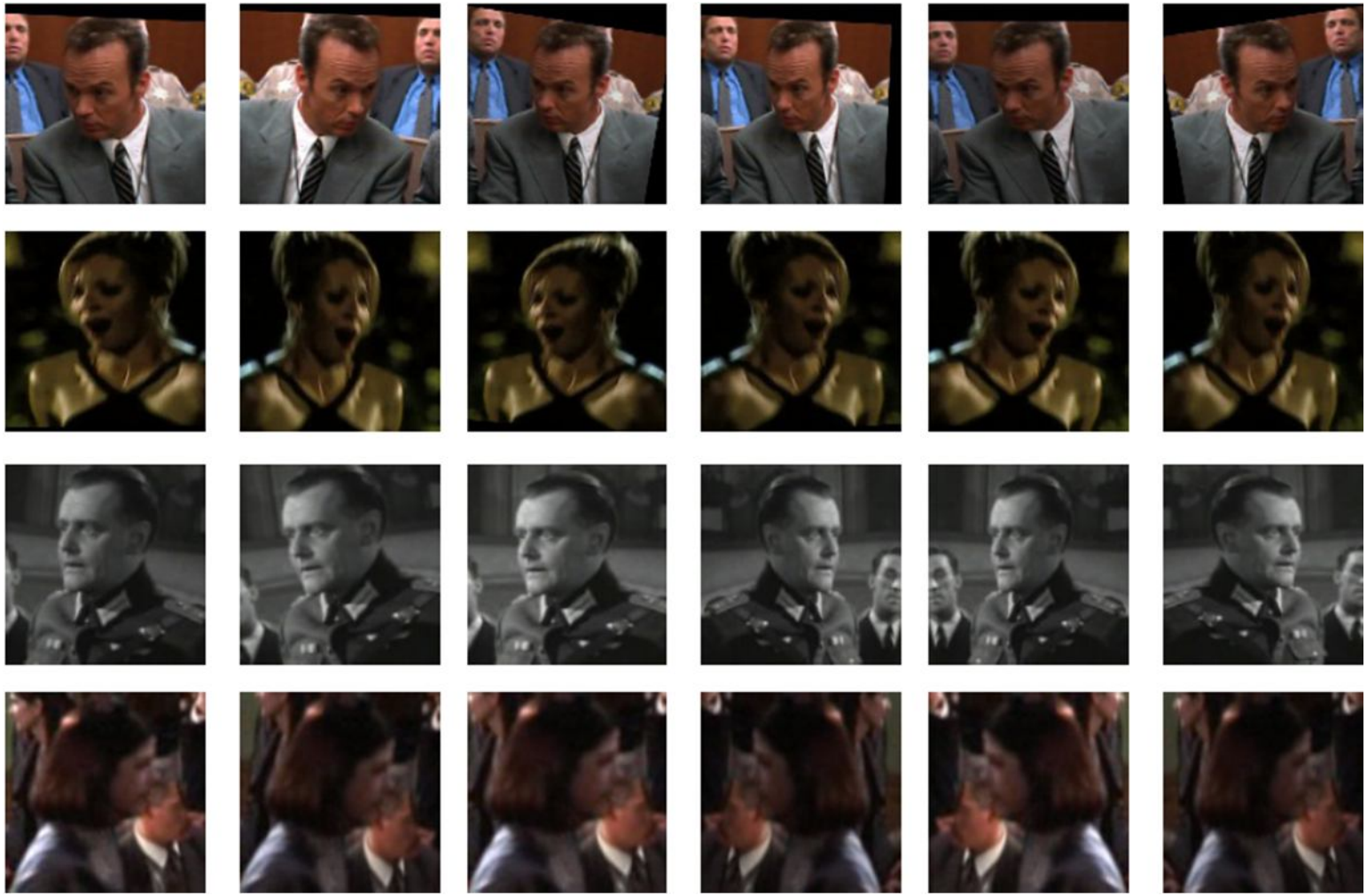



Положительные окна

- Внимание: похожие положение и ориентация

Искаженные примеры

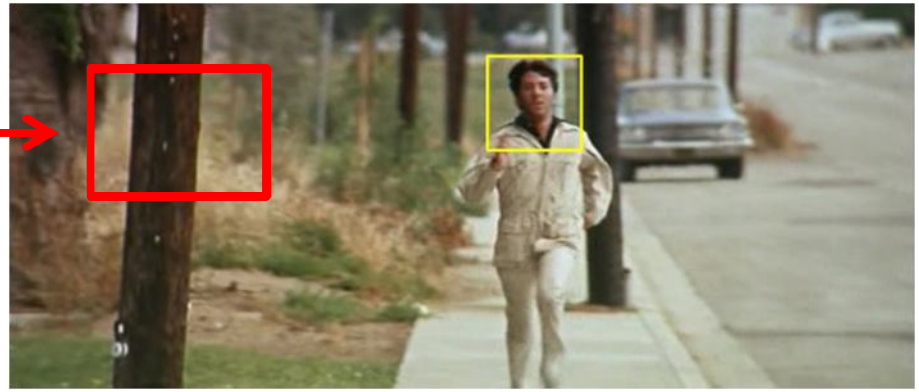


- Небольшие сдвиги, отображения, повороты, изменения масштаба

Random negatives

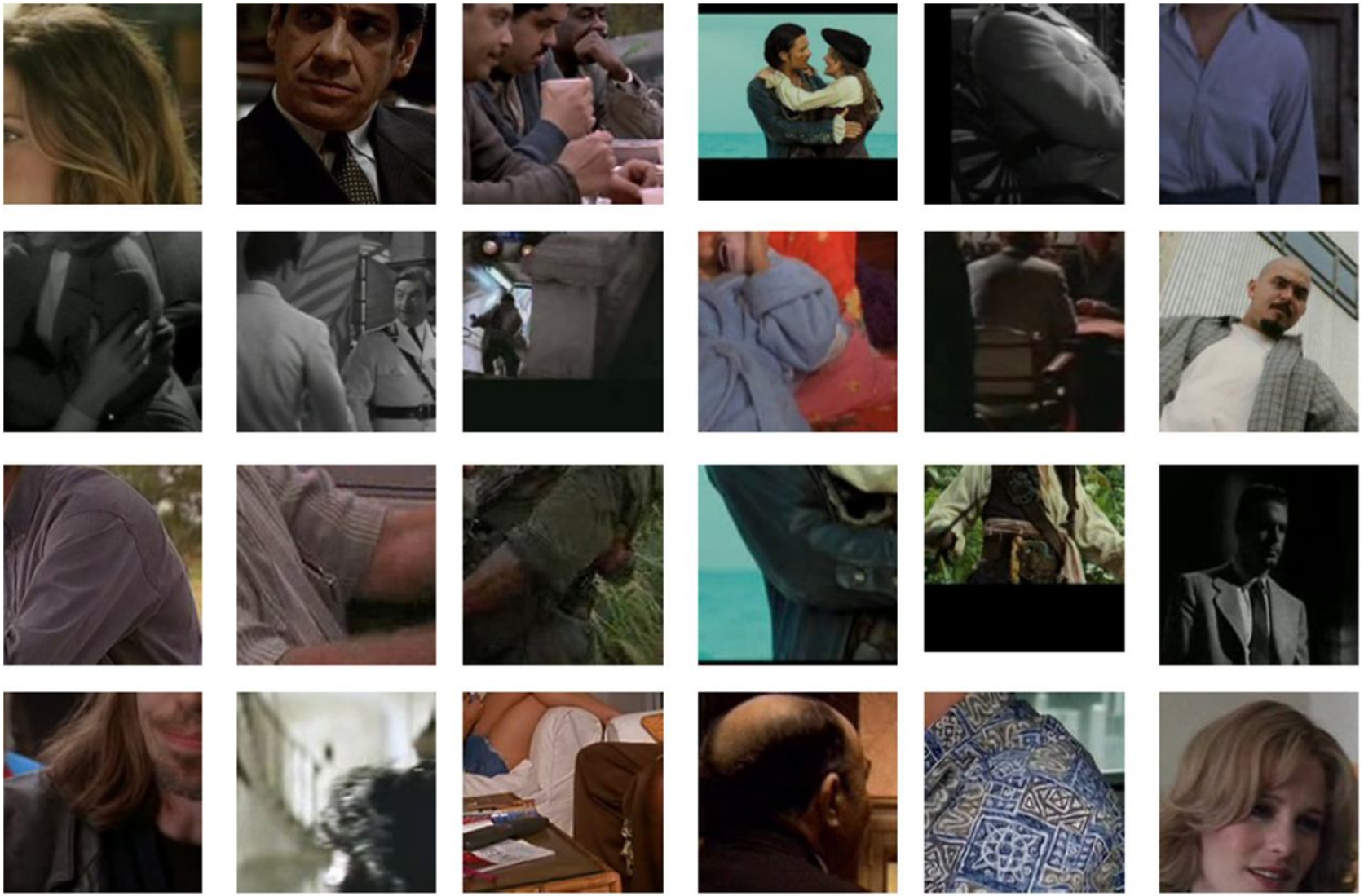
Первая стадия

Трудный отрицательный пример

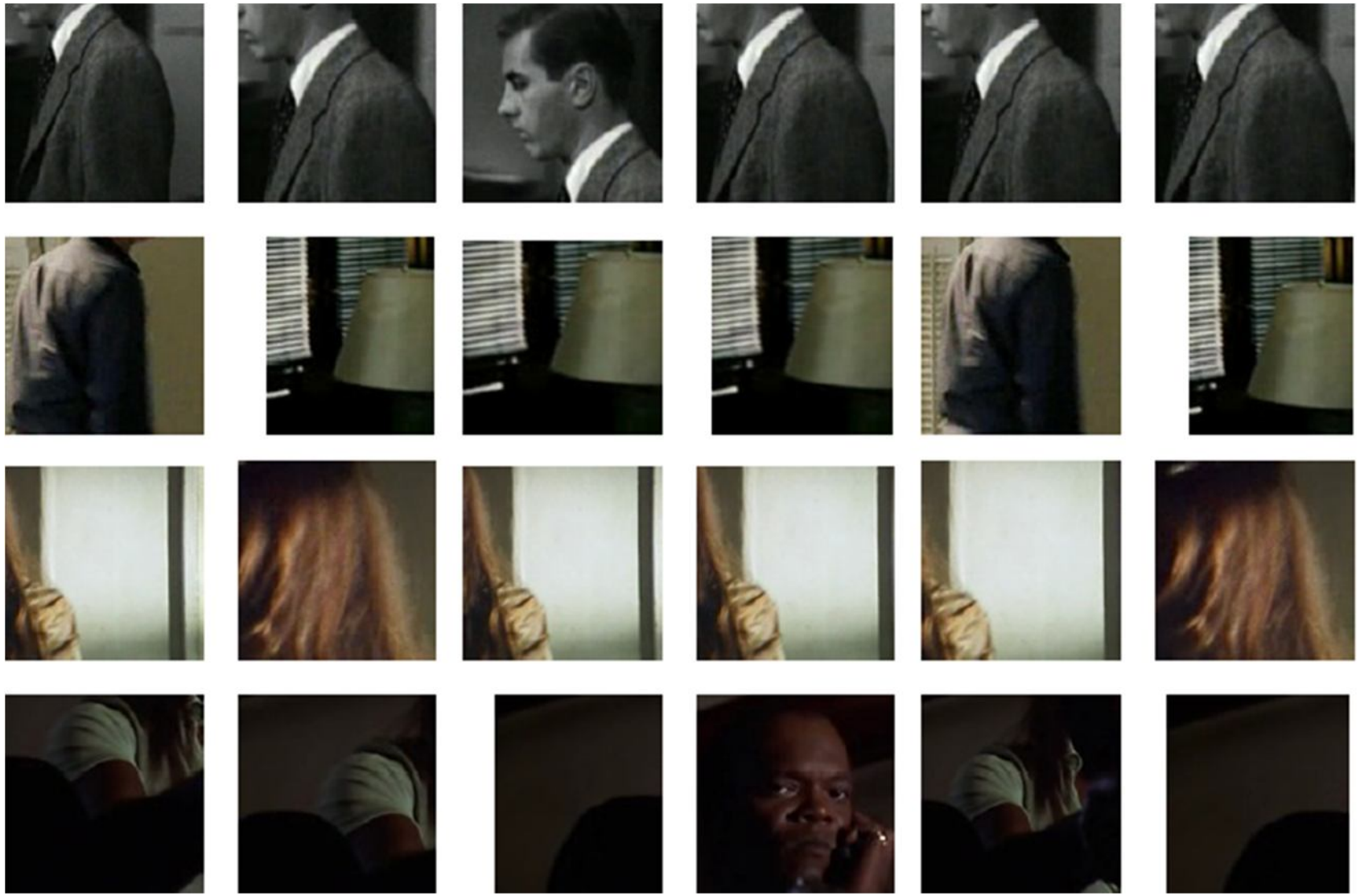


- Ищем ложные обнаружения с высоким рейтингом
- Используем их как трудные отрицательные примеры
- Затраты: # количество изображений x поиск в каждом

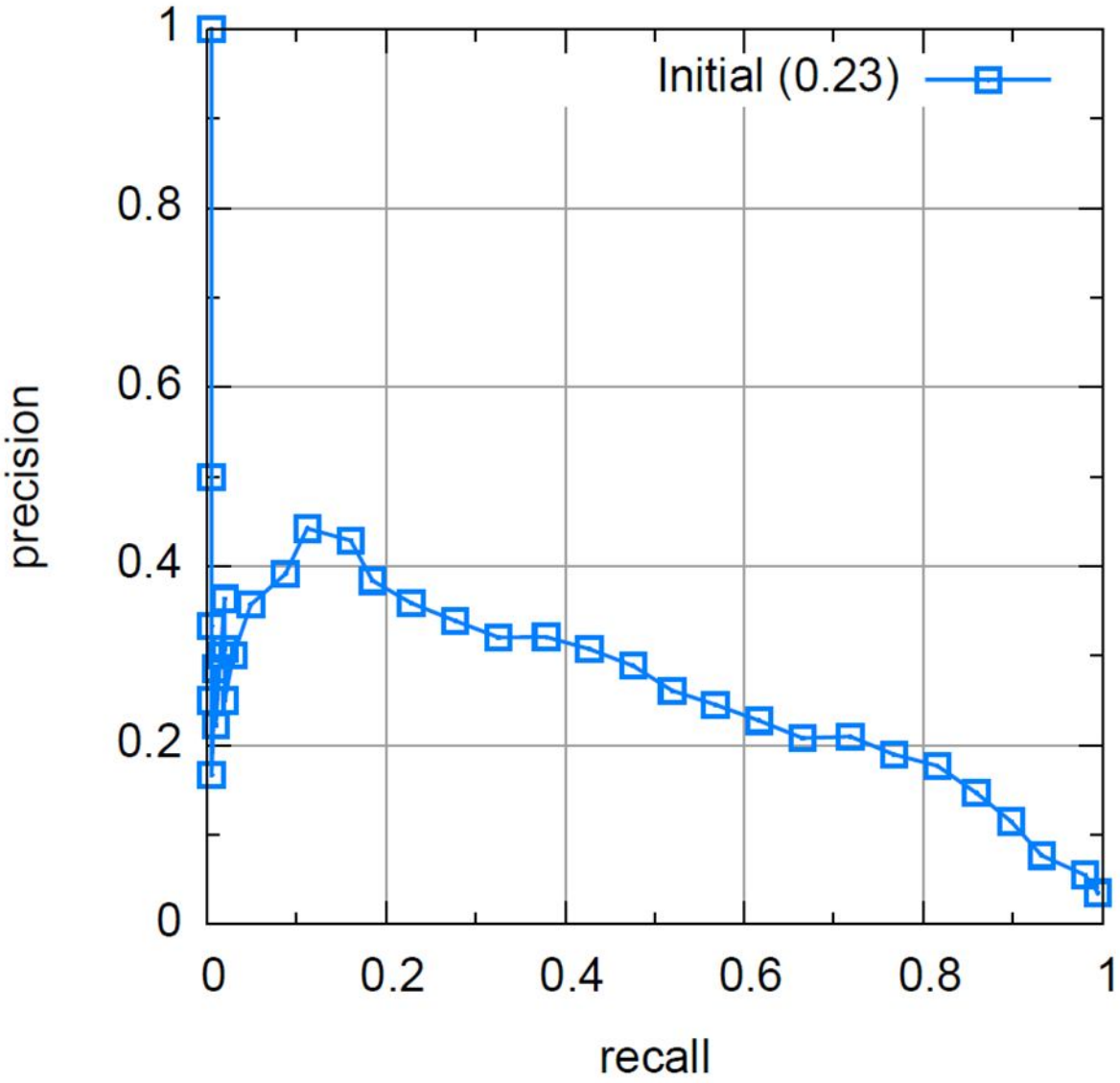
Трудные примеры



Трудные примеры

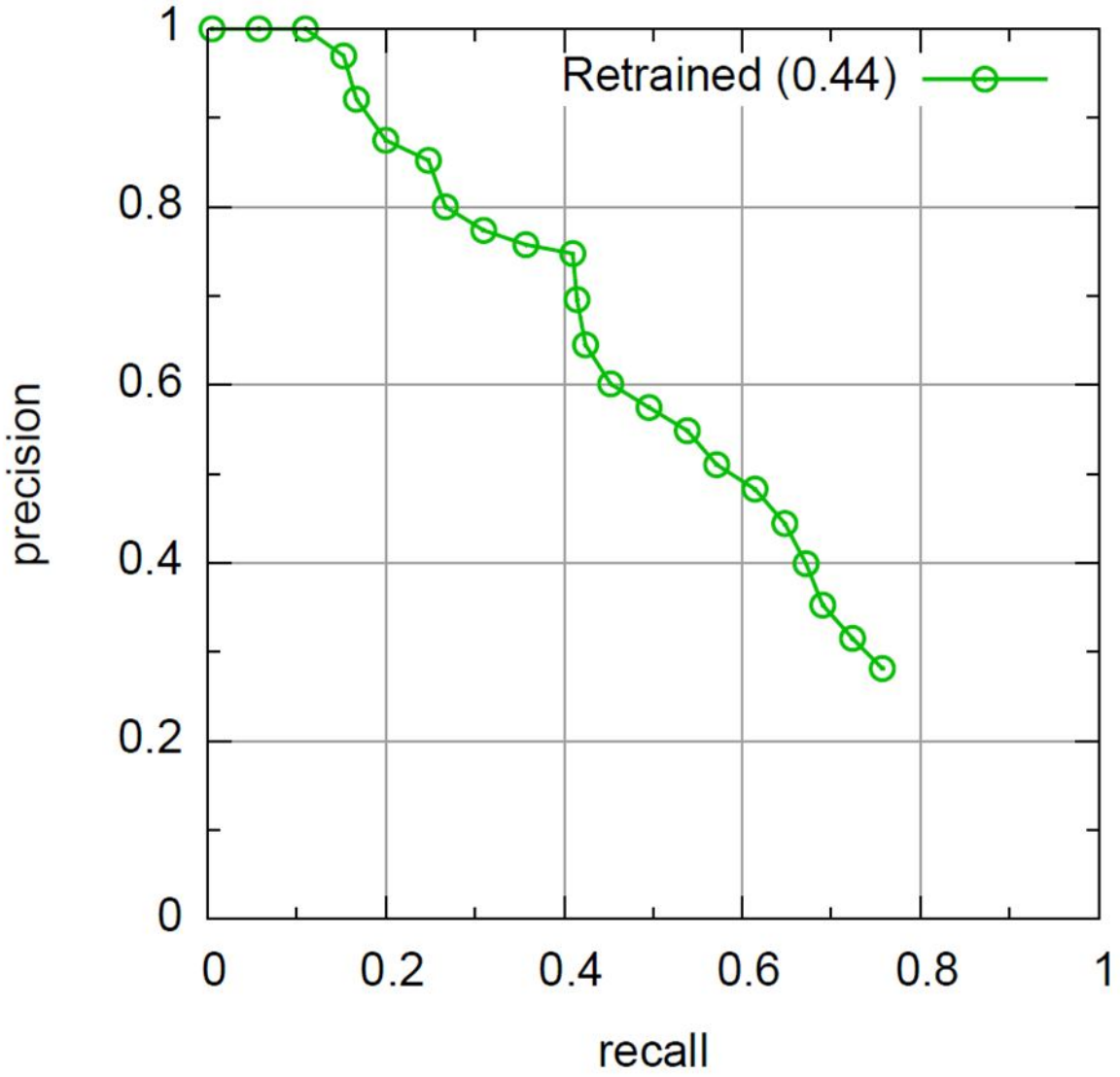


Измерение качества

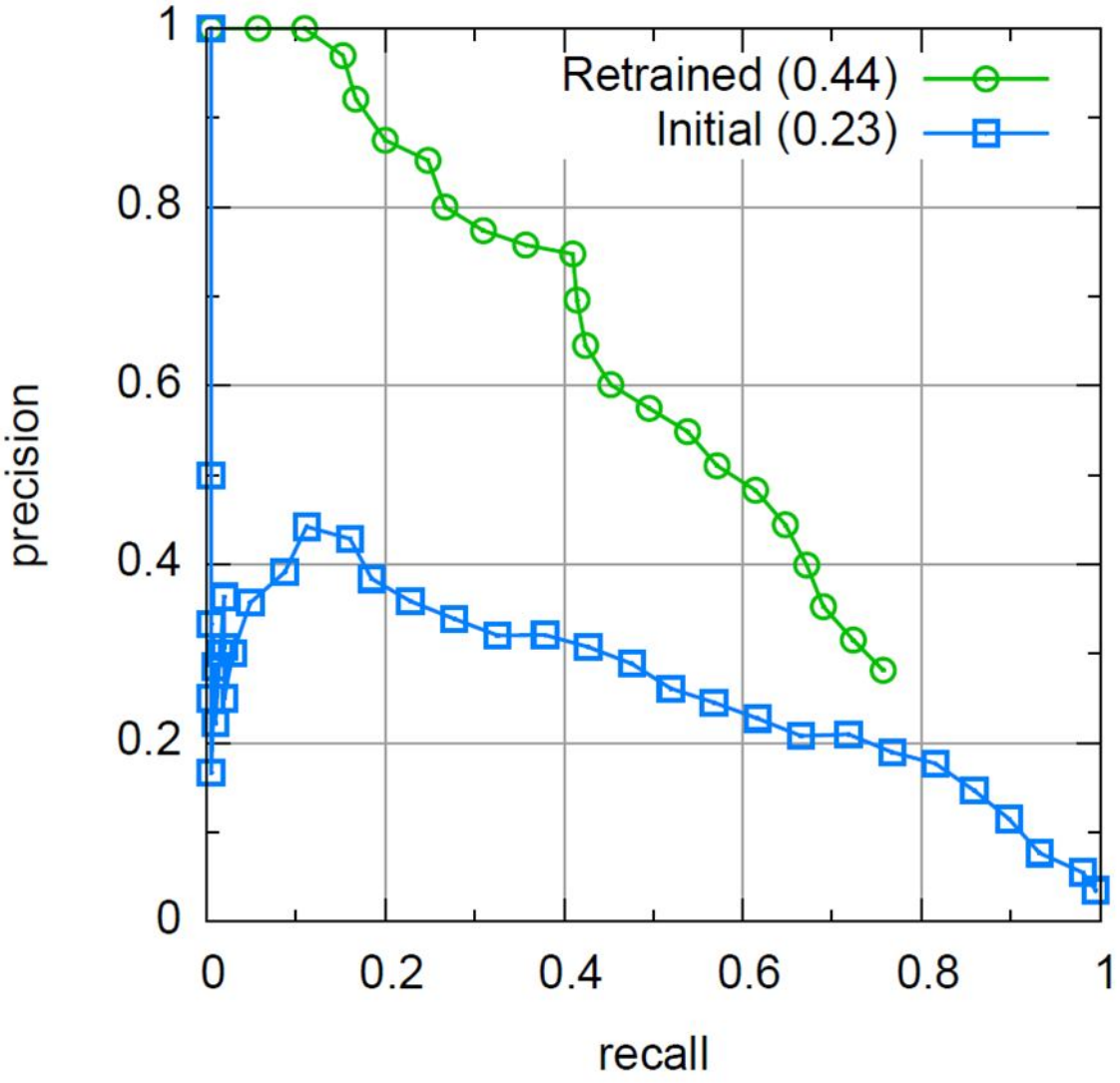


Microsoft Research

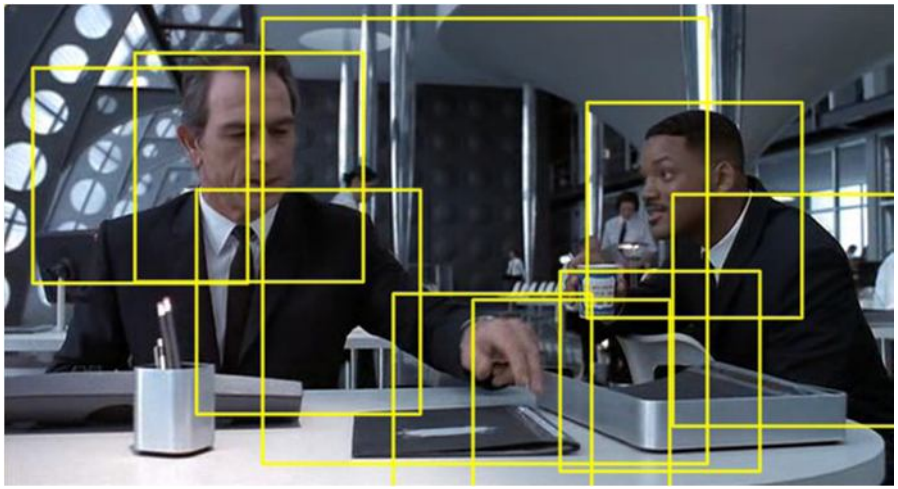
После перетренировки

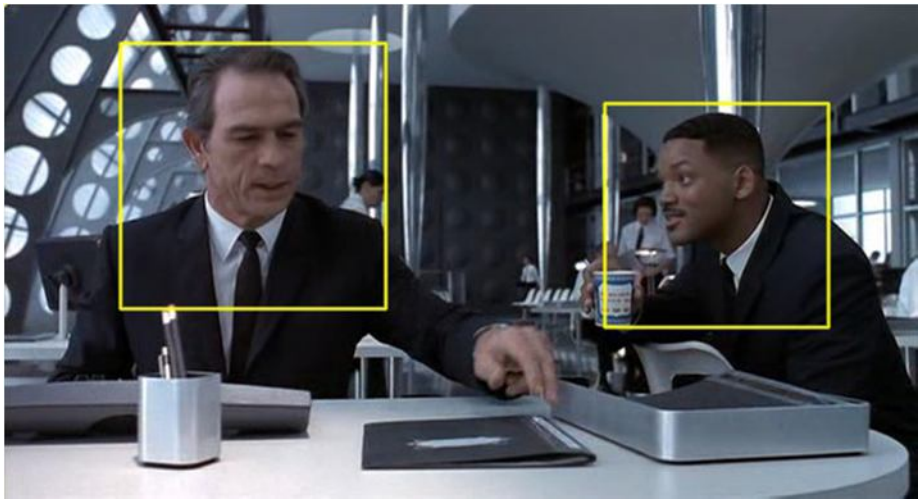


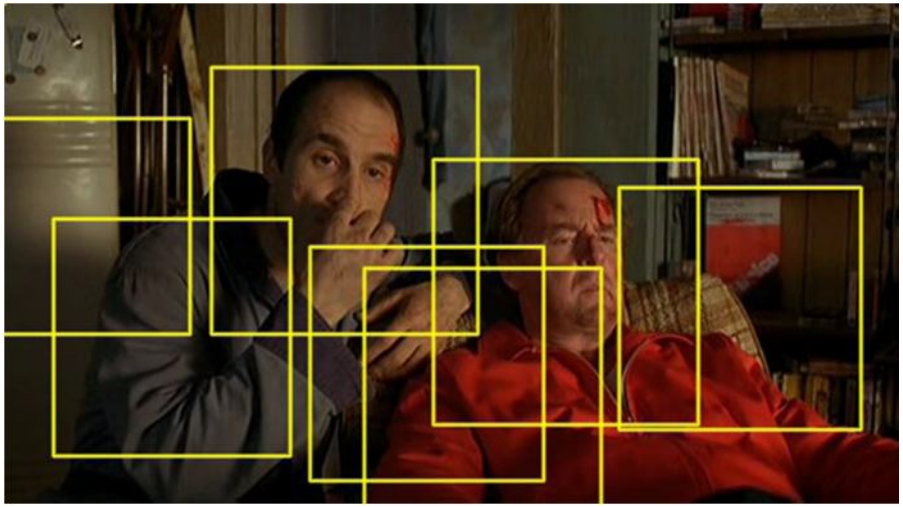
Сравнение

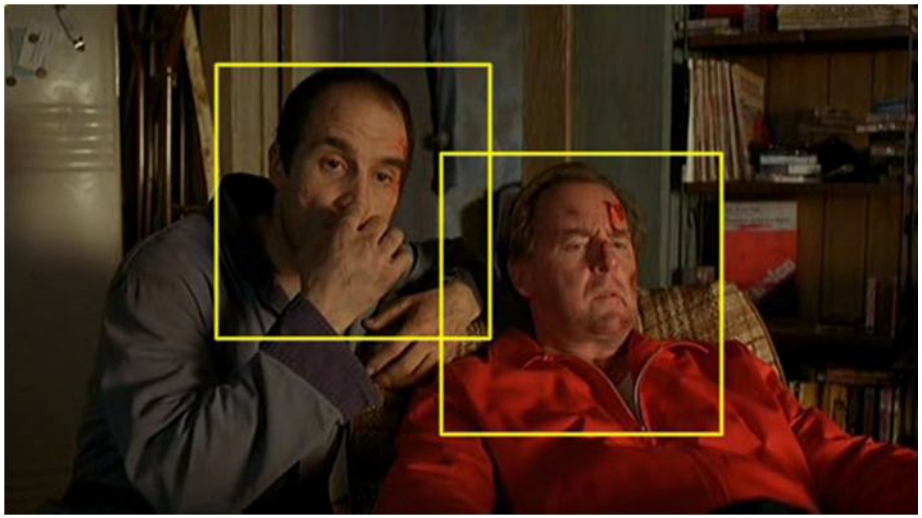


Сравнение

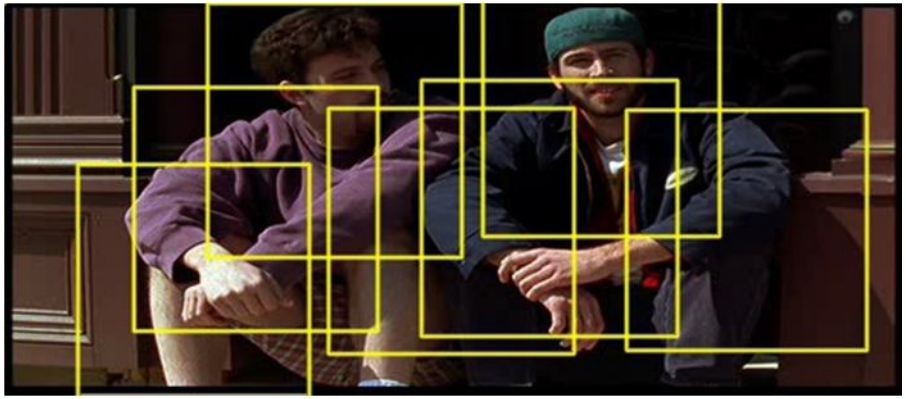


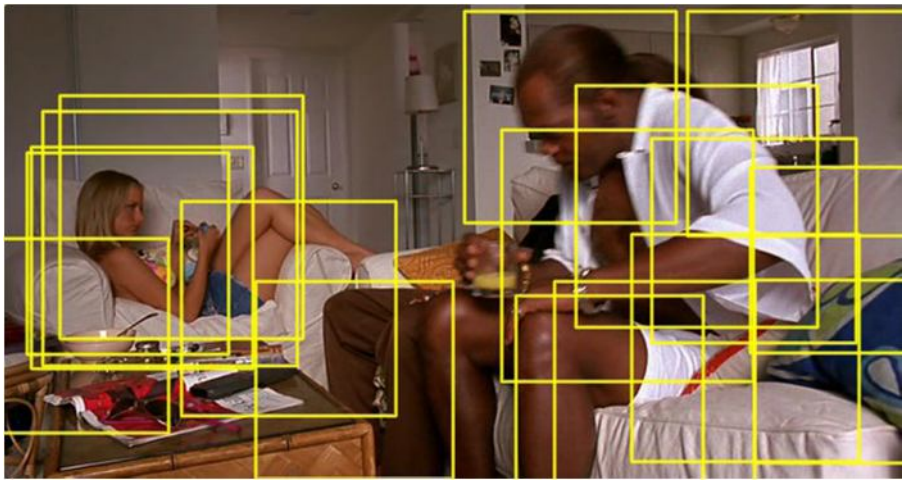




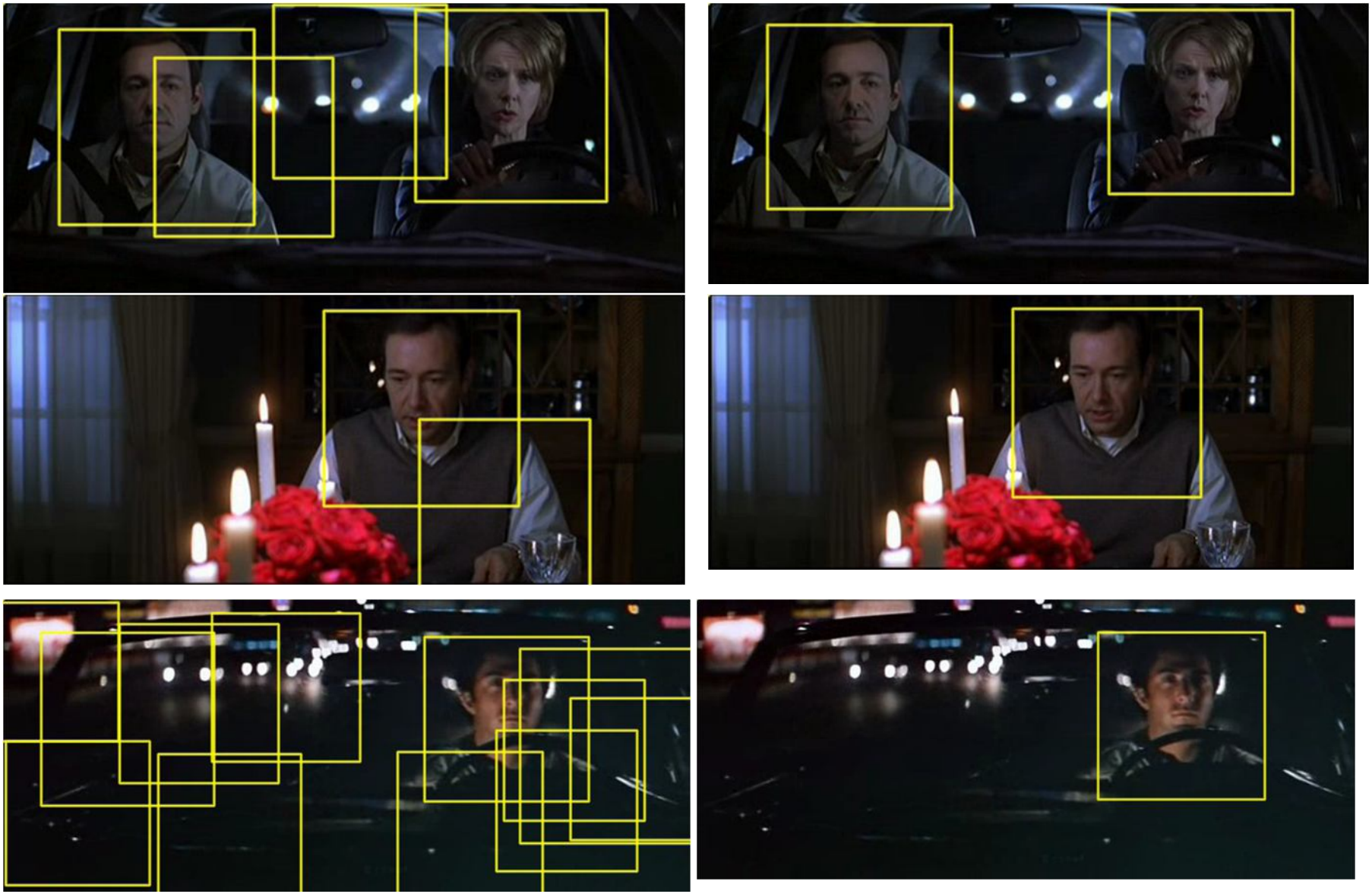


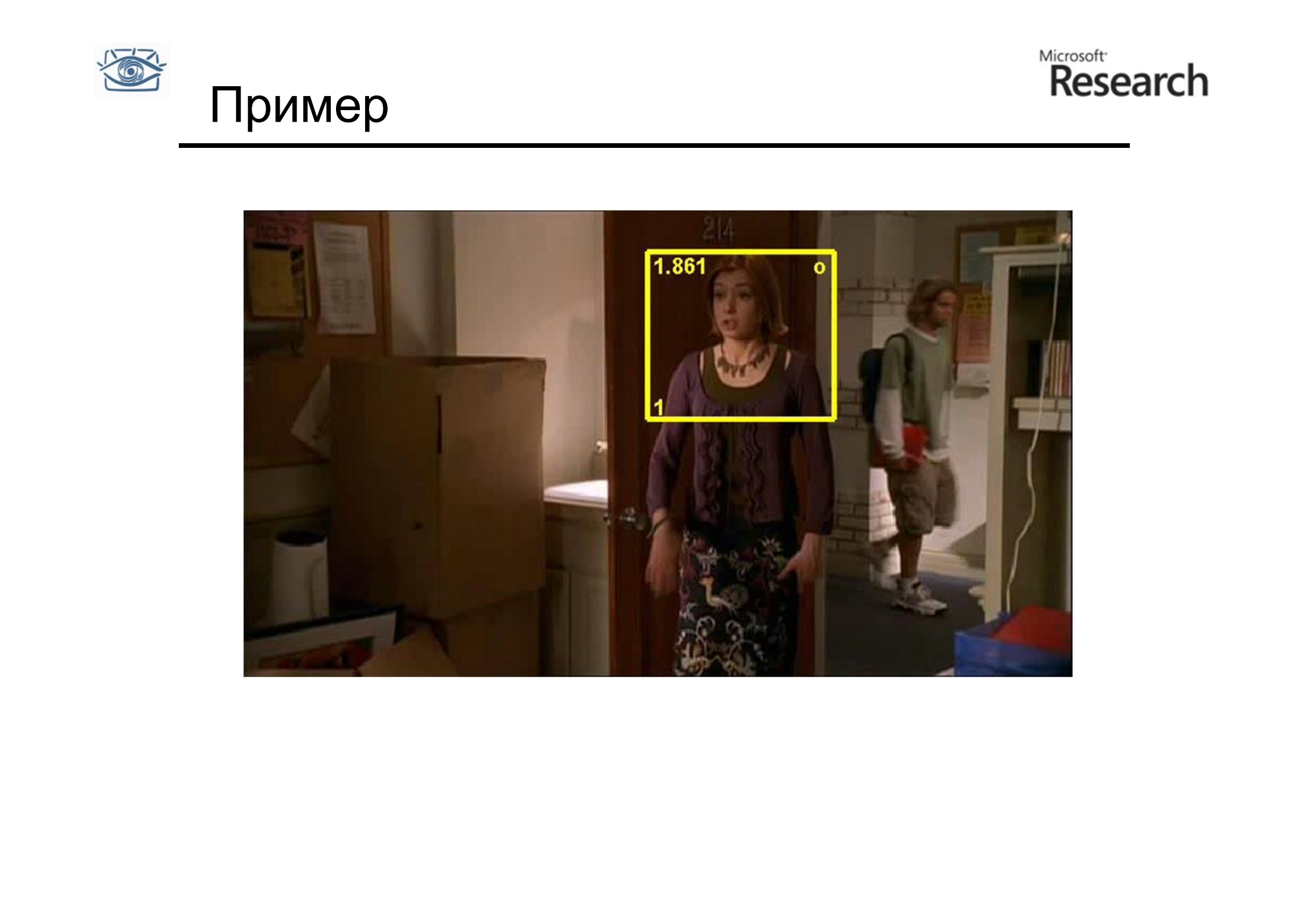
Сравнение





Сравнение




Microsoft
Research
Пример
214
1.861
o
1

Ускорение поиска

- Скользящее окно медленное, т.к. нужно пересмотреть множество окон пр.: x × y × масштаб ≈ 100,000 для картинки 320×240

- Большая часть окон – не объекты
- Как ускорить поиск с учетом этого?

Microsoft Research

Требования к детектору лиц

- Скользящим окно необходимо оценить десятки тысяч возможных комбинаций масштаба / положения
- Лица же достаточно редкий объект – 0-10 лиц на картинке
  - Для достижения вычислительной эффективности отрицательные примеры нужно отбрасывать как можно быстрее
  - На изображении в 1МП сопоставимое с количеством пикселей число возможных положений лица
  - Чтобы избежать ложных обнаружений (false positives) ошибка 2го рода должна быть ниже 10^{-6}

Microsoft Research

Детектор Violo-Jones

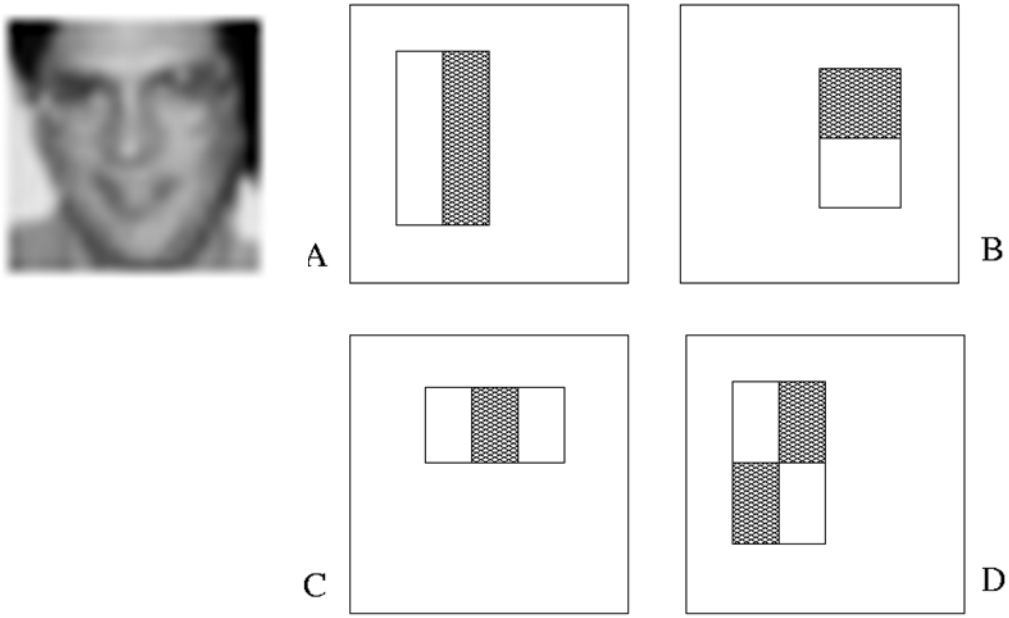
- Основополагающий метод для поиска объектов на изображении в реальном времени
- Обучение очень медленное, но поиск очень быстр
- Основные идеи:
  - *Интегральные изображения* для быстрого вычисления признаков
  - *Бустинг* для выбора признаков
  - *Каскад (Attentional cascade)* для быстрой отбраковки окон без лица

P. Viola and M. Jones. *Rapid object detection using a boosted cascade of simple features.* CVPR 2001.

P. Viola and M. Jones. *Robust real-time face detection.* IJCV 57(2), 2004.

Признаки

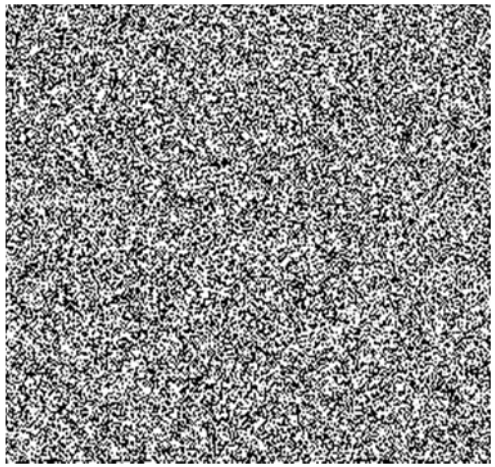
"Прямоугольные фильтры"



Value =

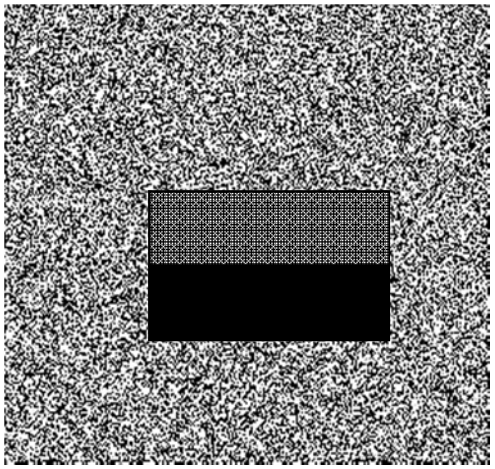
∑ *(pixels in white area)* –
∑ *(pixels in black area)*

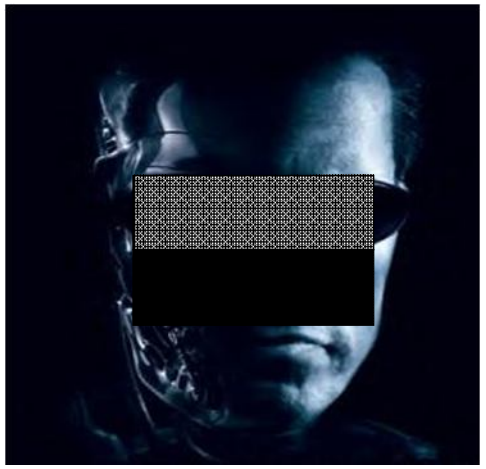
Пример



Source

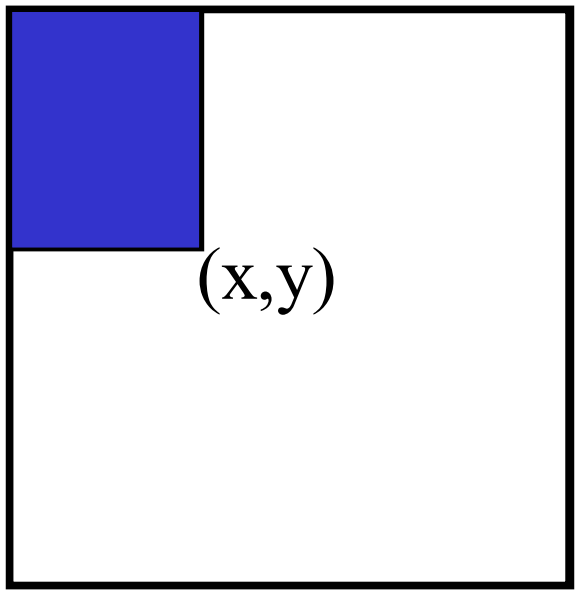
Result





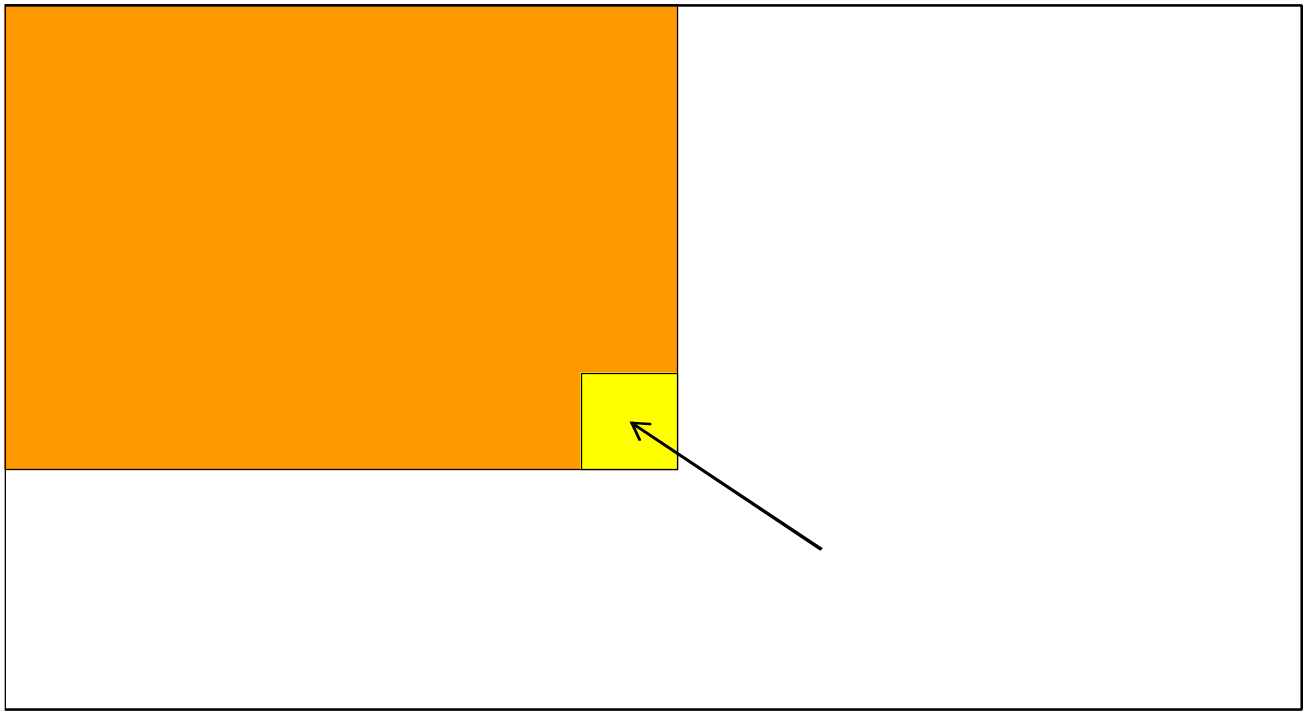
Интегральные изображения

- Значение каждого пиксела (*x*,*y*) равно сумме значений всех пикселов левее и выше пикселя (*x*,*y*) включительно
- Интегральное изображение рассчитывается за один проход



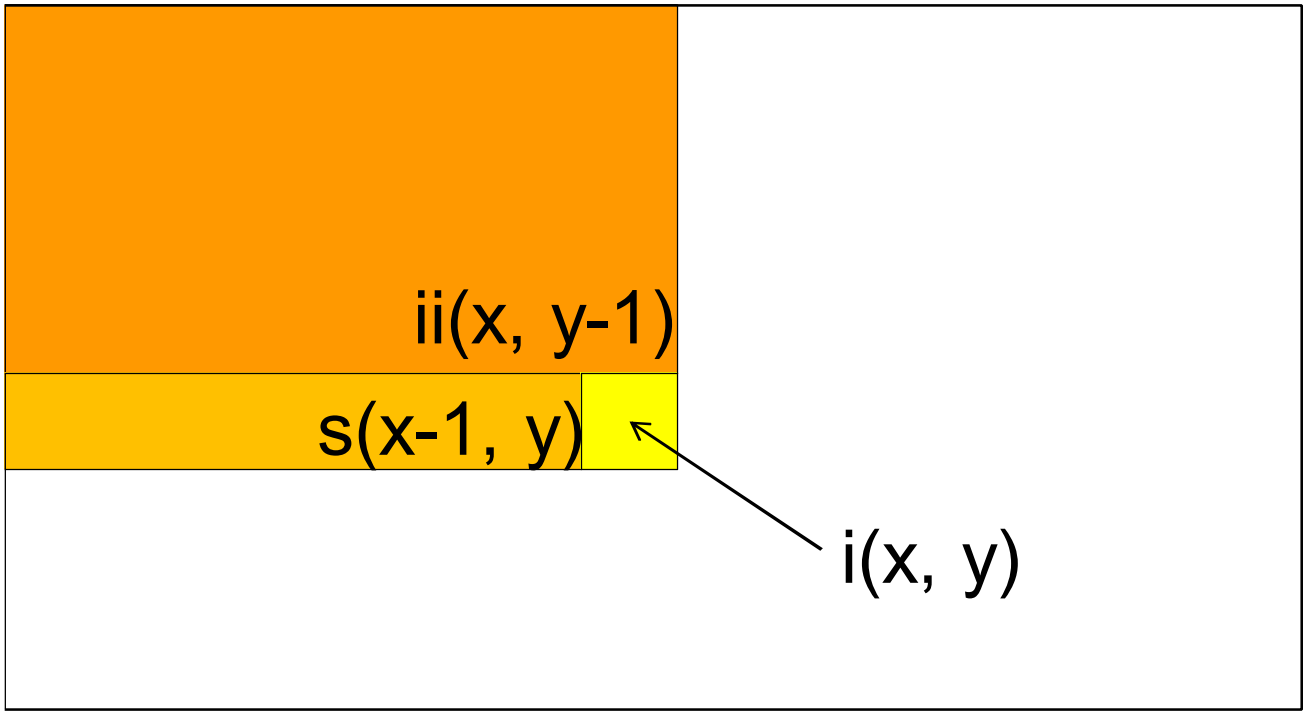
Microsoft Research

Вычисление интегрального изображения



Microsoft Research

Вычисление интегрального изображения



- Сумма по строке: s(x, y) = s(x–1, y) + i(x, y)
- Интегральное изображение: ii(x, y) = ii(x, y−1) + s(x, y)

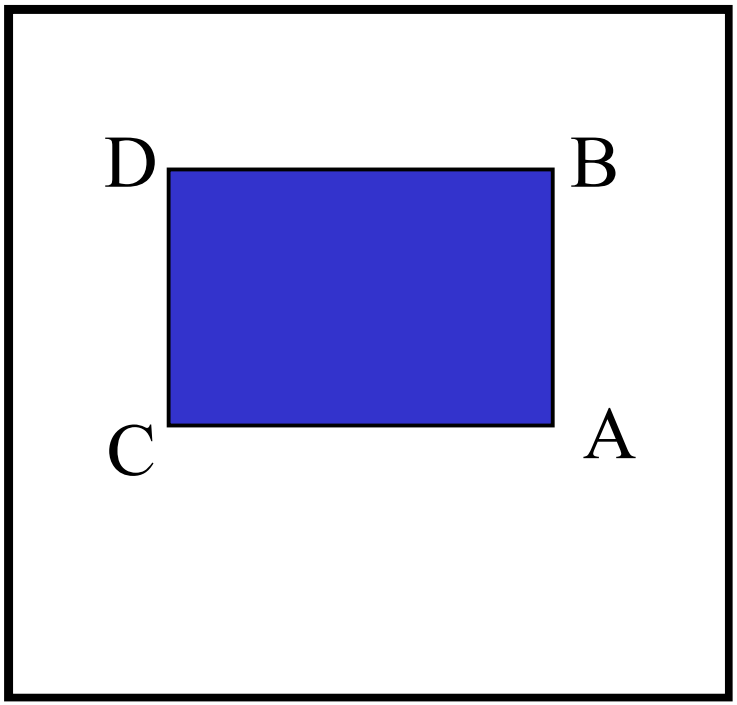
MATLAB: ii = cumsum(cumsum(double(i)), 2);

Microsoft Research

Вычисление суммы в прямоугольнике

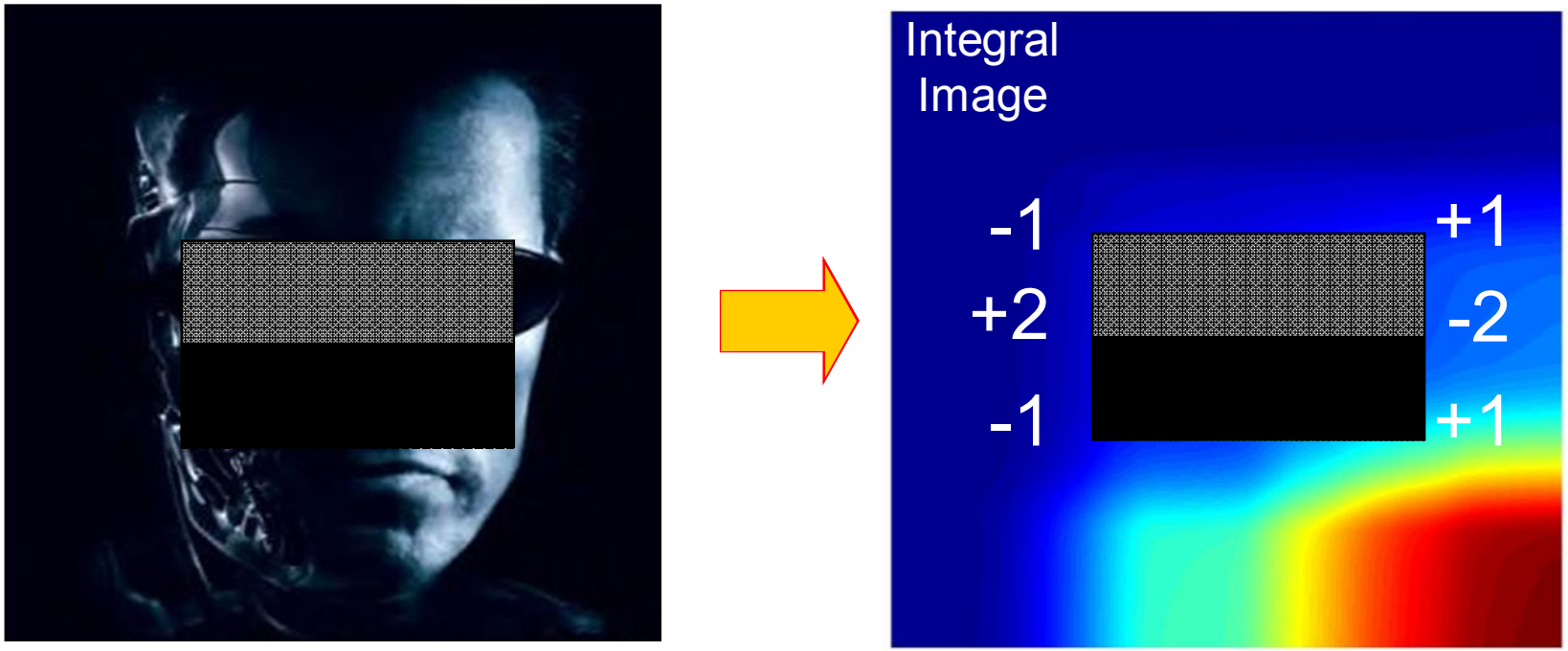
- Пусть A,B,C,D – значения интегрального изображения в углах прямоугольника
- Тогда сумма значений пикселов в исходном изображении вычисляется по формуле:

  sum = A – B – C + D
- 3 операции сложения для любого прямоугольника



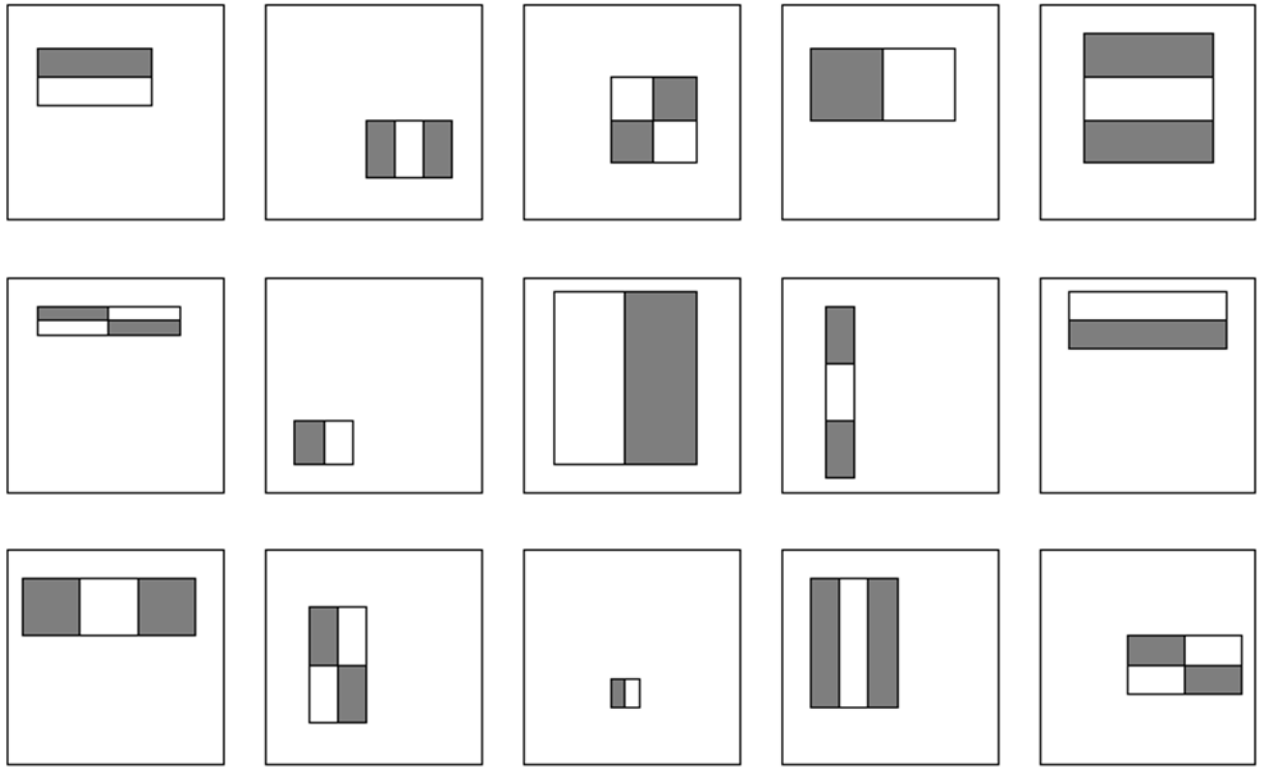
Пример

Microsoft Research



Выбор признаков

- Для окна поиска 24x24 пиксела, число возможных прямоугольных признаков достигает ~160,000!



Microsoft Research

Выбор признаков

- Для окна поиска 24x24 пиксела, число возможных прямоугольных признаков достигает ~160,000!
- В процессе поиска вычислять все признаки нереально
- Хороший классификатор должен использовать лишь маленькое подмножество всевозможных признаков
- Вопрос - как выбрать такое подмножество?

Microsoft Research

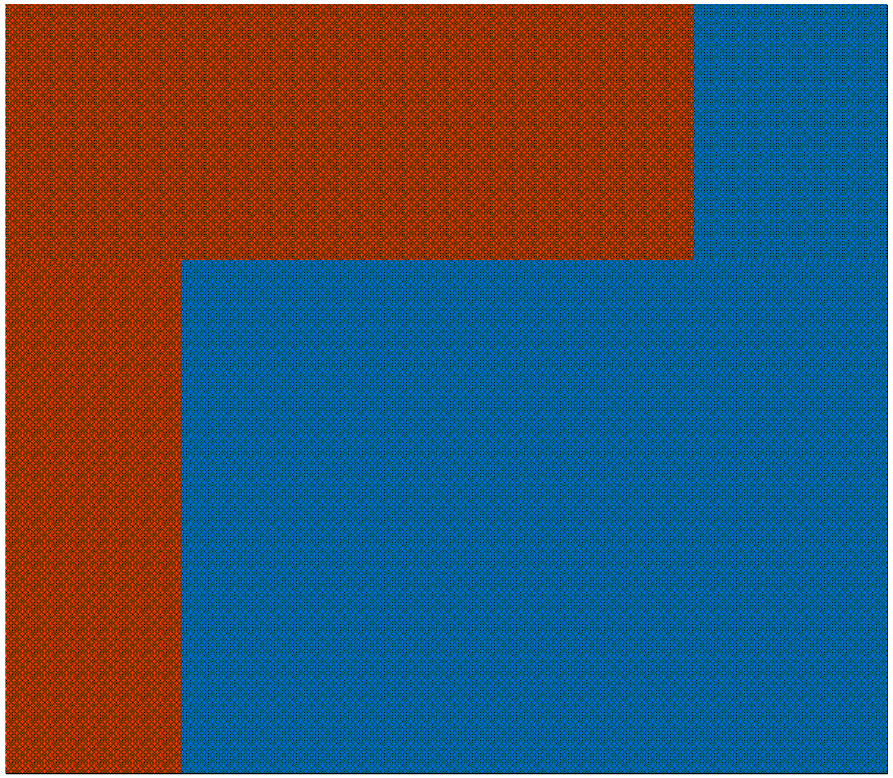
Бустинг

- Бустинг – схема классификации, основанная на комбинировании слабых классификаторов в более точный комитетный
  - Слабый классификатор должен быть лучше монетки
- Обучение состоит из нескольких этапов усиления (*boosting rounds)*
  - На каждом этапе выбираем слабый классификатор, который лучше всех сработал на примерах, оказавшихся трудными для предыдущих классификаторов
  - «Трудность» записывается с помощью весов, приписанных примерам из обучающей выборки
  - Составляем общий классификатор как линейную комбинацию слабых классификаторов

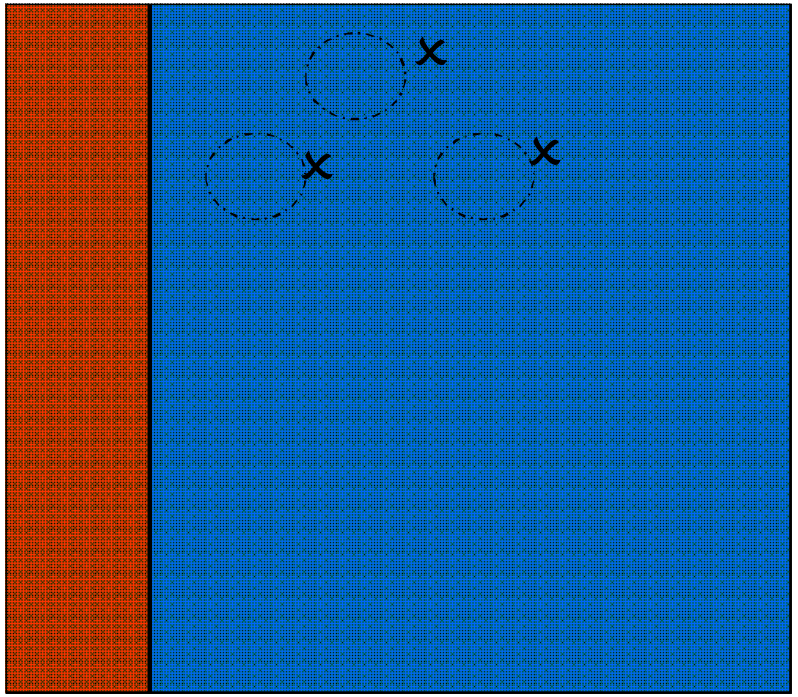
Y. Freund and R. Schapire, A short introduction to boosting, *Journal of Japanese Society for Artificial Intelligence*, 14(5):771-780, September, 1999.

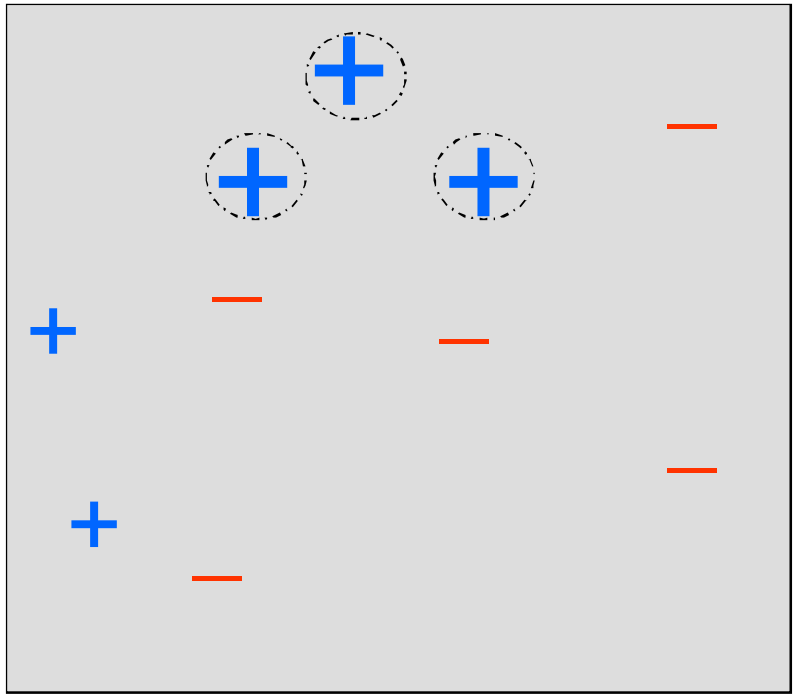
Microsoft Research

Пример хорошего классификатора



Итерация 1 из 3





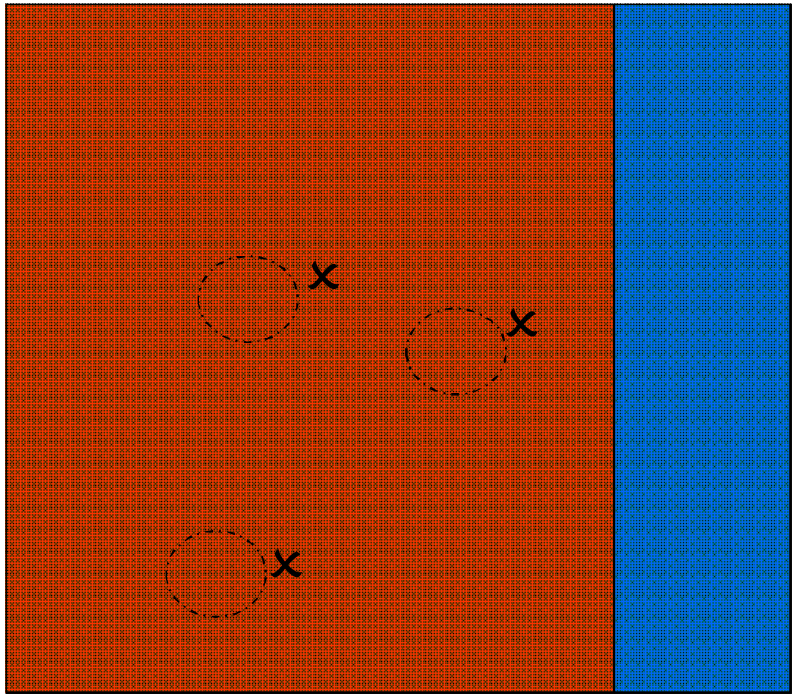
h_1

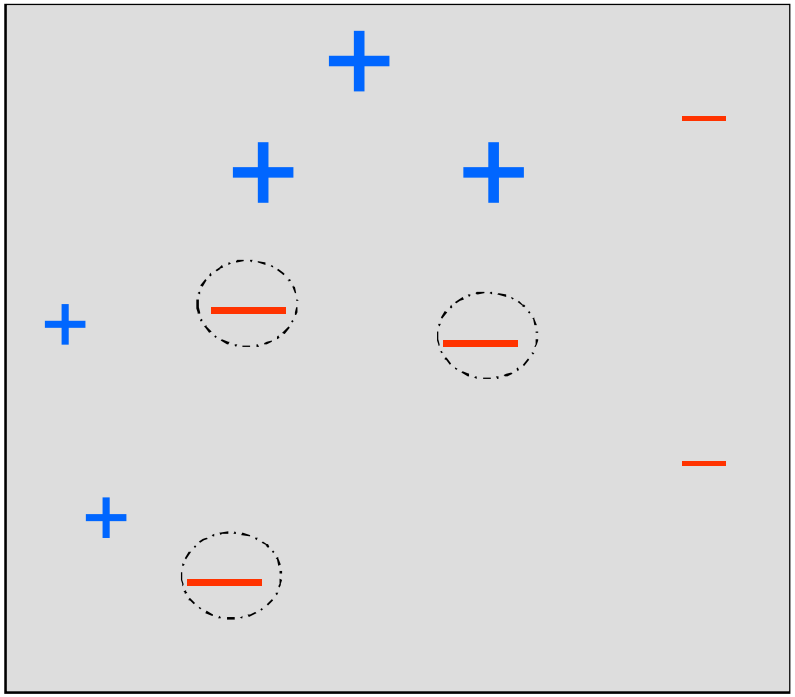
$\varepsilon_1 = 0.300$

$\alpha_1 = 0.424$

D_2

Итерация 2 из 3



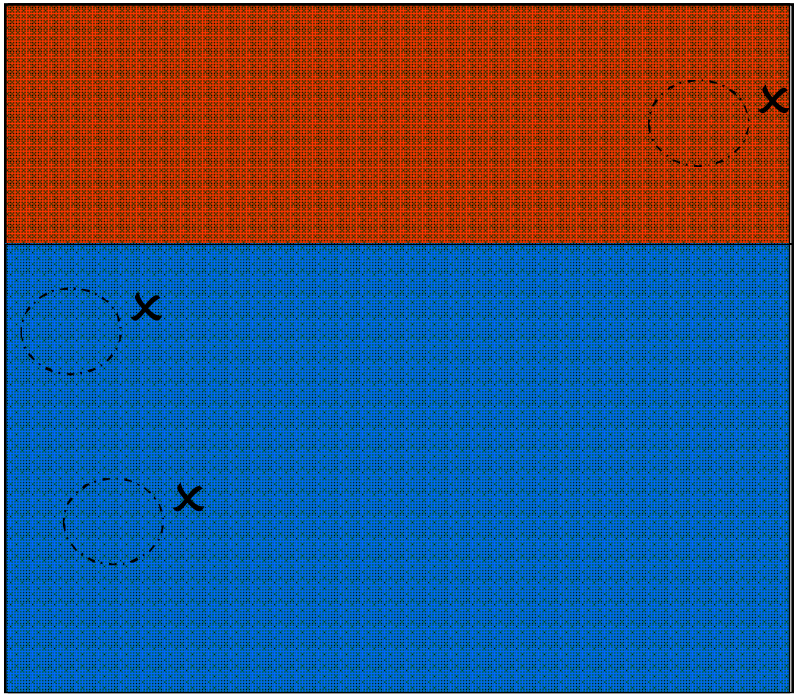


$\varepsilon_2 = 0.196$ h_2

$\alpha_2=0.704$

D_2

Итерация 3 из 3



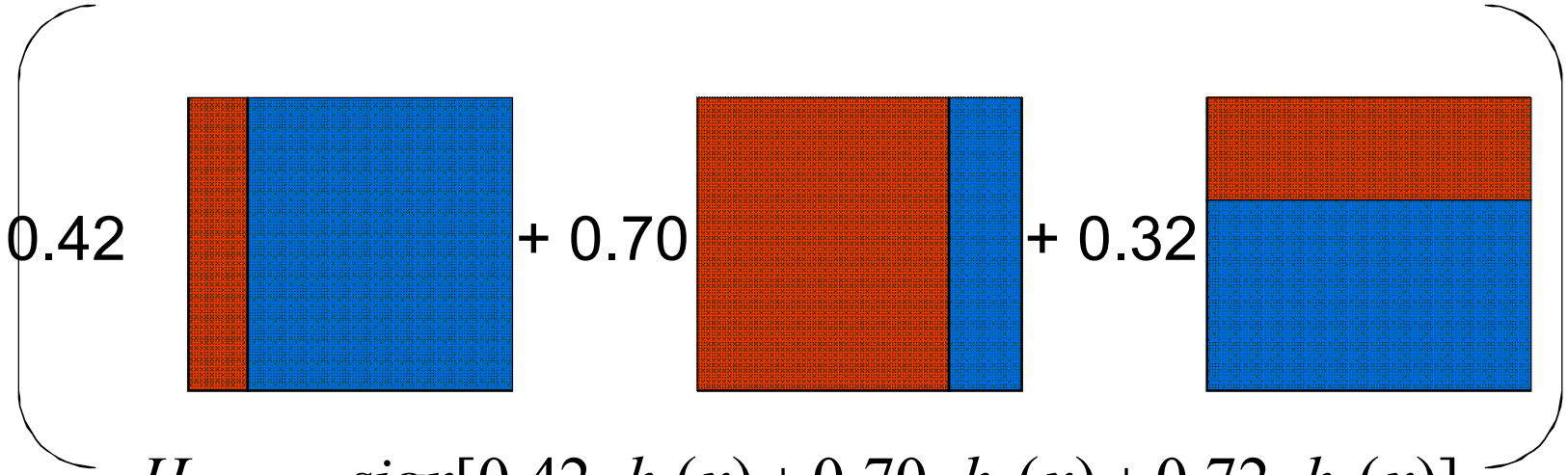
h_3

СТОП

$\varepsilon_3 = 0.344$

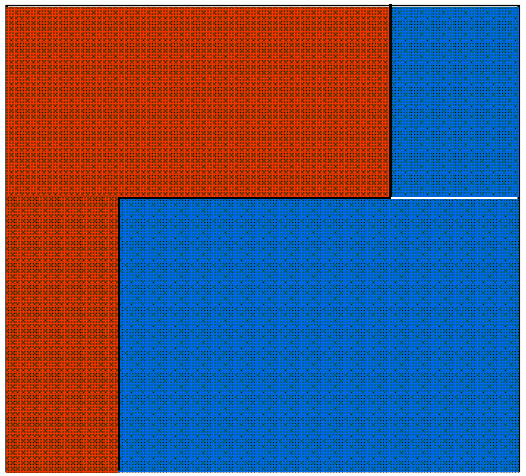
$\alpha_2 = 0.323$

Конечная гипотеза



$$H_{final} = sign[0.42 \cdot h_1(x) + 0.70 \cdot h_2(x) + 0.72 \cdot h_3(x)]$$

$$h_i(x) = \begin{cases} -1, x \to \Theta_0 \\ +1, x \to \Theta_1 \end{cases}$$



AdaBoost (Discreet)

Пусть есть набор *{h}* – слабых классификаторов

Пусть *T*: $(x_1, y_1), \ldots, (x_m, y_m)$ где $x_i \in X, y_i \in \Theta = \{-1, +1\}$

Инициализируем $D_1(i) = 1/m$

Для $k = \overline{1, K}$

1. Обучим h_k с минимальной ошибкой

$$\varepsilon_k = \Pr_{i \sim D_k}[h_k(x_i) \neq y_i]$$

Ошибка h_t рассчитывается с учётом D_t

2. Рассчитаем вес гипотезы

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_k}{\varepsilon_k}\right)$$

Вес Адаптируется. Чем больше ε_k тем меньше α_k

3. Для всех i = 1 to m

$$D_{k+1}(i) = \frac{D_k(i)}{Z_k} \times \begin{cases} e^{-\alpha_k} & \text{if } h_k(x_i) = y_i \\ e^{\alpha_k} & \text{if } h_k(x_i) \neq y_i \end{cases}$$

Увеличиваем вес примера, если на нём алгоритм ошибается

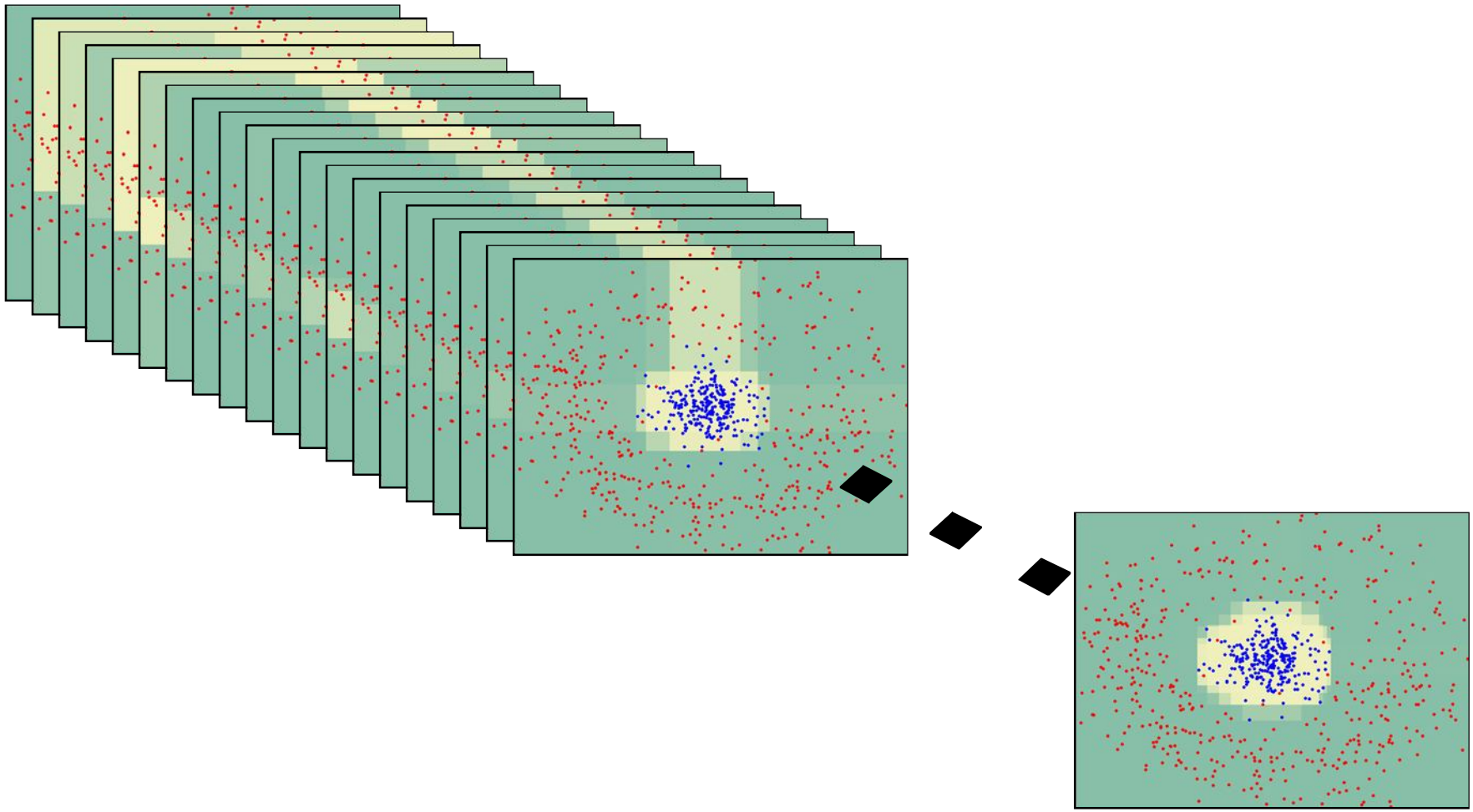
Z_t нормализующий коэф.

Результат:

$$H(x) = \operatorname{sign}\left(\sum_{k=1}^{K} \alpha_k h_k(x)\right)$$

Линейная комбинация

AdaBoost (пороги)



Microsoft Research

Эмпирический риск

- Как показали Freund и Schapire

$$\gamma_k = \frac{1}{2} - \varepsilon_k$$

$$R_{emp} \le \exp\left(-2 \sum_k \gamma_k^2 \right)$$

- Эмпирический риск падает по экспоненте – высокая скорость обучения

Microsoft Research

Бустинг

- Плюсы
  - + Универсальность
  - + Высокая скорость сходимости
  - + Высокая обобщающая способность
  - + Возможность очень эффективной программной реализации и распараллеливания
  - + Простота метода и отсутствия параметров
- Минусы
  - – Трудность определения нужного числа итераций обучения (зачастую, ошибка на контроле продолжает падать и после нуля эмпирического риска)

Слабые классификаторы

- Определяем слабые классификаторы на основе прямоугольных признаков

Значение признака

$$h_t(x) = \begin{cases} 1 & \text{if } f_t(x) > \theta_t \\ 0 & \end{cases}$$

окно

порог

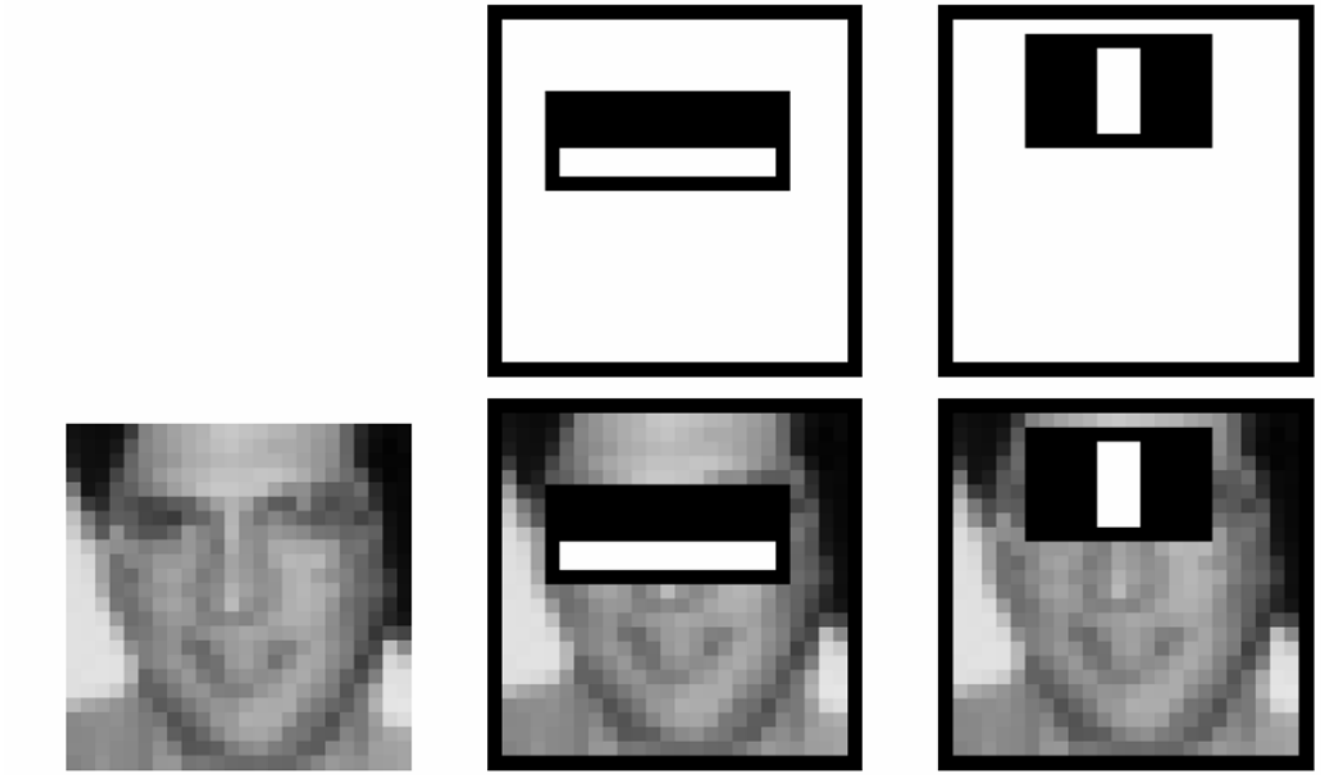
Microsoft Research

Бустинг для поиска лиц

- Определяем слабые классификаторы на основе прямоугольных признаков
- Для каждого этапа бустинга:
  - Вычисляем каждый прямоугольный признак на каждом примере
  - Выбираем наилучший порог для каждого признака
  - Выбираем наилучший признак / порог
  - Перевзвешиваем выборку
- Вычислительная сложность обучения: $O(MNK)$
  - M этапов, N примеров, K признаков

Бустинг для поиска лиц

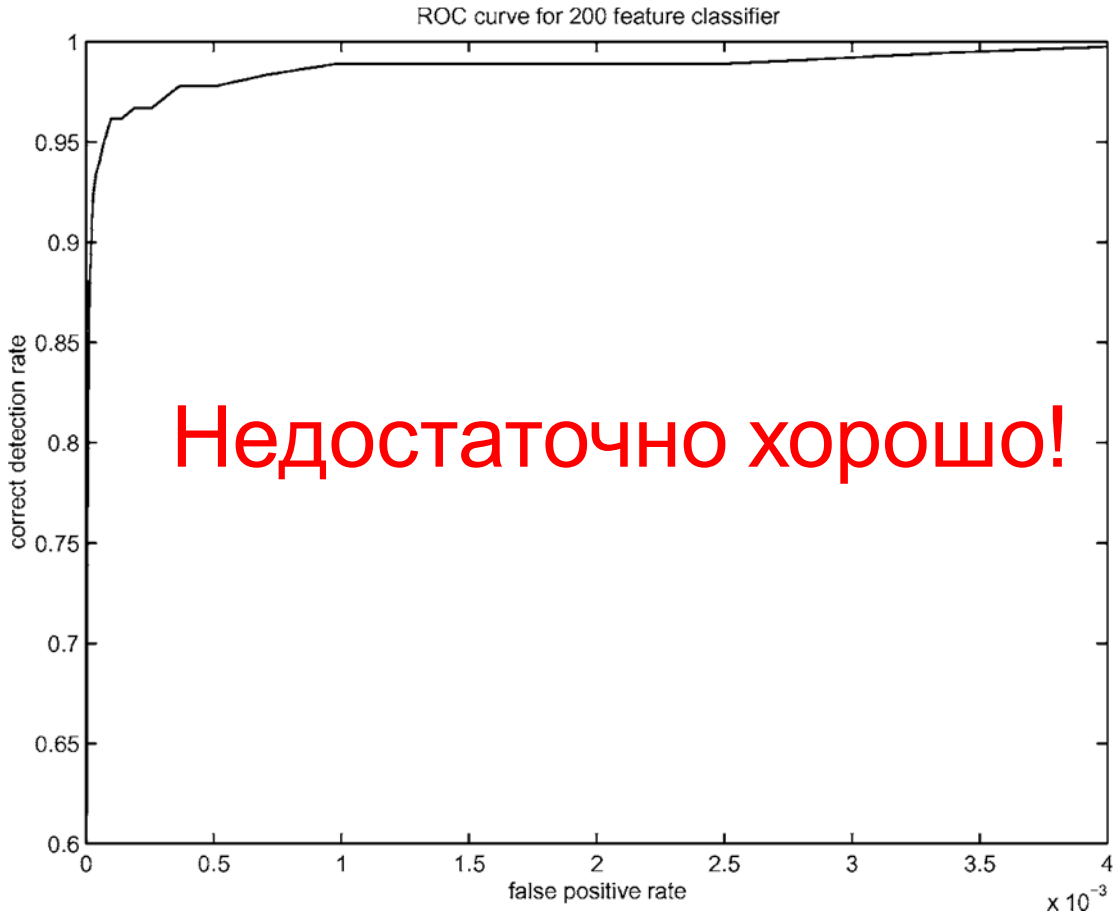
- Первые два признака, выбранные бустингом:



Эта комбинация признаков дает 100% detection rate и 50% false positive rate

Бустинг для поиска лиц

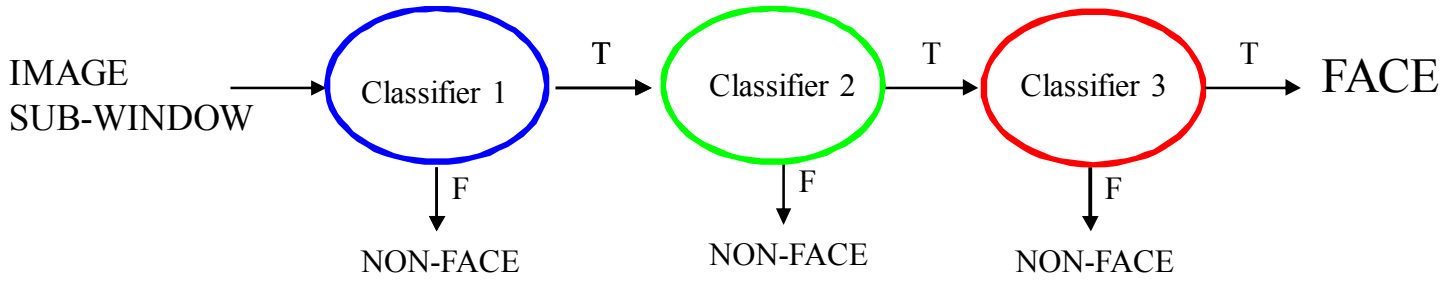
- Классификатор из 200 признаков дает 95% detection rate и a false positive rate of 1 in 14084



Receiver operating characteristic (ROC) curve

Каскад (Attentional cascade)

- Начинаем с простых классификаторов, которые отбрасывают часть отрицательных окон, при этом принимая почти все положительные окна
- Положительный отклик первого классификатора запускает вычисление второго, более сложного, классификатора, и т.д.
- Отрицательный отклик на любом этапе приводит к немедленной отбраковке окна



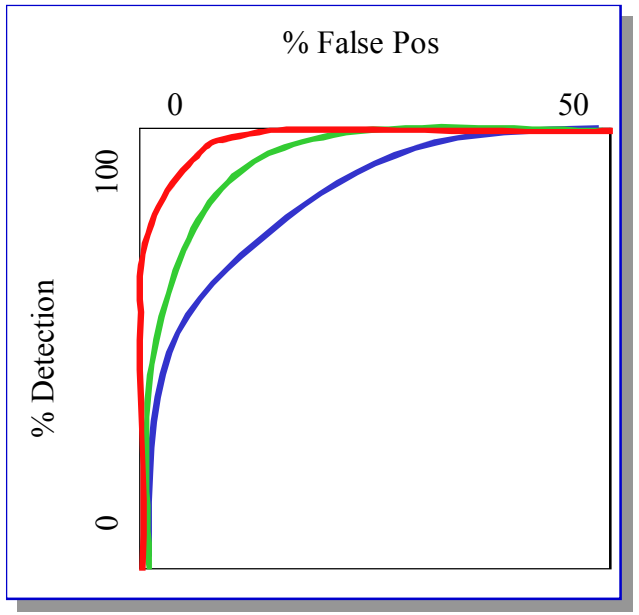
Каскад

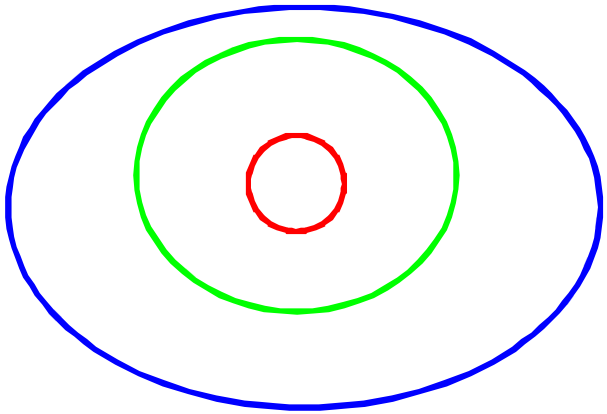
- Медленные классификаторы применяются только к некоторым окнам ⇒ существенное ускорение
- Управляем сложностью/скоростью классификатора:
  - Количество опорных векторов [Romdhani et al, 2001]
  - Количество признаков [Viola & Jones, 2001]
  - Видом ядра MOB [Vedaldi et al, 2009]

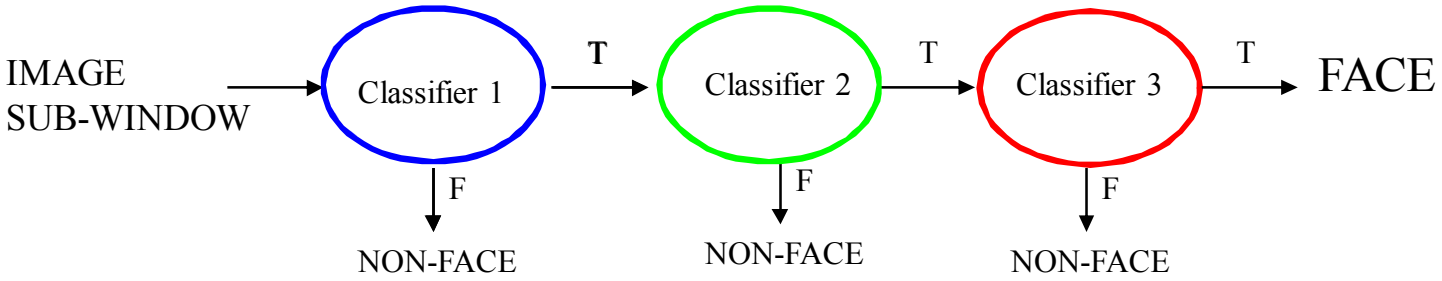
Каскад

- Цепочка классификаторов с каждым уровнем становится более сложной, ошибка второго рода постоянно снижается

Receiver operating characteristic

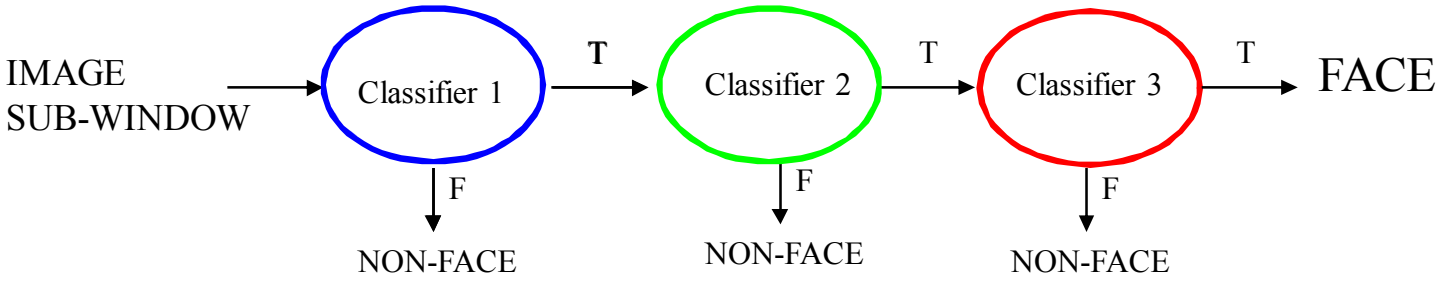






Каскад

- detection rate и false positive rate каскада вычисляются как произведение соответствующих уровней ошибок каждого этапа
- A detection rate 0.9 и false positive rate порядка 10^{-6} достигается с помощью каскада из 10 этапов, если на каждом этапе detection rate примерно равен 0.99 ($0.99^{10} \approx 0.9$) и false positive rate примерно 0.30 ($0.3^{10} \approx 6\times10^{-6}$)



Microsoft Research

Обучение каскада

- Задаем требуемые значения detection and false positive rates для каждого этапа
- Добавляем признаки до тех пор, пока параметры текущего этапа не достигнут заданного уровня
  - Приходится понижать порог AdaBoost для максимизации обнаружения (в противоположенность минимизации общей ошибки классификации)
  - Тестирование на отдельном наборе (*validation set)*
- Если общий уровень false positive rate недостаточно низок, добавляем очередной этап
- Ложные обнаружения на текущем этапе используются как отрицательные примеры на следующем этапе

Тренировочная выборка

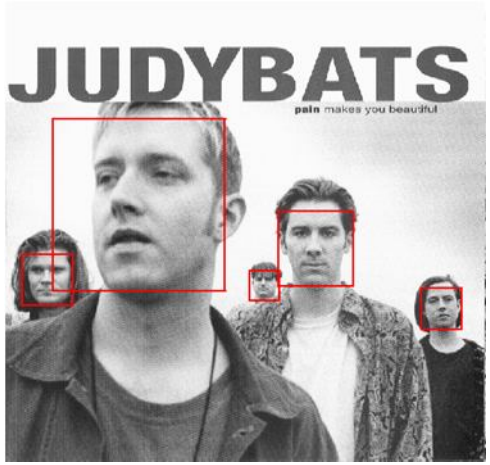
- 5000 лиц
  - Все фронтальные, уменьшенные до 24x24 пикселов
  - Все нормированы
- 300M отрицательных примеров
  - 9500 изображений без лиц
- Большая изменчивость
  - Разные люди
  - Освещение
  - Поза лица

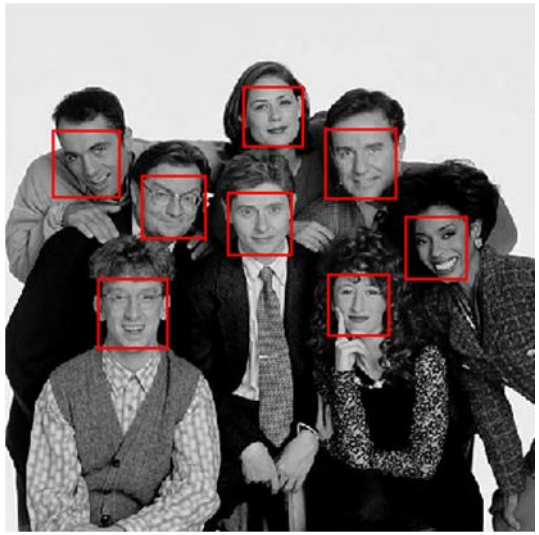
Microsoft Research

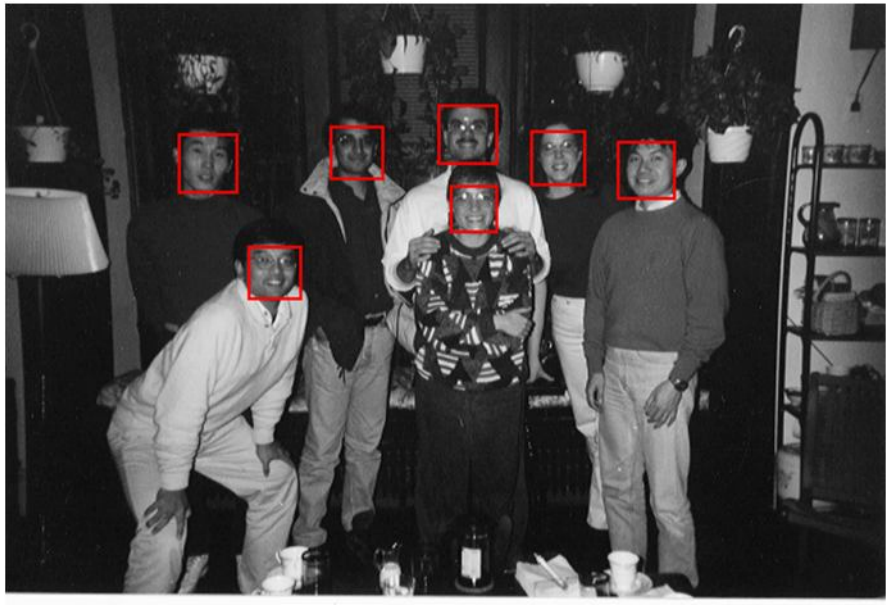
Производительность системы

- Обучение: “недели” на 466 MHz Sun рабочей станции
- 38 этапов, всего 6061 признаков
- В среднем 10 признаков оцениваются для каждого окна на тестовой выборке
- “На 700 Mhz Pentium III, детектор лиц обрабатывает одно изображение 384x288 пикселов за 0.067 секунды”
  - 15 Hz
  - В 15 раз быстрее сравнимого по точности предшествующего метода (Rowley et al., 1998)

Пример работы





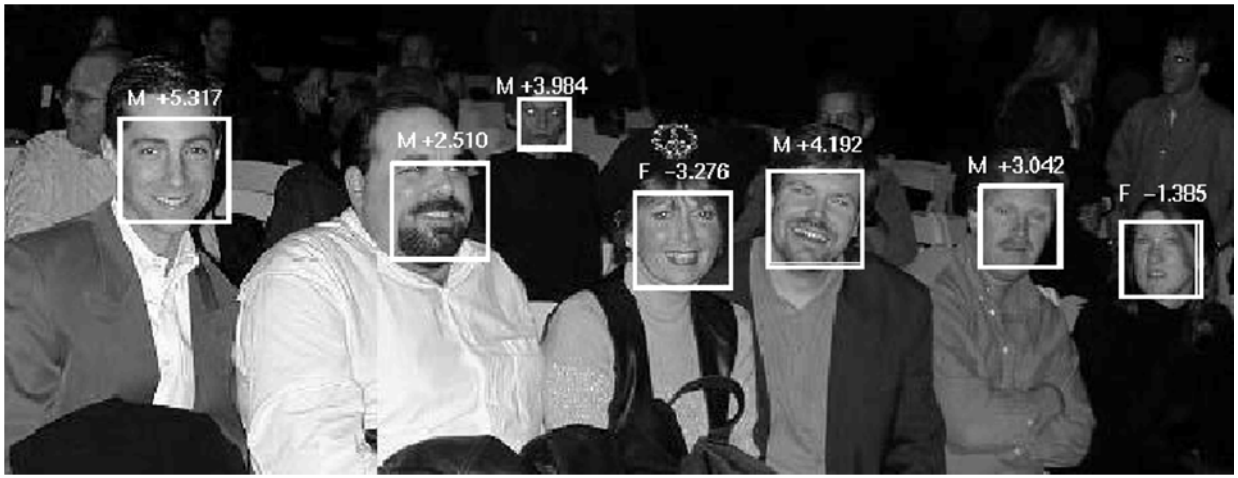


Другие задачи поиска объектов

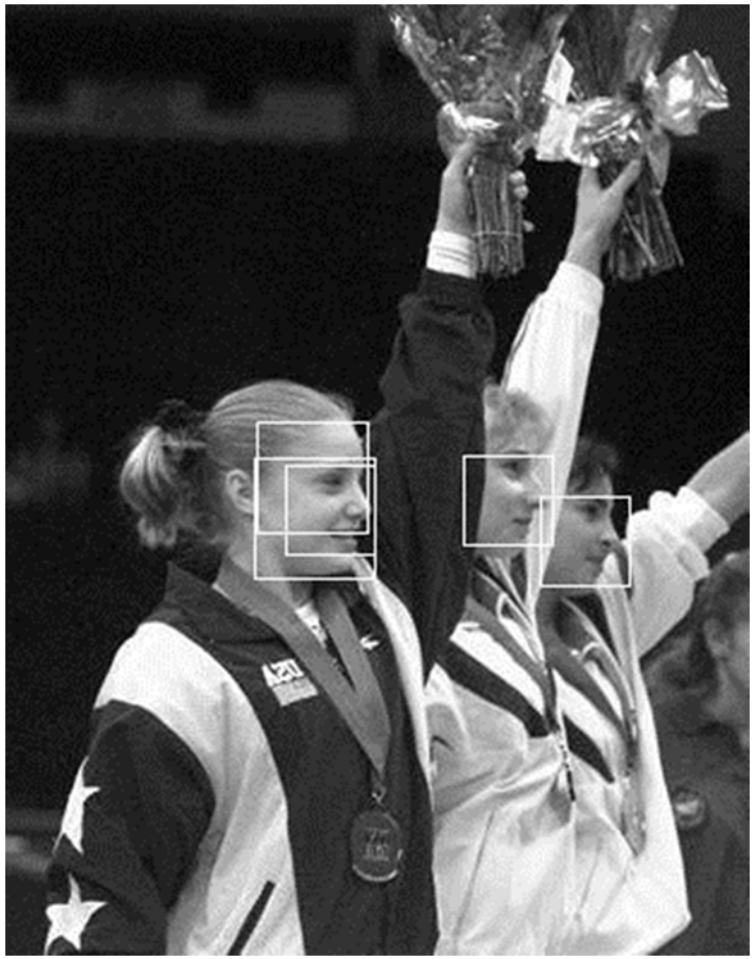
Локализация черт лица

Поиск профилей

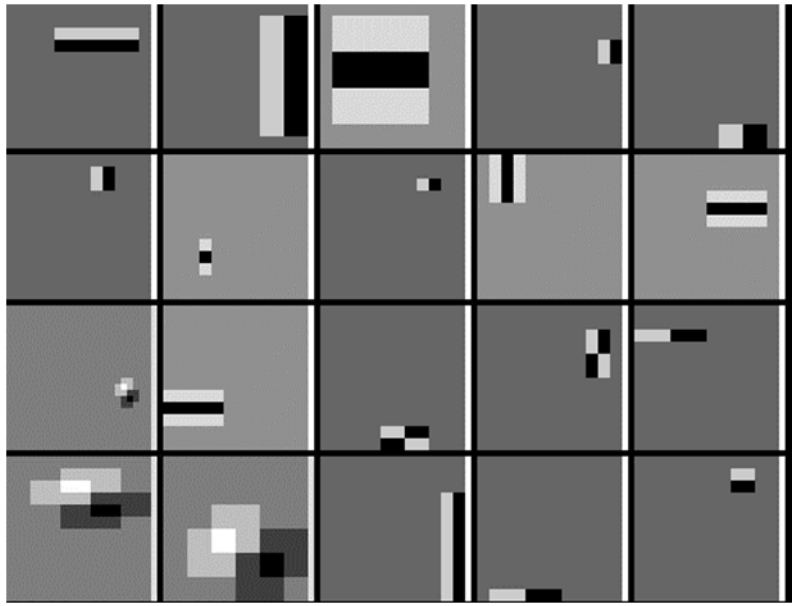
Определение пола



Поиск профилей



Признаки для поиска профилей



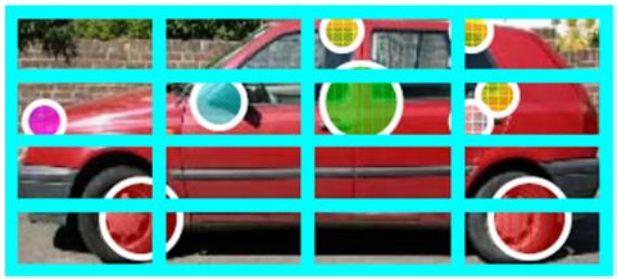
Microsoft Research

Резюме: детектор Violo-Jones

- Прямоугольные признаки
- Интегральные изображения для быстрого вычисления признаков
- Бустинг для выбора признаков
- Каскад классификаторов для быстрого выбраковки отрицательных окон

Резюме скользящего окна

- Скользящее окно позволяет превратить любой классификатор изображений в детектор объектов.
- Требования к инвариантности снижаются за счет поиска по сдвигу и масштабу
- Учет пространственной информации можно внедрить в любой метод за счет разбиение окна



Microsoft Research

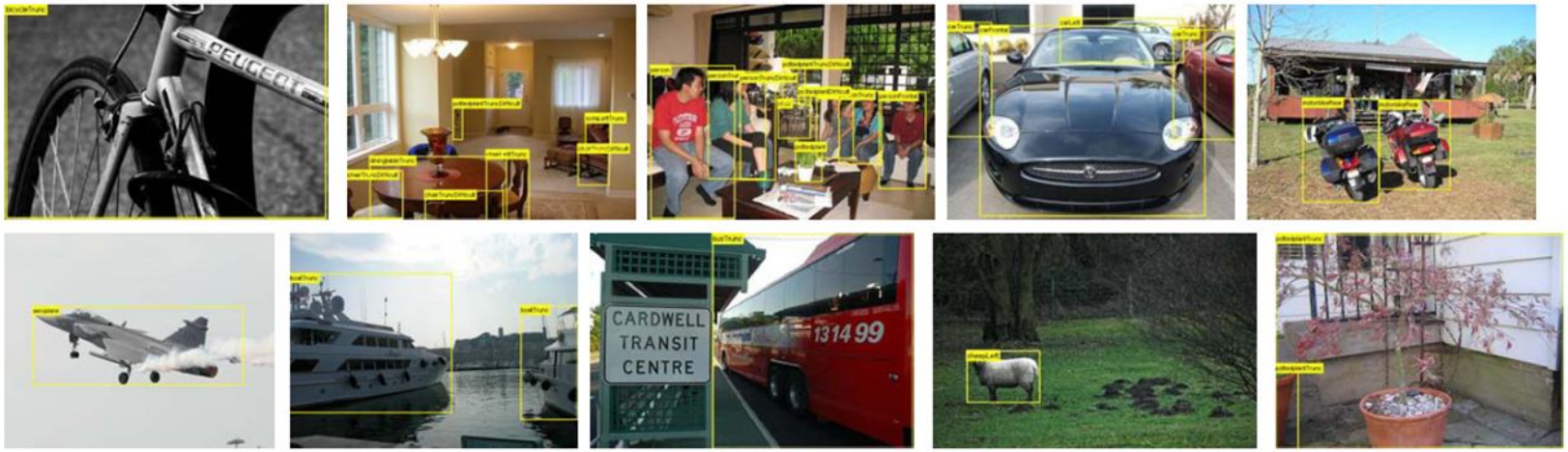
PASCAL VOC

- PASCAL Visual Object Classes (VOC) Dataset and Challenge
  - Mark Everingham
  - Luc Van Gool
  - Chris Williams
  - John Winn
  - Andrew Zisserman

Microsoft Research

Данные

- 20 классов:
  - aeroplane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, TV
- Реальные изображения из flickr, не фильтровались по качеству
- Сложные сцены, разный масштаб, положение, освещение, перекрытие…

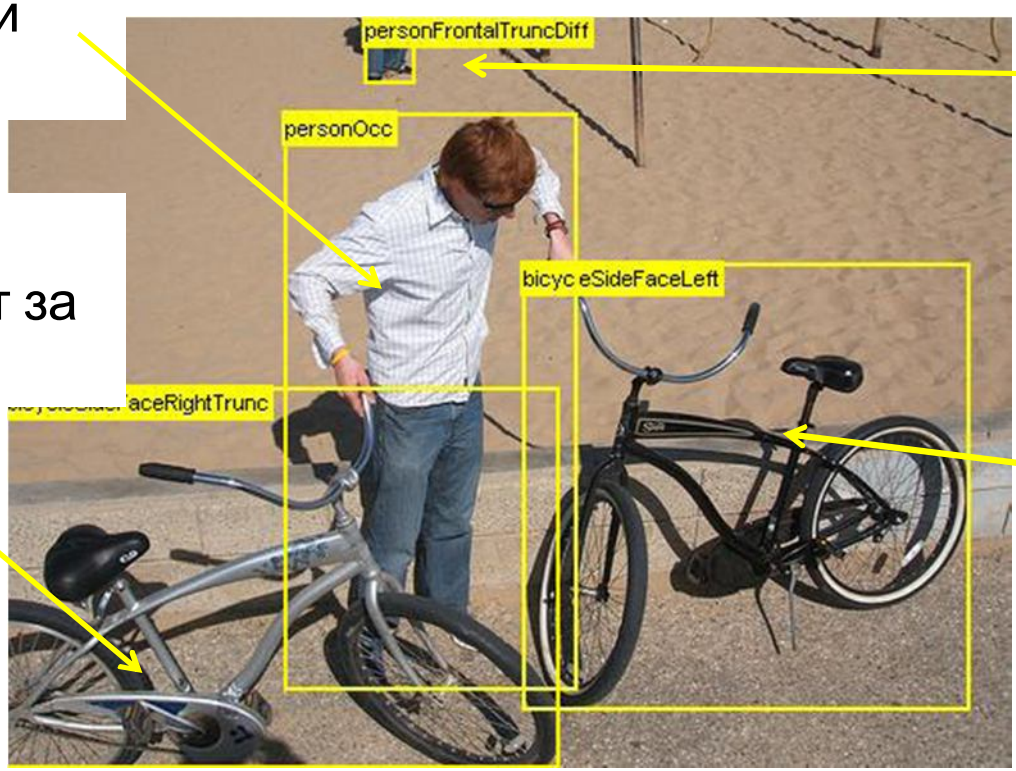


Аннотация

- Полная аннотация всех объектов
  - В одну сессию по указанию

Перекрытый
Объект перекрыт другим внутри рамки

Обрезанный – объект выходит за пределы рамки

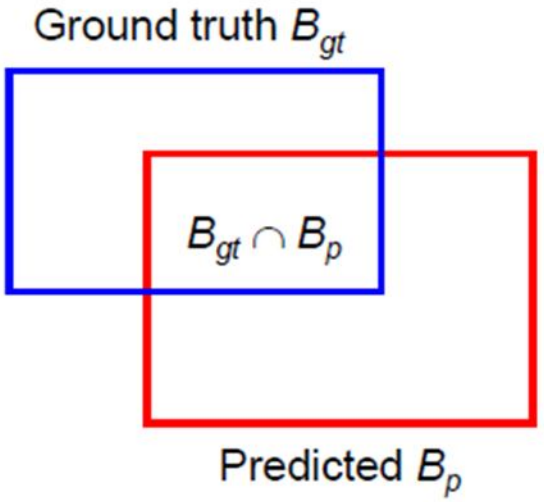


Сложный
Не участвует в оценке

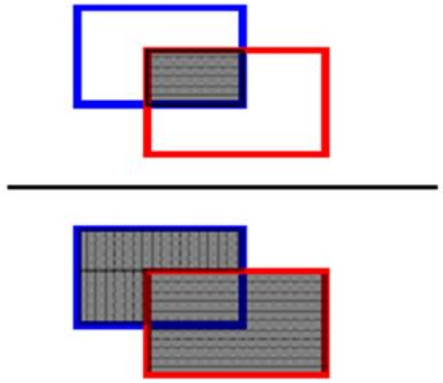
Положение
Смотрит налево

Оценка локализации

- Area of Overlap (AO) Measure

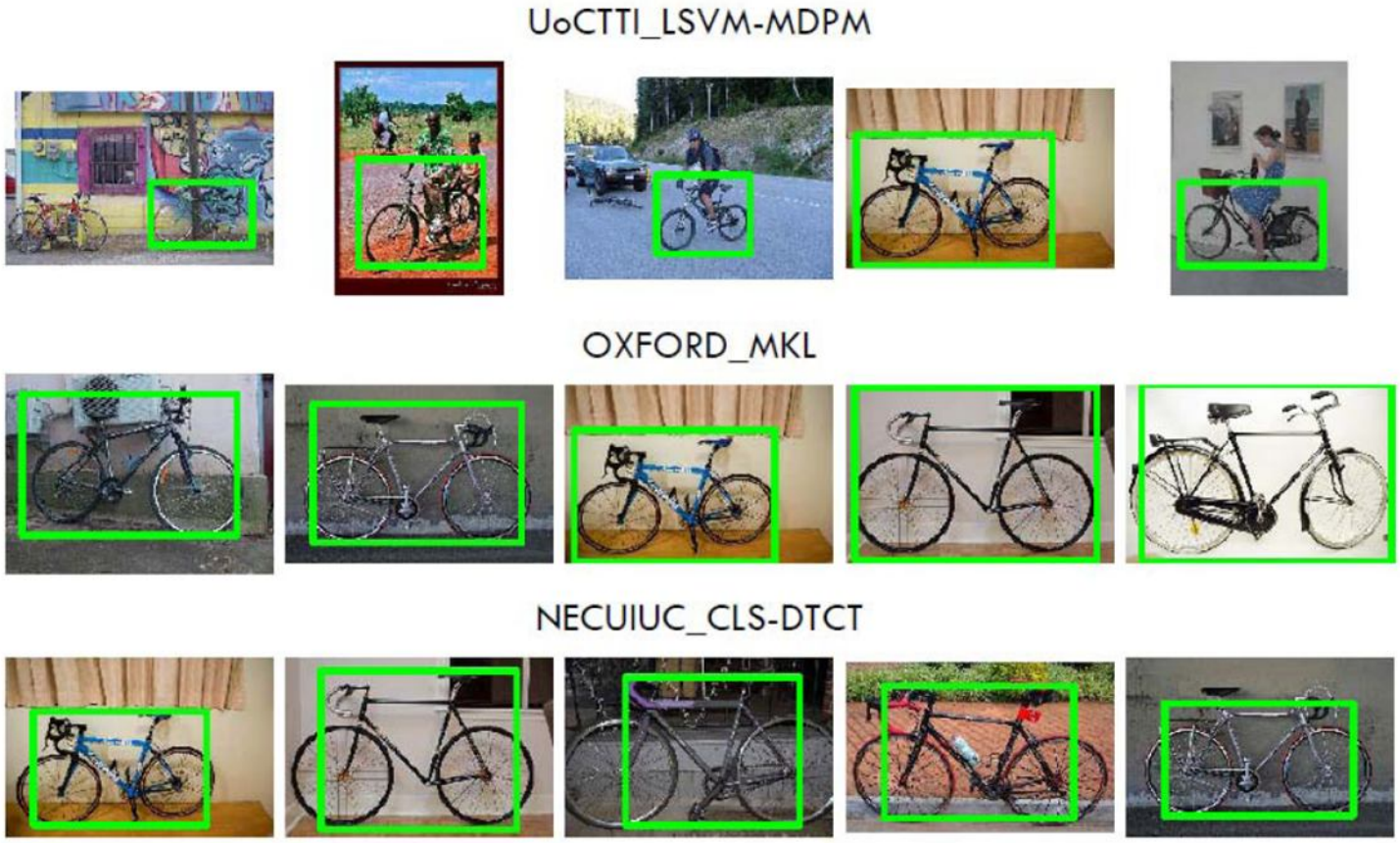


$$AO(B_{gt}, B_p) = \frac{|B_{gt} \bigcap B_p|}{|B_{gt} \bigcup B_p|}$$

Detection if  > Threshold

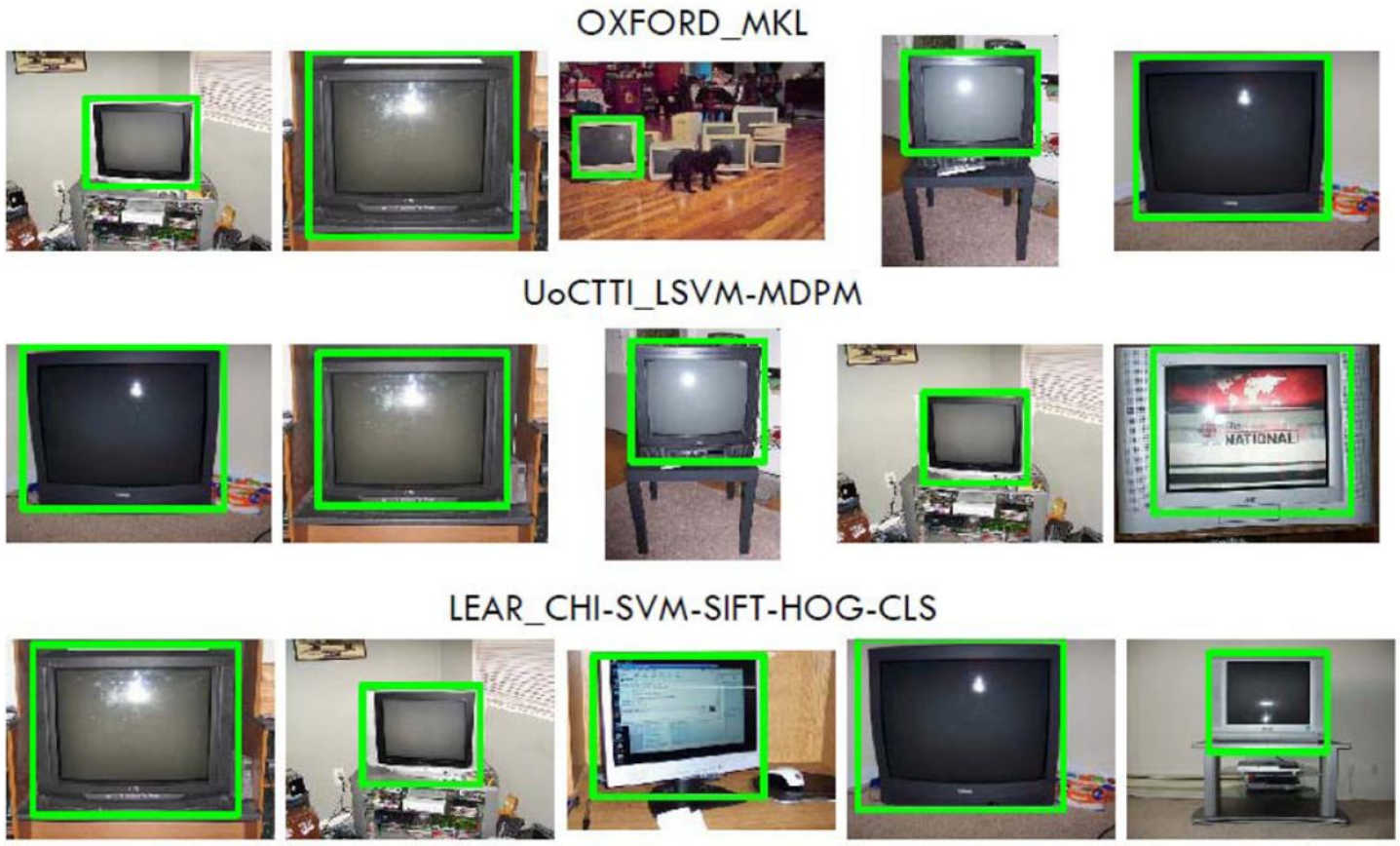
50%

Примеры



Примеры ошибок

Примеры



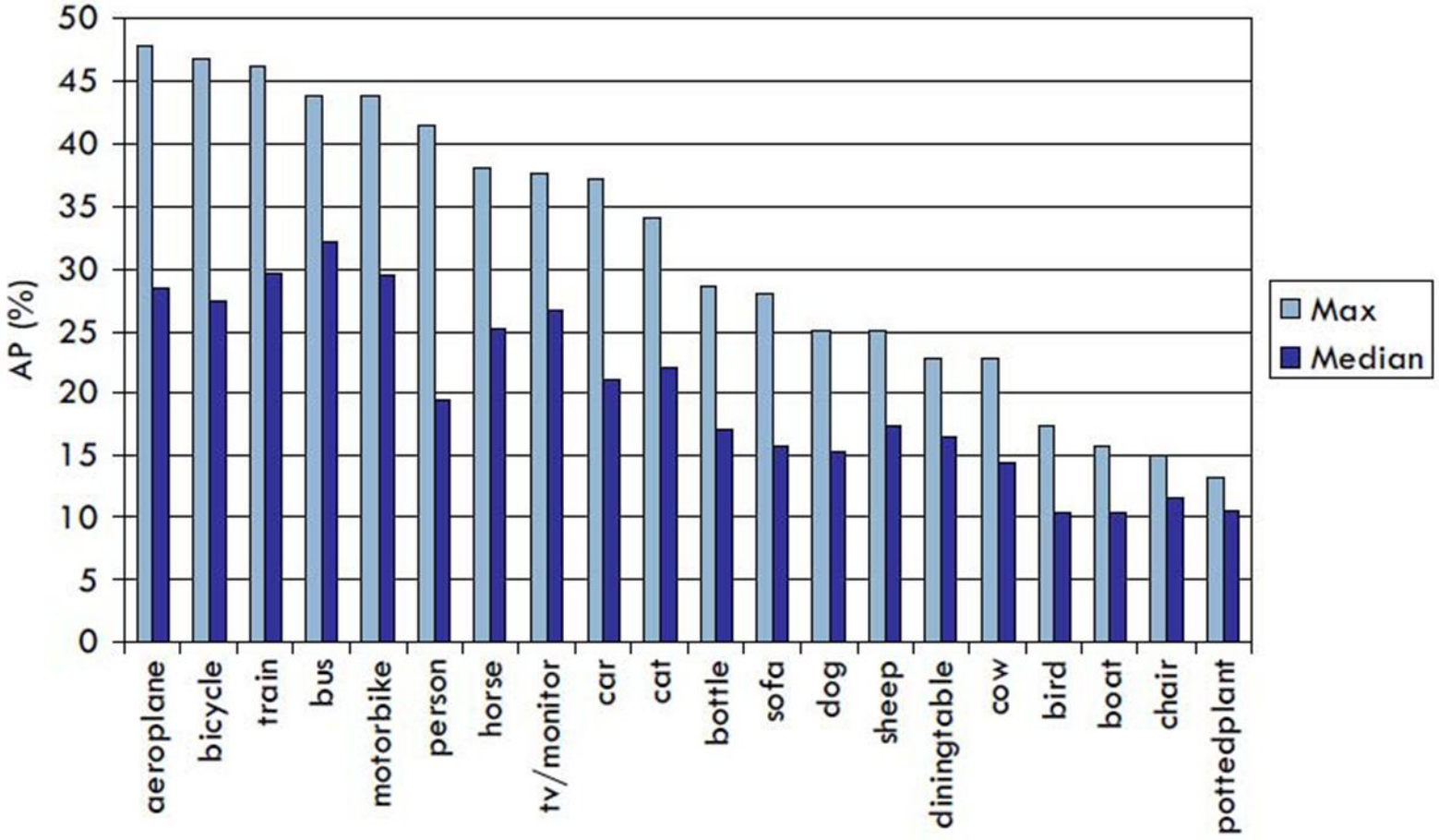
Примеры ошибок

OXFORD_MKL

UoCTTI_LSVM-MDPM

LEAR_CHI-SVM-SIFT-HOG-CLS

AP by Class


AP (%)
50
45
40
35
30
25
20
15
10
5
0
aeroplane
bicycle
train
bus
motorbike
person
horse
tv/monitor
car
cat
bottle
sofa
dog
sheep
diningtable
cow
bird
boat
chair
pottedplant
Max
Median

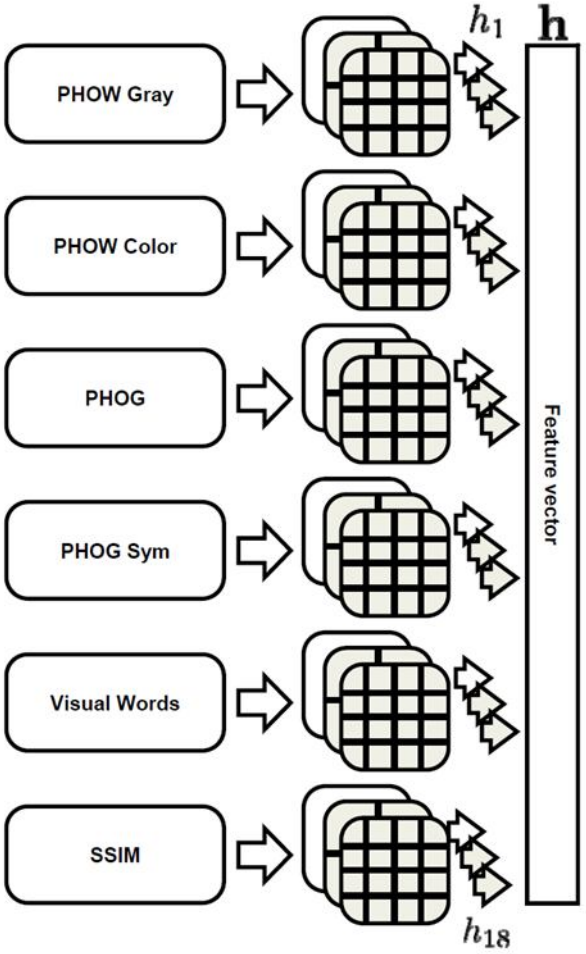
Microsoft Research

Multiple Kernels for Object Detection

Andrea Vedaldi, Varun Gulshan, Manik Varma, Andrew Zisserman

ICCV 2009

STAR-классификатор



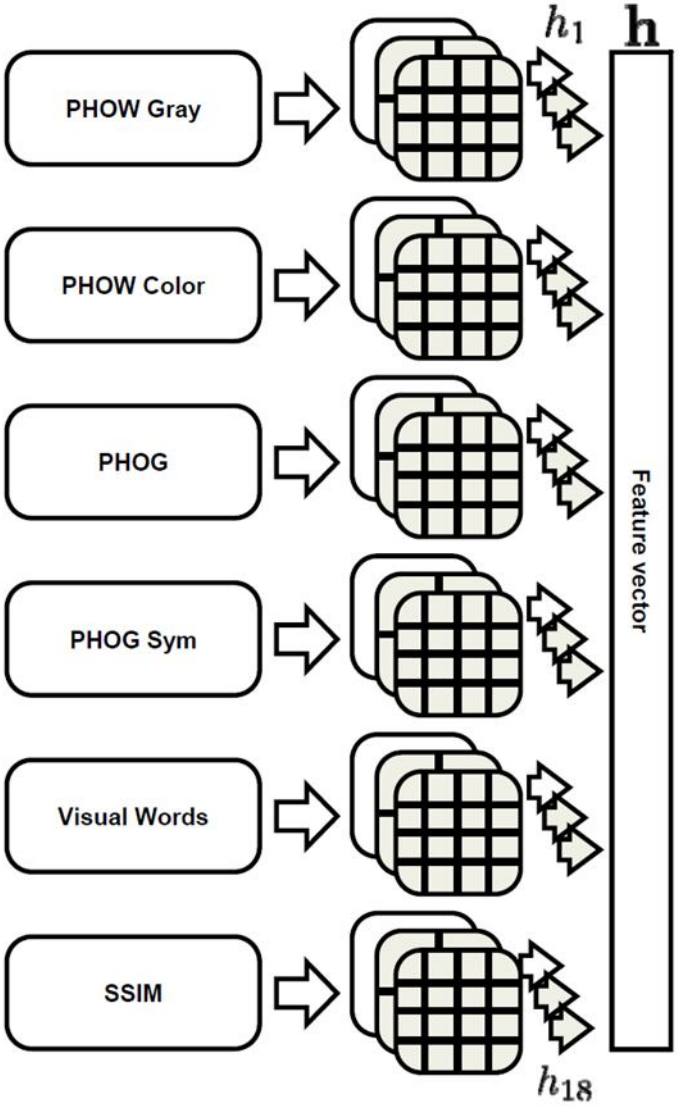
- Хотим использовать SVM со сложным ядром RBX-Chi2 для большого признакового пространства
  - 6 каналов
  - 3 уровня пирамиды (1+4+16 ячеек)

type	example	$f(z)$	$g(x,y)$	eval. complexity
linear	linear	z	xy	$O(N)$
quasi-lin.	χ^2	z	$\frac{2xy}{(x+y)}$	$O(BN)$
non-lin.	RBF-χ^2	$e^{-\gamma z}$	$\frac{(x-y)^2}{(x+y)}$	$O(MBN)$

- N – количество окон (10^5)
- M – количество опорных векторов (10^3)
- B – размерность гистограммы (10^5)

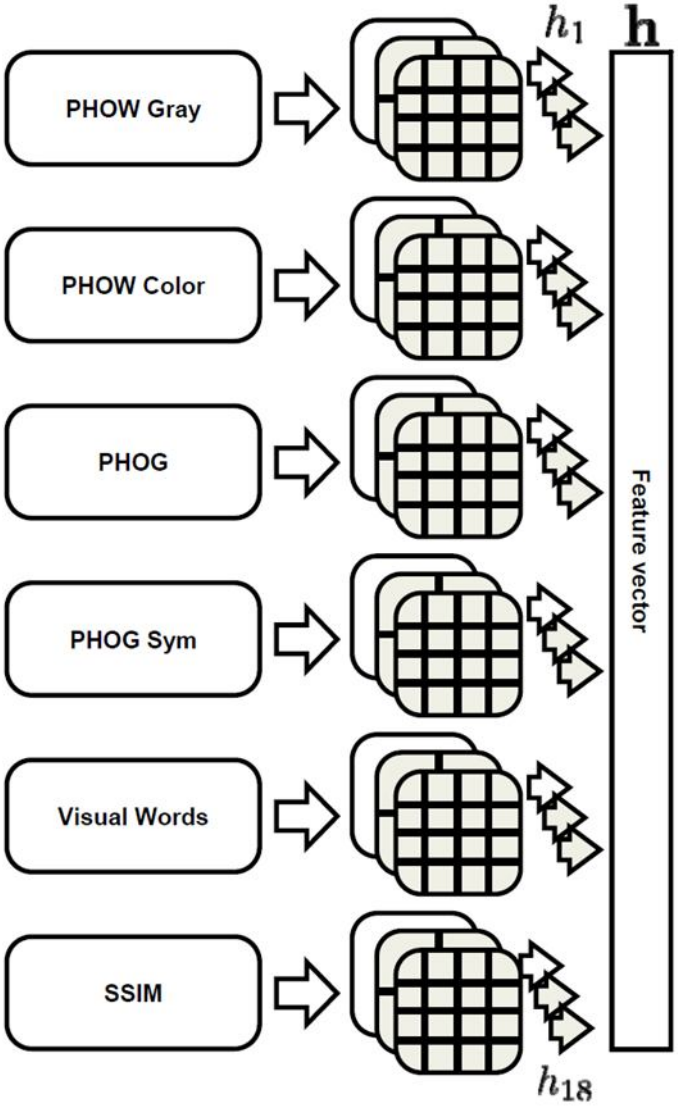
Andrea Vedaldi, Varun Gulshan, Manik Varma, Andrew Zisserman **Multiple Kernels for Object Detection,** ICCV 2009

Признаки



- Мешок слов
  - Harris-Laplace + SIFT, квантованы на 3000 слов, сжаты до 64
- Плотные слова (PhowGray, PhowColor)
  - По регулярной сетке каждые 5 пикселей SIFT
  - 4 масштаба – 10,15,20,25 пикселей
  - 3 HSV канала для цветной версии
  - 300 слов

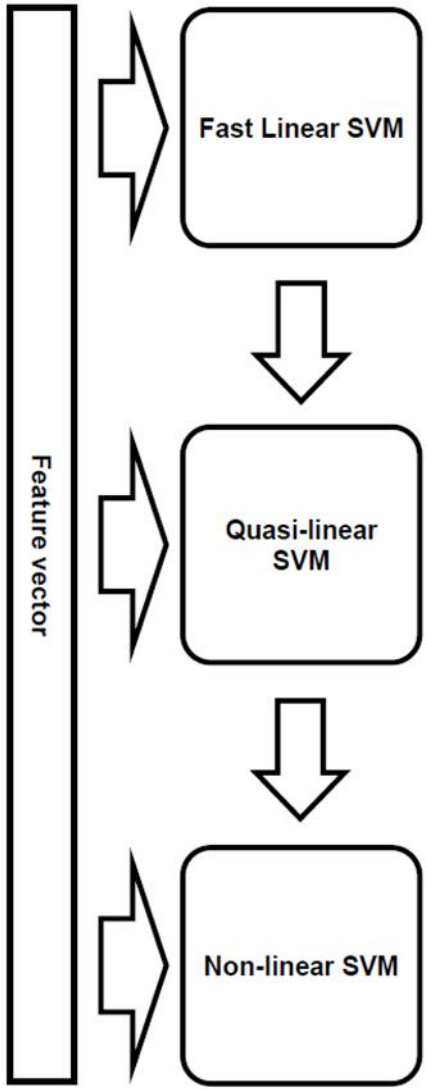
Признаки



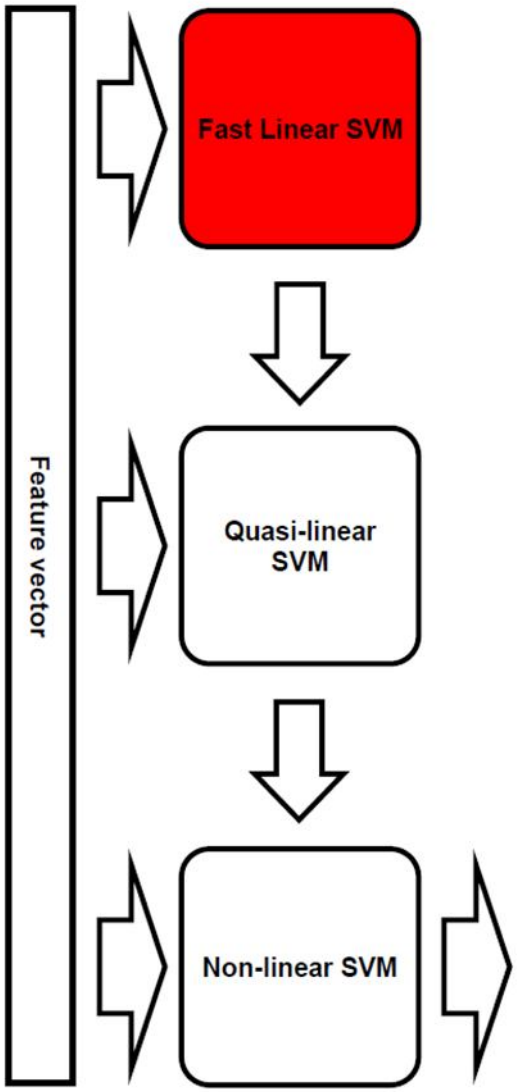
- Гистограмма ориентированных краев
  - Детектор Canny
  - Ориентация и вес каждому пикселю границы по градиенту
  - Гистограмма ориентаций из 8 ячеек
- Самоподобные особенности
  - Self-similarity features (SSIM)
  - По плотной решетке с шагом 5 пикселей
  - Карта корреляции фрагмента 5x5 пикселей в окне радиуса 40
  - Квантование по углу (10 ячеек) и радиусу (3 ячейки)
  - 300 слов

Каскад классификаторов

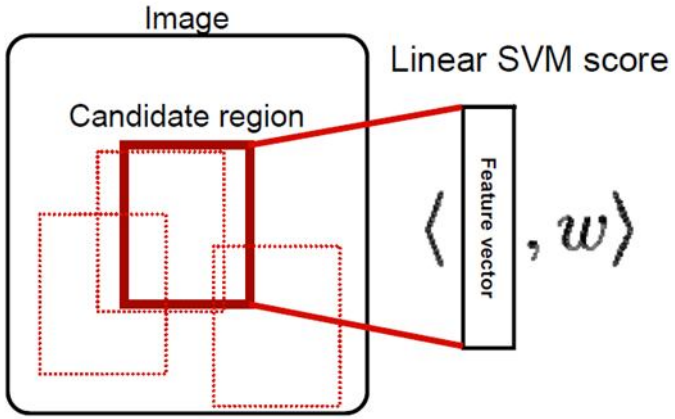
- Задача – оставить только ~100 окон-кандидатов для STAR-классификатора
- 3 этапа каскада
  - Каждый этап использует всё множество признаков
  - Сложность каскада определяется ядром
  - Для простых ядер если алгоритмы ускоренного вычисления
- Стандартная обработка его результатов
  - Выделение локальных максимумов
  - Отброс перекрытий

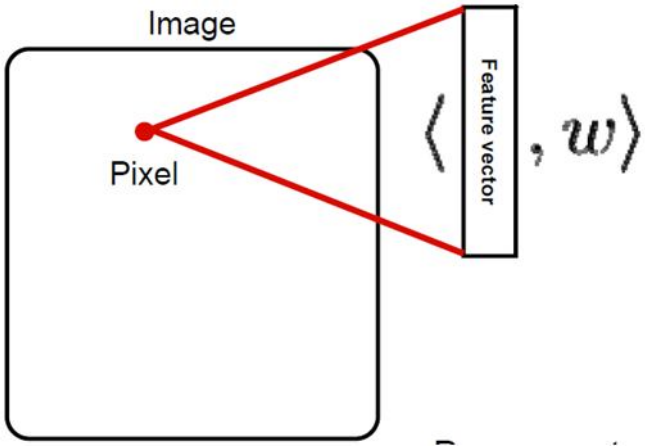


Каскад

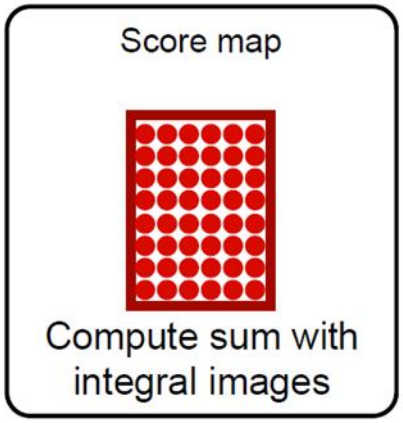


Быстрый линейный MOB

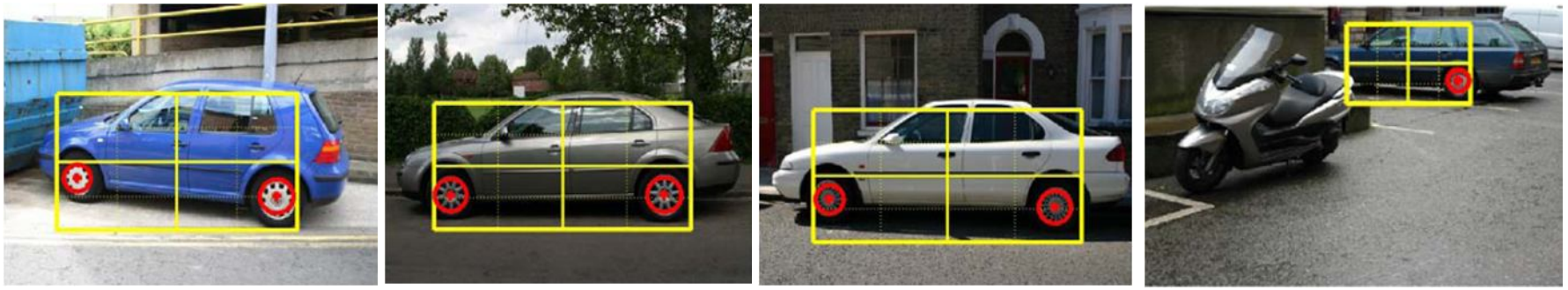




- w можно вычислить заранее, поэтому зависимость от M (числа опорных векторов убирается)
- Если гистограммы не нормализовывать, тогда можно вычислить score для каждого пикселя
- Вычисление MOB – суммирование по пикселям, используем интегральное изображение

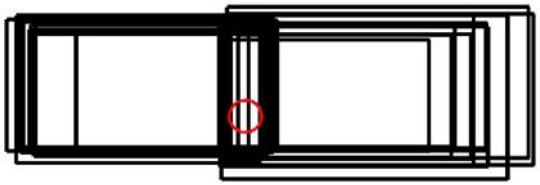


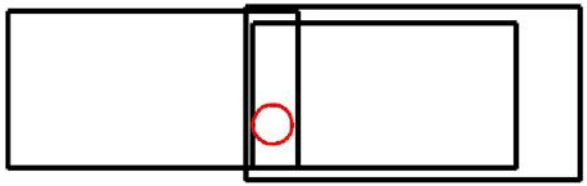
«Прыгающие окошки»

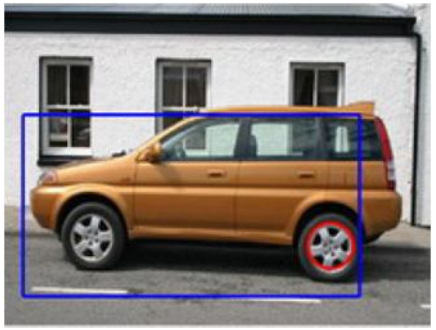


• Идея: выделим «слова», которые «подсказывают» положение объекта

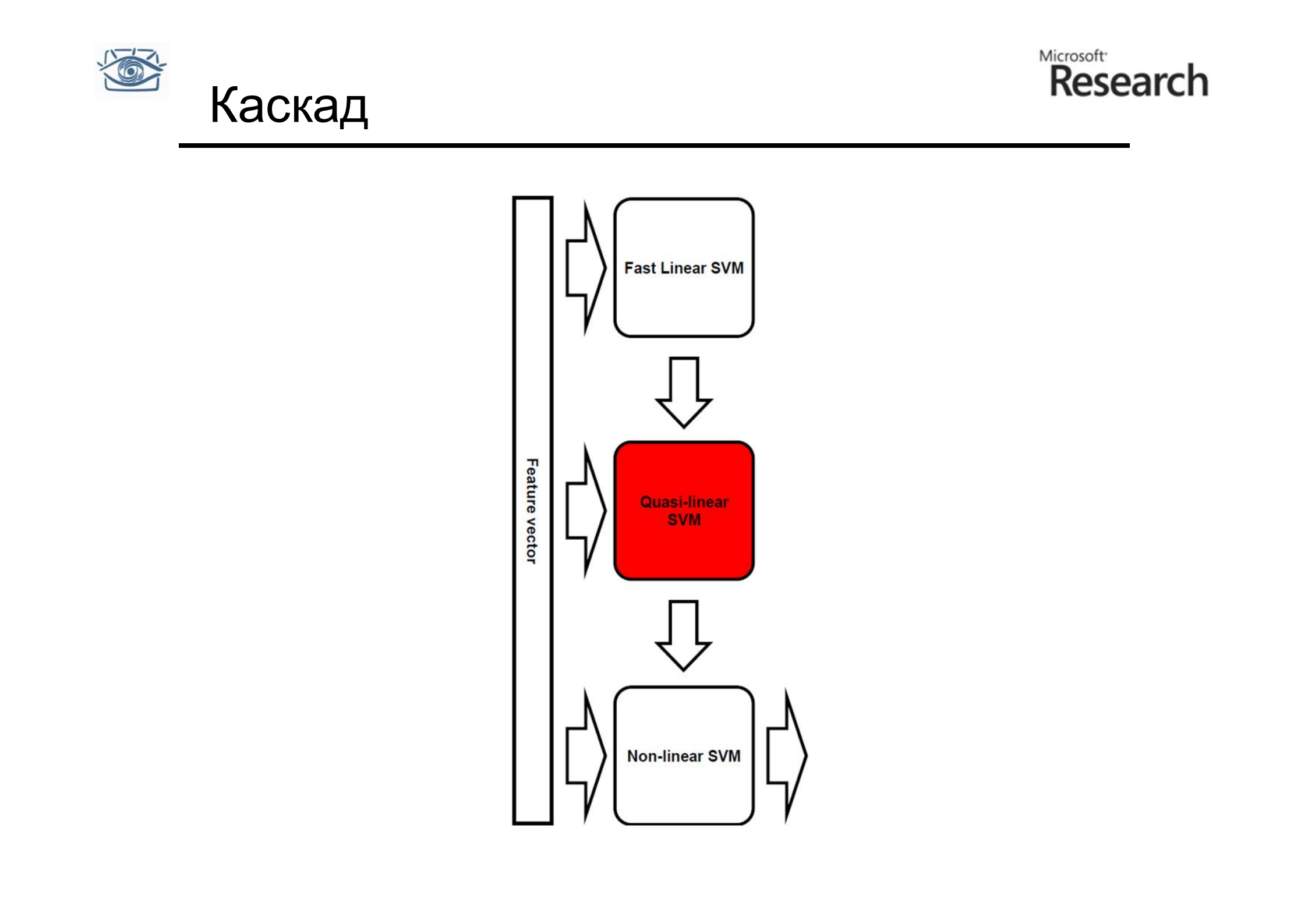
• Обучим положение/масштаб/пропорции «региона интереса» относительно визуального слова



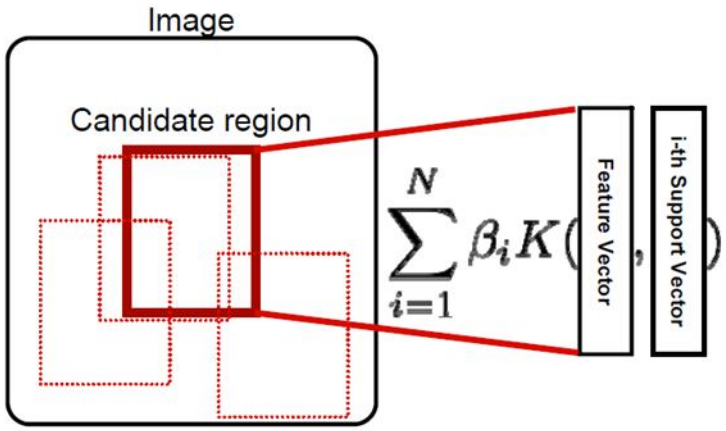




• Голоса будем объединять стандартной процедурой (сдвиг среднего)


Microsoft Research
Каскад
Feature vector
Fast Linear SVM
Quasi-linear SVM
Non-linear SVM

Квази-линейные MOB



- Можем сократить сложность до #окон × #размер гистограммы

Квази-линейное ядро можно разложить в сумму по ячейкам гистограммы:

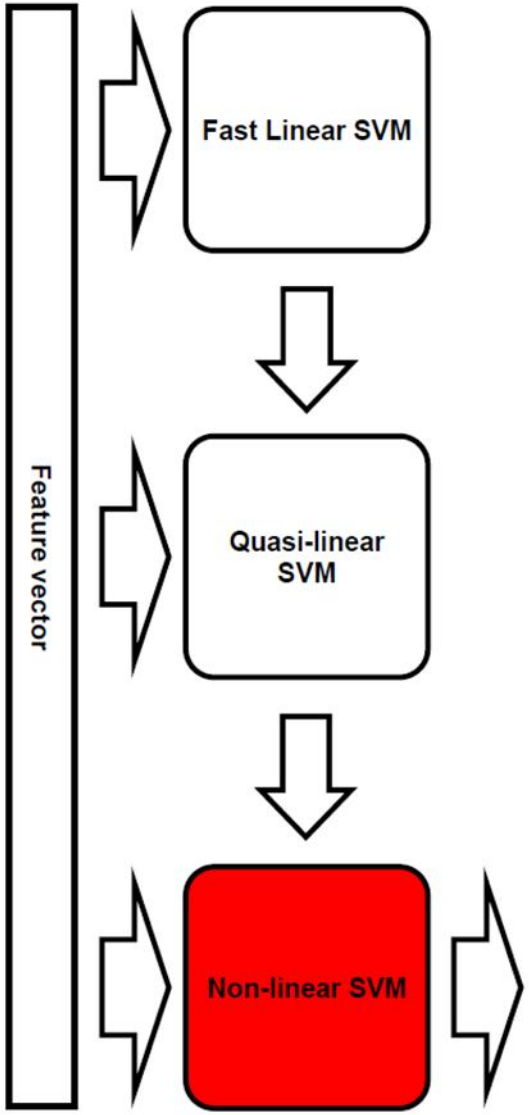
$$K(x,y) = \sum_{j=1}^{d} k(x_j, y_j)$$

Затем просчитать заранее сумму по всем опорным векторам

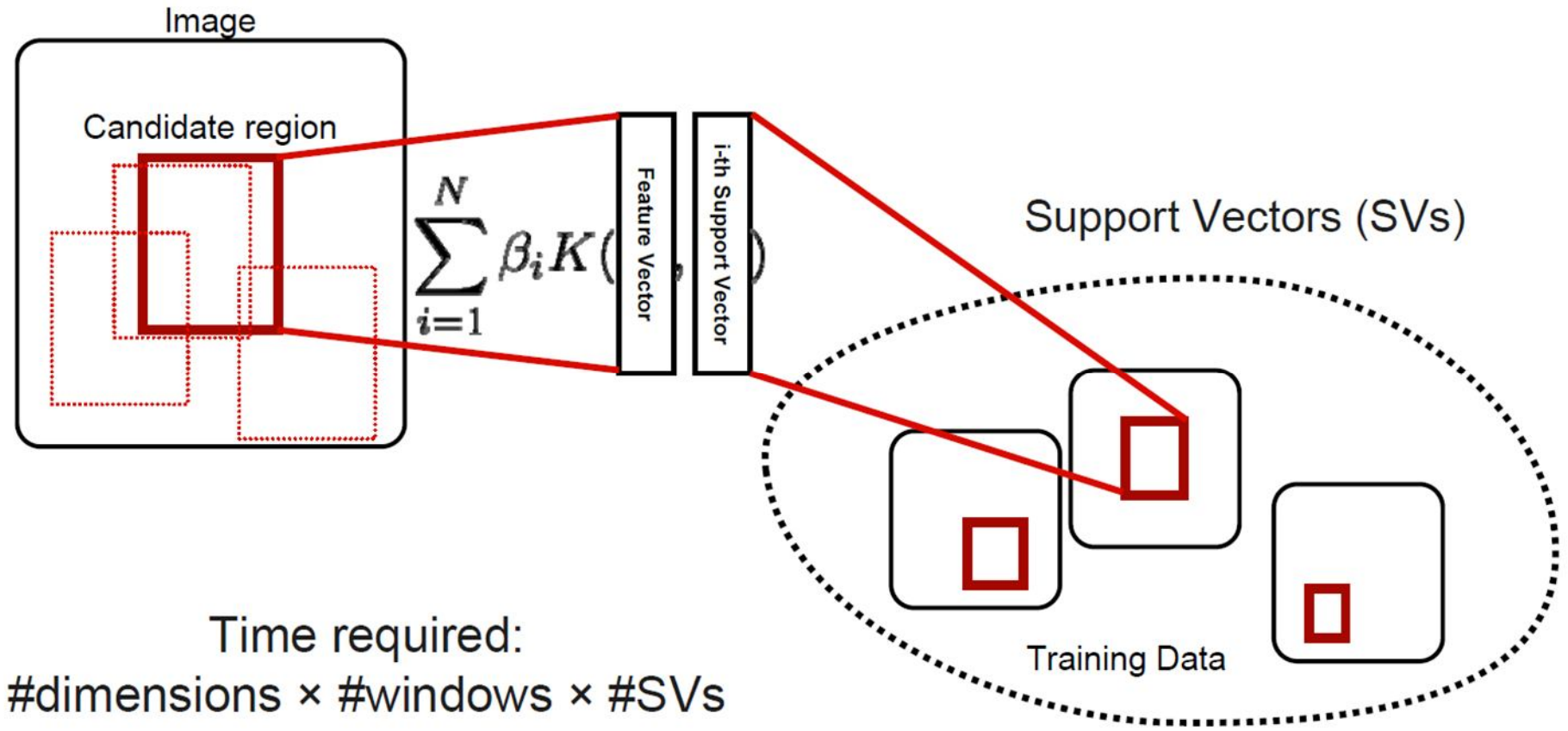
$$\sum_{j=1}^{d} \sum_{i=1}^{N} \beta_j k(x_j, y_j)$$

$$\psi_i(y_i)$$

Каскад


Feature vector
Fast Linear SVM
Quasi-linear SVM
Non-linear SVM

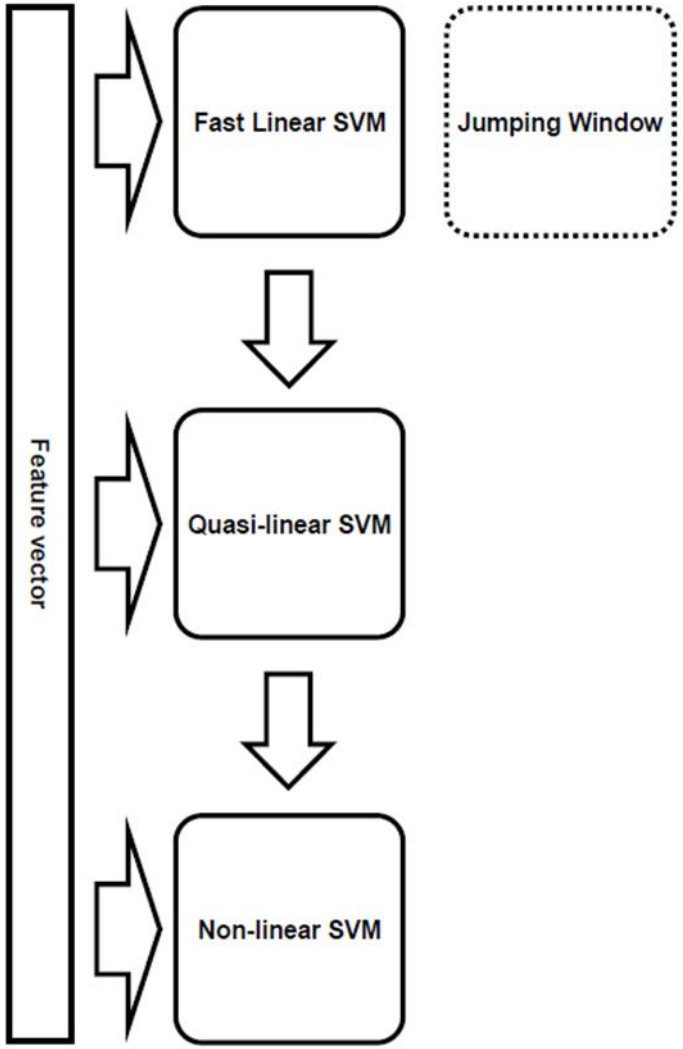
Нелинейные MOB



Time required:
#dimensions × #windows × #SVs

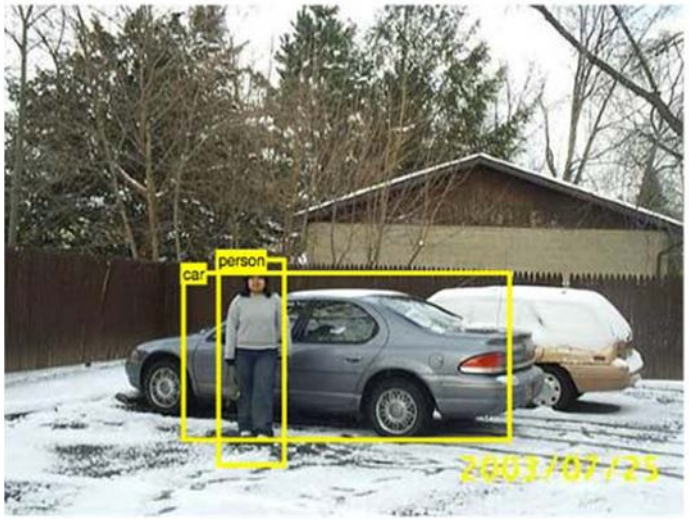
Каскад

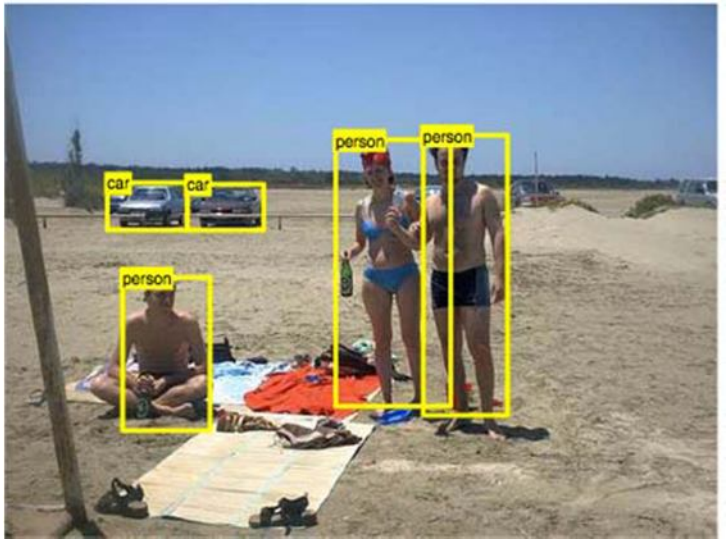
- Первая стадия
  - linear SVM
  - (or jumping window)
  - time: #windows
- Вторая стадия
  - quasi-linear SVM
  - *χ2 kernel*
  - time: #windows × #dimensions
- Третья стадия
  - non-linear SVM
  - *χ2-RBF kernel*
  - time: #windows × #dimensions × #SupportVectors



Результаты

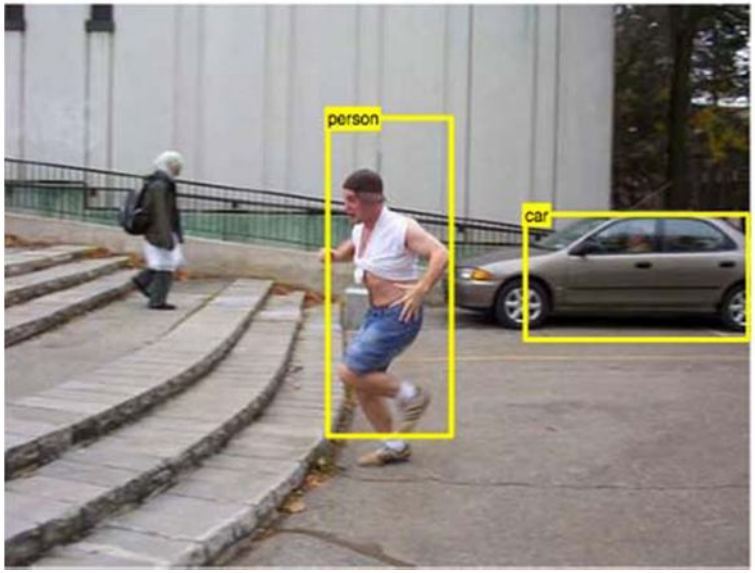
Microsoft Research

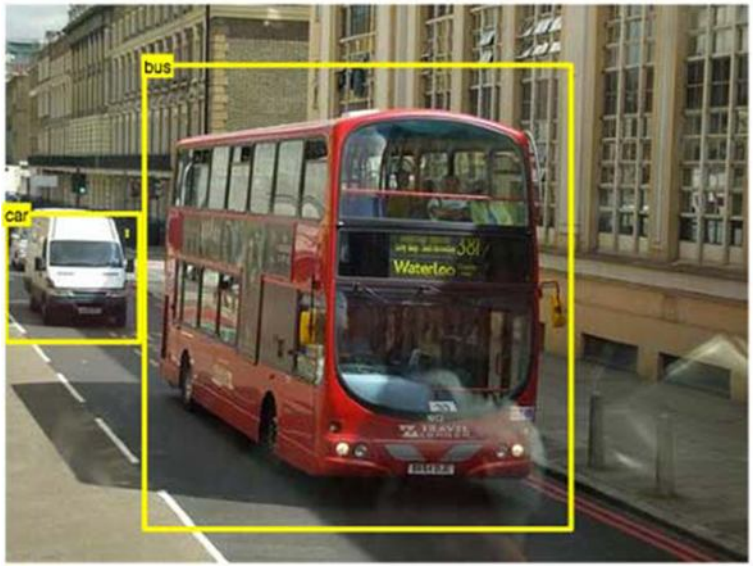




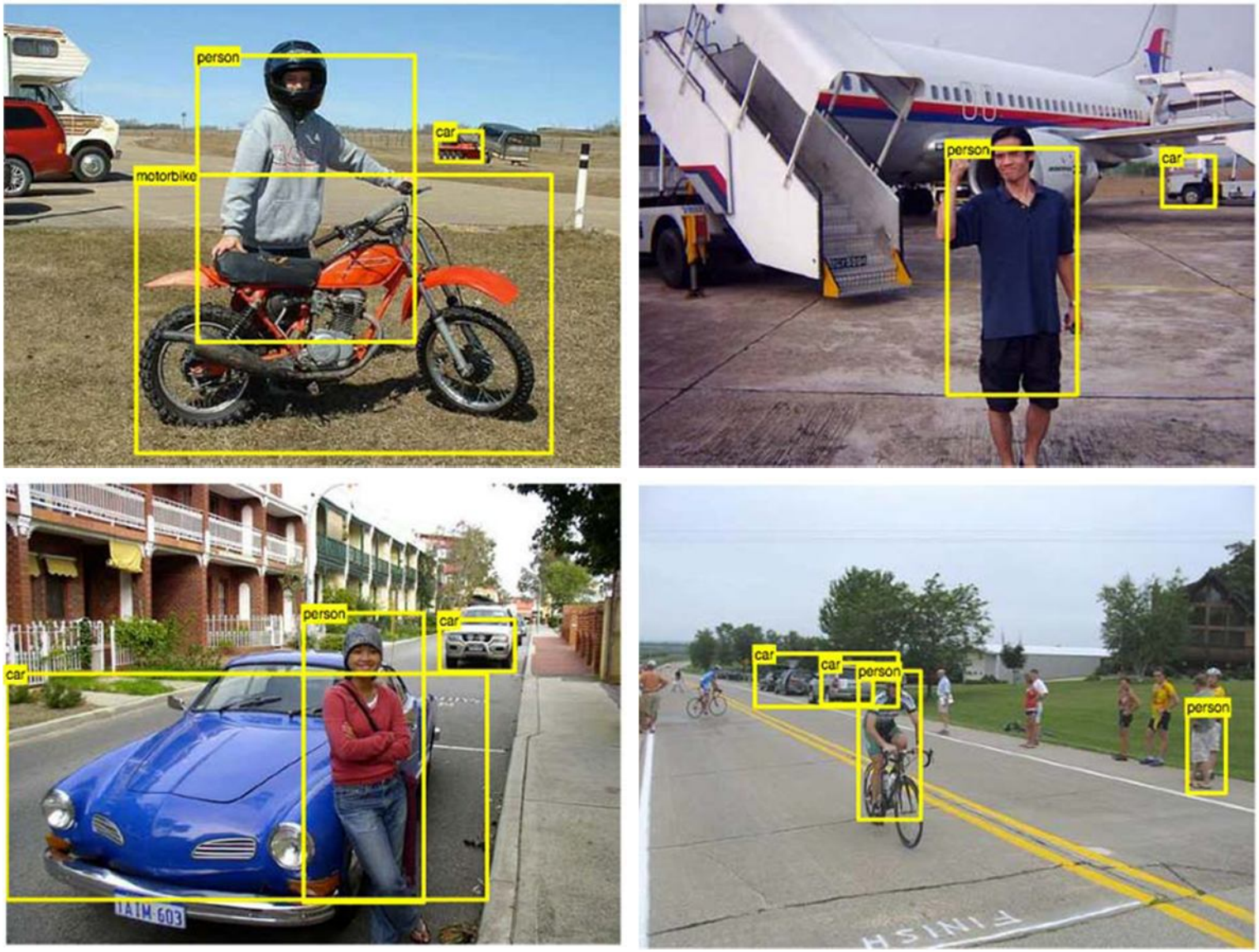
Результаты

Microsoft Research





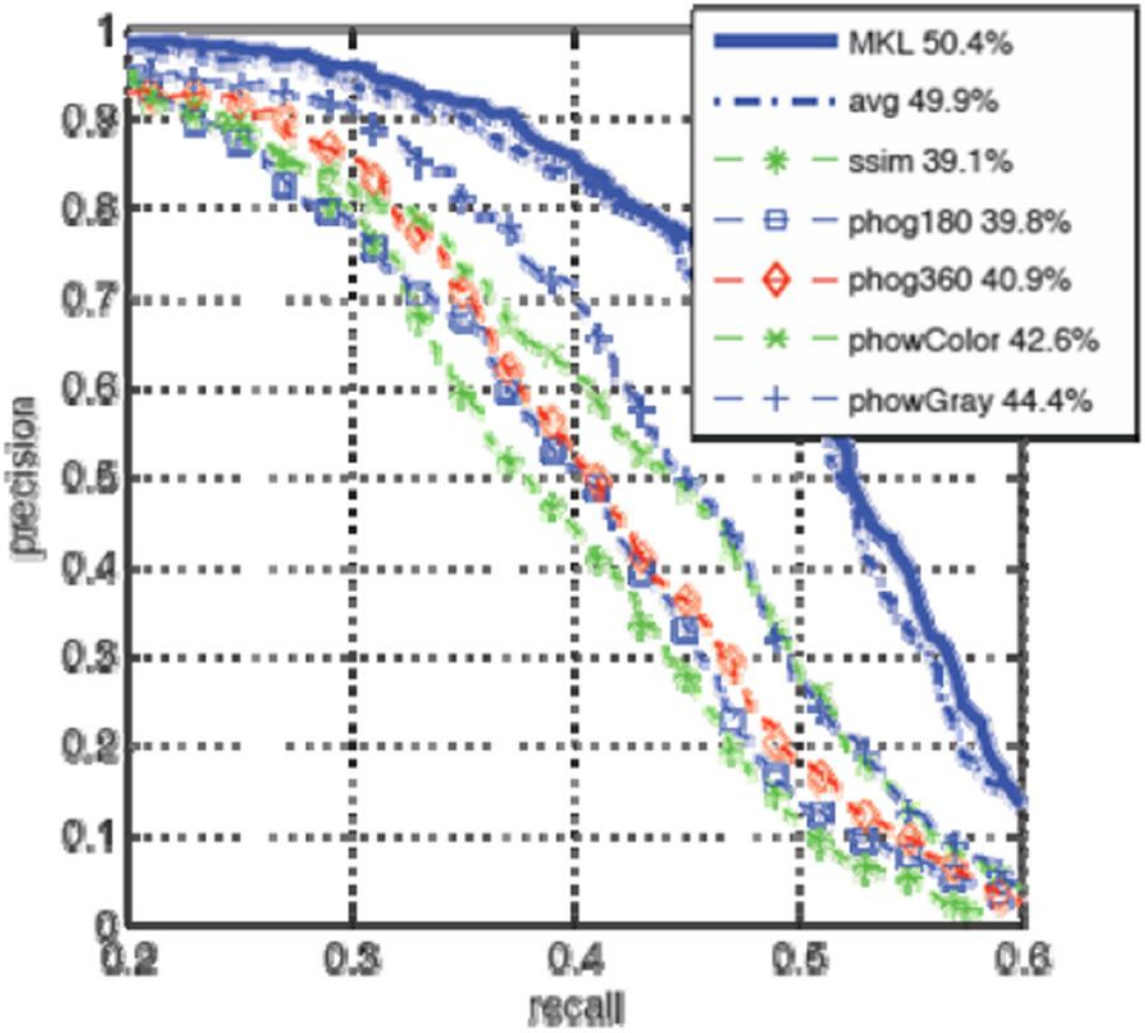
Результаты


person
car
motorbike
person
car
person
car
car
car
person
person

Microsoft Research

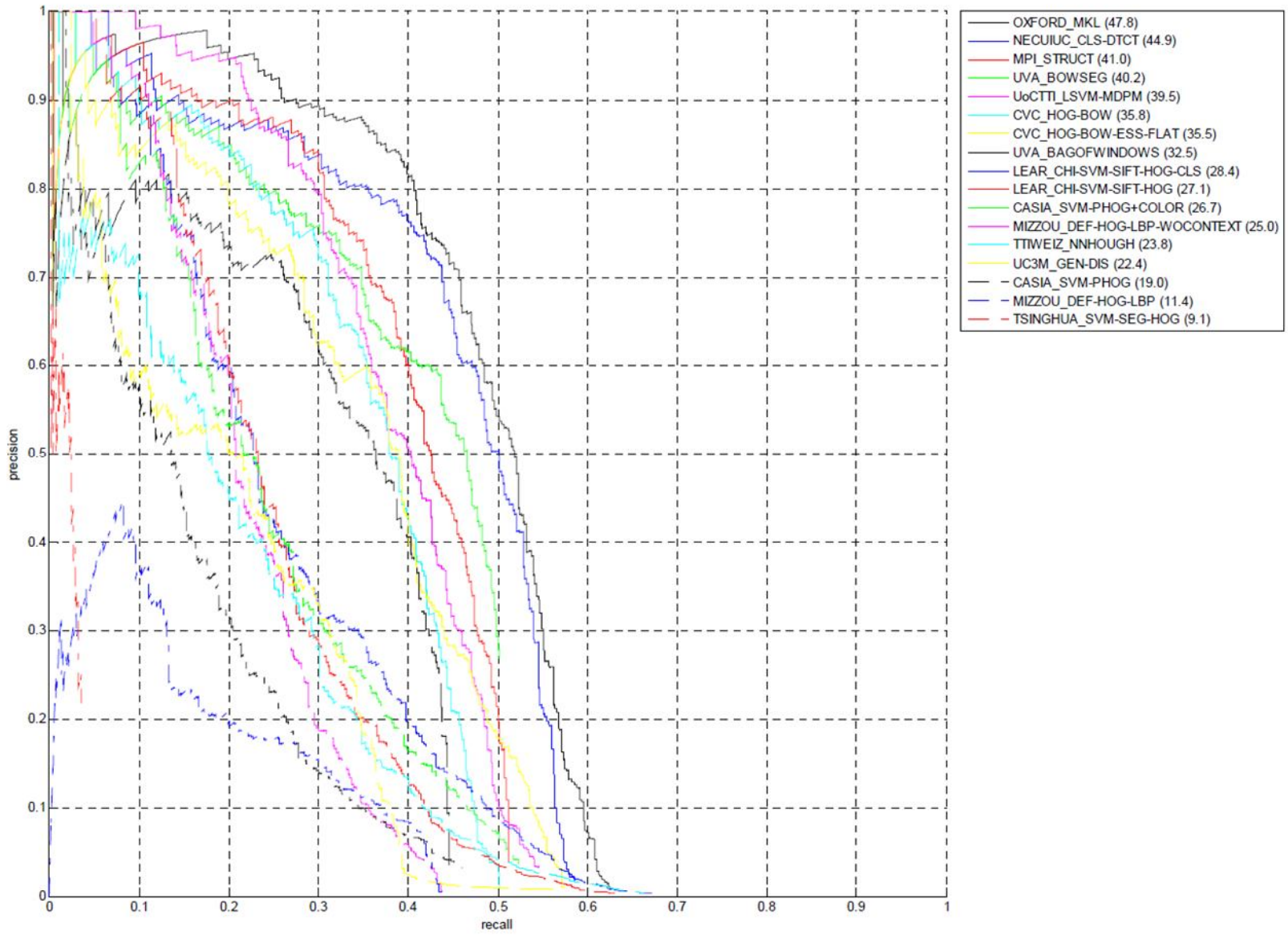
Одно и много ядер

- Несколько ядер дают серьезный прирост качества



Microsoft Research

VOC2009 Aeroplane



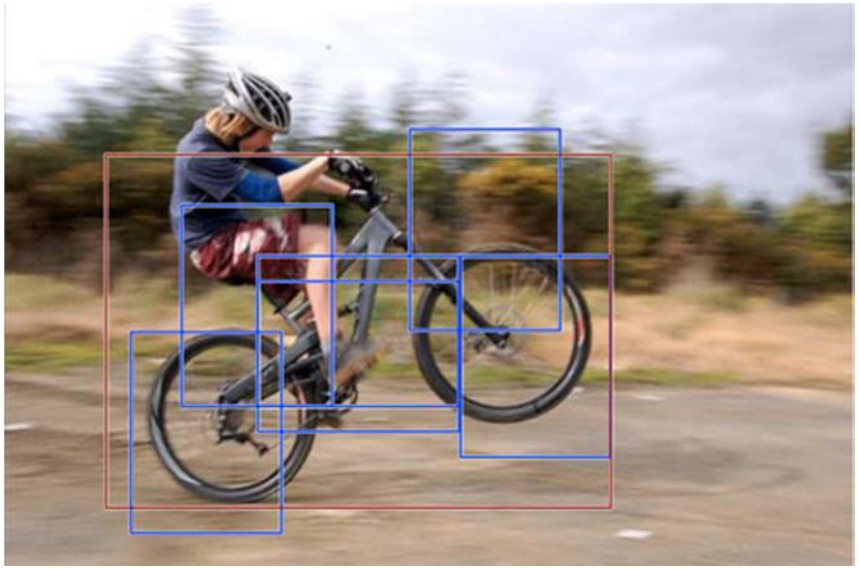
Microsoft Research

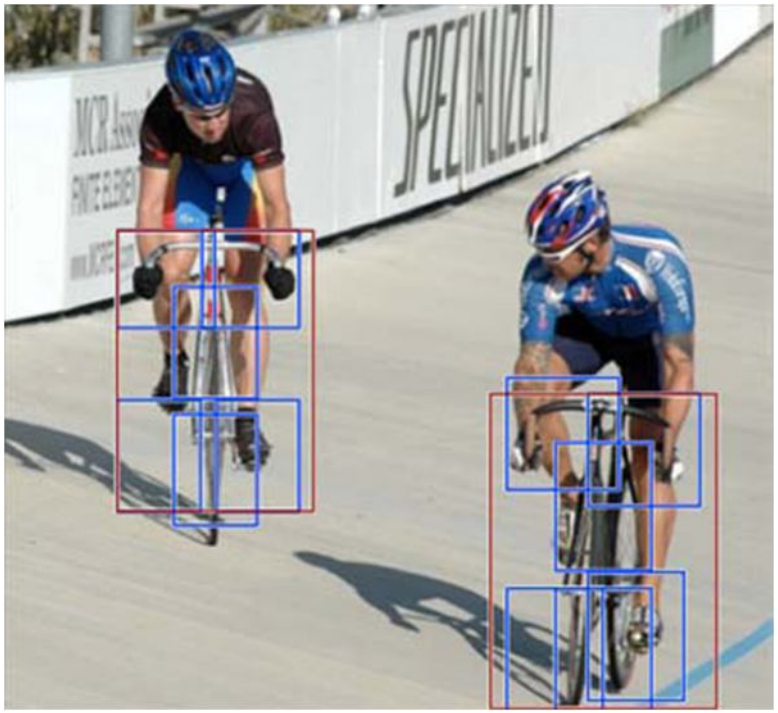
Object Detection with Discriminatively Trained Part Based Models

Pedro F. Felzenszwalb, David Mcallester, Deva Ramanan, Ross Girshick

PAMI 2010

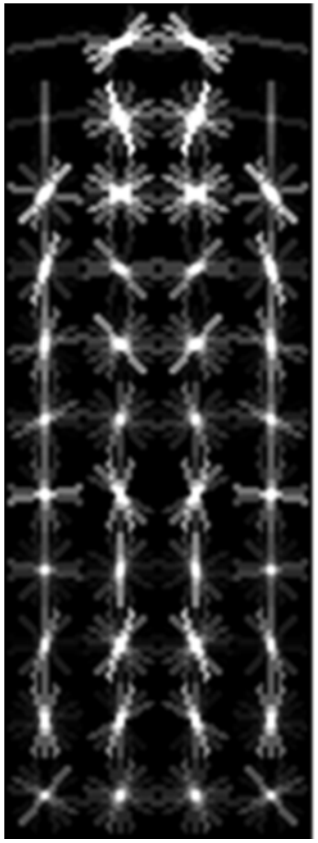
Подход



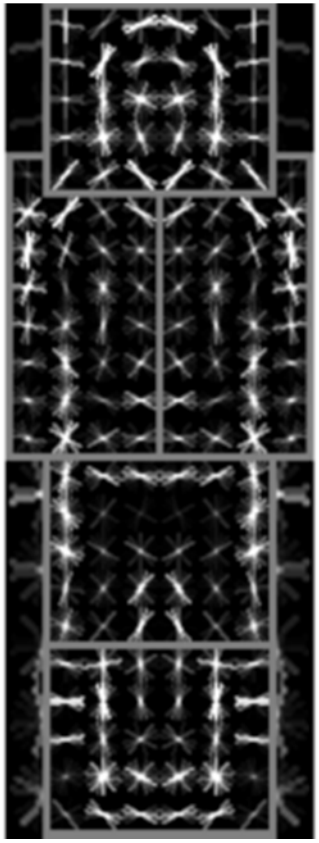


- Деформируемая модель
  - Одна модель для каждой точки обзора
  - Каждая компонента состоит из общего шаблона и деформируемых частей
  - Обучение только по рамкам

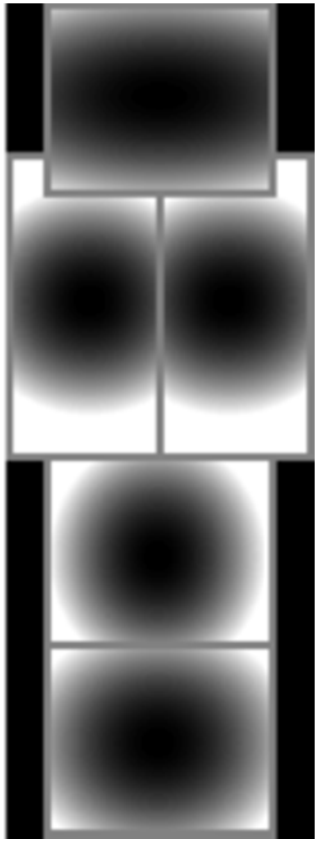
Модель



Корневой фильтр
Низкое разрешение

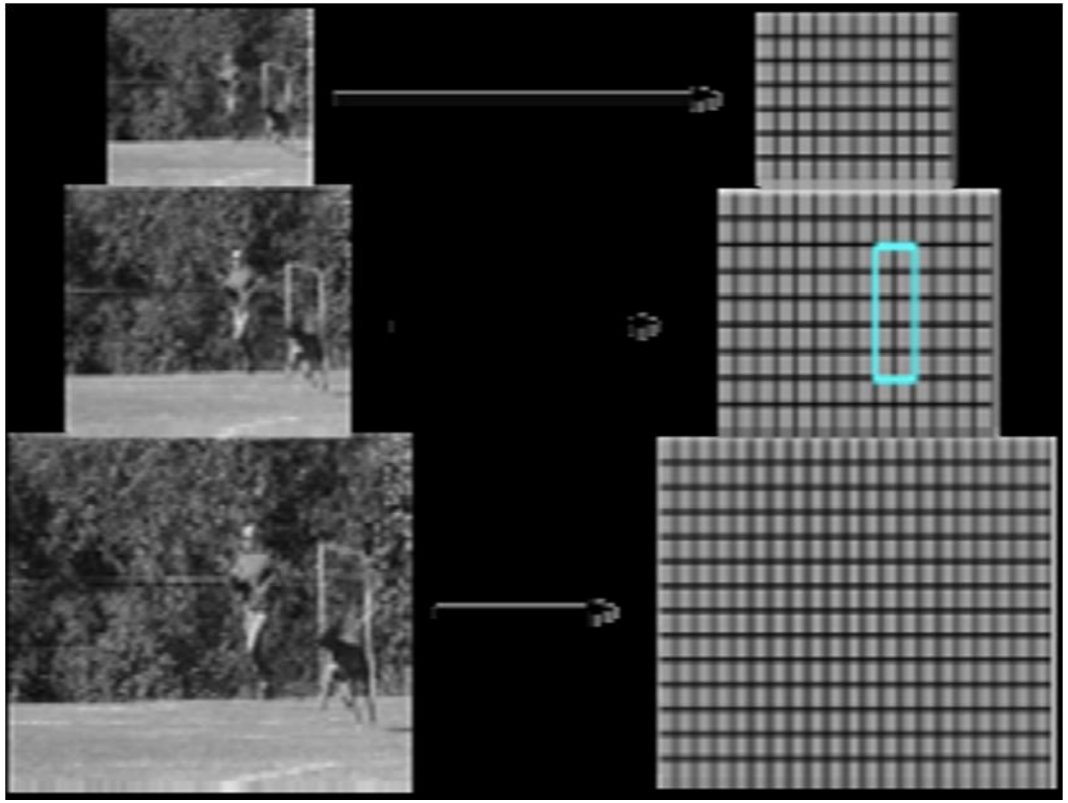


Фильтр части
2x разрешение



Модель
деформации

HOG-фильтр

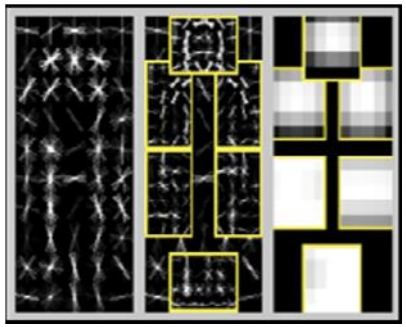


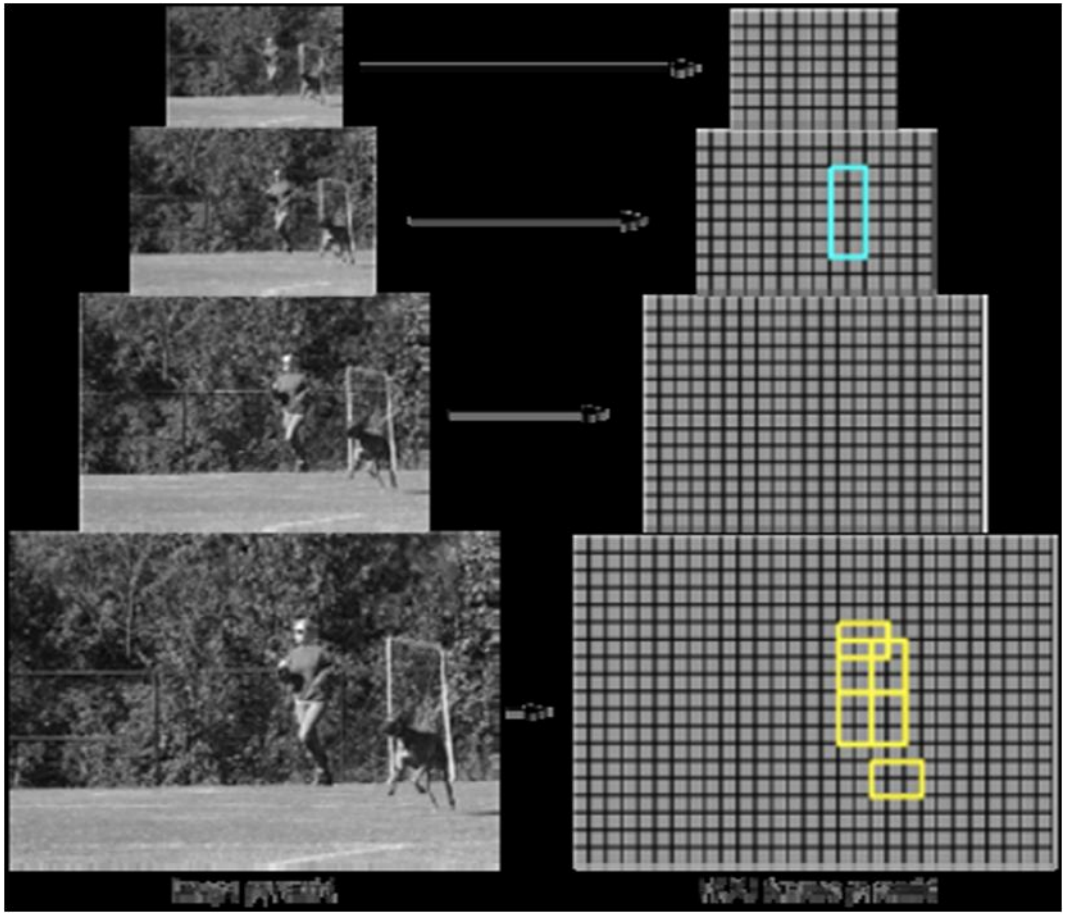
Фильтр

- Поиск: скользящее окно по положению и масштабу
- Фичи: HOG-фильтр
- Классификатор: линейный SVM

Гипотезы

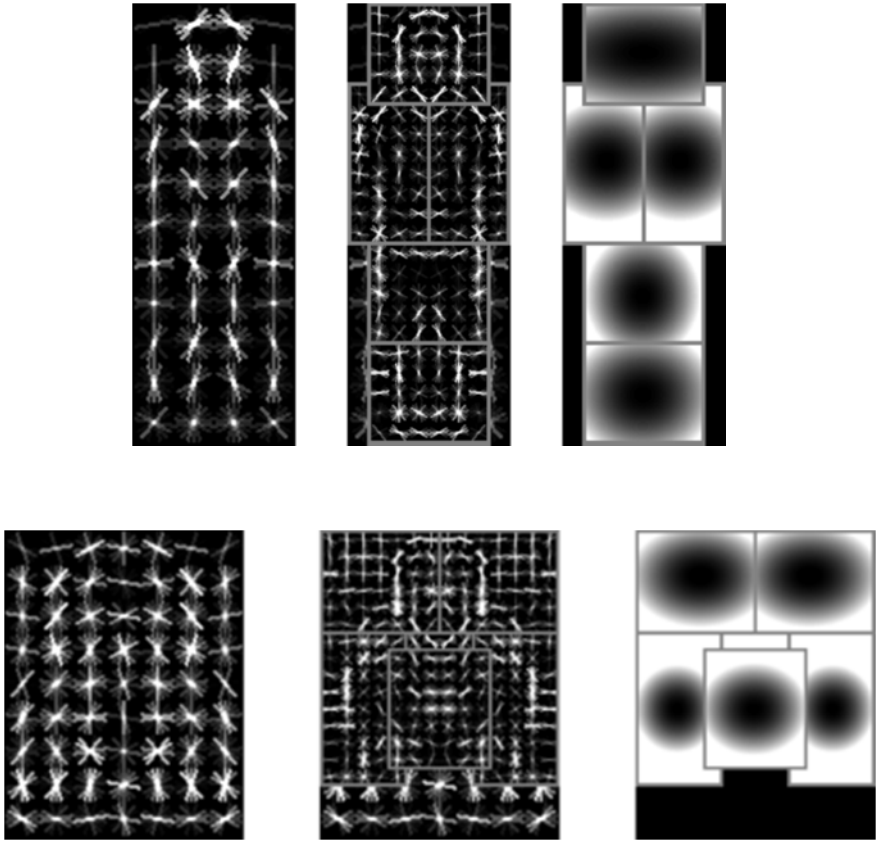
- Каждая часть – HOG-фильтр
- Базовое положение





«Качество» это сумма качеств частей минус пенальти за деформацию

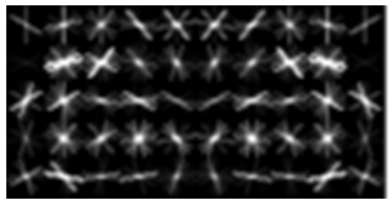
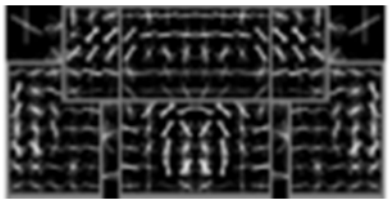
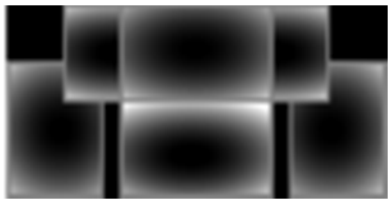
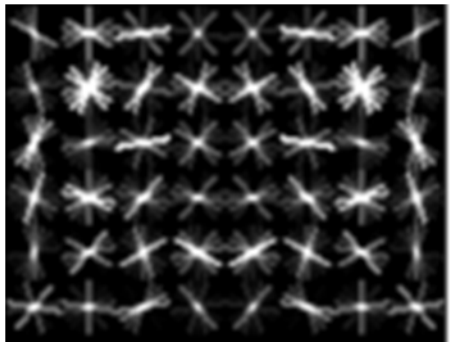
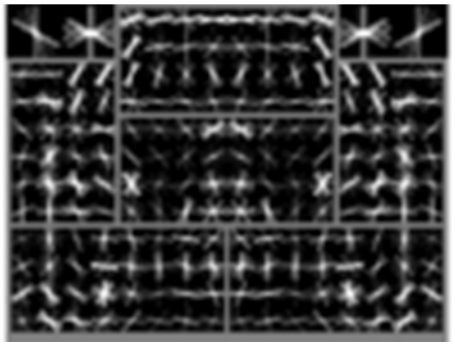
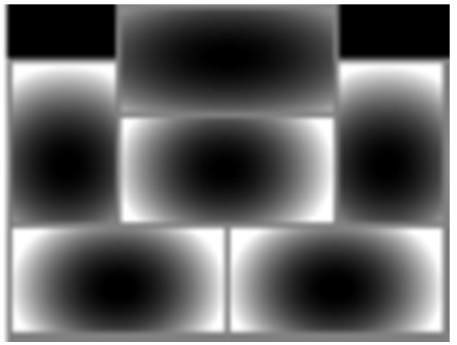
Модель человека



- Обучаем несколько моделей (смесь моделей) для разных случаев
- Можем справится с частичным перекрытием / обрезанинием

Модель машины

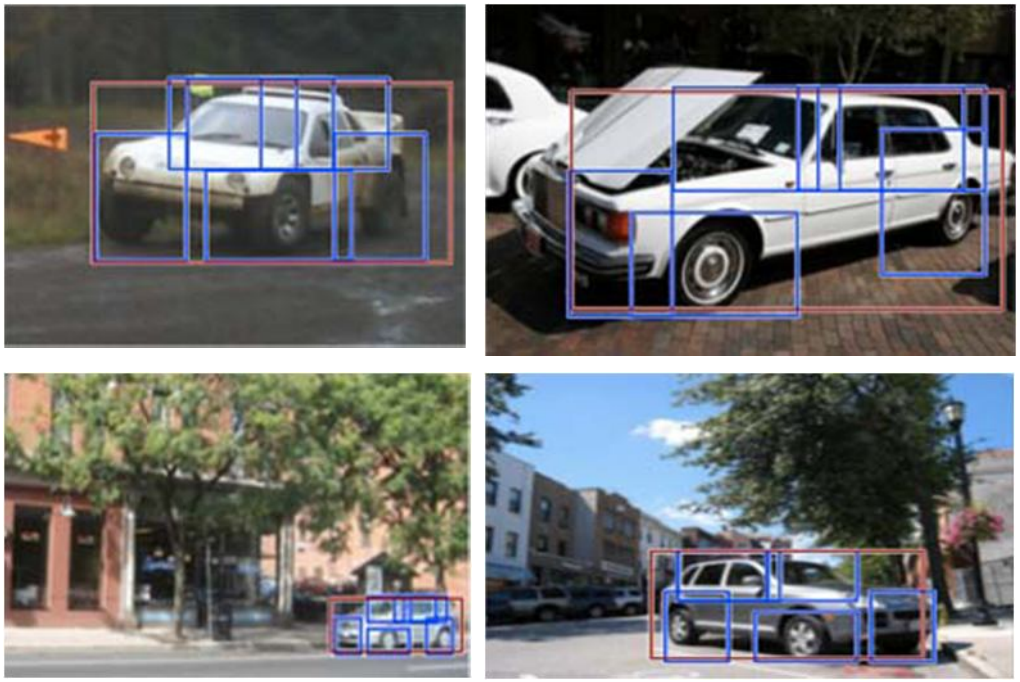
Microsoft Research

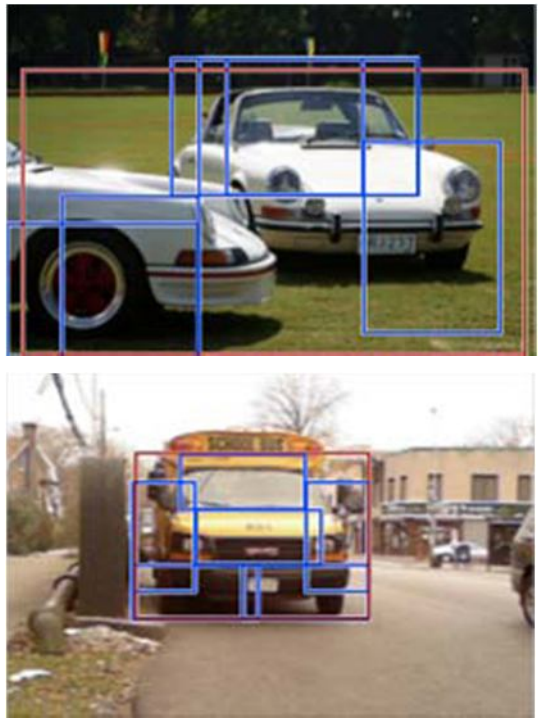
Результаты на машинах

Microsoft Research

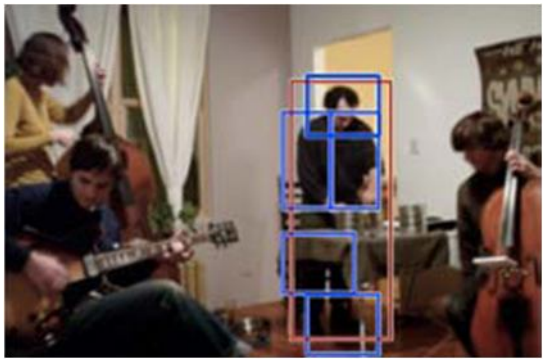
Правильные обнаружения с высокой оценкой

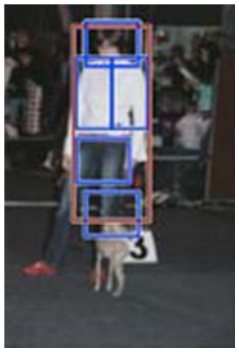
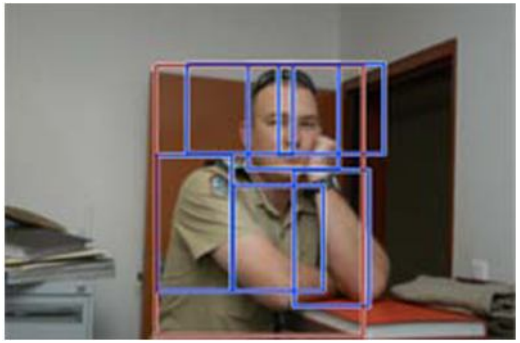
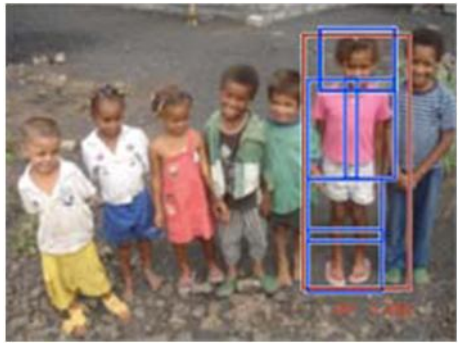


Ложные обнаружения с высокой оценкой



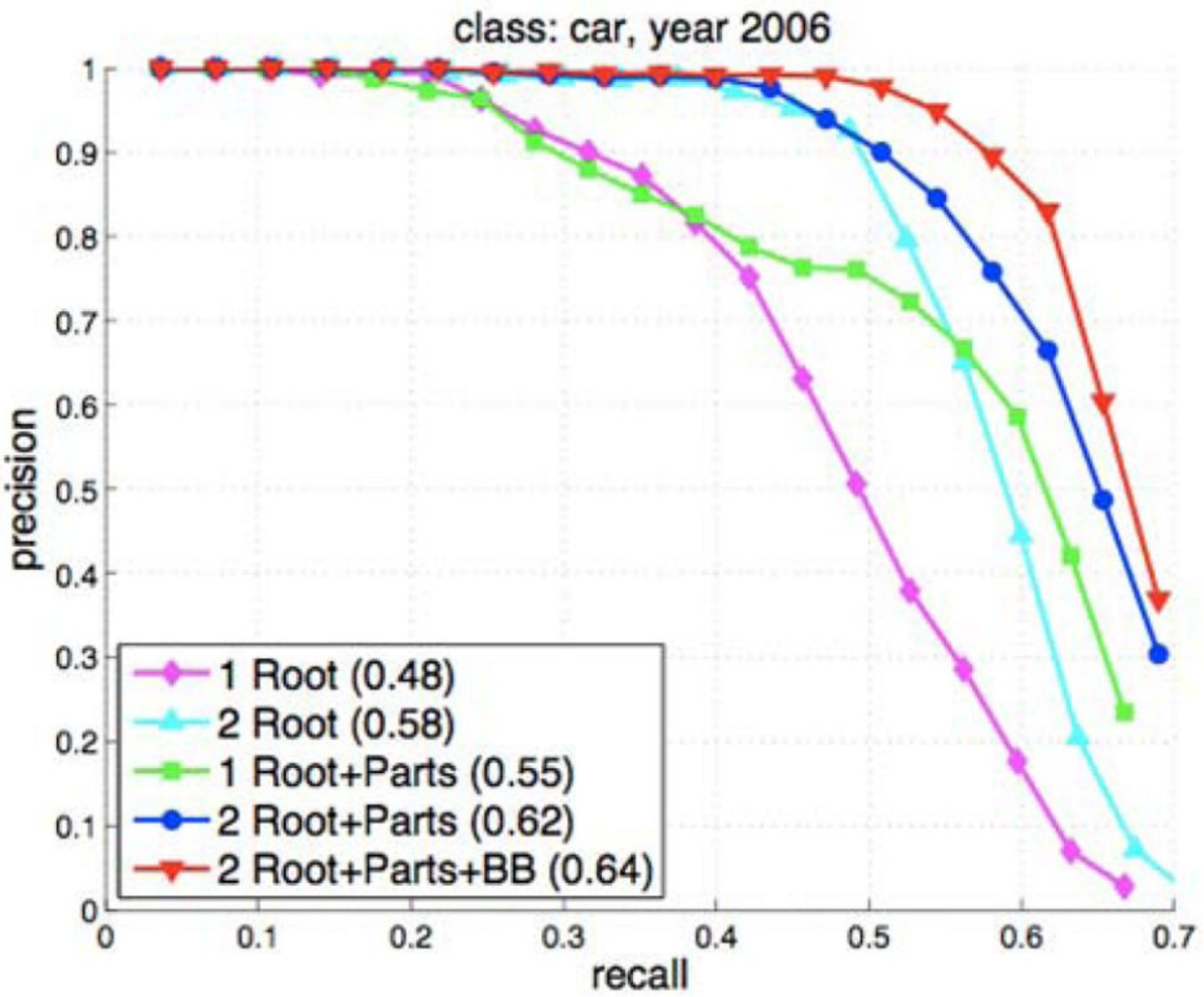
Обнаружение человека



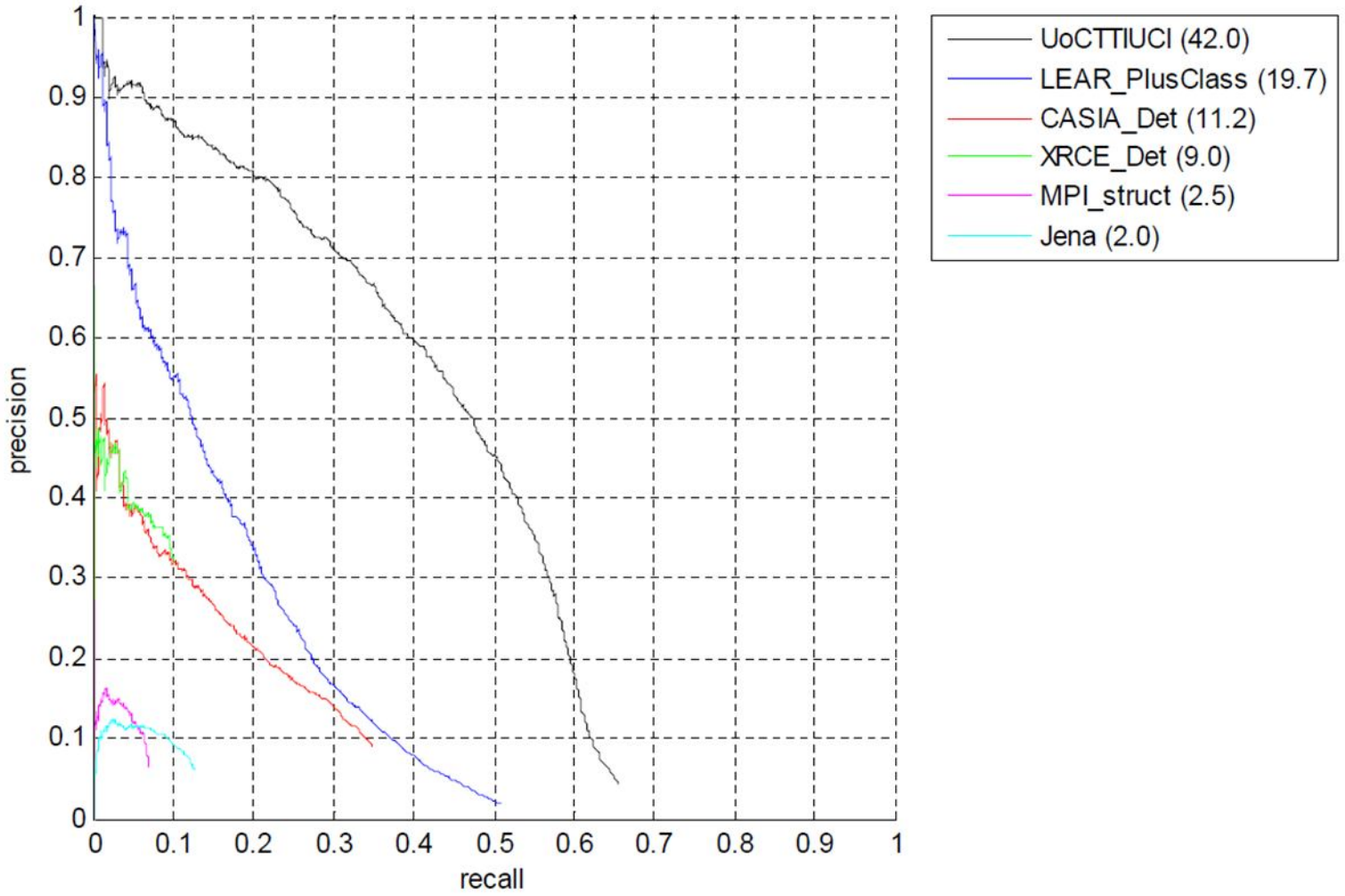





Сравнение моделей

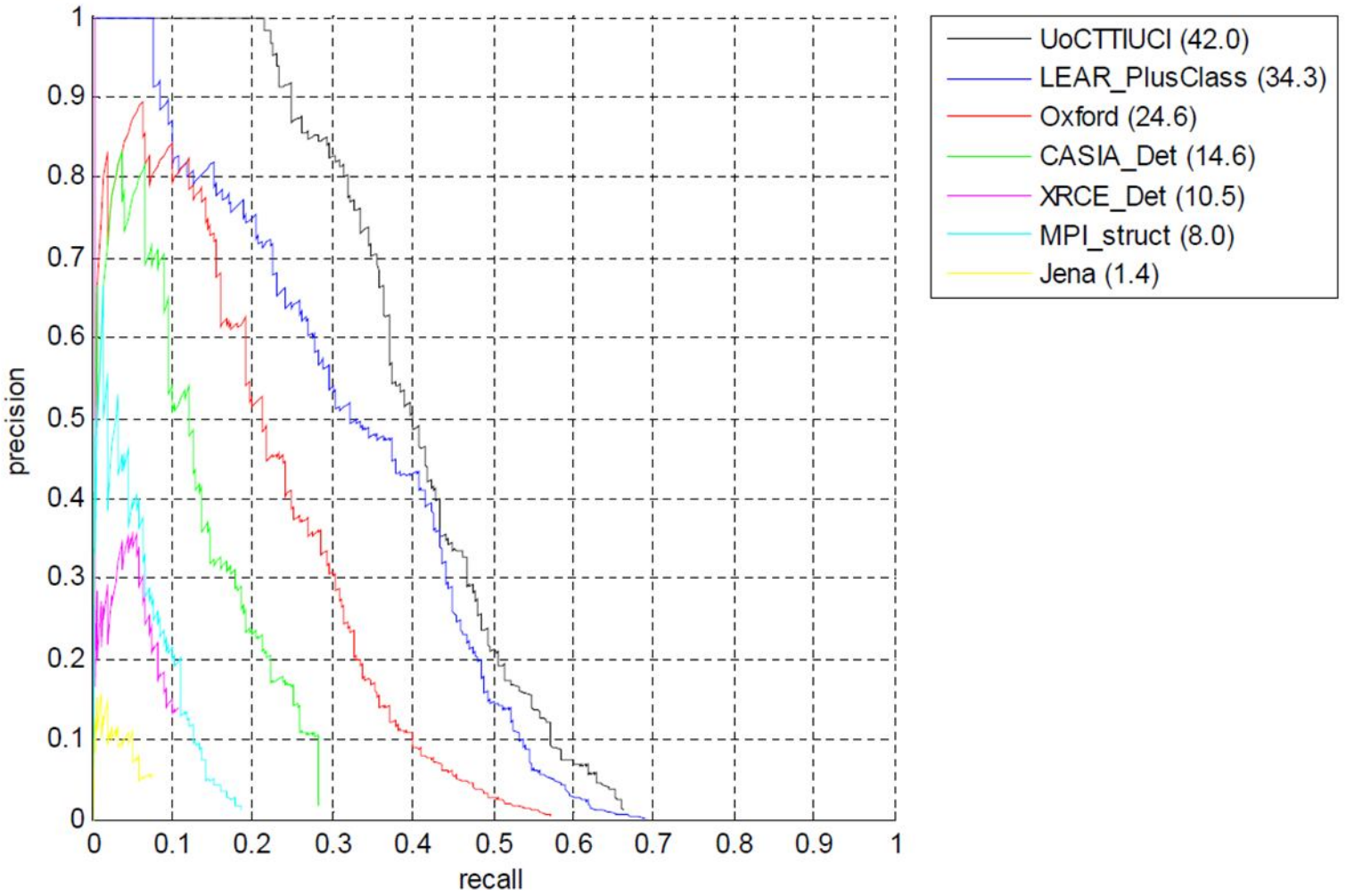
Microsoft Research



Результаты


precision
recall
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
UoCTTIUCI (42.0)
LEAR_PlusClass (19.7)
CASIA_Det (11.2)
XRCE_Det (9.0)
MPI_struct (2.5)
Jena (2.0)

Результаты


precision
recall
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
UoCTTIUCI (42.0)
LEAR_PlusClass (34.3)
Oxford (24.6)
CASIA_Det (14.6)
XRCE_Det (10.5)
MPI_struct (8.0)
Jena (1.4)

Microsoft Research

Направления развития

- Контекст
  - По свойствам сцены: GIST, BoW, stuff
  - По другим объектам
  - По геометрии сцены, пр. Hoiem et al CVPR 06
- Перекрытие / обрезание
  - Winn & Shotton, Layout Consistent Random Field, CVPR 06
  - Vedaldi & Zisserman, NIPS 09
  - Yang et al, Layered Object Detection, CVPR 10
- 3D
- Масштабирование до тысяч классов
  - Torralba et al, Feature sharing
  - ImageNet
- Слабая и шумные разметки

Microsoft Research

На следующей лекции

- It’s all about the Data!
- Применение больших коллекций изображений для решения разных задач