

Детектирование объектов

Осокин Даниил, daniil.osokin@gmail.com

Летняя школа 2016, Itseez/Intel/ННГУ

Июль 2016

Что может рассказать изображение?



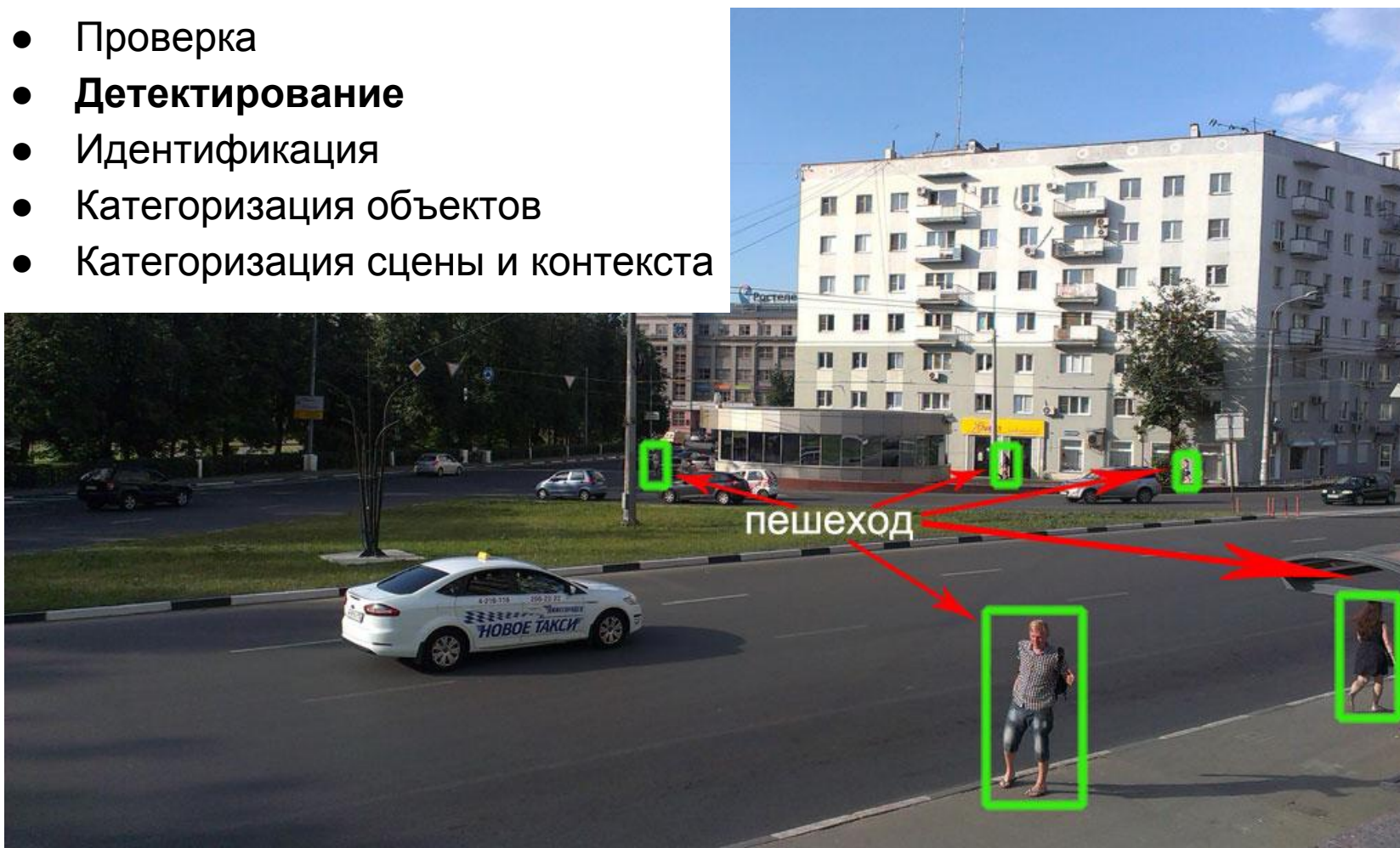
Распознавание объектов

- Проверка
- Детектирование
- Идентификация
- Категоризация объектов
- Категоризация сцены и контекста



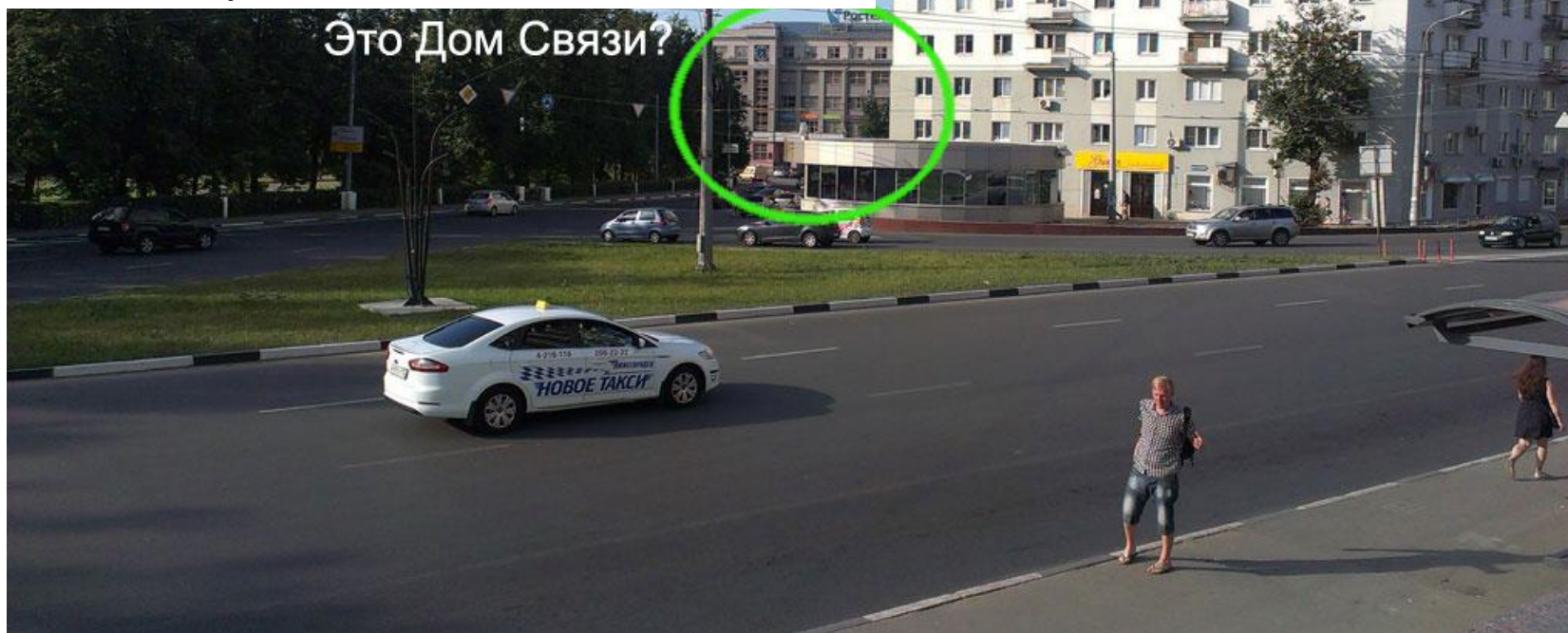
Распознавание объектов

- Проверка
- **Детектирование**
- Идентификация
- Категоризация объектов
- Категоризация сцены и контекста



Распознавание объектов

- Проверка
- Детектирование
- **Идентификация**
- Категоризация объектов
- Категоризация сцены и контекста



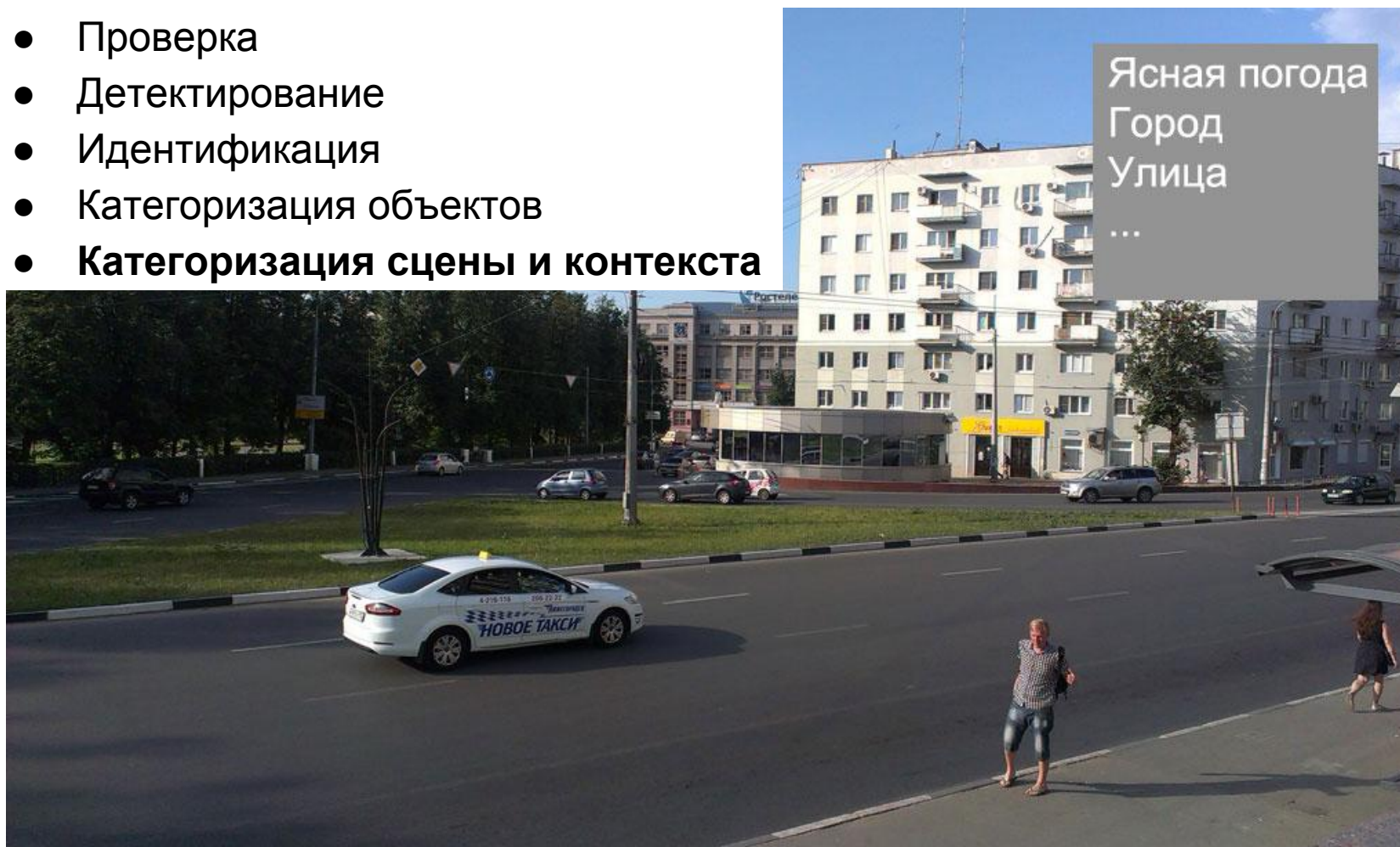
Распознавание объектов

- Проверка
- Детектирование
- Идентификация
- **Категоризация объектов**
- Категоризация сцены и контекста



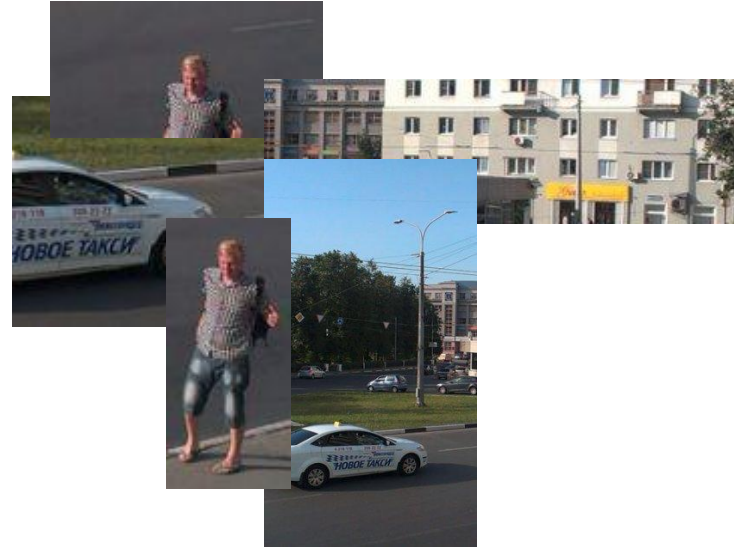
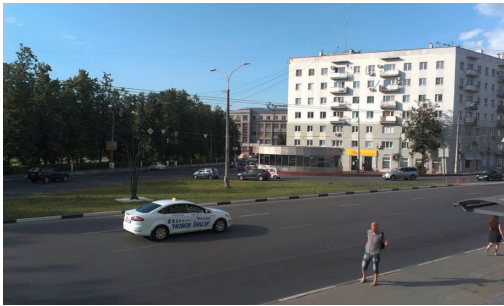
Распознавание объектов

- Проверка
- Детектирование
- Идентификация
- Категоризация объектов
- **Категоризация сцены и контекста**



Детектирование объектов

1. Генерация гипотез



2. Проверка гипотез

$\text{Пешеход}(\text{[image of taxi]}) = 0$

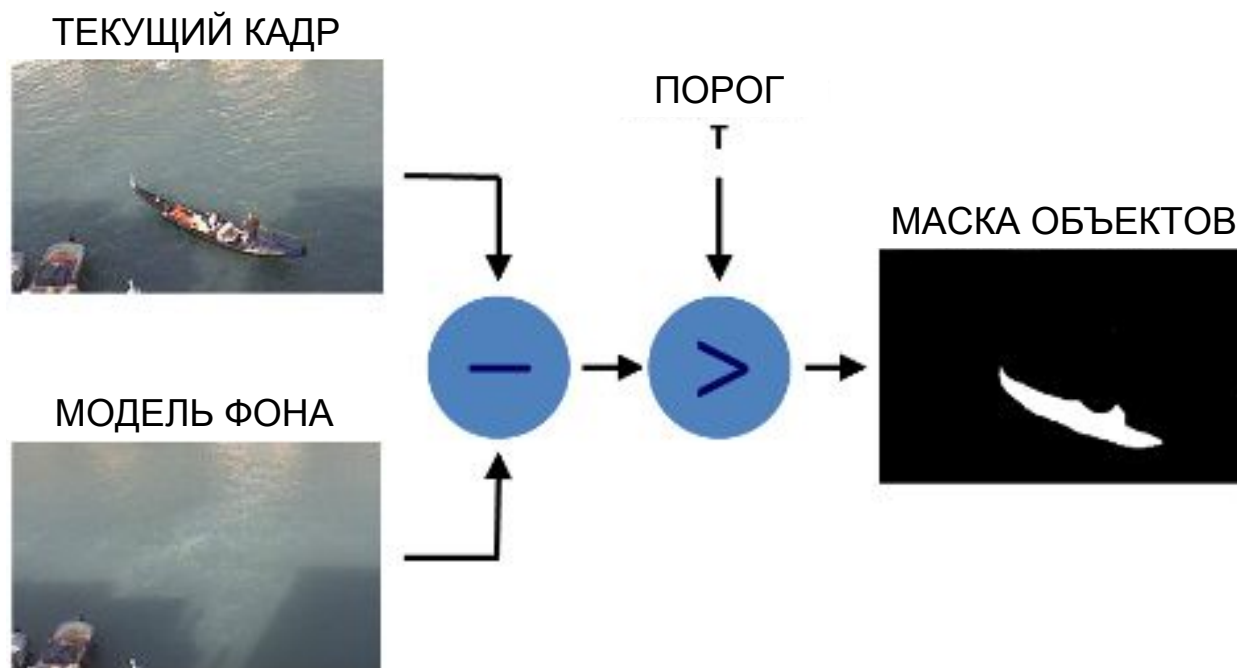


$\text{Пешеход}(\text{[image of pedestrian]}) = 1$



Вычитание фона

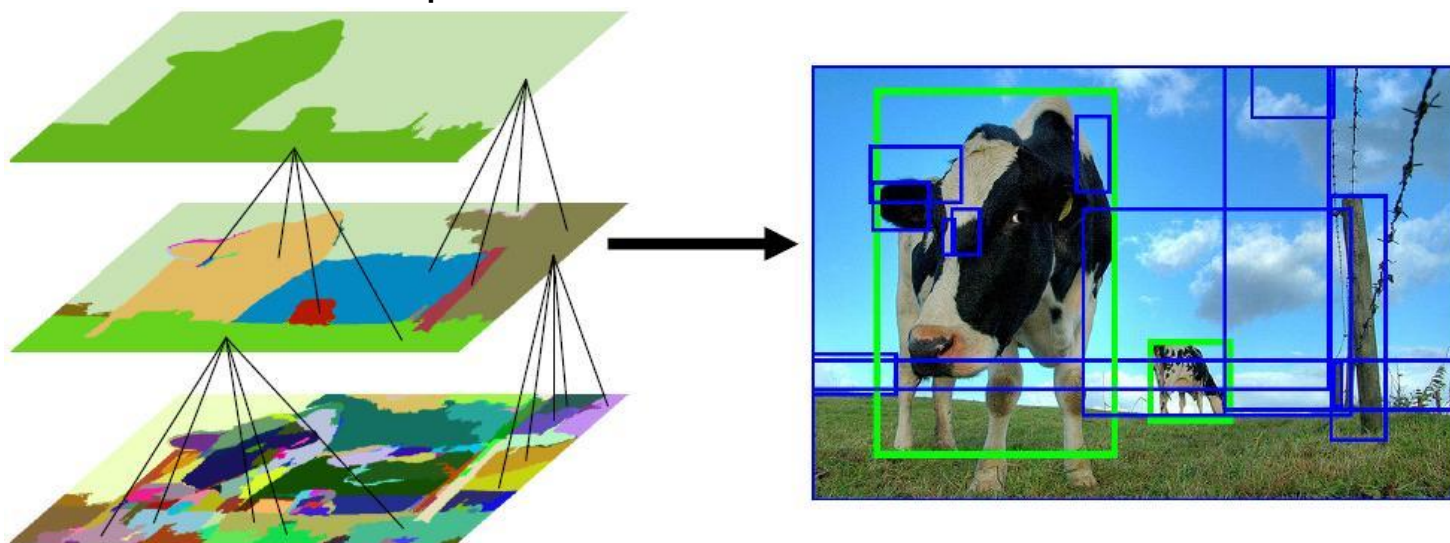
1. Задано изображение фона.
2. Вычтем фон из текущего кадра, отбросим пиксели с незначительной разницей.
3. Полученная маска содержит гипотезы.



Сегментация

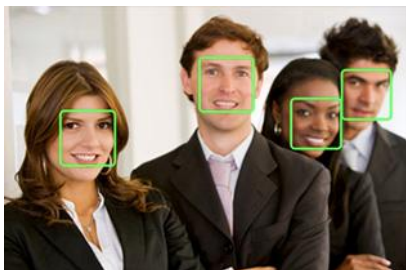
Сегментация - процесс разделения изображения на несколько сегментов.

1. Пусть каждый пиксель - отдельный сегмент.
2. Пока число сегментов > 2
 - a. Объединить похожие по цвету сегменты.
 - b. Объединить похожие по текстуре сегменты.
 - c. Объединить похожие по размеру сегменты.
 - d. Заполнить дырки в сегменте.

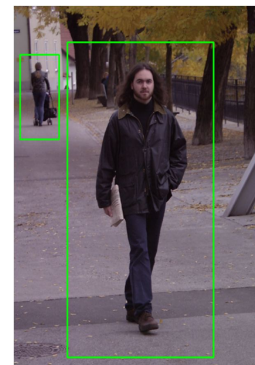


Скользящее окно

1. Окно фиксированного размера.



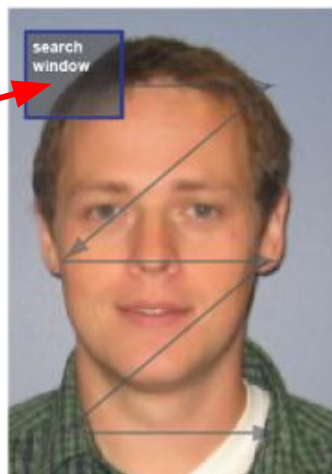
Лицо - квадрат, 24x24 пикселя



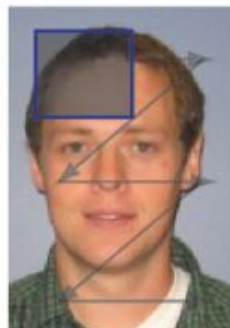
Пешеход - прямоугольник 64x128 пикселей

2. Полный перебор таких окон на исходном изображении и на всех масштабах, вплоть до размера скользящего окна.

Скользящее
окно



Масштаб 1x



Масштаб 0.75x



Масштаб 0.5x

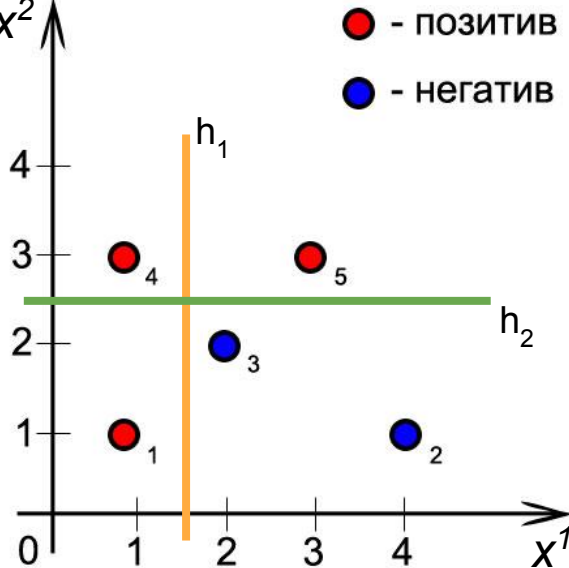
Классификация

$\mathcal{X} = (x^1, x^2, \dots, x^n) \in \mathbb{R}^n$ - вектор признаков объекта

$\mathcal{Y} \in \{0, 1\}$ - класс, 0 - фон, 1 - объект

Классификатор $h_j(x) = \begin{cases} 1, & p^j x^j < p^j \theta^j \\ 0, & \text{иначе} \end{cases}$ θ^j - порог
 p^j - знак

Пример: x^2



$$h_1: \theta^1 = 1.5, p^1 = 1$$

$$h_2: \theta^2 = 2.5, p^2 = -1$$

Если классификатор классифицировал фон как объект, то это ложное срабатывание (*false positive*).

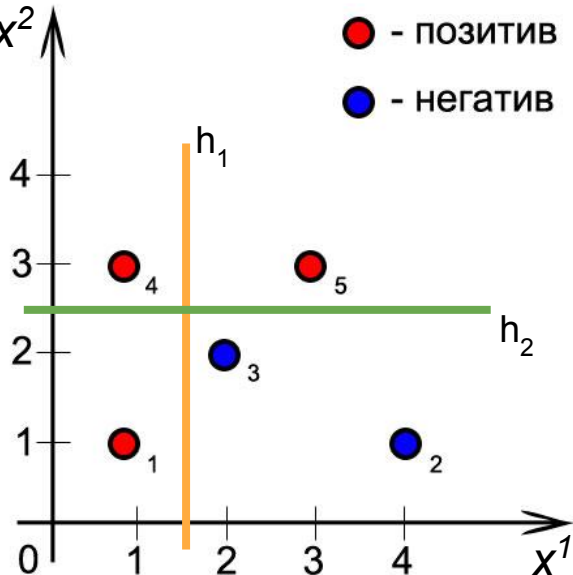
Классификация

$\mathcal{X} = (x^1, x^2, \dots, x^n) \in \mathbb{R}^n$ - вектор признаков объекта

$\mathcal{Y} \in \{0, 1\}$ - класс, 0 - фон, 1 - объект

Классификатор
$$h_j(x) = \begin{cases} 1, & p^j x^j < p^j \theta^j \\ 0, & \text{иначе} \end{cases} \quad \begin{array}{l} \theta^j - \text{порог} \\ p^j - \text{знак} \end{array}$$

Пример: x^2



$$h_1: \theta^1 = 1.5, p^1 = 1$$

$$h_2: \theta^2 = 2.5, p^2 = -1$$

Можно уменьшить ошибку классификации?

Сильный классификатор. Адаптивный бустинг (AdaBoost)

1. Задана обучающая выборка: $(\mathcal{X}_1, \mathcal{Y}_1), \dots, (\mathcal{X}_m, \mathcal{Y}_m)$, веса для каждого объекта $w_i = \frac{1}{m}, \forall i \in \overline{1, m}$

2. Для $t = 1 \dots T$

а. Для каждого j -го признака тренируется слабый классификатор h_j , и

вычисляется взвешенная ошибка классификации ε_j :
$$\varepsilon_j = \sum_{i=1}^m w_i * |h_j(x_i) - y_i|$$

б. Выбирается h_j с наименьшей ошибкой ε .

с. Веса обновляются:

$$w_i = w_i * \beta_t^{1-e_i}, \beta_t = \frac{\varepsilon}{1 - \varepsilon}$$

$e_i = 0$, если \mathcal{X}_i классифицировался корректно, 1 иначе.

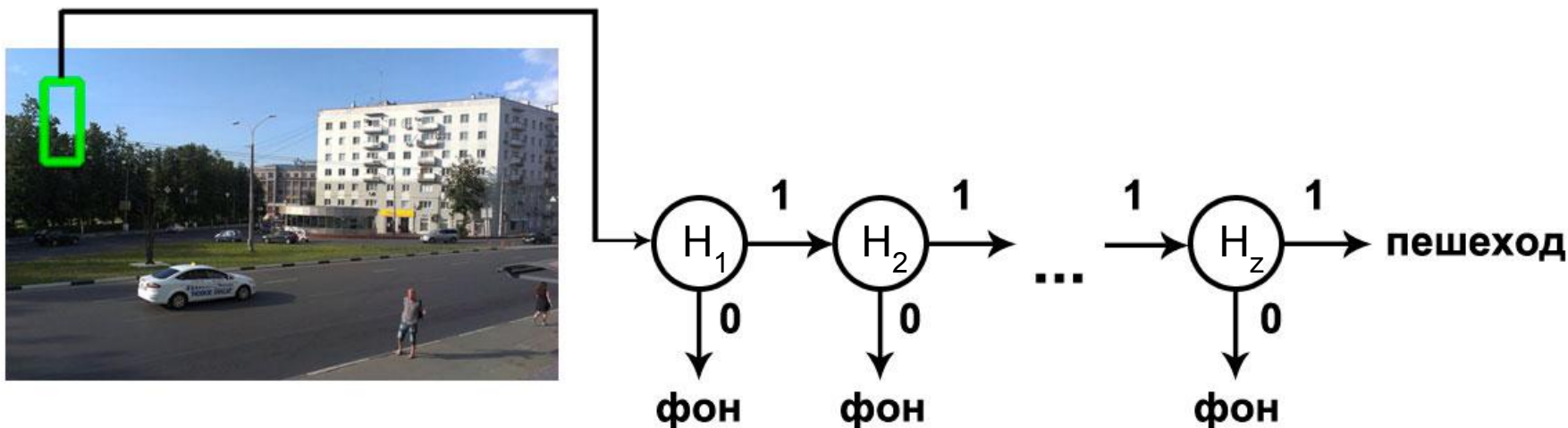
д. Веса нормализуются: $w_i = \frac{w_i}{\sum_{k=1}^m w_k}$

3. Сильный классификатор:

$$H(\mathcal{X}) = \begin{cases} 1, & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t, \alpha_t = \log \frac{1}{\beta_t} \\ 0, & \text{иначе} \end{cases}$$

Каскадный классификатор

- Вычислять сильный классификатор (сумму тысяч слабых) для каждого положения скользящего окна (десятки тысяч) долго.
- Разделим задачу на более простые подзадачи: будем тренировать серию (каскад) классификаторов, компенсирующих ошибки друг друга.



- Зададим желаемый объем ложных срабатываний (*false positive rate*, *FPR*) и степень правильно классифицированных объектов (*detection rate*, *DR*) для каждого H_i .

Тренировка каскада



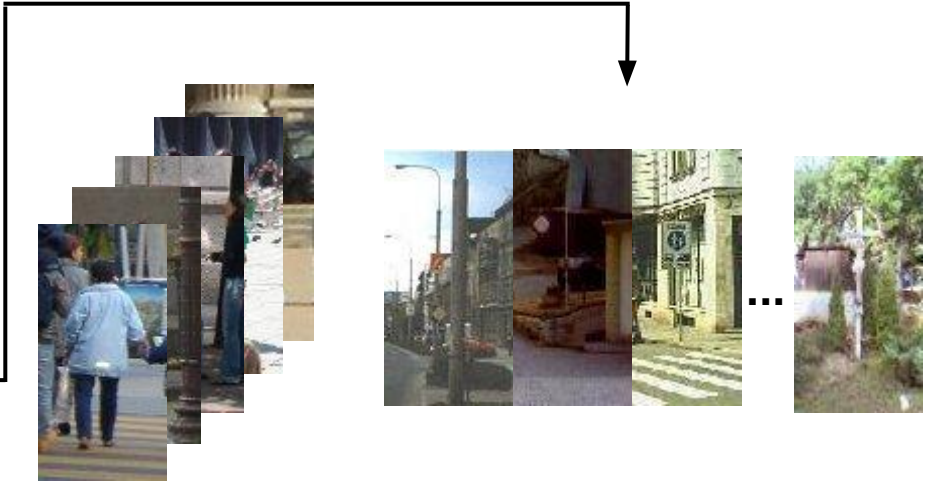
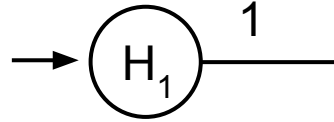
Исходная база изображений



Тренировочная выборка для первого классификатора: позитивы и негативы одинакового размера, негативы вырезаны случайно из исходной базы негативов.

1. Сформируем тренировочную базу, например для детектирования лиц выборка состоит из 10000 лиц и 10000 не лиц.
2. Тренируем сильный классификатор H_1 (*стадию*) до достижения заданных целей, обычно FPR 50%, DR 99.5%.

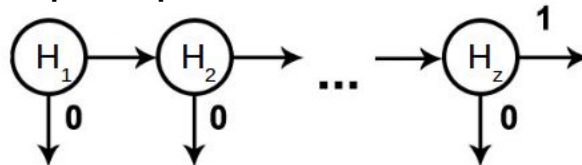
Тренировка каскада



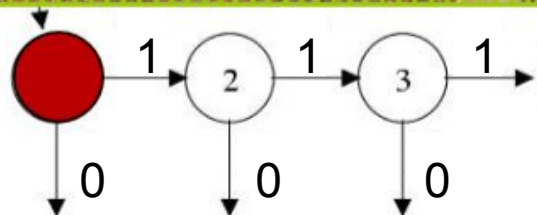
Исходная база негативов

Тренировочная выборка для второго классификатора.

3. Запустим H_1 на негативах, сформируем новую выборку негативов для тренировки (*bootstrapping*). Если число найденных негативов меньше порога, то тренировка завершена, иначе переходим на шаг 2 (тренируем следующую стадию каскада).
4. Составляем из натренированных сильных классификаторов каскадный классификатор.



Каскадный классификатор. Пример



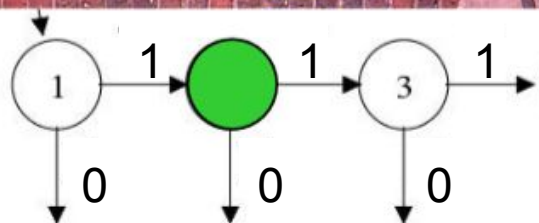
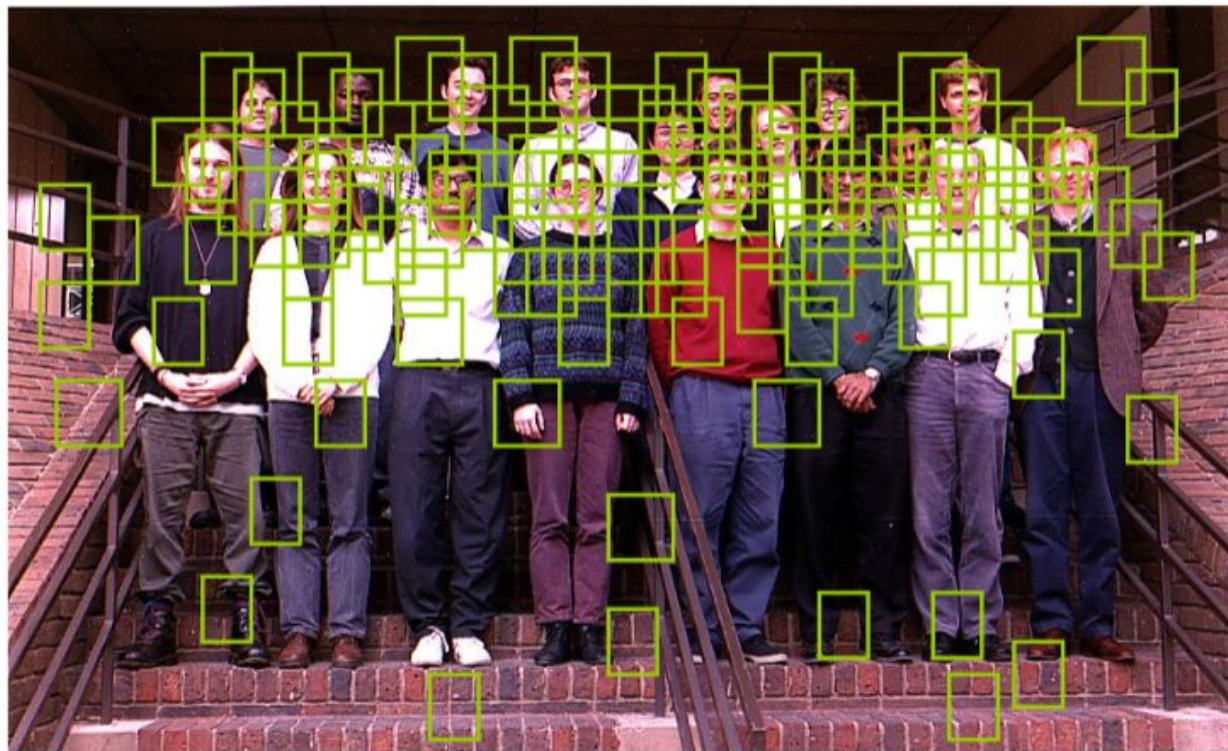
Лицо

8192

Гипотезы

Не лицо

Каскадный классификатор. Пример



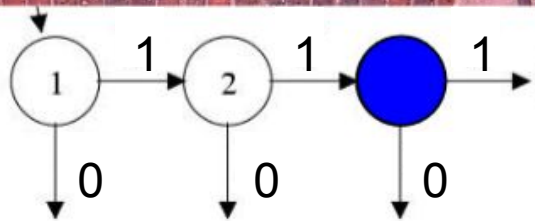
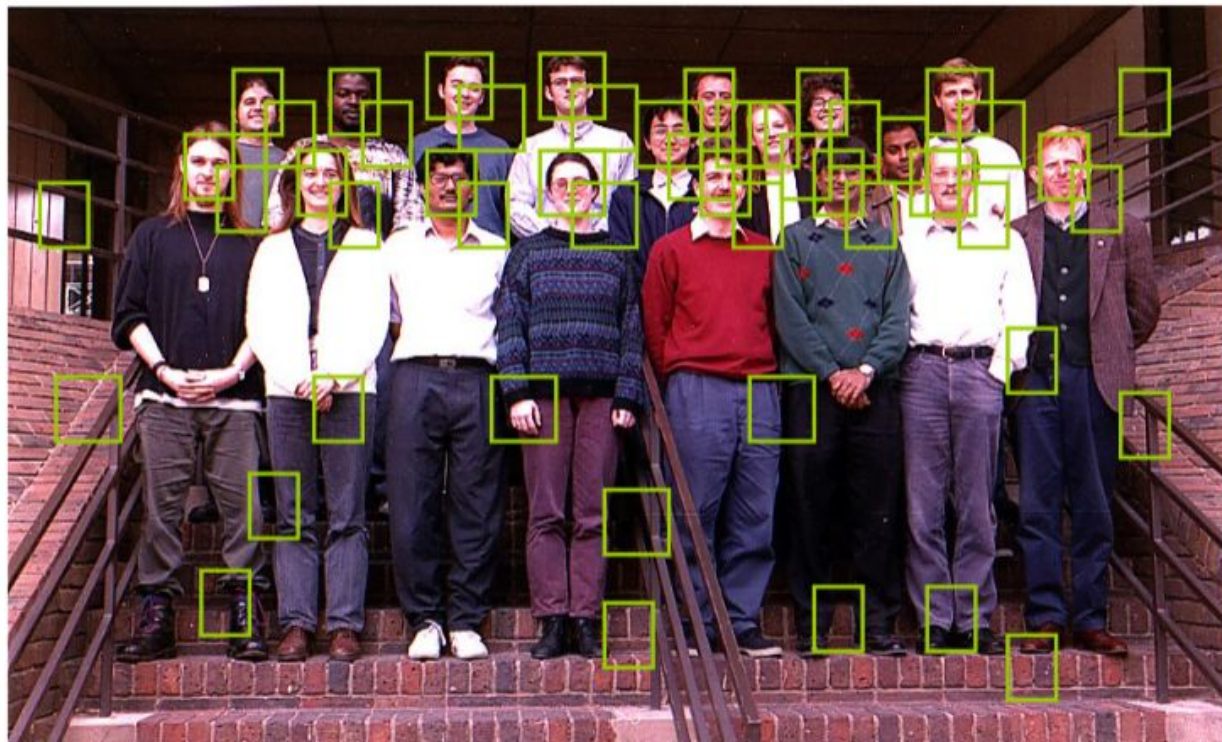
Лицо

188

Гипотез

Не лицо

Каскадный классификатор. Пример



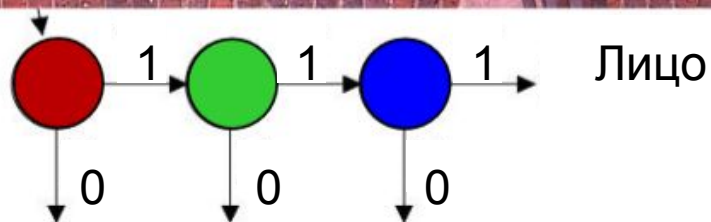
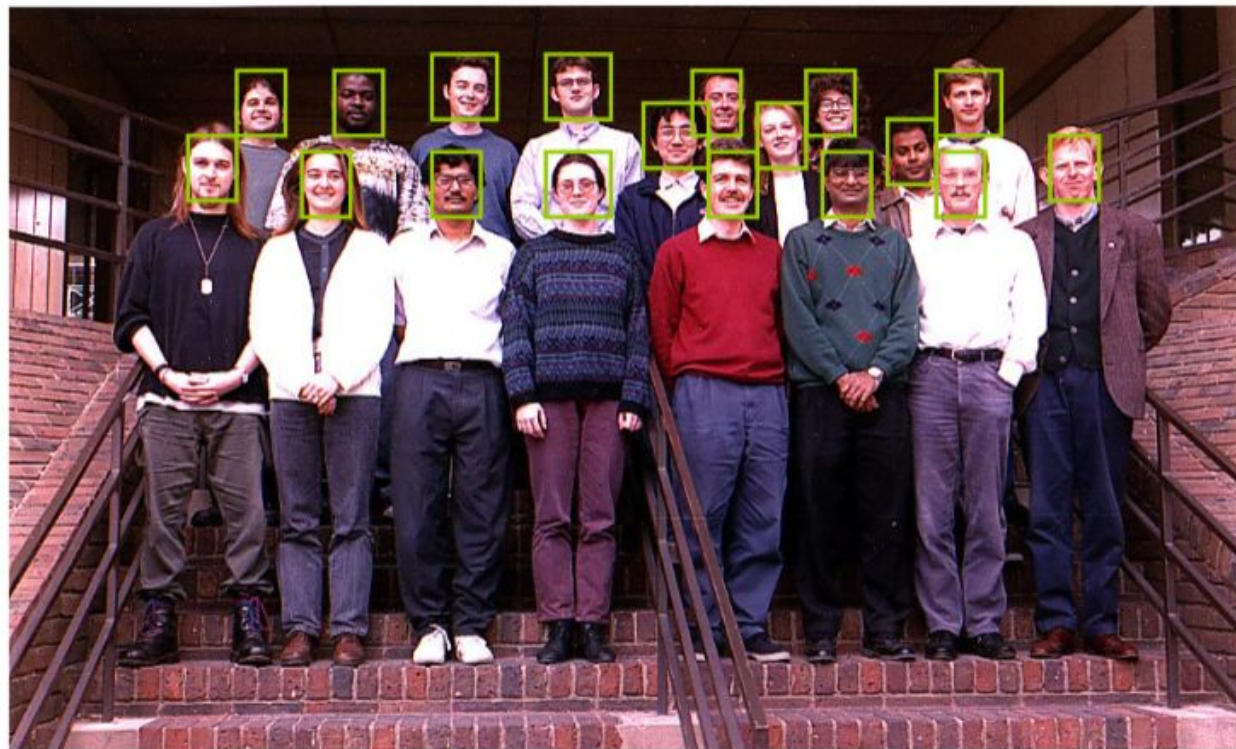
Лицо

48

Гипотез

Не лицо

Каскадный классификатор. Пример



Не лицо

18 лиц

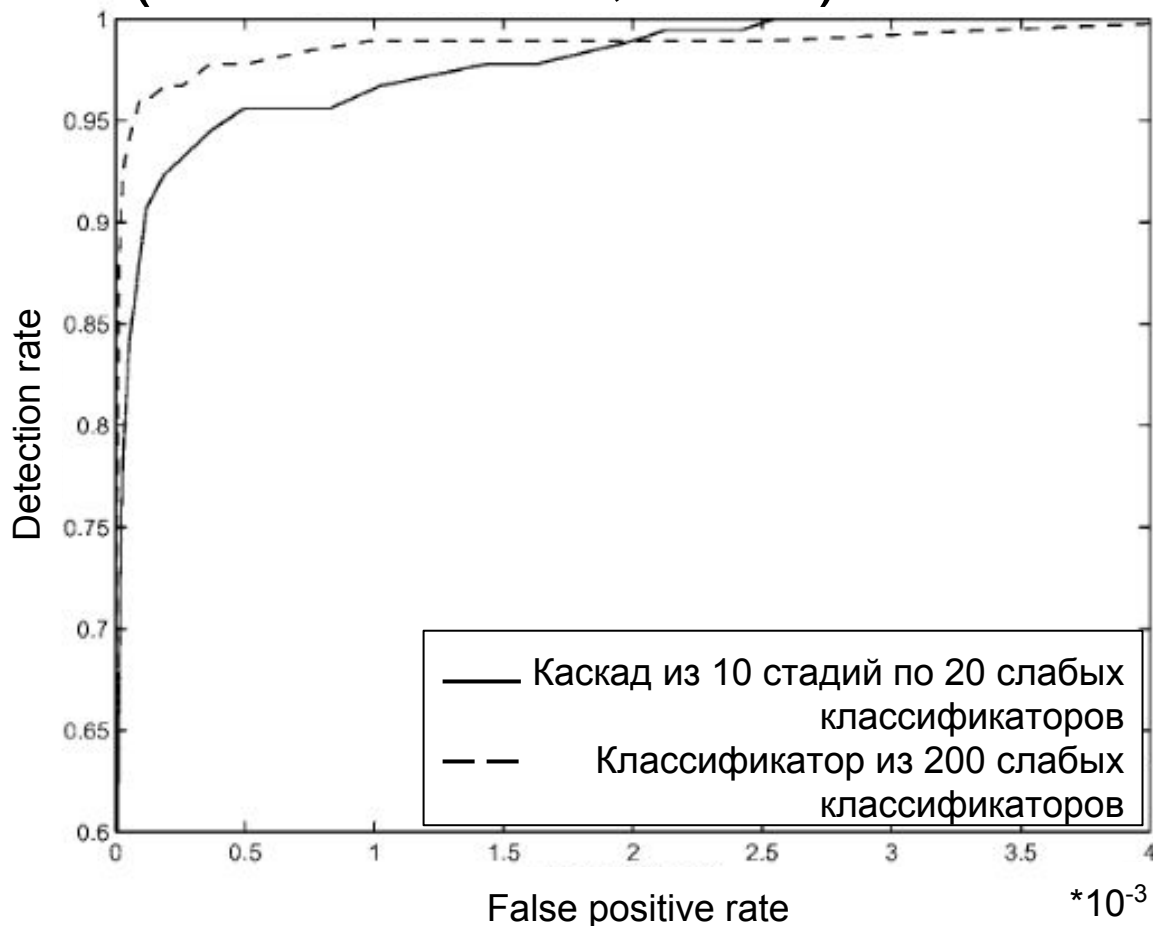
Сравнение каскадного классификатора с монолитным (Viola & Jones, 2001)



Paul Viola,
Microsoft
research

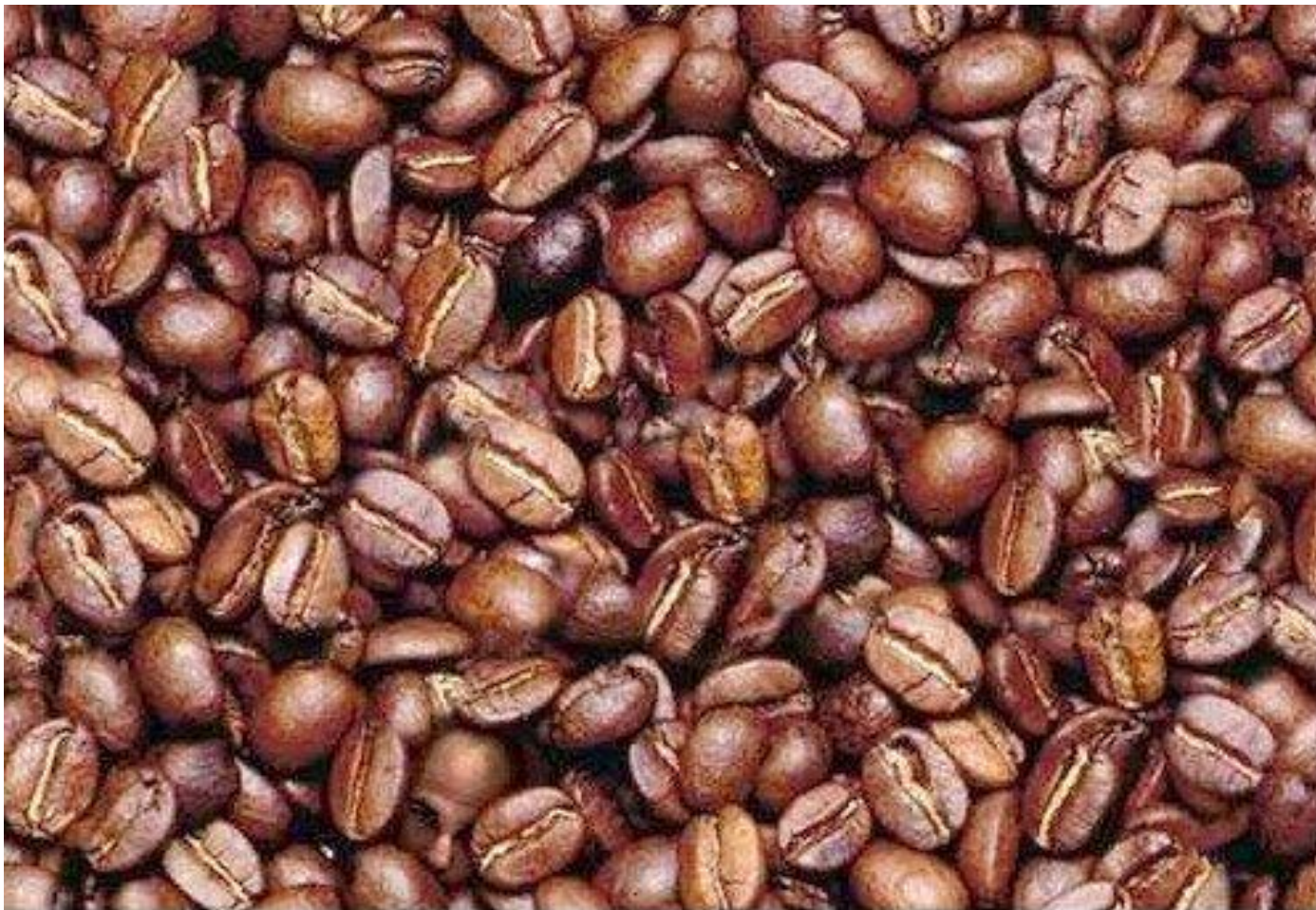


Michael Jones,
Merl



При сравнимом качестве, каскад в ~10 раз быстрее.

Сможете найти лицо среди кофейных зерен?



Простые признаки

Как описать объект?

- Значения пикселей



$$\mathcal{X} = (50, 54, 52, 54, \dots, 186)$$

Пешеход, 16x32 пикселей

Вектор признаков, 512x1 пикселей

- Градиент



$$\mathcal{X} = (0, 0, 38, 0, \dots, 0)$$

Пешеход,
16x32 пикселей

Градиент

Вектор признаков, 512x1 пикселей

Простые признаки

Как описать объект?

- Значения пикселей



$\mathcal{X} = (50, 54, 52, 54, \dots, 186)$

Пешеход, 16x32 пикселей

Вектор признаков, 512x1 пикселей

- Градиент



$\mathcal{X} = (0, 0, 38, 0, \dots, 0)$

Пешеход,
16x32 пикселей

Градиент

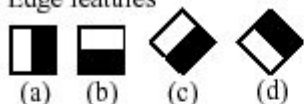
Вектор признаков, 512x1 пикселей

Чем один признак (*feature*) может быть лучше другого?

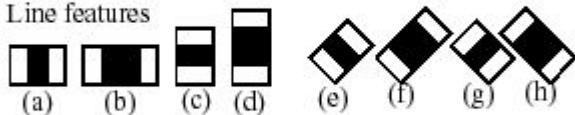
Haar-like признаки

Требование: признаки должны быстро вычисляться в любой точке изображения и при любом масштабе.

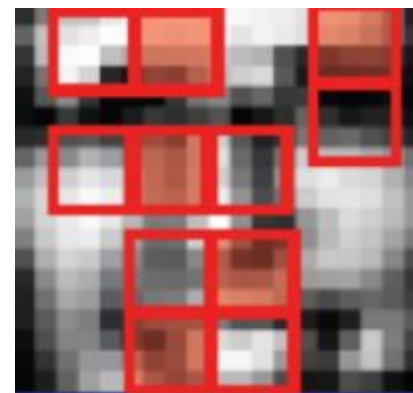
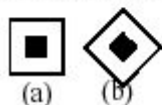
1. Edge features



2. Line features



3. Center-surround features



Для быстрого вычисления используется интегральное изображение:

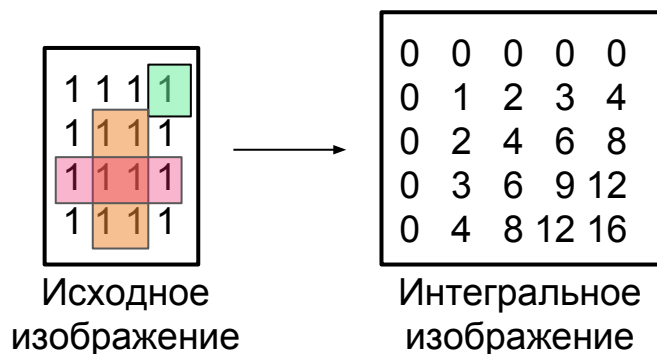
$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'),$$

A	B	
	1	2
C	D	4
	3	

Сумма внутри D:
 $4 + 1 - (2 + 3).$

Интегральное изображение

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'),$$



Сумма в:

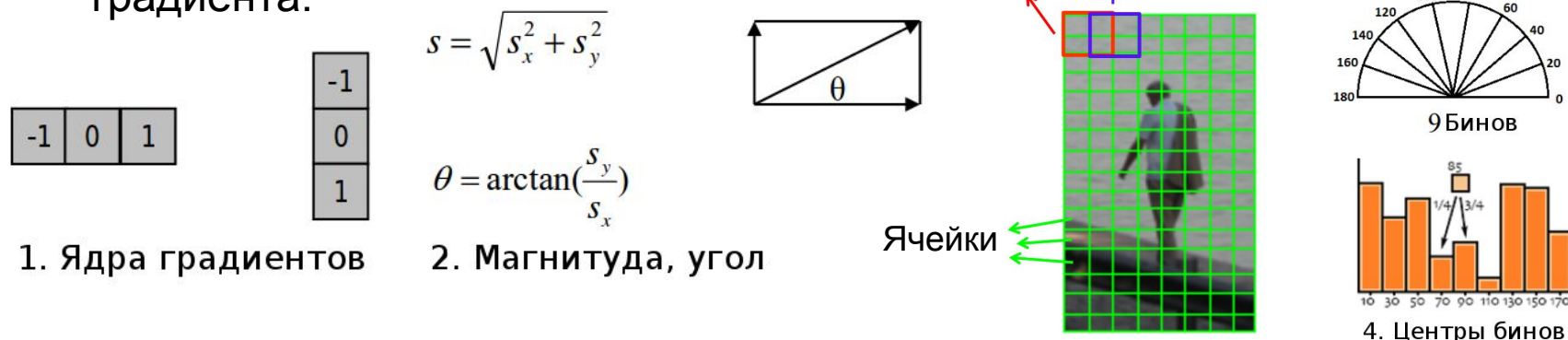
$= (4 + 0) - (3 + 0) = 1$

$= (12 + 0) - (0 + 8) = 4$

$= (12 + 1) - (4 + 3) = 6$

Гистограмма градиентов (HoG)

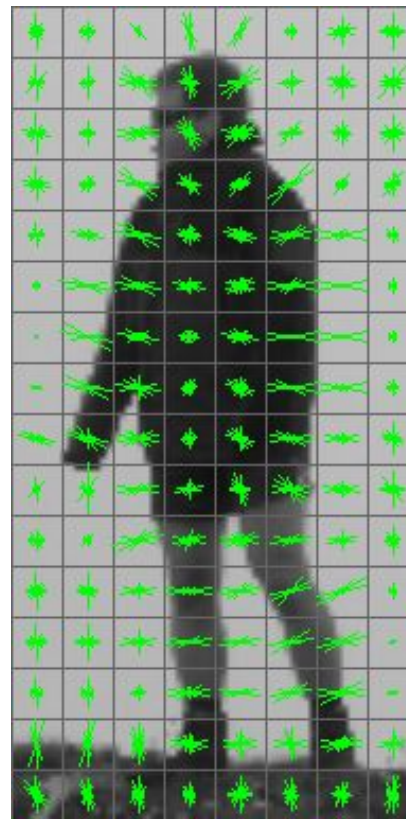
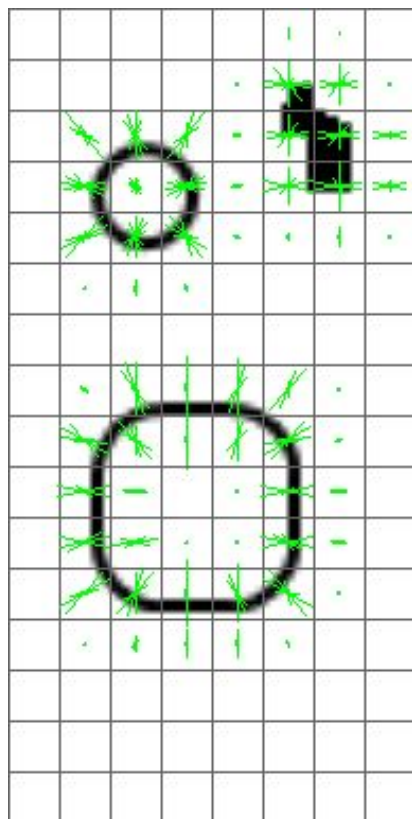
1. Вычислить горизонтальный, вертикальный градиенты в каждом пикселе изображения.
2. Найти магнитуду, ориентацию градиента.
3. Разделить изображение (64x128) на блоки 16x16 пикселей (всего $7 * 15 = 105$ блоков), по 2x2 ячейки, каждая ячейка 8x8 пикселей.
4. Для каждой ячейки каждого блока построить гистограмму ориентации градиента.



5. Поблочная нормализация.
6. Конкатенация всех гистограмм формирует вектор признаков изображения.

$$\left[\begin{array}{ccccccc} \text{H1} & \text{H2} & \text{H3} & \dots & \text{Hn} & \text{Hm} & \text{Hk} \end{array} \right] \in \mathbb{R}^{3780}$$

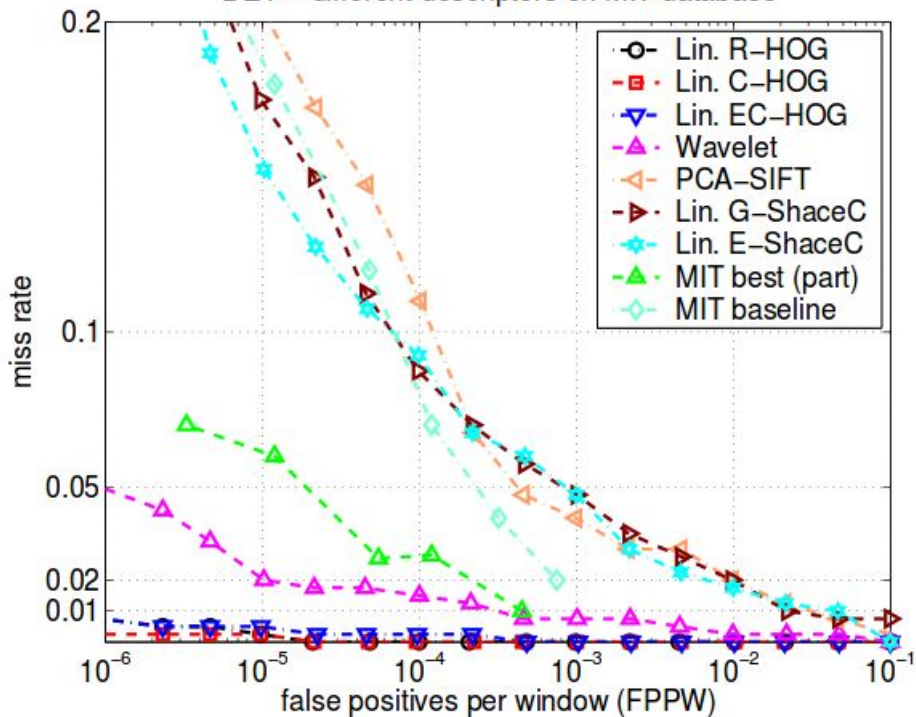
Визуализация HoG



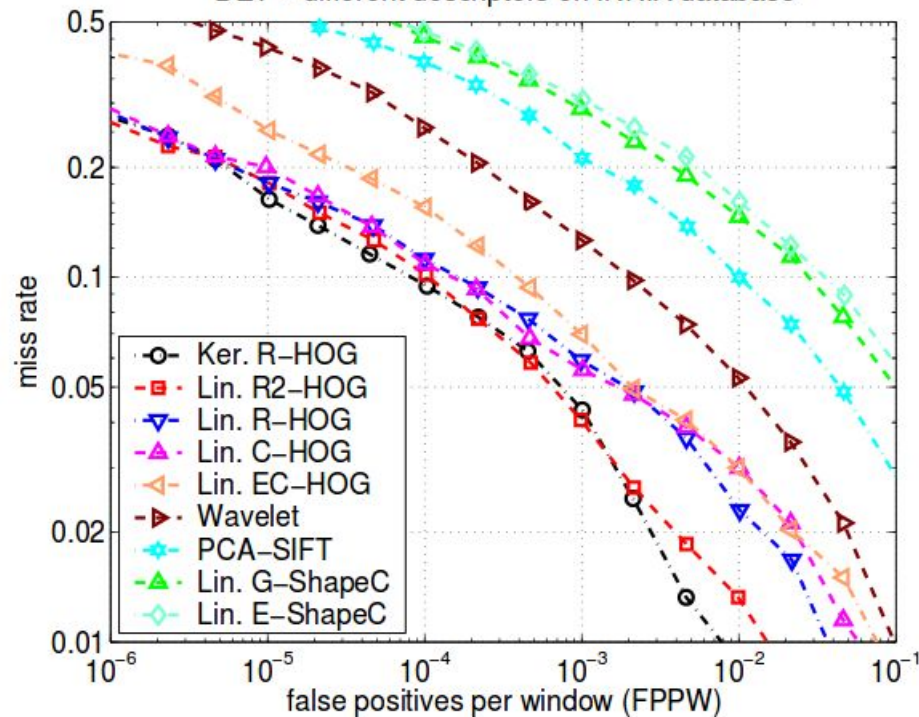
Зеленым отложено насколько сильным был градиент в конкретной ячейке в данном направлении.

HoG для детектирования пешеходов (Dalal & Triggs, 2005)

DET – different descriptors on MIT database



DET – different descriptors on INRIA database



Navneet Dalal and Bill Triggs, INRIA, France



Вопросы

