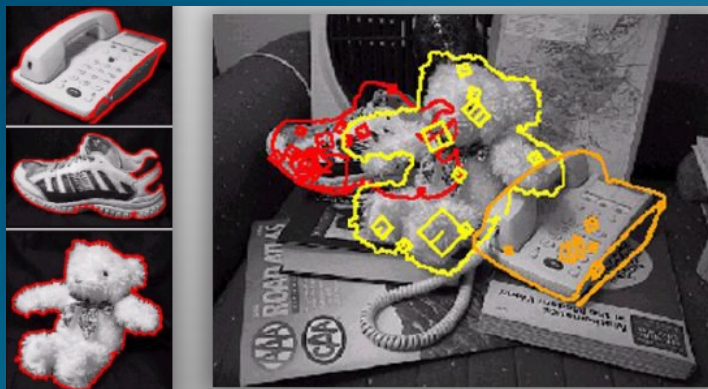


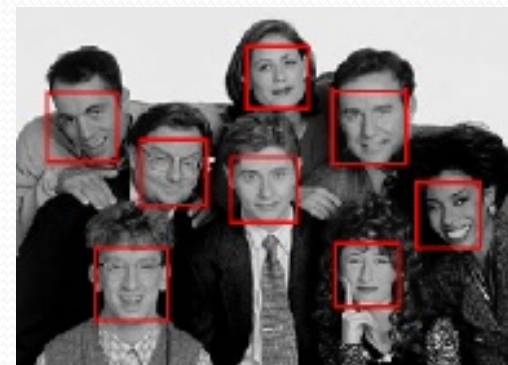
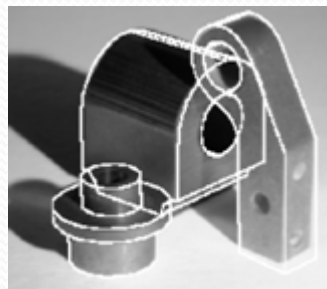
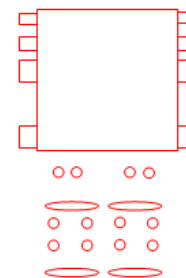
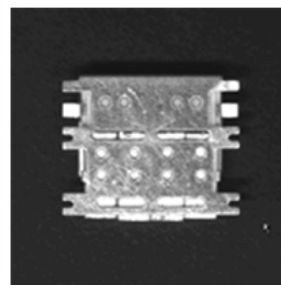
ПОИСК ОБЪЕКТОВ НА ИЗОБРАЖЕНИИ



Александр Бовырин

Постановка задачи

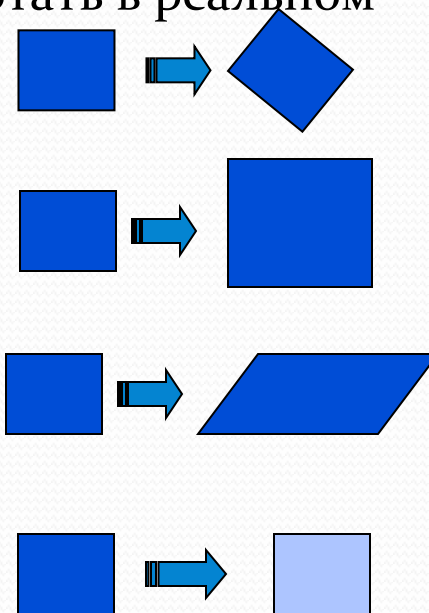
- Задача состоит в обнаружении на изображении заданного объекта(ов).
- Применение:
 - поиск лиц,
 - поиск пешеходов,
 - поиск вещей,
 - сортировка на конвейере,
 - поиск в цифровых базах данных (запрос по шаблону), ...
- Вариации задачи:
 - мало примеров для обучения
 - много примеров для обучения



Постановка задачи

Требования к системе обнаружения

- Ошибки первого рода (ложное детектирование, false positive) должны быть редки, ошибки второго рода (пропуск объекта, false negative) допускаются ещё реже.
- Ограниченное время поиска. Система должна работать в реальном времени (25FPS на VGA)
- Алгоритм поиска может быть инвариантным к
 - Смещению
 - Масштабированию
 - Повороту
 - Аффинным преобразованиям
 - Перспективным преобразованиям
 - Частичному перекрытию объекта
 - Изменению цветовых характеристик
 - Неравномерному освещению
 - Значительному уровню шума
 - 3D повороту объекта



Методы обнаружения

Шаг 1.

Генерация гипотез
(вопрос: “где?”)

Перебор всех возможных
гипотез
(метод скользящего окна)

Оценка позы всего объекта по
позе характерных частей
объекта

Стохастический поиск

Шаг 2.

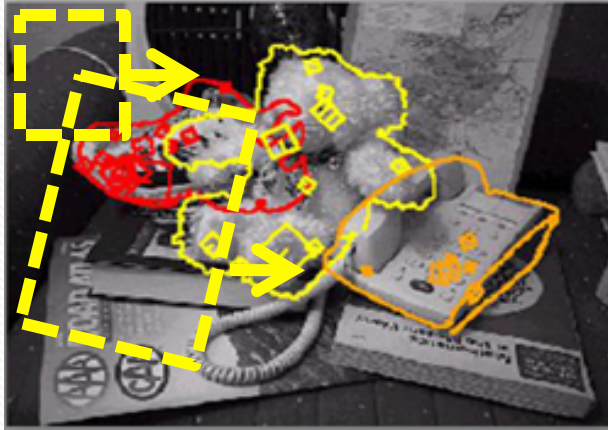
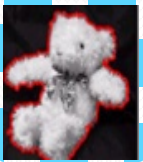
Проверка гипотезы
(вопрос: “что?”)

Проверка корреляции
изображения шаблона и
гипотезы

Проверка согласованности
частных гипотез
(поиск клики, обобщённое
преобразование Хафа)

Машинное обучение
(adaboost, Nnets, SVM,..)

Метод скользящего окна. Поиск прямым сравнением.



Методом скользящего окна проверяются все возможные гипотезы позиции, масштаба и поворота объекта.

Недостатки: низкая скорость, дискретность шага, трудность выбора порога для срабатывания детектора в зависимости от условий освещённости и контрастности объектов.

Эмпирические методы поиска.

- Обнаружение номеров автомобилей



- Возможные эмпирические правила:
 - 1) находится вытянутый светлый сегмент характерного размера.
 - 2) проверяется наличие внутренних контуров (букв), проверяется их форма.

Быстрое детектирование лиц.

- Нужно найти позицию и масштаб фронтальных лиц на изображении.



- Применение:
Распознавание лиц;
Видеонаблюдение;
Digital Signage;
Получение и обработка фотографий.

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola
viola@merl.com
Mitsubishi Electric Research Labs
201 Broadway, 8th FL
Cambridge, MA 02139

Michael Jones
mjones@crl.dec.com
Compaq CRL
One Cambridge Center
Cambridge, MA 02142

Детектирование лиц

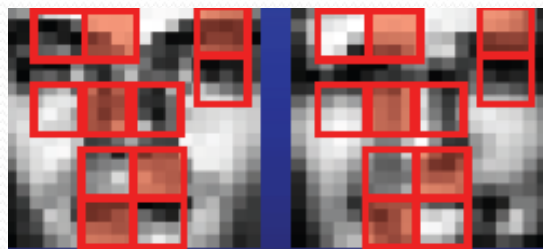
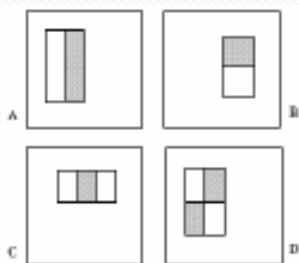
Генерация гипотез

- Цикл по масштабу и по координатам гипотезы
- Фильтруются заведомо ложные гипотезы:
 - Проверяется наличие рёбер/градиентов (они должны присутствовать)
 - Средняя интенсивность в лице не может быть очень большой или очень малой.
- По оставшимся вариантам строится вектор признаков и происходит классификация методом adaboost.



Вектор признаков

- Для классификации используются Хагг вейвлеты (требование к признакам: должны быстро вычисляться в любом положении и любом масштабе).



- Для быстрого вычисления используется интегральное

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

A	B	
	1	2
C	D	
	3	4

The sum within D
 $4 + 1 - (2 + 3).$

“Слабый” классификатор

- Всего признаков > 100000 .
- По любому признаку можно построить “слабый” классификатор

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

где x – вектор размерности 24×24 ,
соответствующий текущему куску изображения.

f_j – j -ый признак

θ_j – порог

p_j – знак

Выбор признаков и классификация

- Дано обучающее мн-во изображений $(x_1, y_1), \dots, (x_n, y_n)$ где $y_i = 0, 1$.
0 – не лица, 1 – изображения лица.
- Инициализируется массив весов $w_{0,i} = \{1/(2m) \text{ or } 1/(2l)\}$ для каждого изображения i , где m и l число негативных или позитивных примеров
- For $t = 1 \dots T$

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$

- 1) Веса w_t нормализуются (пл-ть вер-ти)
- 2) Для каждого признака j тренируется 'слабый' классификатор h_j и вычисляется ошибка классификации ε_j с весом w_t . $\varepsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
- 3) Выбирается классификатор h_j с наименьшей ошибкой.
- 4) Веса обновляются:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i} \quad \beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$$

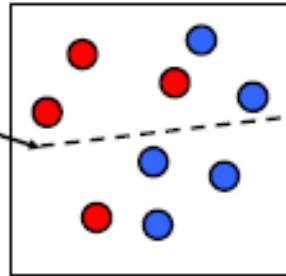
где $e_i = 0$ если x_i классифицировалось корректно, 1 в противном случае

- В итоге получим
'сильный' классификатор:

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

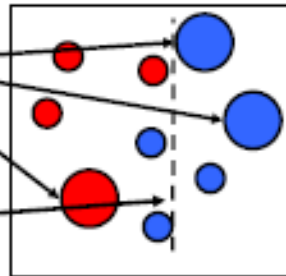
AdaBoost является адаптивным в том смысле, что каждый следующий классификатор строится по объектам неверно классифицированным предыдущими классификаторами

Weak
Classifier 1

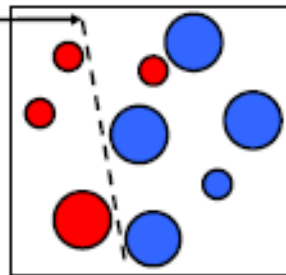


Weights
Increased

Weak
Classifier 2



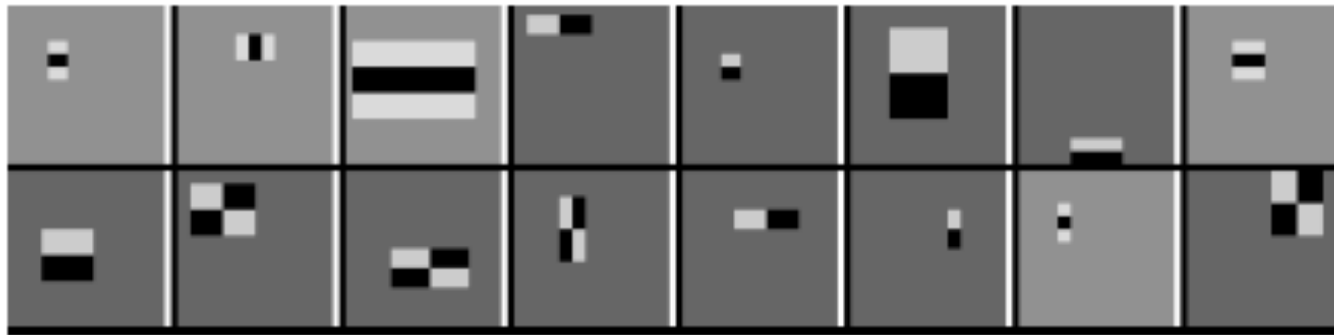
Weak
classifier 3



Final classifier is
linear combination of
weak classifiers

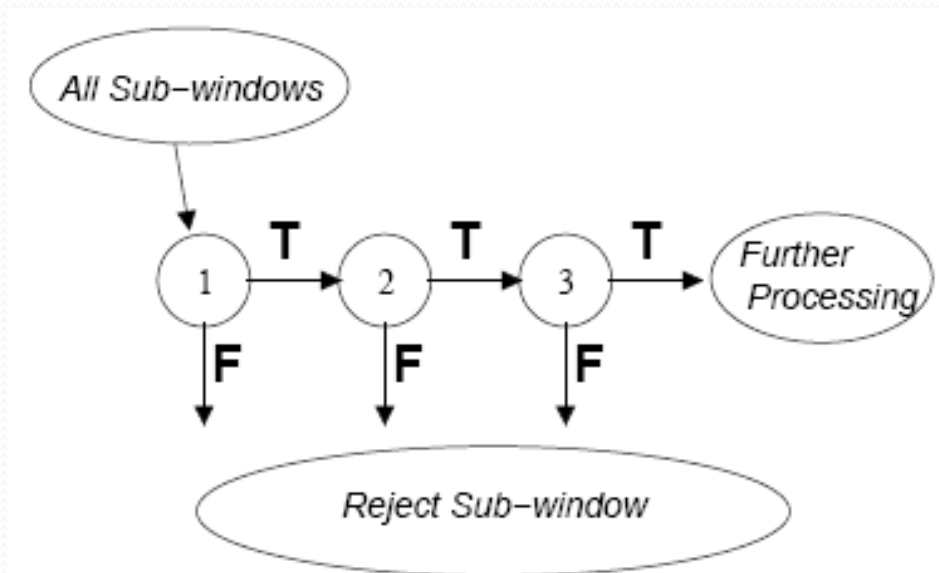
Детектирование лиц

- Ошибка классификации уменьшается с ростом T .



Использование каскада классификаторов

- Подсчёт большого числа признаков в сильном классификаторе и их проверка может быть медленной.
- Декомпозируем задачу на более простые подзадачи. Построим серию (каскад) сильных классификаторов, которые решают задачу нахождения объекта при меньшем требовании на количество ложных срабатываний и учатся на ошибках друг друга.



Детектирование лиц

Тренировка каскада.

- Для каждого классификатора в каскаде используется порядка 10000 примеров лиц и столько же не лиц.
- Тренировка останавливается когда достигается ошибка по частоте ложных срабатываний порядка 50% и правильной классификации лиц 99.5%.
- Для следующего сильного классификатора набираются другие 10000 негативных примеров, на которых не правильно работают предыдущие 'сильные' классификаторы.



Детектирование лиц

- Тренировка каскада

- Example: 32 stage cascade classifier

- ◆ 2-feature classifier in the first stage → rejecting 60% non-faces while detecting 100% faces
- ◆ 5-feature classifier in the second stage → rejecting 80% non-faces while detecting 100 % faces
- ◆ 20-feature classifier in stages 3, 4, and 5
- ◆ 50-feature classifier in stages 6 and 7
- ◆ 100-feature classifier in stages 8 to 12
- ◆ 200-feature classifier in stage 13 to 32

Детектирование лиц

- Каждая гипотеза о положении и масштабе лица оценивается каскадом классификаторов.
- В районе лица получается 'кластер' гипотез положения и масштаба лица.
- Формируется среднее положение искомого объекта.
- Число элементов в кластере отражает меру уверенности в этой гипотезе.



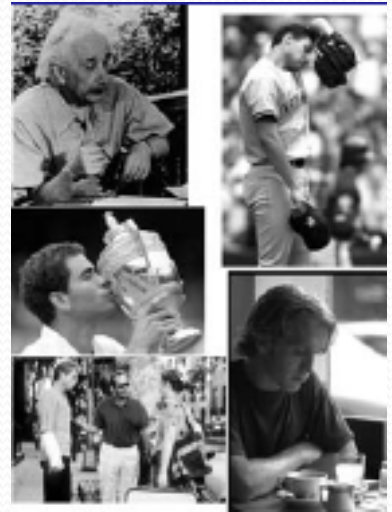
Детектирование лиц.

Проблемы

- Ищет только фронтальные лица.



- Метод требователен к освещенности.



- Производительность (640x480 меньше 25FPS).

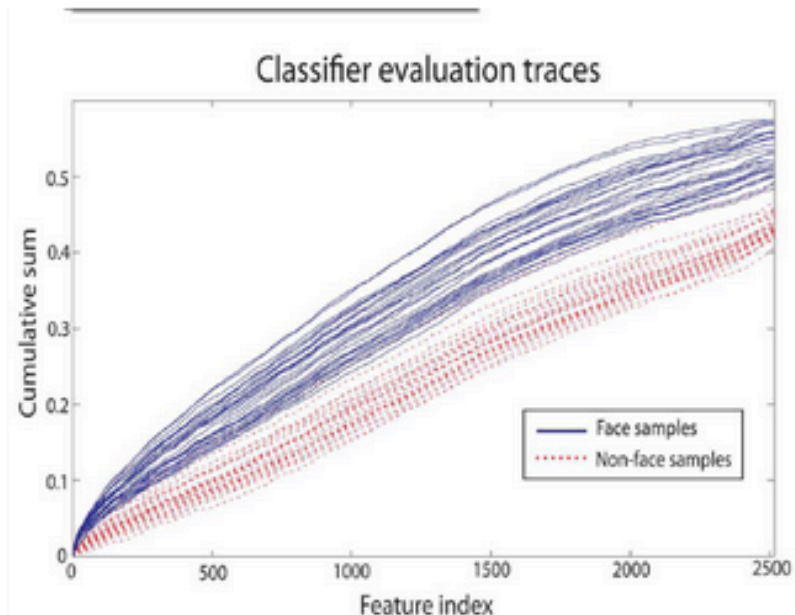
Soft Cascade

- Each new weak classifier forms a new stage

$$H(x) = \sum_{t=1, \dots, T} c_t(x)$$

, where $c_t(x) = \alpha_t h_t(x)$ - weak classifiers with weights,

- Check threshold after each weak classifier evaluation.



Boosting chains

For each pair of integers (i, j) so that the rectangle $R_{ij} = \{\text{left} = x + j \cdot dx, \text{top} = y + i \cdot dy, \text{width} = W_0, \text{height} = H_0\}$ fits entirely in the current image layer, $dx = dy = \{2 \text{ if scale} < 2, 1 \text{ otherwise}\}$, run the boost chain classifier:

$S_{ij} := 0$

if $\text{standard_deviation}(R_{ij}) < T_{\text{stddev}}$ then skip the rectangle, move to the next (i, j) pair.

(*) while $\text{stage} < \text{number_of_stages}$ do

 while $k < \text{number_of_wc_stage}$ do

$S_{ij} := S_{ij} + \text{WeakClassifier}_{\text{stage}, k}(R_{ij})$

 if $S_{ij} < T_{\text{stage}}$ then exit the loop (*)

if the loop (*) was interrupted, skip the rectangle, otherwise, store the scaled rectangle and its score: $\text{FCL} := \text{FCL} \cup (R_{ij} \cdot \text{scale}, S_{ij})$, where $R \cdot \text{scale}$ means that the scale rectangle top-left position and the size are multiplied by "scale" to represent the object candidate in the original, unscaled, image coordinates.

Детектирование пешеходов с помощью гистограммы ориентаций градиентов (HoG)

People detection is required for:

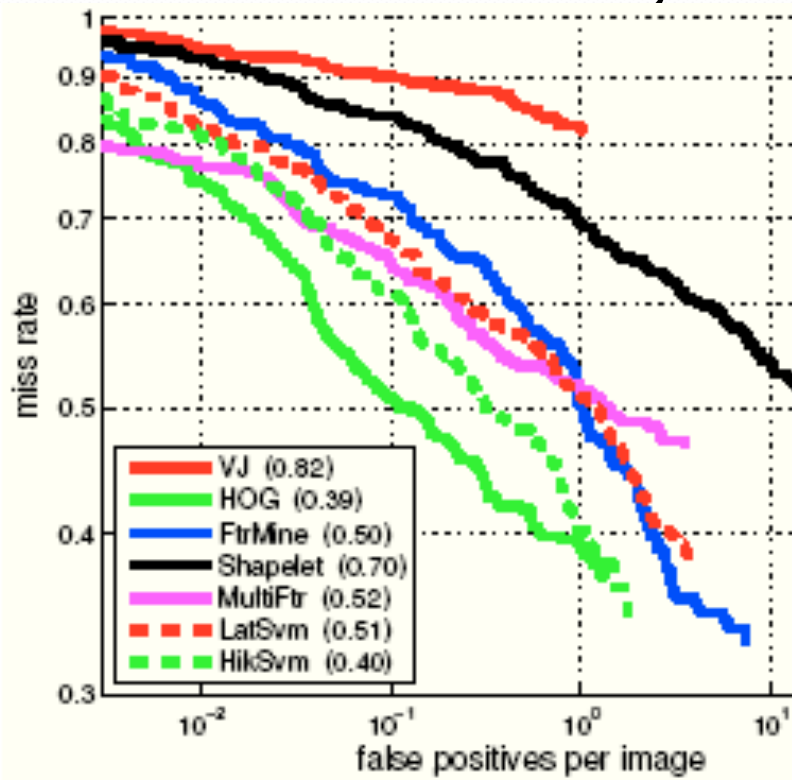
- Driver-assistant systems.
- Intelligent video surveillance systems.



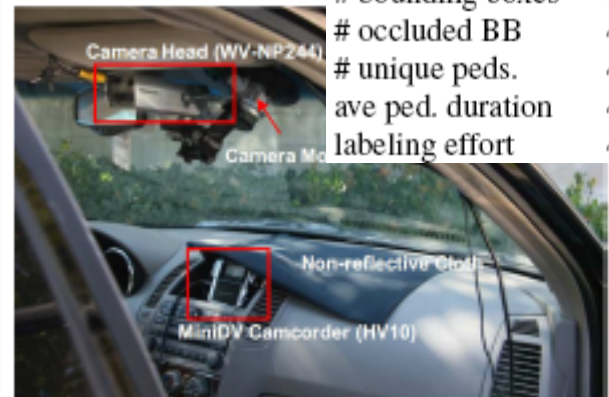
Pedestrian detection based on Histograms of Oriented Gradients

- Pedestrian Detection: A Benchmark (CVPR' 2009)

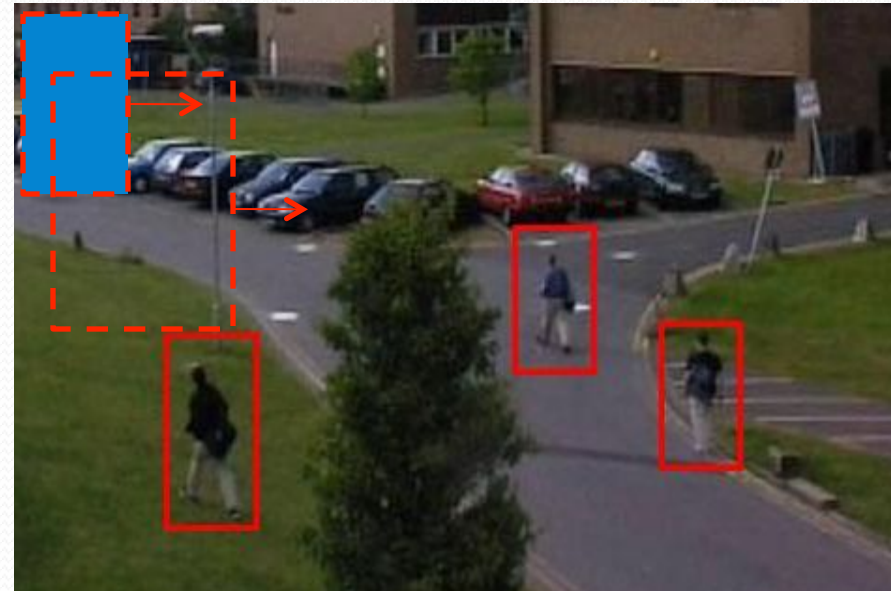
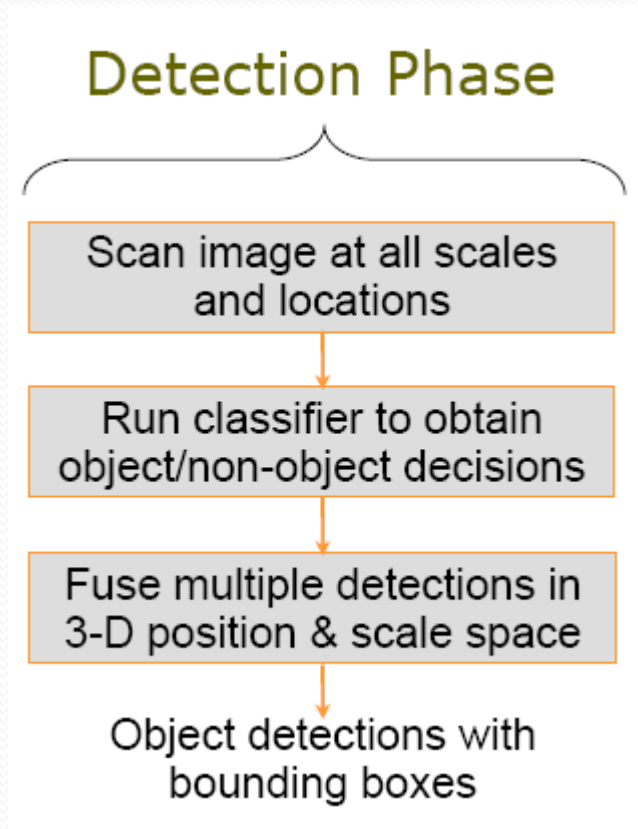
HOG tends to outperform the other methods surveyed.



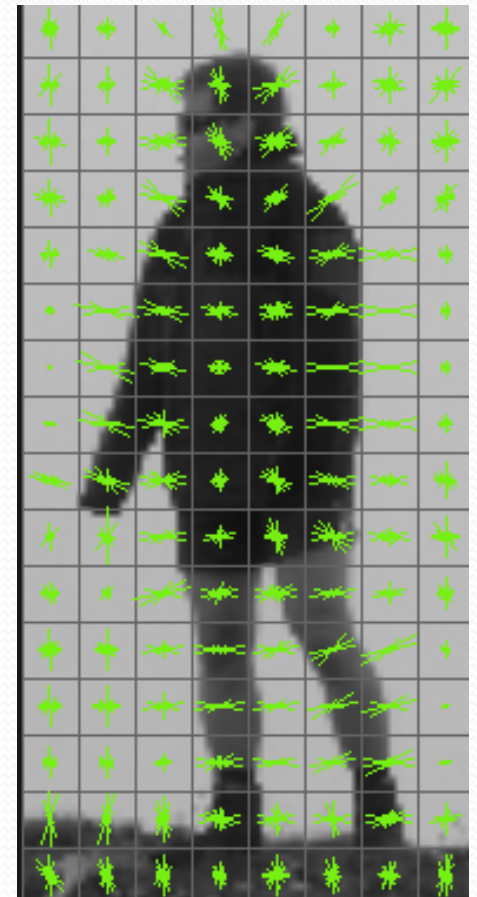
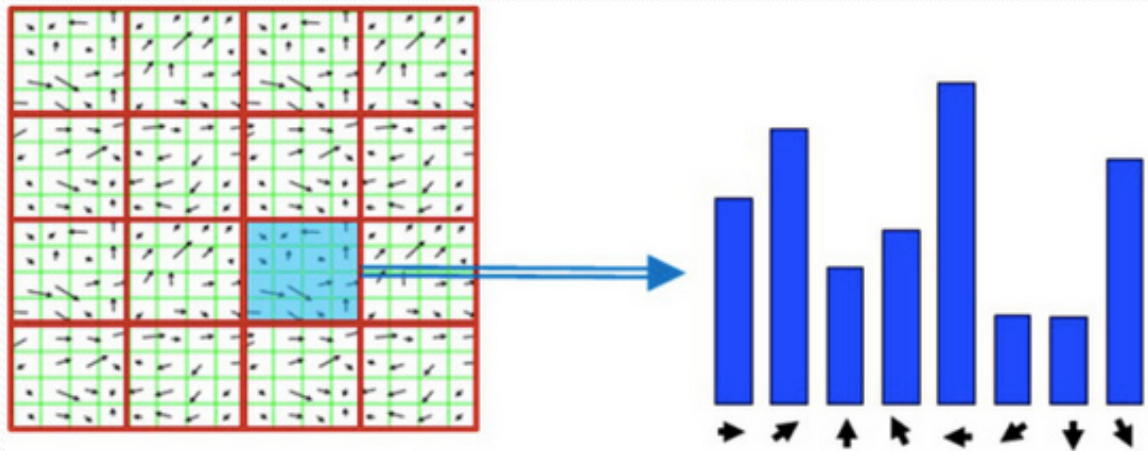
total frames	~1000K
labeled frames	~250K
frames w peds.	~132K
# bounding boxes	~350K
# occluded BB	~126K
# unique peds.	~2300
ave ped. duration	~5s
labeling effort	~400h



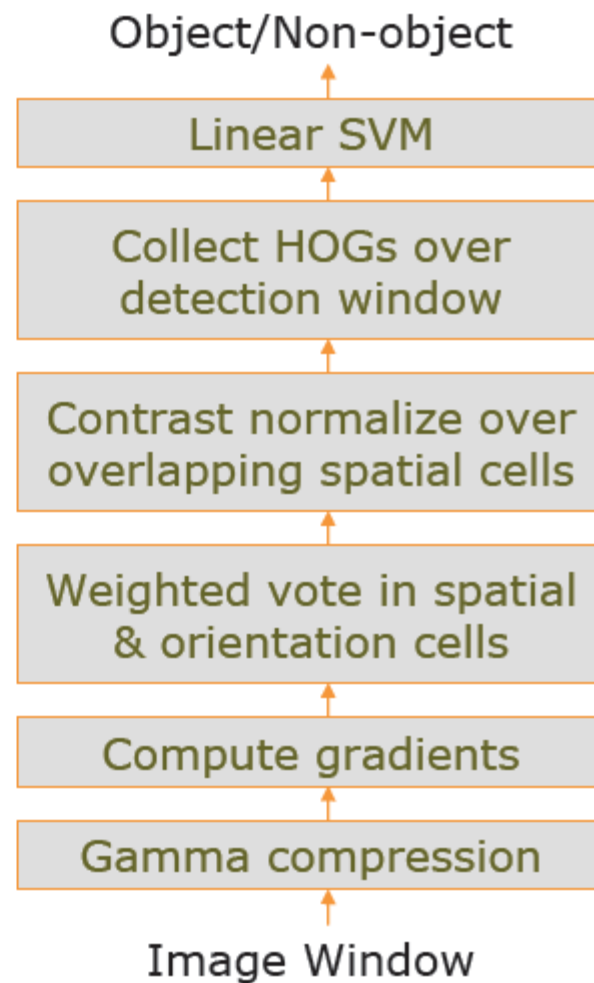
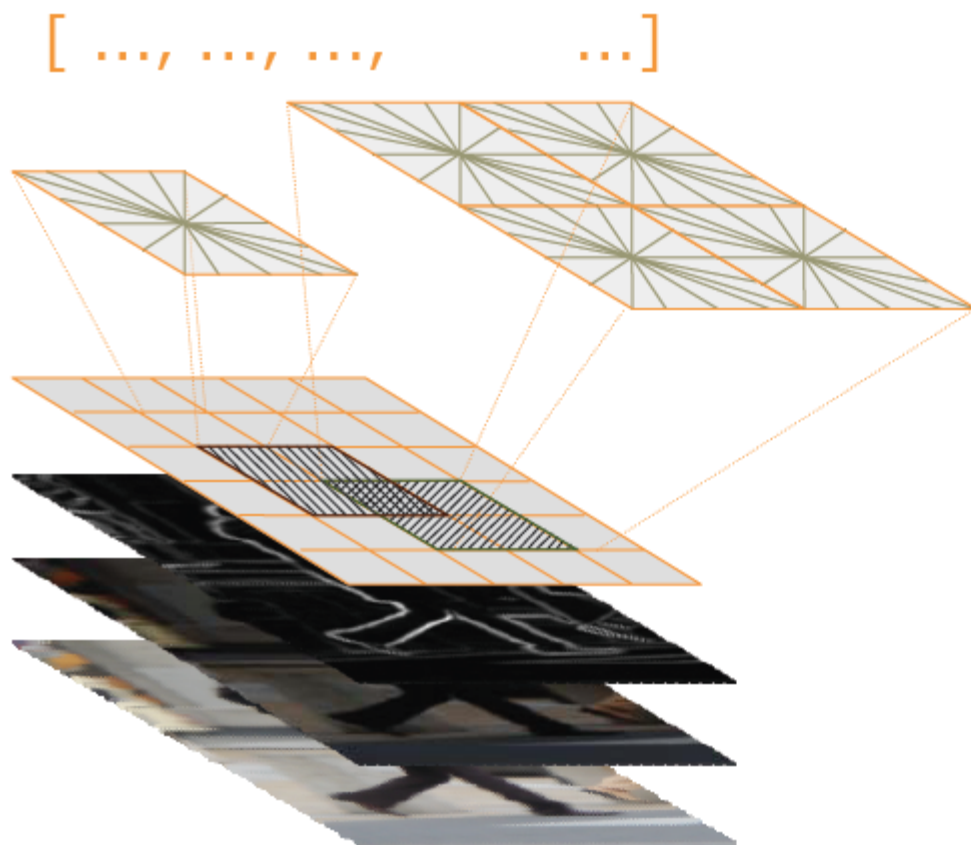
System overview



HOG features



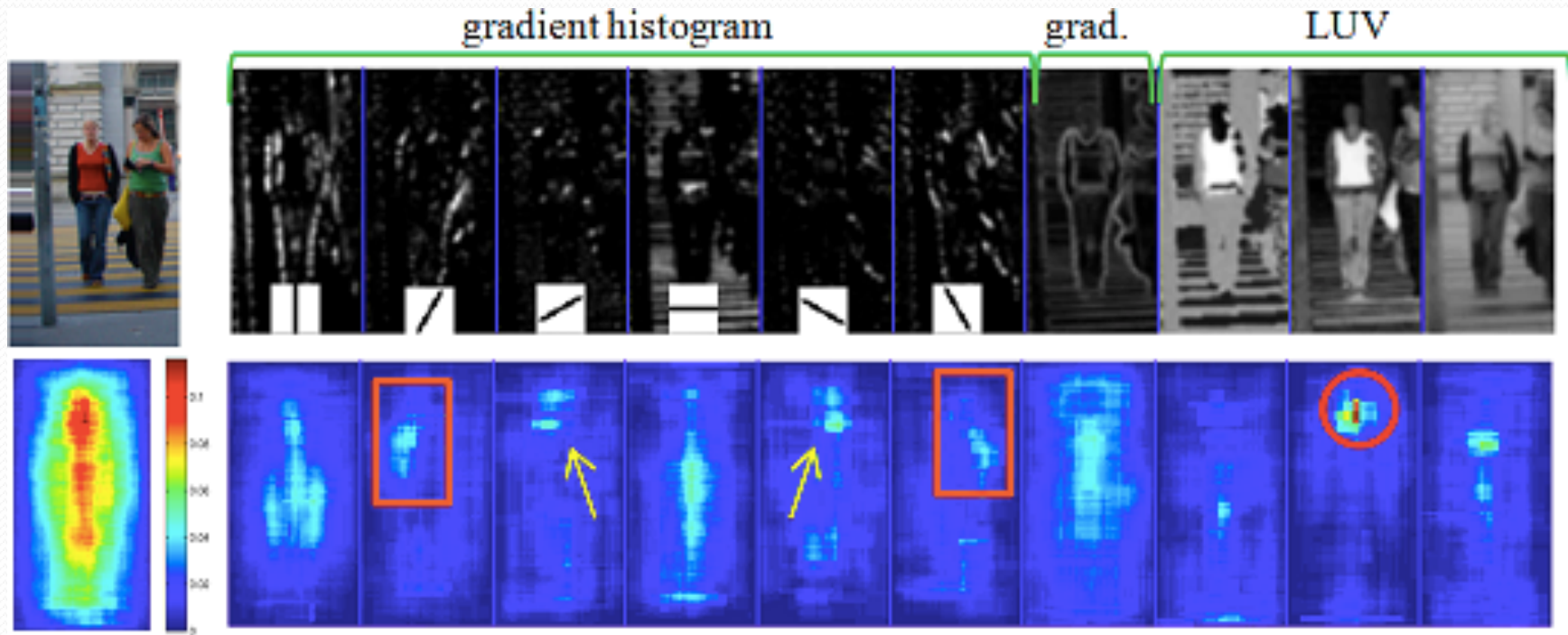
HOG-based detection



ICF – integral channel features

P. Dollár, Z. Tu, P. Perona and S. Belongie Integral Channel Features

BMVC 2009



Методы поиска снизу-вверх. ("Сборочный" метод.)

- Методы поиска объектов из этой группы осуществляют сборку объекта по частям.
(т.е. сначала происходит поиск частей объекта, затем проверяется их взаимное расположение – оно должно удовлетворять допустимым преобразованиям шаблонного объекта (напр. заданы допуски на возможные аффинные преобразования))
- Каждая гипотеза о местоположении части объекта генерирует гипотезу о положении всего объекта.
(аналогично hough transform при поиске линии)

