

# Алгоритм детектирования лица Виолы-Джонса

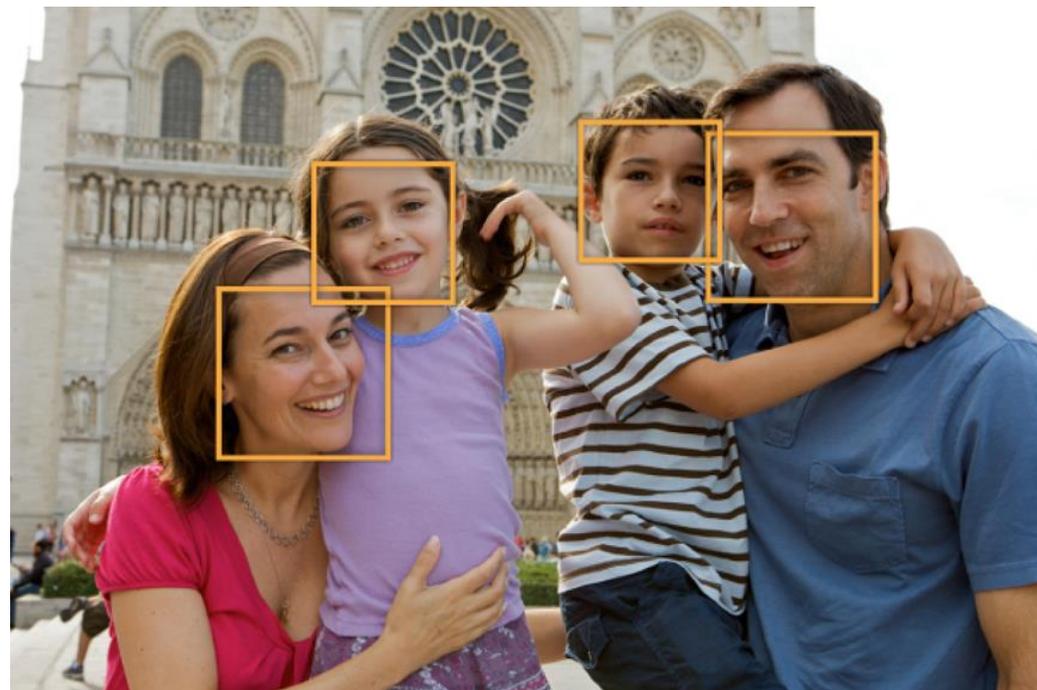
Копелиович Михаил,

аспирант Института математики, механики и компьютерных наук,

[kor@km.ru](mailto:kor@km.ru)

# Алгоритмы детектирования лица

- На входе: изображение.
- На выходе: координаты найденных прямоугольников лица в системе координат изображения.
- Использование:
  - Анализ фото
  - Анализ видео

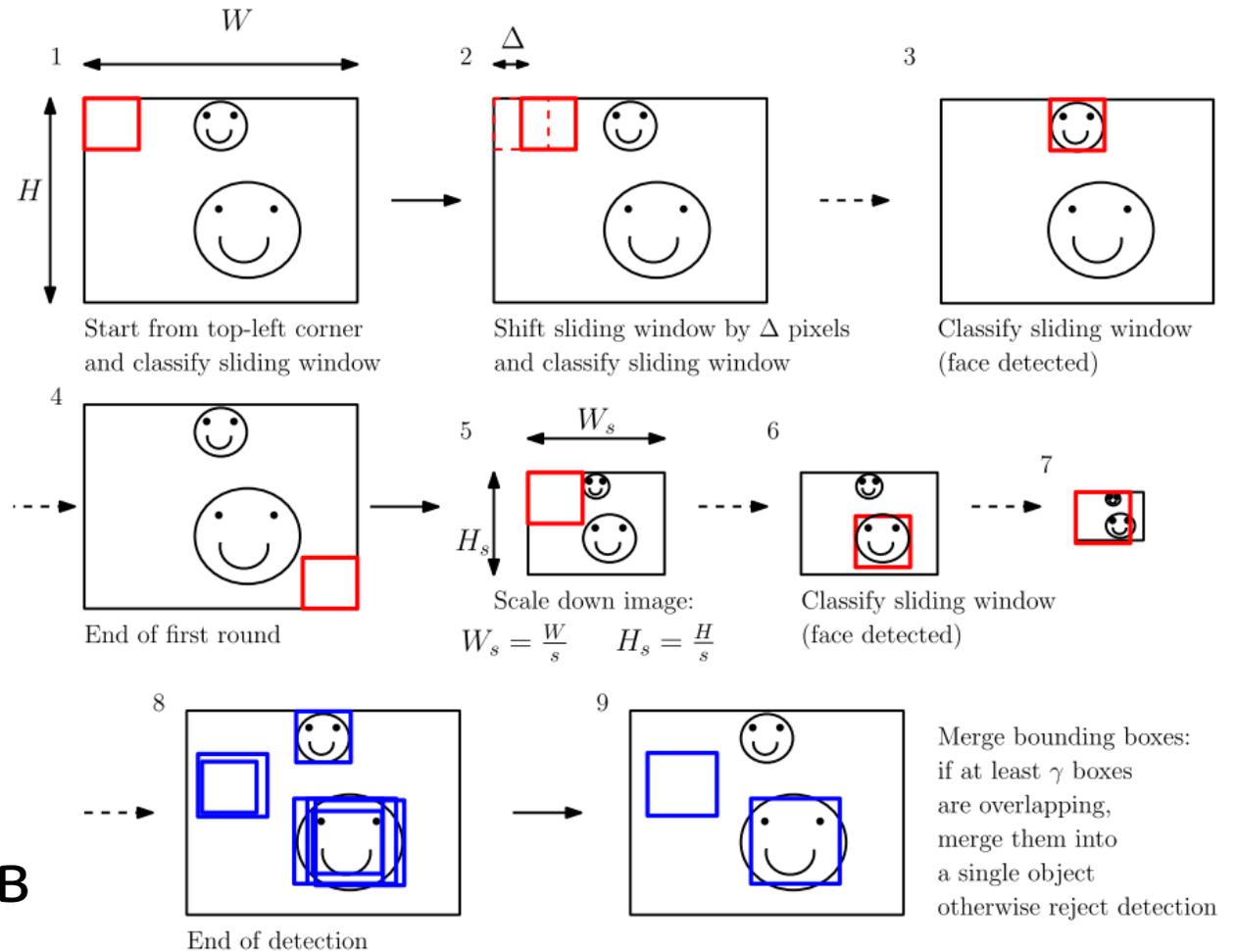


# I. Алгоритм Виолы-Джонса

1. Детектирование лица
2. Обучение алгоритма

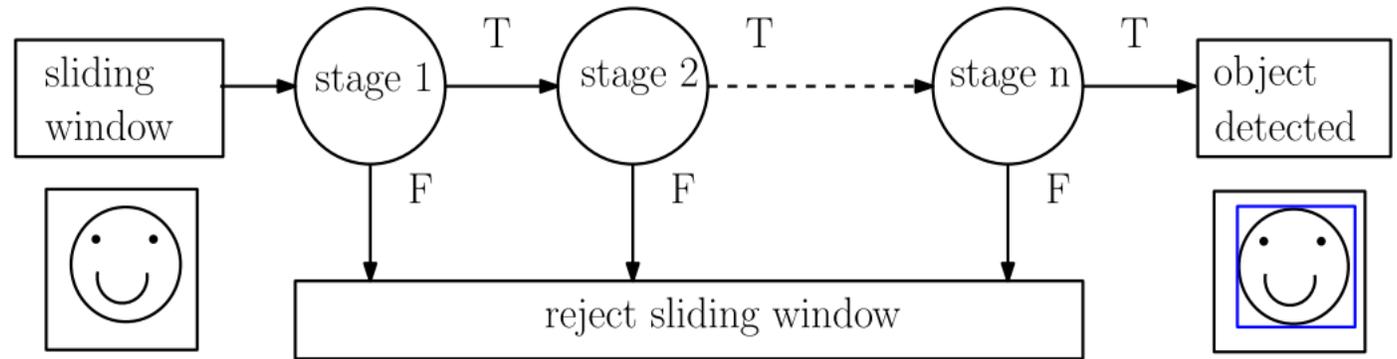
# Детектирование: общая схема

- Скользящее окно:
  - Шаг по X, шаг по Y
- Масштабирование:
  - Пирамида изображений
- Порог количества кандидатов в прямоугольники лица, наложенных друг на друга в одной области



# Детектирование: каскад классификаторов (1)

**Классификатор** – функция, определяющая принадлежность объекта к одному из известных классов.

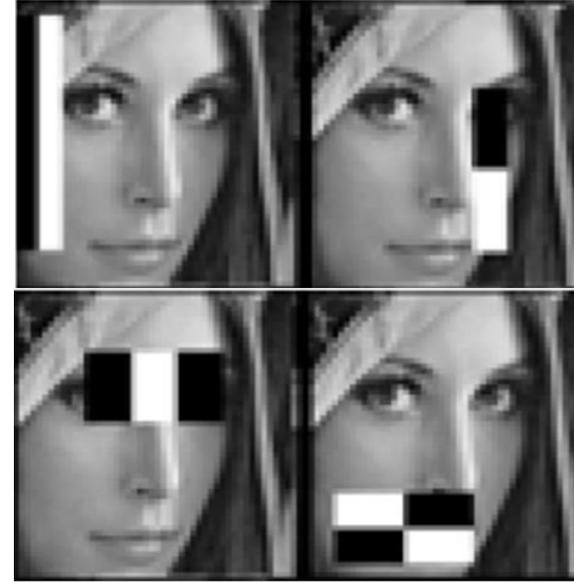
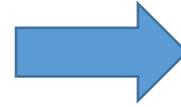
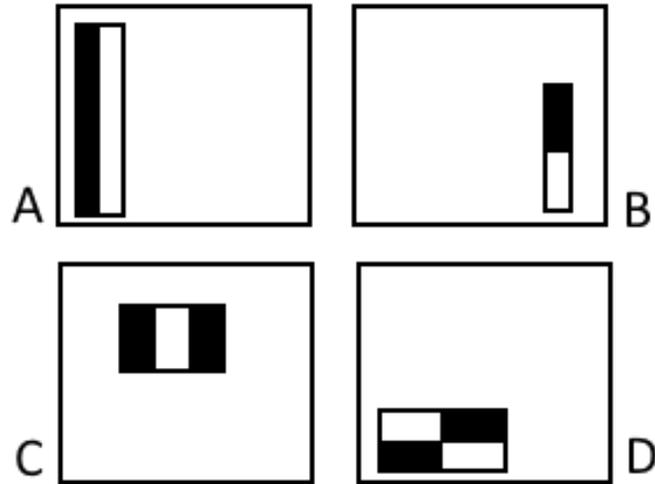


**Сильные** классификаторы:

- Сильный классификатор – функция от слабых классификаторов. В зависимости от ответов слабых классификаторов, возвращает ответ вида:
  - а) Точно не лицо;
  - б) Возможно, лицо.
- Сильные классификаторы применяются последовательно в порядке:
  - 1) уменьшения «подозрительности»;
  - 2) увеличения количества используемых слабых классификаторов.

# Детектирование: каскад классификаторов (2)

**Слабые** классификаторы:



- Слабый классификатор содержит признак Хаара (прямоугольную область, заданную в координатах относительно скользящего окна).
- Ответ слабого классификатора:
  - а) 1, если разность интенсивностей в чёрных и белых областях *больше* порога  $\eta$ ;
  - б) 0, иначе.
- Порог  $\eta$  определяется на этапе обучения.

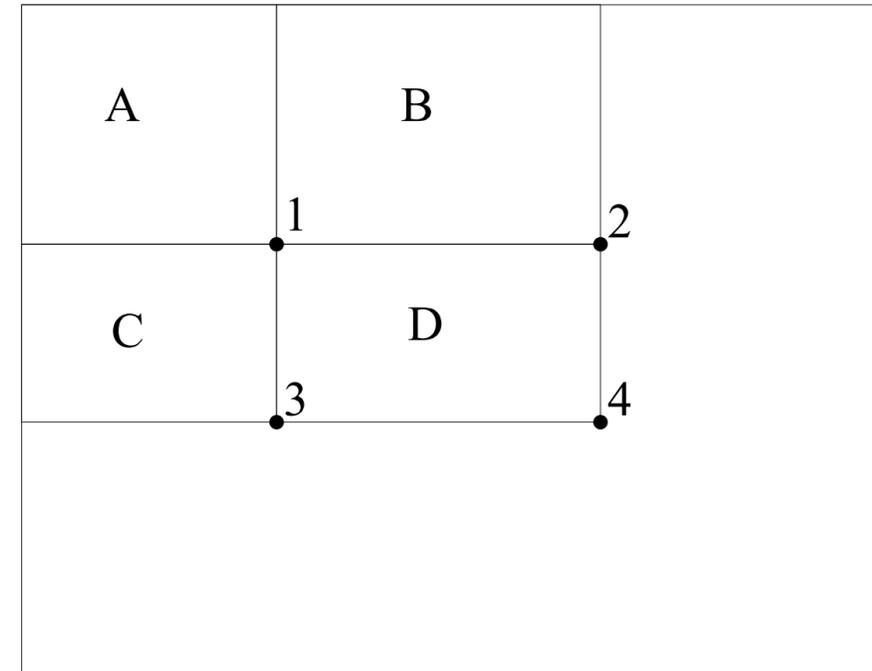
# Детектирование: интегральное изображение

- Интегральное изображение  $ii(N \times N)$ , построенное по исходному изображению  $I(N \times N)$  – матрица, каждый элемент которой равен **сумме** значений яркости соответствующих пикселей на исходном изображении, расположенных **слева** и **сверху** от него:

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

- Сумма яркостей пикселей в прямоугольнике D рассчитывается за константное время по формуле:

$$4 + 1 - 2 - 3.$$



# Детектирование: масштабирование

Алгоритм Reduce(I).

На входе: изображение I (оттенки серого).

1.  $I_{\text{smoothed}} := I * G$ ,

где  $G$  – ядро свёртки – матрица  $M \times M$  ( $3 \times 3$ ,  $5 \times 5$  или  $7 \times 7$ ), определяемая функцией Гаусса с нулевым математическим ожиданием:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

где  $\sigma$  – среднеквадратичное отклонение,

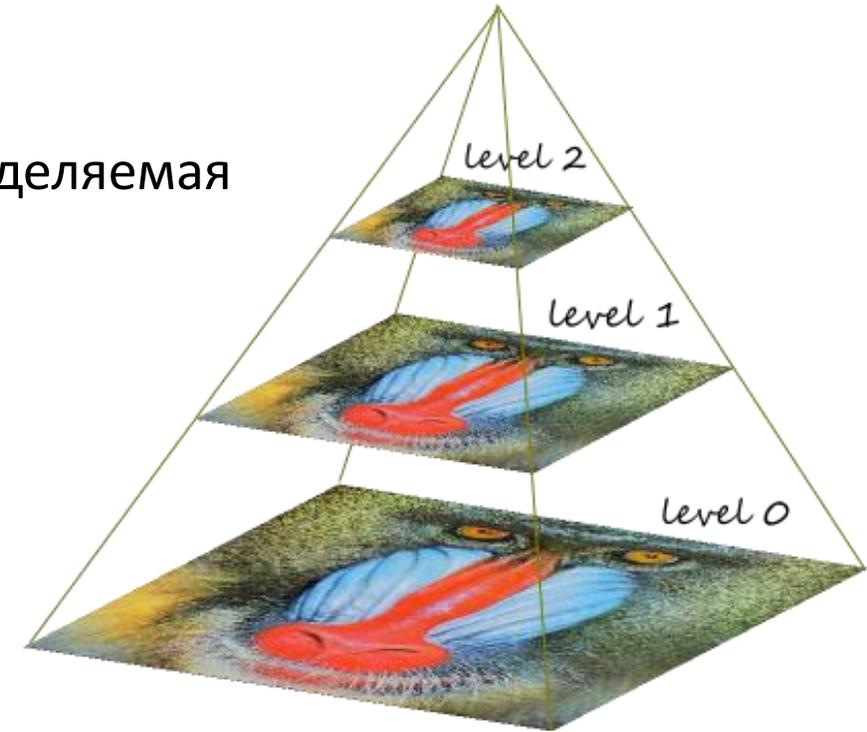
$x, y$  – расстояние до центра ядра;

Свёртка задана формулой:

$$(I * G)[x, y] = \sum_{n, m=-M}^M I[x-n, y-m]G[n, m],$$

2.  $I_{\text{reduced}} := \langle I_{\text{smoothed}} \text{ с уменьшенной в } K \text{ раз (} K=1.2 \text{) шириной и высотой; значения яркостей получены посредством интерполяции} \rangle$

На выходе: масштабированное изображение  $I_{\text{reduced}}$ .



# Детектирование: алгоритм

**На входе:** изображение  $I$  (оттенки серого).

1. Построение интегрального изображения  $I\_integr$ .
2. Обработка изображения  $I\_integr$  скользящим окном размера  $N \times N$  ( $N=24$ ):
  - 2.1. Координаты верхнего левого угла скользящего окна:  $(x,y) := (0,0)$ .
  - 2.2. Последовательное применение сильных классификаторов к скользящему окну. Для каждого сильного классификатора:
    - 2.2.1. Применение слабых классификаторов.
    - 2.2.2. Если общий результат для данного классификатора отрицательный, перейти к **Шагу 2.4**.
  - 2.3. Запомнить координаты текущего окна, как *координаты прямоугольника лица*.
  - 2.4. Сдвиг скользящего окна:
    - 2.4.1.  $x:=x+1$ . Если окно **не вышло** за границы изображения, перейти к **Шагу 2.2**.
    - 2.4.2.  $x:=0$ .  $y:=y+1$ . Если окно **вышло** за границы изображения, перейти к **Шагу 3**.
3. Уменьшение изображения  $I := Reduce(I)$ . Если ширина или высота уменьшенного изображения больше фиксированного порога  $\delta$  ( $\delta \approx 100$ ), возврат на **Шаг 1**.
4. Определение наложенных друг на друга найденных *прямоугольников лица*. Для каждой области наложения:
  - 4.1. Вычисление количества прямоугольников лица, принадлежащих области.
  - 4.2. Если количество прямоугольников больше заданного порогового значения  $\gamma$ , *результатирующий прямоугольник лица* вычисляется путём усреднения координат соответствующих точек прямоугольников лица, принадлежащих данной области наложения.

**На выходе:** координаты результирующих прямоугольников лица.

# Обучение

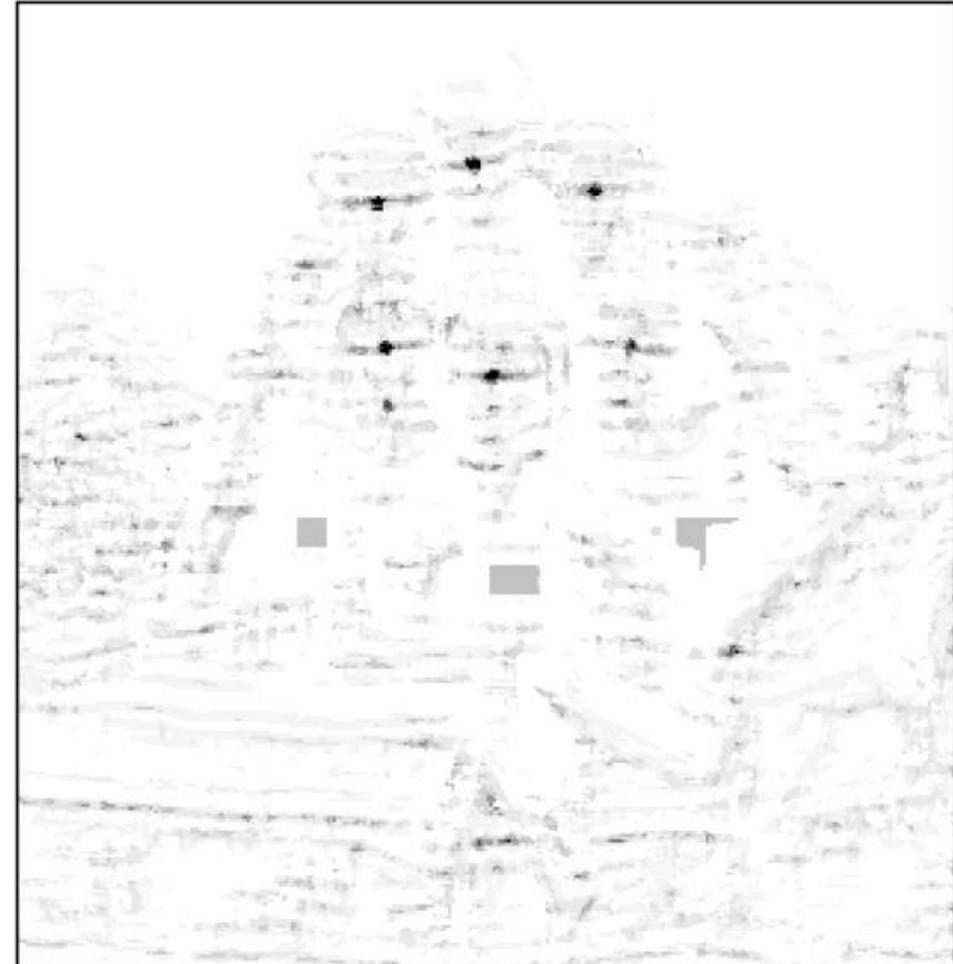
- Последовательная генерация сильных классификаторов из множества слабых классификаторов.
- Можно варьировать:
  - Для каждого слабого классификатора:
    - Вид признака Хаара
    - Координаты признака Хаара
    - Масштаб признака Хаара
    - Весовой коэффициент
  - Количество слабых классификаторов для каждого сильного классификатора
  - Количество сильных классификаторов

## II. Ускорение алгоритма Виолы-Джонса

### 1. Метод ускорения без распараллеливания

# Адаптивное скользящее окно в реальном времени (RASW)

- Подбор шага скользящего окна в зависимости от номера отклонённого сильного классификатора



# ССЫЛКИ

- **Оригинальная работа по алгоритму Виолы-Джонса:**

P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proc. CVPR (Kauai, HI, 2001), pp. 511–518.

- **Описание алгоритма Виолы-Джонса и его обучения на русском языке:**

<http://habrahabr.ru/post/133826/>

- **Метод RASW (адаптивный шаг скользящего окна):**

F. Comaschi, S. Stuijk, T. Basten and H. Corporaal "Rasw: A runtime adaptive sliding window to improve viola-jones object detection", Distributed Smart Cameras (ICDSC), 2013 Seventh International Conference.

- **Перевод работы по методу RASW на Хабрахабре:**

<http://habrahabr.ru/post/216019/>