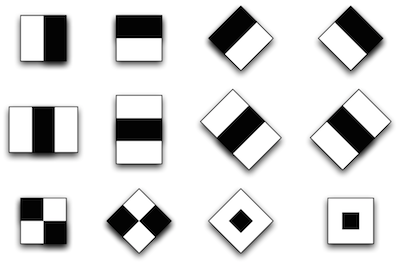
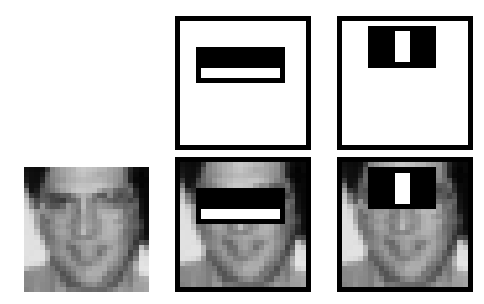
Метод Виолы-Джонса

Обычно у каждого метода есть основа, то, без чего этот метод не мог бы существовать в принципе, а уже над этой основой строится вся остальная часть. В методе Виолы-Джонса эту основу составляют примитивы Хаара, представляющие собой разбивку заданной прямоугольной области на наборы разнотипных прямоугольных подобластей:  
  
В оригинальной версии алгоритма Виолы-Джонса использовались только примитивы без поворотов, а для вычисления значения признака сумма яркостей пикселей одной подобласти вычиталась из суммы яркостей другой подобласти [1]. В развитии метода были предложены примитивы с наклоном на 45 градусов и несимметричных конфигураций. Также вместо вычисления обычной разности, было предложено приписывать каждой подобласти определенный вес и значения признака вычислять как взвешенную сумму пикселей разнотипных областей [2]:   
  
https://habrastorage.org/getpro/habr/post_images/412/e91/fab/412e91fabeb24e4a352cfa522f17cb52.gif  
  
Почему в основу метода легли примитивы Хаара? Основной причиной являлась попытка уйти от пиксельного представления с сохранением скорости вычисления признака. Из значений пары пикселей сложно вынести какую-либо осмысленную информацию для классификации, в то время как из двух признаков Хаара строится, например, первый каскад системы по распознаванию лиц, который имеет вполне осмысленную интерпретацию [1]:  
  
Сложность вычисления признака так же как и получения значения пикселя остается O(1): значение каждой подобласти можно вычислить скомбинировав 4 значения интегрального представления (Summed Area Table — SAT), которое в свою очередь можно построить заранее один раз для всего изображения за O(n), где n — число пикселей в изображении, используя формулу [2]:   
  
https://habrastorage.org/getpro/habr/post_images/403/cfb/8db/403cfb8db58b1c33919002155e3b9f75.gif  
https://habrastorage.org/getpro/habr/post_images/3cd/12d/327/3cd12d3270f8c11b3076876dd84eac81.gif  
  
Это позволило создать быстрый алгоритм поиска объектов, который пользуется успехом уже больше десятилетия. Но вернемся к нашим признакам. Для определения принадлежности к классу в каждом каскаде, находиться сумма значений слабых классификаторов этого каскада. Каждый слабый классификатор выдает два значения в зависимости от того больше или меньше заданного порога значение признака, принадлежащего этому классификатору. В конце сумма значений слабых классификаторов сравнивается с порогом каскада и выносится решения найден объект или нет данным каскадом. Ну хватит теории, перейдем к практике!   
Мы уже давали ссылку на XML нашего классификатора автомобильных номеров, который можно найти в мастере проекта opencv ([ссылка](https://github.com/Itseez/opencv/blob/master/data/haarcascades/haarcascade_russian_plate_number.xml)). Посмотрим на его первый каскад:

<maxWeakCount>6</maxWeakCount>

<stageThreshold>-1.3110191822052002e+000</stageThreshold>

<weakClassifiers>

<\_>

<internalNodes>

0 -1 193 1.0079263709485531e-002</internalNodes>

<leafValues>

-8.1339186429977417e-001 5.0277775526046753e-001</leafValues></\_>

<\_>

<internalNodes>

0 -1 94 -2.2060684859752655e-002</internalNodes>

<leafValues>

7.9418992996215820e-001 -5.0896102190017700e-001</leafValues></\_>

<\_>

<internalNodes>

0 -1 18 -4.8777908086776733e-002</internalNodes>

<leafValues>

7.1656656265258789e-001 -4.1640335321426392e-001</leafValues></\_>

<\_>

<internalNodes>

0 -1 35 1.0387318208813667e-002</internalNodes>

<leafValues>

3.7618312239646912e-001 -8.5504144430160522e-001</leafValues></\_>

<\_>

<internalNodes>

0 -1 191 -9.4083719886839390e-004</internalNodes>

<leafValues>

4.2658549547195435e-001 -5.7729166746139526e-001</leafValues></\_>

<\_>

<internalNodes>

0 -1 48 -8.2391249015927315e-003</internalNodes>

<leafValues>

8.2346975803375244e-001 -3.7503159046173096e-001</leafValues></\_></weakClassifiers>

На первый взгляд кажется, что здесь куча непонятных цифр и странной информации, но на самом деле все просто: weakClassifiers — набор слабых классификаторов, на основе которых выносится решение о том, находится объект на изображении или нет, internalNodes и leafValues — это параметры конкретного слабого классификатора. Расшифровка internalNodes слева направо: первые два значения в нашем случае не используется, третье — номер признака в общей таблице признаков (она располагается дальше в XML файле под тегом features), четвертое — пороговое значение слабого классификатора. Так как у нас используется классификатор, основанный на одноуровневых решающих деревьях ([decision stump](http://en.wikipedia.org/wiki/Decision_stump)), то если значение признака Хаара меньше порога слабого классификатора (четвертое значение в internalNodes), выбирается первое значение leafValues, если больше — второе.

Ссылка на источник: <https://habr.com/ru/company/recognitor/blog/228195/>