

ФЕДЕРАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ЦЕНТР  
«ИНФОРМАТИКА И УПРАВЛЕНИЕ» РОССИЙСКОЙ АКАДЕМИИ НАУК  
ИНСТИТУТ СИСТЕМНОГО АНАЛИЗА

На правах рукописи  
УДК 004.93'11, 004.93'12, 004.932.2

*УСИЛИН Сергей Александрович*

Алгоритмическое развитие Виола-Джонсовских  
детекторов для решения прикладных задач  
распознавания изображений

Специальность 05.13.01 – «Системный анализ, управление и  
обработка информации (информационно-вычислительное обеспечение)»

Диссертация на соискание ученой степени  
кандидата технических наук

Научный руководитель  
д-р техн. наук, проф., чл.-кор. РАН  
Арлазаров Владимир Львович

Москва – 2017

## ОГЛАВЛЕНИЕ

Введение.....	5
1 Метод Виолы и Джонса детектирования объектов на изображении.....	17
1.1 Понятие цифрового изображения.....	17
1.2 Задача поиска объектов и основы статистического обучения .....	19
1.2.1 Задача построения классификатора .....	20
1.2.2 Бустинг в задачах классификации.....	22f
1.2.3 Оценка качества работы детекторов .....	23
1.3 Оригинальный методы Виолы и Джонса.....	26
1.3.1 Прямоугольные признаки Хаара .....	27
1.3.2 Обучение классификаторов с помощью AdaBoost.....	29
1.3.3 Каскадная структура классификаторов .....	30
1.3.4 Результаты практического тестирования метода.....	34
1.4 Анализ существующих модификаций метода Виолы и Джонса .....	35
1.4.1 Модификации в части пространства признаков .....	35
1.4.2 Модификации в части обучающего алгоритма .....	42
1.4.3 Модификации в части каскадной структуры классификатора .....	44
1.5 Применение метода Виолы и Джонса в индустриальных распознающих системах.....	46
1.6 Конкретизация цели и задачи исследования .....	47
2 Построение устойчивых признаков для алгоритма Виолы и Джонса, учитывающих геометрические особенности объектов .....	50
2.1 Карта направленных границ.....	52
2.2 Контурные признаки Хаара.....	55
2.3 Численный эксперимент и результаты .....	57

2.3.1 Распознавание образов колес .....	57
2.3.2 Распознавание номера кредитной карты .....	61
2.4 Выводы по второму разделу .....	63
3 Высокоуровневая структура классификатора Виолы и Джонса в виде решающего дерева сильных классификаторов .....	64
3.1 Решающее дерево сильных классификаторов.....	65
3.2 Метрика сравнения классификаторов.....	72
3.3 Численный эксперимент и результаты .....	73
3.3.1 Распознавание образов колес .....	73
3.3.2 Распознавание логотипа платежной системы VISA.....	74
3.4 Выводы по третьему разделу .....	78
4 Способ ускорения процедуры поиска объектов в видеопотоке путем адаптивного выбора распознающих классификаторов .....	79
4.1 Задача об n-руском бандите .....	80
4.2 Постановка задачи адаптивного выбора распознающих классификаторов.....	82
4.3 Алгоритм адаптивного выбора распознающих классификаторов .....	83
4.4 Численный эксперимент и результаты .....	86
4.5 Выводы по четвертому разделу .....	92
5 Проектирование и реализация программного комплекса детектирования объектов методом Виолы и Джонса.....	94
5.1 Существующие программные решения детектирования объектов методом Виолы и Джонса .....	94
5.2 Технические и функциональные требования к разрабатываемому программному комплексу детектирования объектов .....	97
5.3 Разработка программного комплекса objed.....	99
5.3.1 Выбор средств разработки .....	99

5.3.2 Архитектура программного комплекса objed .....	101
5.4 Использование программного комплекса objed в индустриальных распознающих системах .....	104
5.5 Выводы по пятому разделу .....	107
Заключение .....	108
Список сокращений и условных обозначений .....	110
Список литературы .....	111
Список публикаций автора.....	123
Приложение А Модуль обучения программного комплекса objed .....	127
Приложение Б Алгоритм AdaBoost.....	138
Приложение В Охранные документы на результаты интеллектуальной деятельности .....	144

## Введение

Практически во всех областях человеческой деятельности ежедневно возникает необходимость решения задач анализа данных, выявления зависимостей между атрибутами, прогнозирования ситуаций, явлений и процессов. Причем с каждым днем таких задач становится все больше, объем доступной информации увеличивается, благодаря бурному развитию данных центров и облачных технологий совершенствуются технологии ее сбора, хранения и организации. В связи с этим требуются универсальные подходы и методы, пригодные для решения прикладных задач анализа данных. В качестве такого решения в настоящее время активно используются подходы математической теории распознавания.

Содержательная постановка задачи обучения машин распознаванию образов появилась в конце 50-х годов и заключалась в построении способа обучения машин решать классификационные задачи так же, как это делают живые существа. В это же появились первые нейросетевые модели распознавания, в частности перцептрон Розенблатта [1]. К концу 60-х годов уже были разработаны различные подходы для решения задач распознавания в рамках статистических и персептронных моделей. Большой вклад в развитие теории распознавания внесли советские (в последующем российские) ученые и научные школы. Здесь необходимо, прежде всего, отметить школы: Айвазяна С.А. – ЦЭМИ АН СССР (Айвазян С.А., Бухштабер В.М., Енюков И.С. и др.), Айзermana M.A. – ИАТ АН СССР (Айзerman M.A., Раверман Э.М., Розонеэр Л.И. и др.), Бонгарда М.М. – ИППИ АН СССР (Бонгард М.М., Вайнцвайг М.Н., Максимов В.В. и др.), Вапника В.Н. – ИАТ АН СССР (Вапник В.Н., Червоненкис А.Я., Стефанюк А.Р. и др.), Журавлева Ю.И. – ВЦ АН СССР (Журавлев Ю.И., Рудаков К.В., Воронцов К.В. и др.), Загоруйко Н.Г. – Институт математики СО АН СССР, Новосибирский ГУ (Загоруйко Н.Г., Елкина В.Н., Лбов Г. С. и др.), Ковалевского В.А. – Институт кибернетики АН УССР, Киев (Ковалевский В.А., Винценок Т.К., Шлезингер М.И. и др.), Сойфера В.А. – СГАУ (Сойфер В.А., Сергеев В.В., Храмов А.Г. и др.), Якубовича В.А. – Ленинградский ГУ (Якубович В.А., Тимофеев А.В., Шмидт А.А. и др.).

В результате такого активного развития теории распознавания образов было разработано множество алгоритмов построения классификаторов. Дальнейшее развитие этого направления, особенно при решении сложных практических задач, привело к разработке нового типа классификаторов, использующих для построения решающих правил композицию («коллектив») алгоритмов. Здесь необходимо выделить результаты, полученные учёными из научной школы Ю.И. Журавлева (см., например, [2,3]). Позднее аналогичные исследования появились и в других странах. Так, например, в 90-е годы Йоавом Фрайндом и Робертом Шапиром в работах был предложен достаточно эффективный итеративный алгоритм построения «сильного» распознавателя путем объединения нескольких «слабых», названный авторами AdaBoost (adaptive boosting) [4–6]. Алгоритм представляет собой процедуру пошагового объединения «слабых» классификаторов в «сильные», при этом на каждой итерации особое внимание обращается на неверно классифицированные прецеденты предыдущих этапов. Благодаря своей простоте и эффективности AdaBoost быстро набрал популярность и начал активно применяться для решения различных задач распознавания.

К концу XX века область, связанная с распознавания образов, находилась на достаточно высоком научном уровне. Однако большинство алгоритмов были предназначены для решения задачи классификации (например, для определения присутствия целевого объекта на изображении), не возвращая каких-либо знаний о точном местоположении. При этом локализация объекта (которая может быть определена с помощью обрамляющего прямоугольника, путем указания центра и характерного размера, задания контура и т.п.) является важной задачей в области автоматической обработки цифровых изображений и находит свое применение в таких областях, как сегментация изображений вида объект/фон, анализ пространственных взаимоотношений объектов на изображении, определение и восстановление траекторий движения и т.п.

Для случая, когда локализация объектов определяется с помощью обрамляющего прямоугольника, на протяжении многих лет применяют метод сканирующего (скользящего) окна (англ. sliding window, scan window) [7–11].

Данный метод заключается в последовательном анализе всевозможных прямоугольных подокон изображения, взятых с разным сдвигом и разным масштабом, с помощью бинарного классификатора и последующим объединением результатов классификации для определения локализации объектов. Тонким местом для данного метода является жесткое требование к вычислительной сложности используемого детектора: для изображения, содержащего  $n$  пикселей общее количество подокон, которое должно быть проанализировано с помощью классификатора, – порядка  $n^2$ . Такое жесткое ограничение препятствовало практическому применению алгоритмов поиска объектов в индустриальных распознающих системах, где зачастую требуется быстрое (за доли секунды) принятие решения.

Революционный прорыв в области детектирования объектов был сделан американскими учеными Полом Виолой и Майклом Джонсом в 2001 году, предложившими алгоритм поиска лица на изображениях, обеспечивающий обработку до 15 кадров VGA разрешения в секунду [7,8]. Суть предложенного алгоритма заключается в объединении в единую конструкцию четырех подходов: использование вычислительно легковесных и обладающих хорошей обобщающей способностью признаков Хаара, обучение классификаторов с помощью бустинга, обеспечение высокой производительности за счет использования каскада классификаторов, а также применение метода скользящего окна для определения локализации объектов. Несмотря на то, что данный метод изначально разрабатывался для поиска лиц, он оказался применим для детекции широкого класса объектов, обладающих жесткой геометрией. В результате алгоритм Виолы и Джонса стал классическим подходом для решения задач локализации объектов на изображении [12–25].

На сегодняшний день степень разработанности метода Виолы и Джонса достаточно высока. Исследователями по всему миру предложено множество модификаций оригинального алгоритма и отдельных его частей. В частности, Лиенхард и Майдт предложили расширить оригинальное пространство прямоугольных признаков Хаара за счет повернутых на  $45^\circ$  в паре с быстрым

способом подсчета таких признаков [26]. Позже Мессом и Барзак дополнили концепцию наклонных признаков Хаара, предложив подход к вычислению признаков под произвольными углами [27]. В 2010 развитие яркостных признаков Хаара достигло своего пика за счет работы [28], в которой исследователи из института в Гилфорде (Мин-Три Фам и Ян Гао) вместе с коллегами из Университета в Сингапуре (Виет-Дунг Хуан и Тат-Джен Чам) предложили алгоритм построения полигональных признаков Хаара. Помимо расширения допустимой формы признаков в настоящее время предложен ряд модификаций относительно самой природы вычисляемых признаков. Так, Далал и Триггс предложили использовать гистограммы направленных градиентов [10] для построения классификаторов, устойчивых к яркостным изменениям, исследователи из Microsoft (Жан, Сан и Танг) объединили концепцию HoG и признаков Хаара [29], а Пиотр Доллар с коллегами из Калифорнийского института предложил использовать мультиспектральное представление изображения для одновременного вычисления признаков разной природы [30,31].

Немало работ посвящено модификациям метода Виолы и Джонса в части алгоритма обучения сильных классификаторов. Большое количество работ посвящено различным вариантам аппроксимации функционала качества композиции непрерывно дифференцируемыми оценками сверху [32–38]. Отдельное внимание стоит обратить на алгоритм LogitBoost [35], сформулированный Фридманом, Хасти и Тибширани, который представляет собой адаптацию подхода логистической регрессии к идеи бустинга. В результате этого построенный классификатор обладает набором интересных особенностей: обученный таким образом классификатор оказывается оптимальным с точки зрения байесовской классификации, позволяет строить более устойчивые к шуму классификаторы, возникает возможность получать численные оценки вероятности принадлежности исследуемых объектов к найденным классам.

В дополнение к новым методам обучения сильных классификаторов, учеными были предложены альтернативные способы построения каскадов. Одна из первых модификаций была предложена учеными из Microsoft (Сяо, Чжу, Чжан)

и заключалась в использовании ответов классификации ранних уровней каскада в качестве специального признака на очередном уровне [39]. В результате получается своего рода «связанный каскад», который в англоязычной литературе встречается в виде термина *chain cascade* [39] или *nesting-structure cascade* [40]. Позднее, в 2005 Боурдев и Брандт представили модель так называемого «легкого каскада» (англ. *soft cascade*), которая структурно похожа на сильный классификатор, обученный с помощью бустинга, однако способна отклонять отрицательные области после вычисления очередного слабого классификатора. В 2007 году Пол Виола и Ча Жанг в работе [41] развили идею такого каскада, добавив алгоритм подбора промежуточных порогов.

Несмотря на наличие приведенных выше модификаций алгоритма Виолы и Джонса, применение его для решения большого диапазона прикладных задач в индустриальных системах распознавания часто оказывается затруднительным. Во-первых, значения классических признаков Хаара, даже в известных нормализованных модификациях, оказываются неинвариантны к изменению освещенности, а модификации признакового пространства, оперирующие с граничными точками, оказываются либо вычислительно трудоемкими, либо чувствительны к шумовым выбросам, либо неприспособленными к масштабированию. Во-вторых, алгоритм Виолы и Джонса, как и представленные модификации, решает задачу поиска объектов в «лабораторной» постановке: наборы прецедентов известны, зафиксированы и не требуют дополнительной кластеризации на отдельные подтипы, детекция объектов выполняется на отдельных стационарных изображениях. Однако индустриальные системы распознавания предъявляют к алгоритмам поиска объектов дополнительные требования. Так, например, в качестве источника данных часто выступает видеокамера, обеспечивающая вместо отдельных стационарных изображений коррелированную последовательность кадров, которые могут быть использованы для повышения производительности алгоритма. В качестве обучающих примеров часто выступают «сырые» данные, требующие предварительного разделения на подтипы. И, наконец, хотя в соответствии с оригинальным методом обучение

детектора ведется в «пакетном режиме» (обучающая выборка известна и зафиксирована), на практике регулярно возникает задача «дообучения» детектора в связи с появлением новых данных.

Таким образом, алгоритмическое развитие метода Виолы и Джонса, устраняющее указанные недостатки, является высоко востребованным, а тема исследований диссертационной работы – **актуальной**. Метод Виолы и Джонса активно применяется в области робототехники [42,43], разработки в которой входят в перечень приоритетных направлений развития науки, технологий и техники Российской Федерации, утвержденный Указом Президента РФ № 899 от 07.07.2011 года. Исследования по теме диссертации поддержаны Российским фондом фундаментальных исследований как в части инициативных научно-исследовательских проектов, так и в области междисциплинарных исследований. Основные результаты диссертации были получены в процессе выполнения работ по следующим научным грантам РФФИ:

- 17-29-03161\_офи\_м Каскадные алгоритмы локализации и прослеживания протяженных объектов на мобильных устройствах;
- 17-29-03297\_офи\_м Исследование возможности создания энергоэффективных аппаратных устройств для мобильных устройств комплексов идентификации и верификации личности в составе систем технического зрения наземных робототехнических комплексов;
- 16-29-09508\_офи\_м Алгоритмическое развитие Виола-Джонсовских детекторов в задачах идентификации объектов в потоках гетерогенных данных;
- 13-01-12106\_офи\_м Методы обучения и дообучения для систем видеоклассификации сложноструктурированных объектов в неконтролируемых условиях;
- 13-07-00870\_А Алгоритмы трекинга и детекции ригидных трёхмерных объектов.

**Объектом исследования** является метод Виолы и Джонса поиска объектов на изображении.

**Предметом исследования** является алгоритмическое развитие Виола-Джонсовских детекторов для решения прикладных задач распознавания изображений.

Основная цель диссертационной работы заключается в повышении эффективности метода Виолы и Джонса (повышение производительности и качества детектирования объектов) при использовании в индустриальных системах распознавания.

Для ее достижения требуется решить следующие задачи:

1. Провести критический анализ оригинального метода Виолы и Джонса и существующих актуальных модификаций, а также исследовать индустриальные распознающие системы на предмет дополнительных требований к методам детектирования объектов.

2. Разработать семейство вычислительно эффективных признаков, обеспечивающих высокий уровень обобщения объектов и устойчивых к различным параметрам освещенности, применимых для использования в индустриальных распознающих системах.

3. Построить математическую модель и разработать алгоритм обучения высокоуровневого классификатора, обеспечивающую функциональность дообучения при расширении обучающей выборки.

4. Построить математическую модель и разработать алгоритм многоклассовой детекции объектов на видеопоследовательности для случая поочередной демонстрации целевых объектов.

5. Разработать инструментарий, содержащий имплементацию полученных в рамках диссертации результатов и применимый для использования в индустриальных распознающих системах.

**Методы исследования**, используемые в диссертационной работе, включают в себя системный анализ, методы численного и натурного экспериментов, методы обработки цифровых изображений и распознавания образов, объектно-ориентированный анализ и проектирование.

**Основные положения, выносимые на защиту:**

1. Разработано семейство признаков для алгоритма Виолы и Джонса, устойчивых к различным параметрам освещенности и учитывающих геометрические особенности объектов.

2. Предложена математическая модель высокоуровневого классификатора Виолы и Джонса в виде решающего дерева сильных классификаторов, реализующая функциональность дообучения при расширении набора обучающих прецедентов, а также обеспечивающая лучшее по сравнению с каскадом качество детектирования за счет наличия нескольких положительных выходов.

3. Разработан алгоритм, сводящий обучение древовидного классификатора к последовательному обучению классических каскадных классификаторов, а также позволяющий выполнять дообучения высокоуровневого классификатора при расширении набора обучающих прецедентов.

4. Разработаны математическая модель и алгоритм адаптивного выбора распознающего классификатора Виолы и Джонса в задаче многоклассовой детекции объектов в видеопоследовательности.

5. Создан программный комплекс обучения классификаторов, реализующий представленные в диссертационной работе подходы и алгоритмы.

**Научная новизна** выносимых на защиту результатов состоит в следующем:

– в рамках диссертационной работы формулируется и решается задача адаптации метода Виолы и Джонса к использованию в индустриальных условиях, требующая разработки новых моделей и подходов;

– представлено новое семейство признаков, устойчивых к различным параметрам освещенности и учитывающих геометрические особенности объектов за счет использования в качестве исходных данных карты направленных границ;

– предложена новая модель высокоуровневого классификатора в виде решающего дерева, обеспечивающая несколько выходов с положительным исходом, а также представлен алгоритм его обучения и дообучения в случае расширения выборки прецедентов;

– впервые предложен алгоритм многоклассовой детекции объектов на видеопоследовательности, использующий методы обучения машин с подкреплением.

**Теоретическая и практическая значимость работы.** Диссертационная работа имеет как теоретическую, так и практическую значимость.

Теоретическая значимость работы заключается в первую очередь в постановке исследуемой задачи, предложенном семействе признаков, а также в разработанных моделях и алгоритмах для описания структуры высокоуровневого классификатора в виде решающего дерева и задачи многоклассовой детекции объектов на видеопоследовательности. Полученные результаты могут быть использованы для дальнейшего развития науки в данной области.

Практическая значимость работы подтверждается использованием полученных результатов в следующих индустриальных системах распознавания. Результаты диссертационной работы внедрены в систему классификации автомобилей по геометрическим характеристикам АКТС-4 (эксплуатируется на 200 полосах пунктов взимания платы автомагистралей России М1 «Беларусь», М4 «Дон» и Западного Скоростного Диаметра Санкт-Петербурга), программные продукты распознавания документов Smart IDReader и Smart PassportReader (интегрированы в информационные решения крупнейших российских банков, страховых компаний и организаций различных государственных структур), а также в интеллектуальную систему автономного вождения C-Pilot (система содействия водителю, предназначена для установки на коммерческий транспорт).

**Степень достоверности и апробация результатов.** Основные положения и результаты диссертационной работы докладывались в виде очного доклада на следующих международных научных конференциях:

- Международной IEEE конференции «Image Processing» (ICIP'2010), Гонконг, 2010;
- Международной конференции «The 7th International Conference on Machine Vision» (ICMV'2014), Милан, Италия, 2014;

- Международной конференции «The 9th International Conference on Machine Vision» (ICMV'2016), Ницца, Франция, 2016;
- Международной конференции «The 10th International Conference on Machine Vision» (ICMV'2017), Вена, Австрия, 2017;
- Междисциплинарной школе-конференции Института проблем передачи информации им. А. А. Харкевича Российской академии наук (ИППИ РАН) «Информационные технологии и системы» (ИТИС) в 2015, 2014 и 2012 годах;
- Открытом германо-российским семинаре «8th Open German-Russian Workshop «Pattern Recognition and Image Understanding», г. Нижний Новгород, 2011.
- Научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук» в 2010, 2012 и 2013 годах.
- Открытом германо-российским семинаре «8th Open German-Russian Workshop «Pattern Recognition and Image Understanding», г. Нижний Новгород, 2011.
- Научной конференции МФТИ «Современные проблемы фундаментальных и прикладных наук» в 2010, 2012 и 2013 годах.

Помимо научных конференций результаты докторской работы были обсуждены на международном научно-исследовательском семинаре Национального комитета при Президиуме РАН по распознаванию образов и анализу изображений «Анализ и понимание изображений (Математические, когнитивные и прикладные проблемы анализа изображений и сигналов)», заседания которого на базе Федерального исследовательского центра «Информатика и управление» Российской академии наук.

**Публикации.** Основные научные результаты докторской работы изложены в 23 публикациях, в том числе: пять статей в изданиях, рекомендованных ВАК РФ, три публикации в изданиях, входящих в международные базы цитирования Scopus и Web of Science, патент на изобретение, два патента на полезную модель, а также три свидетельства на программу для электронных вычислительных машин (ЭВМ).

**Личный вклад соискателя** в работы с соавторами состоит в разработке нового семейства признаков, реализации алгоритма построения и исследовании свойств высокоуровневого классификатора в виде решающего дерева, разработке и реализации алгоритма многоклассовой детекции объектов на видеопоследовательности, архитектурном проектировании и имплементации модулей программного комплекса детекции объектов objed.

**Структура и объем диссертации.** Диссертация состоит из введения, пяти основных разделов, заключения, списка используемых по тексту работы сокращений и условных обозначений, списка использованной литературы, список публикаций автора по теме диссертации, а также трех приложений. Общий объем работы составляет 149 страниц (с учетом всех приложений), содержит 36 рисунков и 10 таблиц. Список литературы представлен 128 наименованиями.

В **первом** разделе содержится обзор оригинального метода поиска объектов Виолы и Джонса, представлены определения и теоремы, необходимые для подробного описания метода. Описаны существующие модификации алгоритма в части пространства признаков, обучающего алгоритма и каскадной структуры.

Во **втором** разделе описываются предложенные автором признаки Хаара, вычисляемые поверх карты направленных границ, позволяющие существенно улучшить обобщающую силу и повысить эффективность метода Виолы и Джонса для решения задачи поиска симметричных объектов. Показана эффективность предложенных признаков по сравнению с классическими яркостными на примере двух практических задач: распознавание образов колес и распознавание номера кредитной карты.

В **третьем** разделе описана модель высокоуровневого классификатора в виде решающего дерева, а также изложен алгоритм обучения такого классификатора, обеспечивающего несколько выходов (листьев) с положительным ответом. Благодаря такой высокоуровневой структуре классификатора обеспечивается лучшее качество детектирования объектов, а также возможность дообучения классификатора при расширении обучающих наборов.

В **четвертом** разделе представлен оригинальный алгоритм адаптивного выбора распознающих классификаторов Виолы и Джонса, который использует информацию, накопленную по уже обработанным видеокадрам с целью выбора наилучшего классификатора для текущего видеокадра. Показана эффективность алгоритма на примере задачи распознавания логотипа банковской карты в видеопотоке.

В **пятом** разделе приведено описание проектирования и реализации программного комплекса детекции объектов методом Виолы и Джонса, содержащего программную имплементацию изложенных в диссертационной работе теоретических результатов.

**Приложение А** настоящей диссертации содержит описание модуля обучения классификаторов программного комплекса *objed*, приведена краткая документация по настройке и запуску отдельных утилит.

**Приложение Б** содержит подробное описание алгоритма AdaBoost, включающее формальную постановку задачи, математическое обоснование способа выбора очередного слабого классификатора, а также достаточные условия успешности построения сильного классификатора.

**Приложение В** содержит сканы охранных документов на результаты интеллектуальной деятельности, в которых применяются методы, изложенные в диссертации.

# 1 Метод Виолы и Джонса детектирования объектов на изображении

Метод Виолы и Джонса [7,8] представляет собой подход для решения задачи поиска объектов на изображении. Несмотря на то, что алгоритм, предложенный Полом Виолой и Майклом Джонсом в 2001 году, изначально разрабатывался для поиска лиц на изображениях в режиме реального времени, в настоящее время его различные модификации активно применяются в задачах поиска на изображениях пешеходов [19–22], транспортных средств [22–24], дорожных знаков [14,25] и множества других объектов с условно-фиксированной проекцией [12,13]. Здесь под словом «модификация» подразумевается тот факт, что при изменении постановки задачи приходится вносить изменения и в сам метод Виолы и Джонса (например, изменить пространство признаков или метод обучения). Такая популярность обусловлена тем фактом, что алгоритм показывает отличные результаты и высокую стабильность к геометрическим искажениям и яркостным изменениям.

Прежде, чем приступить к описанию самого метода, введем понятие цифрового изображения [44–48] и рассмотрим базовые основы задачи поиска объектов.

## 1.1 Понятие цифрового изображения

Несмотря на то, что понятие изображения интуитивно понятно, для дальнейшего описания необходимо дать определение более формально, а также ввести обозначения, которые будут использоваться на протяжении всей работы.

**Определение 1.** Под *изображением* будем понимать действительнозначную функцию  $f(y, x)$ , определенную на некотором подмножестве пространства  $\mathbb{R}^2$ . Значение функции  $f(y, x)$  называется интенсивностью или яркостью, реже уровнем серого.

**Определение 2.** Изображение  $f(y, x)$  будем называть *цифровым изображением*, если величины  $y$  и  $x$  принимают конечное число дискретных значений.

В силу того, что величины  $y$  и  $x$  принимают конечное число дискретных значений, цифровое изображение можно представить в виде матрицы из  $M$  строк и  $N$  столбцов, где  $(y, x)$  задают координаты, а  $f(y, x)$  – значение элемента матрицы. В дальнейшем для большего удобства будем использовать целочисленные значения  $y = \{0, 1, 2, \dots, M - 1\}$  и  $x = \{0, 1, 2, \dots, N - 1\}$ , принимая за начало координат левый верхний угол изображения, где  $(y, x) = (0, 0)$ , ось  $y$  направлена вниз, а ось  $x$  – вправо. Область действительной координатной плоскости, охватываемая координатами изображения, называется пространственной областью, а  $y$  и  $x$  – *пространственными переменными* или *пространственными координатами*.

Таким образом, цифровое изображение в виде матрицы можно записать следующим образом:

$$f(y, x) = \begin{pmatrix} f(0, 0) & \cdots & f(0, M - 1) \\ \vdots & \ddots & \vdots \\ f(N - 1, 0) & \cdots & f(N - 1, M - 1) \end{pmatrix}. \quad (1)$$

Каждый элемент матрицы (1) называются *элементами изображения* или *пикселями* (образовано от английского словосочетания picture element – «элемент изображения»).

В процессе получения цифрового изображения помимо определения величин  $M$  и  $N$  (пространственного размера изображения) возникает еще число  $L$  – количество дискретных уровней (градаций) яркости. Из соображений удобства и особенностей хранения цифровых изображений значение  $L$  обычно выбирают равным целочисленной степени двойки:  $L = 2^k$ . Предполагается, что дискретные уровни яркости расположены с постоянным шагом и принимают целые значения в интервале  $[0, L - 1]$ . Тогда для хранения цифрового изображения необходимо следующее количество бит:  $b = M \cdot N \cdot k$ .

Для вычислений или разработки алгоритмов бывает полезно отобразить шкалу  $L$  значений яркости на интервал  $[0, 1]$ , так что они перестают быть целочисленными. Однако при хранении и визуализации изображений в

большинстве случаев полученные значения масштабируются обратно на интервал целых значений  $[0, L - 1]$ .

Многоканальные изображения (например, цветные изображения RGB) размера  $M \times N$  могут быть представлены как совокупность изображений-компонент тех же размеров или путем представления пикселей в виде вектора:

$$\vec{f}(y, x) = \begin{pmatrix} f_1(y, x) \\ \vdots \\ f_n(y, x) \end{pmatrix}, \quad (2)$$

где  $f_i(x, y), i = \{1, 2, \dots, n\}$  – изображение  $i$ -ой компоненты,  $n$  – количество компонент многоканального изображения.

Далее в работе под *изображением в градациях серого* (или просто *серым изображением*) будет пониматься одноканальное цифровое изображение с  $2^8 = 256$  уровнями яркости. Под *цветным изображением* будет пониматься трехканальное цифровое изображение, сформированное в пространстве RGB (количество уровней яркости каждой цветовой компоненты составляет 256).

Введенные в данном разделе понятия, определения и обозначения являются фундаментом для дальнейшего описания алгоритмов.

## 1.2 Задача поиска объектов и основы статистического обучения

Задача поиска (детектирования) объектов (*object detection*) образно заключается в следующем: на каждом изображении необходимо выделить все искомые объекты. Естественно поэтому считать, что детектор должен возвращать список каких-то информационных структур, описывающих каждый конкретный экземпляр найденного объекта. Точное оконтуривание объекта требует сложных структур данных, да и, как правило, не требуется. Поэтому в системах компьютерного зрения при решении задач детектирования объектов обычно считается, что найденный объект однозначно задается обрамляющим прямоугольником со сторонами, выровненными по осям изображения. Часто для

каждого найденного экземпляра детектор помимо прямоугольника возвращает величину собственной уверенности (*confidence*) в ответе, что дает больше информации алгоритмам системы, использующих результат детектирования. Именно такой интерфейс используется, например, в рамках соревнования The PASCAL Visual Object Classes (VOC) Challenge [49].

В методе Виолы и Джонса [7] задача детектирования объектов на изображении сводится к задаче бинарной классификации в каждой точке изображения. Для каждой прямоугольной области изображения, взятой со всевозможными сдвигами и масштабами, при помощи заранее обученного классификатора (детектора) проверяется гипотеза о наличии в области искомого объекта. При таком подходе количество запусков классификатора на одном изображении исчисляется сотнями тысяч. Отсюда следуют два жестких требования на классификатор: высочайшая производительность и экстремально малое число ложных срабатываний.

### 1.2.1 Задача построения классификатора

Современные методы классификации практически без исключений опираются на механизмы обучения машин. При этом на практике чаще всего применяют статистические методы, когда решающее правило строится на основе обучающего набора прецедентов, для которых принадлежности к классам заведомо известны. Дадим формально описание задачи обучения классификатора (распознавателя) по прецедентам [32–34].

**Определение 3.** Пусть  $\mathbb{X}$  – множество исследуемых объектов. Будем называть *признаком* (*feature*) следующее отображение  $f: \mathbb{X} \rightarrow \mathbb{D}_f$ ,  $\mathbb{D}_f \subset \mathbb{R}$  – множество допустимых значений признака  $f$ .

На практике обычно множество  $\mathbb{D}_f$  представляет собой либо конечное множество (в этом случае признак называется *номинальным*), либо совпадает с множеством действительных значений (в этом случае признак называется *количественным*).

**Определение 4.** Пусть имеется набор признаков  $f_1, \dots, f_n$ . Тогда вектор  $(f_1(x), \dots, f_n(x))$  называется признаковым описанием объекта  $x \subseteq \mathbb{X}$ .

Совокупность признаковых описаний всех объектов подвыборки  $\mathbb{X}^l = \{x_1 \in \mathbb{X}, \dots, x_l \in \mathbb{X}\}$ , записанную в виде матрицы, называют *матрицей объектов-признаков*:

$$F = \begin{pmatrix} f_1(x_1) & \cdots & f_1(x_l) \\ \vdots & \ddots & \vdots \\ f_n(x_1) & \cdots & f_n(x_l) \end{pmatrix}. \quad (3)$$

Матрица объектов-признаков является наиболее распространённым способом представления исходных данных в прикладных задачах обучения машин.

Далее будем считать, что множество  $\mathbb{X}$  разбивается на  $M$  непересекающихся классов. Введем множество  $\mathbb{Y} = \{1, \dots, M\}$  – множество индексов классов. Тогда существует отображение (которое называется *целевая функция*)  $y^*: \mathbb{X} \rightarrow \mathbb{Y}$ .

Пусть есть подмножество  $\mathbb{X}^l = \{x_1 \in \mathbb{X}, \dots, x_l \in \mathbb{X}\}$ , для которого известны значения целевой функции  $y_i = y^*(x_i)$ . Задача обучения по прецедентам заключается в том, чтобы по выборке  $\mathbb{X}^l$  и соответствующим значениям целевой функции построить решающую функцию  $a: \mathbb{X} \rightarrow \mathbb{Y}$ , которая приближала бы целевую функцию  $y^*(x)$  на всём множестве  $\mathbb{X}$  наилучшим образом.

**Определение 5.** Пары  $(x_i, y_i)_{i=1}^l$  называются *прецедентами*. Совокупность всех прецедентов называется *обучающей выборкой*.

Так как по смыслу решающая функция  $a(x)$  должна давать ответ на вопрос «к какому классу принадлежит объект  $x$ », то будем называть ее *классификатором*.

Как было сказано выше, при постановке задачи обучения по прецедентам необходимо найти наилучшую решающую функцию  $a(x)$ . При формализации понятия «наилучшая» большинством упомянутых во Введении научных школ вводится в рассмотрение функционал качества классификации. Тогда наилучшей считается функция, которая доставляет оптимальное значение выбранному функционалу качества.

В диссертации используется функционал качества, предложенный в работах В.Н. Вапника, А.Я. Червоненкиса (см., например, [50,51]).

**Определение 6.** Будем называть *функцией потерь* (*loss function*) неотрицательную функцию  $L(a, x)$ , характеризующую величину ошибки решающей функции  $a$  на объекте  $x$ . Если  $L(a, x) = 0$ , то ответ  $a(x)$  называется *корректным (достоверным)*.

**Определение 7.** *Функционал качества* решающей функции  $a$  на выборке  $\mathbb{X}^l$  определяется следующим образом:

$$Q(a, \mathbb{X}^l) = \frac{1}{l} \sum_{i=1}^l L(a, x_i). \quad (4)$$

Функционал  $Q(a, \mathbb{X}^l)$  также называют функционалом средних потерь или эмпирическим риском [52], так как он вычисляется по эмпирическим данным.

Что касается функции потерь, то в задачах классификации на практике наиболее часто используется функция-индикатор ошибки:

$$L(a, x) = [a(x) \neq y^*(x)] = \begin{cases} 1 & a(x) \neq y^*(x) \\ 0 & a(x) = y^*(x) \end{cases}. \quad (5)$$

## 1.2.2 Бустинг в задачах классификации

При решении сложных задач классификации часто оказывается, что ни одна из имеющихся решающих функций не обеспечивает желаемого качества. Вместо того, чтобы из получившихся «слабых» классификаторов (weak classifiers) выбирать наименее плохой, можно попробовать их «усилить», ища самый лучший классификатор не только среди имеющихся, но и среди функций от них. Чаще всего используются линейные комбинации классификаторов. Так появился термин *бустинг (boosting)*.

В середине 90-х годов в работах [4,5] был предложен удачный итеративный алгоритм AdaBoost (Adaptive Boosting), обучающий слабые классификаторы постепенно, с учетом результатов голосования уже обученных классификаторов. Подробное описание алгоритма AdaBoost, включающее формальную постановку задачи, математическое обоснование способа выбора очередного слабого классификатора, а также достаточные условия успешности построения сильного классификатора приведено в приложении Б.

С момента своего появления и до настоящего времени бустинг остаётся одним из наиболее популярных методов обучения машин, наряду с нейронными сетями и машинами опорных векторов (Support Vector Machine, SVN). Причинами этому служат простота, универсальность, гибкость (возможность построения различных модификаций), а также высокая обобщающая способность.

### 1.2.3 Оценка качества работы детекторов

После обучения детектора объектов законно возникает задача оценить качество его работы (распознавания). Для этого обученный детектор проверяют на независимой *валидационной выборке*, для которой известны правильные ответы. Такую выборку в зарубежной литературе называют *ground truth*, в отечественной литературе часто встречается термин *идеал*.

С помощью валидационной выборки можно посчитать следующие величины:

- *TP (True Positive)* – число верно найденных объектов;
- *TN (True Negative)* – число верно распознанных пустых областей;
- *FP (False Positive)* – число ложно найденных объектов;
- *FN (False Negative)* – число ложно распознанных пустых областей.

Заметим, что справедливы следующие равенства:

- $P = TP + FN$  – истинное число объектов на валидационной выборке;
- $N = TN + FP$  – истинное число пустых областей на валидационной выборке.

На основании вычисленных величин можно посчитать следующие статистики: доля верного распознанных объектов  $TPR = \frac{TP}{P} = \frac{TP}{TP+FP}$  и доля ложно распознанных объектов  $FPR = \frac{FP}{N} = \frac{FP}{TN+FP}$ . Заметим, что указанные статистики не являются независимыми величинами. Большинство алгоритмов распознавания обладает настроечными коэффициентами, с помощью которых можно улучшать одну статистику (например,  $TPR$ ), ухудшая при этом вторую (в нашем случае  $FPR$ ). Поэтому для анализа качества работы детектора используют так называемую ROC-кривую (*receiver operating characteristic*, рабочая характеристика приёмника), которая представляет собой график зависимости доли верно распознанных объектов от доли ложно распознанных объектов. Пример ROC-кривой для двух детекторов представлен на рисунке 1.

Количественную интерпретацию ROC-кривой дает площадь, ограниченная ROC-кривой и осью абсцисс, которая в литературе называется AUC (area under ROC curve, площадь под ROC-кривой) [53]. Чем выше показатель AUC, тем качественнее детектор, при этом значение 0,5 соответствует случайному гаданию.

При решении практических задач в части детектирования объектов бывает затруднительно оценить статистику  $FPR$  из-за сложности подсчета общего числа

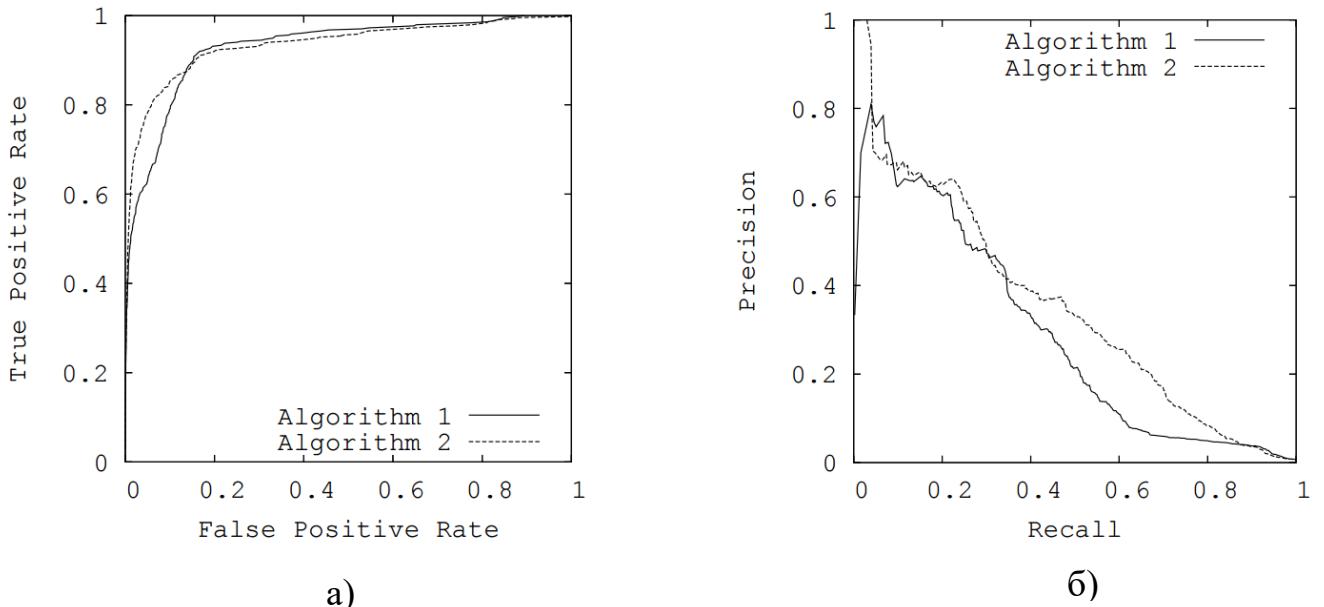


Рисунок 1 — Пример а) ROC-кривой и б) Precision-Recall кривой  
(источник [54])

пустых областей. В таком случае прибегают к Precision-Recall анализу [54], в соответствии с которым вычисляются точность  $Precision = \frac{TP}{TP+FP}$  и полнота  $Recall = \frac{TP}{TP+FN}$ . Для сравнения качества работы нескольких детекторов строится Precision-Recall кривая (см. рисунок 1), для которой аналогичны образом исследуется площадь под кривой.

**Определение 8.** Будем говорить, что детектор  $a_1$  доминирует над детектором  $a_2$  в смысле ROC анализа, если ROC-кривая детектора  $a_1$  лежит не ниже ROC-кривой детектора  $a_2$ .

Аналогичным образом вводится определение доминирования в смысле Precision-Recall анализа. Справедлива следующая теорема [54].

**Теорема 1.** Детектор доминирует в ROC анализе тогда и только тогда, когда он доминирует в Precision-Recall анализе.

**Доказательство.** Рассмотрим два детектора  $a_1$  и  $a_2$ . Предположим обратное. Пусть для определенности детектор  $a_1$  доминирует над детектором  $a_2$  в смысле ROC анализа, но не доминирует при этом в смысле Precision-Recall анализа. Это значит, что существует такое значение  $Recall$  (обозначим его  $r^*$ ), для которого значение точности детектора  $a_1$  будет меньше значения точности детектора  $a_2$ , то есть  $Precision_{a_1}(r^*) < Precision_{a_2}(r^*)$ . Исходя из того факта, что детектор  $a_1$  доминирует над  $a_2$  в терминах ROC анализа и что  $TPR \equiv Recall$ , получаем, что должно быть справедливо неравенство  $FPR_{a_1}(r^*) \leq FPR_{a_2}(r^*)$ . Из данного неравенства и определения  $FPR$  следует, что  $FP_{a_1}(r^*) \leq FP_{a_2}(r^*)$ . Но в таком случае  $Precision_{a_1}(r^*) = \frac{TP}{FP_{a_1}(r^*)+TP} \geq \frac{TP}{FP_{a_2}(r^*)+TP} = Precision_{a_2}(r^*)$ , что противоречит предположению. Теорема доказана.

Доказанная выше теорема 1 существенно упрощает процедуру сравнения качества работы детекторов. Тем не менее до сих пор не было обозначено как считать количество верно распознанных объектов. Удачное с практической точки зрения решение было предложено в рамках соревнования The PASCAL Visual Object Classes (VOC) Challenge [49].

**Определение 9.** Будем говорить, что детектор  $a$  верно распознал объект  $x$ , если доля перекрытия (*overlap ratio*) найденного обрамляющего прямоугольника  $B_a(x)$  и идеального обрамляющего прямоугольника  $B_{gt}(x)$  превышает 0.5, то есть:

$$\text{overlap ratio} = \frac{\text{area}(B_a \cap B_{gt})}{\text{area}(B_a \cup B_{gt})} \geq 0.5 \quad (6)$$

Такое низкое значение порога для доли перекрытия было установлено намеренно, чтобы объяснить неточности в идеальных валидационных выборках, которые готовятся ручным образом. Дело в том, что понятие обрамляющего прямоугольника слишком субъективно для сильно невыпуклого объекта (например, такого, как человек). Однако значение порога 0.5 позволяет покрыть такие неточности.

### 1.3 Оригинальный методы Виолы и Джонса

Метод Виолы и Джонса представляет из себя схему построения детекторов объектов с достаточно жесткой геометрией статистическим образом (опираясь на прецеденты).

В своей основополагающей работе авторы описывают построение детектора человеческого лица анфас. Обучающая выборка состояла из 4916 изображений лиц, приведенных к разрешению  $24 \times 24$  пикселя (положительная выборка) и 9500 полноразмерных изображений, не содержащих лиц, из которых произвольным образом вырезались области, не содержащие лица (отрицательная выборка).

Для построения признакового пространства использовались признаки Хаара, которые хорошо выделяют характерные особенности объектов, связанные с перепадами яркости, а также могут быть легко вычислены с помощью интегрального представления изображения. При этом оказалось, что с изображением размером  $24 \times 24$  пикселей связано множество, содержащее 45 396 признаков, которое заведомо является переполненным.

В своем подходе Виола и Джонс применили алгоритм AdaBoost, который одновременно позволил выбрать наиболее эффективные признаки и построить бинарный классификатор. Для этого с каждым признаком авторы связали слабый классификатор, представляющий одноуровневое дерево решений. Такие слабые классификаторы впоследствии подаются на вход алгоритму AdaBoost.

Последней важной частью метода Виолы и Джонса является способ объединения набора сильных классификаторов в каскадную структуру, которая позволяет резко увеличить производительность путем сосредоточения внимания на регионах исследуемого изображения, потенциально содержащие лица.

В следующих подразделах мы рассмотрим каждую составляющую часть данного метода более подробно.

### 1.3.1 Прямоугольные признаки Хаара

В алгоритме Виолы и Джонса при обучении классификаторов для описания объектов используется семейство признаков Хаара [55]. Такой выбор обусловлен в первую очередь тем фактом, что признаки Хаара позволяют описать характерные особенности объектов, связанные с перепадами яркости. Например, с помощью признаков Хаара легко отразить факт, что на изображении лица человека область глаз темнее области носа (рисунке 2г). Если более конкретно, то в описываемом

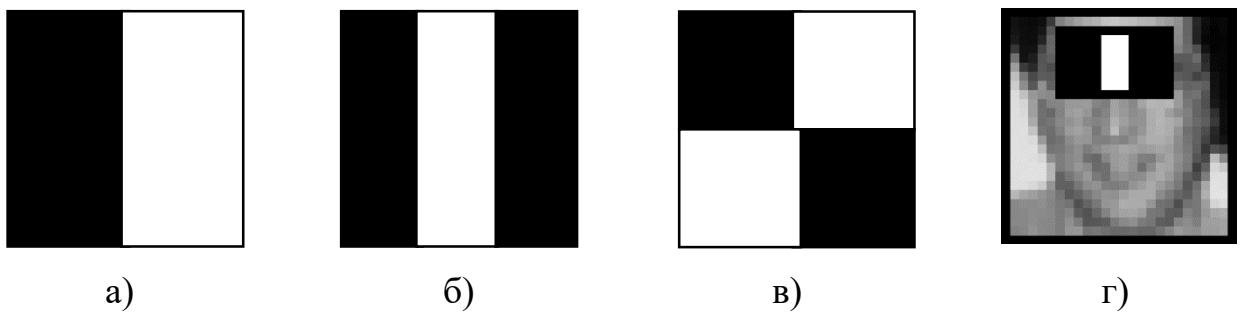


Рисунок 2 — Прямоугольные признаки Хаара, используемые при обучении классификатора: а) 2-прямоугольный признак; б) 3-прямоугольный признак; в) 4-прямоугольный признак; г) пример расположение признака относительно сканирующего окна (источник [7])

подходе используется три типа признаков: 2-прямоугольные признаки (рисунок 2а), 3-прямоугольные признаки (рисунок 2б) и 4-прямоугольные признаки (рисунок 2в).

Значение признаков Хаара вычисляется как разность сумм пикселей областей изображения внутри черных и белых прямоугольников равного размера. 2-прямоугольные и 3-прямоугольные признаки Хаара могут иметь как вертикальную (представленную на рисунке 2), так и горизонтальную ориентации.

Для эффективно вычисления признаков Хаара используется интегральное представление изображения [56].

**Определение 10.** Пусть есть серое изображение  $f(y, x)$  с размерами  $M \times N$ . Под интегральным представлением  $I_f(y, x)$  изображения  $f(y, x)$  будем понимать цифровое изображение размером  $(M + 1) \times (N + 1)$ , значение пикселей которого вычисляется следующим образом:

$$I_f(y, x) = \begin{cases} 0, & \text{при } y = 0 \text{ или } x = 0 \\ \sum_{y' < y, x' < x} f(y', x'), & \text{при } x > 0 \text{ и } y > 0. \end{cases} \quad (7)$$

Интегральное представление изображения может быть построено за один проход с помощью следующего рекурсивного метода:

1. Положим значение пикселей первой строки и первого столбца интегрального представления изображения равным нулю, то есть  $I_f(0, x) = 0$  и  $I_f(y, 0) = 0$ .

2. Для остальных пикселей  $y > 0$  и  $x > 0$  вычислим значение по рекурсивной формуле  $I_f(y, x) = I_f(y - 1, x) + I_f(y, x - 1) - I_f(y - 1, x - 1) + f(y - 1, x - 1)$ .

Используя интегральное представление изображения сумму пикселей изображения внутри прямоугольной области можно выполнить за четыре арифметические операции вне зависимости от размера прямоугольного изображения с помощью следующей формулы:

$$\sum_{\substack{y_1 \leq y < y_2 \\ x_1 \leq x < x_2}} f(y, x) = I_f(y_2, x_2) - I_f(y_1, x_2) - I_f(y_2, x_1) + I_f(y_1, x_1). \quad (8)$$

Таким образом, используя интегральное представление изображения, можно вычислить признаки Хаара за несколько арифметических операций вне зависимости от размера прямоугольных областей.

На сегодняшний день интегральное представление изображения широко применяется и других областях обработки цифровых изображений, например, при расчёте дескрипторов особых точек или вычислении вейвлет-преобразований.

### 1.3.2 Обучение классификаторов с помощью AdaBoost

В методе Виолы и Джонса в качестве обучающего алгоритма используется алгоритм AdaBoost, описанный в разделе 1.2.2 и приложении Б. При этом в качестве слабых классификаторов применяются распознающие деревья с одним ветвлением и двумя листьями (которые часто в литературе называются *decision stump* [57], которые сравнивают значение соответствующего признака с пороговым значением. Формально слабый классификатор  $b_j(x)$ , построенный поверх признака  $f_j$  с порогом  $\theta_j$  и четностью  $p_j$  выглядит следующим образом:

$$b_j(x) = \begin{cases} +1, & \text{если } p_j f_j(x) < p_j \theta_j \\ -1, & \text{если } p_j f_j(x) \geq p_j \theta_j \end{cases} \quad (9)$$

Построенные таким образом классификаторы формируют множество слабых классификаторов  $\mathbb{B}$ , которое вместе с обучающей выборкой (описанной в разделе 1.3) подается на вход алгоритму AdaBoost.

Используя описанную процедуру, авторы построили сильный классификатор, содержащий 200 слабых классификаторов. Полученный классификатор показывал неплохое качество классификации (доля верного обнаружения  $TPR \approx 0.95$ ) при низком уровне ошибок (доля ложного обнаружения

$FPR \approx 7 \cdot 10^{-5}$ ). Однако, производительность такого классификатора составляла 0.7 секунд на изображение размером  $384 \times 288$ . Хотя такая производительность сравнима с производительностью других алгоритмов того времени [58–61], она являлась недостаточной для решения задачи детектирования объектов в режиме реального времени.

В следующем разделе будет описана каскадная структура классификатора, которая позволяет резко увеличить производительность путем сосредоточения внимания на регионах исследуемого изображения, потенциально содержащих лица.

### 1.3.3 Каскадная структура классификаторов

Каскадная структура классификаторов (далее в работе будем также использовать название *каскадный классификатор* или *каскад*) разработана исходя из следующего эвристического факта: для любого исследуемого изображения количество «отрицательных» (не содержащих искомый объект) областей на несколько порядков превосходит количество «положительных» (содержащих искомый объект) областей. Следовательно, для улучшения быстродействия детектора объектов необходимо как можно раньше «отсекать» отрицательные области изображения. Другими словами, количество вычисленных признаков для анализа отрицательных областей должно быть значительно меньше, чем количество признаков, вычисленных для положительных областей.

Для решения поставленной задачи Виола и Джонс использовали каскадную структуру классификаторов. Принцип каскадного классификатора изображен на рисунке 3. Каждая исследуемая на предмет наличия искомого объекта область изображения подается на вход упорядоченной последовательности классификаторов (на рисунке 3 представлен каскад из трех классификаторов). Будем называть эти классификаторами *уровнями* (*level*) или *стадиями* (*stage*) каскада. Вычислительная сложность уровней увеличивается по мере продвижения вглубь каскада. Если на очередном уровне анализа классификатор откажется от

распознавания исследуемой области (признает ее отрицательной), то данная область уже не анализируется оставшимися классификаторами каскада.

Опыты авторов показывают, что если использовать в качестве первого уровня каскада сильный классификатор, обученный с помощью алгоритма AdaBoost и содержащий всего два слабых классификатора, то около 40% отрицательных областей будет верно отсечено уже на первом уровне.

Объективно возникает вопрос о способе обучения каскадного классификатора. Рассмотрим способ, описанный Виолой и Джонсом в своей основополагающей работе [7].

Заметим, что для каскадного классификатора справедливы следующие соотношения:

$$FPR = \prod_{i=1}^K fpr_i, \quad TPR = \prod_{i=1}^K tpr_i, \quad (10)$$

где  $FPR$  – доля ложного обнаружения каскада,  $TPR$  – доля верного обнаружения каскада,  $K$  – количество уровней каскада,  $fpr_i$  – доля ложного обнаружения  $i$ -го уровня каскада,  $tpr_i$  – доля верного обнаружения  $i$ -го уровня каскада.

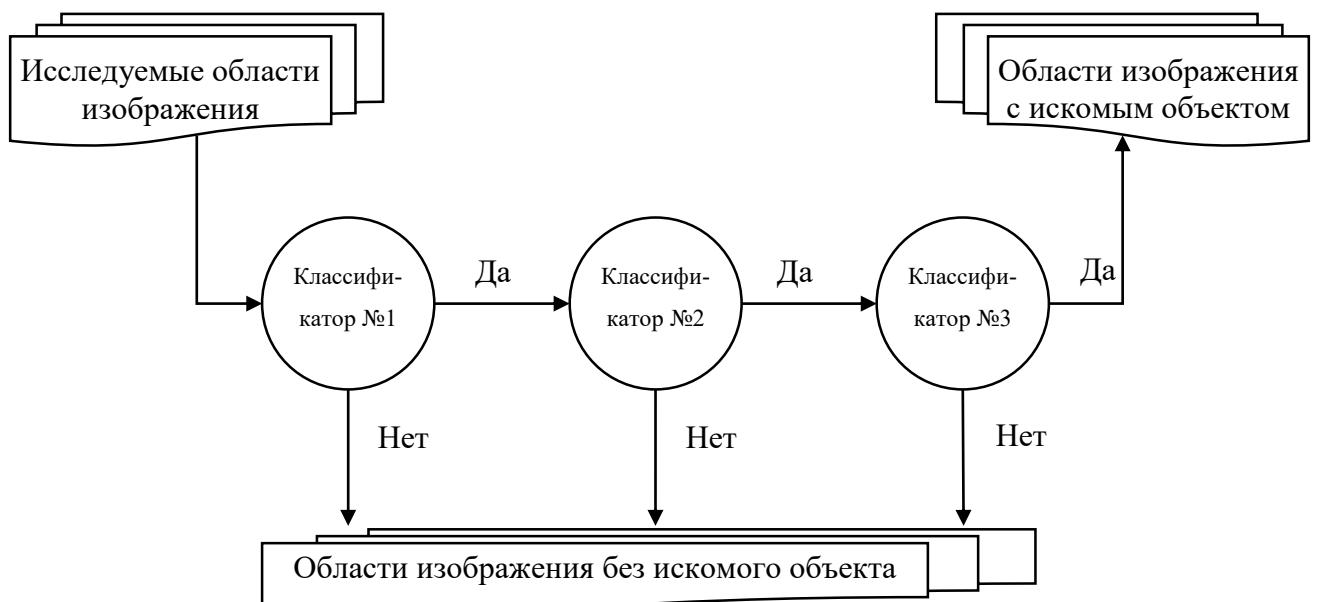


Рисунок 3 — Схематическое описание каскадной структуры

Ожидаемое количество вычисляемых признаков, которые будут вычислены для каждой исследуемой области вычисляется по формуле:

$$N = n_1 + \sum_{i=2}^K \left( n_i \prod_{j=1}^{i-1} pr_j \right), \quad (11)$$

где  $n_i$  – количество слабых классификаторов  $i$ -го уровня,  $pr_i$  – доля срабатываний (positive rate)  $i$ -го уровня.

В определенных таким образом терминах алгоритм обучения классификатора должен решать оптимизационную задачу, стремясь построить каскад с наилучшими показателями  $FPR$  и  $TPR$ , минимизируя при этом ожидаемое количество вычисляемых признаков  $N$ . Однако решить такую задачу (равно как и строго сформулировать) чрезвычайно сложно в связи с отсутствием явной связи между величинами  $FPR$ ,  $TPR$  и  $N$ .

В своих работах авторы предлагают следующую простую идею обучения каскада. Пусть есть обучающая выборка, состоящая из положительных и отрицательных примеров. Заранее указывается три параметра: доля ложных срабатываний ( $fpr^*$ ), доля верных срабатываний для каждого уровня каскада ( $tpr^*$ ), а так же доля ложных срабатываний для всего каскада в целом ( $FPR^*$ ). Тогда каскадный классификатор строится итерационно, где каждая стадия обучается с помощью алгоритма AdaBoost, описанного в разделе 1.2.2 и приложении Б. Уровни каскада добавляются до тех пор, пока не будет достигнуто целевое качество, заданное величиной  $FPR^*$ . Важно отметить, что на вход обучения очередного уровня каскада подается фильтрованная обучающая выборка, в которой отрицательные примеры представляют собой ложные срабатывания уже обученного на текущий момент каскадного классификатора. Блок-схема описанного алгоритма представлена на рисунке 4.

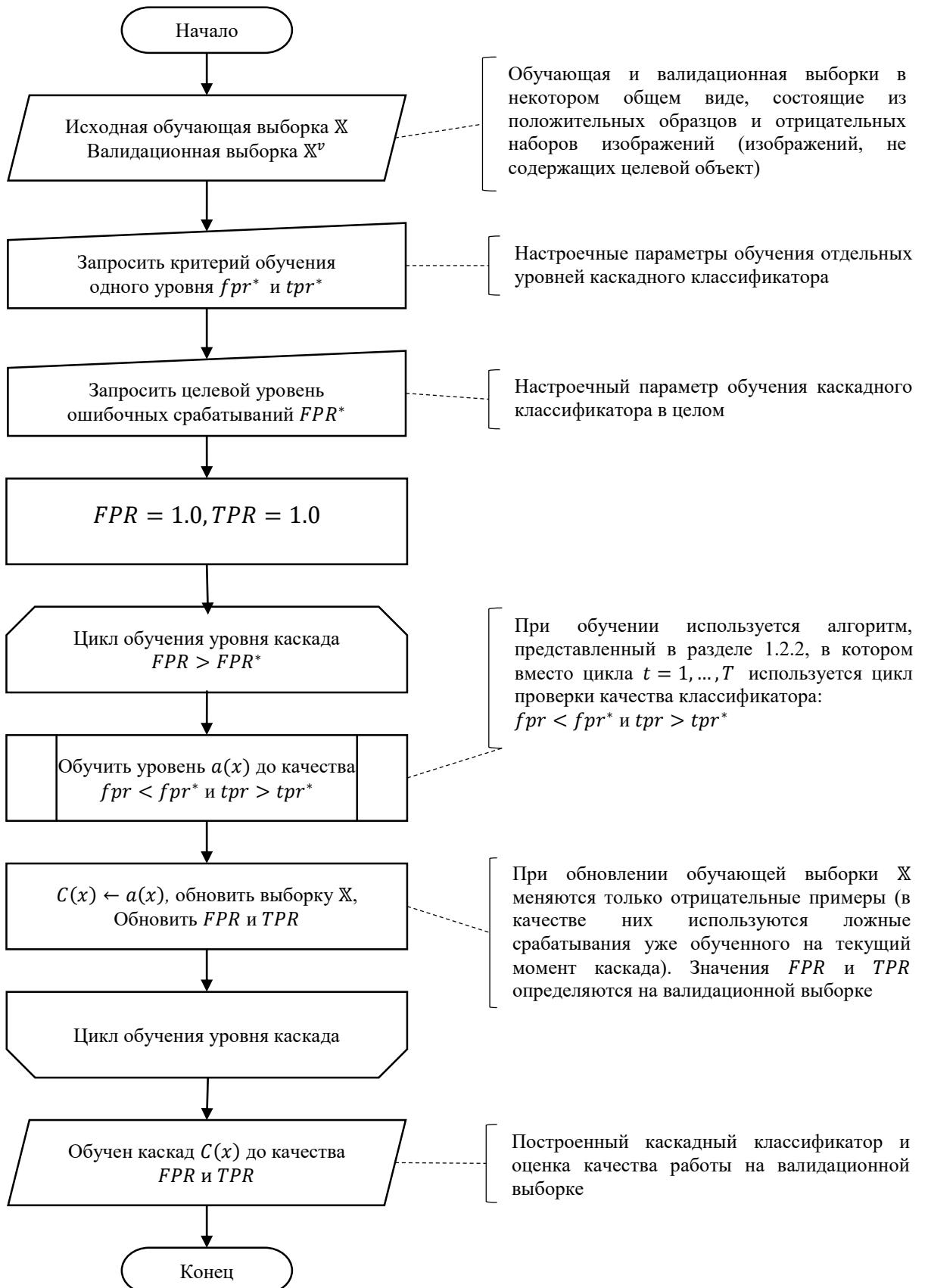


Рисунок 4 — Блок-схема обучения каскадной структуры классификатора

### 1.3.4 Результаты практического тестирования метода

Описанный подход к обучению детектора объектов был протестирован в задаче детектирования лиц. В качестве обучающей выборки использовался набор из 4916 вырезанных лиц и 9500 изображений, не содержащих человеческих лиц.

В результате был обучен каскадный классификатор, состоящий из 32 уровней и содержащий в общей сложности 4297 слабых классификаторов. Обученный каскадный классификатор применялся для детектирования объектов путем «сканирования» изображения при различных масштабах. При этом в методе масштабируется не изображение, а классификатор с некоторым шагом. Шаг сканирования также зависит от текущей величины классификатора, то есть для

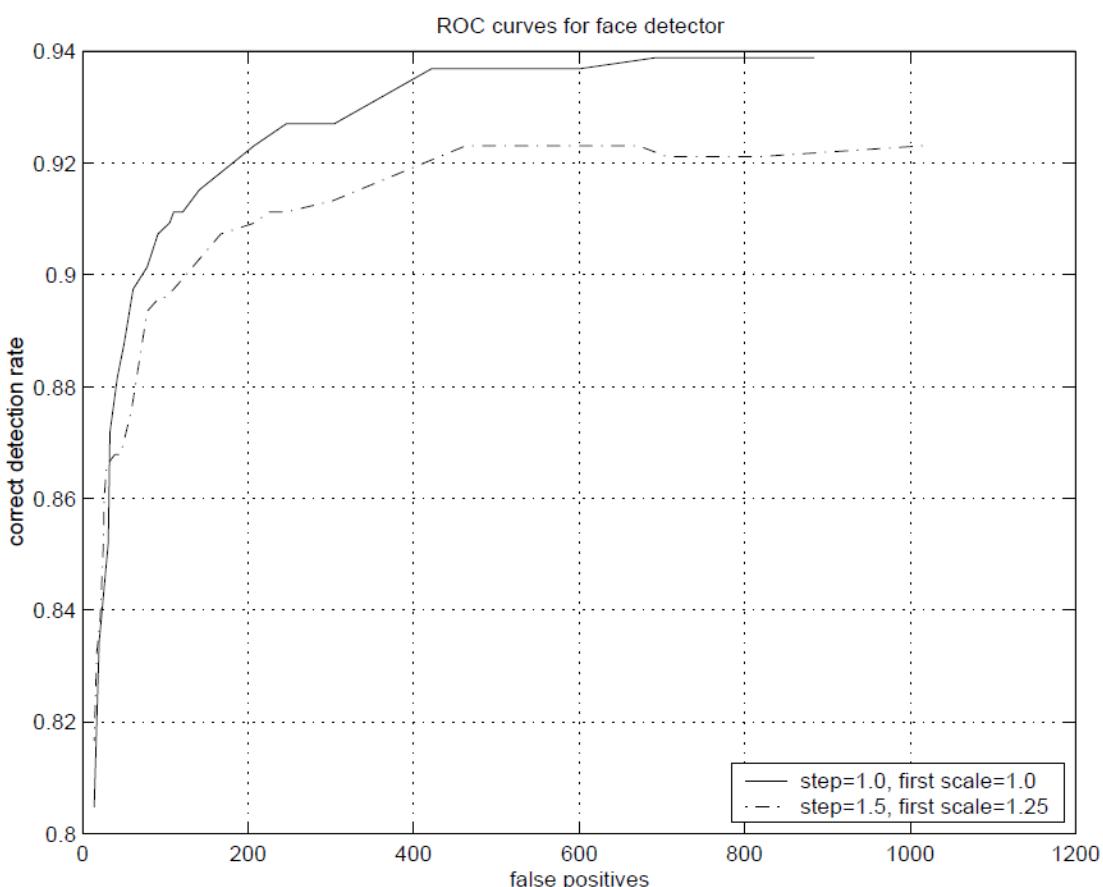


Рисунок 5 — ROC-кривая оригинального метода Виолы и Джонса, посчитанная при различных параметрах на валидационном наборе MIT+CMU (источник [7])

некоторого масштаба  $s$  шаг сканирования вычисляется как  $\Delta s$ , где  $\Delta$  – наперед заданный коэффициент сканирования.

Задание начального значения масштаба классификатора, шага масштабирования и коэффициента сканирования позволяет регулировать производительность и качество работы детектора. На рисунке 5 представлены ROC-кривые, посчитанные при тестировании каскадного детектора на валидационном наборе MIT+CMU [61] для двух вариантов параметров (с тем отличием, что вместо  $FPR$  используется  $FP$ ). В первом случае ( $step = 1.0, first scale = 1.0, \Delta = 1.0$ ) количество просканированных областей на одном изображении составило 75 081 800, во втором случае ( $step = 1.5, first scale = 1.25, \Delta = 1.0$ ) – 18 901 947 областей.

## **1.4 Анализ существующих модификаций метода Виолы и Джонса**

Как было сказано в разделе 1.3 подход Виолы и Джонса состоит из следующих основополагающих частей: пространство признаков Хаара, обучающий алгоритм AdaBoost и каскадная структура классификатора. Соответственно, развитие оригинального подхода заключается в улучшении одной или нескольких составляющих частей. В данном подразделе будут рассмотрены основные работы, представляющие собой развитие оригинального алгоритма.

### **1.4.1 Модификации в части пространства признаков**

Пространство прямоугольных признаков Хаара хорошо описывает характерные особенности объектов, связанные с перепадами яркости вдоль вертикального и горизонтального направления. Однако существуют задачи, в которых объект обладает большим количеством перепадов вдоль диагональных направлений. Для решения такого рода задач Линхарт и Майд предложили расширить пространство признаков путем добавления наклоненных под 45 градусов признаков Хаара. На рисунке 6 проиллюстрированы все типы признаков, используемые авторами в работе [26].

Для быстрого вычисления новых признаков Линхарт и Майд предложили использовать дополнительно наклоненное интегральное представление изображения (Rotated Summed Area Table), которое определяется следующим образом:

$$I_f^r(y, x) = \begin{cases} 0, & \text{при } y = 0 \text{ или } x = 0 \\ \sum_{\substack{x' < x \\ x' < x - |y - y'|}} f(y', x'), & \text{при } x > 0 \text{ и } y > 0. \end{cases} \quad (12)$$

Утверждается, что наклонное интегральное представление изображения может быть быстро посчитано за два прохода по пикселям исходного изображения. Благодаря использованию  $I_f^r(y, x)$  сумма пикселей внутри наклонного прямоугольника может быть вычислена за 4 арифметические операции вне зависимости от размера прямоугольника.

Предложенные Линхартом и Майдом наклонные признаки Хаара на сегодняшний день широко используются при обучении детекторов разных по своей природе объектов [26,62,63].

Позже Мессом и Барзак дополнили концепцию наклонных признаков Хаара, предложив подход вычислять признаки под произвольными углами [27]. Хотя идея верна с математической точки зрения, на практике при использовании признаков под разными углами возникают проблемы. Для ускорения вычислений, детектор использует изображения низкого разрешения, что приводит к ошибке округления.

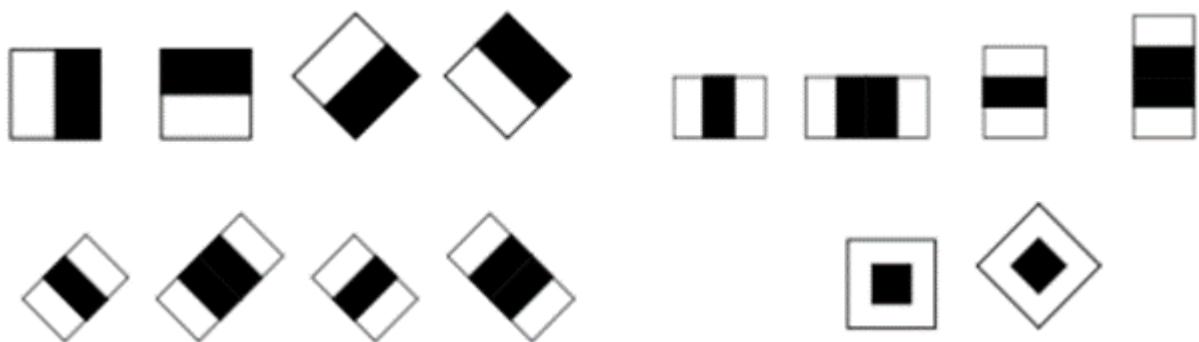


Рисунок 6 — Расширенное пространство признаков Хаара

Кроме того, благодаря высокой устойчивости к геометрическим искажениям, вычисленные под точными углами признаки не приносят существенного прироста качества. Исходя из этого, наклонные признаки Хаара обычно не используются.

Дальнейшим развитием классических признаков Хаара стали так называемые полигональные признаки, впервые предложенные в 2010 году на международной конференции компьютерного зрения и распознавания образов CVPR'2010 [28]. Исследователи из института в Гилфорде вместе с коллегами из Университета в Сингапуре обобщили метод вычисления прямолинейного многоугольного интеграла изображения на случай многоугольных интегралов общего вида, продемонстрировав в своей работе способ представления произвольного многоугольного интеграла в виде взвешенной суммы предварительно вычисленных правых треугольных интегралов.

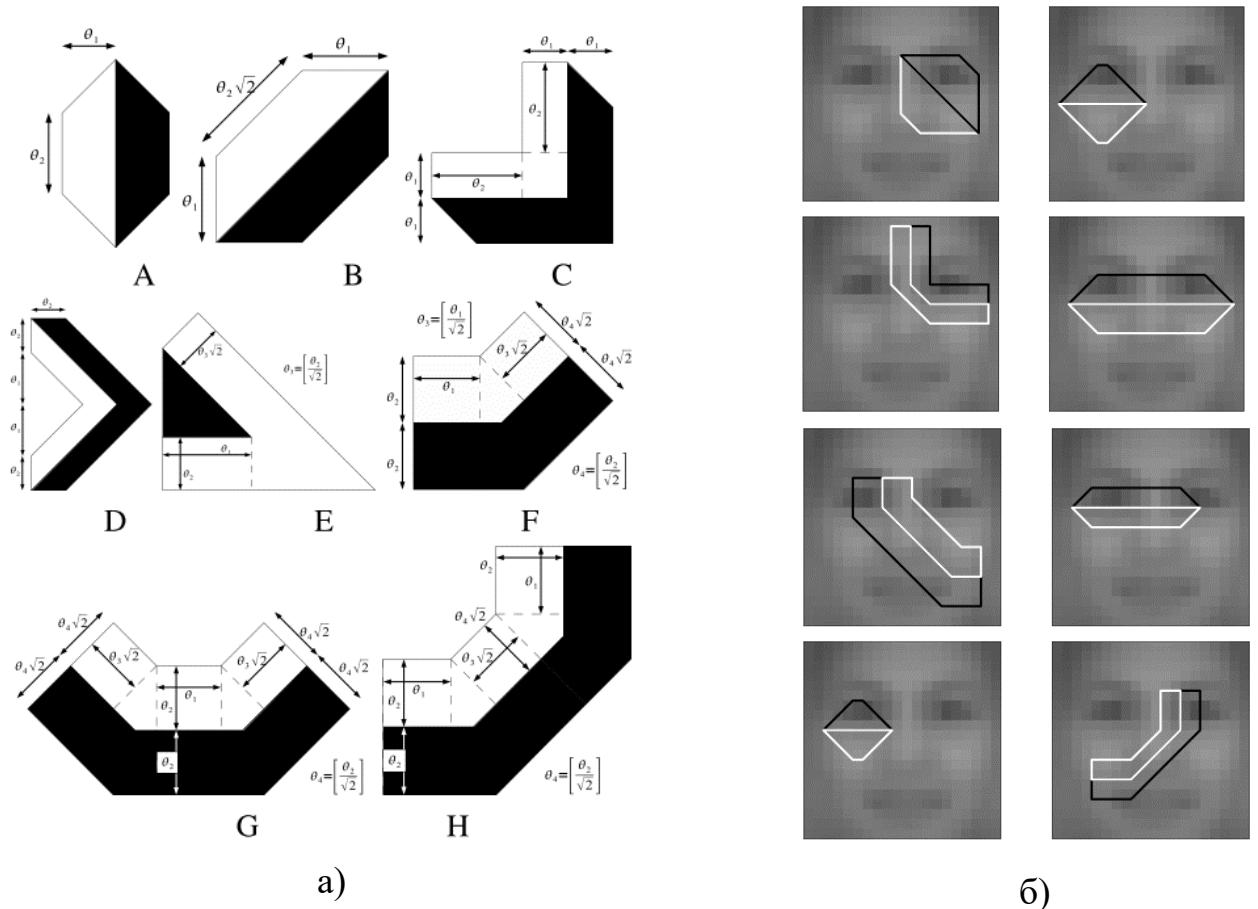


Рисунок 7 — Полигональные признаки Хаара: а) варианты представления признаков; б) примеры применимости признаков для задачи поиска лиц

Полученный метод вычисления интегралов ученые применили для построения полигональных признаков Хаара (см. рисунок 7) и расширения за счет них общего пула признаков в алгоритме Виолы и Джонса, в результате чего удалось обучить детектор лица, обеспечивающий лучшие показатели как по качеству, так и по скорости детектирования.

Несмотря на описанные преимущества признаков Хаара, они обладают одним существенным недостатком – будучи посчитаны поверх яркостного изображения признаки не устойчивы к изменению освещенности. Конечно, исследователи предлагают различные пути обхода данной проблемы (например, Виола и Джонс предложили использовать предварительную нормализацию каждого исследуемого региона перед подачей его на распознавание), однако принципиального решения у данной проблемы нет.

Одним из видов признаков, устойчивыми к изменению яркости являются признаки, построенные с помощью локальных бинарных шаблонов (Local Binary Patterns, LPB) [64–66]. Первое упоминание данного вида признаков было в 1996 году [64], однако в задачах распознавания образов они начали использоваться сравнительно недавно.

Оператор LPB представляет каждый пиксель изображения в виде бинарного числа, зависящего от интенсивностей соседних пикселей изображения. Классический LBP использует восемь пикселей окрестности (см. рисунок 8а). Он рассматривает центральный пиксель в качестве порога и сравнивает в каждом соседнем пикселе окрестности значение яркости с ним. Если соседнее значение

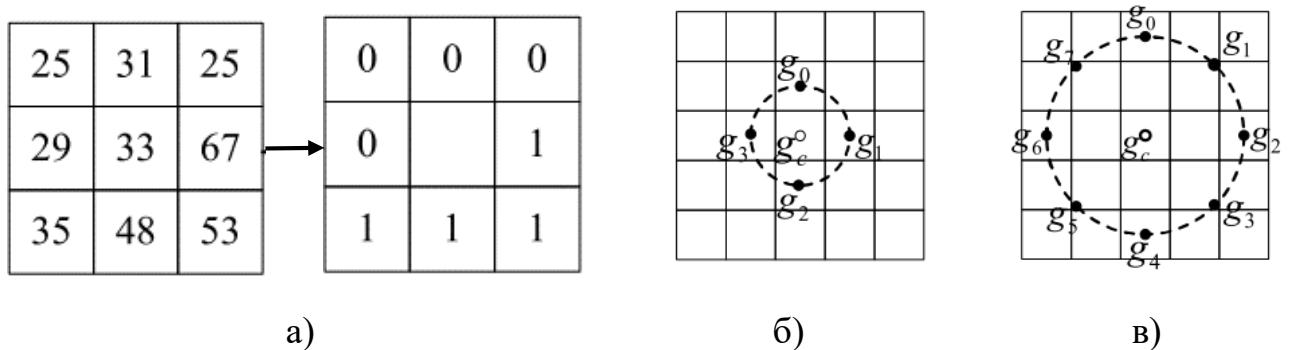


Рисунок 8 — Иллюстрация LBP: а) классический оператор LBP; б) вид расширенного оператора LBP; в) вид расширенного оператора LBP

яркости не меньше яркости центрального пикселя, то соседний пиксель принимает значение 1, иначе — 0. Полученное восьми битное число характеризует окрестность центрального пикселя и может быть закодировано с помощью одного байта.

Позже в 2002 году в работе [67] был предложен расширенный вид оператора LBP, позволяющий вычислять его по круглой окрестности произвольного радиуса, а не только по квадратному окну  $3 \times 3$  (см. рисунок 8б и рисунок 8в).

Для того, чтобы применить идею LBP в задаче детектирования объектов, была предложена идея строить признаки в виде гистограммы по окну изображения, к которому применен LBP (Local Binary Patterns Histogram, LBPH) [65]. Поверх таких признаков строится множество слабых классификаторов (в самом простом случае путем сравнения значения отдельных ячеек гистограммы с порогом [68]) с дальнейшим обучением финального детектора с помощью Boosting-метода [65,68].

Третий класс признаков, также устойчивый к изменению яркости, активно применяемый в настоящий момент для детектирования объектов на изображении – признаки, построенные на базе ориентированных градиентов (Oriented Gradients). Впервые эта идея была применена в работе [10] для построения классификатора пешеходов поверх гистограмм ориентированных градиентов (Histogram of Oriented Gradients, HOG) [10,69]. Идея применения данного класса признаков основывается на том допущении, что форма объекта на изображении может быть описана направлением краев и распределением градиентов интенсивности (см. рисунок 9). Следовательно, если разбить область изображения с исследуемым объектом на небольшие участки и для каждого такого участка построить гистограмму направленных градиентов, то комбинация полученных гистограмм будет характеризовать исследуемый объект.

При работе с HOG признаками в самом начале вычисляется градиент изображения. Градиент изображения представляет собой цифровое изображение, каждый пиксель которого отражает направление наискорейшего возрастания яркости оригинального изображения. Так как цифровое изображение дискретно и

не является дифференцируемым в классическом смысле, применяются различные разностные алгоритмы:

$$\begin{aligned}
 G^I(y, x) &= (G_y^I, G_x^I), \\
 G_y^I(y, x) &= \frac{I(y+1, x) - I(y-1, x)}{2}, \\
 G_x^I(y, x) &= \frac{I(y, x+1) - I(y, x-1)}{2}.
 \end{aligned} \tag{13}$$

Гистограммы ориентированных градиентов в описанном виде отлично отражают краевые особенности объектов. Однако, в отличие от признаков Хаара, HOG не способны подчеркнуть пространственные изменения исследуемого объекта. Исследователи из Microsoft предложили совместить признаки Хаара и мощь ориентированных градиентов, назвав их Haar of Oriented Gradients (HoOG). В своей работе [29] авторы предлагают по одному изображению построить набор

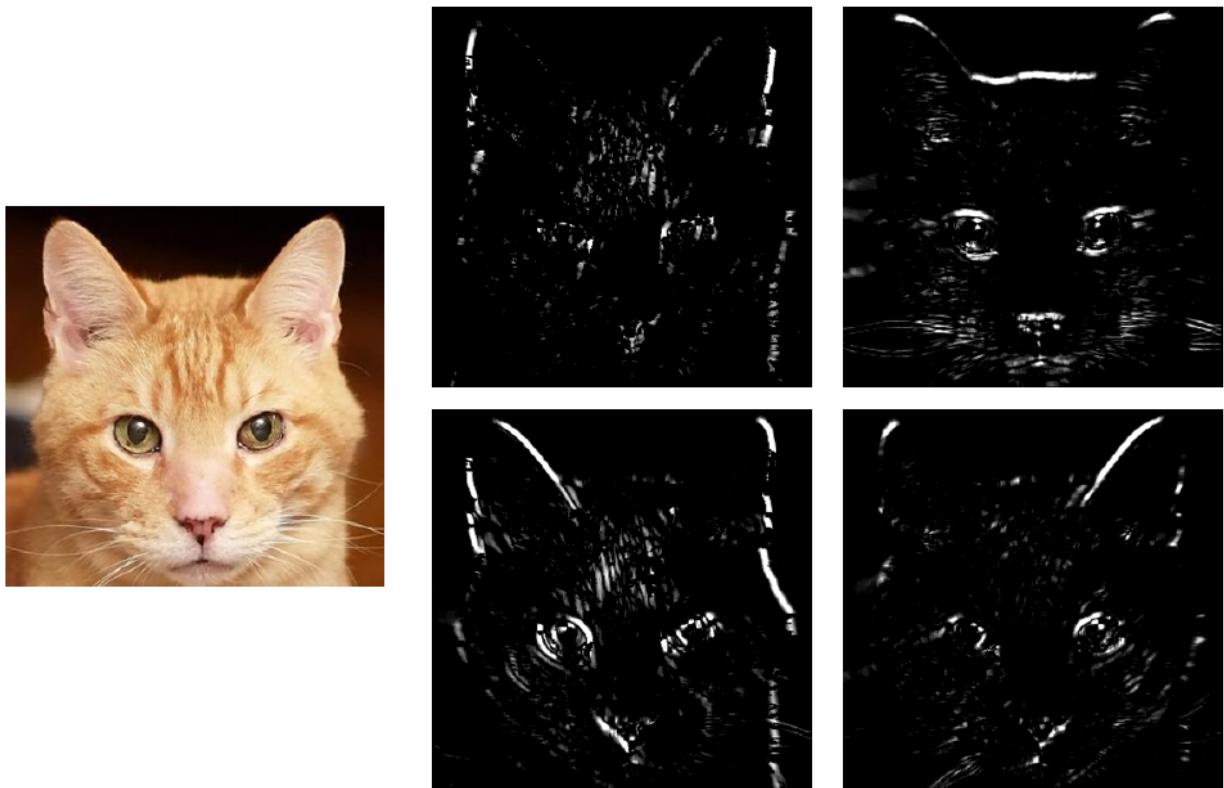


Рисунок 9 — Пример изображений ориентированных градиентов для четырех направлений (источник [29])

изображений ориентированных градиентов. На рисунке 9 проиллюстрированы изображения ориентированных градиентов для четырех направлений. Поверх построенных изображений вычисляются классические признаки Хаара. При этом суммы значений яркости может вычисляться как на одном градиентном изображении (*In-channel Haar Feature*), так и на двух изображениях градиентов ортогонального направления (*Orthogonal-channel Haar Feature*). Авторами утверждается, что использование описанных признаков позволяет хорошо обобщить объекты, обладающие псевдо-риgidной структурой.

Еще один способ модификации алгоритма Виолы и Джонса в части пространства признаков предложен в серии работ [15,30,31]. В них вводятся так называемые *интегральные поканальные признаки* (*integral channel features*), общая

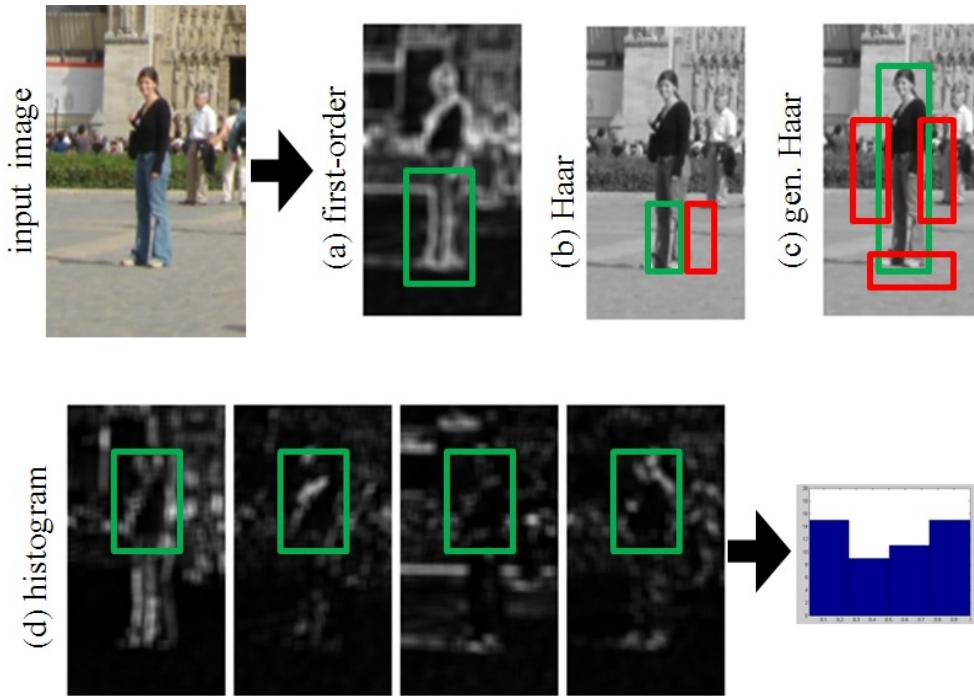


Рисунок 10 — Пример интегральных поканальных признаков: а) признак первого порядка – сумма яркостей пикселей внутри заданного прямоугольника; б) признак Хаара второго порядка – аппроксимация локального перепада яркости; в) обобщенный признак Хаара, вычисляющий взвешенную сумму яркостей пикселей внутри заданных прямоугольников; г) гистограмма направленных градиентов, посчитанная по изображениям направленных границ (источник [30])

идея которых заключается в следующем: с помощью линейных и нелинейных преобразований пикселей цифрового изображения вычисляются новые изображения (*зарегистрированные каналы изображения*, англ. *registered image channels*), а затем различные признаки (признаки Хаара, гистограммы, локальные суммы и т.п.) эффективно вычисляются, используя интегральные представления построенных зарегистрированных каналов (см. рисунок 10). Авторы в работах демонстрируют высокую эффективность предложенных признаков, превосходство над другими эффективными признаками (в частности, над гистограммами направленных градиентов), а также отражают следующие особенности представленных интегральных поканальных признаков:

- естественным образом объединяют гетерогенные источники информации, полученной по цифровому изображению;
- обладают малым количеством настроек параметров;
- допускают более точную пространственную локализацию объектов в процессе детектирования;
- обеспечивают возможность построения быстрых детекторов в сочетании с каскадными классификаторами.

#### **1.4.2 Модификации в части обучающего алгоритма**

Алгоритм AdaBoost является одним из алгоритмов, реализующих подход бустинга. На сегодняшний день наиболее популярны следующие методы:

- Discrete AdaBoost [5]. Исторически первая и наиболее популярная версия бустинга, именно она используется в оригинальном подходе Виолы и Джонса;
- Real AdaBoost [4,70]. Развитие классической версии AdaBoost в части области допустимого значение классификаторов. В отличии от классической версии, классификаторы возвращают степень уверенности в принадлежности к классу;
- LogitBoost [35]. Применение теории логистической регрессии к задаче классификации;

- GentleBoost [35]. Использует менее агрессивную политику для выбросовых точек, благодаря чему менее склонен к переобучению;
- BrownBoost [71]. В отличии от классического подхода использует невыпуклую функцию потерь, благодаря чему показывает лучшие результаты на зашумленной обучающей выборке.
- LPBoost [72]. С помощью аппарата линейного программирования максимизирует «зазор» между различными классами в обучающей выборке;
- MadaBoost [36]. Представляет собой модификацию AdaBoost в которой перманентно ограничены веса примеров. Данная модификация позволяет частично адаптировать обучающий алгоритм к работе на зашумленной выборке, а также позволяет использовать алгоритм в качестве инструмента по фильтрации выборки (filtering framework).

Несмотря на такое разнообразие алгоритмов бустинга, в прикладных системах в основном используются два алгоритма: Real AdaBoost и LogitBoost. Дополнительное преимущество данным алгоритмам придает наличие качественной программной реализации в библиотеке с открытыми исходными кодами OpenCV [73]

Основная идея алгоритма Real AdaBoost заключается в развитии классического AdaBoost на случай использования вещественно-значных слабых классификаторов. В частности, при бинарной классификации, когда  $\mathbb{Y} = \{+1, -1\}$  слабый классификатор представляет собой отображение  $b: \mathbb{X} \rightarrow [0,1]$ , при этом значение классификатора интерпретируется как вероятность принадлежности объекта к «положительному» классу. Алгоритм Real AdaBoost широко применяется в задачах детектирования объектов разной природы [16,37,40,70,74]

Алгоритм LogitBoost представляет собой адаптацию подхода логистической регрессии к идее бустинга, в результате чего построенный классификатор обладает целым набором интересных последствий. Во-первых, он оказывается оптимальным с точки зрения байесовского классификатора. Во-вторых, он позволяет строить более устойчивый к шуму классификатор. В-третьих, помимо результата классификации, возникает возможность получать численные оценки вероятности

принадлежности исследуемых объектов к найденным классам. Алгоритм LogitBoost в настоящее время менее распространен в силу относительной новизны, тем не менее активно используется в современных работах [35, 75–77].

### 1.4.3 Модификации в части каскадной структуры классификатора

Несмотря на то, что каскадная структура классификатора позволяет существенно ускорить производительность детектирования объектов, она не лишена недостатков.

Во-первых, в соответствии с архитектурой каскада, информация, полученная на текущем уровне каскада, ни в каком виде не передается на следующие уровни. Соответственно, решение о том, отбраковывать ли на очередном уровне каскада рассматриваемый участок изображения никак не зависит от того, насколько хорошо он был распознан на предыдущих уровнях. Такой подход иногда приводит к построению «хрупкого» классификатора, который «полностью ломается» при

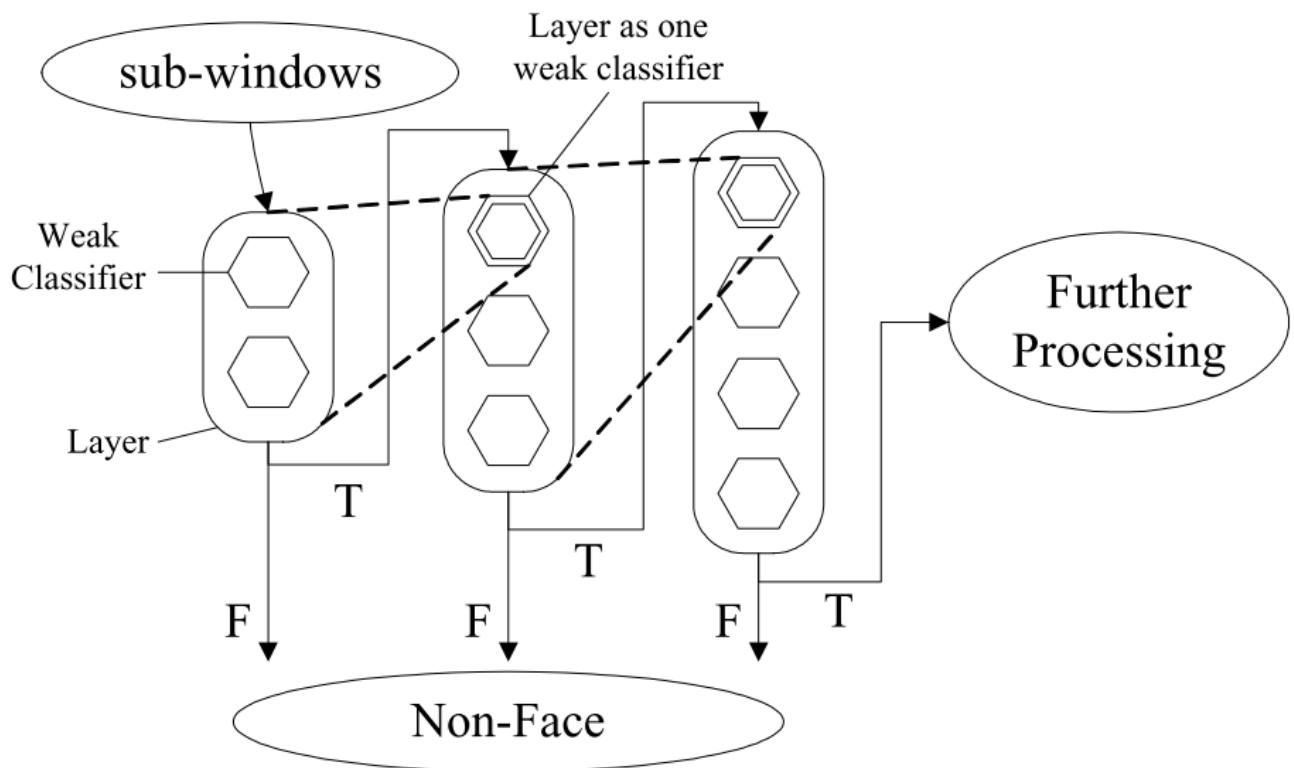


Рисунок 11 — Схема связанного каскадного классификатора (источник [40])

малейших колебаниях на отдельных уровнях. Для решения данной проблемы ученые предлагают использовать в качестве степень уверенности классификатора на предыдущем уровне в качестве слабого классификатора при построении классификатора текущего уровня [39,40]. В результате получается своего рода «связанный каскад», который в англоязычной литературе встречается в виде термина Chain Cascade или Nesting-structure Cascade (см. рисунок 11).

Второй недостаток классического каскада – это отсутствие оптимального способа построения уровней каскада. Дело в том, что при обучении классического каскадного классификатора для каждого уровня должны указываться несколько коррелирующих между собой параметров: количество слабых классификаторов, долю ложных срабатываний, долю ложных пропусков. Использование различных наборов данных параметров позволяют достигать компромисса между производительностью и качеством детектирования. В 2005 году был предложен новый вид каскадного классификатора, получивший название «легкий каскад» (Soft Cascade) [30,41,78]. Структурно, легкий каскад похож на сильный классификатор, обученный методом бустинга, однако способный отклонить отрицательные области после вычисления очередного слабого классификатора. Достигается это за счет сравнения частичных сумм линейных комбинаций слабых

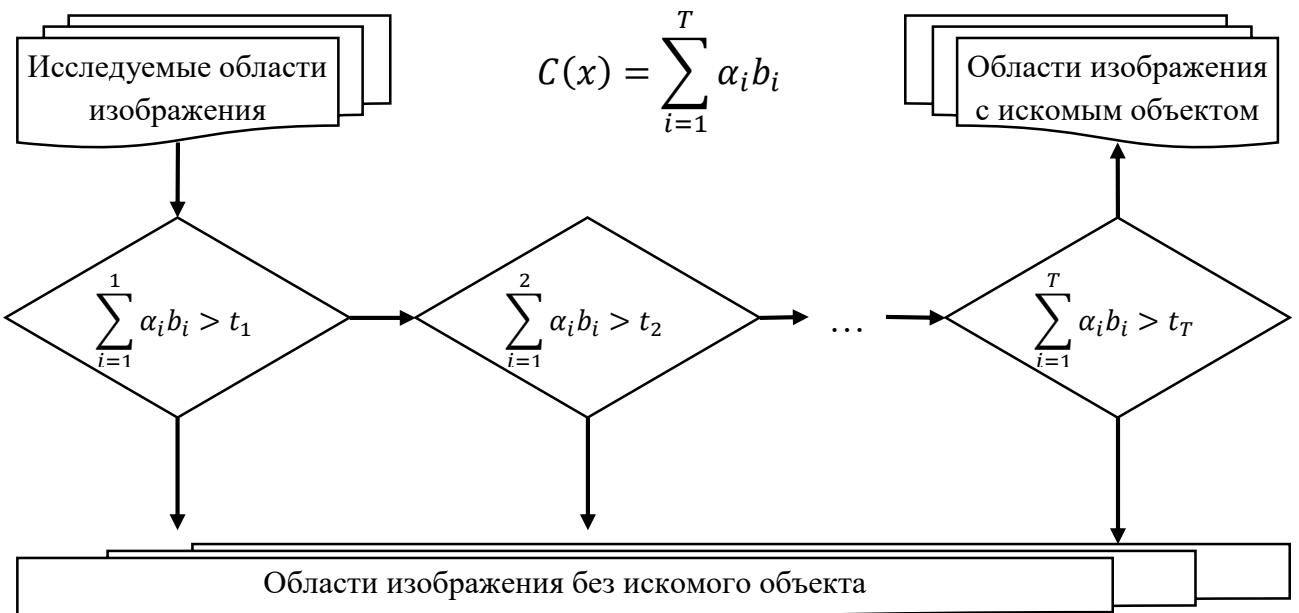


Рисунок 12 — Схема легкого классификатора

классификаторов с пороговыми значениями в процессе вычисления значения сильного классификатора. Процедура построения легкого каскада состоит из двух этапов: построение сильного классификатора и поиск пороговых значений отсечения.

Третий недостаток, присущий классическому каскадному классификатору – это неприменимость в случае большой вариативности обучающих прецедентов. В этом случае обычно задача решается путем кластеризации обучающей выборки, обучения различных каскадов для каждого кластера, и использования в некоторых случаях пре-классификатора для выбора целевого каскада [37,40,79].

## **1.5 Применение метода Виолы и Джонса в индустриальных распознающих системах**

В современном мире инновации в части компьютерных технологий делают реальным использование распознающих программных систем в индустриальном масштабе. Часто алгоритмы распознавания встраиваются в уже готовые системы, обеспечивая автоматическую регистрацию различных событий, транслируемых далее в отдельные параметры в терминах системы управления. Примерами индустриальных программных комплексов, неотъемлемой частью которых стали системы распознавания, являются различные комплексы контроля дорожного движения [80], системы работы с документами [81,82], модули видеонаблюдения в системах безопасности [83].

Метод Виолы и Джонса также нашел свое применение в индустриальных программных комплексах [84–86]. При этом, в связи с особенностями промышленных распознающих систем, к методу Виолы и Джонса предъявляется ряд требований. Во-первых, исходные данные (изображения и видеопоследовательности), которые поступают в алгоритм в «живых» условиях существенно разнятся с теми, которые рассматриваются в лабораторной среде. Большую роль на качество исходных данных влияет окружающая среда (погодные условия для видеосъемки, качество сканирующего оборудования в случае системах документооборота и т.п.). Поэтому вычисляемые в алгоритме поиска объектов

признаки должны быть робастны относительно таких искажений. Во-вторых, индустриальные программные комплексы предъявляют высокие требования к скорости обработки данных. Причем, помимо высокой скорости «в среднем» алгоритм должен отличаться низкой дисперсией времени обработки однородных данных (в идеале время обработки данных одинакового размера должно совпадать). В-третьих, жизненный цикл промышленного программного обеспечения подразумевает регулярное обновление модулей (в том числе, распознающих алгоритмов) в процессе эксплуатации, с целью адаптации работы системы к новым данным. Следовательно, алгоритм распознавания должен обеспечивать возможность быстрого «дообучения» при расширении обучающей выборки за счет новых данных.

Таким образом, несмотря на то, что метод Виолы и Джонса уже сейчас применяется в индустриальных системах распознавания, задача развития алгоритма с целью удовлетворения перечисленным требованиям является высоко востребованной.

## **1.6 Конкретизация цели и задачи исследования**

В первом разделе был подробно описан метод Виолы и Джонса поиска объектов на изображениях. Благодаря использованию легких с точки зрения вычислений признаков Хаара в паре с идеей бустинга для построения классификаторов и каскадной схемой, данный алгоритм может быть применим для решения задач детектирования в режиме реального времени.

Однако несмотря на общность, декларированную в работах Виолы и Джонса, описанный в оригинальном виде алгоритм хорошо подходит для решения конкретной задачи: быстрого детектирования лиц на изображениях. Прямое применение описанного метода для решения других прикладных задач не приводит к высоким результатам. Причин для этого несколько.

Во-первых, значения классических признаков Хаара, даже в известных нормализованных модификациях, оказываются неинвариантны к изменению освещенности, а модификации признакового пространства, оперирующие с

граничными точками, оказываются либо вычислительно трудоемкими, либо чувствительны к шумовым выбросам, либо неприспособленными к масштабированию.

Во-вторых, алгоритм Виолы и Джонса, как и представленные модификации, решает задачу поиска объектов в «лабораторной» постановке: наборы прецедентов известны, зафиксированы и не требуют дополнительной кластеризации на отдельные подтипы, детектирование объектов выполняется на отдельных стационарных изображениях. Однако индустриальные системы распознавания предъявляют к алгоритмам поиска объектов дополнительные требования. Так, например, в качестве источника данных часто выступает видеокамера, обеспечивающая вместо отдельных стационарных изображений коррелированную последовательность кадров, которые могут быть использованы для повышения производительности алгоритма. В качестве обучающих примеров часто выступают «сырые» данные, требующие предварительного разделения на подтипы. И, наконец, хотя в соответствии с оригинальным методом обучение детектора ведется в «пакетном режиме» (обучающая выборка известна и зафиксирована), на практике регулярно возникает задача «дообучения» детектора в связи с появлением новых данных.

Из сказанного следует, что задача алгоритмического развития Виола-Джонсовских детекторов для решения прикладных задач распознавания изображений, особенно в части применения метода в индустриальных программных комплексах, крайне актуальна. В соответствии с целями диссертационной работы и учетом выше сказанного, конкретизируем исходно поставленные задачи диссертационной работы следующим образом.

1. Разработать семейство вычислительно эффективных признаков, обеспечивающих высокий уровень обобщения объектов и устойчивых к различным параметрам освещенности, применимых для использования в индустриальных распознающих системах.

2. Разработать высокоуровневую структуру классификатора, применимую для поиска вариативных объектов (с большой внутриклассовой вариацией), а также

обеспечивающую функциональность дообучения при расширении обучающего набора данных. Следует разработать алгоритм построения такой структуры, исследовать эффективность по сравнению с классическим каскадом.

3. Построить математическую модель многоклассовой детекции объектов на видеопоследовательности. Разработанная модель должна существенно улучшить производительность детектора объектов в случае поочередной демонстрации целевых объектов. Используя построенную модель требуется разработать алгоритм многоклассовой детекции объектов на видеопоследовательности для указанного случая.

4. Разработать программный инструментарий, содержащий имплементацию полученных результатов. Инструментарий должен с архитектурной точки зрения алгоритмически расширять и обеспечивать полный цикл обучения классификаторов Виолы и Джонса (подготовку обучающих данных, непосредственно обучение классификатора, оценка качества работы построенного классификатора).

## 2 Построение устойчивых признаков для алгоритма Виолы и Джонса, учитывающих геометрические особенности объектов

Классический подход Виолы и Джонса [7] с использованием признаков Хаара, построенных по яркостному изображению реагирует на характерные для объекта области яркости – когда одна часть объекта на изображении всегда темнее другой (например, область глаз на изображении всегда темнее области лба).

Тем не менее, на практике часто оказывается, что ригидные объекты, обладающие «в теории» характерным яркостным распределением, «в жизни» выглядят несколько иначе. Рассмотрим в качестве примера автомобильное колесо, которое в классическом варианте состоит из стального диска и каучуковой покрышки. Если представить изображение автомобильного колеса сбоку, то область шины будет выглядеть темнее, чем область стального диска. Однако, анализируя реальные фотографии колес (см. рисунок 13), видно, что характерный яркостной контраст между покрышкой и диском не соблюдается. Основными причинами этого являются непостоянные условия освещенности (яркое солнце,

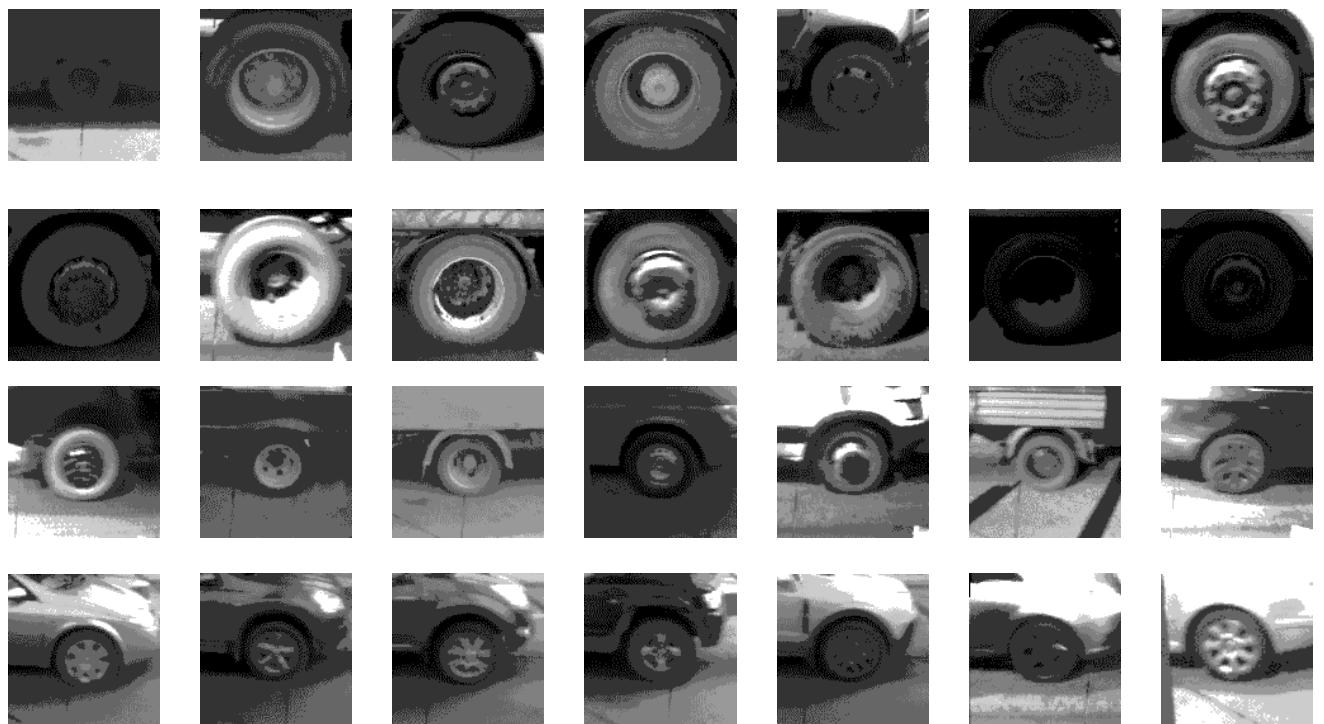


Рисунок 13 — Примеры изображений колес

тень, дым) и общее загрязнение колес. Следовательно, построение детектора, оперирующего яркостными признаками, для такого рода объектов неэффективно.

Другим примером, в котором не применимы признаки, основанные на яркостных характеристиках, является задача распознавания символов, нанесенных на поверхность выдавливанием или чеканкой (надписи на пластиковых картах, агрегатные номера на автомашинах и станках и т.п.). На рисунке 14 продемонстрировано изображения символа «2» на различных пластиковых картах, снятые при различных условиях наблюдения.

Таким образом далеко не во всех задачах можно опираться на распределение яркости при построении детекторов. Альтернативный путь – при построении детектора учитывать геометрические особенности исследуемых объектов. В первом разделе было показано, что детекторы, построенные на базе признаков, оперирующих градиентами, обеспечивают лучшее по сравнению с яркостными признаками качество обнаружения таких объектов. Тем не менее, использование градиента изображения в качестве описателя геометрических особенностей не совсем корректно: во-первых, модуль градиента, как и яркость, зависит от параметров освещенности, а, во-вторых, при использовании классических методов построения градиента изображения порождается много шумовых «границных точек».

В рамках диссертационной работы разработано новое семейство признаков, устойчивых к различным параметрам освещенности и не обладающих характерными яркостными особенностями. Данные признаки представляют собой прямоугольные признаки Хаара, вычисляемые поверх карты направленных границ, что позволило существенно улучшить обобщающую силу. В качестве детектора



Рисунок 14 — Примеры изображений цифры «2», нанесенных на кредитную пластиковую карту

границ предлагаемые признаки используют модифицированный метод Кэнни [87], в котором знания о направлении границы учитываются путем «раскраски» пикселей изображения карты направленных в разные цвета в зависимости от направления.

## 2.1 Карта направленных границ

Карта направленных границ представляет собой модификацию метода Кэнни [87], в результате которой каждой найденной граничной точки дополнительно приписывается ее преимущественное направление. В работе используется четыре преимущественных направления: горизонтальное, вертикальное,  $+45^\circ$  и  $-45^\circ$ . Как было сказано выше, знания о направлении границы учитываются путем «раскраски» пикселей изображения карты направленных в разные цвета в зависимости от направления. Более того, так как поверх построенной карты направленных границ в соответствии с предлагаемым методом будут вычисляться прямоугольные признаки Хаара (определенные как разность сумм пикселей смежных прямоугольников), то для повышения обобщающей способности признаков разумно раскрашивать пиксели, соответствующие ортогональным границам в максимально разнесенные (в соответствии с используемой шкалой яркости) цвета – белый и черный. Все остальные пиксели (пиксели фона или пиксели, обладающие другими преимущественными направлениями) – в нейтральный серый цвет. При таком подходе возникает два типа карты направленных границ:

- карта прямых границ, содержащая преимущественно горизонтальные и вертикальные границы;
- карта диагональных границ, содержащая преимущественно границы вдоль главной и побочной диагоналей (другими словами, границы с преимущественным направлением  $\pm 45^\circ$ ).

На рисунке 15 продемонстрированы примеры карты прямых и диагональных границ.

Далее при описании алгоритма построения карты направленных границ будем использовать следующее необходимое условие, которому должны удовлетворять граничные точки

**Определение 11.** Если точка  $(y_e, x_e)$  изображения  $f(y, x)$  принадлежит искомой границе (является *граничной точкой*), то в ней достигается локальный максимум производной в направлении градиента функции  $f(y, x)$ .

Теперь опишем пошагово алгоритм построения карты направленных границ. Первые пять шагов алгоритма совпадают как при построении карты прямых границ, так при построении карты диагональных границ. Отличия возникают только на шаге 6.

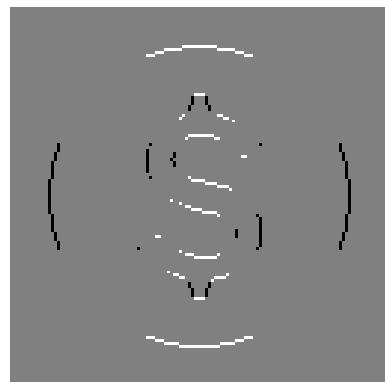
*Шаг 1. Сглаживание изображения с целью уменьшения шума.* Для этих целей применяется двумерная гауссова фильтрация, эффективно подавляющая белый аддитивный шум.

*Шаг 2. Вычисление частных производных  $\partial f / \partial y$  и  $\partial f / \partial x$ .* Поскольку  $f(y, x)$  является функцией с дискретными величинами, необходимо использовать дискретную аппроксимацию частных производных в окрестности данной точки:

$$g_y(y, x) = \frac{\partial f(y, x)}{\partial y} = \frac{f(y + 1, x) - f(y - 1, x)}{2}, \quad (14)$$



a)



б)



в)

Рисунок 15 — Иллюстрация карты направленных границ: а) исходное изображение; б) карта прямых границ; в) карта диагональных границ

$$g_x(y, x) = \frac{\partial f(y, x)}{\partial x} = \frac{f(y, x+1) - f(y-1, x)}{2}. \quad (15)$$

*Шаг 3. Обнаружение потенциально граничных точек в соответствии с необходимым условием.* Зная частные производные в каждой точке, определяется модуль  $g(y, x)$ , направление градиента  $\alpha(y, x)$  и преимущественное направление  $d(y, x)$ , определяемые следующим образом:

$$g(y, x) = |\nabla f(y, x)| = \sqrt{g_y^2 + g_x^2}, \quad (16)$$

$$\alpha(y, x) = \arctg\left(\frac{g_y}{g_x}\right), \quad (17)$$

$$d(y, x) = \begin{cases} 0^\circ, & \text{если } |\alpha(y, x)| \leq \frac{\pi}{8} \\ 90^\circ, & \text{если } |\alpha(y, x)| \geq \frac{3\pi}{8} \\ +45^\circ, & \text{если } \frac{\pi}{8} \leq \alpha(y, x) \leq \frac{3\pi}{8} \\ -45^\circ, & \text{если } -\frac{3\pi}{8} \leq \alpha(y, x) \leq -\frac{\pi}{8} \end{cases}. \quad (18)$$

Для поиска локального максимума производной в направлении градиента функции  $f(y, x)$  используются окрестность  $3 \times 3$  и дискретизация всевозможных направлений на горизонтальное, вертикальное,  $+45^\circ$  и  $-45^\circ$ .

*Шаг 4. Двойная фильтрация потенциальных граничных точек.* Значение модуля градиента каждой потенциальной граничной точки сравнивается с двумя заданными пороговыми значениями  $T_H$  и  $T_L$ ,  $T_H > T_L$ . Если,  $g(y, x) > T_H$ , то потенциальная граничная точка считается «сильной». Если  $T_H \geq g(y, x) > T_L$ , то потенциальная граничная точка считается «слабой». Все «сильные» точки

считываются истинными граничными точками, тогда как «слабые» подвергаются трассировке на следующем шаге.

*Шаг 5. Трассировка области неоднозначности.* Каждая «слабая» потенциальная точка принадлежит истинной границе, если она связана с любой другой истинной граничной точкой в смысле 8-связности. 8-связные компоненты связности, состоящих из граничных точек, будем называть *границами*.

*Шаг 6. Построение изображения направленных границ.*

В случае построения изображения карты диагональных границ  $f_{de}(y, x)$  значение в точке  $(y, x)$  определяется следующим образом:

$$f_{de}(y, x) = \begin{cases} 0, & \text{если } (y, x) - \text{гранична точка и } d(y, x) = +45^\circ \\ 255, & \text{если } (y, x) - \text{гранична точка и } d(y, x) = -45^\circ \\ 128, & \text{иначе} \end{cases}. \quad (19)$$

В случае построения изображения карты прямых границ  $f_{se}(y, x)$  значение в точке  $(y, x)$  определяется следующим образом:

$$f_{se}(y, x) = \begin{cases} 0, & \text{если } (y, x) - \text{гранична точка и } d(y, x) = 90^\circ \\ 255, & \text{если } (y, x) - \text{гранична точка и } d(y, x) = 0^\circ \\ 128, & \text{иначе} \end{cases}. \quad (20)$$

Блок-схема описанного выше алгоритма представлена на рисунке 16.

## 2.2 Контурные признаки Хаара

Поверх построенной карты направленных границ (диагональной или прямой) вычисляются прямоугольные признаки Хаара. Благодаря этому контурные признаки Хаара наследуют все преимущества классических яркостных признаков Хаара. Во-первых, существует метод быстрого вычисления значения признака вне зависимости от размера с помощью интегрального представления изображения [56].



Рисунок 16 — Блок-схема алгоритма построения карты направленных границ

Во-вторых, такого рода признаки слабо чувствительны к незначительным искажениям обучающей выборки, что помогает акцентировать внимание на характерных особенностях классифицируемых объектов, снижая при этом влияние шума. В-третьих, прямоугольные признаки линейно масштабируемые, что облегчает задачу поиска объектов различных размеров (масштабируя обученные классификаторы, а не исходное изображение).

Кроме того, границы, найденные с помощью детектора Кэнни [87], инвариантны к монотонному преобразованию пикселей изображения. Следовательно, значения представленных в диссертационной работе признаков, посчитанных поверх карты направленных границ, не требуют дополнительной нормировки для компенсации различной параметров освещенности, что позволяет использовать такие признаки в индустриальных системах распознавания для детектирования объектов в неконтролируемых условиях.

### **2.3 Численный эксперимент и результаты**

Эффективность описанных признаков была испытана в процессе решения двух задач распознавания: распознавание образов колес на фотографиях в боковом ракурсе и распознавание номера кредитной карты. В обоих случаях на одной и той же обучающей выборке с помощью одного и того же инструментария *objed* (раздел 5, приложение А) строились классификаторы типа Виолы и Джонса на базе яркостных и контурных признаках Хаара. Оценку эффективности классификаторов будем выполнять с помощью Precision-Recall анализа, описанному в разделе 1.2.3. Для подсчета верно распознанных объектов будем использовать методику, предложенную в рамках соревнования The PASCAL Visual Object Classes (VOC) Challenge [49].

#### **2.3.1 Распознавание образов колес**

Анализ задачи детектирования колес показал, что на изображениях автомобилей в данном ракурсе имеется большое количество вертикальных и



Рисунок 17 — Изображение колеса и соответствующих карт границ:

а) исходное изображение; б) карта границ Кэнни;

в) карта диагональных границ

горизонтальных границ (днище автомобиля, бензобак грузовика и т.д.), в то время как границ, направленных по диагоналям мало – и значительное количество из них как раз относится к колесам (см. рисунок 17). Поэтому для задачи детектирования колес использовались контурные признаки Хаара, посчитанные поверх карты диагональных краев.

Для их обучения классификатора были подготовлены положительная и отрицательная выборки. Положительная выборка состояла в основном из изображений «светлых» (снятых при нормальных условиях освещенности) колес (всего 460 изображений). Отрицательная выборка состояла из 7420 изображений, не содержащих образа колеса.

При обучении использовались следующие параметры:

- доля ложных срабатываний при обучении очередного слоя каскадного классификатора  $fpr^* = 0,2$ ;
- доля верных срабатываний при обучении очередного слоя каскадного классификатора  $fpr^* = 0,98$ ;
- количество обучаемых слов  $S = 15$ ;
- физический размер классификатора в пикселях  $60 \times 60$ ;
- набор масштабов классификатора от 1,0 до 2,5 с аддитивным шагом 0,1.

Оценка качества детектирования выполнялась на специальной тестовой выборке, состоящей из 10477 полноразмерных изображений, содержащих колеса

при всевозможных условиях освещенности (при обычных условиях, яркое солнце, колеса частично или полностью покрываемые тенью и т.п.). Результаты тестирования такого «контурного» классификатора для различных порогов классификации представлены в таблице 1.

Таблица 1 — Результат работы классификатора колес (контурные признаки)

<b>Порог</b>	<b>True Positive</b>	<b>False Positive</b>	<b>False Negative</b>	<b>Precision</b>	<b>Recall</b>
39	3168	32	7309	0,99	0,30
35	3663	37	6814	0,99	0,35
31	4950	50	5527	0,99	0,47
27	5940	60	4537	0,99	0,57
23	7821	79	2656	0,99	0,75
19	7848	327	2629	0,96	0,75
15	8417	443	2060	0,95	0,80
11	8562	605	1915	0,93	0,82
9	8649	961	1828	0,90	0,83
7	8806	1235	1671	0,88	0,84
5	8961	1339	1516	0,87	0,86
3	9222	1564	1255	0,86	0,88
1	9348	2485	1129	0,79	0,89

Для сравнения был обучен также классификатор с использованием классических яркостных признаков на данной обучающей выборке, с использованием таких же параметров обучения. Результаты тестирования такого «яркостного» классификатора для различных порогов классификации представлены в таблице 2.

Таблица 2 — Результат работы классификатора колес (яркостные признаки)

Порог	True Positive	False Positive	False Negative	Precision	Recall
39	2079	21	8398	0,99	0,20
35	2376	24	8101	0,99	0,23
31	2772	28	7705	0,99	0,26
27	3168	32	7309	0,99	0,30
23	4851	99	5626	0,98	0,46
19	6370	197	4107	0,97	0,61
15	6632	298	3845	0,96	0,63
11	6688	352	3789	0,95	0,64
9	6919	426	3558	0,94	0,66
7	6936	686	3541	0,91	0,66
5	7056	1344	3421	0,84	0,67
3	7284	1599	3193	0,82	0,70
1	7576	2662	2901	0,74	0,72

Используя приведенные табличные данные, построим Precision-Recall кривые на одном графике для обоих классификаторов (см. рисунок 18).

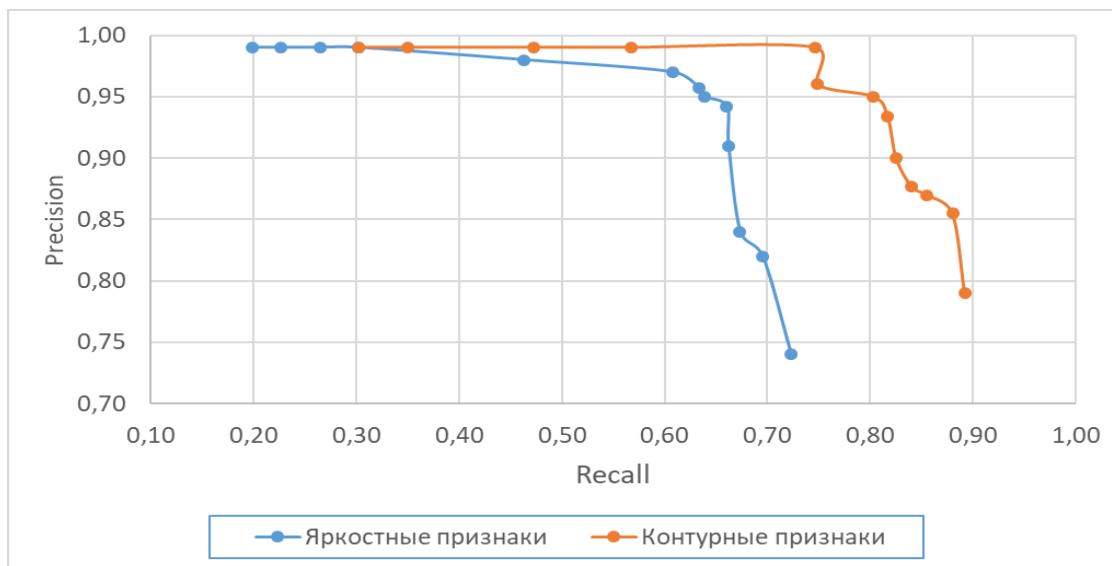


Рисунок 18 — Precision-Recall кривые классификаторов колес

Из графика видно, что при одном и том же значении полноты точность «контурного» классификатора существенно выше. Следовательно, классификатор на основе разработанных контурных признаков Хаара более устойчив к различным условиям освещенности.

### 2.3.2 Распознавание номера кредитной карты

Анализ задачи детектирования и распознавания цифирных символов, нанесенных выдавливанием и чеканкой, показал, что для нанесения данных символов как правило используется небольшой набор шрифтов, в нашей конкретной задаче – шрифты, большинство направлений границ в которых ориентированы вдоль вертикали и горизонтали (см. рисунок 19). В то же время направление контрастов фонового изображения примерно равномерно распределен по всем направлениям. Поэтому для задачи детектирования цифр использовались контурные признаки Хаара, посчитанные поверх карты прямых границ.

В рамках задачи распознавания для каждой цифры тренировались бинарные классификаторы двух типов (на базе контурных и яркостных признаков Хаара соответственно). Происходило обучение принципу «один против всех», то есть при обучении классификатора на узнавание какой-то цифры, изображения остальных цифр помещались в отрицательную выборку. Потом на основе нескольких таких бинарных классификаторов можно сконструировать общий распознаватель,

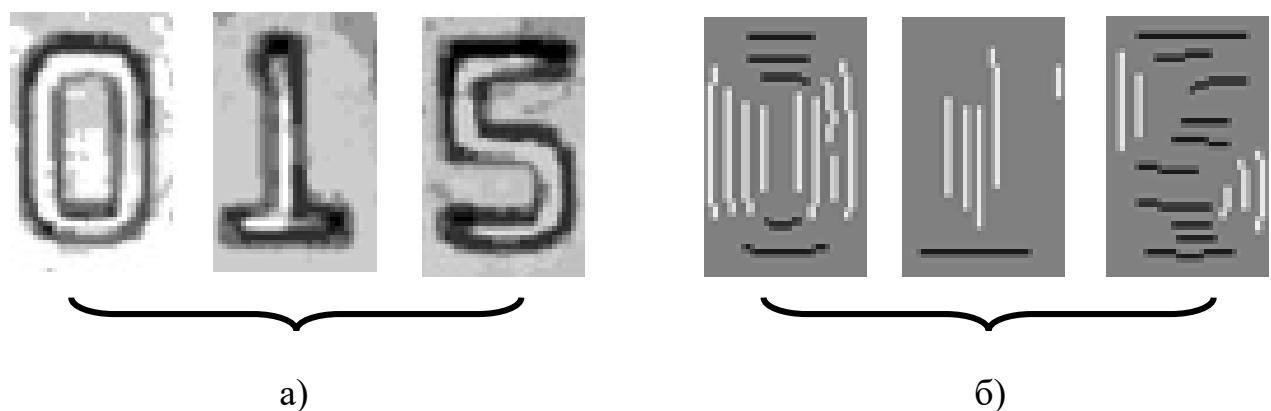


Рисунок 19 — Изображение цифр номера кредитной карты:  
а) исходные изображения; б) карты прямых границ

который будет соотносить показанное ему изображение к одному из классов, соответствующих цифрам.

При обучении использовались следующие параметры:

- доля ложных срабатываний при обучении очередного слоя каскадного классификатора  $fpr^* = 0,3$ ;
- доля верных срабатываний при обучении очередного слоя каскадного классификатора  $fpr^* = 0,97$ ;
- количество обучаемых слов  $S = 15$ ;
- физический размер классификатора в пикселях  $30 \times 46$ ;
- набор масштабов классификатора от 1,0 до 1,2 с аддитивным шагом 0,05.

В положительные примеры для каждой цифры отбирались те изображения цифр, где сама цифра светлая, а фон темный. В тестовой выборке для каждой цифры были представлены изображения с совершенно разными условиями освещенности.

Как и в первом эксперименте «контуры» классификаторы для каждой цифры продемонстрировали более высокое качество распознавания, чем соответствующие «яркостные» классификаторы. В таблице 3 приведено сравнение полноты «контурного» и «яркостного» классификаторов цифр при практически идеальной точности (то есть, когда  $Precision = 0.99$ ).

Таблица 3 — Сравнение полноты классификаторов

Цифра	Полнота «контурного» классификатора	Полнота «яркостного» классификатора
0	0,91	0,45
1	0,89	0,57
2	0,72	0,49
3	0,64	0,52
4	0,96	0,54
5	0,83	0,52

Таблица 3 — Сравнение полноты классификаторов

Цифра	Полнота «контурного» классификатора	Полнота «яркостного» классификатора
6	0,86	0,54
7	0,78	0,68
8	0,86	0,48
9	0,75	0,41

Полученные результаты объясняются тем, что в тестовой выборке примерно половина изображений была похожа на положительные образцы, а оставшиеся изображения существенно отличались от них по своим цветовым характеристикам. Таким образом, «яркостные» классификаторы распознали только первую часть тестовой выборки, в то время как «контурные» классификаторы показали хороший результат в целом на всей выборке.

## 2.4 Выводы по второму разделу

В рамках диссертационной работы разработан новый вид признаков для задач детекции и классификации объектов на изображении. Представленный вид признаков для алгоритма Виолы и Джонса показал себя эффективным и устойчивым в случаях, когда детектируемые объекты не обладают характерными яркостными контрастами, но имеют границы характерной формы. При этом сохраняются преимущества классических признаков Хаара – устойчивость к шуму и вычислительная эффективность.

Описанные признаки были применены для решения двух задач распознавания: распознавание образов колес на фотографиях в боковом ракурсе и распознавание номера кредитной карты. В обеих задачах контурные признаки позволил обучить более эффективный классификатор по сравнению с классическими яркостными признаками Виолы и Джонса.

### **3 Высокоуровневая структура классификатора Виолы и Джонса в виде решающего дерева сильных классификаторов**

Как упоминалось ранее в диссертационной работе, в последнее время поиск и распознавание объектов на цифровых изображениях и видеопоследовательностях из стадии лабораторных исследований переросли в стадию использования в индустриальных системах. Как следствие, от алгоритма поиска объектов ожидается устойчивая работа в части детектирования вариативных объектов, наличие которых обуславливается различными параметрами освещенности, косвенным влиянием разных элементов сцены на изображение объекта и т.п. Оригинальный метод Виолы и Джонса строился в предположении, что карта яркостных контрастов на объекте не меняется от раза к разу. Это не так в случае, например, наличия контрастных теней от других объектов. В качестве одного из решений может быть использована предварительная кластеризация обучающей выборки и последующее обучение отдельного независимого каскада для каждого кластера. Однако, такой подход омрачен двумя проблемами: во-первых, обученные каскады будут «бороться» против однотипных пустых подокон, а, во-вторых, не ясным остается сам способ кластеризации.

Другой особенностью индустриальных распознающих систем является необходимость постоянного «дообучения» классификатора в связи с непрерывным расширением обучающей выборки. В соответствии с оригинальным методом обучение классификатора ведется в «пакетном режиме» (обучающий набор прецедентов известен и зафиксирован). Так как процедура обучения классификатора «с нуля» достаточно продолжительная (в зависимости от сложности объекта и размера обучающей выборки может занимать от нескольких часов до десятков суток), то востребованной оказывается задача построения высокоуровневой структуры классификатора и разработка алгоритма его обучения, обеспечивающего быструю процедуру «дообучения» классификатора при расширении обучающего набора.

В данном разделе представлена модель классификатора Виолы и Джонса в виде решающего дерева, содержащая несколько выходов с положительным результатом. В рамках диссертационной работы разработан алгоритм обучения такого классификатора, который автоматически выполняет кластеризацию данных в процессе обучения, обеспечивает быстрое дообучение классификаторов при расширении обучающих наборов, а также позволяет строить более производительные в сложных случаях классификаторы.

### 3.1 Решающее дерево сильных классификаторов

Решающее дерево сильных классификаторов представляет собой вид бинарного решающего дерева [88,89]: узел дерева – это сильный классификатор, на правое ребро которого попадают подокна, предположительно содержащие объект, а на левое – те, которые не распознались как объект, соответственно. Окончательный ответдается только в листьях.

Заметим, что каскадный классификатор, описанный в оригинальной работе Виолы и Джонса – это, по сути, древовидный классификатор, содержащий лишь один «положительный» выход (лист) и множество «отрицательных» выходов.

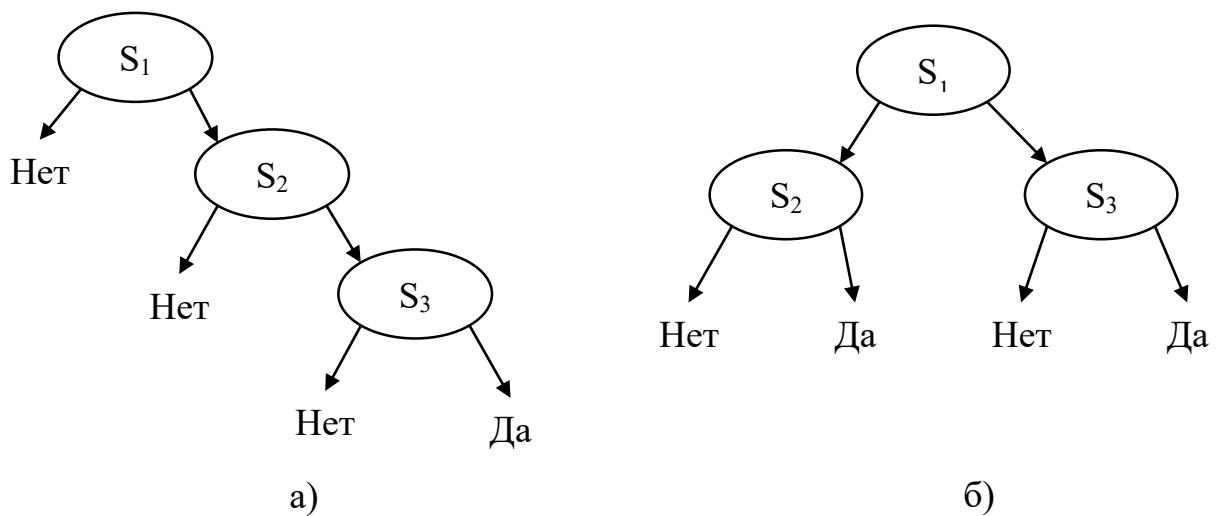


Рисунок 20 — Пример классификаторов Виолы и Джонса:  
а) классический каскадный б) в виде решающего дерева

На рисунке 20б показан древовидный классификатор более общего вида, каждая его ветвь состоит из 2 сильных классификаторов.

Разработанный метод построения (обучения) классификатора в виде решающего дерева основан на следующих предположениях.

*Предположение 1.* Построение каскадного классификатора происходит итерационно. Каждый следующий уровень обучается, используя знания предыдущих уровней.

*Предположение 2.* Обученный каскадный классификатор представим в виде конъюнкции сильных классификаторов. Следовательно, каскадный классификатор, состоящий из  $N$  уровней, формально можно представить следующим образом:

$$\text{Cascade}(x) = S_1(x) \wedge S_2(x) \wedge S_3(x) \wedge \dots \wedge S_N(x). \quad (21)$$

Будем считать, что каждый уровень  $S_i$  возвращает  $+1$ , если искомый объект найден, и  $-1$  в противном случае. Тогда формула (21) может быть переписана следующим образом:

$$\text{Cascade}(x) = [S_1(x) > 0] \cdot [S_2(x) > 0] \cdot \dots \cdot [S_N(x) > 0] = \prod_{i=1}^N [S_i(x) > 0], \quad (22)$$

где  $[S_i(x) > 0] = \begin{cases} 1, & \text{если } S_i(x) > 0 \\ 0, & \text{если } S_i(x) \leq 0 \end{cases}$  — индикатор найденного объекта.

*Предположение 3.* Любой путь от корня до самого нижнего узла древовидного классификатора может быть представлен как каскад, в котором отдельные сильные классификаторы входят в конъюнкцию с отрицанием. Например, на рисунке 20б правый путь представим в виде  $\text{Cascade}_{right}(x) = S_1(x) \wedge S_3(x)$ , тогда как левый путь определяется следующим выражением:  $\text{Cascade}_{left}(x) = \overline{S_1(x)} \wedge S_2(x)$ . Следовательно, обучение отдельных путей

древовидного классификатора может быть выполнено аналогично обучению классического линейного каскада.

Для дальнейшего описания введем следующие определения.

**Определение 12.** Под *взвешенным каскадом* будем подразумевать классический линейный каскад, у которого для каждого уровня (сильного классификатора)  $S_i(x)$  дополнительно задан вес (знаковый множитель)  $w_i = \pm 1$  с которым данный уровень участвует в произведении при вычислении значения каскада.

Исходя из определения, формула вычисления взвешенного каскада записывается следующим образом:

$$WeightedCascade(x) = \prod_{i=1}^N [w_i \cdot S_i(x) > 0]. \quad (23)$$

Для упрощения записи в дальнейших рассуждениях, будем схематично обозначать взвешенный каскад как последовательную конъюнкцию уровней, дополненных оператором отрицания тех из них, которые входят в общее произведение с отрицательным весом.

Как и в случае с классическим линейным каскадом, взвешенный каскад строится итерационно. Но в отличии от классического каскада, где при обучении очередного уровня обновляется только отрицательная выборка, состоящая из ложных срабатываний уже обученного каскада, при обучении очередного уровня взвешенного каскада обновляется так же и положительная выборка, состоящая из успешно пройденных сквозь уже обученный взвешенный каскад положительных примеров.

Опишем алгоритм обучения решающего дерева сильных классификаторов.

*Шаг 1 (Инициализация).* Пусть есть обучающая выборка  $\mathbb{X}$  и валидационная выборка  $\mathbb{X}^v$ , состоящие из положительных образцов и отрицательных наборов изображений (изображений, не содержащих целевой объект). Пусть задан параметр

$FPR^*$  – доля ложного срабатывания. Также пусть задано значение  $N_{wc}$  – мощность уровня дерева (количество слабых классификаторов для каждого уровня дерева).

*Шаг 2 (Обучение вершины дерева).* Обучаем очередную вершину решающего дерева методом «в ширину», то есть, последовательно, справа налево, начиная с вершины-источника (см. рисунок 21). Для подготовки обучающей выборки используем взвешенный каскад, построенный из уже обученных вершин, находящихся на пути к обучаемой вершине. Так, например, для обучения вершины  $S_4$  соответствующий взвешенный каскад определяется как  $WC_{S_4} = S_1 \wedge S_2$ , для обучения вершины  $S_5$  соответствующий взвешенный каскад определяется как  $WC_{S_5} = S_1 \wedge \bar{S}_2$ , для обучения вершины  $S_6$  соответствующий взвешенный каскад имеет вид  $WC_{S_6} = \bar{S}_1 \wedge S_3$  и так далее. Если при подготовке обучающей выборки для очередного уровня древовидного классификатора множество положительных или отрицательных примеров оказалось пустым, то рассматриваемая ветвь считается обучена полностью и дальнейшего обучения не требует.

*Шаг 3 (Проверка текущего состояния обучения).* Если достигнуты целевые параметры обучения ( $FPR < FPR^*$ ), оцененные на валидационной выборке  $\mathbf{X}^v$ , то закончить процедуру обучения. Иначе перейти на Шаг 2.

Блок-схема представленного алгоритма изображена на рисунке 22.

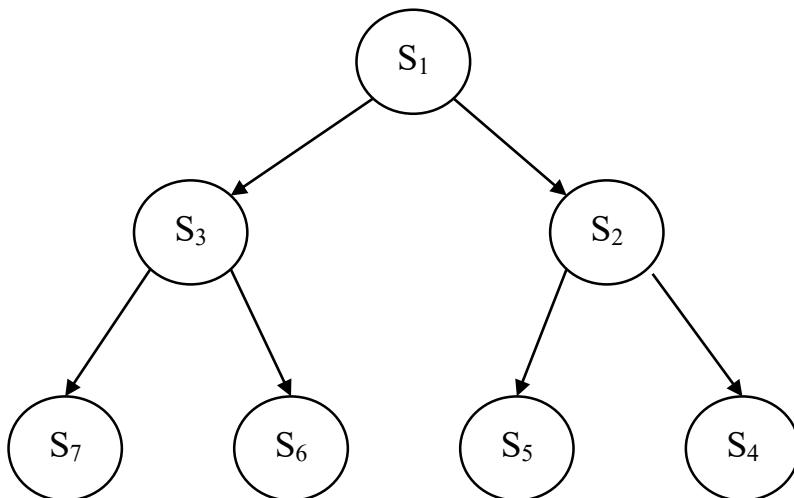


Рисунок 21 — Иллюстрация последовательности обучения «в ширину».

Внутри каждого уровня дерева указан его индекс при обучении

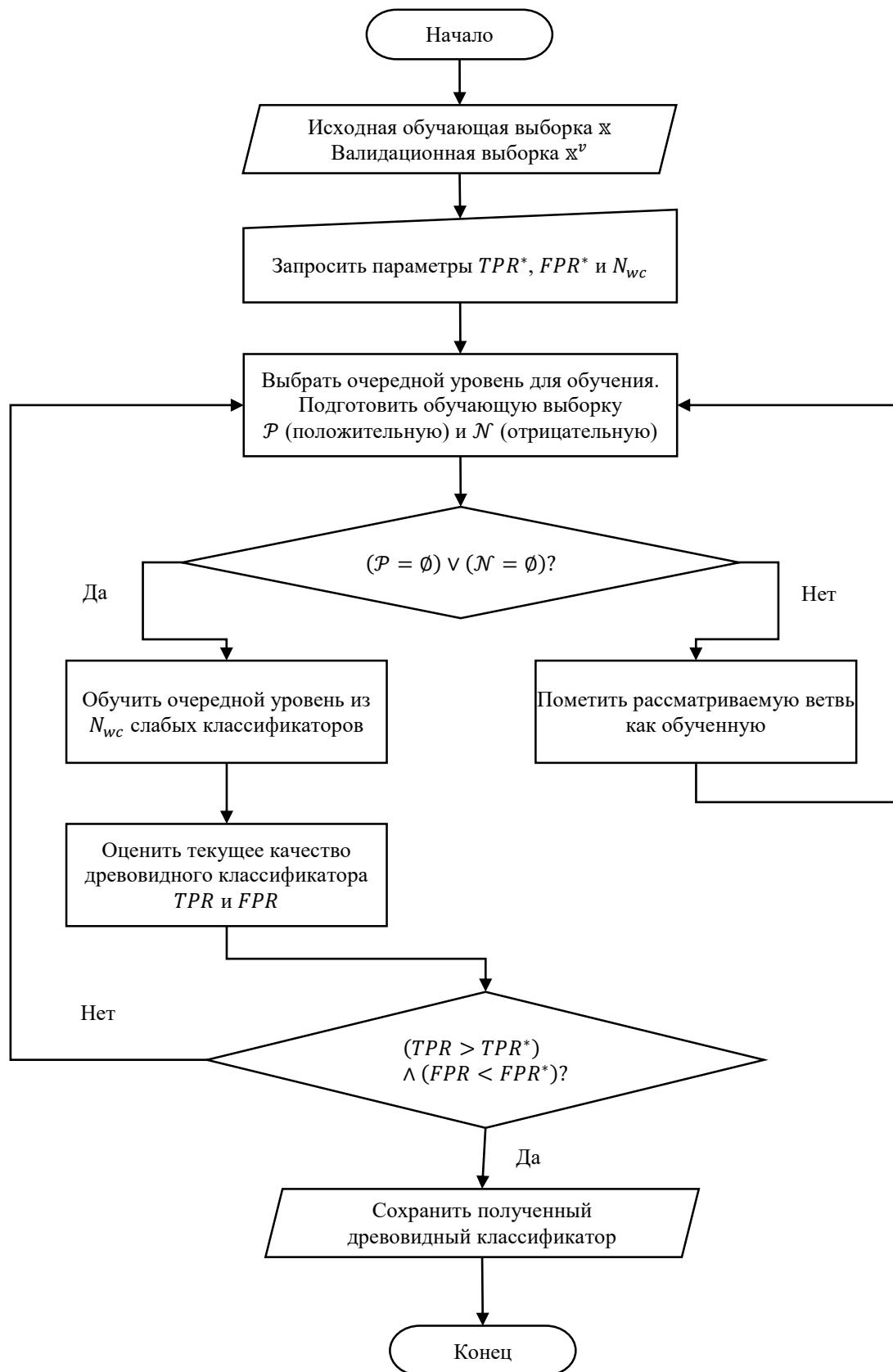


Рисунок 22 — Блок-схема алгоритма обучения высокоровневого классификатора в виде решающего дерева

В отличии от классического каскада, для которого справедливо сильное монотонное падение доли верных срабатываний с каждым новым уровнем (напомним, что доля верных срабатываний для каскада, состоящего из  $K$  уровней, оценивается как  $TPR = \prod_{i=1}^K tpr_i$ , где  $tpr_i$  – доля верных срабатываний для  $i$ -го уровня), древовидный классификатор обеспечивает более высокую долю верных срабатываний. Связано это с наличием нескольких путей в классификаторе, возвращающих положительный ответ (сумма всех долей верного срабатывания взвешенных каскадов, данный составляющих древовидный классификатор). Исходя из определения некоторого взвешенного каскада  $WC$ , доля верных срабатываний для него определяется следующим образом:

$$TPR_{WC} = \prod_{i:w_i > 0} trp_i \cdot \prod_{j:w_j < 0} (1 - tpr_j), \quad (24)$$

где  $tpr_i$  – доля верных срабатываний  $i$ -го уровня взвешенного каскада.

Тогда доля верных срабатываний для всего древовидного классификатора задается следующим образом:

$$TPR = \sum_{WC} \left( \prod_{i:w_i > 0} trp_i \cdot \prod_{j:w_j < 0} (1 - tpr_j) \right). \quad (25)$$

Например, для древовидного классификатора, изображенного на рисунке 206 и состоящего из двух взвешенных каскадов  $WC_{left}(x) = \overline{S_1(x)} \wedge S_2(x)$  и  $WC_{right}(x) = S_1(x) \wedge S_2(x)$  доля верных срабатываний оценивается следующим образом:  $TPR = (1 - tpr_1) \cdot tpr_2 + tpr_1 \cdot tpr_3$ .

В качестве дополнительного развития описанного алгоритма можно использовать порог разделения положительной выборки при обучении очередного уровня древовидного классификатора. Введение такого порога позволяет уже на этапе обучения строить прореженное дерево решений без существенной потери в качестве детектирования.

Представленный в данном разделе алгоритм позволяет обучать классификаторы, обладающие высокой вариативностью. Действительно, при обучении классического каскада для детекции таких сложных объектов (слабо обобщаемых с точки зрения выбранного признакового пространства) начиная с некоторого момента мощности имеющихся признаков оказывается недостаточно для разделения положительной и отрицательных выборок. И хотя бустинг в конце концов позволит построить очередной уровень, обладающий высокой сложностью (содержащий большое количество слабых классификаторов), с высокой степенью вероятности это приведет к переобучению. Описанный в данном разделе алгоритм построения древовидного классификатора, благодаря априорно установленному ограничению на сложность каждого обучаемого уровня, периодически «перебрасывает» положительные образцы на отрицательную ветку, выполняя тем самым естественную кластеризацию положительной выборки.

Еще одной важной особенностью изложенного алгоритма построения классификатора Виолы и Джонса в виде решающего дерева является возможность дообучения при расширении обучающей выборки. Классическая каскадная структура позволяет дообучение в части уменьшения доли ложных срабатываний за счет добавления новых уровней при расширении отрицательного обучающего набора. Улучшение же доли верного обнаружения возможно только путем полного переобучения. В противоположность каскаду, дообучение древовидного классификатора при расширении обучающей выборки может быть выполнено как с точки зрения улучшения ложных срабатываний, так и точки зрения повышения доли верного обнаружения за счет наличия нескольких путей с положительным ответом.

Далее в главе будет проведено сравнение классического каскадного классификатора и классификатора в виде решающего дерева, построенного в соответствии с описанным алгоритмом с точки зрения эффективности детектирования. Предварительно, выберем метрику для сравнения качества классификаторов, позволяющую перейти от пары чисел (точности и полноты) к одному.

### 3.2 Метрика сравнения классификаторов

Выбор метрики сравнения для построенных классификаторов Виолы и Джонса был произведен исходя из несбалансированности задачи поиска объекта с помощью бинарного классификатора. Дело в том, что в таких задачах поиска объекта при анализе каждого изображения количество отрицательных прецедентов (пустых подокон) превосходит количество положительных прецедентов (подокон с целевым объектом) на несколько порядков. Например, при поиске объекта с характерным линейным размером  $100 \times 100$  пикселей на изображении FullHD разрешения даже без учета возможного масштабирования количество отрицательных прецедентов превосходит количество положительных прецедентов  $(1920 - 100) \cdot (1080 - 100) > 10^6$  раз. В связи с этим выбор метрики, которая при вычислении оперирует количеством верно распознанных отрицательных прецедентов или производных от них функций, не допустимо. В результате, в качестве искомой метрики была выбрана  $F$ -мера.

$F$ -мера (также часто встречается как  $F$ -метрика) [90] представляет собой взвешенное гармоническое среднее между точностью и полнотой:

$$F = \frac{1}{\alpha \cdot \frac{1}{Precision} + (1 - \alpha) \cdot \frac{1}{Recall}}, \quad (26)$$

где  $\alpha \in [0, 1]$  определяет взаимный вес точности и полноты. Обычно,  $F$ -меру записывают в следующем виде:

$$F_\beta = \frac{(\beta^2 + 1) \cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall}, \quad (27)$$

где  $\beta^2 = \frac{1-\alpha}{\alpha}$ ,  $\beta \in [0, \infty)$ . При  $\beta \in [0, 1)$  предпочтение отдается точности, при  $\beta \in (1, \infty)$  больший вес приобретает полнота, а при  $\beta = 1$   $F$ -мера придает одинаковый вес точности и полноте.

### 3.3 Численный эксперимент и результаты

Эффективность разработанного алгоритма оценивалась при решении двух задач: распознавание образов колес на фотографиях в боковом ракурсе и в задаче поиска логотипа платежной системы VISA на изображениях банковских карт.

#### 3.3.1 Распознавание образов колес

Как упоминалось ранее в диссертационной работе, автомобильное колесо в боковом ракурсе – достаточно вариативный объект (см. рисунок 23), следовательно, для качественного детектирования такого объекта необходим сложный классификатор.

В процессе решения задачи на одном и том же обучающем наборе было проведено два эксперимента: обучение классического каскадного классификатора и обучение классификатора в виде решающего дерева в соответствии с представленным в данной главе алгоритмом.

Обучающая выборка состояла из положительной и отрицательной выборок. Положительная выборка состояла из 587 изображений колеса размером 25x33

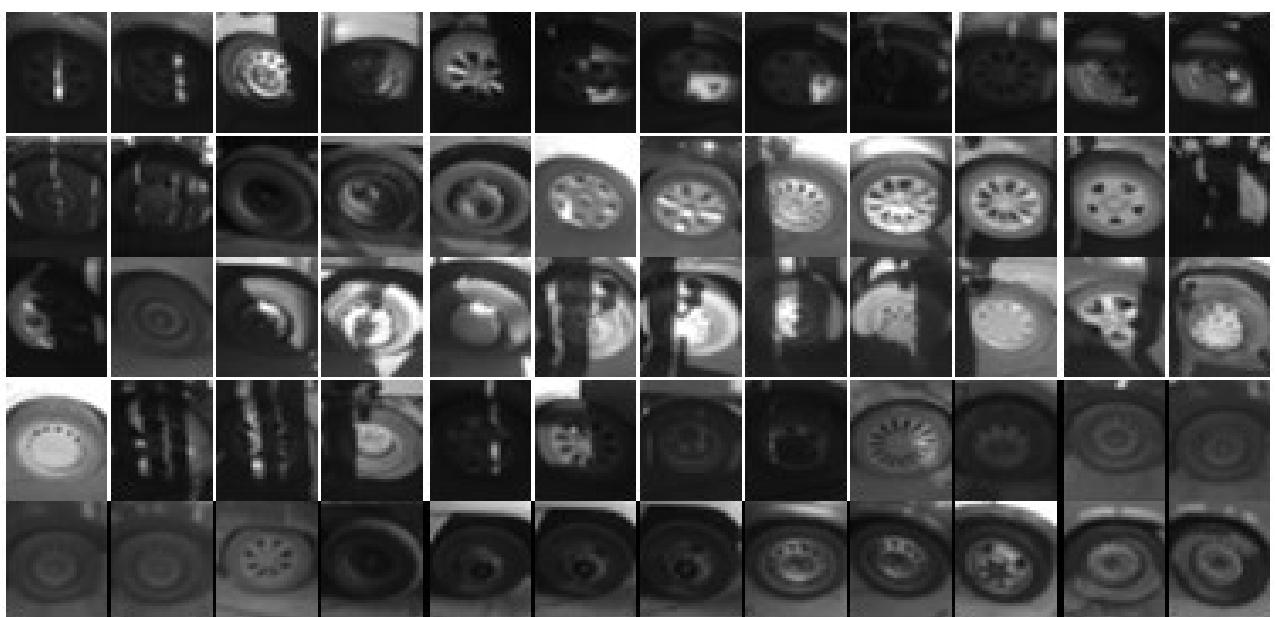


Рисунок 23 — Примеры изображений колес в боковом ракурсе

пикселя. Отрицательная выборка состояла из 4234 изображений размером 136x220 пикселей, на которых отсутствовал образ колеса.

Качество работы обученных классификаторов оценивалось на тестовом наборе, состоящем из 12437 изображений и содержащем 2621 образов колес.

В процессе решения задачи было проведено два эксперимента.

Первый эксперимент E1 представлял собой обучение классического каскада Виолы и Джонса, в результате которого был построен каскад, состоящий из 20 сильных классификаторов.

Второй эксперимент E2 заключался в обучении древовидного классификатора с помощью разработанного алгоритма. В результате был построен древовидный классификатор, содержащий в общей сложности 17 сильных классификаторов (см. рисунок 25а).

Результаты измерений качества обученных детекторов представлены в таблице 4.

Таблица 4 — Качество детектирования каскадного и древовидного классификаторов автомобильного колеса в боковом ракурсе

Эксперимент	TP	FP	FN	Precision	Recall	F-мера
E1	2452	57	169	0,9773	0,9355	0,9559
E2	2587	61	34	0,9770	0,9870	0,9820

### 3.3.2 Распознавание логотипа платежной системы VISA

Несмотря на кажущуюся однотипность, используемые на банковских картах логотипы с точки зрения обучения классификаторов достаточно разнообразны (см. рисунок 24). Это связано как с большой вариативностью фона, так и со способом нанесения и печати логотипа на банковской карте.

В рамках эксперимента обучались два типа классификаторов Виолы и Джонса: классический каскадный классификатор и классификатор в виде решающего дерева в соответствии с представленным в данной главе алгоритмом.

В качестве положительной обучающей выборки использовался набор изображений, состоящий из 1744 предварительно вырезанных цветных образцов логотипов платежной системы. В качестве отрицательной обучающей выборки использовались 7020 фрагментов цветных фотографий банковских карт, не содержащих логотипа платежной системы VISA. Оба детектора использовали в качестве пространства признаков семейство яркостных 2-прямоугольных признаков Хаара. Для упрощения подсчетов каждый уровень (сильный классификатор) каскада и древовидного классификатора состоял из 5 слабых классификаторов.

В качестве тестового набора данных использовались цветные фотографии банковских карт платежной системы VISA разрешением  $800 \times 600$  пикселей. Площадь банковской карты на каждом изображении составляло не менее 60% общей площади фотографии. Объем тестового набора составлял 1176 фотографий.

Было выполнено два эксперимента.



Рисунок 24 — Пример используемых логотипов платежной системы VISA

Первый эксперимент (E1) состоял в обучении классического линейного каскада. Мощность (общее количество уровней, содержащихся в каскаде) обученного каскада составила 10 уровней.

Второй эксперимент (E2) представлял собой процедуру «дообучения» линейного каскада до общей древовидной структуры (см. рисунок 25б). Целью данного эксперимента было повышение качества детектирования за счет «восстановления» потерянных в процессе обучения каскада положительных образцов. В результате был обучен древовидный классификатор, мощность которого составила 21 уровень.

Качество детектирования всех обученных классификаторов было проверено на подготовленном тестовом наборе, состоящем из 1176 фотографий банковских карт платежной системы VISA. Для подсчета верно найденных логотипов использовалась методика, предложенную в рамках соревнования The PASCAL Visual Object Classes (VOC) Challenge [49]. Результаты измерений представлены в таблице 5.

Таблица 5 — Качество детектирования каскадного и древовидного классификаторов логотипа платежной системы VISA

Эксперимент	TP	FP	FN	Precision	Recall	F-мера
E1	1080	111	96	0,9068	0,9184	0,9125
E2	1090	115	86	0,9046	0,9269	0,9156

Таким образом, анализируя полученные результаты, отметим следующие особенности разработанного алгоритма:

- разработанный алгоритм позволяет обучать эффективные классификаторы для вариативных объектов, не проводя предварительную кластеризацию обучающей выборки;
- дообучение линейного каскада до древовидной структуры позволяет повысить полноту детектора, немного потеряв при этом в точности.

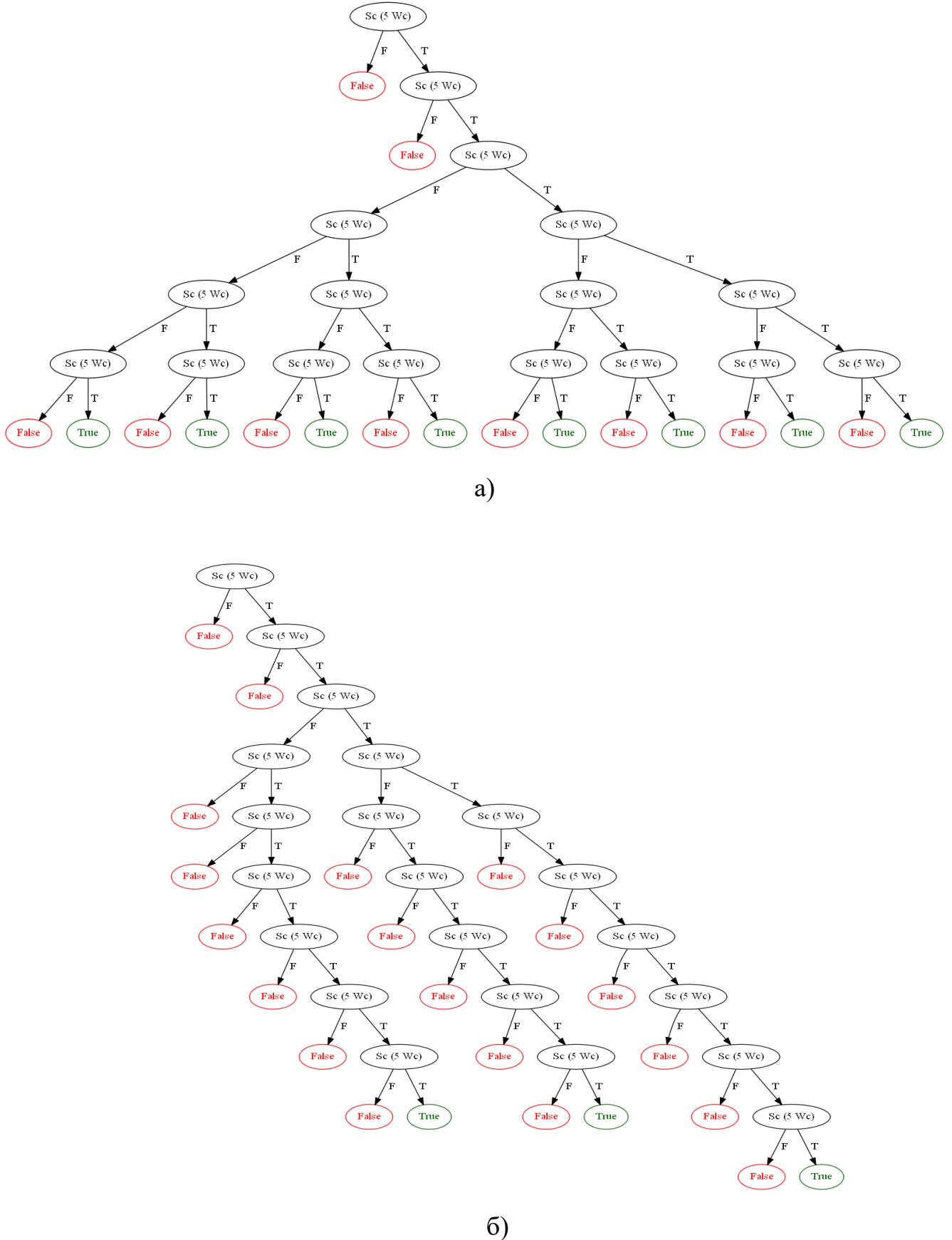


Рисунок 25 — Визуализация древовидных классификаторов: а) классификатор образа колеса; б) классификатор логотипа платежной системы VISA

### 3.4 Выводы по третьему разделу

Представленный в данном разделе алгоритм обучения классификатора Виолы и Джонса в виде решающего дерева сильных классификаторов показал себя эффективным для решения задач построения детекторов сложных объектов. Описанный алгоритм позволяет:

1. Выполнять естественную кластеризацию обучающих данных в процессе обучения для построение более эффективного классификатора.
2. Проводить быстрое дообучение классификаторов при расширении обучающих наборов.

Алгоритм был испытан для решения двух задач:

- обучение детектора образа колеса автомобиля в боковом ракурсе;
- обучение детектора логотипа платежной системы VISA.

Алгоритм продемонстрировал свою эффективность как с точки зрения улучшения качества детектирования.

## **4 Способ ускорения процедуры поиска объектов в видеопотоке путем адаптивного выбора распознающих классификаторов**

В задачах поиска объектов важное значение имеет скорость обработки одного изображения. В простых ситуациях, когда ведется поиск объектов только одного типа метод Виолы и Джонса [7,8] обеспечивает стабильную скорость детектирования, достаточную для решения задачи в режиме реального времени. При этом проблему детектирования нескольких разнородных объектов решают с помощью обучения нескольких независимых классификаторов, каждый для своего типа объектов [37,40,76,79,91–93]. В этом случае, закономерно, процедура поиска объектов замедляется линейно с ростом искомых объектов и часто выходит за допустимые временные рамки.

Обычной с практической точки зрения случаем является ситуация, когда несмотря на глобальную необходимость детектировать несколько разнородных объектов, в каждый момент времени на кадре может присутствовать не более одного объекта. В качестве примера рассмотрим задачу, когда банковский служащий с помощью веб-камеры регистрирует выдаваемые клиентам банковские карты, при этом требуется по нанесенному логотипу определить тип карты и распознать номер. Несмотря на то, что в настоящее время существует более десяти различных видов платежных систем (а, следовательно, более 10 различных логотипов), в каждый момент времени оператор показывает только одну карту. В этом случае поиск всевозможных логотипов на видеокадре заведомо избыточен.

В приведенной в качестве примера задаче возникает проблема выбора «правильного» распознающего классификатора для текущих видеокадров, которая может быть решена аналогично задаче об *n-руком бандите* [94–96], который в отличии от классического игрового автомата («однорукого бандита»), содержит n рычагов, при этом в каждой игре участвует только один рычаг. Можно провести аналогию между задачей выбора «правильного» классификатора и задачей выбора «лучшего» рычага n-ру�ого бандита, если заметить, что успешное детектирование

объекта на очередном кадре в некотором смысле аналогично хорошему вознаграждению в игре, а видеофрагмент – последовательной серии игр.

В данном разделе описан оригинальный метод адаптивного выбора распознающих классификаторов Виолы и Джонса, который использует информацию, накопленную по уже обработанным видеокадрам с целью выбора наилучшего классификатора для текущего видеокадра.

#### **4.1 Задача об n-руком бандите**

Исходная форма задачи об n-руком бандите формулируется следующим образом. Пусть приходится многократно осуществлять выбор одной из  $n$  различных альтернатив (вариантов действий). Каждый выбор влечет за собой получение определенного вознаграждения, зависящего от выбранного действия. Среднее вознаграждение за выбор данного действия называется *ценностью действия*. Целью последовательности действий является максимизация ожидаемого полного вознаграждения за заданный период времени. Часто каждая такая попытка называется *игрой*.

Если бы ценность действий была бы известна, то решение задачи об n-руком бандите было бы тривиальным: в каждой игре необходимо выбирать действие с наивысшей ценностью. Поэтому разумно считать, что точные значения ценности действий не известны, но с каждой игрой могут быть получены оценки таких значений. Тогда в каждый момент времени найдется хотя бы одно действие, для которого такая оценка будет наибольшей (такие действия называются *жадными*). Под *применением* понимается выбор жадного действия в данной игре. Под *изучением* понимается выбор одного из нежадных действий в данной игре с целью более точной оценкой ценности нежадного действия.

Из-за того, что невозможно одновременно изучать и применять знания в рамках единственного действия в игре, часто говорят о «конфликте» между изучением и применением. В каждом конкретном случае выбор между изучением и применением должен зависеть от текущих оценок ценности действий, имеющихся неопределенностей и количества сыгранных (оставшихся) игр.

Перейдем к формальному описанию задачи об n-руком бандите и классических методов выбора баланса между изучением и применением.

Обозначим через  $V_t(a)$  предполагаемое значение ценности действия  $a$  в  $t$ -ой игре. Если к моменту времени  $t$  начала  $t$ -ой игры действие  $a$  было выбрано  $k_a$  раз, что привело к последовательному получению вознаграждения  $r_1, r_2, \dots, r_{k_a}$ , то ценность действия будет оцениваться по следующей формуле:

$$V_t(a) = \begin{cases} \frac{r_1 + r_2 + \dots + r_{k_a}}{k_a}, & k_a > 0 \\ 0, & k_a = 0 \end{cases}. \quad (28)$$

Несмотря на то, что проще всего в каждой игре выбирать действие, обладающее максимальной предполагаемой ценностью, такой подход не приведет к успеху из-за отсутствия процедуры исследования. Проще всего избавиться от этого недостатка следующим способом: будем использовать жадный алгоритм выбора действия большую часть времени, но иногда, например, с вероятностью  $\epsilon$ , выбирать действие произвольно, равномерным образом, вне зависимости от ценности выбиравшего действия. Такого рода модификации жадного правила выбора действия называются  $\epsilon$ -жадными методами [97–99].

Хотя  $\epsilon$ -жадный выбор действия является достаточно эффективным и распространенным способом достижения баланса между изучением и применением в задаче о n-руком бандите, его недостаток состоит в том, что при изучении выбор среди всех действий осуществляется равновероятно. Если для некоторых задач выбор наихудшего действия даже на этапе изучения является нежелательным, то  $\epsilon$ -жадный выбор действия может не подойти. В этом случае можно варьировать вероятность выбора действия посредством некоторой функции от предполагаемых значений действий. Для жадного действия вероятность выбора по-прежнему будет наибольшей, но все остальные действия будут ранжированы согласно их предполагаемой ценности. Такой принцип выбора действия называется правилом на основе операции *softmax* или *softmax-методом* [98,100]. В самом

распространенном варианте softmax-метода используется распределение Гиббса, тогда выбор действия  $a$  в игре  $t$  осуществляется с вероятностью

$$\pi(a) = \frac{e^{\frac{V_t(a)}{\tau}}}{\sum_{b=1}^n e^{\frac{V_t(b)}{\tau}}}, \quad (29)$$

где  $\tau$  – некоторый положительный параметр, называемый температурой. Высокие температуры делают все действия примерно равновероятными. Низкие температуры вызывают более значительное различие в вероятности выбора действий, имеющих разную ценность.

## 4.2 Постановка задачи адаптивного выбора распознающих классификаторов

Перейдем от общего описания задачи об  $n$ -руском бандите к постановке конкретной задачи детектирования объектов. Хотя алгоритм адаптивного выбора распознающего классификатора разрабатывался для работы в режиме реального времени, для замера производительности предлагаемого алгоритма и сравнения с существующими методами в дальнейшем будем рассматривать *видеофрагмент* (ограниченная времененная последовательность кадров).

Итак, рассмотрим следующую задачу. Пусть есть видеофрагмент, содержащий  $T$  кадров. На каждом кадре видеопоследовательности может находиться не более одного объекта, который требуется детектировать. Всего  $N$  различных типов объектов, причем для каждого типа обучен собственный классификатор Виолы и Джонса. Объекты на видеопоследовательности появляются и исчезают естественным путем, не существует мгновенного появления или исчезновения объектов (например, в задаче регистрации банковских карт с помощью веб-камеры, описанной во введение, естественность заключается в плавном последовательном появлении и исчезновении карты в кадре). Необходимо построить алгоритм выбора распознающего классификатора для

каждого кадра видеофрагмента, обеспечивающий максимально точное детектирование объектов.

Пусть  $\mathbb{V} = \{F_i\}_{i=1}^T$  – видеофрагмент, содержащий  $T$  кадров. На каждом кадре может присутствовать не более одного объекта, то есть каждому кадру  $F_i$  можно поставить в соответствие число  $j_i^* \in \{0, 1, \dots, N\}$ , причем  $j_i^* = 0$  соответствует отсутствию какого-либо объекта на рассматриваемом кадре. Пусть  $\mathbb{C} = \{C^j(F) : F \rightarrow \{0, j\}\}_{j=1}^N$  – множество обученных классификаторов Виолы и Джонса. Требуется построить правило адаптивного выбора классификатора для очередного кадра из видеофрагмента  $a : \mathbb{V} \rightarrow \mathbb{C}$  так, чтобы максимизировать количество правильно найденных объектов.

Предположим, на очередном кадре  $F_i$  в качестве действия выбран классификатор  $C_i = C^{j_i^*}$ . Будем считать, что данный кадр распознан правильно, если  $C_i(F_i) = j_i^*$ . Тогда функционал качества для правила адаптивного выбора классификатора  $a$  определяется следующим образом:

$$Q(a, \mathbb{V}) = \sum_{i=1}^T L(a, F_i), \quad (30)$$

где  $L(a, F_i)$  – индикатор ошибки, определяемый следующим образом:

$$L(a, F_i) = \begin{cases} 1, & a(F_i) = j_i^* \\ 0, & a(F_i) \neq j_i^* \end{cases}. \quad (31)$$

### 4.3 Алгоритм адаптивного выбора распознающих классификаторов

Описываемый в данном разделе алгоритм адаптивного выбора распознающих классификаторов реализует идею жадного метода, описанного в разделе 4.1 с учетом особенностей задачи поиска объектов.

Как говорилось выше, в жадном методе учитываются оценки ценности действий, которые вычисляются по вознаграждениям в отдельных играх. И если в задаче об  $n$ -руком бандите вознаграждение возникает естественным образом, то в задаче поиска объектов понятие «вознаграждение» требует отдельного определения. Интуитивно понятно, что вознаграждение за выбор действия (распознающего классификатора) должно быть положительным, если удалось правильно найти объект, и нулевым в противном случае. Однако в момент игры достоверной информации об имеющихся на кадрах объектах нет, так как в противном случае задача поиска объектов на изображении не имеет смысла. Тогда, предполагая, что имеющиеся классификаторы обладают хорошей точностью и полнотой, будем поощрять классификаторы только за найденные на изображении объекты (то есть вознаграждение в очередной игре равно 1 при нахождении объекта на кадре и 0 в противном случае).

Вторым отличием предлагаемого алгоритма от оригинального жадного метода является модификация в части адаптации к нестационарной задаче. Способ вычисления ценности действия по формуле (28) хорошо подходит только для стационарных задач. В нестационарных задачах более поздние вознаграждения обладают большим приоритетом, чем более ранние. Наиболее распространенный способ этого добиться состоит в использовании экспоненциального среднего [101,102]. Тогда ценность действия  $a$  при получении очередного вознаграждения  $r_{k+1}$  определяется по следующей рекурсивной формуле:

$$V_{k+1}(a) = \alpha r_{k+1} + (1 - \alpha)V_k(a), \quad (32)$$

где  $\alpha \in [0; 1]$  – размер шага (чем больше значение  $\alpha$ , тем больший вес в ценности действия имеет новое вознаграждение). С физической точки зрения параметр  $\alpha$  регулирует насколько быстро текущее действие становится жадным.

В рассматриваемой задаче поиска объектов величина параметра  $\alpha$  может быть оценена при наличии некоторых знаний о входной видеопоследовательности. Значение параметра  $\alpha$  должно быть таким, чтобы за время смены объектов в зоне

интереса, текущий жадный классификатор перешел из состояния с большим значением ценности (которое не может превышать 1) в начальное (когда значение ценности близко к нулю). Тогда, если предположить, что смена объекта в зоне видимости происходит за  $M$  кадров (то есть  $M$  кадров в зоне видимости нет никаких объектов), вознаграждение на таких кадрах нулевое (то есть  $r_i = 0$  для  $i = 1, 2, \dots, M$ ), справедливо следующее выражение:

$$\begin{aligned} V_M &= \alpha r_M + (1 - \alpha)V_{M-1} = \alpha r_M + \alpha(1 - \alpha)r_{M-1} + (1 - \alpha)V_{M-2} = \\ &= \alpha r_M + \alpha(1 - \alpha)r_{M-1} + \cdots + \alpha(1 - \alpha)^{M-1}r_1 + (1 - \alpha)^M V_0 = \\ &= (1 - \alpha)^M V_0 + \sum_{i=1}^M \alpha(1 - \alpha)^{M-i}r_i = (1 - \alpha)^M V_0. \end{aligned} \tag{33}$$

Зная, что начальное значение ценности  $V_0 \leq 1$ , а конечное –  $V_M = \epsilon \approx 0$ , то для параметра  $\alpha$  справедливо следующее выражение:

$$\alpha \approx 1 - \sqrt[M]{\epsilon} \tag{34}$$

В формуле (34) в качестве  $\epsilon$  выступает действительное число, близкое к нулю. При реализации алгоритма на современных персональных ЭВМ в качестве такого  $\epsilon$  можно использовать так называемый машинный эпсилон (англ. Machine epsilon), который, например, для типа double (64 бита) в языках Си и C++ равен  $\epsilon = 2^{-52} \approx 2.20 \cdot 10^{-16}$ . Важно, чтобы выбор  $\epsilon$  был строго согласован с выбором типа данных, используемых в программной реализации для хранения величин  $V_i$ ,  $r_i$  и  $\alpha$ . В противном случае, выбор классификатора с максимальным текущим значением ценности будет вычисляться неверно.

С учетом вышесказанного алгоритм адаптивного выбора классификатора представим в виде следующей последовательности шагов.

*Шаг 1 (Инициализация).* Пусть есть  $N$  классификаторов. Зададим начальное значение ценности для классификатора  $V_1 = V_2 = \cdots = V_N = 0$ . Оценим исходя из имеющихся данных значение параметра  $\alpha$ .

*Шаг 2.* Выберем классификатор  $C$  с максимальным текущим значением ценности  $V_{curr}$ . Если таких классификаторов несколько, выберем один из них произвольным образом.

*Шаг 3.* Применим на очередном кадре  $F$  выбранный классификатор  $C$ . Определим выигрыш классификатора следующим образом:

$$r = \begin{cases} 1, & \text{если на кадре } F \text{ найден объект} \\ 0, & \text{если на кадре } F \text{ не найден объект} \end{cases} \quad (35)$$

*Шаг 4.* Обновим значение ценности  $V_{new}$  для классификатора  $C$  по следующей формуле:

$$V_{new} = \alpha r + (1 - \alpha)V_{curr} \quad (36)$$

*Шаг 5.* Перейдем на шаг 2, если есть еще кадры для поиска объектов. Иначе заканчиваем обработку

Реализация алгоритма в виде блок-схемы показана на рисунке 26.

#### 4.4 Численный эксперимент и результаты

Эффективность описанного алгоритма оценивалась в рамках решения задачи определения типа банковской карточки в видеопотоке. Оператор фотографирует с помощью web-камеры банковскую карту, а система распознавания должна определить тип карты по имеющемуся логотипу. Всего тестовый видеофрагмент содержал 29 представлений банковских карт следующих пяти типов: VISA, MasterCard, American Express, Discover и UnionPay. Общее количество кадров видеофрагмента составило 1116. Минимальный интервал между демонстрациями соседних карт составляет 16 изображений. Распределение кадров по содержанию объектов, а также общее количество кадров тестового видеофрагмента представлено в таблице 6.

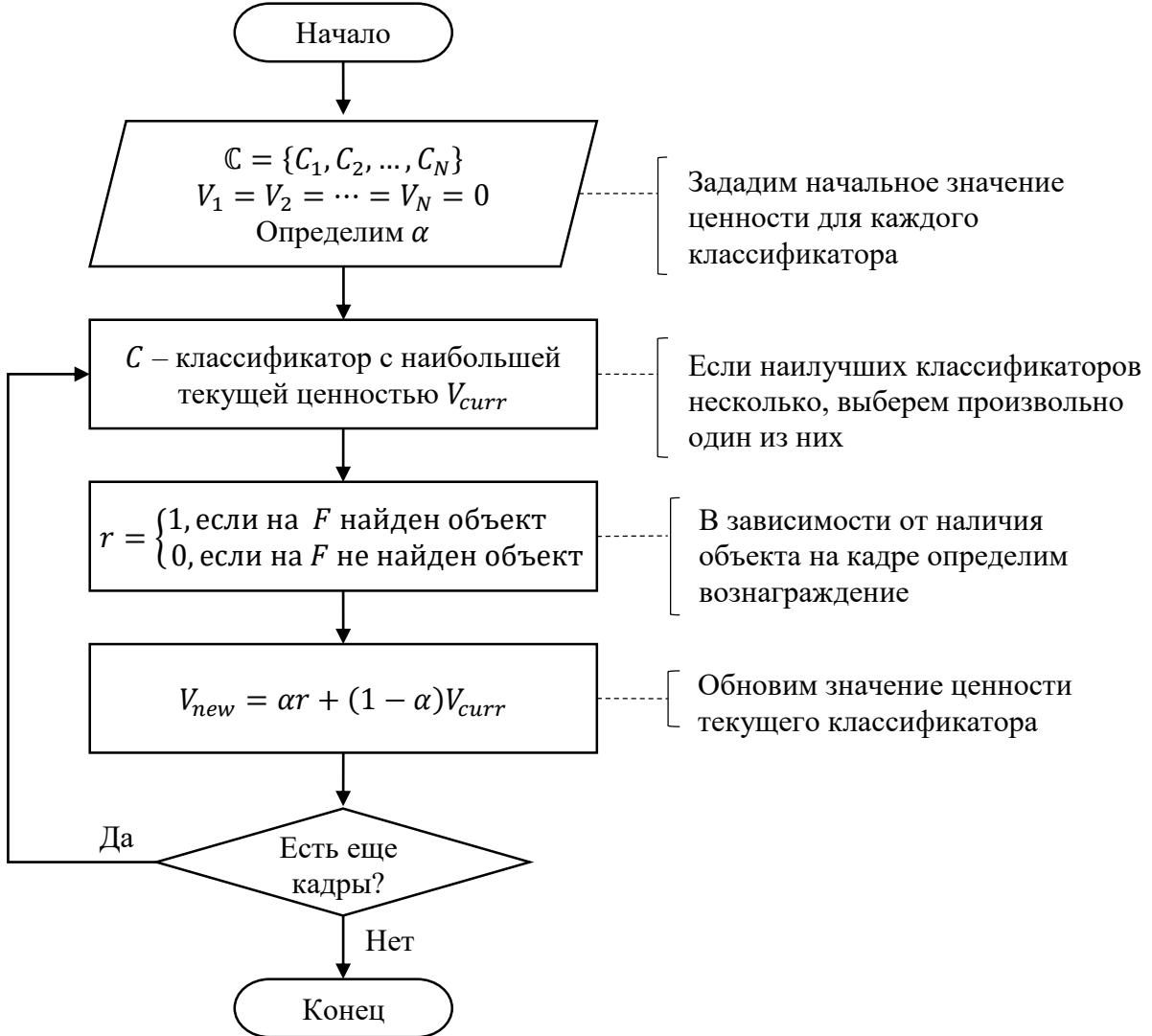


Рисунок 26 — Блок-схема алгоритма адаптивного выбора  
распознающих классификаторов

Детектирование карт выполнялось путем использования классификаторов логотипов соответствующих платежных систем, построенных с помощью инструментария objed (раздел 5, приложение А). В качестве положительной обучающей выборки использовались изображения логотипов, вырезанные из фотографий соответствующих банковских карт (данные о количестве фотографий представлены во втором столбце таблицы 7). Из-за малого количества обучающих образцов использовалась аугментация (синтез обучающих данных на основе существующих) [103–105] данных с помощью геометрических и яркостных преобразований: поворот изображения логотипа вокруг центра на углы  $\pm 5^\circ$  и  $\pm 3^\circ$ ,

и применение гамма-коррекции с коэффициентами 0,5 и 2,0, что позволило увеличить обучающий набор изображений в 15 раз (см. третий столбец таблицы 7). В частности, использовался поворот изображения логотипа вокруг центра на углы  $\pm 5^\circ$  и  $\pm 3^\circ$ , и применение гамма-коррекции с коэффициентами 0,5 и 2,0, что позволило увеличить обучающий набор изображений в 15 раз (см. третий столбец таблицы 7).

Таблица 6 — Содержимое тестового видеофрагмента

<b>Тип объекта</b>	<b>Количество демонстраций</b>	<b>Общее количество кадров</b>
VISA	16	302
MasterCard	8	139
American Express	2	37
UnionPay	1	17
Discover	2	34
Кадры без банковской карты	-	587
<b>Итого:</b>	<b>29</b>	<b>1116</b>

Таблица 7 — Состав положительной обучающей выборки

<b>Тип объекта</b>	<b>Количество изображений (изначально)</b>	<b>Количество изображений (после аугментации)</b>
VISA	190	2850
MasterCard	160	2400
Am. Express	81	1215
UnionPay	8	120
Discover	9	135
Без карты	1792	-
<b>Итого:</b>	<b>2240</b>	<b>6720</b>

В качестве отрицательной обучающей выборки использовались фрагменты фотографий банковских карточек, на которых отсутствовал логотип платежной системы. Общее объем отрицательной обучающей выборки составил 1814 изображений.

Состав тестовой выборки и показатели качества работы полученных классификаторов представлены в таблице 8.

Таблица 8 — Характеристики классификаторов логотипов

Тип объекта	Количество тестовых образцов	Точность (Precision)	Полнота (Recall)
VISA	90	0,99	0,96
MasterCard	105	0,97	0,97
American Express	41	0,95	0,93
UnionPay	4	1,00	0,75
Discover	5	0,83	1,00

Чтобы оценить эффективность, качество работы представленного алгоритма выбора распознающих классификаторов (далее – АВРК) со значением параметра  $\alpha = 0.9$  было сравнено с качеством работы следующих трех сопоставимых по скорости методов: поочередным методом (обученные классификаторы применяются к кадрам по очереди),  $\epsilon$ -жадным методом при  $\epsilon = 0.1$  и softmax-методом при  $\tau = 0.1$ . Параметры для  $\epsilon$ -жадного метода и softmax-метода подобраны таким образом, чтобы обеспечить максимальное значение функционала качества на  $Q$  на тестовой выборке. Значение параметра  $\alpha$  для описанного в данном разделе алгоритма было рассчитано по формуле (34), опираясь на тот факт, что минимальный интервал между демонстрациями соседних карт  $M$  составляет 16 кадров, а значение  $\epsilon = 2^{-52} \approx 2.20 \cdot 10^{-16}$  (что соответствует значению машинного эпсилона для типа double (64 бита) в языках Си и C++). Кроме того, так как был проведен эксперимент по применению к каждому изображению видеопотока всех обученных классификаторов (одновременный метод), который

использовался как теоретическая оценка максимально возможного результата детектирования с помощью имеющихся каскадных классификаторов, так как в эксперименте использовались не «идеальные» классификаторы, для которых значение точности и полноты меньше единицы.

Графические характеристики и поведение различных методов показано на рисунке 27. В таблице 9 представлено значение функционала качества  $Q$ , а также отношение функционала качества к общему числу обработанных кадров  $Q_r$  для отдельных интервальных точек. Полученные результаты свидетельствуют о том, что методы решения задачи об  $n$ -руском бандите применимы для повышения быстродействия алгоритмов поиска объектов (простой поочередный алгоритм демонстрирует результаты существенно хуже всех остальных методов). Кроме того, важно отметить, что разработанный алгоритм АВРК опережает  $\varepsilon$ -жадный и softmax методы на всех уровнях эксперимента.

Помимо значения функционала качества  $Q$  с практической точки зрения отдельный интерес представляет доля верно распознанных изображений, непосредственно содержащих карту.

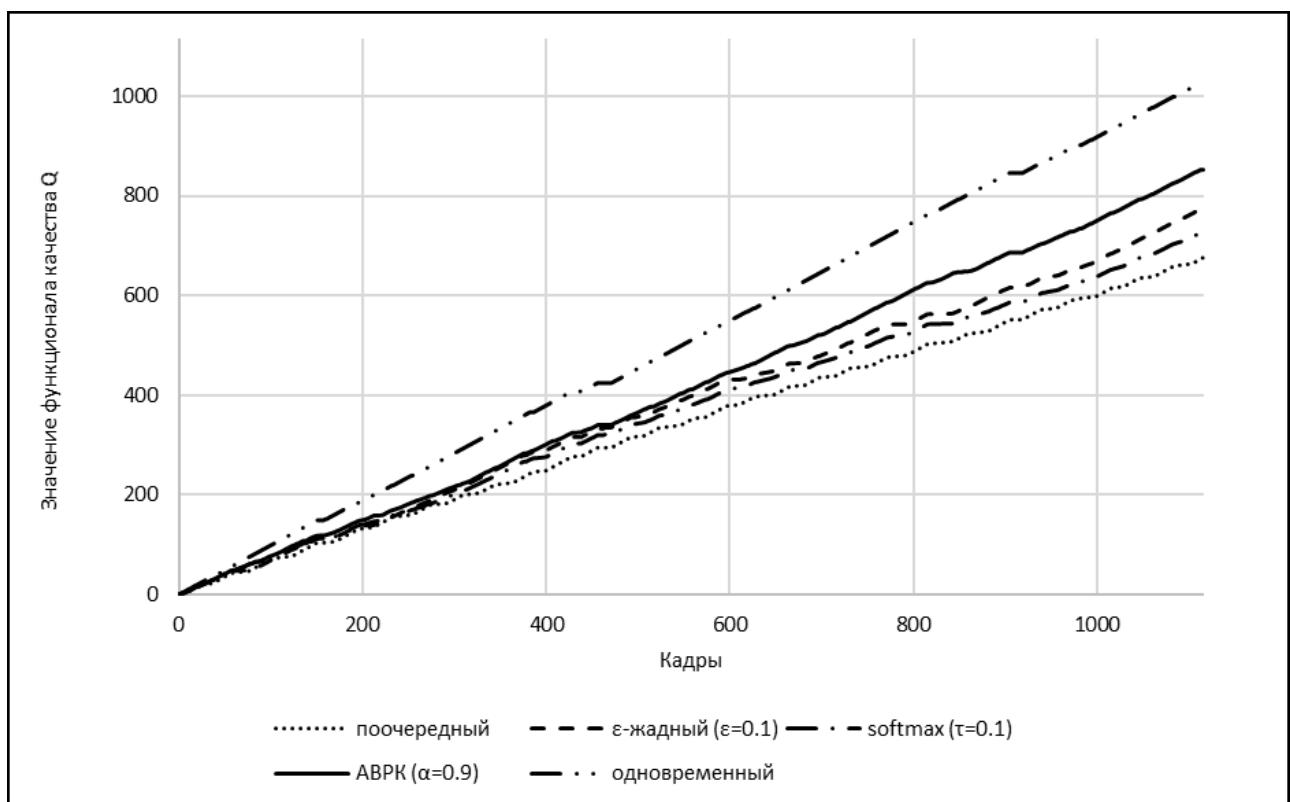


Рисунок 27 — График зависимости значения функционала качества  $Q$  от номера кадра

Таблица 9 — Значение функционала качества в отдельных точках проверочной видеопоследовательности

Метод	$Q(Q_r)$					
	200 кадров	400 кадров	600 кадров	800 кадров	1000 кадров	1116 кадров
поочередный	133 (0.66)	249 (0.62)	379 (0.63)	487 (0.61)	599 (0.60)	676 (0.61)
$\epsilon$ -жадный ( $\epsilon = 0.1$ )	138 (0.69)	289 (0.72)	432 (0.72)	548 (0.68)	670 (0.67)	773 (0.69)
softmax ( $\tau = 0.1$ )	140 (0.70)	276 (0.69)	411 (0.68)	527 (0.66)	637 (0.64)	724 (0.65)
АВРК ( $\alpha = 0.9$ )	149 (0.74)	301 (0.75)	446 (0.74)	611 (0.76)	750 (0.75)	852 (0.76)
одновременный	188 (0.94)	380 (0.95)	550 (0.92)	748 (0.94)	919 (0.92)	1030 (0.92)

В тестовой выборке таких количество таких изображений составляет 529 шт. Как уже неоднократно упоминалось в работе, связано это с тем, что в промышленных распознающих системах высокое значение полноты детектора ценится сильнее, чем точность. На рисунке 28 изображена диаграмма, показывающая количество верно распознанных изображений, содержащих банковскую карту с помощью различных методов. Из представленных данных видно, что предложенный алгоритм адаптивного выбора распознающего классификатора существенно превосходит  $\epsilon$ -жадный и softmax методы, приближаясь по данному параметру к методу одновременного применения всех каскадов.

В заключение оценим насколько алгоритм адаптивного выбора распознающего классификатора, описанный в данном разделе, повышает производительность по сравнению с одновременным методом. Так как каждый классификатор представляет собой с вычислительной точки зрения коллекцию



Рисунок 28 — Диаграмма количества верно распознанных изображений, содержащих банковскую карту с помощью различных методов

признаков Хаара, то объективной единицей измерения производительности можно считать суммарное количество вычисленных признаков при обработке тестовой видеопоследовательности указанным методом. Такая статистика представлена в таблице 10. Из полученных данных видно, что представленный в данном разделе алгоритм практически в 4 раз быстрее, чем одновременный метод.

Таблица 10 — Производительность различных методов на проверочной видеопоследовательности

Метод	Общее количество вычисленных признаков на видеопоследовательности	Среднее количество вычисленных признаков на одном изображении
одновременный	$12.82 \cdot 10^9$	$11.49 \cdot 10^6$
ABPK ( $\alpha = 0.9$ )	$3.36 \cdot 10^9$	$3.01 \cdot 10^6$

#### 4.5 Выводы по четвертому разделу

В задачах поиска объектов в видеопотоке важное значение имеет скорость обработки отдельного кадра. Существуют задачи, для которых даже применение

быстрых алгоритмов, таких как метод Виолы и Джонса, не обеспечивает необходимой скорости. В качестве важного с практической точки зрения примера можно привести проблему многоклассовой детекции объектов в видеопотоке с помощью классификаторов Виолы и Джонса.

В данном разделе разработан алгоритм адаптивного выбора распознающего классификатора, в основе которого лежит стратегия жадного выбора действия в задаче  $n$ -руского бандита. Эффективность описанного алгоритма была продемонстрирована на примере поиска и распознавания логотипов банковских карт в видеопотоке путем сравнения с классическими для задачи  $n$ -руского бандита стратегиями:  $\epsilon$ -жадным методом и softmax-методом.

Представленный в работе алгоритм может быть эффективно применен в таких задачах, как локализация и идентификация документов, распознавание элементов дорожной сцены, а также для решения ряда других задач идентификации ригидных объектов в потоках гетерогенных данных.

## **5 Проектирование и реализация программного комплекса детектирования объектов методом Виолы и Джонса**

Существующие в настоящее время и доступные для свободного использования программные решения поиска объектов методом Виолы и Джонса являются частью инструментариев, используемых преимущественно для академических целей. Наиболее популярными считаются:

- OpenCV: Object Detection;
- Computer Vision System Toolbox.

Несмотря на широкую популярность данных программных комплексов поиска объектов методом Виолы и Джонса, они обладают рядом недостатков, препятствующих их промышленному использованию. В данном разделе описано проектирование и программная реализация программного комплекса поиска объектов методом Виолы и Джонса, пригодная для эксплуатации в индустриальном масштабе.

### **5.1 Существующие программные решения детектирования объектов методом Виолы и Джонса**

Пакет OpenCV: Object Detection [106–108] является частью библиотеки компьютерного зрения с открытым исходным кодом OpenCV и предназначен для обучения каскадных классификаторов поиска объектов методом Виолы и Джонса. Пакет состоит из двух частей: инструментарий обучения каскадного классификатора (`opencv_traincascade`) и модуль применения обученного классификатора к изображению (`cv::CascadeClassifier`). Инструментарий `opencv` позволяет управлять следующими параметрами обучения:

- указывать семейство признаков (поддерживаются признаки Хаара, локальные бинарные шаблоны LBP);
- использовать расширенный комплект признаков Хаара (поворнутые на 45 градусов признаки Хаара);

– задавать алгоритм обучения сильного классификатора (Discrete AdaBoost, Real AdaBoost, LogitBoost, Gentle AdaBoost);

– задавать параметры обучения сильного классификатора (максимальное количество слабых классификаторов, целевое качество, вид алгоритма обучения);

– задавать количество обучаемых уровней каскада.

Альтернативой инструментарию OpenCV: Object Detection является пакет Computer Vision System Toolbox [109], включенный в продукт MATLAB & Simulink. Пакет включает в себя различные алгоритмы выявления деталей, детектирования движения, обнаружения и отслеживания объектов, алгоритмы регистрации изображений и работы со стереозрением. В части обнаружения объектов Computer Vision System Toolbox поддерживает несколько: поиск по шаблонам, анализ блоков (сегментация и определение свойств блока для идентификации объектов, представляющих интерес) и алгоритм Виолы и Джонса. Аналогично предыдущему рассмотренному инструментарию, пакет Computer Vision System Toolbox позволяет управлять следующими параметрами обучения:

– указывать семейство признаков (поддерживаются признаки Хаара, гистограммы направленных градиентов HOG, локальные бинарные шаблоны LBP);

– задавать параметры обучения сильного классификатора (при этом вид обучающего алгоритма не специализирован);

– задавать количество обучаемых уровней каскада.

Несмотря на широкую популярность рассмотренных программных комплексов поиска объектов методом Виолы и Джонса, они обладают следующими недостатками:

1. Рассмотренные программные комплексы «тяжеловесны» для подключения промышленным распознавающим системам, а модуль применения обученных детекторов к изображению трудноотделим (в случае Computer Vision System Toolbox не отделим вовсе) от исходного инструментария.

2. Допустимые параметры не обеспечивают необходимой гибкости настройки процесса обучения. Несмотря на то, что оба рассмотренных инструментария позволяют выбирать семейство признаков, более точная

спецификация вида признаков не поддерживается (например, нельзя сузить множество признаков Хаара до 2-прямоугольных или задать радиус для локальных бинарных шаблонов). Указание вида слабых классификаторов не доступно изначально, в обоих инструментариях используется слабый классификатор типа «ступенька» (decision stump). Аналогично, рассмотренные программные комплексы позволяют обучить только каскадный высокоуровневый классификатор.

3. Оба программных комплекса не позволяют обучить классификатор, оперирующий разнотипными признаками (например, комбинирующий HOG и признаки Хаара).

4. Рассмотренные инструментарии не обеспечивают полный цикл обучения и оценку качества обученных классификаторов, предоставляя пользователю только часть, связанную непосредственно с обучением. Предполагается, что работы, связанных с подготовкой обучающих наборов, визуализацию результатов обучения, тестирование обученных классификаторов, пользователь выполняет с помощью других доступных инструментов.

5. Рассмотренные программные комплексы не обеспечивают возможность алгоритмического расширения из-за закрытого исходного кода (Computer Vision System Toolbox) или большого количества сложноструктурированных элементов (OpenCV: Object Detection).

6. Оба программных комплекса не позволяют выполнять «дообучение» уже существующего классификатора. Данная функциональность полезна по следующим соображениям:

– процедура обучения классификатора занимает длительное время (измеряется часами, а иногда даже сутками), поэтому при возникновении технического сбоя должна быть возможность продолжить обучения (дообучения) последней сохраненной версии;

– в процессе тестовой эксплуатации («пилотирования») обученного классификатора обычно коллекционируются ошибки детекции, которые должны быть использованы для дообучения классификатора.

## 5.2 Технические и функциональные требования к разрабатываемому программному комплексу детектирования объектов

К разрабатываемому программному комплексу детектирования объектов методом Виолы и Джонса предъявлялись следующие требования.

### **Требования к функциональным характеристикам**

Программный комплекс детектирования объектов методом Виолы и Джонса должен состоять из следующих функциональных модулей:

- модуль, обеспечивающий полный цикл обучения и тестирования классификаторов (далее – «Модуль обучения»);
- модуль, обеспечивающий применение обученного классификатора к изображению (далее – «Модуль воспроизведения»).

#### *Требования к Модулю обучения*

Модуль обучения должен удовлетворять следующим требованиям:

– включать инструментарий подготовки наборов данных для обучения и тестирования, инструментарий обучения (построения) классификаторов заданного типа, инструментарий визуализации работы классификатора, инструментарий оценки качества работы классификатора, инструментарий анализа ошибок классификатора;

– обеспечивать возможность гибкой настройки процедуры обучения, включая:

- возможность задания типа и характеристик признаков;
- возможность одновременного использования признаков разных типов (семейств);
- возможность выбора вида слабого классификатора;
- возможность тонкой настройки процедуры обучения сильного классификатора;
- возможность выбора алгоритма обучения;
- возможность выбора вида высокоуровневого классификатора;

- обеспечивать возможность легкого добавления новых признаков и классификаторов (то есть, обеспечивать возможность алгоритмического расширения);
- обеспечивать возможность дообучения имеющегося классификатора;
- содержать реализацию представленных в данной работе контурных признаков Хаара;
- содержать реализацию представленной в данной работе высокоуровневой структуры классификатора Виолы и Джонса в виде решающего дерева сильных классификаторов.

### Требования к Модулю воспроизведения

Модуль воспроизведения должен удовлетворять следующим требованиям:

- иметь C++ интерфейс;
- не использовать тяжеловесные библиотеки (например, Boost, OpenCV, Leptonica и аналогичные им);
- обеспечивать возможность компиляции исходных текстов Модуля воспроизведения, а также используемых сторонних библиотек с использованием компиляторов GCC (версии 4.8 и выше), Microsoft Visual C++ (версии 14.0 и выше), MinGW (версии 4.8 и выше);
- содержать реализацию представленных в данной работе контурных признаков Хаара;
- содержать реализацию представленной в данной работе высокоуровневой структуры классификатора Виолы и Джонса в виде решающего дерева сильных классификаторов.

### **Требования к надежности**

Модуль обучения должен выполнять автоматическое сохранение классификатора при завершении очередной стадии обучения, а также обеспечивать возможность загрузки таких классификаторов с дальнейшим обучением.

Модуль воспроизведения должен содержать обработку всех возможных исключений и не допускать выброса их за пределы модуля.

## **Требования к составу и параметрам технических средств**

Модуль обучения должен нормально функционировать на ПК со следующими минимальными аппаратными и программными характеристиками:

- 1 процессор с архитектурой x86-64, тактовая частота процессора 1,8 ГГц, процессор обладает 2 ядрами;
- оперативная память, занимаемая Модулем обучения: не более 512 МБ;
- операционная система: Microsoft Windows 7 и выше.

## **Требования к информационной и программной совместимости**

Разработка программного комплекса детектирования объектов методом Виолы и Джонса должна вестись на языке программирования C++ версии стандарта C++11.

Модуль обучения классификаторов должен обеспечивать параллельные вычисления для повышения производительности на многоядерных процессорах.

### **5.3 Разработка программного комплекса objed**

В рамках диссертационной работы разработан инструментарий objed – программный модуль поиска объектов методом Виолы и Джонса, удовлетворяющий сформулированным в разделе 5.2 техническим и функциональным требованиям. В данном разделе описаны подходы, применяемые при разработке инструментария objed, представлена архитектура, а также описаны способы работы с инструментарием.

#### **5.3.1 Выбор средств разработки**

В качестве среды разработки была выбрана Microsoft Visual Studio (MSVS) 2015 в редакции Community [110]. Данная редакция распространяется бесплатно и предназначена в первую очередь для работы над различными научно-исследовательскими проектами и проектами с открытым исходным кодом. Несмотря на свободное распространение, она предоставляет усовершенствованные

средства для отладки, профилирования и диагностики кода, позволяет управлять исходным кодом в репозиториях Git (включая ресурс GitHub), а также обеспечивает доступ к широкому набору бесплатных инструментов разработчика.

Для обеспечения кроссплатформенной автоматизации сборки был выбран генератор проектных файлов CMake [111]. Данная утилита способна создавать «родные» сборочные файлы для различных сред разработки под разными операционными системами (MS Visual Studio, MinGW, C++ Builder, GNU и др.). Кроме того, CMake стремится максимально использовать фирменные средства генерации сборочных файлов – например, для генерации сборочных файлов проекта Qt используется qmake.

В качестве основного языка разработки был выбран C++ [112,113]. Язык C++ широко используется для разработки вычислительно эффективных программ, операционных систем и прочих других системных утилит. Язык сочетает в себе свойства как высокоуровневых, так и низкоуровневых языков, поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование.

В качестве основного фреймворка при создании приложений с графическим пользовательским интерфейсом (GUI), а также для работы с файловой системой, данными в формате XML и JSON использовался Qt [114,115]. Qt позволяет запускать написанное с его помощью программное обеспечение в большинстве современных операционных систем под разными процессорными архитектурами путём простой компиляции программы для каждой системы без изменения исходного кода.

Для реализации многопоточности использовались средства OpenMP [116]. OpenMP реализует параллельные вычисления с помощью многопоточности, в которой «главный» (master) поток создает набор подчиненных (slave) потоков и задача распределяется между ними. При этом задачи, выполняемые потоками параллельно, также как и данные, требуемые для выполнения этих задач, описываются с помощью специальных директив препроцессора соответствующего языка — прагм.

### 5.3.2 Архитектура программного комплекса objed

С учетом технических требований была разработана следующая архитектура программного комплекса обучения классификаторов.

Модуль обучения objed состоит из следующих программных инструментов:

1. *Инструмент разметки набора изображений и анализа результатов детектирования objedmarker* представляет собой программу с графическим пользовательским интерфейсом, выполняющую две основные задачи: разметка набора изображений на предмет наличия объекта и сравнение результатов детектирования с эталонным (то есть, созданными вручную оператором).

2. *Инструмент подготовки обучающего набора изображений objedcutcli* представляет собой программу с интерфейсом командной строки и предназначен для подготовки обучающих наборов изображений на основе размеченного набора изображений

3. *Инструмент обучения классификатора objedtraincli* представляет собой программу с интерфейсом командной строки и предназначен для обучения бинарных классификаторов типа Виолы и Джонса, описанных в данной работе (включая контурные признаки Хаара, обучение высокоуровневой структуры классификатора Виолы и Джонса в виде решающего дерева сильных классификаторов).

4. *Инструмент визуального контроля работы классификатора objedcheck* представляет собой программу с графическим пользовательским интерфейсом, предназначенную для органолептического контроля работоспособности обученных классификаторов и построенных на их базе детекторов.

5. *Инструмент анализа качества работы классификатора objedruncli* представляет собой программу с интерфейсом командной строки и предназначен для тестирования обученных классификаторов и построенных на их базе детекторов на размеченном тестовом наборе изображений.

Руководство пользователя инструментария обучения и тестирования классификаторов objed представлено в приложении А.

При разработке Модуля обучения применялся подход компонентно-ориентированного программирования. Исходный код (в виде терминальных классов и отдельных функций), предназначенный для повторного использования, объединялся в специальные компоненты и интегрировался в таком виде в отдельные элементы Модуля обучения.

Модуль воспроизведения представлен в виде отдельной программной библиотеки `objed`, содержащей 3 заголовочный файла и обладающей C++ интерфейсом. Библиотека `objed` использует в качестве единственной внешней зависимости библиотеку `jsoncpp` [117].

Модуль воспроизведения содержит имплементацию следующих сущностей метода Виолы и Джонса:

- 1-прямоугольные, 2-прямоугольны и 3-прямоугольные признаки Хаара, вычисляемые как поверх яркостных изображений, так и поверх карты направленных границ;
- слабый классификатор вида «ступенька»;
- слабый классификатор вида «кусочно-заданная функция»;
- сильный классификатор (как линейный классификатор над слабыми классификатором);
- высокоуровневый классификатор в виде каскада сильных классификаторов;
- высокоуровневый классификатор в виде решающего дерева сильных классификаторов;
- детекторы, решающие с помощью заданного бинарного классификатора, а также заданных параметров масштабирования и сдвига, задачу поиска целевого объекта на изображении.

При разработке Модуля воспроизведения использовались следующие шаблоны проектирования [118]:

- фабричный метод (*factory method*);
- пул объектов (*object pool*).

Диаграмма основных классов модуля `objed` представлена на рисунке 29.

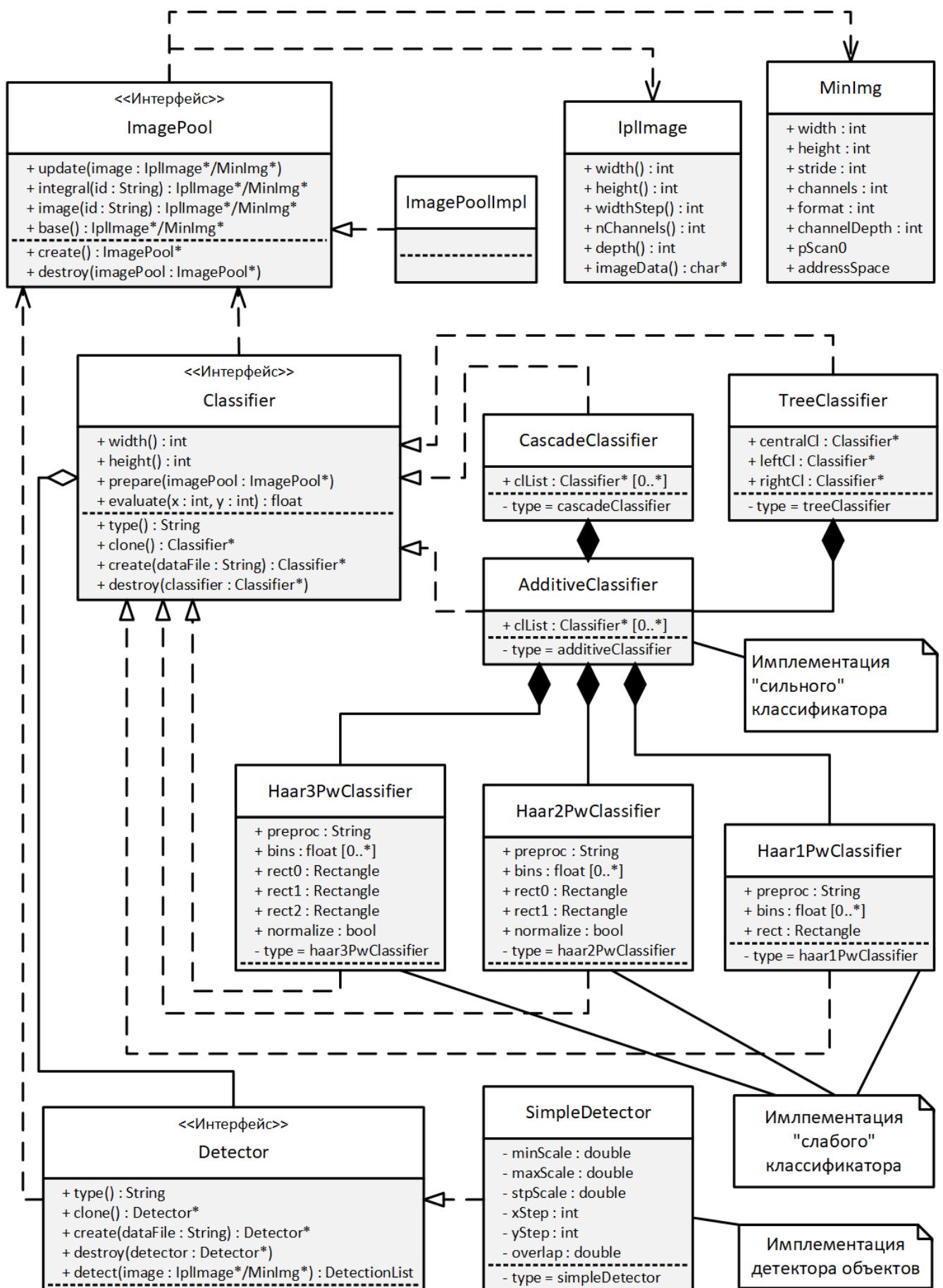


Рисунок 29 — UML диаграмма библиотеки objed (основных компонентов)

## 5.4 Использование программного комплекса objed в индустриальных распознающих системах

Разработанный в рамках данной диссертационной работы программный комплекс objed активно применяется в ряде индустриальных распознающих систем для решения различных задач поиска объектов.

С помощью инструментария objed для автоматического классификатора транспортных средств AKTC-4 [119–121] были обучены детекторы колес в ракурсе «вид сбоку» для последующего построения подсистемы определения колесных пар проезжающего транспортного средства (см. рисунок 30). С помощью разработанного инструментария было размечено более 20 000 изображений колес, а также обучен древовидный классификатор, использующий контурные признаки Хаара. Система AKTC-4 не имеет мировых аналогов, работает на данный момент на 200 полосах российских дорог с ежедневной пропускной способностью около 500 тысяч автомобилей.



Рисунок 30 — Пример визуализации работы системы автоматической классификации транспортных средств AKTC-4

Другим индустриальным распознающим решением, использующим разработанный инструментарий objed, является интеллектуальная система автономного вождения Cognitive C-Pilot [122,123]. В рамках данной системы на базе программного комплекса objed была построена подсистема детектирования попутных и двигающихся навстречу транспортных средств, а также система локализации и распознавания основных дорожных знаков (см. рисунок 31). Построенные детекторы целевых объектов отличались высокой скоростью работы (достаточной для обработки видеопотока в разрешении FullHD с частотой не менее 10 кадров в секунду), а также высокой устойчивостью к различным погодным условиям.

Помимо индустриальных распознающих систем, связанных с транспортными средствами, разработанный инструментарий применяется в программных комплексах распознавания документов, удостоверяющих личность, Smart PassportReader [124] и Smart IDReader [125], а также серии программно-аппаратных комплексов для сканирования и распознавания документов граждан PassportBox [126], построенных с использованием указанного программного обеспечения. В рамках данных распознающих систем с помощью программного комплекса детектирования объектов objed решается задача локализации, ориентации и идентификации документов фиксированной геометрией как на сканах, так и на

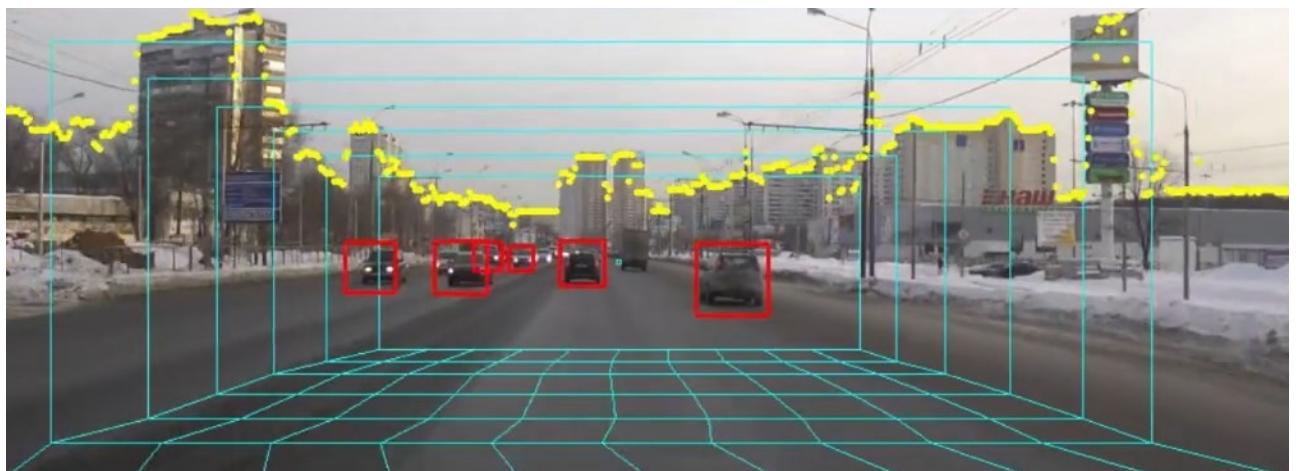


Рисунок 31 — Пример визуализации работы интеллектуальной системы автономного вождения C-Pilot

отдельных кадрах видеопоследовательности. С помощью objed успешно обучены и подключены в систему следующие типы документов: вторая и третья страницы внутреннего паспорта гражданина РФ, водительское удостоверение гражданина РФ (образцов 1972 года, 1998 года, 2011 года), СНИЛС (2 различных типа), миграционная карта. Благодаря кроссплатформенности модуля воспроизведения objed полученное программное решение работает как на стационарных ПК, так и на мобильных устройствах (см. рисунок 32). Системы распознавания Smart PassportReader и Smart IDReader интегрированы в информационные системы крупных российских банков (Почта Банк, Тинькофф Кредитные Системы, БинБанк), страховых компаний (АльфаСтрахование, Ренессанс Страхование, УралСиб Страхование, Сбербанк Страхование), авиакомпаний (Аэрофлот), за счет чего общее количество пользователей систем распознавания Smart PassportReader и Smart IDReader превышает 10 миллионов человек.

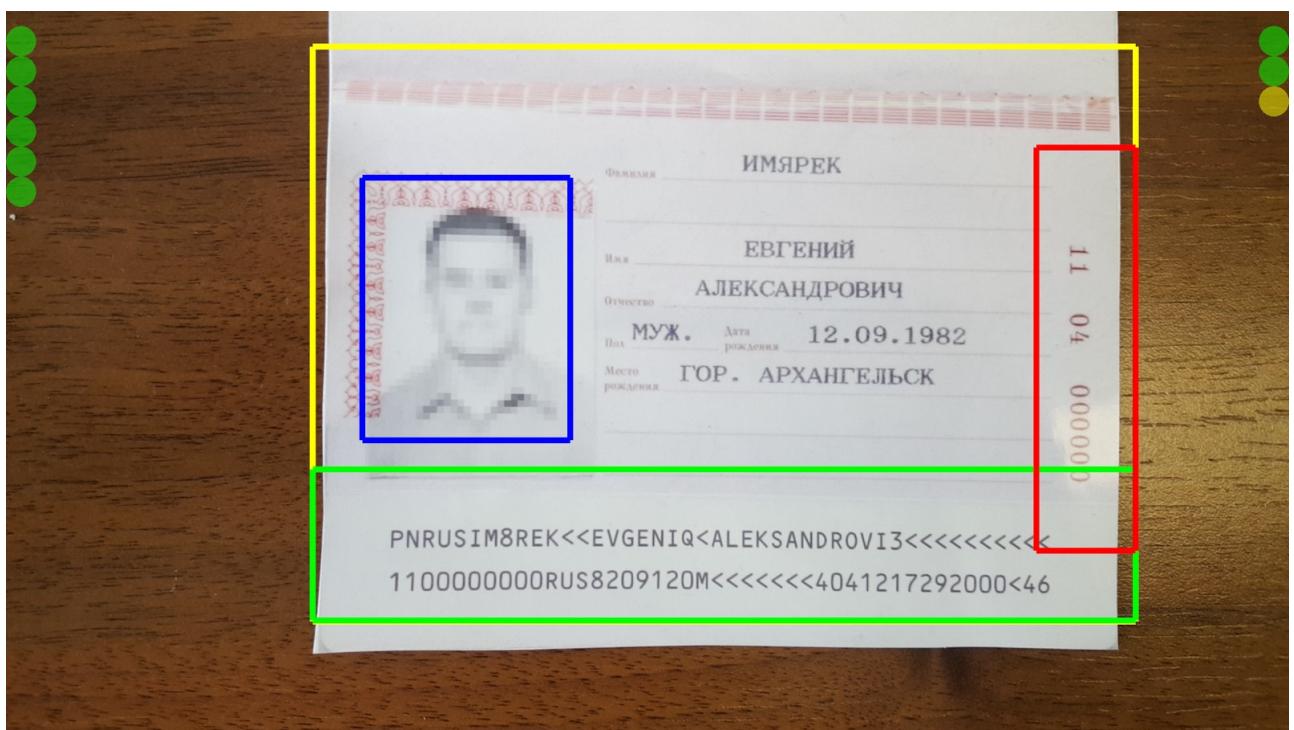


Рисунок 32 — Пример визуализации работы системы распознавания документов Smart PassportReader

## 5.5 Выводы по пятому разделу

В рамках данной диссертационной работы был создан программный комплекс `objed`, обладающий следующими особенностями:

1. Программный комплекс архитектурно состоит из двух частей: модуль обучения классификаторов и модуль воспроизведения (применения) классификаторов.
2. Модуль воспроизведения `objed` представляет собой кроссплатформенную программную библиотеку с C++ интерфейсом, не зависящую от каких-либо «тяжеловесных» сторонних библиотек (например, Boost, Qt, OpenCV и пр.).
3. Модуль обучения представляет собой коллекцию утилит, с помощью которых решается весь комплекс подзадач в процессе обучения классификатора, включая разметку изображений, подготовку обучающих наборов, обучение классификатора и оценку качества работы классификатора.
4. За счет абстрагирования сущностей и активного применения различных паттернов программирования программный комплекс `objed` обеспечивает возможность легкого алгоритмического расширения (добавления новой функциональности).
5. Программный комплекс `objed` содержит имплементацию представленных в данной работе подходов и алгоритмов.

Программный комплекс вместе с исходным кодом размещен в репозитории GitHub и доступен для ознакомления в свободном режиме.

## Заключение

В ходе выполнения диссертационной работы были получены следующие результаты:

1. Представлено семейство признаков для алгоритма Виолы и Джонса, устойчивых к различным параметрам освещенности и учитывающих геометрические особенности объектов. Вычислительная сложность представленных признаков сопоставима со сложностью яркостных признаков Хаара. Эффективность контурных признаков продемонстрирована на примере распознавания образов колес в рамках задачи распознавания типа транспортного средства, а также на примере распознавания номера банковской карты в рамках задачи распознавания документов. В обоих случаях качество детектирования с помощью контурных признаков оказалось выше более, чем на 15% по сравнению с яркостными признаками.

2. Разработан алгоритм обучения древовидного классификатора Виолы и Джонса, который автоматически выполняет кластеризацию данных в процессе обучения, а также обеспечивает быстрое дообучение классификаторов при расширении обучающих наборов. Алгоритм был испытан для решения задачи обучения детектора образов автомобильных колес и детектора логотипа платежной системы VISA. Построенный древовидный классификатор в обоих задачах показывал лучшее качество детектирования по сравнению с классическим линейным каскадом.

3. Представлен алгоритм адаптивного выбора распознающего классификатора в задаче многоклассовой детекции объектов на видеопоследовательности, отвечающей модели поочередной демонстрации целевых объектов. В основе алгоритма лежит стратегия жадного выбора действия в задаче n-руского бандита. Эффективность описанного алгоритма была продемонстрирована на примере поиска и распознавания логотипов банковских карт в видеопотоке путем сравнения с классическими для задачи n-руского бандита стратегиями: ε-жадным методом и softmax-методом.

4. Разработан программный комплекс детекции объектов objed, содержащий программную имплементацию результатов диссертационной работы. Программный комплекс архитектурно состоит из двух частей: модуль обучения классификаторов и модуль воспроизведения (применения) классификаторов и позволяет выполнять полный цикл обучения классификаторов. За счет абстрагирования сущностей и активного применения различных паттернов программирования программный комплекс objed обеспечивает возможность легкого алгоритмического расширения (добавления новой функциональности). Программный комплекс вместе с исходным кодом размещен в репозитории GitHub и доступен для ознакомления в свободном режиме.

5. Практическая ценность полученных результатов подтверждается их использованием при решении ряда задач в индустриальных системах распознавания. Результаты диссертационной работы внедрены в систему классификации автомобилей по геометрическим характеристикам АКТС-4 (эксплуатируется на 200 полосах пунктов взимания платы автомагистралей России М1 «Беларусь», М4 «Дон» и Западного Скоростного Диаметра Санкт-Петербурга), программные продукты распознавания документов Smart IDReader и Smart PassportReader (интегрированы в информационные решения крупнейших российских банков, страховых компаний и организаций различных государственных структур), а также в интеллектуальную систему автономного вождения C-Pilot (система содействия водителю, предназначена для установки на коммерческий транспорт).

Основные результаты диссертационной работы представлены публикациях в журналах, рецензируемых ВАК, журналах, входящих в международную базу цитирования Scopus, а также сборниках трудов международных и всероссийских конференций.

## Список сокращений и условных обозначений

TP	Верные обнаружения (True Positives). Число объектов, присутствующих на изображении и обнаруженных алгоритмом.
TN	Верные пропуски (True Negative). Число верно не обнаруженных алгоритмом областей изображения.
FP	Ложные обнаружения (False Positive). Число объектов, обнаруженных алгоритмом и отсутствующих на изображении в действительности.
FN	Ложные пропуски (False Negative). Число объектов, присутствующих на изображении, но не обнаруженных алгоритмом.
TPR	Доля верных срабатываний (True Positive Rate). Отношение числа верных обнаружений к истинному числу объектов.
FPR	Доля ложных срабатываний (False Positive Rate). Отношение числа ложных срабатываний к истинному количеству ложных областей.
Recall	Полнота. То же самое, что и доля верных срабатываний.
Precision	Точность. Отношение числа верных обнаружений к сумме чисел верных и ложных обнаружений.
ROC кривая	Кривая Receiver Operator Characteristic. Кривая выражает соотношение уровней верных и ложных обнаружений.

## Список литературы

1. Rosenblatt, F. The Perceptron: A Perceiving and Recognizing Automaton / F. Rosenblatt // Cornell Aeronaut. — 1957. — Report 85-60-1.
2. Журавлев, Ю.И. Об алгебраическом подходе к решению задач распознавания и классификации / Ю.И. Журавлев // Проблемы кибернетики. — 1978. — Т. 33. С. 5–68.
3. Журавлев, Ю.И. Избранные научные труды / Ю.И. Журавлев. — М.: Магистр, 1998. — 420 с.
4. Freund, Y. Experiments with a New Boosting Algorithm / Y. Freund, R.E. Schapire // International Conference on Machine Learning. — 1996. — P. 148–156.
5. Freund, Y. A Decision-theoretic Generalization of On-line Learning and an Application to Boosting / Y. Freund, R.E. Schapire // Comput. Learn. theory. — 1997. — Vol. 55. — № 1. — P. 119–139.
6. Freund, Y. Boosting a Weak Learning Algorithm by Majority / Y. Freund // Inf. Comput. — 1995. — Vol. 121. — № 2. — P. 256–285.
7. Viola, P. Robust Real-time Object Detection / P. Viola, M. Jones // International Journal of Computer Vision. — 2001.
8. Viola, P. Rapid object detection using a boosted cascade of simple features / P. Viola, M. Jones // Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001). — 2001. — Vol. 1. — P. 511–518.
9. Rowley, H.A. Human Face Detection in Visual Scenes / H.A. Rowley, S. Baluja, T. Kanade // Adv. Neural Inf. Process. Syst. — 1995. — № 11 — P. 875–881.
10. Dalal, N. Histograms of Oriented Gradients for Human Detection / N. Dalal, W. Triggs // IEEE Computer Society Conference on Computer Vision and Pattern Recognition CVPR'05. — 2005. — Vol. 1. — № 3. — P. 886–893.
11. Chum, O. An Exemplar Model for Learning Object Classes / O. Chum, A. Zisserman // IEEE Conference on Computer Vision and Pattern Recognition. — 2007. — P. 1–8.

12. Усилин, С.А. Распознавание гильоширных элементов: определение страниц паспорта РФ / С.А. Усилин и др. // Труды Института системного анализа РАН. Обработка информационных и графических ресурсов. — 2013. — Т. 63. — № 3. — С. 106–110.
13. Усилин, С.А. Локализация, ориентация и идентификация документов с фиксированной геометрией на изображении / С.А. Усилин, Д.П. Николаев, В.В. Постников // Труды Института системного анализа РАН. Обработка информационных и графических ресурсов / под ред. В.Л. Арлазарова. — М.: КРАСАНД, 2010. — Т. 58. — С. 248–261.
14. Chen, S. Boosted road sign detection and recognition / S. Chen, J. Hsieh // 2008 Int. Conf. Mach. Learn. Cybern. — 2008. — № July. — P. 3823–3826.
15. Dollár, P. Pedestrian detection: A benchmark / P. Dollár et al. // 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009. — 2009. — P. 304–311.
16. Broggi, A. Vehicle detection for autonomous parking using a Soft-Cascade AdaBoost classifier / A. Broggi et al. // IEEE Intelligent Vehicles Symposium. — 2014. — P. 912–917.
17. Paisitkriangkrai, S. Fast pedestrian detection using a cascade of boosted covariance features / S. Paisitkriangkrai, C. Shen, J. Zhang // IEEE Trans. Circuits Syst. Video Technol. — 2008. — Vol. 18, — № 8. — P. 1140–1151.
18. Lin, C. Boosted Vehicle Detection Using Local and Global Features / C. Lin et al. // J. Signal Inf. Process. — 2013. — Vol. 4. — № 3. — P. 243–252.
19. Viola, P. Detecting pedestrians using patterns of motion and appearance / P. Viola, M. Jones, D. Snow // Int. J. Comput. Vis. 2005. — Vol. 63. — № 2. — P. 153–161.
20. Zhu, Q. Fast Human Detection Using a Cascade of Histograms of Oriented Gradients / Q. Zhu et al. // IEEE Conference on Computer Vision and Pattern Recognition. — 2006. — Vol. 2. — P. 1491–1498.
21. Jones, M. Pedestrian detection using boosted features over many frames / M. Jones, D. Snow // 2008 19th International Conference on Pattern Recognition. — 2008. — P. 1–4.

22. Moutarde, F. Real-time visual detection of vehicles and pedestrians with new efficient adaBoost features / F. Moutarde, B. Stanciulescu, A. Breheret // 2008 IEEE International Conference on Intelligent RObots Systems (IROS 2008). — 2008.
23. Lee, D.C. Boosted Classifier for Car Detection / D.C. Lee, T. Kanade. — 2007.
24. Sivaraman, S. A general active-learning framework for on-road vehicle recognition and tracking / S. Sivaraman, M.M. Trivedi // IEEE Trans. Intell. Transp. Syst. — 2010. — Vol. 11. — № 2. — P. 267–276.
25. Escalera, S. Fast greyscale road sign model matching and recognition / S. Escalera, P. Radeva // Recent Adv. Artif. Intell. Res. Dev. — 2004. — P. 69–76.
26. Lienhart, R. An extended set of Haar-like features for rapid object detection / R. Lienhart, J. Maydt // Image Processing. 2002. Proceedings. 2002 International Conference on. — 2002. — Vol. 1. — P. I-900-I-903 vol.1.
27. Messom, C. Fast and Efficient Rotated Haar-like Features using Rotated Integral Images / C. Messom, A. Barczak. — 2017.
28. Pham, M.T. Fast polygonal integration and its application in extending haar-like features to improve object detection / M.T. Pham et al. // 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. — 2010. — P. 942–949.
29. Zhang, W. Cat head detection - How to effectively exploit shape and texture features / W. Zhang, J. Sun, X. Tang // Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). — 2008. — Vol. 5305 LNCS. — № PART 4. — P. 802–816.
30. Dollár, P. Integral Channel Features / P. Dollár et al. // BMVC 2009 London Engl. — 2009. — P. 1–11.
31. Dollár, P. The Fastest Pedestrian Detector in the West / P. Dollár, S. Belongie, P. Perona // Proceedings Br. Mach. Vis. Conf. 2010. — 2010. — P. 68.1-68.11.
32. Воронцов, К.В. Машинное обучение (курс лекций) [Электронный ресурс] / К.В. Воронцов. — Режим доступа: <http://www.machinelearning.ru/> (дата обращения: 01.09.2015).

33. Мерков, А. Распознавание образов. Введение в методы статистического обучения / А. Мерков. — М. : Едиториал УРСС, 2011. — 256 с.
34. Мерков, А. Распознавание образов. Построение и обучение вероятностных моделей / А. Мерков. — М. : Ленанд, 2014. — 240 с.
35. Friedman, J. Additive logistic regression: A statistical view of boosting / J. Friedman, T. Hastie, R. Tibshirani // Annals of Statistics. — 2000. — Vol. 28. — № 2. — P. 337–407.
36. Domingo, C. MadaBoost: A Modification of AdaBoost / C. Domingo, O. Watanabe // Conference on Computational Learning Theory (COLT). — 2000. — P. 180–189.
37. Huang, C.H.C. Vector boosting for rotation invariant multi-view face detection / C.H.C. Huang et al. // Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1. — 2005. — Vol. 1. — P. 446–453.
38. Li, S.Z. FloatBoost learning and statistical face detection / S.Z. Li, Z.Q. Zhang // IEEE Trans. Pattern Anal. Mach. Intell. — 2004. — Vol. 26. — № 9. — P. 1112–1123.
39. Xiao, R. Boosting Chain Learning for Object Detection / R. Xiao, L. Zhu, H.-J. Zhang // ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision. — 2003. — P. 709.
40. Wu, Bo. Fast rotation invariant multi-view face detection based on real AdaBoost / Bo Wu et al. // In Sixth IEEE International Conference on Automatic Face and Gesture Recognition. — 2004. — P. 79–84.
41. Zhang, C. Multiple-instance pruning for learning efficient cascade detectors / C. Zhang, P. Viola // Adv. Neural Inf. Process. — 2008. — P. 1–8.
42. Lee, S. Recent Progress in Robotics: Viable Robotic Service to Human / S. Lee. — Springer Science & Business Media, 2008. — 410 p.
43. Robotics: Joint Conference on Robotics, LARS 2014, SBR 2014, Robocontrol 2014, São Carlos, Brazil, October 18-23, 2014. Revised Selected Papers (Communications in Computer and Information Science). — Springer, 2015.
44. Гонсалес, Р. Цифровая обработка изображений. Издание 3-е, исправленное и дополненное / Р. Гонсалес, Р. Вудс. — М. : Техносфера, 2012. — 1104 с.

45. Претт, У. Цифровая обработка изображений: Пер. с англ. Кн. 1 / У. Претт. — М. : Мир, 1982. — 312 с.
46. Красильников, Н.Н. Цифровая обработка 2D- и 3D-изображений: учеб. пособие / Н.Н. Красильников. — 2011. — 608 с.
47. Яне, Б. Цифровая обработка изображений / Б. Яне. — М. : Техносфера, 2007. — 584 с.
48. Гашников, М.В. Методы компьютерной обработки изображений. 2-е изд. / М.Б. Гашников и др. / под ред. Сойфера В.А. — М. : ФИЗМАТЛИТ, 2003. — 784 с.
49. Everingham, M. The pascal visual object classes (VOC) challenge / M. Everingham // Int. J. Comput. Vis. — 2010. — Vol. 88. — № 2. — P. 303–338.
50. Вапник, В.Н. Теория распознавания образов (статистические проблемы обучения) / В.Н. Вапник, А.Я. Червоненкис. — М. : Наука, 1974. — 416 с.
51. Вапник, В.Н. О методе упорядоченной минимизации риска / В.Н. Вапник, А.Я. Червоненкис // Автоматика и телемеханика. — 1994. — № 8. — С. 21–31. — № 9. — С. 29–40.
52. Вапник, В.Н. Восстановление зависимостей по эмпирическим данным / В.Н. Вапник. — М. : Наука, 1979. — 449 с.
53. What you wanted to know about AUC [Электронный ресурс]. — Режим доступа: <http://fastml.com/what-you-wanted-to-know-about-auc/> (дата обращения: 01.09.2015).
54. Davis, J. The Relationship Between Precision-Recall and ROC Curves / J. Davis, M. Goadrich // Proceedings of the 23rd International Conference on Machine learning — ICML’06. — 2006. — P. 233–240.
55. Papageorgiou, C.P. A general framework for object detection / C.P. Papageorgiou, M. Oren, T. Poggio // Sixth Int. Conf. Comput. Vis. IEEE Cat No98CH36271. — 1998. — Vol. 6. — № January. — P. 555–562.
56. Crow, F.C. Summed-area tables for texture mapping / F.C. Crow // ACM SIGGRAPH Computer Graphics. — 1984. — Vol. 18. — № 3. — P. 207–212.

57. Iba, W. Induction of One-Level Decision Trees / W. Iba, P. Langley // ML92: Proceedings of the Ninth International Conference on Machine Learning, Aberdeen, Scotland, 1–3 July 1992. — 1992. — P. 233–240.
58. Roth, D. A SNoW-based face detector / D. Roth, M. Yang, N. Ahuja // Adv. Neural Inf. Process. Syst. — 2000. — Vol. 12. — P. 855–861.
59. Schneiderman, H. A statistical method for 3D object detection applied to faces and cars / H. Schneiderman, T. Kanade // IEEE Conference on Computer Vision and Pattern Recognition. — 2000. — Vol. 1. — P. 746–751.
60. Sung, K.K. Example-based learning for view-based human face detection / K.K. Sung, T. Poggio // IEEE Trans. Pattern Anal. Mach. Intell. — 1998. — Vol. 20. — № 1. — P. 39–51.
61. Rowley, H.A. Neural network-based face detection / H.A. Rowley, S. Baluja, T. Kanade // IEEE Trans. Pattern Anal. Mach. Intell. — 1998. — Vol. 20. — № 1. — P. 23–38.
62. Huang, C.-C. An Extended Set of Haar-like Features for Bird Detection Based on AdaBoost / C.-C. Huang, C.-Y. Tsai, H.-C. Yang // Signal Processing, Image Processing and Pattern Recognition SE – 17 / ed. Kim T. et al. — Springer Berlin Heidelberg, 2011. — Vol. 260. — P. 160–169.
63. Francke, H. Real-Time Hand Gesture Detection and Recognition Using Boosted Classifiers and Active Learning / H. Francke, J. Ruiz-del-solar, R. Verschae // Lect. Notes Comput. Sci. — 2007. — Vol. 4872. — P. 533–547.
64. Ojala, T. A comparative study of texture measures with classification based on feature distributions / T. Ojala, M. Pietikäinen, D. Harwood // Pattern Recognit. — 1996. — Vol. 29. — № 1. — P. 51–59.
65. Yang, Z. Demographic Classification with Local Binary Patterns / Z. Yang, H. Ai // Proceedings of the 2007 International Conference on Advances in Biometrics. — Berlin, Heidelberg: Springer-Verlag, 2007. — P. 464–473.
66. Оператор Local Binary Patterns в задаче классификации текстур [Электронный ресурс]. Режим доступа: <http://habrahabr.ru/post/153109/> (дата обращения: 01.09.2015).

67. Ojala, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns / T. Ojala, M. Pietikäinen, T. Mäenpää // IEEE Trans. Pattern Anal. Mach. Intell. — 2002. — Vol. 24. — № 7. — P. 971–987.
68. Chang-yeon, J. Face Detection using LBP features / J. Chang-yeon // Stanford.Edu. — 2008. — P. 1–4.
69. HOG (Histogram of Oriented Gradients) with Matlab Implementation [Электронный ресурс]. Режим доступа: <http://farshbafdoustar.blogspot.com.tr/2011/09/hog-with-matlab-implementation.html> (дата обращения: 01.09.2015).
70. Schapire, R.E. Improved Boosting Algorithms Using Confidence-Based Predictions / R.E. Schapire, Y. Singer // Mach. Learn. — 1999. — Vol. 37. — № 3. — P. 297–336.
71. Freund, Y. An adaptive version of the boost by majority algorithm / Y. Freund // Mach. Learn. — 2001. — Vol. 43. — № 3. — P. 293–318.
72. Demiriz, A. Linear programming boosting via column generation / A. Demiriz, K.P. Bennett, J. Shawe-Taylor // Mach. Learn. — 2002. — Vol. 46. — № 1–3. — P. 225–254.
73. OpenCV [Электронный ресурс]. — Режим доступа: <http://opencv.org/> (дата обращения: 01.09.2015).
74. Ruan, C. Real Adaboost feature selection for Face Recognition / C. Ruan, Q. Ruan, X. Li // International Conference on Signal Processing Proceedings, ICSP. — 2010. — P. 1402–1405.
75. Duan, S. The LogitBoost Based on Joint Feature for Face Detection / S. Duan, X. Wang, W. Wan // Image and Graphics (ICIG), 2013 Seventh International Conference on. — 2013. — P. 483–488.
76. Demirkir, C. Multi pose face detection and pose estimation using Multi-class LogitBoost algorithm / C. Demirkir, B. Sankur // Signal Process. Commun. Appl. Conf. (SIU), 2010 IEEE 18th. — 2010.
77. Gualdi, G. Multi-stage sampling with boosting cascades for pedestrian detection in images and videos / G. Gualdi, A. Prati, R. Cucchiara // Lecture Notes in Computer

Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). — 2010. — Vol. 6316 LNCS. — № PART 6. — P. 196–209.

78. Bourdev, L. Robust Object Detection via Soft Cascade / L. Bourdev, J. Brandt // Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). — Washington, DC, USA: IEEE Computer Society, 2005. — Vol. 2. — P. 236–243.
79. Li, S.Z. Statistical learning of multi-view face detection / S.Z. Li et al. // Proc. European Conf. on Computer Vision. — 2002. — P. 67–81.
80. Жукова, П.Н. О профилактике правонарушений и безопасности дорожного движения / П.Н. Жукова, В.А. Насонова // Проблемы правоохранительной деятельности. — Белгород: Белгородский юридический институт Министерства внутренних дел Российской Федерации им. И.Д. Путилина, 2014. — № 1. — С. 34–36.
81. Арлазаров, В.Л. Алгоритмы распознавания и технологии ввода текстов в ЭВМ / В.Л. Арлазаров, О.А. Славин // Информационные технологии и вычислительные системы. — 1996. — Т. 1. — № 6. С. 48–54.
82. Арлазаров, В.В. Cognitive Forms система массового ввода структурированных документов / В.В. Арлазаров, В.В. Постников, Д.Л. Шоломов // Управление информационными потоками. Сборник трудов Института системного анализа РАН. — М.: Едиториал УРСС, 2002. — С. 35–46.
83. Шепелев, К.В. Применение методов распознавания образов при модернизации существующих систем видеонаблюдения в системах безопасности / К.В. Шепелев // Современный мир опыт, проблемы и перспективы развития. — Ставрополь: Центр научного знания “Логос,” 2016. — № 2. — С. 9–12.
84. Zafeiriou, S. A survey on face detection in the wild: Past, present and future / S. Zafeiriou, C. Zhang, Z. Zhang // Comput. Vis. Image Underst. — 2015. — Vol. 138. — P. 1–24.
85. Zhang, C. Boosting-Based Face Detection and Adaptation / C. Zhang, Z. Zhang. — 2010. — 140 p.

86. Comaschi, F. RASW: A run-time adaptive sliding window to improve Viola-Jones object detection / F. Comaschi et al. // 2013 7th International Conference on Distributed Smart Cameras, ICDSC 2013. — 2013.
87. Canny, J. A Computational Approach to Edge Detection / J. Canny // IEEE Trans. Pattern Anal. Mach. Intell. — 1986. — Vol. 8. — № 6. — P. 679–698.
88. Rokach, L. Data mining with decision trees: theory and applications / L. Rokach, O. Maimon // Data Mining and Knowledge Discovery. — 2007. — 242 p.
89. Норушис, А. Построение логических (древообразных) классификаторов методами нисходящего поиска (обзор) / А. Норушис // Статистические проблемы управления. — Вильнюс, 1990. — Т. 93. — С. 131–158.
90. POWERS, D.M.W. Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation / D.M.W. POWERS // J. Mach. Learn. Technol. — 2011. — Vol. 2. — № 1. — P. 37–63.
91. Thompson, W.R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples / W.R. Thompson // Biometrika. — 1933. — Vol. 25. — № 3–4. — P. 285–294.
92. Jones, M. Fast Multi-view Face Detection / M. Jones, P. Viola // Mitsubishi Electr. Res. Lab TR2000396. — 2003. — № July.
93. Lienhart, R. A detector tree of boosted classifiers for real-time object detection and tracking / R. Lienhart, L. Liang, A. Kuranov // Proceedings - IEEE International Conference on Multimedia and Expo. — 2003. — Vol. 2. — P. II277-II280.
94. Herbert, R. Some Aspects of the Sequential Design of Experiments / R. Herbert // Bull. Am. Math. Soc. — 1952. — Vol. 58. — № 5. — P. 527–535.
95. Саттон, Р.С. Обучение с подкреплением / Р.С. Саттон, Э.Г. Барто. — М.: БИНОМ. Лаборатория знаний, 2012. — 399 с.
96. Gittins, J.C. Bandit processes and dynamic allocation indices / J.C. Gittins // Statistics (Ber). — 1979. — Vol. 41. — № 2. — P. 148–177.
97. Thathachar, M.A.L. A New Approach to the Design of Reinforcement Schemes for Learning Automata / M.A.L. Thathachar, P.S. Sastry // IEEE Trans. Syst. Man Cybern. — 1985. — Vol. SMC-15. — № 1. — P. 168–175.

98. Tokic, M. Value-difference based exploration: Adaptive control between epsilon-greedy and softmax / M. Tokic, G. Palm // Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). — 2011. — Vol. 7006 LNAI. — P. 335–346.
99. Watkins, C.J.C.H. Learning from Delayed Rewards / C.J.C.H. Watkins. — Cambridge University, 1989.
100. Luce, R.D. Individual choice behavior / R.D. Luce // Individ. Choice Behav. — 1959.
101. Гришилов А.А. Математические методы построения прогнозов / А.А. Гришилов, В.А. Стакун, А.А. Стакун. — М.: Радио и связь, 1997. — 112 с.
102. Nikolaev, D.P. Diamond recognition algorithm using two-channel x-ray radiographic separator / D.P. Nikolaev et al. // Proc. SPIE 9445, Seventh Int. Conf. Mach. Vis. (ICMV 2014). — № 944507.
103. Жуковский, А.Е. Синтез обучающей выборки на основе реальных данных в задачах распознавания изображений / А.Е. Жуковский и др. // Информационные технологии и системы (ИТИС'12): сборник трудов конференции. — М., 2012. — С. 377–382.
104. Ilin, D. Creating training datasets for OCR in mobile device video stream / D. Ilin, V. Krivtsov // Proceedings 29th European Conference on Modelling and Simulation. — 2015.
105. Арлазаров, В.В. Формирование множества графических образов символов в задачах обучения классификатора символов / В.В. Арлазаров, Н. Решетняк, О.А. Славин // Труды Института системного анализа РАН. Обработка информационных и графических ресурсов. — 2014. — Т. 64. — № 4. — С. 73–79.
106. OpenCV: Cascade Classifier Training [Электронный ресурс]. — Режим доступа: [http://docs.opencv.org/2.4/doc/user\\_guide/ug\\_traincascade.html](http://docs.opencv.org/2.4/doc/user_guide/ug_traincascade.html) (дата обращения: 01.09.2017).
107. Bradski, G. Learning OpenCV 3. Computer Vision in C++ with the OpenCV Library / G. Bradski, A. Kaehler. — O'Reilly Media, 2016. — 1024 p.

108. Howse, J. OpenCV Blueprints. / J. Howse et al. — Packt Publishing, 2015. 366 p.
109. Computer Vision System Toolbox [Электронный ресурс]. — Режим доступа: <https://www.mathworks.com/help/vision/> (дата обращения: 01.09.2017).
110. Visual Studio Community [Электронный ресурс]. — Режим доступа: <https://www.visualstudio.com/ru/vs/community/> (дата обращения: 01.09.2017).
111. CMake [Электронный ресурс]. Режим доступа: <https://cmake.org/> (дата обращения: 01.09.2017).
112. Страуструп, Б. Язык программирования C++ / Б. Страуструп. — БИНОМ, 2015. — 1136 с.
113. JTC1/SC22/WG21 – The C++ Standards Committee – ISO/CPP [Электронный ресурс]. Режим доступа: <http://www.open-std.org/jtc1/sc22/wg21/> (дата обращения: 01.09.2017).
114. Шлее, М. Qt 5.3 Профессиональное программирование на C++ / М. Шлее. — СПб.: БХВ-Петербург, 2015. — 928 с.
115. Саммерфилд, М. Qt. Профессиональное программирование. Разработка кроссплатформенных приложений на C++ / М. Саммерфилд. — СПб.: Символ-Плюс, 2011. — 560 с.
116. Гербер, Р. Оптимизация ПО. Сборник рецептов / Р. Гербер и др. — СПб.: Питер, 2010. — 352 с.
117. A C++ library for interacting with JSON [Электронный ресурс]. — Режим доступа: <https://github.com/open-source-parsers/jsoncpp> (дата обращения: 01.09.2017).
118. Гамма, Э. Приемы объектно–ориентированного проектирования. Паттерны проектирования / Э. Гамма и др. — СПб.: Питер, 2007. — 366 с.
119. Khanipov, T. Vision-based industrial automatic vehicle classifier / T. Khanipov et al. // Proceedings of SPIE vol. 9445 (ICMV-2014). — 2015. — P. 9445-9445-5.
120. Коптелов, И. Макет автоматического классификатора транспортных средств / И. Коптелов и др. // Информационные технологии и системы (ИТиС'14): сборник трудов конференции. — 2014. — С. 350–256.

121. Григорьев, А.С. Архитектура системы детекции и классификации автомобилей средствами технического зрения в естественных условиях / А.С. Григорьев и др. // Сенсорные системы. — 2017. — Т. 31. — № 1. — С. 72–84.
122. Когнитив Автороботы - Cognitive Technologies - Разработка беспилотных транспортных средств [Электронный ресурс]. — Режим доступа: <http://cognitivepilot.com/> (дата обращения: 01.09.2017).
123. Прун, В.Е. Разработка мультисенсорного комплекса компьютерного (технического) зрения на базе грузового автомобиля КАМАЗ / В.Е. Прун, В.В. Постников, А.В. Панченко // БЕСПИЛОТНЫЕ ТРАНСПОРТНЫЕ СРЕДСТВА: ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ сборник материалов 94 международной научно-технической конференции Ассоциации автомобильных инженеров. — Нижний Новгород: Нижегородский государственный технический университет им. Р.Е. Алексеева, 2016. — С. 246–248.
124. Распознавание Паспорта РФ на мобильном телефоне [Электронный ресурс]. Режим доступа: <https://habrahabr.ru/company/smartengines/blog/252703/> (дата обращения: 01.09.2017).
125. Smart IDReader [Электронный ресурс]. Режим доступа: <http://smartidreader.ru/> (дата обращения: 01.09.2017).
126. RUDevices [Электронный ресурс]. Режим доступа: <http://rudevices.ru/> (дата обращения: 01.09.2017).
127. Полевой, Д.П. Задача контроля качества при создании и развитии систем оптического распознавания печатного текста / Д.П. Полевой, О.С. Самойлов // Технологии программирования и хранения данных. — 2009. — Т. 45. — С. 251–259.
128. Kim, K. Computer Vision Final Project: Viola-Jones & Morphology-based Face Detector [Электронный ресурс] / K. Kim. — 2005. — Режим доступа: [http://www.cc.gatech.edu/~kihwan23/imageCV/Final2005/FinalProject\\_KH.htm](http://www.cc.gatech.edu/~kihwan23/imageCV/Final2005/FinalProject_KH.htm) (дата обращения: 01.09.2015).

## Список публикаций автора

### Публикации в изданиях, входящих в Перечень ВАК РФ

1. Усилин, С.А. Использование метода преследования для повышения быстродействия алгоритма многоклассовой детекции объектов в видеопотоке каскадами Виолы-Джонса / С.А. Усилин // Труды Института системного анализа Российской академии наук (ИСА РАН). — М.: Поли Принт Сервис, 2017. — Т. 67. — №1. — С. 75-82.
2. Поляков, И.В. Построение оптимальных каскадов Виолы–Джонса при помощи «жадных» алгоритмов перебора управляющих параметров с промежуточным контролем по валидационной выборке / И.В. Поляков, Е.Г. Кузнецов, С.А. Усилин, Д.П. Николаев // Сенсорные системы. — М.: Наука, 2016. — Т. 30. — №3. — С. 241-248.
3. Усилин, С.А. Распознавание гильоширных элементов: определение страниц паспорта РФ / С.А. Усилин, Д.П. Николаев, Д.Л. Шоломов, В.В. Арлазаров // Труды Института системного анализа РАН. Обработка информационных и графических ресурсов. — М.: URSS, 2013. — Т. 63. — № 3. — С. 106-110.
4. Усилин, С.А. Локализация, ориентация и идентификация документов с фиксированной геометрией на изображении / С.А. Усилин, Д.П. Николаев, В.В. Постников // Труды Института системного анализа РАН. Обработка информационных и графических ресурсов. — М.: URSS, 2010. — Т. 58. — С. 248-261.
5. Гладилин, С.А. Построение устойчивых признаков детекции и классификации объектов, не обладающих характерными яркостными контрастами / С.А. Усилин, Д.П. Николаев, А.А. Котов, С.А. Гладилин // Информационные технологии и вычислительные системы. — М.: ФИЦ ИУ РАН, 2014. — Т. 1. — С. 61-72.

## **Публикации в трудах профильных конференций**

6. Fedorenko, F. Real-time object-to-features vectorisation via Siamese neural networks / F. Fedorenko, S. Usilin // Proceedings of the Ninth International Conference on Machine Vision (ICMV 2016). — SPIE, 2017. — Vol. 10341. — 103411R. (SCOPUS)
7. Minkina, A. Generalization of the Viola-Jones method as a Decision Tree of Strong Classifiers for Real-time Object Recognition in Video Stream / A. Minkina, D. Nikolaev, S. Usilin, V. Kozyrev // Proceedings of the Seventh International Conference on Machine Vision (ICMV 2014). — SPIE, 2015. — Vol. 9445. — 944517. (SCOPUS)
8. Usilin, S. Visual Appearance Based Document Image Classification / S. Usilin, D. Nikolaev, V. Postnikov, G. Schaefer // Proceedings of the 17th IEEE International Conference on Image Processing (ICIP 2010). — IEEE, 2010. — P. 2133-2136. (SCOPUS, WoS)
9. Кузнецова, Е.Г. Модификация слабых классификаторов машины Виолы-Джонса для мультиспектральных изображений / Е.Г. Кузнецова, С.А. Усилин, А. Г. Минкина, Д.П. Николаев // Информационные технологии и системы (ИТиС'15): сборник трудов конференции. — М.: ИППИ, 2015. — С. 329-337.
10. Минкина, А.Г. Обобщение метода Виолы и Джонса в виде решающего дерева сильных классификаторов для распознавания объектов в видеопотоке в режиме реального времени / А.Г. Минкина, А.С. Григорьев, С.А. Усилин, Д.В. Полевой, Д.П. Николаев // Информационные технологии и системы (ИТиС'14): сборник трудов конференции. — М.: ИППИ, 2014. — С. 158-163.
11. Григорьев, А.С. Ускорение поиска объектов в видеопотоке методом Виолы–Джонса путем адаптивного выбора распознающих каскадов / А.С. Григорьев, С.А. Усилин, Н.Д. Петрович // Труды 55-й научной конференции МФТИ. — Москва-Долгопрудный-Жуковский : МФТИ, 2012. — С. 11-12.
12. Жуковский, А.Е. Синтез обучающей выборки на основе реальных данных в задачах распознавания изображений / А.Е. Жуковский, Н.А. Тарасова, С.А. Усилин, Д.П. Николаев // Информационные технологии и системы (ИТиС'12): сборник трудов конференции. — М.: ИППИ, 2012. — С. 377-382.

13. Котов, А.А. Построение устойчивых признаков для алгоритма Виолы и Джонса в задаче классификации транспортных средств / А.А. Котов, С.А. Усилин, Д.П. Николаев // Информационные технологии и системы (ИТиС'12): сборник трудов конференции. — М.: ИППИ, 2012. — С. 383-388.

14. Usilin, S. Guilloche Elements Recognition Applied to Passport Page Processing / S. Usilin, D. Nikolaev, D. Sholomov // Proceedings of the 8th Open German-Russian Workshop «Pattern Recognition and Image Understanding» OGRW-8-2011. — Nizhny Novgorod : Lobachevsky State University of Nizhny Novgorod, 2011. — P. 303-306.

15. Жуковский, А.Е. Применение активного обучения к методу Виола-Джонса поиска объектов на изображении / А.Е. Жуковский, Д.П. Николаев, С.А. Усилин // Информационные технологии и системы (ИТиС'11): сборник трудов конференции. — М.: ИППИ, 2011. — С. 234-240.

16. Усилин, С.А. Поиск объектов в видеопотоке при известных кинематике и геометрической модели сцены / С.А. Усилин, Д.П. Николаев, В.В. Постников // Труды 53-й научной конференции МФТИ. — Москва-Долгопрудный-Жуковский : МФТИ, 2010. — С. 67–69.

17. Усилин, С.А. Идентификация изображений документов методами, предназначенными для поиска лиц / С.А. Усилин, Д.П. Николаев, В.В. Постников // Информационные технологии и системы (ИТиС'10): сборник трудов конференции. — М.: ИППИ, 2010. — С. 132–137.

### **Патенты, свидетельства о регистрации программ для ЭВМ**

18. Автономное автоматизированное рабочее место контроля паспортных документов : Патент на полезную модель № 166152 Российская Федерация / В.В. Арлазаров, А.П. Гладков, Д.П. Николаев, С.А. Усилин ; патентообладатель Общество с ограниченной ответственностью «СМАРТ ЭНДЖИНС СЕРВИС». — № 2016122432; заявл. 07.06.2016 ; опубл. 26.10.2016.

19. Автоматизированное рабочее место контроля паспортных документов : Патент на полезную модель № 166038 Российская Федерация / В.В. Арлазаров,

А.П. Гладков, Д.П. Николаев, С.А. Усилин ; патентообладатель Общество с ограниченной ответственностью «СМАРТ ЭНДЖИНС СЕРВИС». — № 2016106183; заявл. 25.02.2016 ; опубл. 24.10.2016.

20. Программа для распознавания идентификационных карт личности «Smart IDReader» : свидетельство о государственной регистрации программы для ЭВМ № 2016616961 / В.В. Арлазаров, Д.П. Николаев, С.А. Усилин, К.Б. Булатов, Т.С. Чернов, Д.Г. Слугин, Д.А. Ильин, П.В. Безматерных, А.А. Муковозов, Е.Е. Лимонова. — № 2016612014 ; заявл. 10.03.2016 ; опубл. 22.06.2016.

21. Библиотека для распознавания в видеопотоке паспорта гражданина Российской Федерации «Smart PassportReader» : свидетельство о государственной регистрации программы для ЭВМ № 2015616071 / В.В. Арлазаров, К.Б. Булатов, Д.А. Ильин, А.В. Куроптев, Д.П. Николаев, Д.В. Полевой, С.А. Усилин, И.А. Фараджев, Т.С. Чернов. — № 2015612880 ; заявл. 10.04.2015 ; опубл. 29.05.2015.

22. Программа для распознавания и детектирования автомобилей, с учетом их ракурса и направления движения : свидетельство о государственной регистрации программы для ЭВМ № 2015610891 / Д.В. Полевой, С.А. Усилин, П.В. Безматерных. — № 2014662041 ; заявл. 26.11.2014 ; опубл. 20.01.2015.

23. Способ автоматической классификации транспортных средств : Пат. 2486597 С1 Российская Федерация / Д.П. Николаев, В.В. Постников, Т.М. Ханипов, С.А. Усилин, А.С. Григорьев ; патентообладатель Общество с ограниченной ответственностью «Техно-траффик». — № 2012105146/11 ; заявл. 09.02.2012 ; опубл. 27.06.2013, Бюл. № 18. — 15 с.

## Приложение А

### Модуль обучения программного комплекса objed

Модуль обучения программного комплекса objed представляет собой набор программ, предназначенных для осуществления полного цикла построения классификатора методом Виолы и Джонса: подготовка обучающего набора изображений, обучение классификатора, тестирование и оценка качества работы построенного классификатора.

Инструментарий objed является открытым программным обеспечением, исходный код objed доступен по адресу: <https://github.com/usilinsergey/objed>.

В данном приложении представлено краткое описание основных составляющих инструментарий objed.

### Содержание

A.1. Инструмент разметки набора изображений и анализа результатов детектирования objedmarker.....	128
A.2. Инструмент подготовки обучающего набора изображений objedcutcli .....	130
A.3. Инструмент обучения классификатора objetraincli .....	131
A.4. Инструмент визуального контроля работы классификатора objedcheck .....	135
A.5. Инструмент анализа качества работы классификатора objedruncli .....	136

## А.1. Инструмент разметки набора изображений и анализа результатов детектирования objedmarker

Инструмент разметки набора изображений и анализа результатов детектирования objedmarker представляет собой программу с графическим пользовательским интерфейсом, выполняющую две основные задачи:

- разметка набора изображений на предмет наличия объекта (в русскоязычной литературе данная процедура часто упоминается как «изготовление идеала» [127]);
- сравнение результатов детектирования с «идеальными» (то есть, созданными вручную оператором).

Пользовательский интерфейс программы objedmarker представлен на рисунке А.1. Главное окно разделено на две основные области: слева находится область панелей инструментов, справа – область просмотра изображения и разметки.

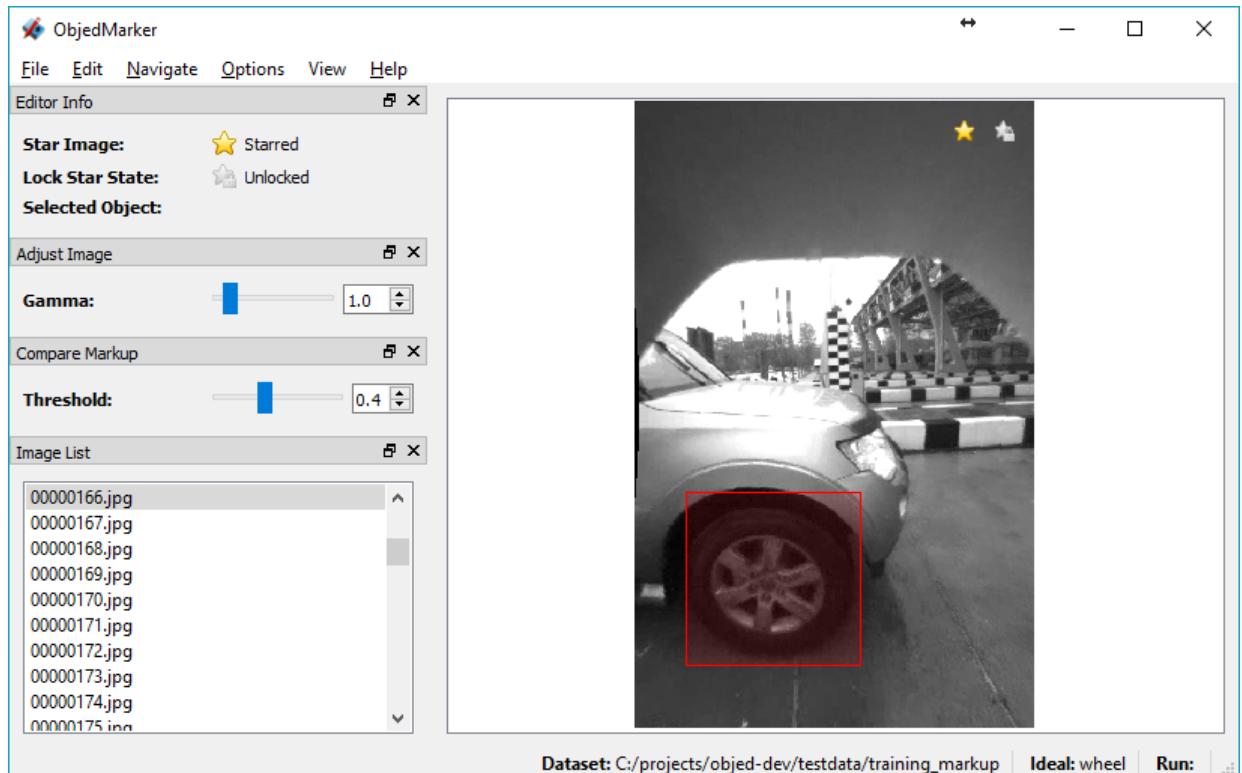


Рисунок А.1 — Внешний вид программы objedmarker

Открытие набора изображений выполняется с помощью команды меню File→Open Dataset... (Ctrl+O). При открытии указывается директория изображений (Directory), а также имя размечаемых данных (Ideal Markup). Опционально можно указать имя результата детектирования (Run Markup) для сравнения результата детектирования с «идеальными» значениями.

С помощью диалогового окна Option Size Properties следует указать характерные размеры размечаемых объектов (базовый размер задает соотношение сторон размечаемого объекта, минимальный и максимальный размеры – границы размечаемых объектов). Программа objedmarker при снятии флажка Base Object Size возможна разметка объектов с произвольным соотношением сторон. Цвет размечаемых объектов указывается с помощью диалогового окна Editor Properties.

Процесс разметки каждого изображения состоит из следующих двух шагов:

Шаг 1. С помощью левой кнопки мыши (устанавливает границы нового объекта) и клавиш W и S (используются при нажатой кнопке мыши и позволяют увеличивать или уменьшать размер нового объекта) выполняется разметка всех видимых объектов. Удаление случайного объекта выполняется с помощью правой кнопки мыши. Перемещение и изменение размера выполняется также с помощью кнопки мыши).

Шаг 2. Если на изображении все присутствующие объекты размечены и нет никаких сомнительных элементов (например, присутствует объект наполовину видимый, слишком маленький, размытый и т.п.), то такое изображение считается размеченным полностью и с помощью клавиши Space помечается «звездочкой». В противном случае изображение считается размеченным частично и не помечается «звездочкой».

Навигация между изображениями выполняется с помощью клавиш A и D. Сохранение результатов разметки выполняется автоматически в директорию <директория изображений>/<имя размечаемых данных>. Для повышения визуального качества изображения присутствует возможность контрастирования с помощью панели Adjust Image.

При сравнении результата детектирования с «идеальной разметкой» необходимо с помощью панели Compare Markup указать порог взаимного перекрытия объектов (от 0.0 до 1.0). Далее, с помощью комбинаций клавиш Shift+D и Shift+A можно перемещаться к следующему и предыдущем отлинию соответственно. С помощью левой кнопки мыши можно добавить результат детектирования в «идеал» (полезно при условии непреднамеренного пропуска объекта при создании «идеальной» разметки).

## А.2. Инструмент подготовки обучающего набора изображений objedcutcli

Инструмент подготовки обучающего набора изображений objedcutcli представляет собой программу с интерфейсом командной строки и предназначен для подготовки обучающих наборов изображений на основе размеченного набора изображений.

При запуске программы objedcutcli необходимо передать в качестве аргумента путь к конфигурационному файлу в формате INI. Чтобы сгенерировать шаблон конфигурационного файла запустите программу следующим образом:

```
objedcutcli -g [config_path]
```

Ниже представлено краткое описание основных элементов конфигурационного файла.

**Mode.** Задает режим функционирования программы objedcutcli. В режиме *cutting* происходит подготовка обучающего набора данных. В режиме *statistics* выполняется подсчет статистической информации по размеченному изображению:

- общее количество изображений в размеченном наборе;
- количество изображений, размеченных полностью;
- количество размеченных объектов;
- характерные размеры размеченных объектов;

**MarkupName.** Задает имя размечаемых данных (должно совпадать со значением Ideal Markup, указанным при разметке).

**LeftMargin, RightMargin, TopMargin, BottomMargin.** Указывают насколько необходимо расширить изображение размеченного объекта при вырезании слева, справа, сверху, снизу соответственно. Данные величины задаются в долях от ширины (высоты).

**PositiveDataset, NegativeDataset.** Указывают директории для сохранения положительной и отрицательной обучающих выборок. Если данные параметры не указаны, то соответствующая обучающая выборка не будет сгенерирована.

**OutputFormat.** Указывают формат сохранения обучающей выборки. Поддерживается TIFF и JPEG.

**MinObjectWidth, MaxObjectWidth, MinObjectHeight, MaxObjectHeight.** Указывают минимальные и максимальные размеры вырезываемых объектов.

**DatasetList.** Содержит список размеченных наборов данных. Указывается путь к директориям с изображениями, внутри которых должны быть соответствующие директории с разметкой.

### A.3. Инструмент обучения классификатора objedtraincli

Инструмент обучения классификатора objedtraincli представляет собой программу с интерфейсом командной строки и предназначен для обучения бинарных классификаторов типа Виолы и Джонса, описанных в данной работе.

При запуске программы objedtraincli необходимо передать в качестве аргумента путь к конфигурационному файлу в формате INI. Чтобы сгенерировать шаблон конфигурационного файла запустите программу следующим образом:

```
objedtraincli -g [config_path]
```

Ниже представлено краткое описание основных элементов конфигурационного файла.

**ClassifierType.** Задает высокоуровневую структуру классификатора. Возможно обучение каскадного классификатора (cascadeClassifier) и древовидного классификатора (treeClassifier) в соответствии с методом, описанным в разделе 3.

**ClassifierPath.** Указывает путь и имя файла, в который будет сохранен классификатор. Если по указанному пути уже существует классификатор, то он будет «дообучен».

**ClassifierWidth, ClassifierHeight.** Задает физические размеры в пикселях обучаемого классификатора. Размеры должны быть нечетные.

**MinimumScale, MaximumScale, StepScale.** Задает параметры масштабирования, при которых будет обучаться классификатор, то есть при обучении будут использоваться все размеры начиная с  $ClassifierSize \cdot MinimumScale$  до  $ClassifierSize \cdot MaximumScale$  с множителем  $StepScale$ .

**NegativeCount.** Задает количество отрицательных примеров, которые должны быть собраны для обучения очередного сильного классификатора. В соответствии со спецификой программы objed реальное количество собранных отрицательных примеров будет не меньше, чем  $NegativeCount$ .

**PositiveCountThreshold.** Задает порог обучения очередного сильного классификатора по количеству положительных примеров. Другими словами, если для обучения очередного сильного классификатора удалось собрать положительных примеров  $PositiveCount < PositiveCountThreshold$ , то обучение будет остановлено. В основном, применяется при обучении древовидного классификатора.

**NegativeCountThreshold.** Задает порог обучения очередного сильного классификатора по количеству отрицательных примеров. Другими словами, если для обучения очередного сильного классификатора удалось собрать отрицательных примеров  $NegativeCount < NegativeCountThreshold$ , то обучение будет остановлено.

**WeightShift.** Задает первоначальный сдвиг ценности обучающей выборки. Если  $WeightShift$  ближе к 1.0, то положительные примеры считаются более ценными, если  $WeightShift$  ближе к 0.0, то отрицательные примеры считаются более ценными.

**LevelCount.** Задает количество сильных классификаторов, которые следует обучить в текущем процессе (запуске). Если *LevelCount* = 0, то ограничений по количеству обучаемых сильных классификаторов отсутствует.

**WcCount.** Задает количество слабых классификаторов, которые будут добавлены в сильный классификатор на очередной итерации обучения при условии установки флага *StoppingCriterion* = *Count* или *StoppingCriterion* = *Any*. Если *WcCount* = 0, то ограничений по количеству добавляемых слабых классификаторов отсутствует.

**FalseNegativeRate, FalsePositiveRate.** Задают критерии обучения очередного сильного классификатора в качественных характеристиках. Оценка качественных характеристик вычисляется на основе актуальной обучающей выборки. Данные критерии действуют при условии установки флага *StoppingCriterion* = *Rate* или *StoppingCriterion* = *Any*.

**PositiveDatasetList, NegativeDatasetList.** Задают списки положительных и отрицательных наборов изображений для обучения.

**WcLineList.** Задает семейства слабых классификаторов, которые могут быть использованы при обучении. Указание соответствующего семейства выполняется с помощью текстовой строки следующего вида:

<ТипКлассификатора> -preproc=<ВидПрепроцессинга> <ДопПараметры>

В настоящий момент поддерживаются следующие типы классификаторов:

- Haar1StumpWc – ступенчатый классификатор поверх 1-прямоугольных признаков Хаара, вычисляемых поверх одноканального изображения;
- Haar1PwWc – кусочно-постоянный классификатор поверх 1-прямоугольных признаков Хаара, вычисляемых поверх одноканального изображения;
- Haar2StumpWc – ступенчатый классификатор поверх 2-прямоугольных признаков Хаара, вычисляемых поверх одноканального изображения;
- Haar2PwWc – кусочно-постоянный классификатор поверх 2-прямоугольных признаков Хаара, вычисляемых поверх одноканального изображения;

– Haar3StumpWc – ступенчатый классификатор поверх 3-прямоугольных признаков Хаара, вычисляемых поверх одноканального изображения;

– Haar3PwWc – кусочно-постоянный классификатор поверх 3-прямоугольных признаков Хаара, вычисляемых поверх одноканального изображения;

Вид препроцессинга представляет собой набор последовательных инструкций приведения изображения до поддерживаемого классификатором формата. В настоящий момент поддерживаются следующие виды препроцессинга:

– gray – «осерение» изображения путем усреднения каналов, на входе – трехканальное изображение, на выходе – одноканальное изображение;

– saturation – «осерение» путем вычисления насыщенности, на входе – трехканальное изображение, на выходе – одноканальное изображение;

– channel<index> – «осерение» путем взятия канала <index>, на входе – трехканальное изображение, на выходе – одноканальное изображение;

– gradient<direction> – вычисление модуля градиента в трех режимах: при  $\langle\text{direction}\rangle=0$  вычисляется классическое значение модуля градиента, при  $\langle\text{direction}\rangle=1$  вычисляется градиент прямых границ, при  $\langle\text{direction}\rangle=2$  вычисляется градиент диагональных границ;

– canny<direction> – вычисление карты границ в трех режимах: при  $\langle\text{direction}\rangle=0$  вычисляется классическая карта границ, при  $\langle\text{direction}\rangle=1$  вычисляется карта прямых границ, при  $\langle\text{direction}\rangle=2$  вычисляется карта направленных границ;

Препроцессинг может состоять из нескольких элементов, при этом элементы должны быть разделены вертикальной чертой, например: gray|gradient2.

Для кусочно-постоянных классификаторов необходимо указать кол-во интервалов с помощью параметра bin, например –bin=16.

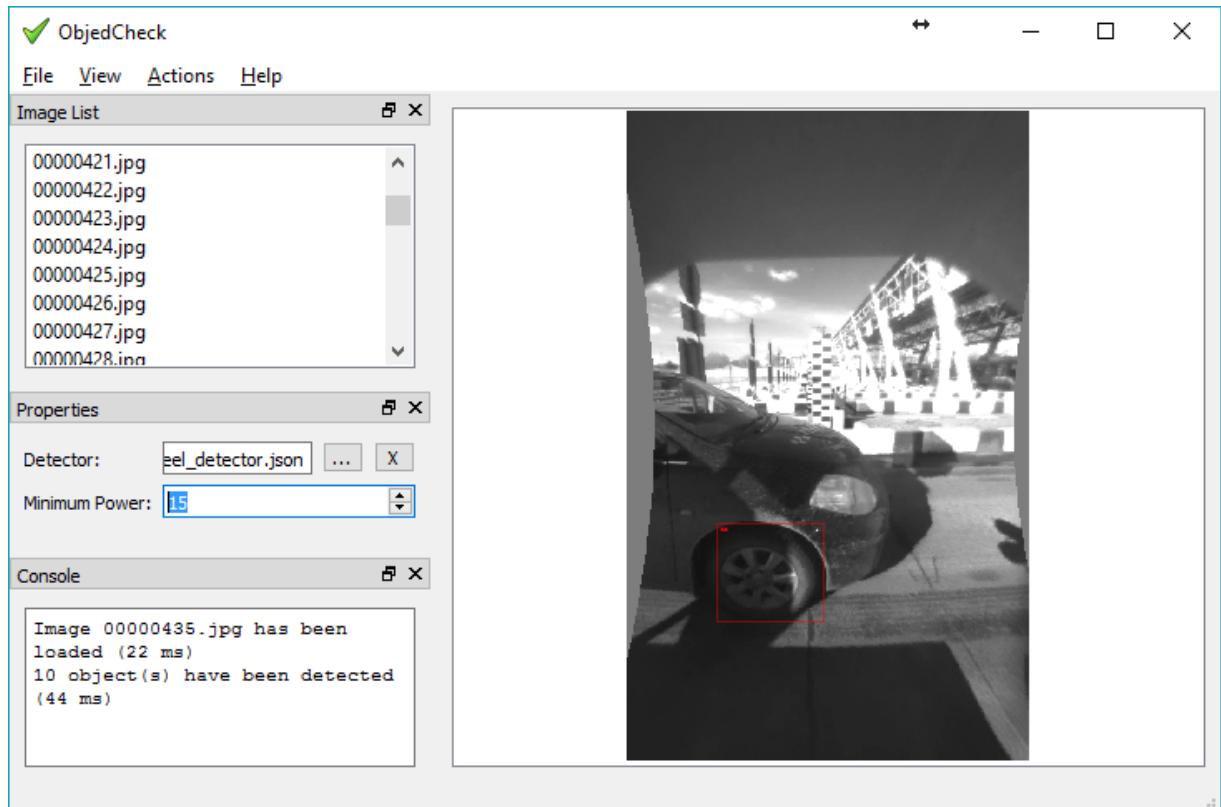
Ширины и высоты образующих прямоугольных областей, а также допустимые зазоры для признаков Хаара задаются с помощью диапазонов следующим образом: <мин\_знач>:<макс\_знач>/<шаг>. Например, запись – wd=4:16/4 задает диапазон ширин, состоящий из 4, 8, 12 и 16.

С помощью дополнительного параметра normalize, принимающего булевое значение, можно управлять нормализацией значения признака. Если –normalize=true, то значение признака нормализуется на сумму яркостей внутри образующих прямоугольников. Если–normalize=false, то значение признака нормализуется исходя их максимальных допустимых значений.

#### **A.4. Инструмент визуального контроля работы классификатора objedcheck**

Инструмент визуального контроля работы классификатора objedcheck представляет собой программу с графическим пользовательским интерфейсом, предназначенную для органолептического контроля работоспособности обученных классификаторов и построенных на их базе детекторов.

Программа позволяет указать директорию с тестовыми изображениями, а также детектор (сущность инструментария objed, которая дополнительно к



**Рисунок А.2 — Внешний вид программы objedcheck**

обученному с помощью `objedtraincli` классификатора содержит информацию о допустимых масштабах, шаге применения и способе объединения множественных «срабатываний» вокруг одного и того же объекта).

Под детекцией будем понимать обрамляющий прямоугольник и количество «срабатываний», которые были объединены в данную детекцию (мощность детекции). Программа `objedcheck` позволяет фильтровать детекции с малым значением мощности с помощью панели `Properties`.

Дополнительно к визуализации результатов детектирования, `objedcheck` позволяет визуализировать различные варианты препроцессинга, описанного в разделе А.3. Для этого необходимо открыть диалоговое окно `Image View` с помощью пункта меню `View→Open Image View...` и в появившемся диалоговом окне указать необходимую строку препроцессинга.

## A.5. Инструмент анализа качества работы классификатора `objedruncli`

Инструмент анализа качества работы классификатора `objedruncli` представляет собой программу с интерфейсом командной строки и предназначен для тестирования обученных классификаторов и построенных на их базе детекторов на размеченном тестовом наборе изображений.

При запуске программы `objedruncli` необходимо передать в качестве аргумента путь к конфигурационному файлу в формате INI. Чтобы сгенерировать шаблон конфигурационного файла запустите программу следующим образом:

```
objedruncli -g [config_path]
```

Ниже представлено краткое описание основных элементов конфигурационного файла.

**DetectorPath.** Путь к детектору (аналогично программе `objedcheck`).

**MinimumPower.** Минимальное значение мощности детекций, которые принимаются в расчет.

**IdealMarkupName.** Задает имя размечаемых данных (должно совпадать со значением Ideal Markup, указанным при разметке данного тестового набора).

**SaveRealMarkup.** Указывает следует ли сохранять результат детектирования для последующего анализа с помощью objedmarker. Если true, то результат детектирования будет сохранен в директорию <IdealMarkupName>-ууууммдд-ННММсс, где ууууммдд-ННММсс задает дату и время начала процедуры детектирования.

**Threshold.** Указывает минимальный порог перекрытия найденного объекта и «идеального» для принятия решения о детектировании. Подробнее про оценку качества детекции изложено в разделе 1.2.3.

**DatasetList.** Задает список тестовых наборов изображений

В качестве результата работы программы objedruncli для каждого тестового набора изображений и для всех наборов изображений в целом выведет следующую посчитанную информацию:

- TP – true positive, количество верно найденных объектов;
- FP – false positive, количество ошибочно найденных объектов;
- FN – false negative, количество ненайденных объектов;
- IC – image count, количество обработанных изображений.

## Приложение Б

### Алгоритм AdaBoost

Алгоритм AdaBoost (сокр. от adaptive boosting) – алгоритм машинного обучения, предложенный Йоавом Фройндом (Yoav Freund) и Робертом Шапиром (Robert Schapire). Является мета-алгоритмом, в процессе обучения строит композицию из базовых алгоритмов обучения для улучшения их эффективности. AdaBoost является алгоритмом адаптивного бустинга в том смысле, что каждый следующий классификатор строится по объектам, которые плохо классифицируются предыдущими классификаторами.

Изучим принцип работы алгоритма AdaBoost более подробно на примере задачи классификации на два класса, то есть  $\mathbb{Y} = \{-1, +1\}$ . Пусть есть множество слабых классификаторов  $b_t: \mathbb{X} \rightarrow \mathbb{Y}$ . Будем строить сильный классификатор с помощью алгоритма AdaBoost следующего вида:

$$a(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t b_t(x)\right), \quad x \in \mathbb{X}. \quad (\text{Б.1})$$

Определим функционал качества композиции как число ошибок, допускаемых ею на обучающей выборке  $\mathbb{X}^l$ :

$$Q_T = \sum_{i=1}^l [y_i a(x_i) < 0] = \sum_{i=1}^l \left[ y_i \sum_{t=1}^T (\alpha_t b_t(x_i)) < 0 \right], \quad x_i \in \mathbb{X}^l. \quad (\text{Б.2})$$

В итеративном алгоритме AdaBoost используются два основополагающих приема:

1. При добавлении в композицию очередного слагаемого  $\alpha_t b_t(x)$  оптимизируется только базовый алгоритм  $b_t(x)$  и коэффициент при нём  $\alpha_t$ , а все предыдущие слагаемые  $\alpha_1 b_1(x), \dots, \alpha_{t-1} b_{t-1}(x)$  остаются нетронутыми.

2. Пороговая функция потерь в функционале качества заменяется непрерывно дифференцируемой оценкой сверху:

$$[y_i a(x_i) < 0] \leq e^{-y_i a(x_i)} \quad x_i \in \mathbb{X}^l. \quad (\text{Б.3})$$

Оценим функционал  $Q_T$  сверху:

$$\begin{aligned} Q_T \leq \tilde{Q}_T = \sum_{i=1}^l e^{-y_i \sum_{t=1}^T (\alpha_t b_t(x_i))} &= \sum_{i=1}^l e^{-y_i \sum_{t=1}^{T-1} (\alpha_t b_t(x_i))} e^{-y_i \alpha_T b_T(x_i)} \quad x_i \\ &\in \mathbb{X}^l. \end{aligned} \quad (\text{Б.4})$$

Обозначим  $w_i = e^{-y_i \sum_{t=1}^{T-1} \alpha_t b_t(x_i)}$ . Заметим, что величина  $w_i$  (назовем ее «весом» обучающего примера  $x_i$ ) не зависит от  $\alpha_T b_T$  и может быть вычислена перед построением очередного слабого классификатора  $b_T$ .

Введем понятие нормированного веса  $\tilde{w}_i = \frac{w_i}{\sum_{j=1}^l w_j}$ , а также определим два функционала качества слабого классификатора  $b(x)$ : суммарный вес ошибочных (negative) классификаций  $N(b)$  и суммарный вес правильных (positive) классификаций  $P(b)$ :

$$N(b) = \sum_{i=1}^l \tilde{w}_i [b(x_i) = -y_i]; \quad (\text{Б.5})$$

$$P(b) = \sum_{i=1}^l \tilde{w}_i [b(x_i) = y_i]. \quad (\text{Б.6})$$

Заметим, что в силу нормировки веса справедливо равенство  $N + P = 1$ .

**Теорема Б.1.** Пусть для нормированных весов  $\tilde{w}_1, \dots, \tilde{w}_l$  существует базовый классификатор  $b$ , для которого справедливо  $N(b) < \frac{1}{2}$ . Тогда минимум функционала  $\tilde{Q}_T$  достигается при  $b_T = \operatorname{argmin}_b(N(b))$  и  $\alpha_T = \frac{1}{2} \ln \frac{1-N(b)}{N(b)}$ .

**Доказательство.** Заметим, что для любых  $a \in \mathbb{R}$  и  $b \in \{-1, +1\}$  справедливо следующее тождество:  $e^{-ab} = e^{-a}[b = 1] + e^a[b = -1]$ . Тогда, так как  $y_i \in \mathbb{Y} = \{-1, +1\}$  преобразуем функционал  $\tilde{Q}_T$  следующим образом:

$$\tilde{Q}_T = \left( e^{-\alpha} \sum_{i=1}^l \tilde{w}_i [b(x_i) = y_i] + e^{\alpha} \sum_{i=1}^l \tilde{w}_i [b(x_i) = -y_i] \right) \sum_{j=1}^l w_j. \quad (\text{Б.7})$$

Из формулы (Б.7) следует:

$$\tilde{Q}_T = (e^{-\alpha} P + e^{\alpha} N) \cdot \tilde{Q}_{T-1} = (e^{-\alpha}(1 - N) + e^{\alpha} N) \cdot \tilde{Q}_{T-1}. \quad (\text{Б.8})$$

Дифференцируя  $\tilde{Q}_T$  по параметру  $\alpha$  и приравнивая к нулю производную получаем оптимальное значение  $\alpha_T = \frac{1}{2} \ln \frac{1-N(b_T)}{N(b_T)}$ . Подставим найденное выражение  $\alpha_T$  обратно в формулу, получим:

$$\tilde{Q}_T = \tilde{Q}_{T-1} \cdot \left( \sqrt{N(1 - N)} + \sqrt{N(1 - N)} \right) = 2 \cdot \tilde{Q}_{T-1} \cdot \sqrt{N - N^2}. \quad (\text{Б.9})$$

Поскольку  $\tilde{Q}_{T-1}$  не зависит от  $\alpha_T$  и  $b_T$ , минимизация  $\tilde{Q}_T$  эквивалентна максимизации  $N - N^2$  при условии  $0 \leq N < \frac{1}{2}$ . То есть  $b_T = \operatorname{argmin}_b(N(b))$ .

Теорема доказана.

Требование, чтобы каждый слабый классификатор распознавал объекты хотя бы немного лучше, чем наугад, является достаточно слабым, и на практике, как правило, выполняется. Более того, этого оказывается достаточно, чтобы гарантировать сходимость алгоритма AdaBoost за конечное число шагов.

**Теорема Б.2.** Если на каждом шаге метод обучения обеспечивает построение слабого классификатора  $b_t$  такого, что  $N(b_t) = \epsilon_t$ , где  $0 < \epsilon_t < \epsilon$  для некоторого  $0 < \epsilon < \frac{1}{2}$ , то за конечное число шагов будет построен корректный сильный классификатор  $a(x)$ .

Доказательство. Из формулы (Б.9) следует:

$$Q_T \leq \tilde{Q}_T \leq 2 \cdot \tilde{Q}_{T-1} \sqrt{\epsilon - \epsilon^2} \leq 2 \cdot \tilde{Q}_1 \left( \sqrt{\epsilon - \epsilon^2} \right)^T \quad (\text{Б.10})$$

Так как  $0 < \epsilon < \frac{1}{2}$ , то  $\sqrt{\epsilon - \epsilon^2} < 1$ . Следовательно, наступит момент, когда значение  $\tilde{Q}_T$  окажется меньше 1. Но тогда функционал  $Q_T$  обратится в нуль, поскольку он может принимать только целые неотрицательные значения. Теорема доказана.

Реализация алгоритма в виде блок-схемы показана на рисунке Б.2.

Остановимся отдельно на идее назначении весов обучающим примерам. На каждом шаге итерации вес объекта увеличивается, если предыдущий слабый классификатор допустил на нем ошибку и уменьшается в противном случае. Таким образом, непосредственно перед настройкой очередного слабого классификатора наибольший вес накапливается у тех объектов, которые чаще оказывались трудными для предыдущих слабых классификаторов. Эта идея отражена на рисунке Б.1.

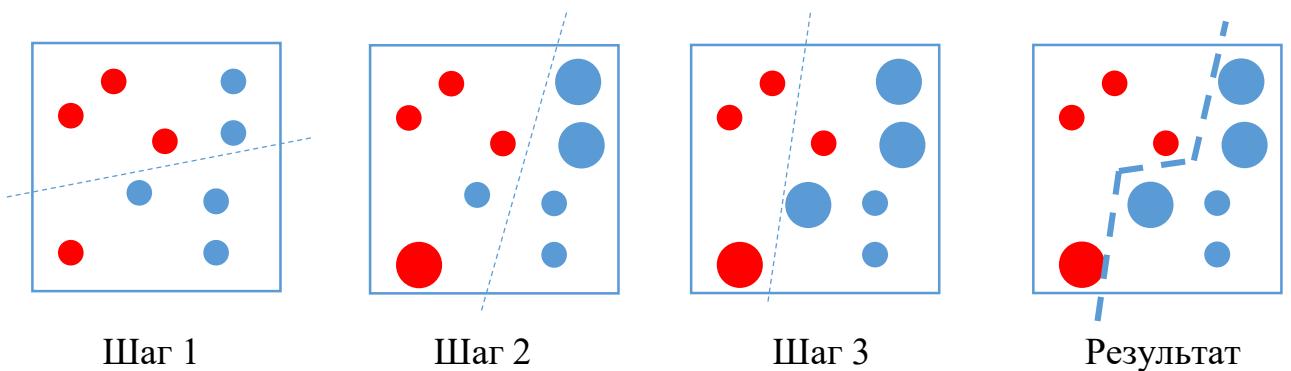


Рисунок Б.1 — Наглядная демонстрация принципа алгоритма AdaBoost: поэтапное построение сильного классификатора из трех слабых (источник [128])

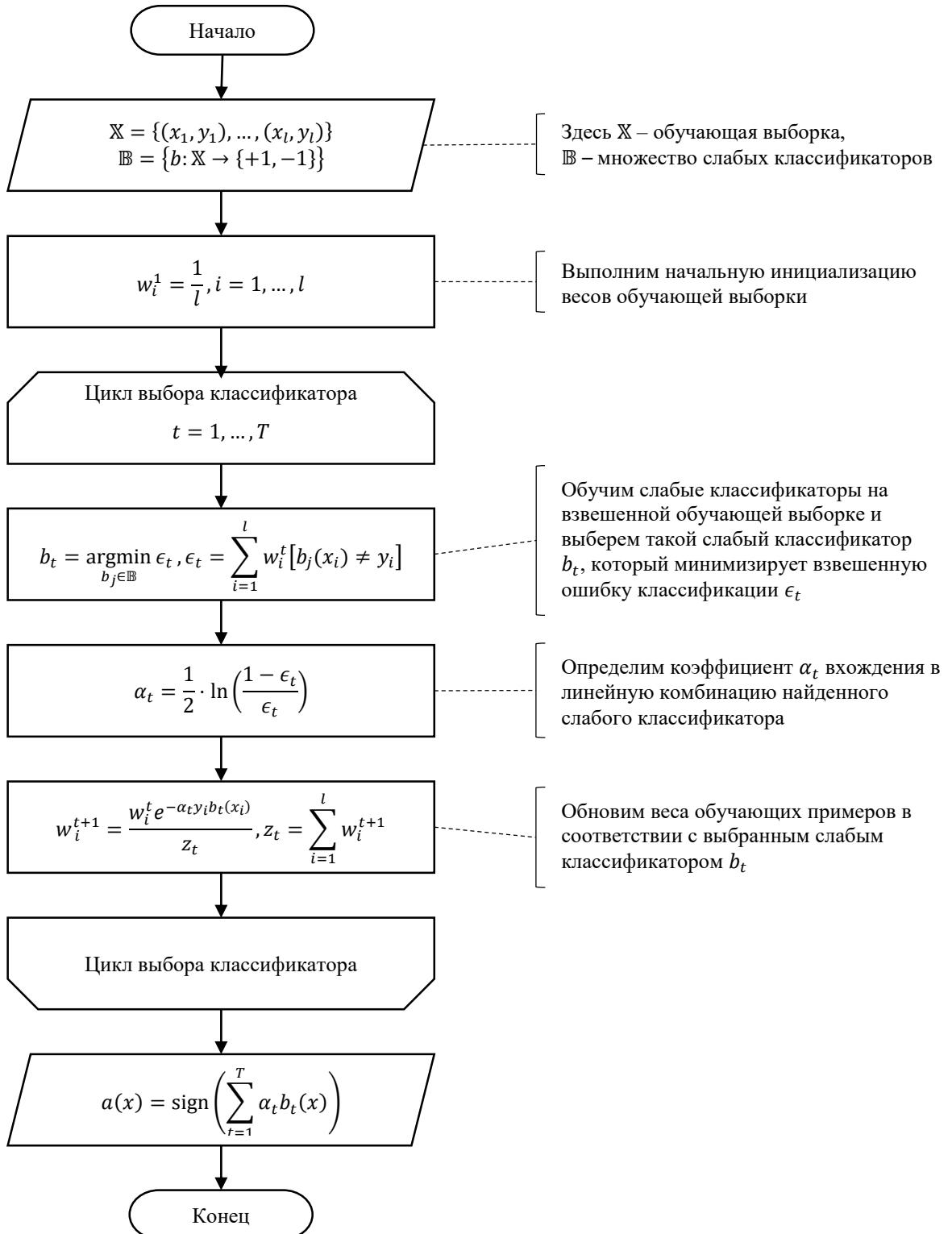


Рисунок Б.2 — Блок-схема алгоритма AdaBoost

Бустинг над решающими деревьями считается одним из наиболее эффективных методов с точки зрения качества классификации. При этом на практике наблюдается уменьшение частоты ошибок на независимой тестовой выборке по мере наращивания композиции. Более того, качество на тестовой выборке часто продолжало улучшаться даже после достижения безошибочного распознавания всей обучающей выборки [4], что перевернуло существовавшие долгое время представления о том, что для повышения обобщающей способности необходимо ограничивать сложность алгоритмов. На примере бустинга стало понятно, что хорошим качеством могут обладать сколь угодно сложные композиции, если их правильно строить.

## Приложение В

### Охранные документы на результаты интеллектуальной деятельности

РОССИЙСКАЯ ФЕДЕРАЦИЯ

(19) RU (11) 2 486 597<sup>(13)</sup> C1(51) МПК  
G08G 1/015 (2006.01)  
G08G 1/04 (2006.01)ФЕДЕРАЛЬНАЯ СЛУЖБА  
ПО ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ

(21)(22) Заявка: 2012105146/11, 09.02.2012

(24) Дата начала отсчета срока действия патента:  
09.02.2012

Приоритет(ы):

(22) Дата подачи заявки: 09.02.2012

(45) Опубликовано: 27.06.2013 Бюл. № 18

(56) Список документов, цитированных в отчете о  
поиске: WO 03096300 A1, 20.11.2003. US 5446291 A,  
29.08.1995. JP 2001283375 A, 12.10.2001. US  
5750069 A, 12.05.1998. SU 557405 A, 05.05.1977.

Адрес для переписки:

119119, Москва, Ленинский пр-т, 42, корп.3,  
оф.32-31, ООО "ПАТЕНТУС", пат.лов. А.В.  
Михайлову, рег.№ 1339 (для ООО "Техно-  
Траффик")

(72) Автор(ы):

Николаев Дмитрий Петрович (RU),  
Постников Василий Валерьевич (RU),  
Ханипов Тимур Маратович (RU),  
Усилин Сергей Александрович (RU),  
Григорьев Антон Сергеевич (RU)

(73) Патентообладатель(и):

Общество с ограниченной  
ответственностью "Техно-траффик" (RU)

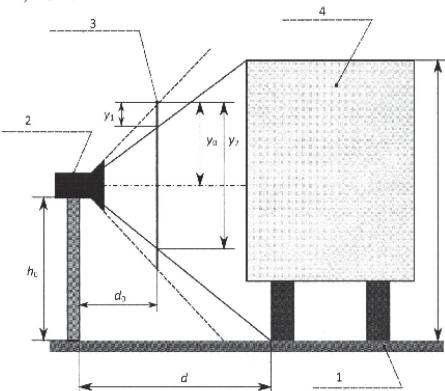
R U 2 4 8 6 5 9 7 C 1

(54) СПОСОБ АВТОМАТИЧЕСКОЙ КЛАССИФИКАЦИИ ТРАНСПОРТНЫХ СРЕДСТВ

(57) Реферат:

Изобретение относится к области контроля движения дорожного транспорта. Предложенный способ предусматривает семь основных этапов: (1) исправление радиальной дисторсии; (2) детекцию въездов и/или выездов; (3) определение высоты транспортного средства; (4) детекцию осей транспортного средства; (5) предварительный подсчет количества осей транспортного средства; (6) фильтрацию результатов предварительного подсчета (статистическую обработку) и (7) окончательный подсчет количества осей транспортного средства. Технический результат - высокоточная, полностью автономная, многокритериальная классификация транспортных средств. 8 з.п. ф-

лы, 16 ил.



Фиг. 3

R U 2 4 8 6 5 9 7 C 1

## РОССИЙСКАЯ ФЕДЕРАЦИЯ



## ПАТЕНТ

НА ПОЛЕЗНУЮ МОДЕЛЬ

№ 166038

АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО КОНТРОЛЯ  
ПАСПОРТНЫХ ДОКУМЕНТОВ

Патентообладатель(ли): *Общество с ограниченной  
ответственностью "СМАРТ ЭНДЖИНС СЕРВИС" (RU)*

Автор(ы): *Арлазаров Владимир Викторович (RU), Гладков  
Андрей Павлович (RU), Николаев Дмитрий Петрович (RU),  
Усилин Сергей Александрович (RU)*

Заявка № 2016106183

Приоритет полезной модели 25 февраля 2016 г.

Зарегистрировано в Государственном реестре полезных  
моделей Российской Федерации 24 октября 2016 г.

Срок действия патента истекает 25 февраля 2026 г.

Руководитель Федеральной службы  
по интеллектуальной собственности

Г.П. Ильин



## РОССИЙСКАЯ ФЕДЕРАЦИЯ



## ПАТЕНТ

НА ПОЛЕЗНУЮ МОДЕЛЬ

№ 166152

АВТОНОМНОЕ АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ  
МЕСТО КОНТРОЛЯ ПАСПОРТНЫХ ДОКУМЕНТОВ

Патентообладатель(ли): *Общество с ограниченной  
ответственностью "СМАРТ ЭНДЖИНС СЕРВИС" (RU)*

Автор(ы): *Арлазаров Владимир Викторович (RU), Гладков  
Андрей Павлович (RU), Николаев Дмитрий Петрович (RU),  
Усилин Сергей Александрович (RU)*

Заявка № 2016122432

Приоритет полезной модели **07 июня 2016 г.**Зарегистрировано в Государственном реестре полезных  
моделей Российской Федерации **26 октября 2016 г.**Срок действия патента истекает **07 июня 2026 г.**

*Руководитель Федеральной службы  
по интеллектуальной собственности*

Г.П. Ивалиев



## РОССИЙСКАЯ ФЕДЕРАЦИЯ



## СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

**№ 2015610891**

**Программа для распознавания и детектирования  
автомобилей, с учетом их ракурса и направления движения**

Правообладатель: *Закрытое акционерное общество "Когнитив"*  
*(RU)*

Авторы: *Полевой Дмитрий Валерьевич (RU), Услин Сергей  
Александрович (RU), Безматерных Павел Владимирович (RU)*

Заявка № **2014662041**Дата поступления **26 ноября 2014 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **20 января 2015 г.**

*Врио руководителя Федеральной службы  
по интеллектуальной собственности*

A handwritten signature in black ink, appearing to read "Л.Л. Кирий".

Л.Л. Кирий



## РОССИЙСКАЯ ФЕДЕРАЦИЯ



## СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

**№ 2015616071**

**Библиотека для распознавания в видеопотоке паспорта гражданина Российской Федерации "Smart PassportReader"**

Правообладатель: *Общество с ограниченной ответственностью  
«СМАРТ ЭНДЖИНС РУС» (RU)*

Авторы: *Арлазаров Владимир Викторович (RU), Булатов Константин  
Булатович (RU), Ильин Дмитрий Алексеевич (RU), Куроптев Андрей  
Владимирович (RU), Николаев Дмитрий Петрович (RU), Полевой Дмитрий  
Валерьевич (RU), Усилин Сергей Александрович (RU), Фараджес Игорь  
Александрович (RU), Чернов Тимофей Сергеевич (RU)*

Заявка № **2015612880**Дата поступления **10 апреля 2015 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **29 мая 2015 г.**

*Врио руководителя Федеральной службы  
по интеллектуальной собственности*

Л.Л. Кирий



# РОССИЙСКАЯ ФЕДЕРАЦИЯ



## СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

**№ 2016616961**

### Программа для распознавания идентификационных карт личности "Smart IDReader"

Правообладатель: *Общество с ограниченной ответственностью «Смарт Эндженс Сервис» (RU)*

Авторы: *Арлазаров Владимир Викторович (RU), Николаев Дмитрий Петрович (RU), Услин Сергей Александрович (RU), Булатов Константин Булатович (RU), Чернов Тимофей Сергеевич (RU), Слугин Дмитрий Геннадьевич (RU), Ильин Дмитрий Алексеевич (RU), Безматерных Павел Владимирович (RU), Муколовоз Арсений Александрович (RU), Лимонова Елена Евгеньевна (RU)*

Заявка № 2016612014

Дата поступления 10 марта 2016 г.

Дата государственной регистрации

в Реестре программ для ЭВМ 22 июня 2016 г.

Руководитель Федеральной службы  
по интеллектуальной собственности

Г.П. Ильин

