На правах рукописи

Чигорин Александр Александрович

Алгоритмы и программная система для выделения и распознавания объектов в видеопоследовательности

Специальность 05.13.11 – математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей

ДИССЕРТАЦИЯ
на соискание учёной степени
кандидата физико-математических наук

Научный руководитель:
к. ф.-м.н., доцент Конушин Антон Сергеевич

Москва – 2014

Введение

Объем информации, получаемой с помощью видеокамер растет экспоненциально от года к году. Например, количество часов видео, загружаемых каждую минуту на сайт youtube.com, увеличилось с 5 часов в 2007 году до 100 часов в 2014. На сегодняшний день подавляющее большинство видеоконтента не подвергается автоматическому анализу. В то же время извлечение информации из видео может быть полезно во многих практических приложениях, таких как мобильное картографирование, видеонаблюдение, поиск по видеоконтенту, робототехника или системы помощи водителю.

Ключевой задачей во всех перечисленных приложениях является задача выделения и распознавания объектов в видеопоследовательности. Для ее решения необходимо:

- выделить объекты на отдельных кадрах видеопоследовательности
- классифицировать объекты
- сопоставить объекты между кадрами.

Для апробации предлагаемых алгоритмов в данной работе рассматривается задача мобильного картографирования - автоматического нанесения объектов на карту. Входные данные представляют собой геопривязанный видеоряд с камеры, установленной на движущейся платформе, и параметры камеры. В результате работы алгоритма должны быть определены тип и положение искомых объектов на карте. В качестве таких объектов могут выступать любые объекты придорожной инфраструктуры: знаки дорожного движения, дорожная разметка, столбы, остановки, здания и так далее. Примеры основных этапов решения задачи приведены на рисунке 1.

В качестве области практического приложения предлагаемых алгоритмов рассматривается задача автоматического картографирования дорожных знаков. Дорожный знак является одним из главных объектов придорожной инфраструктуры. Знание о расположении знаков может быть полезно при построении и обновлении навигационных карт, в системах помощи водителю или при решении задачи учета и управления дорожной инфраструктурой. В первом случае знание положения знаков дорожного движения, их ориентации и классе позволит автоматизировать построение дорожного графа. Уменьшение доли человеческого участия в данном случае позволит уменьшить денежные затраты на обработку данных, а также позволит сократить интервалы между обновлениями дорожного графа. Во втором случае информация о

знаках позволит уведомлять водителя о текущей ситуации на дороге (рекомендуемая скорость, запрещенные повороты, запрет обгона и так далее).

Рисунок 1. Визуализация работы алгоритма автоматического нанесения объектов на карту.

Дорожные знаки сделаны, чтобы быть заметными, и имеют отличительные цвет и форму. Но разнообразие типов знаков и трансформаций над ними, встречаемых при решении задачи в промышленных масштабах, оставляют задачу высокоточного обнаружения и классификации знаков нерешенной до сих пор.

В данной диссертации рассматривается общая схема работы современных алгоритмов автоматического нанесения объектов на карту, которая состоит из четырех этапов:

1. выделение объектов интереса
2. классификация объектов
3. сопоставление изображений одного и того же объекта на соседних кадрах видеопоследовательности
4. локализация и классифицированных обнаруженных объектов на карте.

Предполагается, что на вход алгоритму поступает видеопоследовательность в цветовом формате RGB с разрешением кадров не менее 800x600 пикселей. Частота кадров видеопоследовательности должна быть не меньше 5 кадров в секунду при скорости мобильной платформы в 50 км/час. Если максимальная скорость мобильной платформы может быть выше 50 км/час, то частота кадров должна быть увеличена пропорционально изменению скорости. При этом предполагается, что частота кадров в обрабатываемом фрагменте видеопоследовательности постоянна. Главная ось камеры должна быть сонаправлена с направлением движения мобильной платформы. Минимальный размер объектов интереса на отдельных кадрах видеопоследовательности должен быть равен 24 пикселям по каждой стороне

ограничивающего прямоугольника. Для успешного обнаружения объект интереса должен быть виден минимум на двух последовательных кадрах видеопоследовательности.

На сегодняшний день лучшие методы, используемые на первых двух этапах вышеописанного алгоритма, основаны на машинном обучении и точность их работы напрямую зависит от наличия большой и репрезентативной обучающей выборки. Создание такой выборки требует больших материальных затрат. Например, в случае знаков дорожного движения создание обучающей выборки требует ручной разметки десятков километров проезда. Более того, разметку необходимо повторять для каждой новой страны, так как внешний вид знаков сильно различается в разных странах.

В данной работе исследуется возможность создания искусственных обучающих данных и алгоритмов выделения и классификации объектов, показывающих хорошую обобщающую способность при обучении на искусственных и тестировании на реальных данных.

В результаты проведенного анализа предлагается алгоритм автоматического создания искусственной обучающей выборки и несколько способов оценки качества получаемых данных. Для первых двух этапов решения исходной задачи, предлагаются алгоритмы, способные к обучению на синтетически созданных данных.

Цель диссертационной работы

Целью данной работы является разработка комплекса алгоритмов выделения, классификации и нанесения объектов придорожной инфраструктуры на карту. Разработанные алгоритмы не должны требовать пользовательского ввода и должны позволять обрабатывать большие объемы данных дешевле и быстрее ручной разметки операторами.

Научная новизна работы

В диссертации разработан алгоритм создания искусственной обучающей выборки, позволяющий сократить затраты на разметку. Предложены способы оценки качества получаемых синтетических данных в сравнении с их реальными аналогами. На примере задачи классификации знаков дорожного движения показано, что за счет обучения на большом объеме искусственно созданных данных, можно улучшить точность классификации по сравнению с обучением на реальных данных.

Предложены модификации классического алгоритма обнаружения объектов Viola-Jones [1], позволяющие увеличить скорость и точность его работы.

Также, в рамках работы над задачей классификации объектов, предложен новый алгоритм сегментация объектов от фона. Алгоритм создан для обучения на искусственных данных и позволяет оценивать параметры преобразований, которым был подвержен исследуемый объект. Предложена схема обучения на искусственных данных с использованием глубокой сверточной нейронной сети, позволяющая обучаться на искусственных выборках большого объема.

Практическая значимость и реализация

В рамках работы создана программная реализация разработанных алгоритмов, удовлетворяющая всем требованиям и ограничениям, сформулированным в цели диссертации. Для экспериментальной оценки предложенных алгоритмов была создана база знаков дорожного движения Российской Федерации (Russian Traffic Signs Dataset, RTSD). Данные для разметки были предоставлены ЗАО “Геоцентр-Консалтинг.” Чтобы позволить дальнейшее сравнение с другими методами данные сделаны общедоступными по адресу ftp://anonymous@kiviuq.graphicon.ru/AnonymousFTP/RTSD/.

Предложенные алгоритмы разрабатывались в рамках проекта по автоматическому картографированию знаков дорожного движения в компании ООО “GeoCV”.

Тестирование предложенных алгоритмов обнаружения проводилось на вышеупомянутой базе данных RTSD. Алгоритмы распознавания тестировались на четырех общедоступных база - German Traffic Signs Recognition Benchmark [2], KUL Belgium Traffic Sign Classification Benchmark [3], Sweden Traffic Signs Dataset [4], RTSD.

На всех вышеперечисленных базах были получены согласованные результаты, что подтверждает широкую применимость предложенных методов.

Апробация работы

Основные результаты работы докладывались и обсуждались на:

- 15-й международной конференции по компьютерному зрению ACIVS (Advanced Concepts for Intelligent Vision Systems), Польша, 2013
- международной конференции ISPRS (International Society for Photogrammetry and Remote Sensing), CMRT (City Models Roads And Traffics) workshop, Турция, 2013
- 3-й научно-технической конференции “Техническое зрение в системах управления”, Россия, 2012
- 14-й международной конференции DSPA (Digital Signal Processing and Applications), Россия, 2012

- 15-й международной конференции DSPA (Digital Signal Processing and Applications), Россия, 2013
- 22-й международной конференции по компьютерной графике и машинному зрению Graphicon, Россия, 2012
- научной конференции «Ломоносовские чтения», Россия, 2013.
- 6-й летней школе Microsoft для аспирантов (Microsoft Research PhD Summer School), Англия, Кембридж, 2011
- семинаре аспирантов кафедры АСВК факультета ВМК МГУ под руководством Л. Н. Королева, 2013
- семинаре группы компьютерного зрения лаборатории компьютерной графики и мультимедиа под руководством А. С. Конушина, ВМК МГУ, 2011

Публикации

По теме диссертации автором опубликовано 9 научных работ [5]-[13], из них 3 статьи в рецензируемых журналах, включенных в перечень ВАК [5,6,7]. Статья [5] была опубликована в журнале Lecture Notes in Computer Science издательства Springer. Доклад по статье [8] победил в номинации за лучшую презентацию во время проведения семинара CMRT 2013 на конференции ISPRS2013-SSG.

Содержание работы

Диссертация состоит из введения, четырех глав, заключения об основных результатах работы и списка литературы.

Первая глава посвящена задаче создания искусственной обучающей выборки: предложен алгоритм создания синтетических изображений, позволяющий покрыть большое число возможных трансформаций объекта интереса. Также предложены методы оценки качества получаемых искусственных данных.

Во второй главе рассматривается задача выделения объектов на отдельных кадрах видеопоследовательности и предлагается несколько модификаций широко известного алгоритма обнаружения объектов Viola-Jones [1], позволяющих увеличить точность работы алгоритма, не потеряв при этом в скорости его работы.

Третья глава посвящена задаче классификации обнаруженных объектов. В ней также рассмотрена актуальная на практике проблема неточной локализации объектов за счет множественных срабатываний алгоритма обнаружения в окрестности области интереса. В результате анализа проблемы предложен новый алгоритм, обучаемый на искусственно созданных данных и позволяющий уточнить положение объекта и отделить

его от фона. На примере знаков дорожного движения экспериментально показано, что сегментация от фона позволяет существенно увеличить точность последующего распознавания для широкого набора протестированных классификаторов. Также показано, что использование для классификации глубокой сверточной нейронной сети, способной к обучения на больших выборках за счет использования специализированных аппаратных платформ, позволяет при обучении на искусственных данных получать классификаторы, превосходящие по качеству аналогичные, но обученные на реальных данных.

В четвертой главе более подробно описана задача мобильного картографирования, используемая для апробации предложенных алгоритмов на практике. Предложена схема автоматизированной системы для решения задачи. Дано описание каждого из модулей системы, состоящее из входных и выходных данных, решаемой модулем задачи и описания способа ее решения.

Глава 1. Создание искусственных данных

1.1. Постановка задачи

Задачу создания набора искусственных данных, который в дальнейшем можно использовать для обучения, можно сформулировать следующим образом: по набору синтетических изображений-образцов объектов (пиктограмм) необходимо получить выборку синтетически созданных изображений, максимально приближенную к выборке реальных изображений. Иллюстрация данной задачи приведена на рисунке 2.

Рисунок 2. Слева пиктограмма объекта, к которой применяется набор трансформаций, переводящих ее в набор изображений (справа), максимально приближенных по своим характеристикам к реальным данным.

Сформулируем задачу формально. Пусть дан набор пар (p_i, r_i) , где

p_i - пиктограмма, соответствующая объекту под номером i ,

r_i - реальное изображение объекта под номером i .

Также будем считать, что задан набор возможных трансформаций над пиктограммой $T(\theta)=\{t_1(\theta_1), t_2(\theta_2), ...\}$, в котором каждая трансформация характеризуется параметрами θ_i . Обозначим через $\theta=\{\theta_1, \theta_2, ...\}$ параметры всех трансформаций из набора θ . Целью алгоритма создания искусственных данных на этапе обучения является поиск плотности распределения параметров трансформаций $p(\theta)$ над пиктограммами, позволяющей создавать искусственную выборку, максимально похожую на реальную по некоторому критерию C оценки качества

синтетических данных. В последующих разделах будут предложены и исследованы различные варианты вышеупомянутого критерия.

1.2. Обзор существующих методов

Данный обзор состоит из двух частей. Сначала рассмотрены методы, использующие синтетические данные в различных задачах компьютерного зрения. Затем, более подробно, рассмотрены методы выделения и распознавания дорожных знаков, использующие пиктограммы знаков.

1.2.1. Синтетические данные в задачах компьютерного зрения

Синтетические данные были успешно использованы при решении многих задач. В работе [14] была рассмотрена задача автоматической оценки позы человека по одному изображению с камеры глубины. Для обучения алгоритма случайного леса, в англоязычной литературе известного как random forest [15], с помощью методов компьютерной графики была создана большая выборка синтетических изображений людей, вместе с размеченными частями тела (рисунок 3).

Рисунок 3. Пример синтетических данных (слева) и реальных данных (справа).

Для создания искусственных данных распределения на параметры трансформаций были оценены на большом наборе реальных людей с помощью метода маркерного захвата движения (motion capture). После этого было произведено семплирование из оцененной плотности распределения на параметры трансформаций и по каждому набору параметров был создан синтетический обучающий пример. Использование синтетических данных позволило обучать классификатор на большом количестве примеров (порядка миллиона), что привело к значительному увеличению точности. В отличии от вышеописанного подхода метод, предложенный в данной работе, производит оценку плотности распределения параметров трансформаций, основываясь

только на изображениях реальных данных, без использования методов захвата движения и карт глубины.

Авторы [16] для решения аналогичной задачи определения позы человека также использовали синтетические данные. Различные анимированные позы человека были получены с помощью коммерческого приложения POSER [17].

В работе [18] для обучения классификатора объектов применяются 3D модели объектов из системы автоматизированного проектирования (CAD). Использование 3D моделей позволило обучать детекторы различных частей объектов, а также выводить зависимости о взаимном расположение этих частей в трехмерном пространстве, и, соответственно, на любой из двухмерных проекций (рисунок 4).

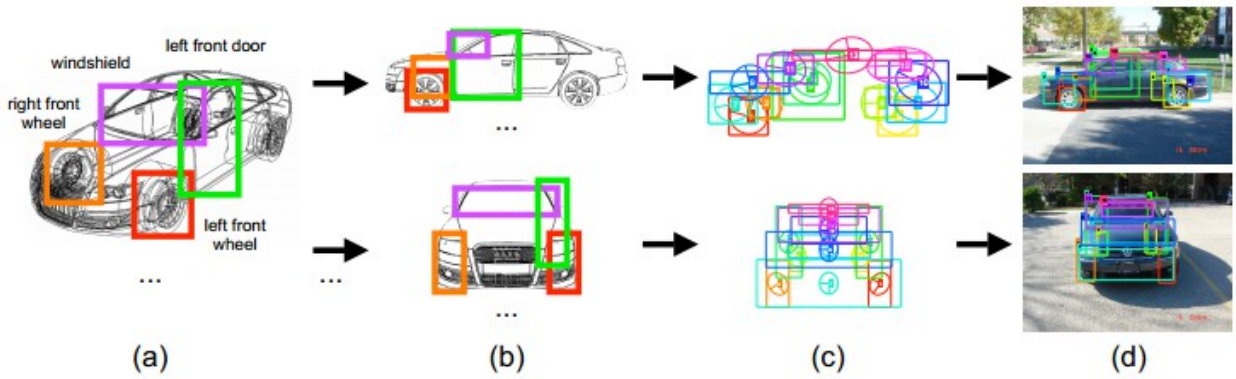


Рисунок 4. (a) Пример трехмерной модели (b) различные синтетические виды, созданные с помощью проецирования (c) модель взаимного расположения частей объектов (d) результаты обнаружения на реальных данных.

Авторы использовали дескрипторы объектов на основе формы [19], что позволило обучать классификатор на изображениях без текстуры. В подходе, предложенном в данной диссертации, параметры трансформаций оцениваются по изображениям реальных объектов. Таким образом метод не требует на входе трехмерную модель объекта, что позволяет работать с большим количеством разновидностей объектов.

Авторы [20] предлагают признаки, позволяющие оценивать трехмерную ориентацию объекта во время обнаружения. Метод является расширением метода неявных моделей формы (implicit shape models) [21] на случай оценки положения объекта в трехмерном пространстве. Использование синтетических трехмерных моделей (рисунок 5) позволило авторам обучить видонезависимый детектор объектов (рисунок 6).

Рисунок 5. Примеры трехмерных моделей объектов, использованных для обучения видонезависимого детектора объектов.

Рисунок 6. Примеры обнаружений видонезависимого детектора. В результате удается выделить не только ограничивающий прямоугольник объекта, но и оценить его расположение в трехмерном пространстве.

Обнаружение пешеходов - это еще одна задача, в которой зачастую используют искусственные данные. В [22] для создания обучающей выборки пешеходов (рисунок 7) было использовано графическое ядро компьютерной игры Half-life 2 [23]. Авторам удалось обучить детектор на искусственных данных, показывающий ту же точность, что и детектор, обученный на реальных данных.

Рисунок 7. Примеры реальных (сверху) и искусственных (снизу) изображений пешеходов.

Авторы [24] использовали технологию захвата движения без маркеров для оценки параметров трансформаций, возможных над телом человека. После этого на вход системе подавался пример человека, взятый из реальных данных, и на его основе создавалось множество обучающих синтетических примеров, на которых человек представал в различных позах и с разным фоном (рисунок 8). В результате экспериментов авторы приходят к выводу, что 11 примеров из реальных данных может быть достаточно для того, чтобы приблизиться к результатам детекторов, обучаемых на выборках из реальных данных гораздо большего объема.

Синтетически созданные примеры символов были использованы в [25] и [26] для обучения классификаторов символов. Отдельные символы обычно встречаются на однородном фоне и имеют однородный цвет, что делает их достаточно простым объектом для генератора синтетических данных (рисунок 9).

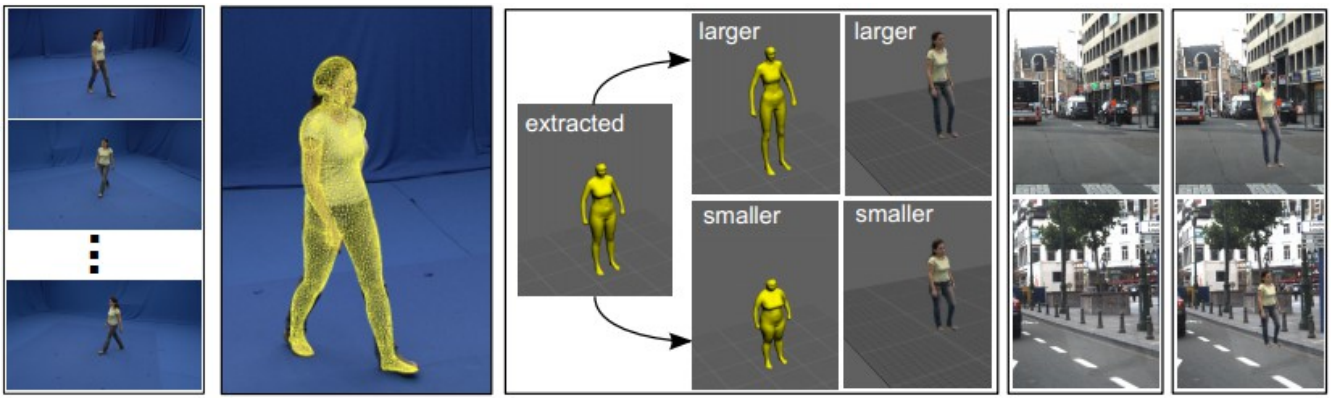


Рисунок 8. Создание синтетических изображений пешеходов. Оценка позы человека без использования маркеров. Оценка вариаций параметров человека и возможных жестикуляций. Создание синтетических изображений пешехода с разным фоном.

Рисунок 9. Сравнение искусственно созданных символов (вверху) с реальными (внизу).

В [27] предложен подход к обучению детектора лиц на основе искусственно созданных данных, получаемых на основе трехмерных моделей лиц с помощью варьирования атрибутов лица, таких как пол, возраст или масса тела. Для получения усредненной 3D модели лица была использована база, состоящая из 512 трехмерных моделей лиц. Каждая модель была описана с помощью набора $S_i=\{x_1, y_1, z_1, \ldots, x_n, y_n, z_n\}$ трехмерных точек (где $n=75972$) и соответствующего ему набора цветов, соответствующих этим точкам $T_i=\{R_1, G_1, B_1, \ldots, R_n, G_n, B_n\}$. Авторы

применяли метод анализа главных компонент (PCA) отдельно к вектору S_i и T_i , снижая размерность до 511 главных компонент. Затем, для получения вариаций в лицах, использовался метод добавления шумов к первым 50 компонентам разложения, содержащим наиболее важные трансформации. Для того, чтобы получить вариации в облике лиц по различным атрибутам авторы вручную разметили обучающую выборку для каждого атрибута и с ее помощью искали направление в пространстве, полученном после применения PCA, варьирование параметров вдоль которого соответствует варьированию одного из атрибутов. Пример варьирования внешнего облика лица по атрибуту “наличие волосяной растительности на лице” приведен на рисунке 10.

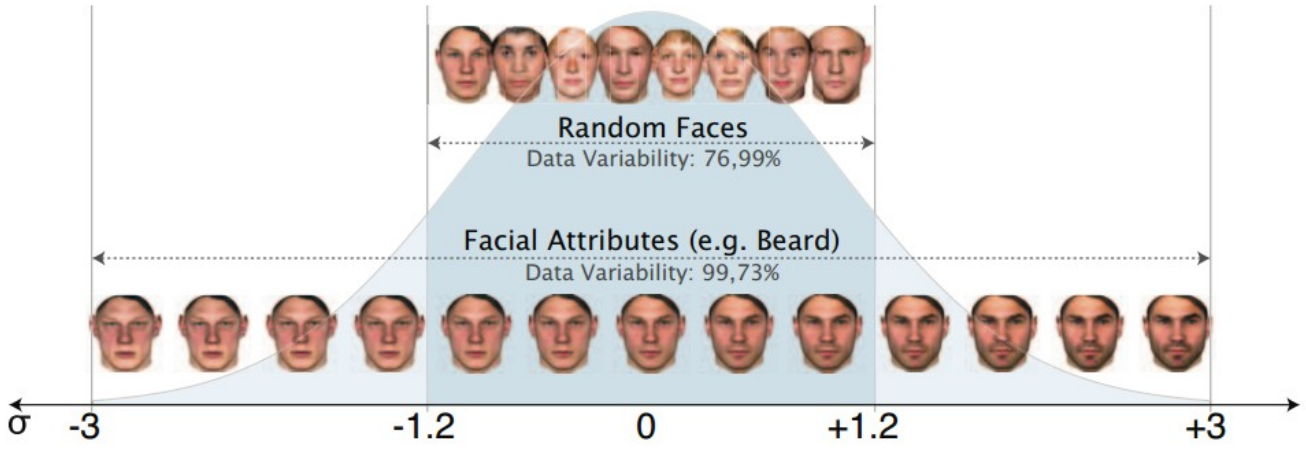


Рисунок 10. Пример синтетических лиц с вариацией по атрибуту лица (в данном случае по наличию волосяной растительности на лице).

1.2.2. Искусственные данные в задаче выделения и классификации знаков дорожного движения

Синтетические обучающие данные были также использованы для решения задач обнаружения и классификации дорожных знаков. В данном случае на вход алгоритму созданию синтетических данных, как правило, поступают пиктограммы знаков, примеры которых легкодоступны в интернете. Также были предложены подходы, работающие непосредственно с пиктограммами и не требующие создания обучающей выборки, но нет свидетельств получения хороших результатов с помощью этих методов на современных общедоступных базах.

В работе [28] обнаружение и распознавание знаков осуществлялось за счет сопоставления контуров пиктограммы, описанных с помощью дескрипторов Фурье [28], с

контурами, извлеченными из реальных изображений. Похожий подход на основе контуров пиктограмм был предложен в [29].

Наиболее близка к предлагаемому в данной диссертации методу работа [30], в которой производится попытка обучения детектора дорожных знаков на синтетически созданной обучающей выборке (рисунок 11).

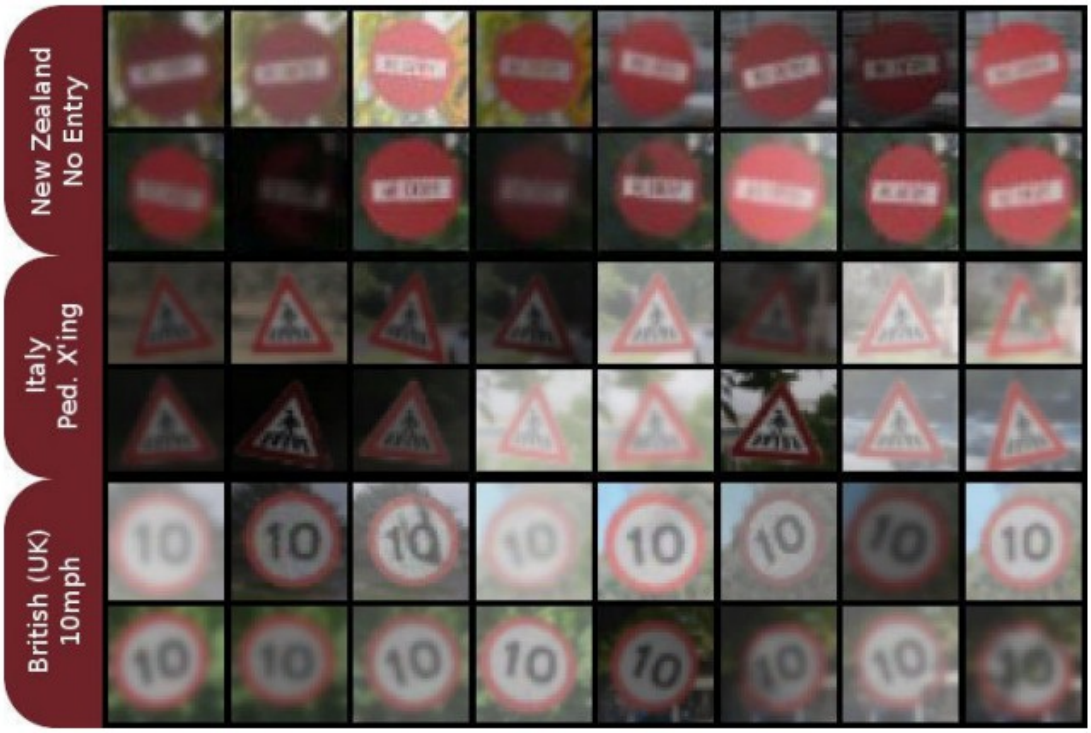


Рисунок 11. Пример получаемых синтетических данных знаков.

В предложенном авторами методе синтетические изображения создавались на основе пиктограммы знака и реальных изображений с фоном, подкладываемых под знак. Авторы предложили эвристический алгоритм создания синтетической выборки, параметры преобразований в котором подобраны таким образом, чтобы детектор знаков на основе метода Viola-Jones [1] был способен обучиться до приемлемой точности. В случае, если итерация обучения детектора проваливалась, авторы предложили уменьшать вариативность в данных, тем самым упрощая задачу детектора. Данная проблема возникает на последних этапах каскада детектора и, в нашем случае, решается за счет использования классификатора на основе сверточной нейронной сети, позволяющего аппроксимировать всё разнообразие вариаций, встречаемых в синтетических данных. Авторам [30] удалось достичь точности работы детектора, обученного на синтетических данных, сравнимой с точностью работы детектора,

обученного на реальных данных. Но экспериментальная оценка была произведена только на трех классах знаков (они показаны на рисунке 11), для каждого из которых был обучен свой детектор (что сильно замедляем скорость работы практической системы, учитывая, что всего существует более 200 классов знаков). Наши эксперименты показали, что данные классы знаков являются наиболее простыми для обнаружения. В данной диссертации эксперименты проводятся на четырех типах знаков (*красные круги*, *синие квадраты*, *красные треугольники* и *синие круги*), каждый из которых включает в себя десятки классов знаков. Это делает предлагаемые алгоритмы более применимыми на практике, так как один обученный детектор единовременно покрывается большое количество классов знаков, тем самым увеличивая скорость работы алгоритма.

1.3. Предлагаемые алгоритмы

Для решения задачи, поставленной в пункте 1.1, набор потенциальных трансформаций $T(\theta)$ (а также порядок их применения) должен быть задан вручную. Оценка качества получаемых искусственных данных производилась на основе одного из перечисленных ниже критериев C :

1. через оценку точности получаемых в результате обучения на искусственных данных классификаторов (оценка точности классификаторов производится на реальных данных)
2. с помощью оценки качества аппроксимации реальных данных
3. с помощью визуальной оценки качества синтетических данных.

Чтобы зафиксировать набор трансформаций $T(\theta)$ мы начали с критерия экспериментальной оценки точности получаемых классификаторов. Для этого были проведены эксперименты (см. раздел 1.4.2.) по оценке качества получаемых классификаторов в зависимости от параметров трансформаций, которым подвергались входные пиктограммы. В результате был зафиксирован набор и порядок трансформаций, через которые должна проходить входная пиктограмма (рисунок 12):

1. сегментация маски знака из входной пиктограммы, исходя из предположения, что фоновые пиксели занимают наибольшую однородную внешнюю область пиктограммы
2. трансформация из цветового пространства RGB в пространство HSV
3. изменение яркости V и насыщенности S
4. вращение пиктограммы и маски знака относительно трех осей координат на углы R_x, R_y, R_z

5. обрезка знака по повернутой маске знака
6. гауссово размытие со среднеквадратичным отклонением σ_B
7. добавление отступов (dx_l, dx_r, dx_u, dx_d) с четырех сторон объекта. Этот шаг особенно важен в случае, если модуль распознавания обрабатывает результаты работы модуля обнаружения, который обычно выдает несколько альтернативных гипотез обнаружений вокруг знака
8. масштабирование до нескольких размеров $s_{1,} s_{2,} \ldots, s_n$ и обратно до целевого размера классификатора (в нашем случае 30x30 пикселей). Данная трансформация моделирует процесс масштабирования окна детектора меньшего по размеру, чем входной размер классификатора (upscale)
9. добавление нормально шума со среднеквадратичным отклонением σ_N к каждому пикселю изображения. Эта трансформация моделирует шум, добавляемый камерой
10. добавление фона из реальных изображений. Изображение фона преобразовывается таким образом, чтобы его средняя интенсивность была равна средней интенсивности знака.

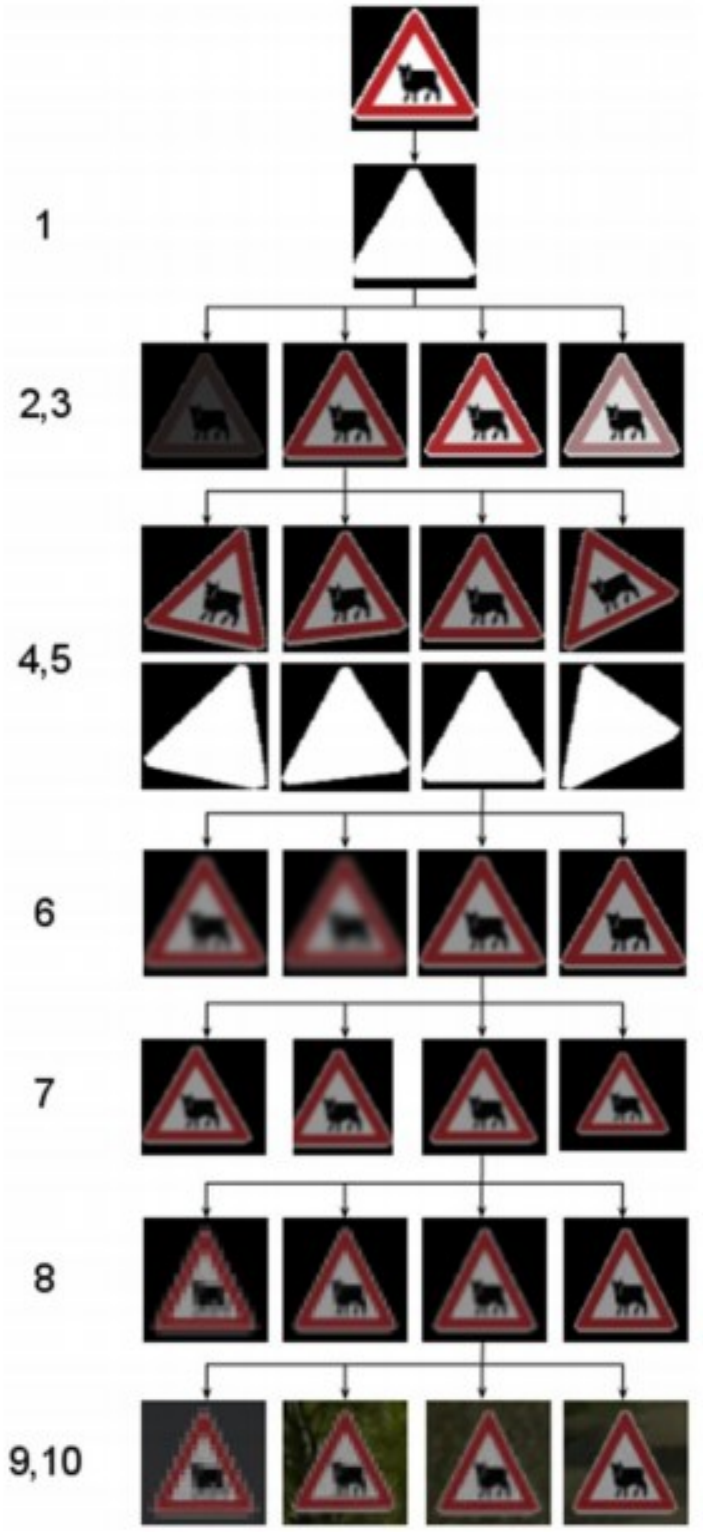


Рисунок 12. Визуализация основных шагов алгоритма создания синтетических данных.

1.3.1. Метод на основе точности получаемых классификаторов

В методе экспериментальной оценки предлагается оценивать плотность вероятности параметров трансформаций $p(\theta)$ с помощью экспериментов, показывающих важность тех или иных трансформаций для достижения итоговой целевой метрики - качества получаемых классификаторов объектов. При этом имеет смысл использовать метод классификации, который бы позволял легко интерпретировать получаемые результаты.

1.3.2. Метод на основе оценки точности приближения реальных данных

Для каждой пары (p_i, r_i) найти $\theta^* = argmin_\theta(\|(r_i - f(p_i, T(\theta)))\|_2)$, где $f(p_i, T(\theta))$ - последовательное применение трансформаций к пиктограмме объекта.

Данную задачу можно решать с помощью методов локальной или глобальной оптимизации. Таким образом для каждого примера из обучающей выборки (p_i, r_i) мы получим оптимальную трансформацию θ^*. Затем с помощью одного из методов оценки плотности распределения вероятности можно оценить $p(\theta)$.

1.4. Экспериментальная оценка

1.4.1. Базы данных, использованные в экспериментах

German traffic sign recognition benchmark (GTSRB) [2] состоит из более чем 50000 изображений немецких знаков дорожного движения и содержит 43 класса знаков. Обучающая и тестовая выборка состоят из 39209 и 12630 изображений, соответственно. Три лучших результата были заявлены в работах [31] - 99.46%, [32] - 98.31%, [33] - 96.14%. Интересно, что точность распознавания человеком (98.84%) на данном наборе данных хуже, чем точность наилучшего из упомянутых методов.

1.4.2. Метод оценки на основе точности получаемых классификаторов

Для оценки важности предлагаемых трансформаций был проведен ряд экспериментов, призванных оценить точность классификации реальных данных при изменении параметров трансформаций. Для этого для фиксированного набора трансформаций создавалась синтетическая обучающая выборка, на которой впоследствии обучался классификатор. После этого точность классификации измерялась на реальной выборке объектов. В качестве дескриптора изображений был выбран HOG-дескриптор [34] с параметрами, приведенными в таблице 1. В качестве

классификатора был использован метод на основе поиска ближайшего соседа в пространстве HOG-дескрипторов. Данный подход удобен тем, что позволяет легко интерпретировать получаемые результаты.

Имя набора	Размер ячейки	Размер блока	Шаг блока	Количество ячеек гистограммы	Точность [%]
HOG1	5x5	10x10	5x5	8	92.35
HOG2	4x4	8x8	2x2	8	94.06
HOG3	3x3	6x6	3x3	9	95.03

Таблица 1. Параметры HOG-дескриптора для эксперимента с оценкой важности параметров трансформаций.

В таблице 2 приведены точности классификации, получаемые при различных вариация в параметрах трансформаций. Яркость V и насыщенность S измеряются в процентах относительно соответствующих значений в исходном изображении. Углы поворота (R_x, R_y, R_z) - в градусах. Система координат, использованная при вращении объектов в 3D, показана на рисунке 13. Параметры отступа относительно края (dx_l, dx_r, dy_u, dy_d) измерены в процентах относительно целевых размеров изображения классификатора (в нашем случае 30x30 пикселей). Размер s получаемых изображений измеряется в пикселях. В таблице использована нотация (*левая_граница* : *шаг* : *правая_граница*) для более компактного описания вариаций параметров. В данной нотации конструкция -10:10:30 означает, что был использован набор значений (-10, 0, 10, 20, 30). Эксперименты были проведены на базе German Traffic Signs Recognition Benchmark [2].

#	V [%]	S [%]	R_x	R_y	R_z	σ_B	dx_l [%]	dx_r [%]	dy_u [%]	dy_d [%]	размер (s) [px]	σ_N	acc. [%]
1	70	50	0	0	0	2	0	0	0	0	30x30	1.5	85.28
2	70	50	0	0	0	2	-4,0,4	-4,0,4	-4,0,4	-4,0,4	30x30	1.5	90.53
3	70	50	0	0	0	2	-4:2:4	-4:2:4	-4:2:4	-4:2:4	30x30	1.5	91.08
4	70	50	-20,0,20	-30,0,30	-6,0,6	2	-4,0,4	-4,0,4	-4,0,4	-4,0,4	30x30	1.5	93.15
5	20,50,80	50	0	0	0	2	-4,0,4	-4,0,4	-4,0,4	-4,0,4	30x30	1.5	90.56
6	70	20,50,80	0	0	0	2	-4,0,4	-4,0,4	-4,0,4	-4,0,4	30x30	1.5	90.83
7	70	50	0	0	0	0,2,4	-4,0,4	-4,0,4	-4,0,4	-4,0,4	30x30	1.5	91.10
8	70	50	0	0	0	2	-4,0,4	-4,0,4	-4,0,4	-4,0,4	15x15, 20x20, 30x30	1.5	90.03

Таблица 2. Зависимость точности классификации от параметров трансформаций.

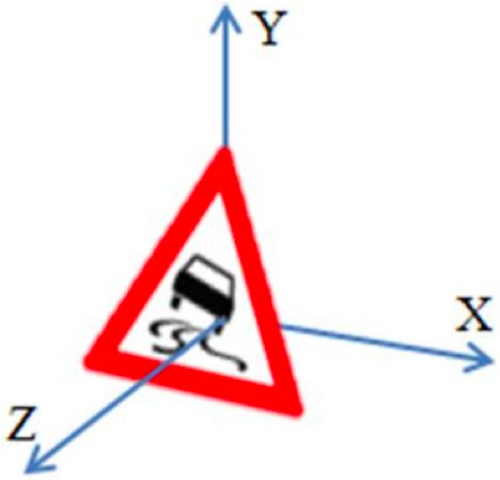


Рисунок 13. Система координат, использованная при вращении объекта в трехмерном пространстве.

Из таблицы 2 можно сделать следующие выводы:

1. если не добавлять дополнительных смещений и поворотов, то получается наиболее низкая точность классификации - 85.28% (набор №1)

2. добавление дополнительных смещений увеличивает точность классификации до 90.53% (набор №2)
3. если увеличить плотность семплирования значения параметра смещения, то удается добиться улучшения точности до 91.08% (набор №3)
4. добавление поворотов к набору №2 увеличивает точность до 93.15%
5. вариации параметров, приведенные в наборах №5-№8, не изменяют точность классификации.

Таким образом данный подход позволяет экспериментально определить набор параметров трансформаций, приводящих к увеличению показателей целевой метрики - точности классификации объектов.

Использованный подход на основе метода ближайшего соседа позволяет легко интерпретировать получаемые результаты классификации, а также строить предположения о том, какие параметры трансформаций необходимо добавить для того, чтобы улучшить точность распознавания. На рисунке 14 приведены примеры входного реального изображения и четырех его ближайших соседей. А на рисунке 15 приведены наиболее частотные случаи ошибок алгоритма, по которым, во многих случаях, легко предположить каких трансформаций недостает в искусственной выборке.

Рисунок 14. Примеры корректного поиска ближайших соседей для входного изображения реального знака. Левый столбец - реальное изображение-запрос. Остальные столбцы - примеры четырех ближайших соседей из синтетической выборки.

Рисунок 15. Примеры ошибок, возникающих при поиске ближайшего соседа. Левый столбец - реальное изображение-запрос. Остальные столбцы - примеры четырех ближайших соседей из синтетической выборки. Видно, что предложенный метод позволяет легко интерпретировать получаемые результаты.

Примеры получаемых синтетических изображений знаков в сравнении с реальными изображениями приведены на рисунке 16.

Рисунок 16. Сверху приведены примеры реальных изображений знаков. Снизу - примеры синтетически полученных изображений.

1.4.3. Метод на основе оценки точности приближения реальных данных

Точность приближения входного реального изображения можно измерить за счет поиска наиболее близкого (по некоторой метрике) изображения в искусственной выборке. В данной работе были проведены эксперименты, использующие в качестве метрики похожести SAD (sum of absolute differences, сумма абсолютных разностей), которую можно записать следующим образом:

$$D = \sum_{p \in pixels} \left| I_1^{(p)} - I_2^{(p)} \right|$$, где

$I_i^{(p)}$ - p -й пиксель первого изображения под номером i .

Для многоканальных изображений вычисления производятся аналогичным образом за счет обработки каждого канала по отдельности и последующего суммирования. Для оценки качества приближения синтетической выборкой набора реальных изображений мы предлагаем использовать усредненное расстояние по всем реальным изображениям до ближайших искусственных примеров:

$$Q = \frac{\sum_{i=1}^{N} min_{j=1,M}\left(SAD\left(I_r^{(i)}, I_s^{(j)}\right)\right)}{N}$$, где

N, M - количество реальных и искусственных изображений, соответственно,

$I_r^{(i)}$ - реальное изображение под номером i ,

$I_s^{(j)}$ - искусственное изображение под номером j .

Помимо оценки качества синтетических данных данный подход также позволяет оценить плотности распределения параметров трансформаций, так как для каждого

реального изображения можно получить параметры трансформаций ближайшего к нему искусственного примера.

Чтобы убедиться, что методы оценки важности параметров на основе точности получаемых классификаторов и на основе точности приближения реальных данных дают похожие результаты - был проведен следующий эксперимент: используя параметры трансформаций из таблицы 3 (ранее использованные для оценки качества синтетических данных по точности классификации) качество приближения реальных данных было оценено на основе метрики SAD.

#	V [%]	S [%]	R_x	R_y	R_z	σ_B	dx_l [%]	dx_r [%]	dy_u [%]	dy_d [%]	размер (s) [px]	σ_N	avg. SAD
1	70	50	0	0	0	2	0	0	0	0	30x30	1.5	40.87
2	70	50	0	0	0	2	-4,0,4	-4,0,4	-4,0,4	-4,0,4	30x30	1.5	36.38
3	70	50	0	0	0	2	-4:2:4	-4:2:4	-4:2:4	-4:2:4	30x30	1.5	36.30
4	70	50	-20,0,20	-30,0,30	-6,0,6	2	-4,0,4	-4,0,4	-4,0,4	-4,0,4	30x30	1.5	34.95
5	20,50,80	50	0	0	0	2	-4,0,4	-4,0,4	-4,0,4	-4,0,4	30x30	1.5	34.85
6	70	20, 50, 80	0	0	0	2	-4,0,4	-4,0,4	-4,0,4	-4,0,4	30x30	1.5	34.91
7	70	50	0	0	0	0,2,4	-4,0,4	-4,0,4	-4,0,4	-4,0,4	30x30	1.5	36.35
8	70	50	0	0	0	2	-4,0,4	-4,0,4	-4,0,4	-4,0,4	15x15, 20x20, 30x30	1.5	35.82

Таблица 3. Зависимость точности приближения по метрике SAD от параметров трансформаций.

Из сравнения таблиц 2 и 3 видно, что оба метода дают скоррелированные результаты по оценке важности используемых трансформаций с точностью до выбранного способа описания изображений (дескриптора). В случае оценки важности трансформаций через точность классификации в качестве дескриптора выступают гистограммы ориентированных градиентов, инвариантные к изменению яркости и насыщенности изображений. Во втором случае изображения были описаны с помощью RGB значений цвета, что привело к увеличению важности параметров яркости и насыщенности (таблица 3, строки 5 и 6).

Из таблицы 4 видно, что точность классификации изображений на базе GTSRB хорошо скоррелирована (коэффициент корреляции Пирсона равен 0.98) с точностью приближения изображений синтетическими данными по метрике SAD, что также является свидетельством в пользу выбранной метрики.

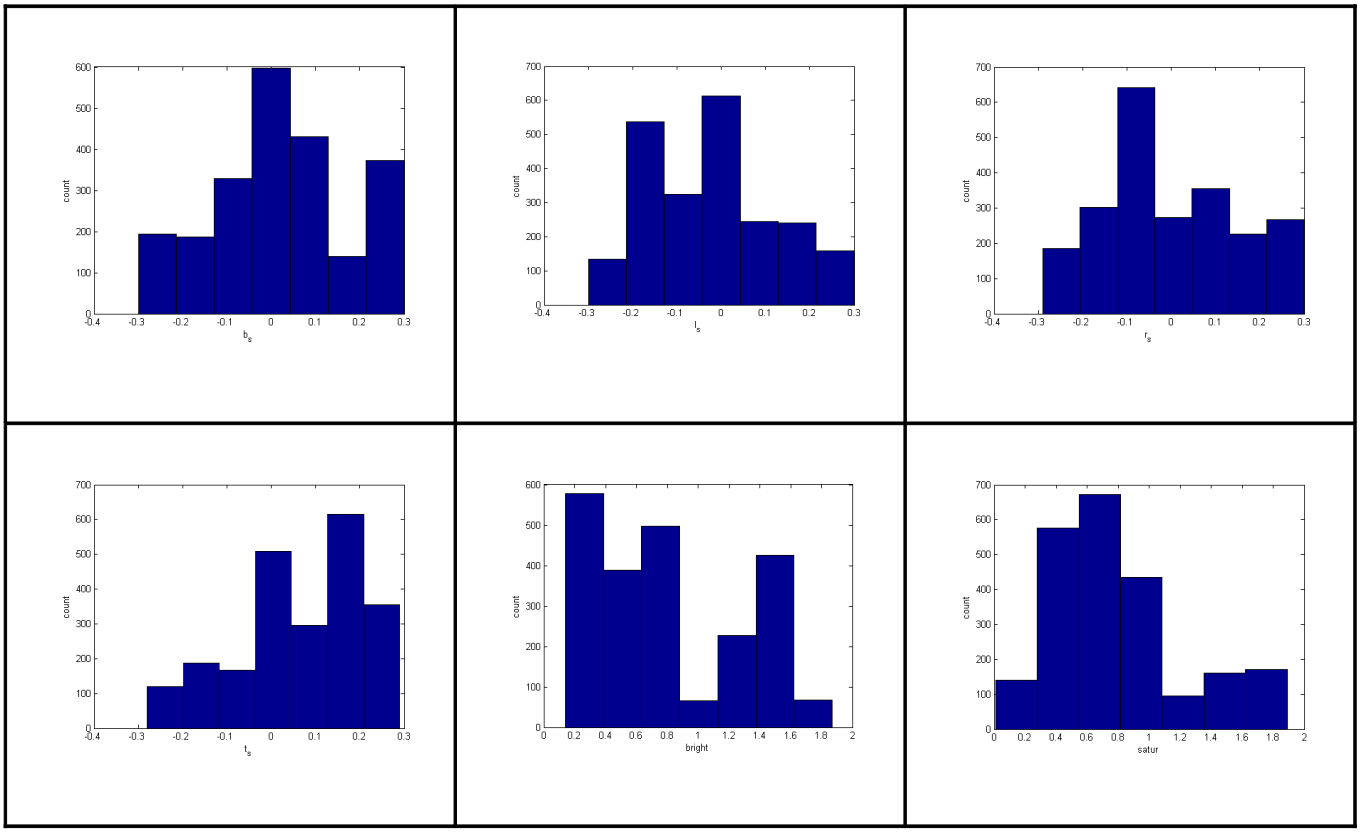
Также, как было сказано ранее, с помощью предложенного подхода можно оценить плотности распределения на параметры трансформации $p(\theta)$. Для этого нужно создать искусственную выборку, состоящую из примеров, прошедших через случайные трансформации. После этого с помощью предложенного метода можно провести поиск ближайшего соседа для каждого реального изображения, что позволит оценить параметры преобразования, которым было подвержено входное изображения. Оценив параметры преобразования для набора изображений, можно построить гистограммы встречаемости тех или иных параметров преобразования, которые в свою очередь можно преобразовать в функции плотности распределения $p(\theta)$ с помощью одного из методов оценки плотности.

Апробация предложенного метода была произведена на знаке “*ограничение скорости 50 км/ч*”, 2250 реальных изображений которого из базы GTSRB были использованы для оценки плотности распределения $p(\theta)$. Для этого была создана выборка искусственных изображений вышеупомянутого класса, состоящая из 1000 изображений с известными параметрами трансформаций. Далее, для каждого реального изображения был найден ближайший сосед в искусственной выборке по метрике SAD. Таким образом для каждого реального изображения были оценены параметры трансформаций, которым оно было подвержено. Полученные гистограммы значений параметров трансформаций приведены в таблице 5.

Количество изображений на класс	Точность аппроксимации (SAD)	Точность классификации (1-NN) [%]
1	35.05	9.0
2	33.30	14.4
4	31.64	19.7
8	30.26	21.9
16	29.15	25.2
32	27.75	30.4

64	26.76	35.8
128	26.14	38.0
256	25.74	44.4
512	24.24	48.2
1024	24.20	52.6
2048	23.63	57.0
4096	22.81	60.8
8192	22.57	62.3
16392	22.14	64.0
32784	21.91	65.1

Таблица 4. Зависимость между количеством изображений на класс, точностью приближения по метрике SAD и точностью классификации изображений на базе GTSRB.



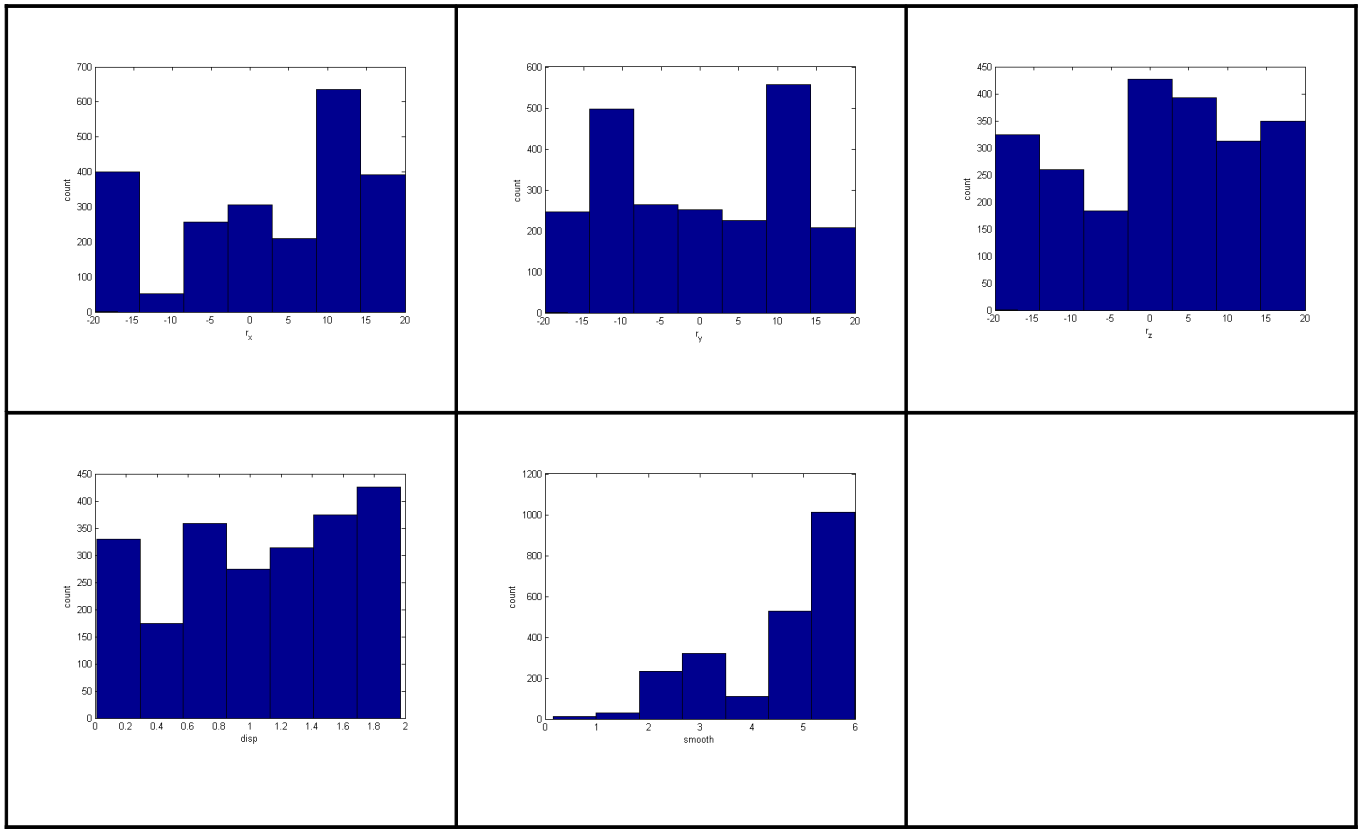


Таблица 5. Гистограммы распределения значений параметров трансформаций. Имена параметров слева-направо сверху-вниз: сдвиг снизу, сдвиг сверху, сдвиг справа, сдвиг слева, коэффициент яркости, коэффициент насыщенности, повороты вдоль осей xyz, дисперсия шума, коэффициент размытия.

После этого было создано два варианта искусственных выборок, состоящих из трех классов знаков “*ограничение скорости 20 и 70*”, “*предупреждающий знак пешеходных переход*”. В первом варианте 1000 искусственных изображений на класс создавалось по тому же принципу, что и для *знака ограничения скорости 50*, то есть параметры трансформаций семплировались из равномерного распределения. Во втором случае были использованы оценки плотности распределения на параметры трансформаций $p(\theta)$ из таблицы 5. Таким образом параметры семплировались из распределений $p(\theta)$, оценка которых была произведена на знаке “*ограничение скорости 50*”. После этого была проведена оценена точность классификации 2430 реальных изображений вышеупомянутых классов. В результате были получены точности классификации, представленные в таблице 6. Видно, что использование оценок на параметры трансформаций, сделанных на основе небольшого количества изображений стороннего класса, помогает значительно увеличить точность классификации изображений других классов.

Распределение на параметры трансформаций	Точность аппроксимации (SAD)	Точность классификации (1-NN) [%]
равномерное	32.99	70.2
оцененное по знаку “*ограничение скорости 50*”	30.88	79.2

Таблица 6. Зависимость точности классификации от плотности распределения параметров трансформаций, использованной при создании искусственных изображений.

1.5. Заключение

В данной главе был проведен обзор методов распознавания изображений, использующих в ходе своей работы искусственные данные. Отдельно, более детально, были рассмотрены методы распознавания знаков дорожного движения. Задача создания синтетической выборки изображений объектов была сформулирована как задача оценки плотности распределения параметров трансформаций, производимых над каноническим изображением объекта (пиктограммой). Было предложено два метода оценки качества получаемых искусственных данных: основанный на оценке точности классификации реальных данных и основанный на оценке среднего расстояния от реальных до искусственных данных по некоторой метрике. Проведен ряд экспериментов, показывающих возможность оценки качества получаемых искусственных данных с помощью предложенных методов. Для второго из предложенных методов также представлены эксперименты, демонстрирующие возможность оценки плотности распределения параметров трансформаций реальных данных. Показана возможность улучшения точности классификации на основе искусственных данных за счет использования плотности распределения параметров трансформаций, оцененной на реальных данных.

Глава 2. Выделение объектов

2.1. Постановка задачи

Пусть на входе дано изображение I со списком присутствующих на нем объектов $(o_{1,}\dots,o_n)$, заданных с помощью ограничивающих прямоугольников $o_i=(x,y,h,w)$.

В задаче выделения объектов, используя только изображение, необходимо найти набор прямоугольников $(r_{1,}\dots,r_n)$ покрывающих как можно больше объектов. Считают, что найденный прямоугольник r_i покрывает объект o_j , если отношение площади пересечения к площади объединения больше некоторого заданного порога τ . Обычно τ берут равным 0.5. Такое же определение корректного обнаружения использовано и в данной диссертации.

2.2. Обзор существующих методов

Данный обзор состоит из двух частей. В первой части описаны подходы, предложенные для выделения объектов любого рода. Во второй части рассмотрены подходы к выделению знаков дорожного движения, учитывающие специфику задачи.

2.2.1. Методы выделения объектов

В настоящее время наиболее популярным подходом к выделению объектов на изображениях является метод на основе идеи “скользящего окна” [1]. В данном подходе классификатор последовательно анализирует небольшие области изображения (называемые окнами) на предмет присутствия в них искомого объекта (рисунок 17). Для выделения объектов разного размера поиск производится на изображениях разного масштаба. Данный подход требует, чтобы вычислительная сложность одного приложения окна была небольшой, так как на одном изображении приходится обрабатывать миллионы окон.

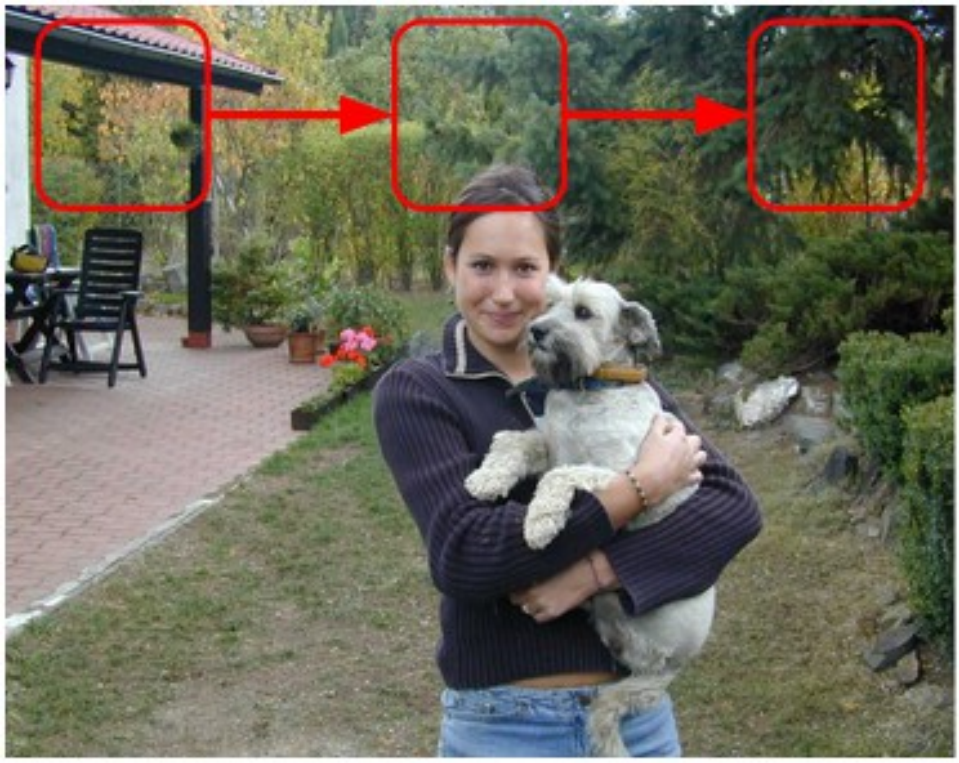


Рисунок 17. Схема работы метода на основе “скользящего окна”.

Наиболее ярким представителем этого направления является работа [1]. Она будет описана подробно, так как является базовой для данной диссертации. Виола и Джонс предложили несколько модификаций, позволивших существенно ускорить время работы детектора, не потеряв при этом в точности его работы. Перечислим предложенные модификации:

1. вместо одноуровневой схемы классификации используется каскад классификаторов. Каждый последующий классификатор работает только с теми окнами, которые были классифицированы всеми предыдущими классификаторами каскада как содержащие объект. Данный подход позволяет использовать на первых этапах каскада простые, но быстрые классификаторы, отклоняющие большую часть окон-кандидатов. А на последних этапах, до которых доходит лишь небольшая часть окон-кандидатов, появляется возможность использовать медленные, но более точные алгоритмы
2. использование фильтров Хаара в качестве признаков (рисунок 18). Данная модификация позволила быстро вычислять признаки, за счет использования интегральных изображений (summed area table, рисунок 19) [35]

3. использование классификатора на основе бустинга [36]. В работе [1] был применен алгоритм обучения AdaBoost. Данный подход позволил строить классификаторы желаемой точности на основе жадного алгоритма, позволяющего выбирать наиболее полезные признаки из большого набора признаков-кандидатов.

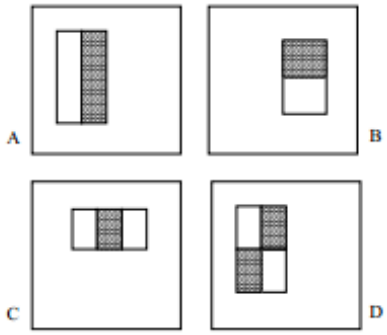


Рисунок 18. Признаки Хаара. Откликом каждого признака Хаара является сумма пикселей внутри прямоугольников. При этом сумма внутри белых прямоугольников берется с положительным, а сумма внутри черных прямоугольников - с отрицательным знаком.

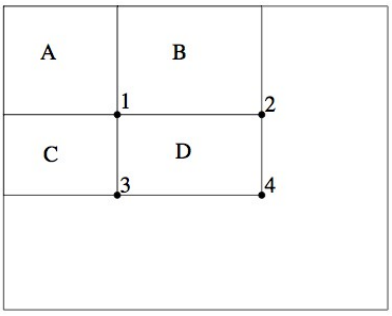


Рисунок 19. Интегральное изображение. Сумма значений пикселей внутри прямоугольника D может быть посчитана за 4 обращения к массиву. Значение интегрального изображения в точке 1 равно сумму пикселей в прямоугольнике A. Значение в точке 2 равно A+B, в точке 3 равно A + C, а в точке 3 сумма равна A + B + C + D. Сумма значений пикселей внутри прямоугольника D может быть посчитана следующим образом 4 + 1 - (2 + 3).

В таблице 7 приведен алгоритм обучения каскада классификаторов, предложенный авторами.

- Пользователь выбирает значение f , максимально допустимый уровень ложных срабатываний на уровне каскада и d , минимальную полноту детектора.
- Пользователь выбирает целевой уровень ложных срабатываний F_{target} .
- P, N - набор положительных и отрицательных примеров, соответственно.
- $F_0 = 1.0, D_0 = 1.0, i = 0.$
- Пока $F_i > F_{target}$
  - $i = i + 1$.
  - $n_i = 0, F_i = F_{i-1}$.
  - Пока $F_i > f \times F_{i-1}$
    - $n_i = n_i + 1$.
    - Использовать P и N , чтобы обучить классификатор на n_i признаках с использованием AdaBoost.
    - Посчитать текущие F_i и D_i на валидационной выборке.
    - Уменьшить порог для i -го классификатора таким образом, чтобы полнота текущего классификатора была не меньше $d \times D_{i-1}$ (это также влияет на F_i).
  - $N = 0$.
  - Если $F_i > F_{target}$, то запустить текущий каскад на наборе изображений, не содержащих лиц и поместить все ложные срабатывания в N .

Таблица 7. Алгоритм обучения детектора лиц, предложенный в [1].

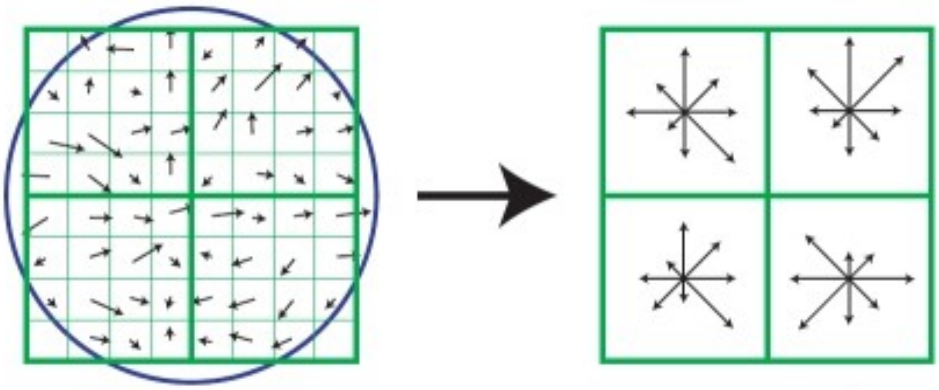


Рисунок 20. Схематическое представление направлений градиентов, посчитанных в отдельных пикселах изображения, которые впоследствии формируют гистограммы ориентированных градиентов, описывающих небольшую область изображения.

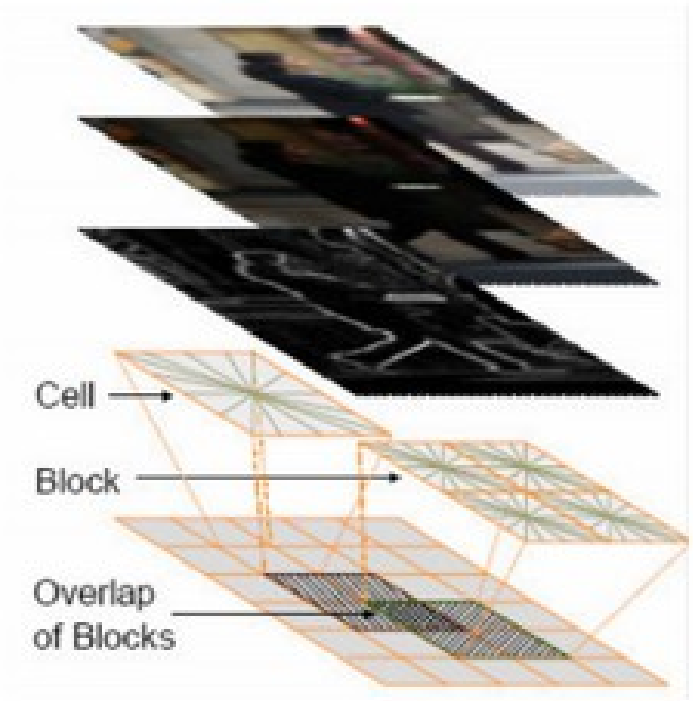


Рисунок 21. Построение гистограмм ориентированных градиентов по изображению. Гистограммы градиентов считаются в отдельных ячейках (cell) и впоследствии нормализуются по соседним ячейкам (block).

Метод [1] хорошо и быстро работает на негибких объектах, не способных значительно изменять свой внешний вид за счет изменения положения своих частей

относительно друг друга. Чтобы обойти это ограничение [34] предложили использовать гистограммы ориентированных градиентов (histogram of oriented gradients, HOG). Идея HOG состоит в разбиении изображения прямоугольной сеткой на ячейки и построении гистограммы градиентов в каждой из ячеек (рисунок 20), с последующей нормализацией гистограмм по соседним ячейкам. Схематически процесс построения гистограмм показан на рисунке 21.

Dalal и Triggs предложили использовать HOG для подсчета признаков в задаче обнаружения людей (рисунок 22). Преимуществом HOG перед методом Viola-Jones является инвариантность к существенным трансформациям объекта. Совместно с обобщающей способность метода опорных векторов (SVM) это позволило достичь наилучшей точности обнаружения людей на изображениях.

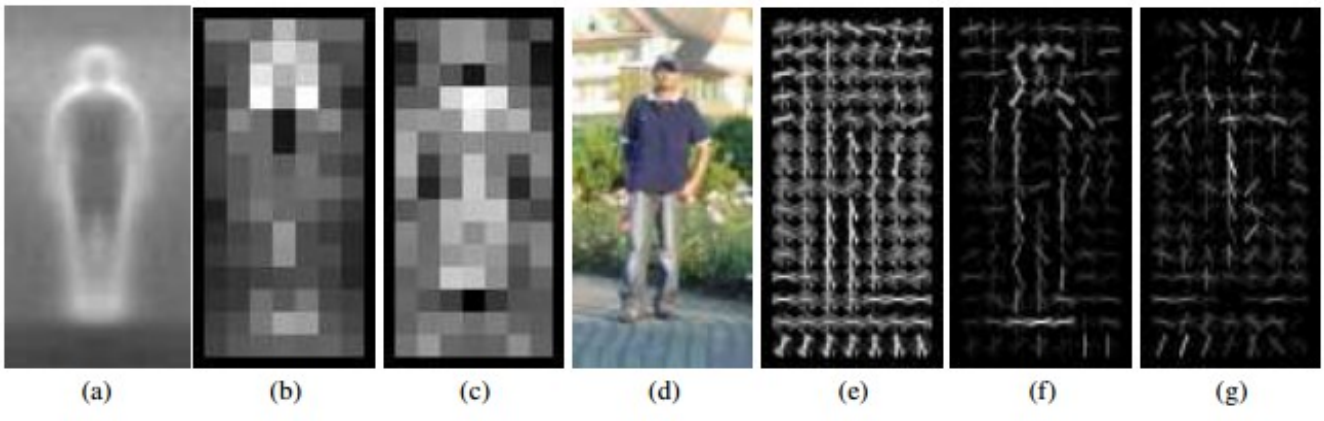


Рисунок 22. Гистограммы ориентированных градиентов для обнаружения людей: a) среднее значение градиента в окрестности точки b) максимальный положительный отклик линейного SVM в окрестности точки c) соответственно, максимальный отрицательный отклик d) исходное изображение e) визуализация гистограмм градиентов в каждой ячейке f,g) гистограммы градиентов, взвешенные обученными весами SVM (слева - положительными, справа - отрицательными).

Подход на основе HOG+SVM был развит в работе [37], в которой было предложено дополнить описание изображения объекта (как в [34]) представлением объекта в виде взвешенной композиции отдельных его частей (рисунок 23). Авторы назвали предложенную модель - деформируемой моделью частей (deformable parts model, DPM). При этом не накладывалось никаких дополнительных ограничений на разметку в обучающей выборке. Таким образом, расположение и возможное отклонение частей

объектов выводилось автоматически из набора ограничивающих прямоугольников, выделяющих объекты целиком.

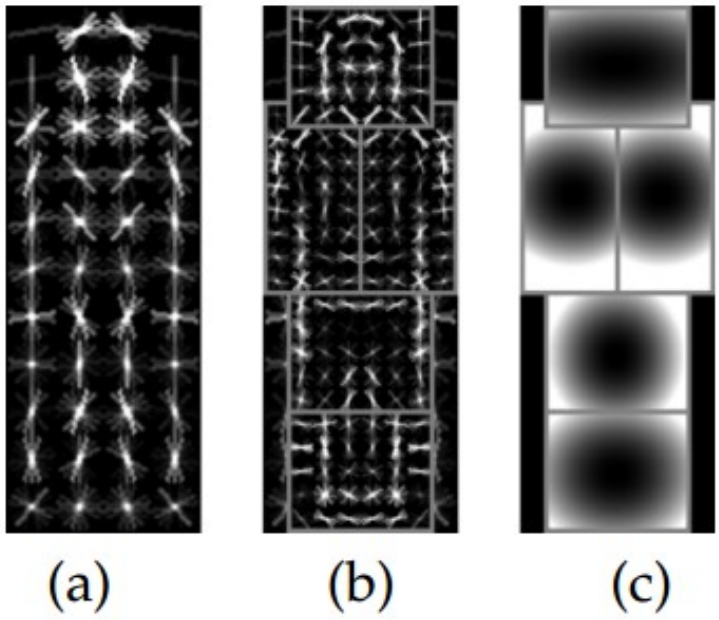


Рисунок 23. a) описание объекта целиком b) автоматические выведенные части объекта c) возможные смещения частей объекта.

В [38] был предложен подход, позволяющий избавиться от необходимости множественной классификации окон-кандидатов за счет использования глубокой нейронной сети, обученной предсказывать положение объектов по полному изображению. Авторы сформулировали задачу обнаружения объектов как задачу регрессии, в которой по входному изображению классификатор должен предсказать области изображения, в которых находится объект интереса (рисунок 24).

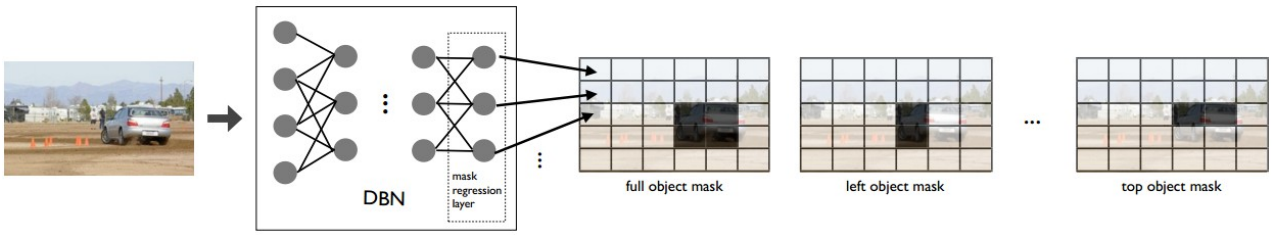


Рисунок 24. Схема работы метода [38]. Входное изображение подается на вход глубокой нейронной сети, которая предсказывает области изображения, в которых находится объект интереса.

С помощью данного подхода удалось добиться точности, сравнимой с точность работы лучших на данный момент методов на базе PASCAL VOC 2007 [39]. При этом он хорошо масштабируется по количеству классов объектов, так как позволяет осуществить этап обнаружения объектов независимо от этапа их последующей классификации. Например, можно обучить детектор всех интересующих классов объектов, а затем провести более точную классификацию только в тех областях, которые были помечены детектором, как содержащие объект. Вышеописанная стратегия используется и в данной работе. Разделение этапа выделения и классификации позволяет существенно снизить время работы всей системы.

Метод для измерения степени “объектности” области изображения был предложен в [40]. В отличии от [38], он основан на подходе “скользящего окна”, классификатор которого измеряет не вероятность принадлежности изображения определенному классу объекта, а вероятность того, что внутри окна находится какой-либо объект. В данной работе предлагается использовать низкоуровневые признаки, что позволяет надеяться на то, что классификатор будет чувствителен именно к понятию “объекта”, а не к конкретным классам объектов, представленным в обучающей выборке. В [40] используются следующие признаки:

1. уникальность объекта в рамках одного изображения, измеряемая с помощью спектральных остатков (spectral residual) [41]. Данный признак измеряется отдельно для каждого канала изображения
2. контрастность по цвету. Мера отличия по цвету области изображения от ее непосредственного окружения
3. количество краев около границы окна. Предполагается, что у окон с объектами этот показатель выше, чем у окон с фоном
4. размеры и положение окна. Априорные распределения на данные параметры считаются по обучающей выборке.

Подобные способы поиска окон-кандидатов для дальнейшей, более детальной, классификации набирают популярность с ростом размеров выборок и количества классов. Например, в конкурсе по локализации объектов ILSVRC 2014 [42] [43] количество классов равно 200, в отличии от 20 классов в популярном на данный момент конкурсе PASCAL VOC 2007 [39].

Еще одна модификация метода на основе скользящего окна предложена в [44]. Подход занимает промежуточную позицию между методами на основе скользящего окна и методами, предсказывающими положение объекта по целому изображению [38]. В [44]

предлагается использовать скользящие окна большого размера, перемещающиеся по изображению с большим шагом. Во время каждого приложения окна используется классификатор на основе сверточной нейронной сети, предсказывающий положение объекта внутри окна. Таким образом, задача сводится к задаче регрессии из окна изображения в четыре числа, характеризующих ограничивающий прямоугольник объекта внутри окна (рисунок 25). Использование больших окон, скользящих по изображению с большим шагом, позволяет существенно ускорить время работы детектора, а также позволяет детектору учитывать контекст.

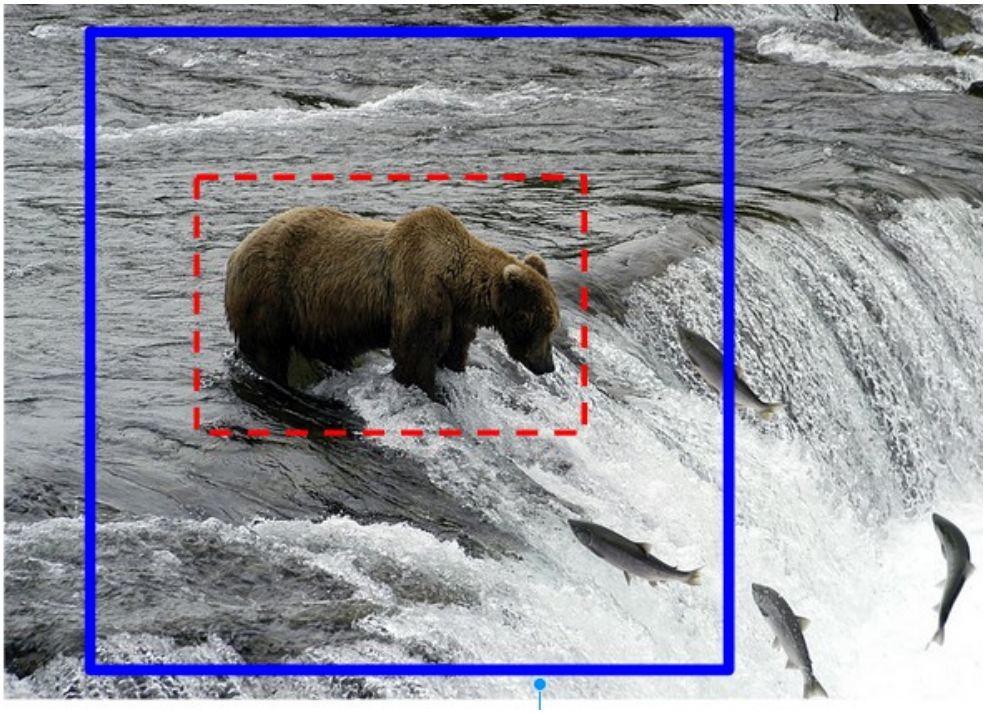


Рисунок 25. Пример работы метода [44]. Сплошной линией показано “скользящее окно” детектора. Пунктирной линией показан предсказанный ограничивающий прямоугольник объекта внутри данного окна.

2.2.2. Методы выделения знаков дорожного движения

Опубликован ряд работ, описывающих системы выделения знаков дорожного движения, нацеленных на применение на практике в больших объемах. В [45] предложена система, основанная на каскаде классификаторов, обученных с помощью алгоритма AdaBoost на разъединенных диполях (dissociated dipoles, рисунок 26) [46]. Система способна распознавать четыре типа знаков, сгруппированных по визуальной похожести (предупреждающие, знаки приоритета, запрещающие и предписывающие

знаки). При использовании монокулярной видеокамеры полнота обнаружения достигала 50-60% в зависимости от типа знака при частоте ложных срабатываний равной одному на 13-52 кадра. В отличии от описанного подхода, в данной работе на разных уровнях каскада предложено использовать классификаторы и признаки различного рода (включая цветовые признаки) . Это позволило значительно уменьшить как ошибку первого, так и ошибку второго рода. Обучение на искусственно созданных данных, предложенное в данной работе, позволяет уменьшить зависимость от размеров и количества классов в обучающей выборке.

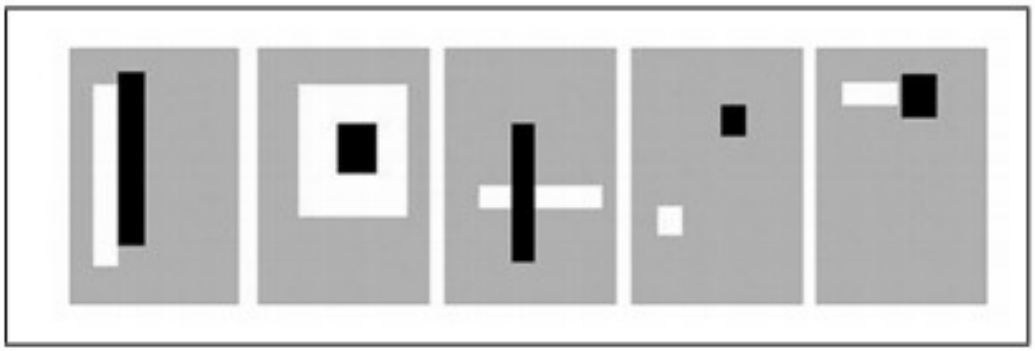


Рисунок 26. Пример возможных вариантов признаков разъединенных диполей.

Другая система обнаружения дорожных знаков описана в [47]. В отличии от данной работы, в качестве входных данных в предложенном алгоритме выступал набор изображений сцены, снятых с различных ракурсов с помощью восьми по-разному расположенных видеокамер. Система была обучена обнаруживать и распознавать 62 класса знаков. В режиме работы с видео только с одной камеры полнота обнаружения равнялась 96.8% знаков при 2 ложных срабатываниях на кадр. Большое количество ложных срабатываний затем снижалось за счет использования информации с разных ракурсов, что позволило достигнуть полноты обнаружения в 95.3% физических знаков с одним ложным срабатыванием на 6000 изображений. В отличии от описанной работы, предложенный в данной диссертации подход работает с входными данными в виде видеоряда с одной камеры и не требует установки специального оборудования, что упрощает его применение в практических задачах.

В [48] был использован “мягкий каскад” (soft cascade) с поканальными признаками (рисунок 27) и приведены результаты работы предложенного алгоритма на двух общедоступных базах немецких и бельгийских знаков, состоящих из 43 и 62 классов соответственно. Была достигнута точность в 99% площади под ROC кривой (area under

the curve, AUC) на немецкой базе и 92.56% AUC на бельгийской базе. К сожалению, доля ложных срабатываний была оценена на маленьком наборе изображений, состоящем из 300 примеров для немецкой и 583 примеров для бельгийской базы.

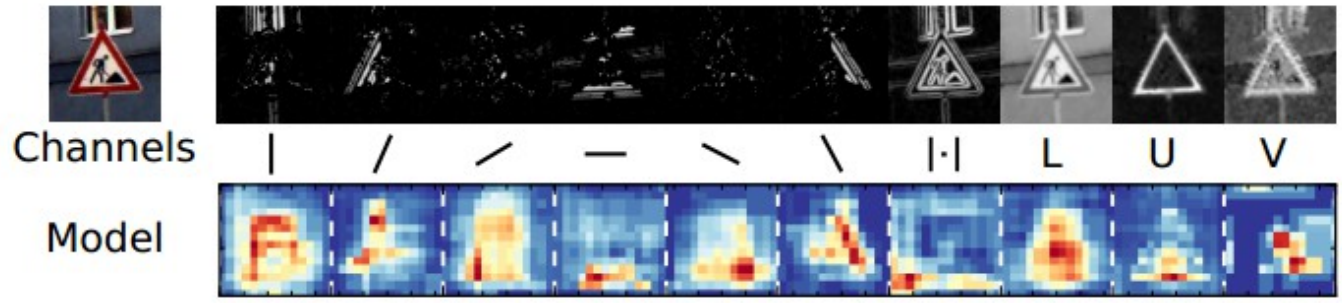


Рисунок 27. Визуализация откликов поканальных признаков, использованных в работе [48].

В работе [49] использование специализированной под аппаратную платформу реализации HOG-дескрипторов и простого, но быстрого, классификатора на основе линейного дискриминантного анализа (Fisher Discriminant Analysis, FDA) позволило создать систему с полнотой в 99% и одним ложным срабатыванием на 10^{10} окон детектора. Методы, предложенные в описанной работе, наиболее близки к алгоритмам, описанным в данной главе, но мы представляем результаты для более чем 140 различных классов дорожных знаков (в отличии от трех классов, представленных в [49]) и показываем, что некоторые классы знаков являются более сложными для обнаружения. Например, синие квадратные знаки (в основном это знаки особых предписаний) являются сложными для обнаружения из-за схожести по цвету с небом, а также из-за отсутствия легко различимой каемки.

2.3. База российских знаков дорожного движения

Для оценки точности предлагаемых алгоритмов совместно с коммерческим партнером была создана база российских знаков дорожного движения, названная Russian Traffic Signs Dataset (RTSD) [50]. Она состоит из 9508 изображений со знаками и 71050 изображений с фоном. Всего размечено 14360 ограничивающих прямоугольников знаков, 6387 из которых также помечены меткой физического знака. Всего размечено 863 физических знака. Таким образом, каждый физический знак в среднем встречается на 7.3 изображениях. Данные разделены на тренировочные и тестовые. Тренировочная

выборка состоит из 4754 изображений со знаками и 44817 изображений с фоном. Остальные изображения отнесены к тестовой выборке.

2.4. Предлагаемый алгоритм

При разработке алгоритма обнаружения объектов были заложены две цели:

1. он должен отрабатывать быстро, чтобы позволять за короткий срок обрабатывать большие объемы данных
2. он должен использовать наличие большого количества искусственно созданных данных.

Подход на основе каскада классификаторов [1] удовлетворяет вышеперечисленным требованиям и хорошо зарекомендовал себя в решении различных задач обнаружения. Он заключается в поэтапном обучении классификаторов. Каждый классификатор C_i должен удовлетворять требованиям на минимальную полноту $R_{min,i}$ и максимальную долю ложных срабатываний $F_{max,i}$. Классификаторы увеличивают $R_{min,i}$ и уменьшают $F_{max,i}$ за счет усложнения модели, а значит - за счет увеличения требования на количество вычислительных ресурсов. На первых этапах каскада требования на $F_{max,i}$ невысокие, что позволяет обучать неточные, но быстрые классификаторы, отбрасывающие большую часть необъектных окон-кандидатов. По мере усложнения решаемой задачи усложняется и структура классификатора (например, он начинает использовать больше признаков). Такой подход позволяет быстро отбросить основную часть окон-кандидатов и сконцентрировать на анализе сложных случаев.

В ходе экспериментов стало ясно, что для решения задачи обнаружения множества различных классов объектов недостаточно иметь однородную структуру каскада, состоящего из классификаторов и признаков одного типа. Гораздо эффективнее дать алгоритму обучения возможность выбирать среди множества классификаторов и множества признаков различной сложности. При этом на ранних этапах каскада целесообразно дать классификатору возможность использовать только быстрые классификаторы и быстрые признаки. В ходе данной работы был разработан программный комплекс для обучения классификаторов, позволяющий задавать разный набор возможных классификаторов и возможных признаков для разных этапов каскада.

На первых этапах в качестве признаков использовались разделенные диполи (dissociated dipoles) [51], хорошо зарекомендовавшие себя в других задачах распознавания. Стоит отметить, что широко известные признаки Хаара [1] являются подмножеством разделенных диполей, если дать возможность различным парам

диполей комбинировать выходы (что и происходит в рамках алгоритма обучения AdaBoost). Пространство перебора в случае разделенных диполей намного больше, чем в случае признаков Хаара, это позволяет алгоритму AdaBoost находить более полезные признаки, что в итоге сказывается на скорости работы классификатора. Таким образом, за счет увеличения времени обучения удается получить выигрыш в точности во время классификации. В качестве классификатора на первых этапах каскада, по-аналогии с [1], был выбран алгоритм AdaBoost [36]. AdaBoost является жадным алгоритмом, что позволяет постепенно добавлять новые признаки (усложняющие классификатор). Как и в [1], при переходе к обучению следующего этапа каскада в качестве негативных примеров используются необъектные окна, дающие наибольший отклик при применении уже обученной части каскада (в англоязычной литературе данный метод известен под названием bootstrapping). Такой подход позволяет классификатору следующего этапа каскада концентрироваться только на тех примерах, которые являются сложными для части каскада, отрабатывающей до него.

Использование быстрых признаков и быстрого классификатора позволяет довести долю ложных срабатываний до определенного уровня, после чего при незначительном уменьшении доли ложных срабатываний начинается резкое падение полноты. Чтобы получить хорошую полноту на выходе каскада и при этом иметь низкую долю ложных срабатываний, мы провели ряд экспериментов по замене простых и быстрых классификаторов на более точные и медленные. Медленные классификаторы отрабатывают лишь на небольшой части окон-кандидатов и практически не влияют на общее время работы системы**.**

2.4.1. Сверточные нейронные сети для классификации изображений

Сверточные нейронные сети [52] показали хорошие результаты на задачах распознавания [53] и являются на сегодняшний день наиболее точным классификатором изображений. Это делает целесообразным их использование в качестве точного, но медленного классификатора на последних этапах обнаруживающего каскада.

Сверточная нейронная сеть состоит из повторяющихся блоков, каждый из которых состоит из операции свертки, сабсемплинга и оператора нелинейности(рисунок 28). Реальные системы состоят из дополнительных слоев, таких как локальная нормализация отклика (local response normalization) [53] или слоев регуляризации, таких как dropout [54].

Сверточные слои характеризуются шириной w_c и высотой фильтра h_c свертки, а также количество фильтров n_f . Еще одним неявным гиперпараметром является

количество каналов в фильтре, оно обычно берется равным количеству каналов входной матрицы (на первом слое входная матрица предствляет собой многоканальное изображение, на всех последующих - трехмерный тензор). Работа сверточного слоя заключается в многократном применении операции свертки с разным ядром к одной и той же входной матрице. Результатом свертки также является трехмерный тензор, обычно называемый картой откликов (features map). При обучении изменяются параметры фильтров в сверточных слоях, что позволяет получить детекторы различных составных частей объектов, полезных при распознавании. На рисунке 29 показаны фильтры, которые удается автоматически обучить на первом сверточном слое, они представляют собой простые детекторы краев, некоторые из них напоминают фильтры Габора.

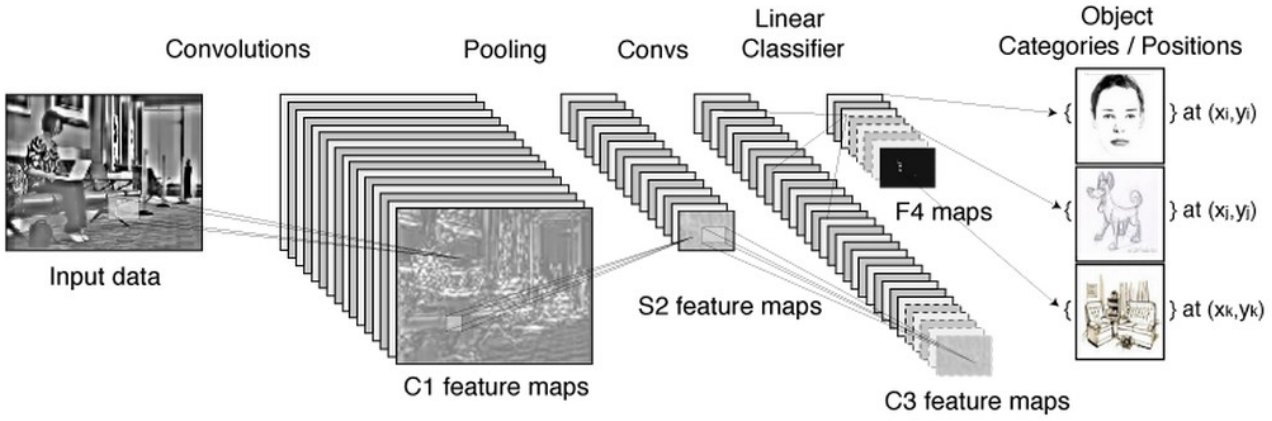


Рисунок 28. Архитектура сверточной нейронной сети. Входное изображение преобразуется с помощью множества операций свертки, извлекающих различные признаки. Затем полученные карты откликов уменьшаются с помощью операции сабсемплинга, что позволяет достигнуть инвариантности к локальным трансформациям. К выходам слоев сабсемплинга применяется функция нелинейности (например ReLU). Данные три шага повторяются несколько раз.

За сверточным слоем обычно размещают слой *сабсемплинга*, характеризуемый шириной w_s и высотой h_s. Самый простой вариант данного слоя (называемый average pooling) представляет собой поканальную свертку с box-фильтром, каждый элемент f_i которого получается по следующей формуле: $f_i = \frac{1}{h_s \cdot w_s}$.

Также часто применяется max-pooling, принцип работы которого заключается в поиске максимального значения внутри рассматриваемого окна. При этом, как и в случае average-pooling, данный фильтр перемещается по изображению аналогично сверточному фильтру.

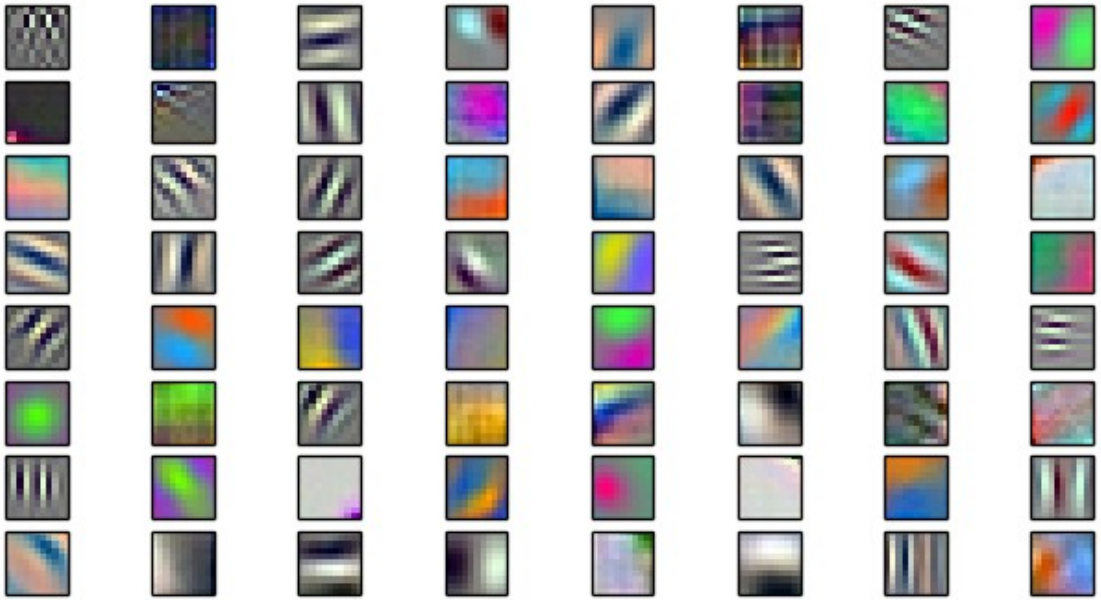


Рисунок 29. Визуализация фильтров, получаемых на первом сверточном слое нейронной сети, обучаемой классифицировать изображения.

После слоя сабсемплинга помещается слой *нелинейного преобразования*, позволяющий строить с помощью нейронной сети сложные нелинейные модели. Вариантов нелинейного преобразования может быть множество. В последнее время часто применяется так называемая ReLU-нелинейность (рисунок 30), которая записывается с помощью следующего уравнения $y = max(0, x)$.

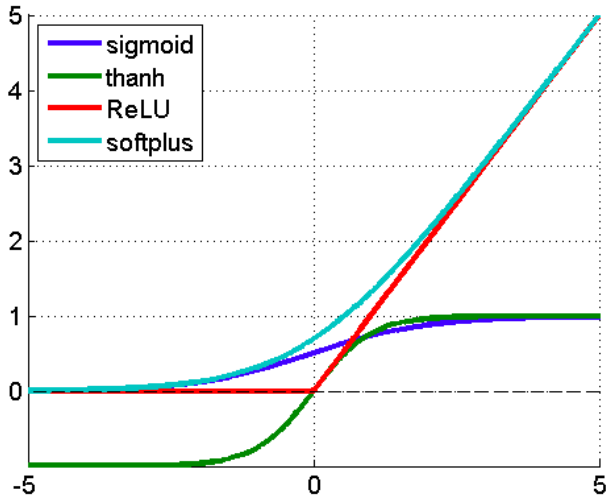


Рисунок 30. Популярные в нейронных сетях функции нелинейного преобразования.

Последним в сети размещается полносвязный слой, который можно рассматривать как частный случай сверточного слоя, с ядром фильтра, у которого $w_c = h_c = 1$, а количество выходных нейронов равно количеству обучаемых фильтров n_f .

В статье [53], выигравшей конкурс LSVRC-2012 [43], был использован слой *локальной нормализации отклика*, увеличивающий отклик тех фильтров, чей выход является максимальный в локальной окрестности. Формально описанное преобразование можно записать следующим образом:

$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=max(0,i-n/2)}^{max(N-1,i+n/2)} \left(a^j_{x,y}\right)^2\right)^\beta$, где

N - общее количество фильтров в ядре,

k, n, α, β - гиперпараметры алгоритма.

Помимо этого на двух последних полносвязных слоях в [53] предложили использовать метод регуляризации dropout [54], заключающийся в обнулении входов с заданной вероятностью p во время обучения сети. Во время тестирования dropout-слой умножает входы на коэффициент $1-p$. Было показано, что в случае однослойной сети данная процедура идентична обучению экспоненциального количества (по количеству нейронов в dropout-слоях в степени экспоненты) разных сетей, с последующим их усреднением с помощью геометрического среднего. В случае многослойных сетей такой способ является хорошей аппроксимацией [55]. Выигравшая конкурс LSVRC-2012 архитектура показана на рисунке 31.

В нашем случае на вход сети поступает область трехканального изображения, размером в 30x30 пикселей. Входное изображение нормализуется с помощью метода глобального выравнивания гистограммы. Архитектура сети состоит из 5 слоев с весами и 2 слоев max-сабсемплинга. Первые два слоя - сверточные, с 64 ядрами свертки размером 5x5 пикселей и шагом свертки в один пиксель. За каждым сверточным слоем следует слой max-сабсемплинга с размером ядра в 3x3 пиксела и шагом в 2 пиксела. За вторым сверточным слоем следуют два локально-связных слоя с размером ядра 3x3 и шагом 1. Последний слой - полносвязный и его выходы поступают на вход в softmax-слой с двумя нейронами, который выдает на выходе распределение вероятности принадлежности изображения к одному из двух классов. Гиперпараметры сети были подобраны с помощью кросс-валидации.

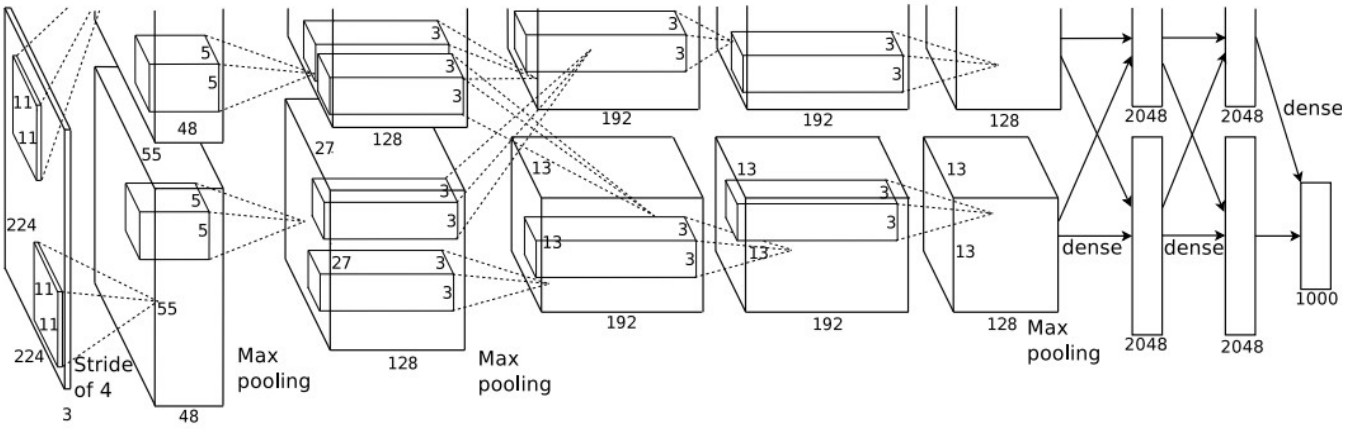


Рисунок 31. Архитектура сверточной нейронной сети, выигравшей конкурс LSVRC 2012 с большим отрывом.

2.4.2. Обучение модуля обнаружения

Мы выделили четыре группы знаков и объединили их по близости по цвету и форме (*красные треугольники*, *красные круги*, *синие круги*, *синие квадраты*, рисунок 32). Далее каждая такая группа знаков именуется *типом* знака. Для примера на рисунке 33 приведены пиктограммы *классов* знаков, попавших в *тип* красные круги. Такое разбиение существенно упрощает задачу детектора. Наши эксперименты показали, что обучить детектор на основе диполей возможно только в случае подобного разбиения. В противном случае выборка оказывается слишком сложной и классификатору не удается подобрать хорошие признаки (общие для всех классов в обучающей выборке), что приводит к плохой полноте детектора.

Рисунок 32. *Типы* знаков, на которых был обучен детектор.

Рисунок 33. Пример пиктограмм всех *классов* знаков, принадлежащих одному *типу* знаков *красный круг*.

Каждый из выделенных типов знаков обучался отдельно. Сначала мы обучали этапы каскада на основе диполей и AdaBoost классификатора. В качестве положительных примеров мы создавали 10000 искусственных примеров знаков, а в качестве отрицательных брали 16000 случайных окон фона. Примеры фона полностью обновлялись после каждой итерации бутстрэппинга, то есть перед каждым запуском обучения очередного этапа каскада. Мы продолжали итерации *бутстрэппинга* (bootstrapping) до тех пор, пока доля ложных срабатываний не опускалась ниже отметки 10^{-7}, дальнейшее обучение каскада на основе AdaBoost на дипольных признаках

приводило к сильной деградации полноты детектора. Бутстрэппинг проводился на вручную размеченной коллекции изображений, не содержащих знаков дорожного движения. Сбор подобной коллекции является более простой задачей по сравнению с разметкой самих знаков, так как нет необходимости выделять знаки ограничивающим прямоугольником и приписывать каждому выделенному знаку его класс. На рисунках 36 и 37 приведены графики зависимости полноты от доли ложных срабатываний для типов знаков *синие квадраты* и *красные треугольники***.**

После обучения каскада на основе диполей мы продолжали обучение на основе сверточной нейронной сети, которая являлась финальным этапом каскада. Обучение проходило в несколько этапов, чередующихся с бутстрэппингом. Использование быстрой реализации сверточной нейронной сети на графическом сопроцессоре (graphical processing unit, GPU) [56] позволило значительно увеличить обучающую выборку синтетических примеров знаков до 200000 изображений. Количество примеров фона, как и раньше, было равно 16000 и дополнялось после каждой итерации бутстрэппинга.

2.5. Экспериментальная оценка

Мы провели ряд экспериментов, проверяющих модификации, призванные улучшить точность классификации и скорость работы алгоритма.

2.5.1. Использование цветовых признаков

В данном эксперименте было обучено два каскада детекторов для типа знаков *синие квадраты*. Первый каскад был обучен на признаках, извлекаемых из изображения, состоящего из оттенков серого (grayscale), второй - на цветных признаках, подчеркивающих синий цвет [57]:

$f_b(x) = max(0, min(\frac{x_b - x_r}{x_r + x_g + x_b}, \frac{x_b - x_g}{x_r + x_g + x_b}))$, где

x_r, x_g, x_b - значения пикселя x в трех каналах RGB-изображения.

Рисунки 34 и 35 демонстрируют превосходство цветных признаков по критериям точности и количества необходимых признаков. Из рисунка 34 видно, что полнота каскада, использующего только оттенки серого начинает сильно деградировать после достижения определенного значения точности. Из рисунка 35 видно, что каскаду на основе цветных признаков для достижения того же соотношения полноты и точности требуется извлечь в 7 раз меньше признаков. Это означает, что получаемые в результате каскады классификаторов будут работать быстрее и точнее.

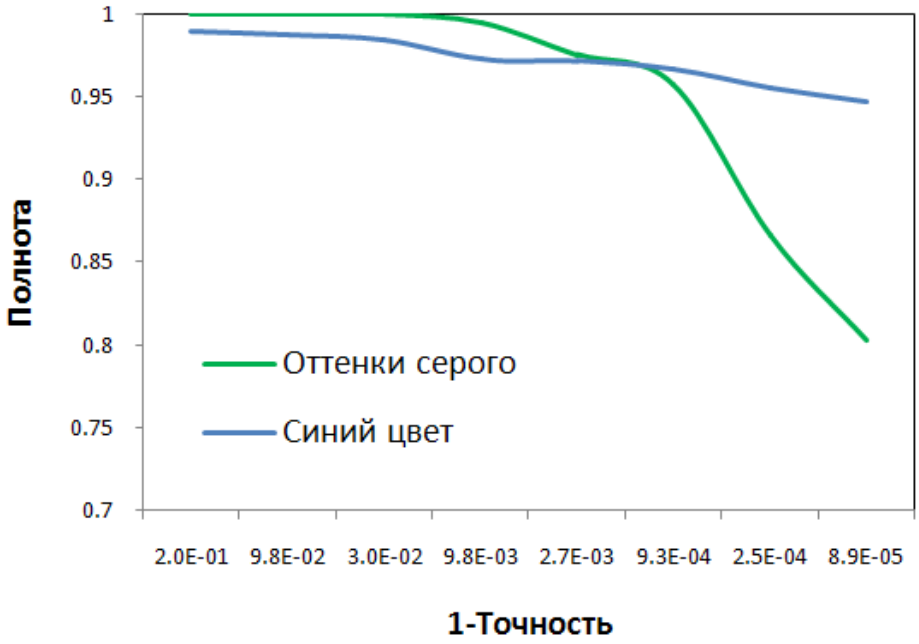


Рисунок 34. Соотношение полноты и точности для случаев использования цветных признаков и признаков на основе оттенков серого.

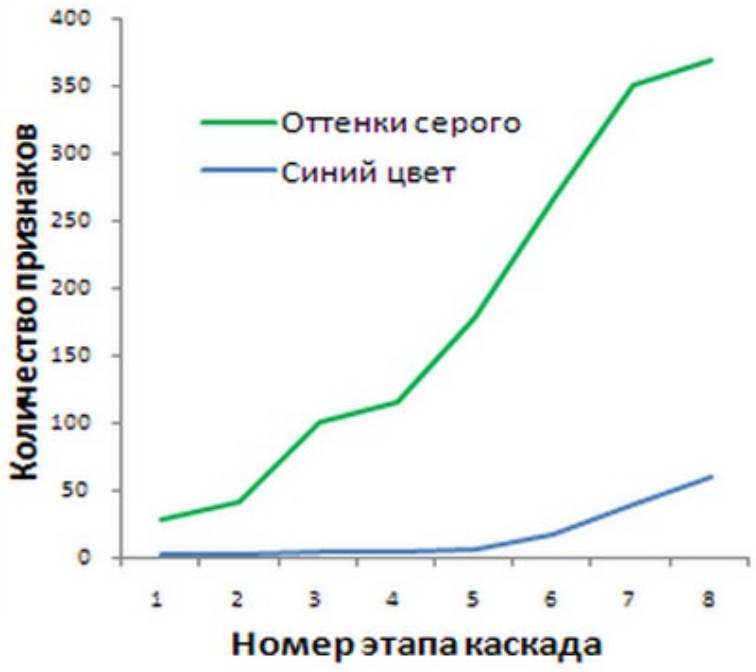


Рисунок 35. Зависимость количества необходимых признаков от этапа каскада для случаев использования цветных признаков и признаков на основе оттенков серого.

2.5.2. Сверточная нейронная сеть на последнем этапе каскада

Разделенные диполи являются быстрыми признаками. В нашем случае они используются на первых этапах каскада, чтобы увеличить скорость его работы. Практически все время работы классификатора сосредоточено на этих этапах, так как на них отвергается более 99% окон-кандидатов. Но наши эксперименты показали, что точность работы классификатора, обученного на диполях, начинает резко деградировать после точки, в которой доля ложных срабатываний на окно детектора опускается ниже отметки в 10^{-7} (рисунки 36 и 37). Возможное решение этой проблемы состоит в использовании более сложных (но медленных) признаков. Мы провели эксперименты с двумя возможными вариантами. Первый заключается в обучении нескольких заключительных этапов каскада с помощью алгоритма AdaBoost на признаках HOG с линейным SVM в качестве базового классификатора. Второй - в обучении глубокой сверточной нейронной сети с несколькими итерациями бутстрэппинга. Из рисунков 36 и 37 видно, что использование HOG-признаков не увеличивает точности классификатора. В то же время, использование сверточной нейронной сети позволяет достичь того же уровня ложных срабатываний с полнотой в среднем выше на 7%. Итоговые соотношения точности и полноты детектора для четырех типов знаков приведены в таблице 8.

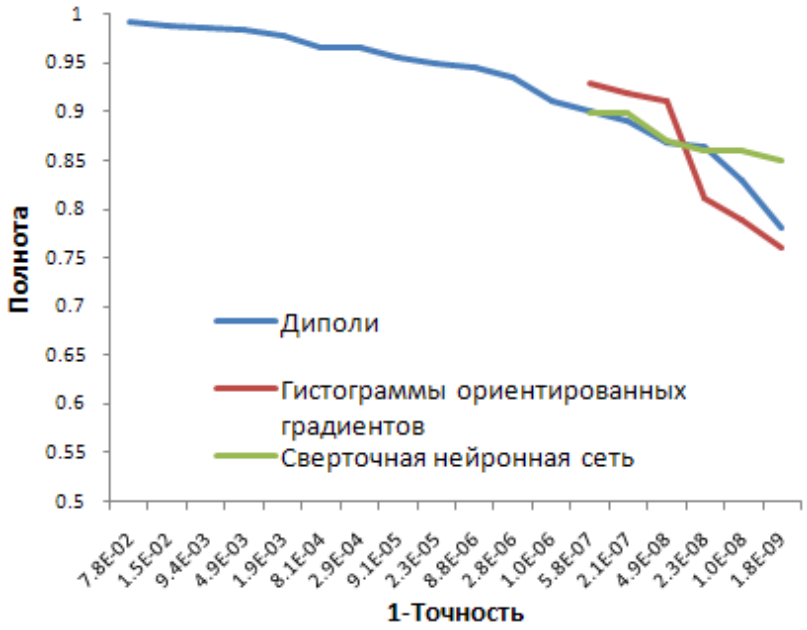


Рисунок 36. Соотношение полноты и точности при обучении каскада классификаторов типа знаков *синие квадраты* на основе различных признаков.

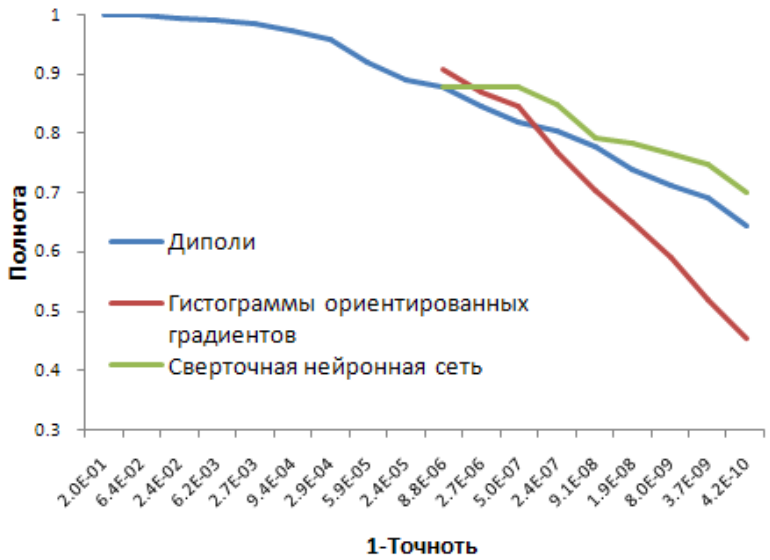


Рисунок 37. Соотношение полноты и точности при обучении каскада классификаторов типа знаков *красные треугольники* на основе различных признаков.

Тип знака	Доля ложных срабатываний	Полнота (по изображениям)	Полнота (по физическим знакам)
Синие квадраты	$7 \cdot 10^{-10}$	77%	92.18%
Красные треугольники	$7 \cdot 10^{-10}$	73.1%	82.35%
Синие круги	$6 \cdot 10^{-10}$	79.2%	83%
Красные круги	$2 \cdot 10^{-9}$	73.8%	84.7%

Таблица 8. Точность работы каскада детектора на четырех типах знаков.

2.5.3. Сравнение с обучением на реальных знаках

В данном эксперименте проведено сравнение соотношения полноты и точности детектора, обученного на искусственных данных, с детектором, обученным на реальных данных. На рисунке 38 видно, что детектор, обученный на реальных данных показывает лучшее соотношение полнота/точность. Но, если сравнивать полноту по физическим знакам, то разница будет относительно небольшой - 96.88% против 92.18%. С практической точки зрения перспективным представляется гибридный подход - обучение на смести реальных и синтетических данных. Он позволит значительно сократить затраты на разметку новых примеров (особенно для редких типов знаков), потенциально сохранив при этом высокую точность и полноту обнаружения.

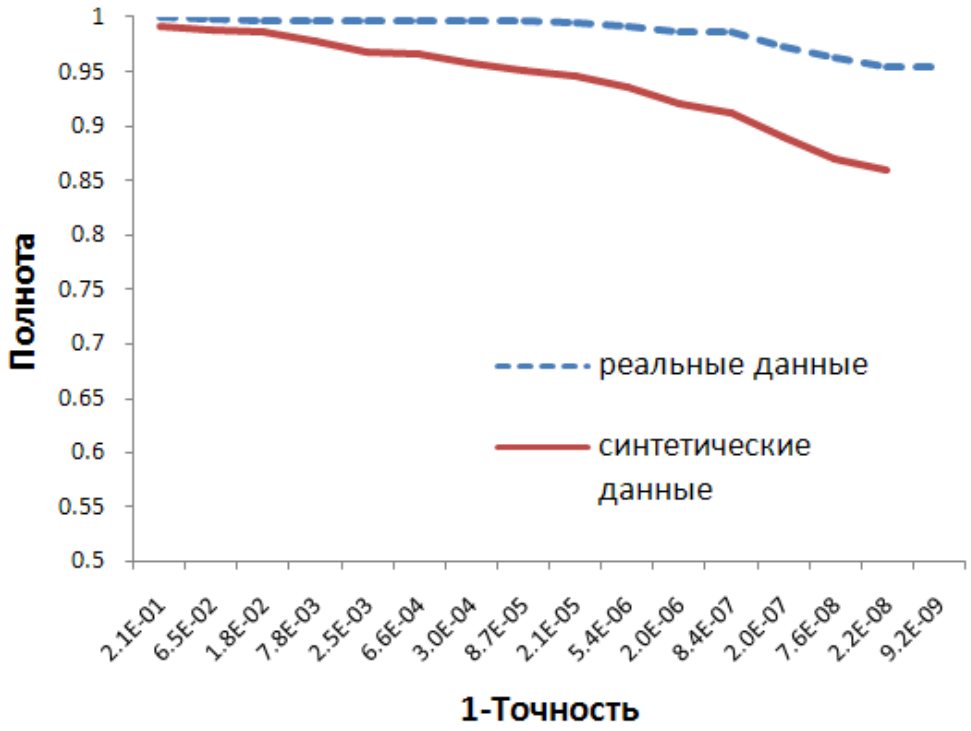


Рисунок 38. Соотношение полноты и точности работы детектора при обучении на реальных и на синтетических данных.

2.6. Заключение

В данной главе была описана и формально поставлена задача выделения объектов. Проведен обзор подходов к обнаружению объектов любого рода. Отдельно были рассмотрены подходы к выделению знаков дорожного движения. Предложен алгоритм выделения объектов, сочетающий в себе высокую скорость и точность работы. Проведены эксперименты, подтверждающие возможность обучения предложенного алгоритма на искусственно созданных данных. Описаны эксперименты, подтверждающие основные дизайнерские решения, принятые при создании алгоритма, такие как:

1. обучение на цветовых признаках
2. использование глубокой сверточной нейронной сети на последних этапах каскада.

Также проведено сравнение с обучением детектора той же архитектуры на реальных данных, которое показало, что хотя детектор, обученный на искусственных данных, работает хуже, его точности достаточно для решения большого количества практических задач. Предложен перспективный гибридный вариант обучения на выборке, состоящей из смести реальных и синтетических данных. Данный подход может позволить добиться хорошей точности при обнаружении часто встречающихся классов

знаков (для которых относительно легко собрать коллекцию реальных примеров) и приемлемой точности для редких классов.

Глава 3. Классификация объектов

3.1. Постановка задачи

Предполагая, что объект интереса занимает большую часть изображения, необходимо определить принадлежность объекта к одному из заранее заданных классов. Визуализация задачи для случая классификации знаков дорожного движения приведена на рисунке 39.

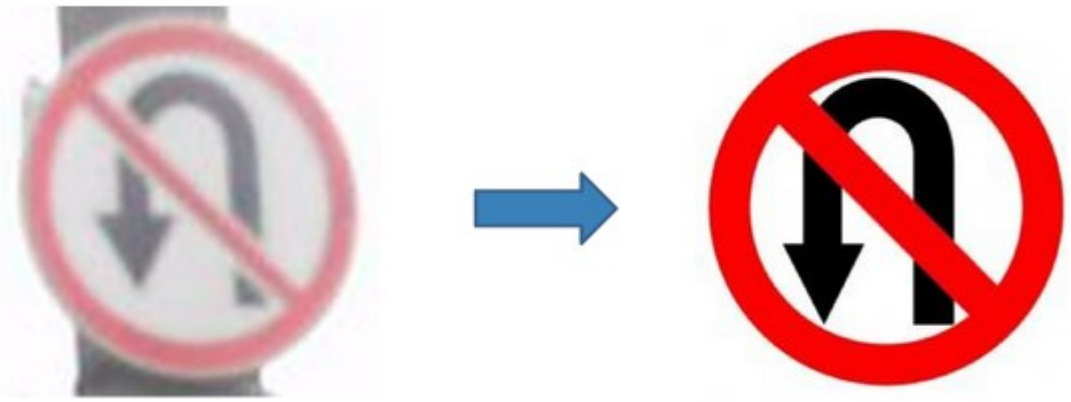


Рисунок 39. Визуализация задачи классификации объектов.

3.2. Обзор существующих методов

В разделе 3.2.1. проведен обзор методов классификации объектов любой природы, не использующих знания из предметной области. В разделе 3.2.2. рассмотрены методы, специфичные для задачи классификации знаков дорожного движения.

3.2.1. Методы классификации объектов

Одним из наиболее популярных подходов к классификации объектов является подход на основе гистограмм ориентированных градиентов (HOG), классифицируемых с помощью метода опорных векторов (SVM). Данный подход, а также его развитие на основе деформируемой модели частей [37] уже описаны в разделе 2.2.1. в контексте задачи выделения объектов.

Другим популярным подходом к классификации изображений является подход, основанный на мешке визуальных слов [58]. Идея метода состоит в представлении изображения с помощью гистограммы визуальных слов, аналогично одноименному подходу, применяемому для представления текстов (рисунок 40). Для получения

визуальных слов из изображений извлекаются интересные точки (interest points), обычно представляющие собой области с высокой нормой градиента по обоим направлениям (уголки). Наиболее популярен подход на основе DoG-детектора и SIFT-дескриптора [59]. Полученные на основе SIFT представления интересных точек кластеризуются. Центры полученных кластеров и называются визуальными словами. Чтобы описать новое изображение из него извлекаются интересные точки и для каждой из них выполняется поиск ближайшего визуального слова. Затем изображение представляется в виде гистограммы визуальных слов (рисунок 41). После этого на полученных представлениях можно обучать классификаторы, решающие различные задачи. Данный подход применяется как для классификации изображений целиком, так и для обнаружения отдельных объектов на изображениях (например, с помощью метода скользящего окна).

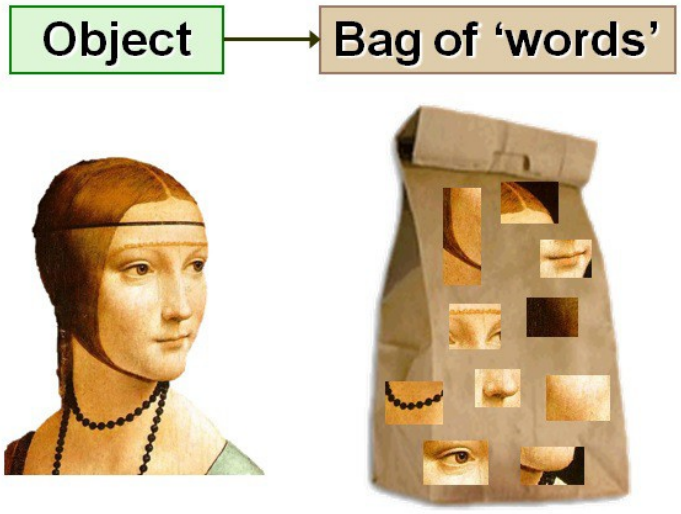


Рисунок 40. Концептуальная визуализация метода на основе мешка визуальных слов. Изображение представляется набором своих частей, при этом теряется информация об их пространственном расположении друг относительно друга.

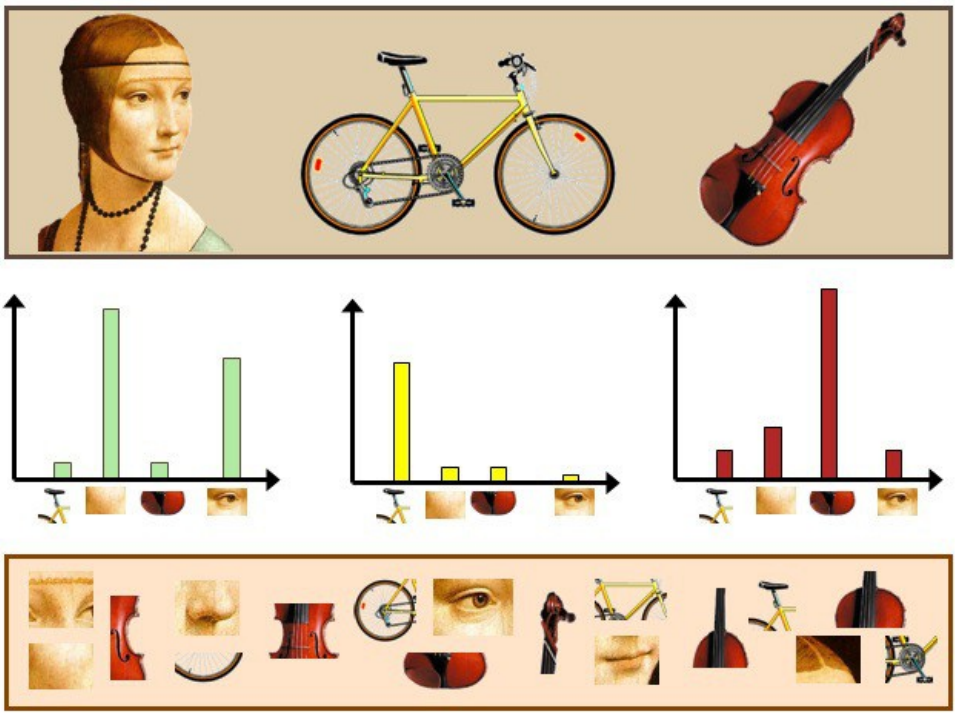


Рисунок 41. Представление изображений (сверху) с помощью обученного словаря визуальных слов (снизу). Каждое изображение описывается с помощью гистограммы визуальных слов (посередине).

Наиболее успешным на данный момент методом классификации изображений является метод на основе глубоких сверточных нейронных сетей, частично описанный в разделе 2.3.1. Сверточные нейронные сети позволяют обучать представления изображений, что является одним из их главных преимуществ перед методами, основанными на придуманных человеком признаках. Помимо этого, реализации на современных сопроцессорах позволяют обучать модели с больших числом параметров на больших выборках. Современные исследования показали зависимость точности работы метода от размеров обучающей выборки при решении многих задач [60],[61],[62]. В работе [53] был описан способ обучения глубокой нейронной сети, предсказывающей принадлежность изображения к одному из 1000 классов (конкурс ImageNet LSVRC 2012). При этом предложенный подход позволил обойти конкурентов с большим отрывом (в два раза, если измерять отрыв по ошибке классификации). Позднее было показано, что признаки, полученные с помощью обучения нейронной сети на задаче ImageNet, можно использовать для решения других задач (отличных от классификации объектов из ImageNet). Получаемые результаты сравнимы, а во многих случаях и превосходят, результаты современных лидеров [62],[63]. Недавно было показано, что использование

предобученных на ImageNet признаков также позволяет значительно увеличить точность работы детектора объектов [61].

3.2.2. Методы классификации знаков дорожного движения

Знаки дорожного движения созданы, чтобы быть хорошо заметными и понятными человеку. В то же время существует ряд трудностей, мешающих их автоматической классификации. Некоторые из них изображены на рисунке 42.

Рисунок 42. Сложности, возникающие при распознавании знаков дорожного движения. Значение строк в визуализации сверху вниз: внутриклассовая изменчивость, межклассовая похожесть, перекрытия, изменение освещения, размытие.

В соответствии с классификацией [64] методы распознавания знаков дорожного движения можно разделить на два типа - основанные на метрике похожести и на признаках.

В методах, основанных на признаках, входному изображению знака присваивается класс ближайшего примера из обучающей выборки в соответствии с заданной метрикой сравнения. В [57] предложен новый вариант метода AdaBoost, названный SimBoost (similarity boosting). SimBoost обучается таким образом, чтобы приписывать большое

значение уверенности паре изображений, содержащих знак одного и того же класса. В режиме тестирования производился поиск ближайшего соседа к распознаваемому изображению на основе инвертированной степени уверенности, возвращаемой SimBoost.

Гибрид классификаторов на основе случайного леса и ближайшего соседа на HOG-дескрипторах предложен в [33]. Подход из [33] на основе ближайшего соседа похож на подход, предлагаемый в данной диссертации, но мы достигаем лучших результатов за счет использования большого количества синтетически созданных примеров.

Еще одной популярной метрикой похожести является нормализованная кросс-корреляция, инвариантная к изменениям освещенности. Она была использована в [65] и [66]. Недостатком данной метрики является ее чувствительность к изменениям фона.

В подходах, основанных на признаках, изображения знаков представляются с помощью какого-либо дескриптора и на данных представлениях обучается классификатор, предсказывающий класс знака. [67] применяли двухэтапный подход к классификации. На первом этапе знак относился к одной из одиннадцати групп на основе цвета и формы (рисунок 43). Затем знак сегментировался от фона на основе фиксированной маски сегментации, заранее заданной для каждой из выделенных групп. Необработанные пиксели, лежащие внутри маски знака, подавались на вход SVM-классификатора с гауссовым ядром. В предлагаемом в данной диссертации подходе маска знака может быть получена точнее за счет использования большого набора возможных вариаций знака.

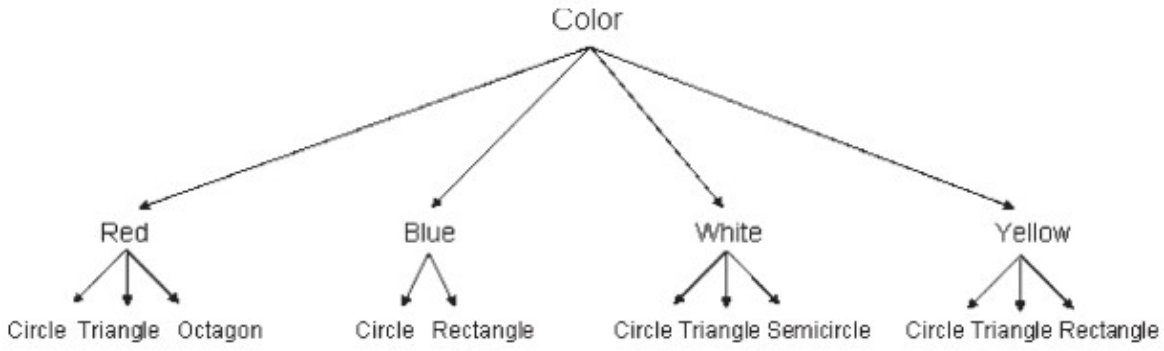


Рисунок 43. Дерево классификации знаков на основе цвета и формы.

Двухэтапная схема классификации была также использована в [68]. На первом этапе SVM-классификатор относил входной знак к одной из шести групп. Затем применялся второй SVM-классификатор (специфичный для каждой группы), обученный

на нормализованных RGB-значениях пикселей изображения, выдающий финальный класс знака. В [69] для получения базиса из 25 наиболее значимым векторов был применен линейный дискриминантный анализ (LDA). Затем функция плотности вероятности для каждого знака была аппроксимирована с помощью гауссианы. Решение о классификации производилось на основе поиска максимума правдоподобия.

Нейронные сети также часто используются в качестве классификатора в статьях по распознаванию знаков дорожного движения. В работе [31] многослойный персептрон (multilayer perceptron, MLP) и сверточная нейронная сеть (convolutional neural network, CNN) были использованы в комитете классификаторов. MLP был натренирован на HOG-признаках, в то время как CNN на случайно отмасштабированных, смещенных и повернутых цветных изображениях, предобработанных с помощью метода CLAHE (Contrast-Limited Adaptive Histogram Equalization).

В [32] также обучали сверточную нейронную сеть на изображениях, преобразованных в цветовое пространство YUV, с Y каналом отдельно предобработанным с помощью глобального и локального метода нормализации яркости. Другой особенностью работы является использование признаков, получаемых на ранних слоях сети, в качестве входа для поздних слоев (рисунок 44). Такой подход позволяет поздним слоям сети получать информацию о низкоуровневых деталях изображения (что актуально в случае классификации похожих классов знаков).

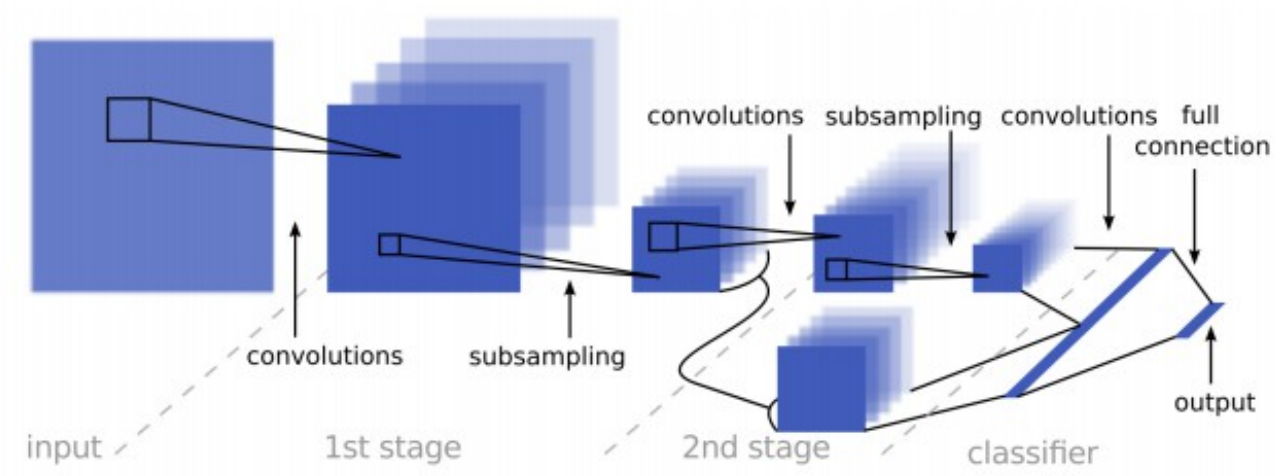


Рисунок 44. Архитектура сверточной нейронной сети, предложенной в [32]. Ее особенность заключается в том, что признаки, полученные после первого применения слоя сабсемплинга попадают на вход финальному слою сети, что позволяет учитывать низкоуровневые детали изображения.

3.3. Алгоритм сегментации объекта от фона

В практической системе модуль классификации неразрывно связан с модулем обнаружения. На их стыке возникают трудности, которые обычно не рассматриваются в публикуемых работах в связи с тем, что задачи обнаружения и классификации рассматриваются отдельно и предполагается, что входные данные подготовлены заранее. Одной из таких трудностей является неоднозначность предсказаний положения объекта, получаемых из модуля обнаружения. На рисунке 45 приведен пример вышеупомянутой неоднозначности.

Рисунок 45. Ограничивающие прямоугольники объекта, получаемые на выходе модуля обнаружения.

Стандартным подходом к решению данной проблемы является метод подавления не максимумов (non-maximal suppression), оставляющий только те ограничивающие прямоугольники, отклик детектора в которых является локальным максимумом в некоторой окрестности. Данный подход позволяет оставить одну гипотезы о положении объекта и за счет этого снизить нагрузку на модуль распознавания. В то же время нет никаких гарантий, что гипотеза наиболее точно представляет объект интереса и что объект центрирован в выбранном прямоугольнике.

Таким образом, ограничивающий прямоугольник объекта, поступающий на вход модулю распознавания, может содержать значительно смещенный относительно центра объект, что накладывает дополнительные требования на используемый метод классификации - локальную инвариантность к смещениям объекта интереса. В данном

разделе предлагается разбить решение вышеописанной проблемы на два этапа - этап сегментации знака от фона и этап классификации. Первый этап призван упростить задачу второго за счет центрирования объекта интереса и сегментации его от фона. Фоновые пиксели часто является одной из главных причин неверной классификации, особенно в случае, если фоновые пиксели занимают значительную часть изображения.

Предлагаемый метод базируется на предположении, что мы можем создать большую выборку изображений объектов интереса, подвергнутых разнообразным трансформациям. Данное предположение важно потому, что мы хотим отделить объект от фона с пиксельной точностью, а для этого необходимо иметь возможность получить точную оценку параметров трансформаций объекта. Метод сегментации состоит из следующих шагов (рисунок 46):

1. создается большая выборка искусственных примеров объектов, покрывающая трансформации объекта, которые влияют на положение маски объекта (отделяющей его от фона). К примеру, такими трансформациями являются поворот, масштабирование и смещение объекта. После создания искусственной выборки для каждого входящего в неё изображения мы знаем параметры трансформаций, а следовательно и маску фона
2. каждое изображение из синтетической выборки описывается с помощью одного из дескрипторов изображений. Важно, чтобы дескриптор ***не*** обладал сильной инвариантностью к трансформациям. Это необходимо для получения точной маски фона. В случае знаков дорожного движения был использован HOG-дескриптор с небольшим размером ячейки и небольшим размером блока
3. на полученных дескрипторах строится поисковый индекс, позволяющий находить похожие изображения (ближайших соседей по некоторой метрике расстояния)
4. входное изображение представляется с помощью того же дескриптора, что используется в пункте 2
5. осуществляется поиск входного изображения в индексе
6. маски фона найденных ближайших соседей комбинируются для получения маски фона входного изображения. В самом простом варианте выбирается маска ближайшего соседа
7. полученная маска фона накладывается на исходное изображение. Все пиксели фона окрашиваются в один, заранее выбранный, цвет. Полученное изображение без фона подается на вход классификатору.

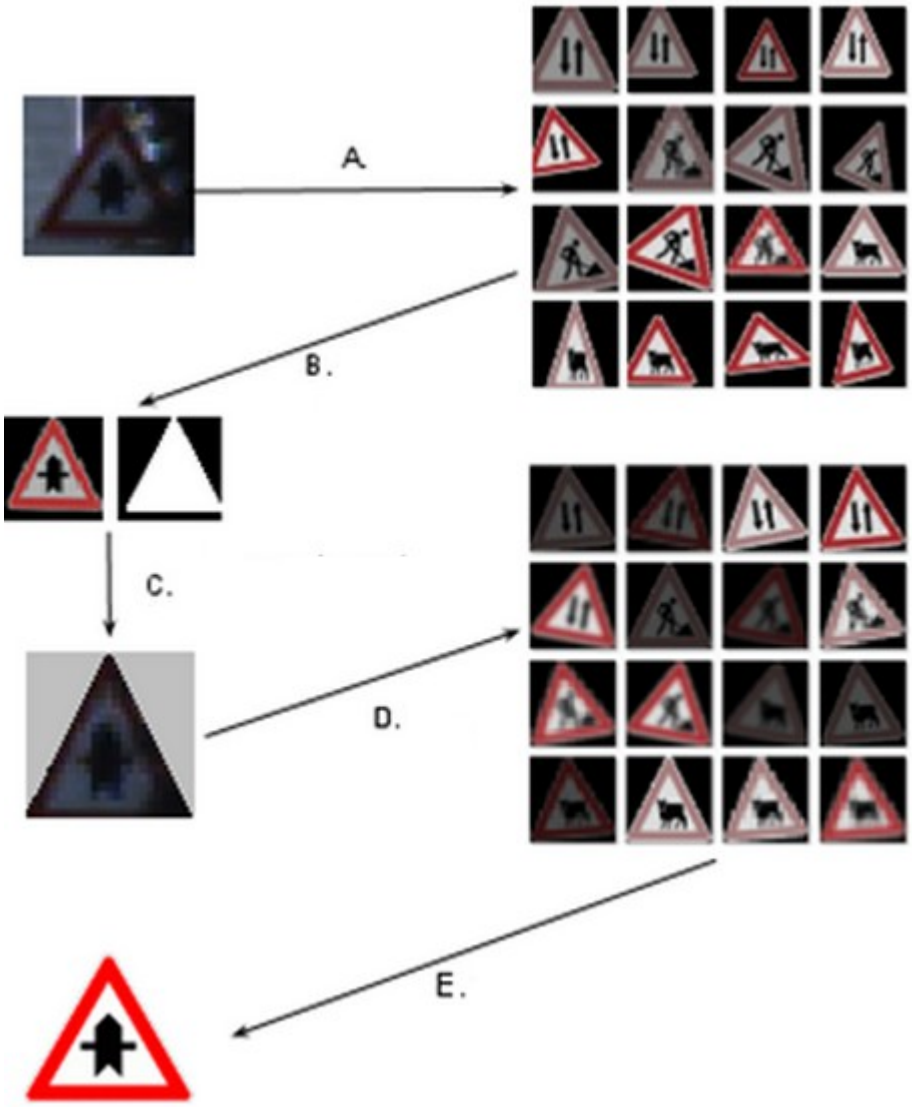


Рисунок 46. Схематическая иллюстрация метода сегментация объекта от фона: (a) поиск ближайшего соседа (b) оценка маски объекта (c) центрирование объекта. В случае применения классификатора на основе ближайшего соседа: (d) поиск ближайшего соседа (e) определение класса знака.

Примеры ближайших соседей, получаемых с помощью предложенного метода приведены на рисунке 47.

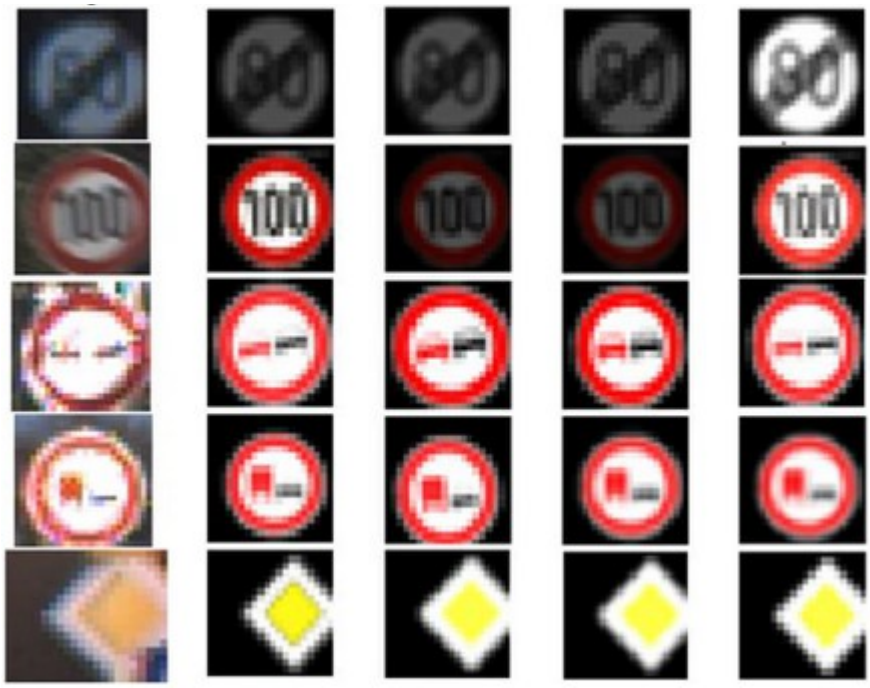


Рисунок 47. Примеры найденных ближайших соседей по метрике евклидова расстояния на HOG-дескрипторах. Левый столбец - входное реальное изображение. Остальные столбцы - найденные ближайшие соседи.

3.4. Алгоритм классификации

После сегментации объекта от фона изображение подается на вход классификатору, для получения класса обнаруженного объекта. В данной работе были экспериментально исследованы несколько вариантов классификации, различающиеся классификатором, обучающей выборкой и наличием или отсутствием вышеописанного этапа сегментации от фона.

Были протестированы следующие классификаторы:

1. классификатор на основе ближайшего соседа (NN) с евклидовым расстоянием в качестве меры похожести
2. линейный дискриминантный анализ (LDA)
3. метод опорных векторов с линейным ядром (linear SVM)
4. сверточная нейронная сеть с двумя сверточными слоями (112 фильтров в каждом, фильтры размеров 5, шаг фильтра 1, сабсемплинг размером 2) и двумя полносвязными слоями (состоящих из 100 и N нейронов, где N равно количеству классов в тестовой выборке).

Первые три классификатора из списка работают в пространстве HOG-дескриптора. Во всех экспериментах мы использовали одни и те же параметры HOG-дескриптора: размер ячейки - 3x3, размер блока - 6x6, шаг блока - 3x3, количество ячеек направления градиентов - 9 (противоположные направления градиентов попадали в одну ячейку). Все параметры были подобраны с помощью кросс-валидации.

3.4.1. Базы данных, использованные в экспериментах

Помимо RTSD и GTSRB, описанных ранее, в данной главе результаты также верифицируются на базе *Sweden traffic signs dataset* [4], которая состоит из 20000 изображений знаков, соответствующих 3488 физическим знакам (около 6 изображений на физический знак). База состоит из двух размеченных частей. Каждая часть содержит изображения знаков, разбитых на четыре группы - размытые, перекрытые, второстепенная дорога (side road) и видимые. Мы протестировали классификаторы на группе “видимые” каждой из частей, исключая классы знаков “OTHER” и “URDBL”. Таким образом, в нашем случае первая часть состояла из 1423 изображений, а вторая часть из 1373 изображений 18 классов знаков. Эта база знаков была также использована в [28], но прямое сравнение результатов методов классификации невозможно, так как авторы [28] использовали данную базу для оценки точности работы модуля обнаружения в связке с модулем распознавания.

В разделе 3.4.5. приведена точность работы метода на базе русских знаков (RTSD).

3.4.2. Параметры создания синтетической выборки

Для создания синтетической выборки мы использовали параметры, приведенные в таблице 9. Значения параметров описаны в разделе 1.3. Одни и те же параметры были использованы для создания синтетических выборок при тестировании на разных базах.

Набор #	V [%]	S [%]	R_x	R_y	R_z	σ_B	dx_l [%]	dx_r [%]	dx_u [%]	dx_d [%]	размер [px]	σ_N
1	70	50	-20:20	-30:30	-6:6	2	-4:4	-4:4	-4:4	-4:4	30x30	1.5
2	70	50	-10:10	-10:10	-3:3	2	-2:2	-2:2	-2:2	-2:2	30x30	1.5

Таблица 9. Параметры создания синтетической выборки для экспериментов с различными классификаторами.

3.4.3. Сравнение классификаторов на GTSRB

Обучающий набор #	Тип обучающей выборки	LDA [%]	Linear SVM [%]	k-NN [%]	CNN [%]
1	синтетическая	43.6	79.01	93.15	97.87
2	реальная	93.28	95.7	72.81	96.3
3	синтетическая, после сегментации от фона	83.22	91	96.91	---

Таблица 10. Сравнение точности работы различных классификаторов, обученных на реальных и синтетических данных базы GTSRB.

Синтетическая обучающая выборка была получена с параметрами из набора 1 (таблица 9) и включала более 100000 изображений. Обученные на данной выборке классификаторы показали точность, представленную в таблице 10 (обучающий набор #1). Обучение LDA и SVM на реальных данных позволяет добиться лучших результатов, если сравнивать с обучением на синтетических данных (таблица 10, обучающий набор #2). Мы объясняем такую значительную разницу в точности линейной природой классификаторов LDA и SVM, простой линейной модели недостаточно для классификации разнообразных изображений из синтетической выборки, что, в свою очередь, не позволяет классификатору достигнуть хорошей обобщающей способности.

Данная гипотеза также подтверждается результатами из таблицы 10, обучающий набор #3. В данном случае классификаторы были обучены на выборке, изображения в которой были подвержены значительно меньшим трансформациям (таблица 9, набор трансформаций #2). Входное изображение центрировалось и сегментировалось от фона с помощью предложенного алгоритма сегментации. Таким образом классификатор работал с упрощенными изображениями объектов. Видно, что уменьшение сложности обучающей выборки значительно увеличило точность классификации линейных классификаторов.

С другой стороны видно, что классификатор на основе ближайшего соседа, обученный на реальных данных показывает точность 72.81%, в то же время тот же классификатор, обученный на большой синтетической выборке, демонстрирует 93.15% точности. Эту разницу можно объяснить тем, что синтетическая выборка покрывает набор всевозможных трансформаций более плотно и равномерно, что позволяет

нелинейному классификатору (такому как NN) построить хорошую разделяющую поверхность.

Использование NN-классификатора совместно с методом сегментации от фона позволяет достичь точности в 95.3% на официальном тестовом наборе GTSRB. Дополнительное раздельное использование трех цветовых каналов и взвешивание HOG-дескриптора (чтобы увеличить важность центральных частей изображение) позволяет достичь точности в 96.91%.

CNN-классификатор, обученный на синтетических данных, также показывает лучшие результаты, по сравнению с CNN-классификатором, обученным на реальных данных. Этот факт можно объяснить также, как и в случае NN-классификатора - сверточная нейронная сеть является нелинейным классификатором, который может извлечь пользу из обучения на больших наборах сильно варьируемых изображений, представленных в синтетической выборке.

3.4.4. Результаты классификации базы шведских знаков

Для тестирования на шведской базе знаков мы использовали тот же набор параметров, что и в экспериментах с GTSRB, и получили точность в 97.47% на первой части и 98.61% на второй части базы, используя в качестве классификатора метод на основе ближайшего соседа. Мы нашли несколько ошибок разметки в каждой из частей базы, учет этих ошибок поднимает результаты классификации до 97.61% и 98.76%, соответственно. Сверточная нейронная сеть показала сравнимую точность в 97.69% и 99.05%.

3.4.5. Результаты классификации базы русских знаков (RTSD)

В таблице 11 приведена точность классификации знаков различных типов при обучении на синтетических данных. Точность была измерена для запуска в связке с модулем обнаружения, таким образом на вход модуля распознавания попадали только обнаруженные знаки.

Тип знака	Количество классов	Количество обучающих примеров	Доля распознанных физических знаков (среди обнаруженных) [%]
Синие квадраты	31	279000	96.6
Красные	46	414000	92.8

треугольники			
Синие круги	16	144000	100
Красные круги	47	423000	93.8

Таблица 11. Точность классификации различных типов обнаруженных знаков, при обучении на синтетической выборке.

3.4.6. Сравнение с обучением на реальных данных

В таблице 12 приведены точности классификации базы RTSD с помощью сверточной нейронной сети (показавшей лучшую точность в предыдущих экспериментах) для случаев обучения на реальных и на искусственно созданных данных. Видно, что обучение на синтетической выборке большого размера позволяет увеличить точность классификации по сравнению с обучением на маленькой выборке реальных данных.

В таблице 13 приведены зависимости точности классификации от типа обучающей выборки для других наборов данных.

Тип данных	Количество классов	Количество обучающих примеров на класс	Точность классификации [%]
Реальные	42	> 15	93.7
Синтетические	42	> 9000	**94.1**

Таблица 12. Сравнение точность классификации RTSD сверточной нейронной сетью при обучении на реальных и на синтетических данных.

База данных	Реальные данные, [%]	Синтетические данные, [%]
RTSD	93.7	**94.1**
GTSRB	96.3	**97.87**
Шведские знаки (часть 1/ часть 2)	---	97.69/99.05

Таблица 13. Сравнение точности работы сверточной нейронной сети, обученной на реальных и на синтетически созданных данных.

3.4.7. Анализ ошибок алгоритма

Визуальная инспекция ошибок алгоритма показывает, что в большинстве случаев ошибки возникают в силу двух причин:

1. незначительные отличия классов друг от друга. В качестве примера можно привести знаки, относящиеся к классам “второстепенная дорога” (рисунок 48)
2. отсутствие преобразования, которому подверглось входное изображение, в наборе преобразований, применяемых к пиктограмме в процессе создания искусственной обучающей выборки (рисунок 49, рисунок 50).

Рисунок 48. Пример классов знаков, представляющих наибольшие трудности для классификатора.

Рисунок 49. Примеры ошибок классификации бельгийской базе знаков. В первом и третьем столбце приведено входное изображение. Во втором и четвертом - пиктограмма предсказанного класса.

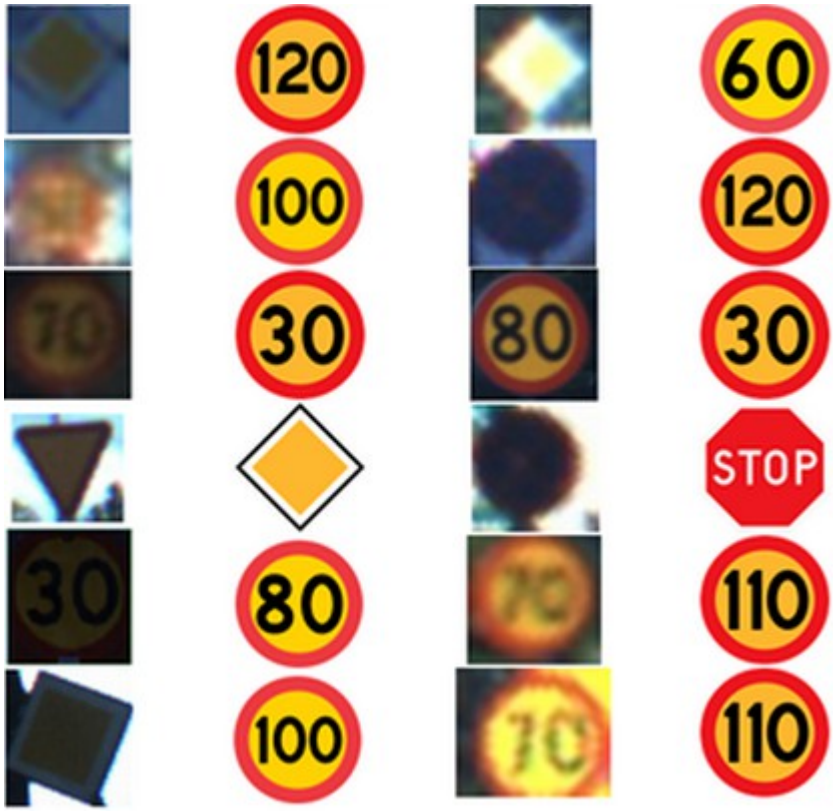


Рисунок 50. Примеры ошибок классификации шведской базе знаков. В первом и третьем столбце приведено входное изображение. Во втором и четвертом - пиктограмма предсказанного класса.

3.5. Заключение

В данной главе был проведен обзор методов классификации объектов, отдельно были рассмотрены методы классификации знаков дорожного движения.

В разделе 3.3. проведен анализ проблем, возникающих на стыке модуля обнаружения и модуля классификации. По результатам анализа предложен алгоритм центрирования и сегментации изображений объекта от фона, использующий наличие синтетически созданной обучающей выборки.

В разделе 3.4. экспериментально показано, что использование предложенного алгоритма совместно с четырьмя различными типами классификаторов позволяет значительно увеличить точность распознавания.

В разделе 3.4.6. проведено сравнение точности работы лучшего классификатора (сверточной нейронной сети) при обучении на реальных и на синтетических данных. Экспериментально показано, что использование большой синтетической выборки позволяет увеличить точность классификации.

В разделе 3.4.7 проведен анализ ошибок классификаторов и выделены две основные причины ошибок - незначительные отличия классов друг от друга и отсутствие преобразования, которому подверглось входное изображение, в наборе преобразований, накладываемых на пиктограмму.

Глава 4. Интеграция отдельных модулей в единую систему мобильного картографирования

4.1. Постановка задачи

Типичный комплекс мобильного картографирования состоит из следующих элементов (рисунок 51):

1. *мобильной платформы*. В большинстве случаев это автомобиль, но также в качестве мобильной платформы может выступать велосипед или мотоцикл
2. *видео или фотокамеры*. Камер может быть несколько - для увеличения угла обзора и точности локализации
3. *модуля геопозиционирования* (GPS, Глоннас). Позволяющего получать координаты мобильной платформы в мировой системе координат
4. *одометра или гиростабилизатора*. Для уточнения позиционирования мобильной платформы, особенно в условиях зашумленного сигнала с модуля геопозиционирования (например, в городах с плотной застройкой)
5. *модуля синхронизации устройств*. Отвечающего за синхронное выполнение измерений.

Рисунок 51. Основные компоненты системы мобильного картографирования.

Задачей комплекса мобильного картографирования является получение геопривязанных изображений исследуемой местности. После (или во время) получения, данные обрабатываются с целью извлечения полезной информации. Например, для нанесения объектов интереса на карту (рисунок 52), снятия метрических данных с объектов, проверки наличия объектов на положенном месте и отсутствия вандальных действий над объектом.

Таким образом, задачу нанесения объектов интереса на карту с помощью мобильной платформы можно сформулировать следующим образом - по набору геопривязанных изображений исследуемой местности необходимо получить список объектов интереса, каждый элемент которого обладает следующими характеристиками:

1. координаты объекта в мировой системе координат
2. класс объекта со степенью уверенности

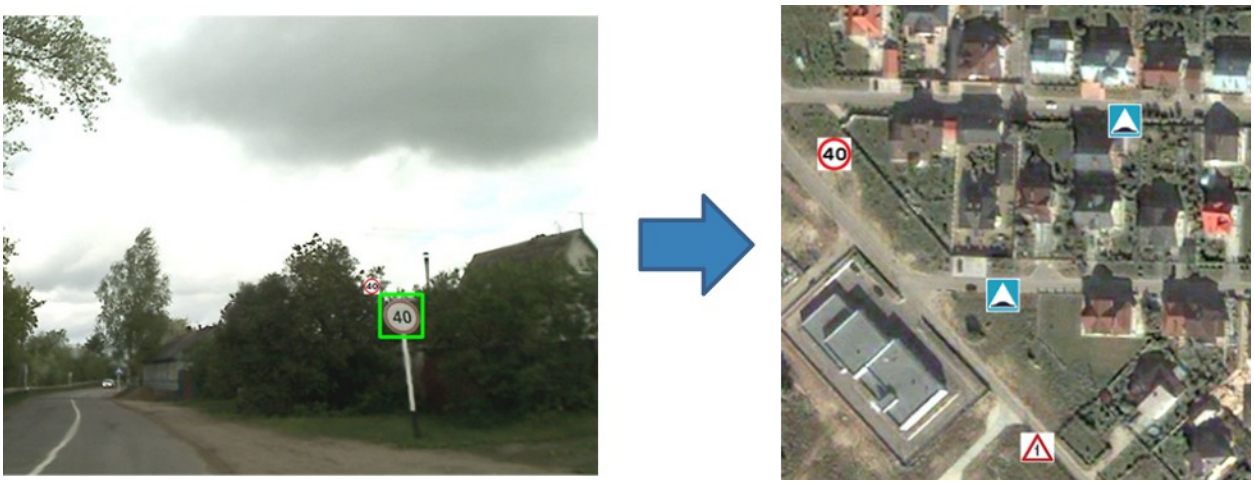


Рисунок 52. Результат работы системы мобильного картографирования на примере нанесения на карту знаков дорожного движения.

Во второй и третьей главах данной диссертации были предложены алгоритмы для решения задач выделения и классификации объектов интереса, способные к обучения на искусственно созданных данных. Предложенные алгоритмы работают на уровне отдельных кадров, не используя информацию по временной шкале видеопоследовательности. Использование данных алгоритмов в рамках системы полного цикла картографирования ставит ряд новых проблем (касающихся объединения результатов работы алгоритмов на отдельных кадрах), но также открывает ряд возможностей по улучшению точности работы алгоритмов за счет использования новой

информации. В дальнейших разделах данной главы описаны модули системы картографирования полного цикла.

4.2. Схема работы системы в целом

Разработанная система мобильного картографирования состоит из следующих модулей:

1. модуль обнаружения объекта
2. модуль сегментации объекта от фона
3. модуль классификации отдельного изображения объекта
4. модуль слежения
5. модуль уточнения класса физического объекта
6. модуль локализации
7. модуль объединения результатов локализации
8. модуль визуализации результатов.

Модули обнаружения, сегментации и классификации работают на отдельных кадрах входной видеопоследовательности и подготавливают результат для анализа модулями более высокого уровня. Модули слежения, уточнения класса физического объекта и локализации работают на уровне соседних кадров видеопоследовательности. Модуль объединения результатов локализации комбинирует результаты работы системы после анализа видеопоследовательностей, полученных с одного и того же участка проезда. Далее приведено более детальное описание каждого из вышеупомянутых модулей. Для упрощения восприятия информации каждый модуль описан по одному шаблону, состоящему из следующих частей:

1. описание входных и выходных данных
2. описание решаемой модулем задачи
3. описание принципов решения поставленной задачи.

4.3. Модуль обнаружения

Входные данные: отдельный кадр видеопоследовательности.

Задача: выделение ограничивающего прямоугольника объекта интереса.

Выходные данные: ограничивающие прямоугольники всех найденных объектов интереса.

Принцип работы: принцип работы модуля описан в Главе 2.

4.4. Модуль сегментации от фона

Входные данные: отдельный кадр видеопоследовательности, ограничивающий прямоугольник объекта интереса.

Задача: уточнить положение объекта интереса в ограничивающем прямоугольнике, отделить его от фона.

Выходные данные: уточненный ограничивающий прямоугольник объекта интереса, маска фона внутри прямоугольника.

Принцип работы: принцип работы модуля описан в разделе 3.3.

4.5. Модуль классификации отдельного изображения объекта

Входные данные: отдельный кадр видеопоследовательности, уточненный ограничивающий прямоугольник объектов интереса, маска фона внутри прямоугольника.

Задача: получить распределение вероятностей классов объекта.

Выходные данные: распределение вероятностей классов объекта.

Принцип работы: принцип работы модуля описан в разделе 3.4.

4.6. Модуль слежения

Входные данные: текущий кадр видеопоследовательности, ограничивающие прямоугольники объектов и распределение вероятностей классов объектов, история слежения по предыдущим кадрам.

Задача: объединить изображения одного и того же физического объекта в последовательность. Отклонить обнаружения, не складывающиеся в последовательность.

Выходные данные: последовательности пар (номер кадра, номер объекта) = (F_i, O_i) , объединенные по принципу принадлежности к одному и тому же физическому объекту.

Принцип работы: визуализация работы алгоритма приведена на рисунках 53-56. На вход алгоритму последовательно поступают кадры F_i с обнаруженными на них объектами интереса, представленными ограничивающими прямоугольниками O_i . Вместе с объектом интереса также поступает распределение вероятностей на классы объекта P_i .

Рассмотрим ситуацию появления новых объектов (O_1, O_2, O_3) в кадре F_1 (рисунок 53). При поступлении в модуль слежения первого кадра F_1 еще не была накоплена история слежения, поэтому на первом кадре алгоритм просто запоминает в

истории слежения текущие положения объектов, то есть записывает в историю три последовательности (S_1, S_2, S_3), состоящие из пар $(F_1, O_1), (F_2, O_2), (F_3, O_3)$ соответственно.

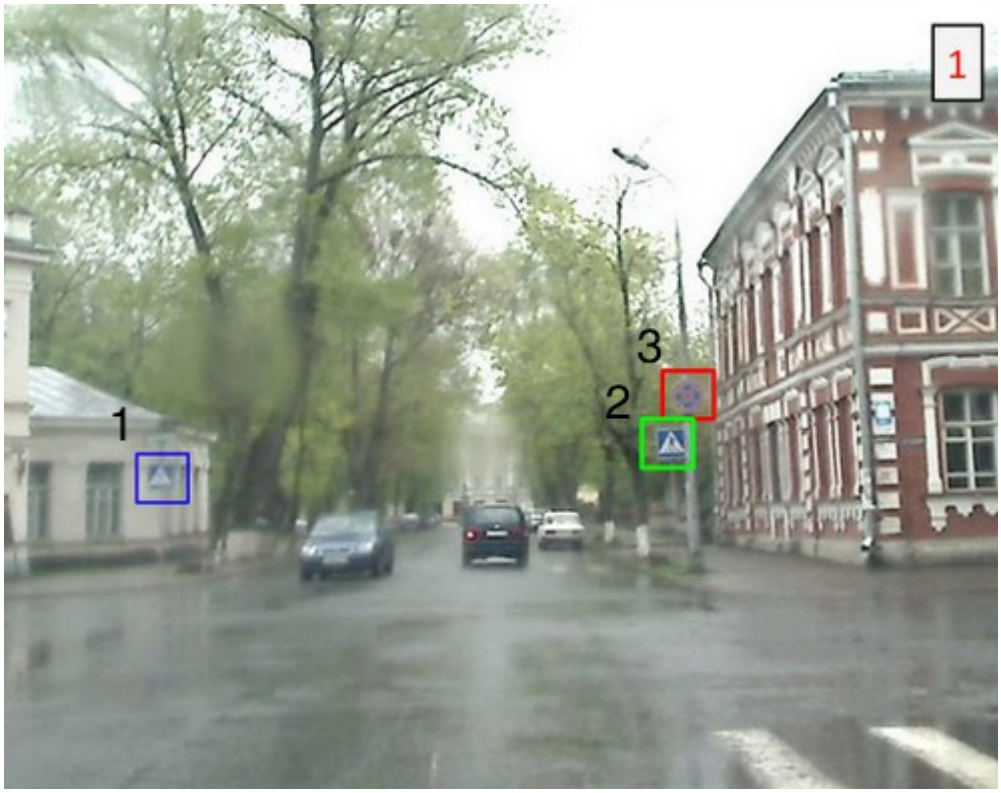


Рисунок 53. Первый кадр видеопоследовательности. Прямоугольниками выделены результаты работы модуля обнаружения.

После появления второго кадра F_2 (рисунок 54) алгоритм производит попытку дополнить существующие последовательности (S_1, S_2, S_3). Для этого на основании информации о предыдущих положениях объектов в каждой из последовательностей производится предсказание положения объекта на следующих кадрах. В случае кадра F_2 была накоплена статистика только по одному предыдущему кадру F_1, поэтому в качестве предсказания берутся положения объектов на кадре F_1 (обозначенные крестами на рисунке 54). После этого производится попытка объединить каждое предсказание с соответствующим ему наблюдаемым положением объектов на кадре F_2. Для этого используется метод минимизации суммы попарных расстояний между наблюдаемым и предсказанным положением объекта с ограничениями на сохранение взаимного положения объектов. Данные ограничения выводятся из априорных знаний о

предметной области. Например, если машина движется прямолинейно по ровной поверхности, то объекты интереса не могут поменять свое взаимное положение по вертикальной оси. То есть если ограничивающий прямоугольник объекта O_3 был расположен выше ограничивающего прямоугольника объекта O_2 , то на следующих кадрах такое взаимное расположение должно сохраниться. В результате сопоставления наблюдаемых данных с предсказаниями в рассматриваемом примере объекты O_2 и O_3 сопоставляются корректно. Объект O_1 , вышедший из поля зрения камеры, сохраняется в истории слежения на фиксированное количество кадров T , что позволяет придать устойчивость алгоритму слежения в случае ошибок второго рода со стороны модуля обнаружения. Если по истечении T кадров не удается продолжить последовательность S_1 , то объект считается вышедшим из поля зрения камеры и последовательность удаляется из истории слежения, поступая на выход модуля слежения. В реальных условиях установка $T=1$ позволяет модулю слежения быть устойчивым к большинству ошибок детектора, в то же время предотвращая некорректное объединение в единую последовательность изображений разных физических знаков.

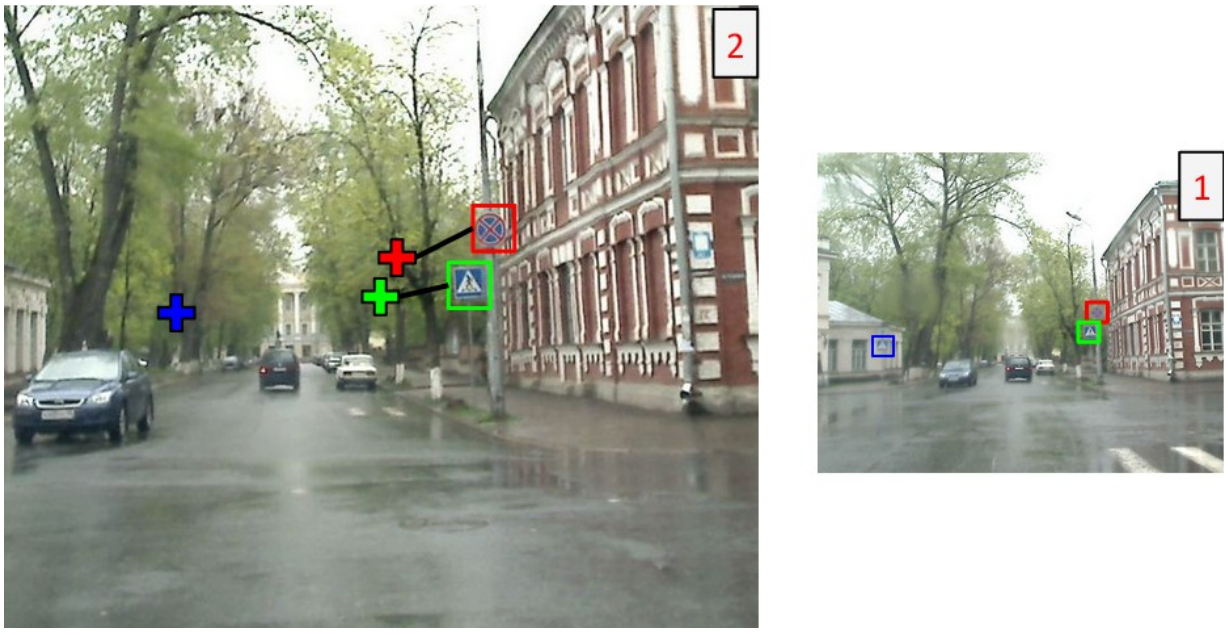


Рисунок 54. Слева - второй, справа - первый кадр видеопоследовательности. Крестами обозначены предсказанные положения объектов на втором кадре, на основании накопленной в истории слежения информации. Также, с помощью соединительных линий, показаны результаты сопоставления наблюдаемых обнаружений с предсказаниями.

При поступлении кадра F_3 алгоритму не удается продолжить последовательность S_1, поэтому она удаляется из истории слежения и поступает на выход модуля (рисунок 55). После этого производится попытка более точного предсказания положения объектов на кадре F_3, за счет использования информации о положении объектов на двух предыдущих кадрах. В наших экспериментах был использован простой метод предсказания положения объекта за счет решения уравнения равноускоренного движения:

$r_i = r_{i-1} + vt_i + \frac{at^2}{2}$, где

r_i - предсказание положения объекта на текущем кадре (неизвестное),

r_{i-1} - наблюдаемое положение объекта на предыдущем кадре,

v - скорость объекта,

a - ускорение объекта,

t - время, прошедшее между i -м и $i-1$ кадрами.

Скорость V и ускорение A объекта можно оценить используя историю слежения:

$$v = \frac{(r_{i-1} - r_{i-2})}{t_{i-1}}$$

$$a = \frac{(v_i - v_{i-1})}{t_{i-1}}$$

Таким образом, для оценки скорости нужно накопить историю по двум, а для оценки ускорения по трем кадрам. В случае, если истории недостаточно, скорость и ускорение принимаются равными нулю. Решая вышеописанное уравнение отдельно для каждой из осей координат изображения мы можем предсказывать положение объекта на следующих кадрах.

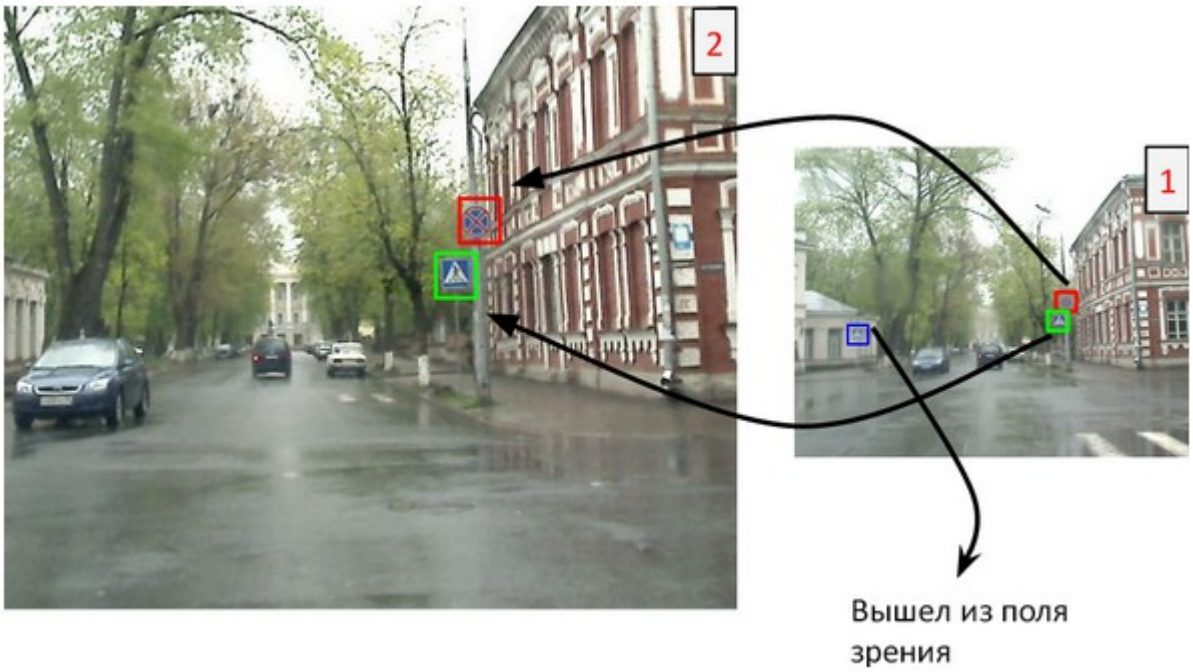


Рисунок 55. Стрелками показаны пары изображений объектов, сопоставленные между первым и втором кадром видеопоследовательности. Один из объектов не удалось сопоставить и он считается вышедшим из поля зрения системы и поступает на выход модуля сопоставления.

Последовательности, полученные в результате применения описанного алгоритма на рассматриваемом примере приведены на рисунке 56.

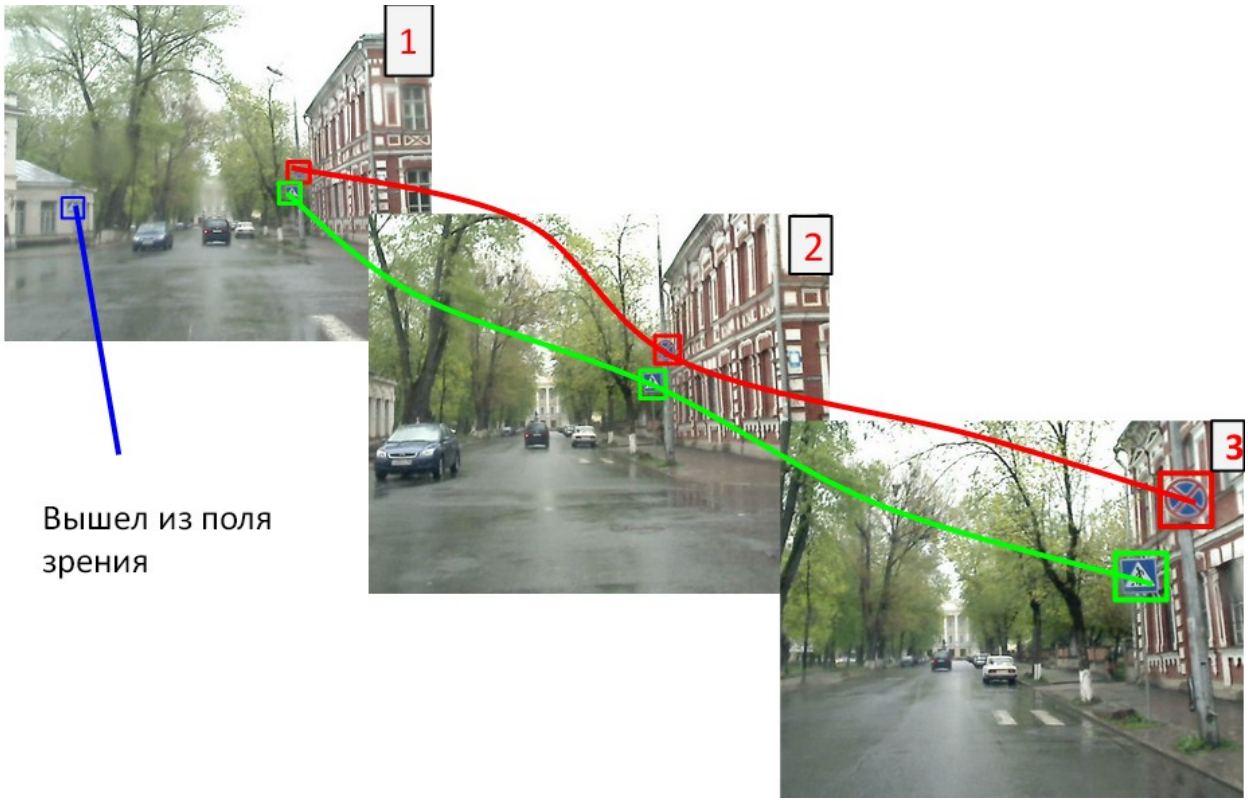


Рисунок 56. Результаты сопоставления на трех кадрах видеопоследовательности.

4.7. Модуль уточнения класса физического объекта

Входные данные: последовательность изображений объекта интереса, объединенных по принципу принадлежности к одному физическому объекту.

Задача: уточнить класс объекта, используя информацию с нескольких кадров.

Выходные данные: уточненный класс объекта.

Принцип работы: Пусть $S=\{s_{1,}s_{2,}...\}$ - последовательность знаков, связанных алгоритмом сопоставления в цепочку, где

$s_i=(d_i,r_i)$,

$d_i=(w_i,h_i)$ - ширина и высота ограничивающего прямоугольника,

$r_i=(p_{i1},p_{i2})$ - распределение вероятностей по классам знаков.

Тогда алгоритм уточнения класса знака возвращает класс, посчитанный по следующей формуле:

$$cl=argmax_c(\sum_i p_{ic}\cdot w_i\cdot h_i) \quad .$$

В результате возвращается класс объекта, имеющий наибольшую сумму вероятностей по отдельным кадрам, взвешенным на размеры ограничивающих

прямоугольников. Взвешивание производится для того, чтобы увеличить доверие к результатам распознавания, полученным по изображениям объекта большого размера.

4.8. Модуль локализации

Входные данные: последовательность изображений объекта интереса, внутренние и внешние параметры камеры.

Задача: получить положение объекта в мировых координатах.

Выходные данные: координаты объекта.

Принцип работы: Зная центры ограничивающих прямоугольников в последовательности и параметры камеры на каждом из рассматриваемых кадров, положение объекта в мировых координатах можно определить за счет широко известного метода триангуляции [70]. Принцип его работы изображен на рисунке 57. Если последовательность состоит из менее чем двух изображений объекта и метод триангуляции применить невозможно, то можно воспользоваться знаниями из предметной области и приблизительно предсказать расстояние до объекта на основе информации о его физических размерах.

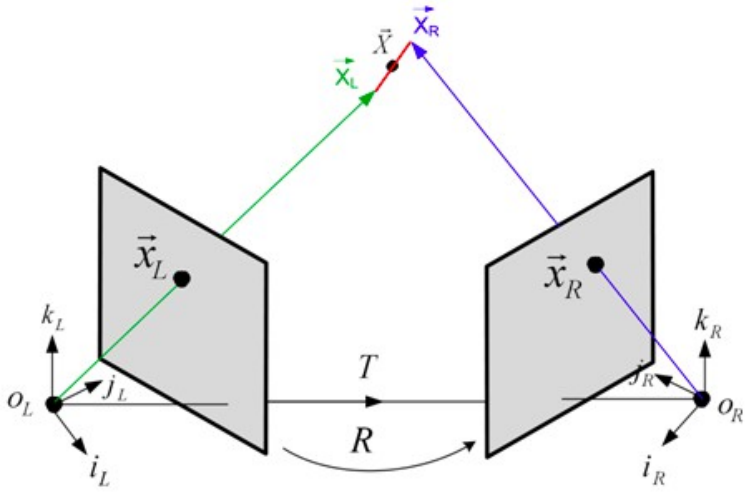


Рисунок 57. Визуализация принципа работы метода триангуляции. Зная внутренние и внешние параметры камеры можно определить положение объекта, за счет поиска в мировых координатах точки, равноудаленной от лучей, проходящих через центр объекта и оптический центр камеры.

4.9. Модуль объединения локализаций

Входные данные: результаты локализации объектов, полученные с нескольких проездов по одному и тому же участку в одном и том же направлении, классы локализованных объектов.

Задача: объединить последовательности, относящиеся к одному и тому же физическому объекту.

Выходные данные: координаты объектов после объединения.

Принцип работы: предполагая, что метод локализации не может ошибаться больше, чем на некоторое, заранее фиксированное расстояние D, метод объединяет в одну последовательность все локализации, находящиеся на расстоянии меньше D и имеющие один и тот же класс объекта.

4.10. Модуль визуализации результатов

Входные данные: результаты локализации объектов, классы локализованных объектов.

Задача: нанесение объектов на карту, визуализация маршрута движения мобильной платформы.

Выходные данные: карта, с нанесенными на неё объектами интереса.

Принцип работы: используя общедоступные системы визуализации геоданных, такие как Google Earth [71], и открытые форматы описания объектов на карте, такие как KML [72], результаты локализации можно отобразить на карте (рисунок 58). Подобного рода отображение упрощает дальнейшее опциональное исправление ошибок алгоритма силами операторов.

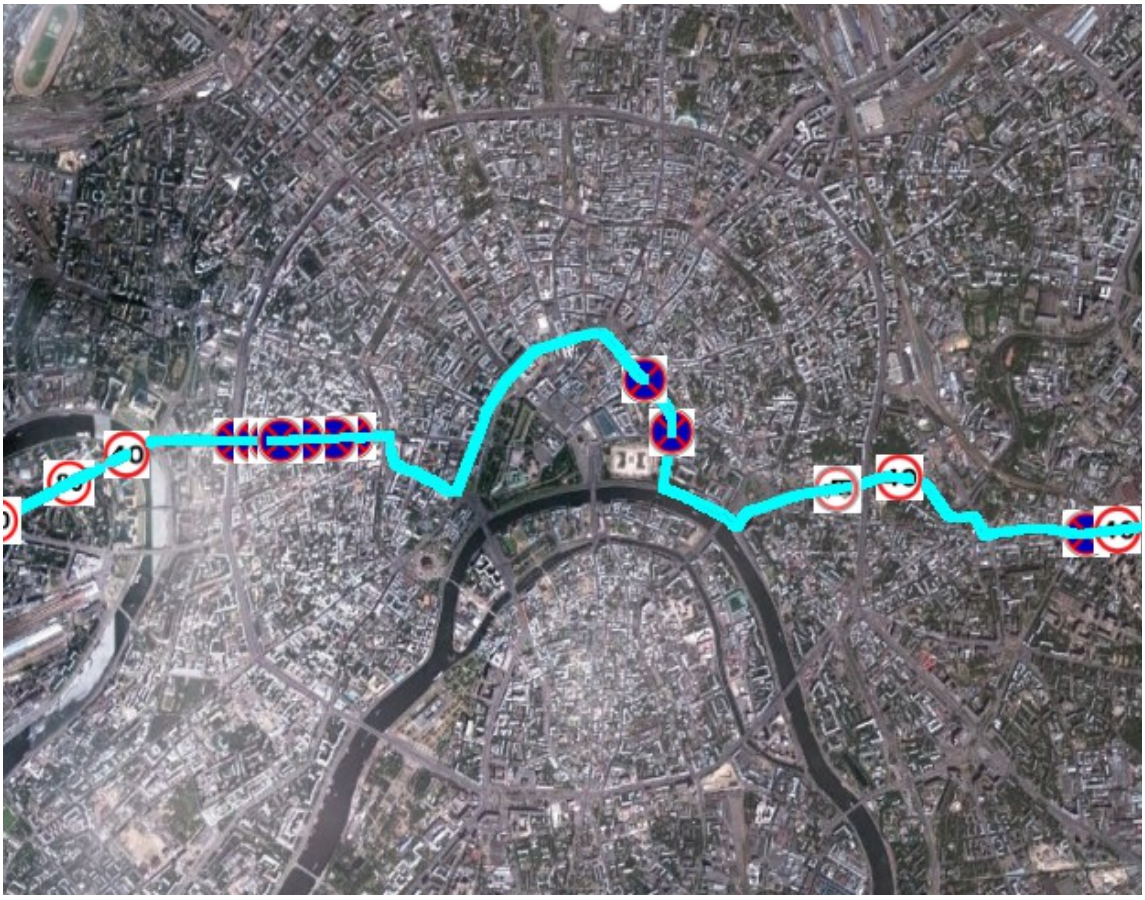


Рисунок 58. Пример визуализации работы автоматического алгоритма мобильного картографирования для случая локализации знаков нескольких выделенных классов.

4.11. Заключение

В данной главе сформулирована задача мобильного картографирования, используемая для апробирования предлагаемых в данной диссертации алгоритмов. В разделе 4.2. описана все компоненты предлагаемой автоматизированной системы и приведена общая схема их взаимодействия.

В разделах 4.3., 4.5., 4.5. описаны модули обнаружения, сегментации и классификации, являющиеся основными местами приложения алгоритмов, предлагаемых в данной диссертации. Подробное описание алгоритмов представлено в главах 2 и 3.

В разделах 4.6., 4.7., 4.8. описаны варианты реализации модулей, работающих на уровне нескольких соседних кадров видеопоследовательности. Описанные способы решения поставленных перед модулями задач не являются новыми и не сравниваются с соответствующими аналогами, а приведены для полноты картины о решаемой задаче, в

рамках практического апробирования предлагаемых методов. В то же время они решают свою задачу на уровне, достаточном для их практического применения.

В разделах 4.9. и 4.10. описаны важные для практической системы модули объединения локализаций и визуализации. Второй модуль особенно важен в случае, если после запуска автоматического алгоритма производится коррекция его результатов человеком.

Результаты работы

Основные результаты работы заключаются в следующем:

- предложен алгоритм создания искусственной обучающей выборки и способы оценки качества получаемых синтетических данных
- предложена модификация алгоритма обнаружения объектов Виолы-Джонса [1], использующая различные признаки на разных этапах каскада. Модифицированный алгоритм работает точнее и быстрее оригинального алгоритма
- на последнем этапе каскада детектора использована глубокая сверточная нейронная сеть, дающая прирост в 7% по полноте обнаружения при одном и том же уровне ошибки первого рода
- предложен метод сегментации объектов от фона, позволяющий увеличить точность распознавания при применении вместе с линейными и нелинейными классификаторами
- предложена схема обучения на искусственных данных глубокой сверточной нейронной сети для распознавания класса объекта. Показано, что обучение на больших объемах синтетических данных позволяет увеличить точность работы метода по сравнению с обучением на реальных данных.

Благодарности

Литература

1. Viola P., Jones M. Robust Real-Time Face Detection // International Journal of Computer Vision. VOL 57(2), P. 137-154.
2. Stallkamp J., Schlipsing M., Salmen J., Igel C. Man vs. Computer: Benchmarking Machine Learning Algorithms for Traffic Sign Recognition // Neural Networks. 2012. VOL 32, P. 323-332.
3. Belgium Traffic Sign Classification Benchmark, http://www.vision.ee.ethz.ch/~timofter/
4. Sweden Traffic Signs Dataset, http://www.cvl.isy.liu.se/research/traffic-signs-dataset
5. Moiseyev B., Konev A., Chigorin A., Konushin A. Evaluation of Traffic Sign Recognition Methods Trained on Synthetically Generated Data // Advanced Concepts for Intelligent Vision Systems. 2013. Springer LNCS. Vol. 8192. P. 576-83.
6. Чигорин А., Конушин А. Система автоматического картографирования знаков дорожного движения // Программные продукты и системы. 2013. С. 288-291.
7. Чигорин А., Конушин А. Эксперименты с обучением методов распознавания дорожных знаков на синтетических данных // Наука и образование: электронное научно-техническое издание. 2013. Т. 8. С. 315-24.
8. Chigorin A., Konushin A. A system for large-scale automatic traffic sign recognition and mapping // City Models, Roads and Traffic 2013. 2013. V. II-3/W3. P. 13-7.
9. Chigorin A., Krivovyaz G., Velizhev A., Konushin A. A method for traffic sign detection in an image with learning from synthetic data // 14th International Conference Digital Signal Processing and its Applications. 2012. V. 2. P. 316-319.
10. Моисеев Б., Чигорин А. Классификация автодорожных знаков с помощью свёрточной нейросети, обученной на синтетических данных // Graphicon. 2012. P. 284-287.
11. Konev A., Chigorin A., Krivovyaz G., Velizhev A., Konushin A. Traffic signs recognition on images with training on synthetic data // Technical vision in computer systems. 2012. P. 65-66.
12. Чигорин А. Автоматическое обнаружение 200 классов российских знаков дорожного движения // 15th International Conference Digital Signal Processing and applications. 2013. V. 2. P. 187-190.
13. Чигорин А., Конушин А. Сборник тезисов конференции Ломоносовские чтения. 2013. P. 42-43.

14. Shotton J., Fitzgibbon A., Cook M., Sharp T., Finocchio M., Moore R., Kipman A., Blake A. Real-Time Human Pose Recognition in Parts from a Single Depth Image // Proceedings IEEE Computer Vision and Pattern Recognition. 2011. P. 1297-1304.
15. Breiman L. Random forests // Machine Learning. 2001. VOL. 45(1). P. 5–32.
16. Grauman K., Shakhnarovichand G., Darrell T. Inferring 3D structure with a statistical image-based shape model // Proceedings Ninth IEEE International Conference on Computer Vision. 2003. P. 641-647.
17. Egisys Co. Curious Labs. Poser 5 : The ultimate 3D character solution. 2002.
18. Stark M., Goeseleand M., Schiele B. Back to the Future: Learning Shape Models from 3D CAD Data // Proceedings of the British Machine Vision Conference. 2010. P. 106.1-106.11.
19. Belongie S., Malik J. and Puzicha J. Shape Matching and Object Recognition Using Shape Contexts // IEEE Transactions on Pattern Analysis and Machine Intelligence. 2002. VOL. 24(4). P. 509-522.
20. Liebelt J., Schmid C., Schertler K. Viewpoint-Independent Object Class Detection using 3D Feature Maps // Proceedings Computer Vision and Pattern Recognition. 2008.
21. Thomas A., Ferrari V., Leibe B., Tuytelaars T., Schiele B., Gool L. V. Towards multi-view object class detection // In Conference on Computer Vision and Pattern Recognition. 2006.
22. Marin J., Vazquez D., Geronimo D., Lopez A. Learning Appearance in Virtual Scenarios for Pedestrian Detection // Proceedings Computer Vision and Pattern Recognition. 2010. P. 137-144.
23. http://ru.wikipedia.org/wiki/Half-Life_2
24. Pishchulin L., Thorm T., Wojek C., Andriluka M., Thormahlen T., Schiele B. Learning People Detection Models from Few Training Samples // Proceedings Computer Vision and Pattern Recognition. 2011. P. 1-8.
25. Wang K.,Babenko B.,Belongie S. End-to-End SceneText Recognition // International Conference on Computer Vision. 2011.
26. Novikova T., Barinova O., Kohli P., Lempitsky V. Large-Lexicon Attribute-Consistent Text Recognition in Natural Images // Proceedings European Conference on Computer Vision. 2012. VOL. 7577, P. 752-765.
27. Scherbaum K., Petterson J., Feris R., Blanz V., Seidel H. Fast Face Detector Training Using Tailored Views // Proceedings International Conference on Computer Vision. 2013. P. 2848-2855.

28. Larsson F., Felsberg M. Using Fourier Descriptors and Spatial Models for Traffic Sign Recognition // Proceedings of Scandinavian Conference on Image Analysis. 2011. P. 238-249.
29. Pauloand C., Correia P. Traffic Sign Recognition Based on Pictogram Contours // Proceedings of 9th International Workshop on Image Analysis for Multimedia Interactive. 2009. P. 67-70.
30. Overett G.,Tychsen-Smith L.,Petersson L.,Andersson L., Pettersson N. Creating Robust High-Throughput Traffic Sign Detectors Using Centre-Surround HOG Statistics // Machine Vision and Applications. 2011. P. 1-14.
31. Ciresan D., Meier U., Masci J. and Schmindhuber J. A Committee of Neural Networks for Traffic Sign Classification // IEEE International Joint Conference on Neural Networks. 2011. P. 1918-1921.
32. Sermanet P. and Lecun Y. Traffic Sign Recognition with Multi-Scale Convolutional Networks // International Joint Conference on Neural Networks. 2011. P. 2809-2813.
33. Zaklouta F., Stanciulescu B., Hamdoun O. Traffic sign classification using K-d trees and Random Forests // IEEE International Joint Conference on Neural Networks. 2011. P. 2151– 2155.
34. Dalal N., Triggs W. Histogram of oriented gradients for human detection // Proc. IEEE Conf. Comput. Vis. and Pattern Recog. 2005. P. 886-893.
35. Crow, Franklin. Summed-area tables for texture mapping // SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques. 1984. P. 207–212.
36. Shapiro R. The Boosting Approach to Machine Learning: An Overview // Nonlinear Estimation and Classification. 2003.
37. Felzenszwalb P., Girshick R., McAllester D. and Ramanan D. Object Detection with Discriminatively Trained Part Based Models // IEEE Transactions on Pattern Analysis and Machine Intelligence, VOL. 32(9).
38. Szegedy C., Toshev A., Erhan D. Deep Neural Networks for Object Detection // Advances in Neural Information Processing Systems. 2013.
39. Everingham M., Van Gool L., Williams C. K. I., Winn J., Zisserman A.. The pascal visual object classes (voc) challenge // International Journal of Computer Vision. 2010. VOL. 88(2), P. 303–338.
40. Alexe B., Deselaers T., Ferrari V. Measuring the objectness of image windows // IEEE Trans. Pattern Anal. Mach. Intell. 2012. VOL 34(11), P. 2189-2202.

41. Hou X. and Zhang L. Saliency Detection: A Spectral Residual Approach // Computer Vision and Pattern Recognition. 2007. P. 1-8.
42. Deng J., Dong W., Socher R., Li-Jia L., Li K., Fei-Fei L. ImageNet: A Large-Scale Hierarchical Image Database // Computer Vision and Pattern Recognition. 2009. P. 248-255.
43. http://image-net.org/challenges/LSVRC/2014/index#introduction
44. Sermanet P., Eigen D., Zhang X., Mathieu M., Fergus R., LeCun Y. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks // International Conference On Representation Learning. 2014.
45. Baro X., Escalera S., Vitria J., Pujol O., Radeva P. Traffic sign recognition using evolutionary Adaboost detection and Forest-ECOC classification // IEEE Transactions on Intelligent Transportation Systems. 2009. VOL 10(1), P. 113-126.
46. Balas B., Sinha P. STICKS: Image-representation via non-local comparisons // Journal of Vision. 2003. VOL 3(9).
47. Timofte R., Zimmermann K., Gool L. Multi-view traffic sign detection, recognition, and 3D localization // Workshop on Applications of Computer Vision. 2009. P. 1-8.
48. Mathias M., Timofte R., Benenson R., Van Gool L. Traffic Sign Recognition – How far are we from the solution? // In International Joint Conference on Neural Networks. 2013.
49. Overett G., Tychsen-Smith L., Petersson L., Andersson L., Pettersson N. Creating Robust High-Throughput Traffic Sign Detectors Using Centre-Surround HOG Statistics // Machine Vision and Applications (special issue paper). 2011. P. 1-14.
50. Russian traffic signs dataset, ftp://anonymous@ki-viuq.graphicon.ru/AnonymousFTP/RTSD/
51. Balas B., Sinha P. Dissociated Dipoles: Image Representation via Non-local Comparisons // CBCL Paper #229, 2003.
52. LeCun Y., Bottou L., Bengio Y., and Haffner P. Gradient-based learning applied to document recognition // Proceedings of the IEEE. 1998. VOL. 86(11). P. 2278–2324.
53. Krizhevsky A., Sutskever I., Hinton G. ImageNet Classification with Deep Convolutional Neural Networks // Advances in Neural Information Processing Systems, 2012, VOL. 25, P. 1097--1105.
54. Hinton G., Srivastava N., Krizhevsky A., Sutskever I., Salakhutdinov R. Improving neural networks by preventing co-adaptation of feature detectors // Technical report, arXiv:1207.0580, 2012.

55. Warde-Farley D., Goodfellow I., Courville A., Bengio Y. An empirical analysis of dropout in piecewise linear networks // arXiv:1312.6197v2, 2014.
56. Cuda-convnet library, https://code.google.com/p/cuda-convnet/
57. Ruta, A., Porikli, F.,Watanabe, S., Li, Y., 2011. In-vehicle camera traffic sign detection and recognition // Mach. Vis. Appl., VOL 22(2), P. 359–375.
58. Dance C., Willamowski J., Fan L., Bray C., Csurka G. Visual categorization with bags of keypoints // ECCV International Workshop on Statistical Learning in Computer Vision, 2004.
59. Lowe D. Object Recognition from local scale–invariant features // International Conference on Computer Vision, 1999.
60. Bissacco A., Cummins M., Netzer Y., Neven H.. PhotoOCR: Reading Text in Uncontrolled Conditions// International Computer Vision Conference, 2014.
61. Girshick R., Donahue J., Darrell T., Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation / / Tech report, 2014. http://arxiv.org/pdf/1311.2524v2.pdf
62. Razavian S., Azizpour H., Sullivan J., CarlssonS. CNN Features off-the-shelf: an Astounding Baseline for Recognition. http://arxiv.org/pdf/1403.6382v2.pdf
63. Sermanet P. Deep ConvNets: “astounding” baseline for vision. http://cs.nyu.edu/~sermanet/papers/Deep_ConvNets_for_Vision-Results.pdf
64. Paclik P., Novovicova J., Duin R. Building Road-Sign Classifiers Using a Trainable Similarity Measure // IEEE Trans. Intell. Transp. Syst. 2006. VOL 7(3), P. 309-321.
65. Miura J., Kanda T. and Shirai Y. An active vision system for real-time traffic sign recognition // IEEE Intelligent Transportation Systems Proceedings. 2000. P. 52-57.
66. Muyan-Ӧzçelik P., Glavtchev V., Ota J., Owens J. A Template- Based Approach for Real-Time Speed-Limit-Sign Recognition on an Embedded System Using GPU Computing // The German Association for Pattern Recognition Symposium. 2010. P. 162-171.
67. Maldonado-Bascón S., Lafuente-Arroyo S., Gil-Jimenez P., Gomez-Moreno H. and Lopez-Ferreras F. Road-Sign Detection and Recognition Based on Support Vector Machines // IEEE Trans. Intell. Transp. Syst. 2007. VOL. 8(2). P. 264-278.
68. Timofte R., Zimmermann K. and Gool L. V. Multi-view traffic sign detection, recognition, and 3D localization // Workshop on Applications of Computer Vision. 2009. P. 1-8.

69. Bahlmann C., Ramesh V., Pellkofer M. and Koehler T. A system for traffic sign detection, tracking, and recognition using color, shape, and motion information // IEEE Proceedings Intelligent Vehicles Symposium. 2005. P. 255-260.
70. Hartley R. and Zisserman A. Multiple View Geometry in computer vision // Cambridge University Press, 2003.
71. Google.Earth, http://www.google.com/earth/
72. KML standart, http://ru.wikipedia.org/wiki/KML