Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский Нижегородский государственный университет им. Н.И. Лобачевского»

Институт информационных технологий, математики и механики

Кафедра математического обеспечения и суперкомпьютерных технологий

Отчёт по производственной практике

Обзор задач компьютерного зрения, возникающих при разработке систем помощи водителю, для решения которых применяются методы глубокого обучения

Выполнил:

студент ИИТММ, гр. 381406-1

Доронин М.А.

Проверил:

ст.преп. каф. МОСТ, ИИТММ

Кустикова В.Д.

Нижний Новгород

2017

Содержание

[Введение 3](#_Toc484961067)

[1 Постановка задачи 4](#_Toc484961068)

[2 Обзор задач компьютерного зрения 5](#_Toc484961069)

[2.1 Общие задачи компьютерного зрения 5](#_Toc484961070)

[2.2 Задачи, возникающие при разработке систем помощи водителю 6](#_Toc484961071)

[2.3 Выбор задачи 7](#_Toc484961072)

[3 Обзор методов решения выбранной задачи 8](#_Toc484961073)

[3.1 Деревья решений 8](#_Toc484961074)

[3.2 SIFT дескриптор 9](#_Toc484961075)

[3.3 HOG дескриптор 9](#_Toc484961076)

[3.4 Метод опорных векторов 9](#_Toc484961077)

[3.5 Сверточные сети 10](#_Toc484961078)

[4 Обзор наборов изображений и видео, используемых для оценки качества решений 12](#_Toc484961079)

[4.1 Сравнение наборов данных 12](#_Toc484961080)

[4.2 Выбор тренировочного и тестового набора 14](#_Toc484961081)

[5 Библиотеки глубокого обучения 15](#_Toc484961082)

[5.1 Caffe 15](#_Toc484961083)

[5.2 Torch 15](#_Toc484961084)

[5.3 Tensorflow 16](#_Toc484961085)

[6 Программная реализация классификатора 17](#_Toc484961086)

[6.1 Используемые инструменты 17](#_Toc484961087)

[6.2 Данные 17](#_Toc484961088)

[6.3 Методы решения задачи 17](#_Toc484961089)

[6.4 Вычислительные результаты 18](#_Toc484961090)

[Заключение 19](#_Toc484961091)

[Литература 20](#_Toc484961092)

[Ссылки 21](#_Toc484961093)

# Введение

Компьютерное зрение – дисциплина (раздел искусственного интеллекта), занимающаяся извлечением информации из изображений. Причем изображения могут быть различного типа и происхождения: фотографии, изображения из видеопотока, медицинский снимок, полученный с помощью МРТ, и так далее. Под извлечением информации понимается обнаружение, отслеживание и классификация объектов.

Мощное развитие этой дисциплины в последние десятилетия связано с несколькими причинами: во-первых, возросла вычислительная мощность компьютеров, что очевидно упрощает задачу обучения сложных ресурсоемких систем. Во-вторых, с развитием индустрий появляется все больше данных для обучения, а также качество самих данных возрастает вместе с ростом качества фото- и видеокамер, медицинских приборов и прочих инструментов, извлекающих визуальную информацию. В-третьих, улучшаются и оптимизируются алгоритмы, и это в первую очередь связано с ростом интереса к области машинного обучения в целом и компьютерного зрения в частности.

Роль компьютерного зрения в настоящее время велико, а учитывая развитие науки и технологий, будет расти быстрыми темпами в будущем. Компьютерное зрение находит широкое применение в науке и индустрии. Прежде всего, конечно, в робототехнике, в том числе системах автономного вождения, о которых пойдет речь в данной работе. Первые системы, использующие компьютерное зрение были связаны с промышленными роботами, которые проверяли детали на заводском конвейере на наличие брака. На сегодняшний день перед роботами стоят задачи намного сложнее: теперь программа, заложенная в робота, должна анализировать сложный реальный мир, что в нем находится, как движется, какие намерения имеют те или иные объекты.

Другой областью применения служат системы, связанные с интернетом и так называемым Big Data. Основная задача таких систем обрабатывать, понимать, индексировать миллиарды изображений, загружаемых в мировую сеть ежедневно. Компьютерное зрение находит применение также в медицине, игровой индустрии, системах видеонаблюдения, биометрии, распознавания лиц и так далее. В настоящий момент, применение машинного обучения в целом и компьютерного зрения в частности становится шире с каждым днем, что делает изучение этой области актуальным.

Особенно актуальными кажутся в настоящий момент задачи, связанные с разработкой автономных систем помощи водителю. Главной причиной их разработки является повышение безопасности на дорогах. Учитывая, насколько часто каждый человек сталкивается с необходимостью пользоваться транспортом или просто перейти дорогу, и насколько высок шанс аварийных ситуаций, такие системы быстро найдут спрос и быстро будут развиваться, что доказывает современное развитие данной индустрии.

# Постановка задачи

В ходе учебной практики требуется:

1. Провести обзор задач компьютерного зрения, возникающих при разработке систем помощи водителю, для решения которых применяются методы глубокого обучения.
2. Выбрать задачу компьютерного зрения, возникающей при разработке систем помощи водителю, для последующего решения.
3. Провести обзор методов решения выбранной задачи.
4. Провести обзор наборов изображений/видео, используемых для оценки качества решений.
5. Выбрать и сконструировать тестовое и тренировочное множество данных.
6. Изучить возможности библиотек глубокого обучения, провести тестовые запуски на простейшей задаче классификации.

# Обзор задач компьютерного зрения

В данной главе описаны общие задачи компьютерного зрения и задачи компьютерного зрения, возникающие при разработке систем помощи водителю, для решения которых применяются методы глубокого обучения.

## Общие задачи компьютерного зрения

Каждая из областей применения компьютерного зрения, связана с рядом задач, описанных в этом разделе. Более или менее хорошо определённые проблемы измерения или обработки могут быть решены с использованием множества методов. Некоторые примеры типичных задач компьютерного зрения представлены ниже.

Распознавание

Классической задачей компьютерного зрения является распознавание некоторого характерного объекта, особенности или активности на изображении или видеоданных. Эта задача достоверно решается человеческим зрением, но до сих пор не решена удовлетворительно в компьютерном зрении в общем случае, то есть детектирование случайных объектов в случайных ситуациях затруднительно для существующих алгоритмов и методов.

Существующие методы решения этой задачи эффективны только для отдельных объектов, таких как простые геометрические объекты, человеческие лица, символы и текст, автомобили и так далее. Тем не менее, вероятность успешного решения задачи с такими объектами высока только при наличии определенных условий, таких как освещение, фон, положение объекта.

В данной задаче компьютерного зрения описывается множество проблем распознавания:

* Распознавание: один или несколько предварительно заданных или изученных объектов или классов объектов могут быть обнаружены на изображении.
* Идентификация: распознается индивидуальные признаки объекта, например, идентификация определенного человеческого лица, отпечатка пальца или автомобиля.
* Обнаружение: данные проверяются на наличие определенного условия, например, обнаружение вредоносного влияния на камеру или объекта, который по каким-либо признакам отличается от остальных.
* Поиск изображений по содержанию: в большом наборе данных необходимо найти те изображение, которые содержат некоторые характерные признаки. Такие признаки могут задаваться разными путями, например, в терминах схожести с конкретным изображением необходимо найти все похожие на него.
* Оценка положения: определение положения или ориентации определённого объекта относительно камеры.
* Оптическое распознавание знаков: распознавание символов на изображениях печатного или рукописного текста, обычно для перевода в текстовый формат, наиболее удобный для редактирования или индексации.

Движение

В данной задаче решаются проблемы оценки движения в некоторой последовательности изображений или в видеопотоке для нахождения скорости движения точек и объектов на сцене. Задача движения решается следующими путями:

1. Определение трехмерного движения камеры, с которой алгоритм получает набор изображений.
2. Слежение: следование за перемещениями определенного объекта, например, автомобилей или людей.

Восстановление сцены

Данная задача сводиться к восстановлению трехмерной модели сцены по набору изображений или видео. В простейших случаях, получаемая модель может состоять из набора характерных точек трехмерного пространства. Более продвинутые методы могут воссоздать полную трехмерную модель.

Восстановление изображений

Задача восстановления изображений сводится к удалению шума и других воздействий с них. Наиболее простым подходом к данной задаче является создания различных фильтров, более сложные алгоритмы используют представления того, как должны выглядеть те или иные участки изображения.

## Задачи, возникающие при разработке систем помощи водителю

Традиционная технология содействия водителю позволяет распознавать некоторые объекты, осуществлять базовую классификацию, извещать водителя об опасных ситуациях и, в некоторых случаях, замедлять или останавливать движение автомобиля. Этот уровень развития технологии отлично справляется с обнаружением мертвых зон, содействием при смене полосы движения и предупреждениями об опасности столкновений.

Программные системы с искусственным интеллектом, устанавливаемые на автомобилях, используют алгоритмы глубокого обучения. Разработки многих ведущих it-компаний и автомобильных компаний позволяют устанавливать на современных автомобилях системы содействия водителю, которые могут отличить полицейский автомобиль от такси, машину скорой помощи от автофургона, припаркованный автомобиль – от того, который через мгновение собирается выехать на полосу движения. На этом ее возможности не ограничиваются: она может идентифицировать все, начиная от движущихся по тротуару велосипедистов и заканчивая невнимательными пешеходами.

Современные вычислительные мощности при небольших размерах вычислительных систем и их относительно низком энергопотреблении предоставляют возможности по внедрению в автомобили таких систем. Ниже представлен список задач, которые решаются такими системами:

* Предупреждение о выходе за пределы полосы движения.
* Распознавание дорожных знаков.
* Интеллектуальную регулировку передних фар.
* Обнаружение и классификация объектов и обнаружение пешеходов.
* Мониторинг слепых пятен.
* Обнаружение сонливости водителя.
* Система предупреждения столкновений.
* Помощь при парковке.
* Автоматическая тормозная система.
* Адаптивный круиз-контроль во всем диапазоне скоростей.

Исходя из данного списка наиболее популярных задач можно сделать вывод о наиболее часто используемых методах их решения. Такими задачами являются задачи детектирования объектов на изображениях, полученных с камеры на автомобиле, классификация этих объектов и получение полной картины окружающей обстановки методом семантической сегментации изображений дорожных сцен.

Детектирование

Детектирование – поиск объектов на изображении. Целью детектирования является определение наличия объекта на изображении и нахождение его положения в системе координат пикселей исходного изображения. Положение объекта может определяться координатами окаймляющего прямоугольника, контуром объекта, либо координатами точек, наиболее характерных для объекта. Выбор способа определения положения зависит от данных и алгоритма детектирования.

Решение задачи детектирования объектов на изображении позволяет провести качественный анализ состава сцены, а также получить информацию о взаимном расположении объектов на ней.

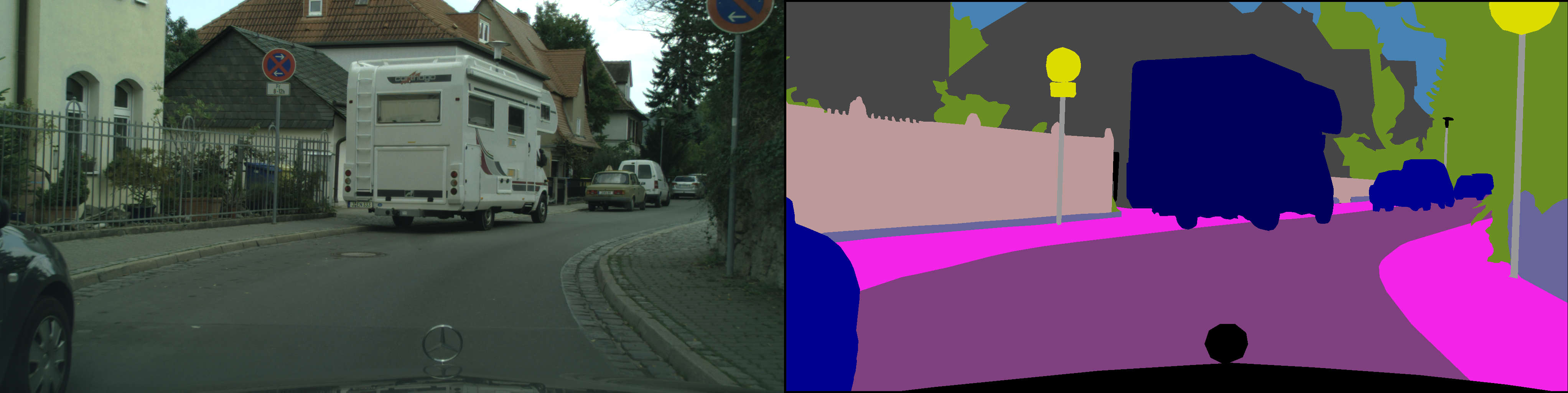
Семантическая сегментация

Семантическая сегментация изображений – это разделение изображения на отдельные группы пикселей, области (так называемые суперпиксели), которые соответствуют одному конкретному объекту с одновременным определением его типа. Задача семантической сегментации является высокоуровневой задачей обработки изображений, относящейся к группе задач так называемых слабого искусственного интеллекта. Данная задача является более сложной, относительно задач классификации и детектирования объектов на изображении, что обусловлено как необходимостью определения классов объектов, так и выявления их структуры и правильного выделения частей и границ этих объектов на изображении.

Конкретным приложением, для которого важны методы семантической сегментации, являются системы помощи водителю с целью решения множества задач из списка, который был указан выше. Семантическая сегментация позволяет наилучшим образом создавать системы автономной навигации автомобиля, поскольку решение именно этой задачи дает наиболее полное описание окружающего мира, представленного в виде изображений или видео с камеры на борту автомобиля.

## Выбор задачи

Благодаря широкой области практического применения в качестве задачи, предлагается рассмотреть в данной работе именно задачу семантической сегментации дорожных сцен (Рис. 1), полученных с камер, установленных на борту автомобиля.



1. Пример семантической сегментации изображения дорожной сцены, взятого из базы Cityscapes [4]

# Обзор методов решения выбранной задачи

В данном разделе будут рассмотрено несколько методов решения задачи семантической сегментации изображения.

Пусть дано множество изображений дорожных сцен и конечный набор классов объектов , например, автомобиль, пешеход, тротуар. Также для тренировочной выборки каждому изображению сопоставлена разметка , то есть для каждого пикселя изображения сопоставлена метка класса . Решением задачи является восстановление разметки для входных изображений . То есть необходимо для всех объектов на изображении определить положение и принадлежность какому-либо классу.

Такой процесс разбиения изображения на сегменты с учетом семантики, то есть с учетом приписывания каждому участку верной метки класса, носит названия семантической сегментации.

Сегментация изображений при помощи «традиционных» методов (3.1–3.4) отличается тем, что в основе их работы лежит информация о пространственных признаках объекта, а также отсутствием нейросетевой составляющей. Относительная простота реализации и теоретической базы делают их широко распространенными

## Деревья решений

В процессе развития интеллектуальных систем, деревья решений стали широко применяться в задачах компьютерного зрения, таких как распознавание объектов, классификация текстов, распознавание жестов, обнаружение спама, обучение ранжированию в информационном поиске, семантическая сегментация и кластеризация данных. Такому развитию и применению деревьев решений способствуют некоторые отличительные особенности деревьев решений, как интерпретируемость, управляемость и возможность автоматического отбора информативность признаков. Однако, имеется и ряд недостатков такого алгоритма при применении его в задачах компьютерного зрения.

Автоматический отбор информативных признаков позволяет автоматически размещать признаки в вершины дерева из набора признаков. Поэтому при решении задачи возможно составить произвольный набор признаков, которые в процессе обучения отфильтруются по информативности, а неинформативные признаки проигнорируются.

Интерпретируемость позволяет строить решающие правила в форме, понятной человеку. Это бывает необходимо в случае, когда человеку требуется понимать, каким образом алгоритм принимает решения, или почему он работает неправильно.

Управляемость позволяет переобучить дерево решений локально, то есть только те вершины, из-за которых примеры классифицируются неправильно. Это может оказаться удобным, когда неправильно классифицируется только часть выборки.

В деревья решений присутствует сильная зависимость от сбалансированности числа обучающих примеров разных классов. При обучении деревьев решений в его вершины выбираются признаки с максимальной информативностью, то есть признаки, порождаемые разбиением тренировочных данных на две части. Таким образом, при обучении особое внимание уделяется классам с большим числом обучающих данных, и могут полностью проигнорироваться классы с малым числом обучающих данных. Эта особенность требует особого внимания к данным и к балансировке дерева, что может привести к необходимости ручной настройки. Также сильная зависимость может привести к явлению переобучения.

Переобучение может возникнуть из-за излишней сложности модели и недостаточного объема обучающей выборки. Для деревьев решений сложность модели выражается в глубине дерева. Экспоненциальное уменьшение обучающей выборки при увеличении глубины дерева сильно может повлиять на шанс возникновения переобучения в какой-либо ветви дерева. Для обучения деревьев решений требуются специальные методы предотвращения переобучения. Зачастую применятся метод удаления тех вершин, которые ухудшают классификацию. Также для решения перечисленных проблем применяется не одно дерево решений, а лес решений. Еще одним способом избавления от переобучения является метод случайного перемешивания и добавления похожих примеров.

Учитывая особенности деревьев решений и его недостатки при решении задач компьютерного зрения следует обратить внимания на другие методы, которые могут оказаться более эффективными.

## SIFT дескриптор

SIFT (Scale Invariant Feature Transform) – дескриптор, представляющий собой локальную гистограмму направлений градиентов на изображении. Он строится по следующему принципу: окрестность некоторой «характерной» точки делится на четыре квадратных сектора, в каждом пикселе которых вычисляется градиент изображения, его направление и значение. Затем значения градиента умножаются на вес, экспоненциально убывающий с удалением от рассматриваемой характерной точки. По каждому сектору вычисляется гистограмма направлений градиентов, а дескриптор SIFT будет представлять вектор, полученный из значений всех элементов гистограмм направлений.

Используя дескрипторы характерных точек можно сопоставлять изображения между собой, поэтому дескрипторы могут находить применение в задачах семантической сегментации.

SIFT дескрипторы имеют также ряд недостатков, например, не все полученные характерные точки будут действительно отвечать некоторым требованиям. Данный факт будет влиять на дальнейшее решении задачи сопоставления изображений. В некоторых случаях, решение может быть не найдено, даже если оно существует.

## HOG дескриптор

HOG (Histogram of oriented gradients) – дескриптор особых точек, который так же как SIFT используется в компьютерном зрении и обработке изображений с целью распознавания объектов. Отличие данного метода от других дескрипторов в том, что вычисляется на плотной сетке равномерно распределенных ячеек и использует нормализацию перекрывающегося локального контраста для увеличения точности.

Идея алгоритма заключается в том, что внешний вид и форма исследуемого объекта на фрагменте изображения могут быть описаны распределением градиентов интенсивности и направлением краев. Реализация HOG может быть осуществлена путем разделения изображения на небольшие области, которые называют «ячейками», в которой далее рассчитывается гистограмма направлений градиентов и направлений краев. В совокупности эти гистограммы и являются дескриптором. Для увеличения точности гистограммы подвергаются нормализации по контрасту.

Дескрипторы характерных точек позволяют получить векторы признаков изображений, который далее необходимо разделить на группы для классификации этих изображений. Такой процесс можно осуществить, например, с помощью метода опорных векторов.

## Метод опорных векторов

Метод опорных векторов (SVM, support vector machine) – набор алгоритмов обучения с учителем, которые используются для задач классификации и регрессионного анализа. Метод принадлежит семейству линейных классификаторов. Основная идея данного метода – перевод исходных векторов признаков в пространство более высокой размерности и поиск разделяющих гиперплоскостей с максимальным зазором в этом пространстве.

Две параллельных гиперплоскости строятся по обеим сторонам гиперплоскости, разделяющих классы признаков. Разделяющей гиперплоскостью тогда будет такая плоскость, которая максимизирует расстояние до двух построенных параллельных плоскостей. Алгоритм работает на предположении, что чем больше расстояние между гиперплоскостями, тем меньше средняя ошибка классификатора.

Обычно SVM применяется для классификации признаков, получаемых на выходе сверточной сети вместо персептрона, о которых пойдет речь далее.

## Сверточные сети

В настоящее время задачи классификации, детектирования и локализации объектов на изображениях или видеопотоке эффективно решаются алгоритмами, использующие сверточные нейронные сети. Архитектура искусственных нейронных сетей была предложена Яном Лекуном для эффективного распознавания изображений. Подробнее о нейронных сетях описано в работе [1].

Сверточные нейронные (convolutional neural network, CNN) сети входят в состав технологий глубокого обучения.

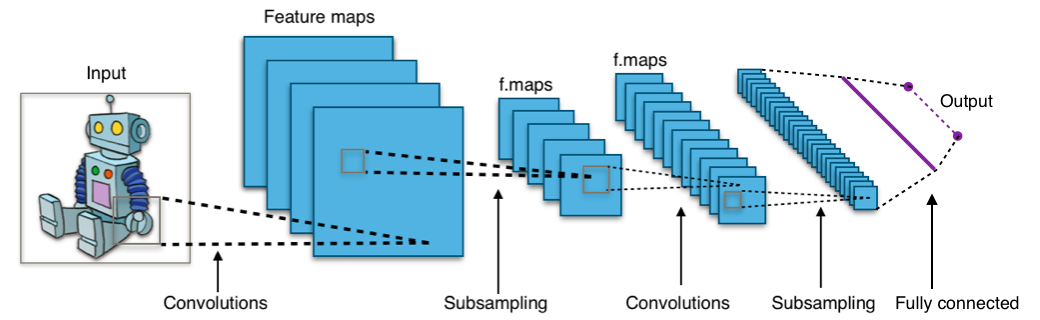
Отличия сверточных сетей заключаются в чередовании сверточных слоев (convolution layers) и слоев субдискретизации (subsampling layers или pooling layers, слоёв подвыборки). Соответственно, название сети получено, благодаря наличию в ней операций свертки. Суть этих операций в том, что к каждому блоку пикселей изображения, то есть некоторому его фрагменту, применяется сверточное ядро. Чаще всего структура сети является однонаправленной, без обратных нейронных связей, к тому же принципиально имеет многослойную архитектуру. Для обучения в основном используются методы обратного распространения ошибки (back-propagation, [1]).

Слой свёртки – это основной блок сверточной нейронной сети. Для каждого канала изображения слой включает в себя свой фильтр, который обрабатывает выход предыдущего слоя по фрагментам. Весовые коэффициенты ядер свертки устанавливаются в процессе обучения сети.

Слой пулинга – по сути операция уменьшения размерности фрагментов различными методами. Пулинг предназначен для отсечения ненужной информации, то есть если на предыдущих операциях свертки были уже выявлены некоторые признаки, то для дальнейшей обработки нет необходимости в столь подробном изображении, поэтому оно сжимается до менее детального.

Рассмотрим более подробно типовую структуру сверточной сети. Как уже сказано, она имеет многослойную архитектуру. После первого начального слоя, то есть входного изображения, сигнал проходит серию сверточных слоев, в который в определенной последовательности чередуются свертка и пулинг. Такое чередование позволяет получить карты признаков, причем на каждом слое карта уменьшается в размере, но увеличивается их количество. На практике это означает способности сети к распознаванию сложных иерархий признаков.

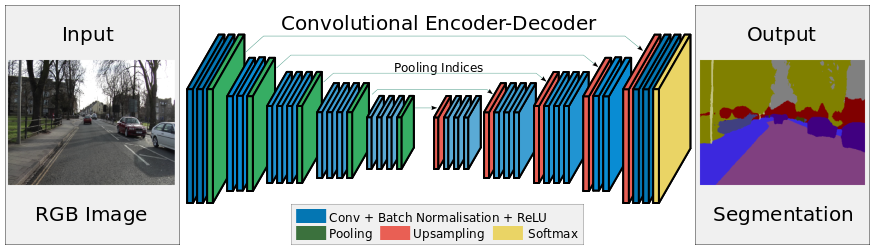
На выходе всех сверточных слоев карты признаков вырождаются в вектор или скаляр, которые далее подаются на вход полносвязным слоям сети (персептрону, [1]), которые обычно дополняют сверточные сети в конце.



1. Иллюстрация изменения карт признаков при прохождении сверточной сети [14].

На сегодняшний день существует множество алгоритмов, которые решают задачу семантической сегментации с использованием сверточных сетей. Однако, большинство таких моделей достаточно трудоемки и ресурсоемки с точки зрения использования их в системах реального времени и для обработки видеопотоков с камеры. Поэтому для ускорения работы таких методов можно применять различные подходы.

Обычно относительно глубокие сети с большим количеством слоев требует продолжительного обучения и большого объема тренировочной выборки, что накладывает некоторые ограничения на наборы изображений, которые применяются для обучения.



1. Пример архитектуры сверточной сети SegNet [15].

# Обзор наборов изображений и видео, используемых для оценки качества решений

В ходе производственной практики были рассмотрены существующие базы изображений и видео для глубоко обучения, связанные с задачами систем помощи водителю. Среди всех баз данных будет выбрана одна, которая в дальнейшем будет использоваться для решения задачи. Данные базы будут анализироваться и сравниваться по нескольким параметрам, таким как:

* Формат данных.
* Размер тренировочной выборки.
* Размер тестовой выборки.
* Непосредственное описание данных.
* Формат разметки.
* Классы детектируемых объектов.

Так же для каждой базы представлена статья, описывающая эту базу.

## Сравнение наборов данных

SUN Database

Представляет объемную базу [3] изображений разного характера: от изображений интерьера помещений до изображений гор и городов. Для конкретной задачи интерес представляют только изображения дорожных сцен.

* Тренировочная выборка: 296 (фотографии улиц) + 272 (фотографии автомагистралей).
* Тестовая выборка: 164 и 589 соответственно.
* Формат разметки: попиксельный.

Cityscapes

Объемная база [4] изображений именно дорожных сцен. Все изображения получены с камеры, установленной на лобовом стекле автомобиля в различных городах и условиях съемки.

* Тренировочная выборка: 2975 изображений с точной разметкой и 20000 с грубой.
* Тестовая выборка: 2025 изображений.
* Формат разметки: попиксельный.
* Классы сегментируемых объектов: более 30 классов. Основные группы классов: дорога, люди, транспорт, здания, объекты типа знаков, растительность, небо.

The KITTI Vision Benchmark Suite

Набор [5] изображений для детектирования дорожного полотна.

* Тренировочная выборка: 289 изображений.
* Тестовая выборка: 290 изображений.

SYNTHIA Dataset

Данный набор [6] изображений характеризуется тем, что изображения сгенерированы программной, благодаря чему выборка имеет большое количество данных, точную разметку и различные погодные и другие условия на изображениях.

* Тренировочная и тестовая выборки не разделены. 200000 изображений с видеопотока.

UIUC Image Database

База [7] содержит боковые изображения автомобилей для их детектирования на изображениях. В базе содержатся изображения обучающей выборки как одинакового масштаба, так и различного.

* Тренировочная выборка: 550 изображений с автомобилями, 500 изображений, не содержащих автомобили.
* Тестовая выборка: 170 изображений с автомобилями того же масштаба, что и в тренировочной выборке, 200 изображений с автомобилями разного масштаба.
* Формат разметки: координата верхнего левого угла окаймляющего прямоугольника.

The German Traffic Sign Detection Benchmark

Дорожные сцены для детектирования знаков и их классификации. База [8] содержит 42 класса различных знаков. Наибольшее количество знаков – запрещающих и ограничивающих, что подчеркивает их наибольшее значение для проектирования систем помощи водителю.

* Тренировочная выборка: 900 изображений знаков.
* Тестовая выборка: 301 изображение.
* Формат разметки: Координаты окаймляющего прямоугольника и идентификатор класса

CamVid Database

База [9] содержит видео с дорожными сценами разных условий освещенности, из которых взяты кадры с периодом 1с и размечены. Видео сняты на камеру из салона автомобиля.

* Тренировочная выборка: 701 изображение.
* Тестовая выборка: 5 видеороликов 30Hz в сумме на 10 минут.
* Формат разметки: попиксельная.
* Классы сегментируемых объектов: перемещающиеся объекты (животные, пешеходы, дети, коляска/тележка, велосипедист, мотоцикл/скутер, автомобиль, внедорожник, грузовик, поезд), дорога (проезжая часть, дорожная разметка, обочина), небо, статичные объекты (здания, стены, дерево, забор, столбы, мосты, знаки, светофор и др.).

CBCL StreetScenes Challenge Framework

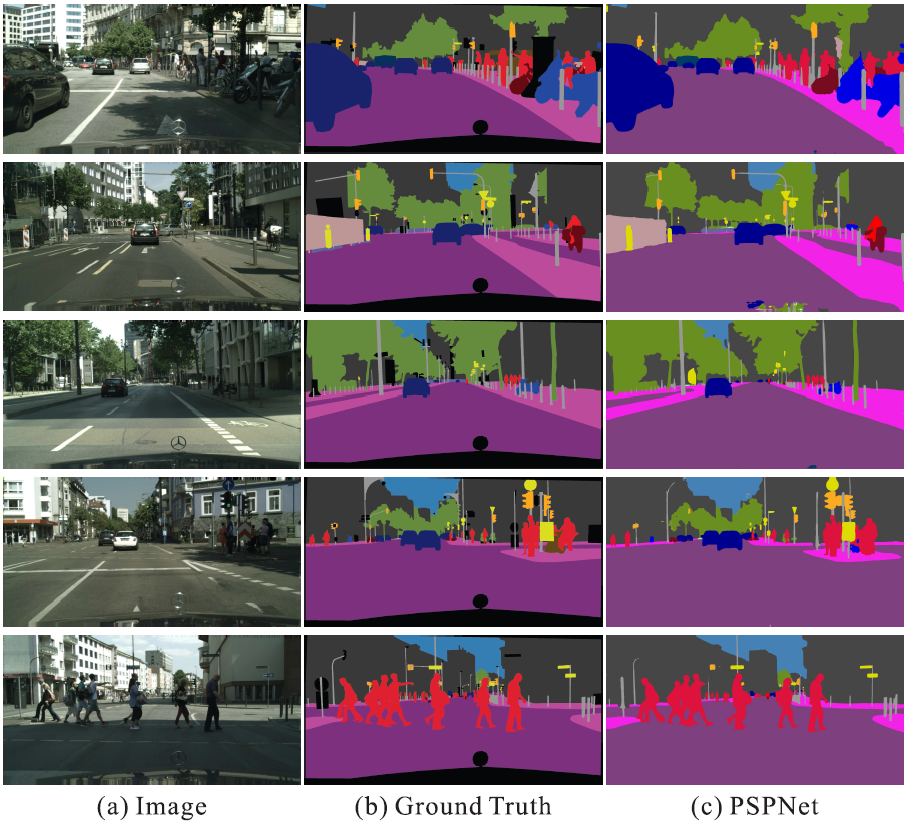
База [10] может использоваться как для детектирования объектов на изображении, так и для семантической сегментации.

* Тренировочная выборка: 3547 изображений.
* Тестовая выборка: отсутствует.
* Формат разметки: точки окаймляющей ломаной.
* Классы сегментируемых объектов: автомобили, пешеходы, велосипеды, здания, деревья, небо, дороги, тротуары и магазины

## Выбор тренировочного и тестового набора

Среди всех наборов изображений и видео для решения задачи семантической сегментации, возникающей при создании систем помощи водителю, наиболее полно удовлетворяет условиях база Cityscapes.

Во-первых, объем тренировочной выборки в данной базе достаточно велик для качественного обучения алгоритма семантической сегментации. Во-вторых, изображения получены с камеры, которая установлена на борту автомобиля, что наиболее приближено к монтированию оборудования в системах помощи водителю. В-третьих, количество классов объектов достаточно велико для получения подробного описания окружающей обстановки.



1. Пример работы сети PSPNet на изображениях из базы Cityscapes [16].

# Библиотеки глубокого обучения

В настоящее время область машинного обучения высоко востребована как в научной сфере, так и в производственной. Поэтому на сегодняшний день существует немалое количество библиотек и фреймворков, нацеленных на машинное обучение. Некоторые из них будут рассмотрены в данном разделе.

## Caffe

Один из наиболее популярных инструментов, которые используется сегодня для решения задач распознавания образов (визуальных и аудио-образов, в том числе речи) – разработка Центра распознаваний образов и машинного обучения университета Беркли, США (Berkeley Visionand Learning Center) и связанных с ним сторонних исследователей под названием Caffe [11].

Библиотека Caffe – фреймворк для глубокого обучения, является open-source С++ проектом, при этом продукт полностью поддерживает написание пользовательским алгоритмов на Python и NumPy, а также совместим с MATLAB. Caffe предлагает широкий инструментарий для создания и применения современных алгоритмов глубокого обучения.

Достоинства библиотеки Caffe:

1. «Чистая» архитектура, которая позволяет осуществить мгновенное развертывание. Сети определяются простыми файлами конфигурации, благодаря чему переключение между центральным и графическим процессором происходит легко. Для ускорения вычислений Caffe может быть запущена на GPU с использованием базовых возможностей технологии CUDA или библиотеки примитивов глубокого обучения cuDNN.
2. Открытый код позволяет не только контролировать внедрение, но и модифицировать его под свои нужды. В первые 6 месяцев активного использования Caffe более 300 независимых разработчиков внесли свой вклад в развитие библиотеки, просто подстраивая ее под себя.
3. На данный момент Caffe лидирует по скорости обучения среди многих библиотек, реализующих сверточные сети.
4. Еще одно преимущество Caffe – наличие уже готовых свободно распространяемых моделей обучения, полностью интегрированных в систему.

Официально поддерживаемые операционные системы — Linux и Mac OS X, также имеется неофициальный порт на Windows.

## Torch

Torch [12] – библиотека для научных вычислений с широкой поддержкой алгоритмов машинного обучения. Разрабатывается Idiap Research Institute, New York University и NEC Laboratories America, начиная с 2000г., распространяется под лицензией BSD.

Библиотека реализована на языке Lua с использованием C и CUDA. Быстрый скриптовый язык Lua в совокупности с технологиями SSE, OpenMP, CUDA позволяют Torch показывать неплохую скорость по сравнению с другими библиотеками. На данный момент поддерживаются операционные системы Linux, FreeBSD, Mac OS X. Основные модули также работают и на Windows. В зависимостях Torch находятся пакеты imagemagick, gnuplot, nodejs, npm и другие.

Библиотека состоит из набора модулей, каждый из которых отвечает за различные стадии работы с нейросетями. Так, например, модуль nn обеспечивает конфигурирование нейросети (определению слоев, и их параметров), модуль optim содержит реализации различных методов оптимизации, применяемых для обучения, а gnuplot предоставляет возможность визуализации данных (построение графиков, показ изображений и т.д.). Установка дополнительных модулей позволяет расширить функционал библиотеки.

Torch позволяет создавать сложные нейросети с помощью механизма контейнеров. Контейнер — это класс, объединяющий объявленные компоненты нейросети в одну общую конфигурацию, которая в дальнейшем может быть передана в процедуру обучения. Компонентом нейросети могут быть не только полносвязные или сверточные слои, но и функции активации ошибки, а также готовые контейнеры.

## Tensorflow

Tensorflow [13] является библиотекой машинного обучения с открытым исходным кодом для решения различных задач, связанных с обучением. Разрабатывается компанией Google с 2015 года. Библиотека представляет APIинтерфейс для Python, а также С/С++, Haskell, Java, Go, Rust. С недавних пор поддерживается интерфейс языка R. Сама библиотека реализована на Python, C++ и CUDA. Поддерживаются платформы Linux, Windows, macOS и Android.

Особенность библиотеки заключается в том, что для представлений вычислений в ней используется графовое представление. Узлы графа представляют собой математические операции, в то время как ребра графа представляют многомерные массивы данных (тензоров), которые взаимодействуют между собой посредством этих операций. Гибкая архитектура Tensorflow позволяет развернуть вычисления на одном или несколько центральных процессорах или графических процессоров в компьютере, на сервере или мобильном устройстве в единый API.

В библиотеке активно добавляются и развиваются различные модули за счет открытости кода и активной поддержки со стороны Google.

* RankBrain – модуль для разработки интеллектуальных систем, работающих с поисковыми запросами в Интернете.
* Inception Image Classification Model – модуль распознавания образов.
* SmartReply – модуль для систем автоматического ответа на email.
* Massively Multitask Networks for Drug Discovery – модуль для исследований в области медицины.
* On-Device Computer Vision for OCR – модуль для распознавания текста с фотографии и его перевода на другие языки.

# Программная реализация классификатора

В целях знакомства с одной из библиотек глубокого обучения в процессе данной работы была поставлена и решена задача по реализации простейшего классификатора. Для решения задачи была выбрана библиотека Tensorflow как наиболее доступная на платформе Windows. Исходный код программы располагается в репозитории на GitHub [1].

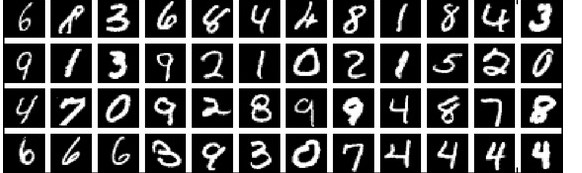
## Используемые инструменты

В ходе работы использовались следующие инструменты:

* Система контроля версий Git.
* Библиотека глубокого обучения Tensorflow.
* Jupyter – веб-оболочка для программирования на языке Python.

## Данные

В качестве решаемой задачи была выбрана одна из классических задач машинного обучения: детектирование рукописных цифр, представленных черно-белыми изображениями 28х28 пикселей. Данные взяты из базы данных MNIST [2].



1. Примеры изображений из базы данных MNIST [17].

Зная размеры каждого изображения, выделим наипростейшие признаки из них – пиксели. Таким образом, необходимо построить классификатор, у которого 784 входа и 10 выходов, где каждый выход соответствует цифре, которую сеть выдает в качестве ответа.

Обучающая выборка состоит из 60000 изображений, представленных в виде строки, где первое число – правильный ответ, следующие за ним 784 числа – значения пикселов изображения.

## Методы решения задачи

Для решения задачи были выбраны два различных метода: полносвязная нейронная сеть и метод логистической регрессии. Использовались реализации методов из библиотеки Tensorflow. Полносвязные нейронные сети подробно описаны в [1].

Логистическая регрессия – метод построения линейного классификатора, позволяющий оценивать апостериорные вероятности принадлежности объектов классам.

Пусть объекты описываются числовыми признаками , тогда пространство признаковых описаний есть . Пусть – конечное множество номеров классов. Пусть задана обучающая выборка .

В простом случае, когда вся выборка разделяется на 2 класса, рассуждения можно провести следующим образом: положим Построим линейный алгоритм :

где – вес -го признака, – порог принятия решения, – вектор весов.

Задача обучения в логистической регрессии сводится к задаче минимизации эмпирического риска с функцией потерь специального вида:

## Вычислительные результаты

Нейронная сеть

Архитектура нейронной сети представляла собой входной, выходной и 1 скрытый слой из 256 нейронов. Скорость обучения 0.01, количество эпох обучения 15. В качестве функции ошибки выбрана кросс-энтропия. Результат – 94,28% правильных откликов сети.

Логистическая регрессия

Скорость обучения 0.01, количество эпох обучения 15. В качестве функции ошибки выбрана кросс-энтропия. Результат – 90,79% правильных ответов.

# Заключение

В ходе данной работы были выполнены все поставленные задачи, а именно, рассмотрены задачи компьютерного зрения, возникающие при разработке автономных систем помощи водителю в частности и общие задачи компьютерного зрения в целом, изучена актуальность данной темы. Была выбрана задача семантической сегментации изображений для дальнейшего решения. Так же были рассмотрены известные методы решения данной задачи, проанализированы достоинства и недостатки каждого метода. Для решения поставленной проблемы рассмотрены несколько наборов изображений, среди которых был выбран наиболее подходящий, а именно база изображений Cityscapes.

Для дальнейшего решения задачи были рассмотрены несколько библиотек машинного обучения и области их применения. Среди них была выбрана одна для тестовых запусков на простейшей задаче классификации. Реализованы алгоритмы с использованием методов, предлагаемых библиотекой Tensorflow. Проведены вычислительные эксперименты.

# Литература

1. Воронцов К. В. Математические методы обучения по прецедентам (теория обучения машин). — 141 с.
2. Хайкин С. Нейронные сети: Полный курс. Пер. с англ. Н. Н. Куссуль, А. Ю. Шелестова. 2-е изд., испр. — М.: Издательский дом Вильямс, 2008, 1103 с.
3. Laurene V. Fausett. Fundamentals of Neural Networks: Architectures, Algorithms And Applications: 1st (first) Edition Paperback — December 9, 1994 — 461 с.
4. Кафтанников И.Л. Особенности применения деревьев решений в задачах классификации / И.Л. Кафтанников, А.В. Парасич // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника». - 2015. - Т. 15, № 3. - С. 26-32. DOI: 10.14529Мсг150304
5. Р.В. Шаповалов, Д.П. Ветров, А.А. Осокин, П. Коли Обучение алгоритма семантической сегментации изображений на выборке с разнообразными типами аннотаций.
6. Григорьев Д.С. Обзор алгоритмов семантической сегментации / Д. С. Григорьев, В. Г. Спицын // ХІІІ Всероссийская научно-практическая конференция «Технологии Майкрософт в теории и практике программирования». – 2015. - С. 108-110.
7. Лекция компании Яндекс по нейронным сетям на портале Хабрахабр [<https://habrahabr.ru/company/yandex/blog/307260/>].
8. Cortes C., Vapnik V. Mach Learn (1995) 20: 273 Machine Learning September 1995, Vol. 20, Iss. 3, С. 273–297.

# Ссылки

1. Репозиторий, содержащий разработанную программную реализацию и отчет по учебной практике предыдущего семестра [<https://github.com/Maxim-Doronin/NeuralNetwork>].
2. Данные, использованные для обучения [<https://www.kaggle.com/c/digit-recognizer/data>].
3. Статья, описывающая базу SUN Database [<http://vision.princeton.edu/projects/2010/SUN/paper.pdf>].
4. Статья, описывающая базу Cityscapes [<https://arxiv.org/pdf/1604.01685.pdf>].
5. Статья, описывающая базу The KITTI Vision Benchmark Suite [<http://www.cvlibs.net/publications/Geiger2013IJRR.pdf>].
6. Статья, описывающая базу The SYNTHIA Dataset [<http://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Ros_The_SYNTHIA_Dataset_CVPR_2016_paper.pdf>].
7. Статья, описывающая базу UIUC Image Database [<http://cogcomp.cs.illinois.edu/papers/AgarwalAwRo04.pdf>].
8. Статья, описывающая базу The German Traffic Sign Detection Benchmark [[http://geza.kzoo.edu/~erdi/IJCNN2013/HTMLFiles/PDFs/P100-1288.pdf].](http://geza.kzoo.edu/~erdi/IJCNN2013/HTMLFiles/PDFs/P100-1288.pdf%5d.)
9. Статья, описывающая базу CamVid Database [<http://www0.cs.ucl.ac.uk/staff/g.brostow/papers/Brostow_2009-PRL.pdf>].
10. Статья, описывающая базу CBCL StreetScenes Challenge Framework [<http://cbcl.mit.edu/projects/cbcl/publications/theses/thesis-bileschi.pdf>].
11. Библиотека Caffe [<http://caffe.berkeleyvision.org>].
12. Библиотека Torch [<http://torch.ch>].
13. Библиотека Tensorflow [<https://www.tensorflow.org/>].
14. Ссылка на источник, содержащий изображение, иллюстрирующий работу сверточной сети [<https://ru.wikipedia.org/wiki/Свёрточная_нейронная_сеть>].
15. Ссылка на источник, содержащий изображение архитектуры сверточной сети [<http://mi.eng.cam.ac.uk/projects/segnet/>].
16. Ссылка на источник, содержащий изображение примера работы сети PSPNet на изображениях из базы Cityscapes [<https://hszhao.github.io/projects/pspnet/>].
17. Ресурс, содержащий изображение данных обучающей выборки [[https://www.researchgate.net/figure/264203784\_fig3\_Samples-from-the-MNIST-database].](https://www.researchgate.net/figure/264203784_fig3_Samples-from-the-MNIST-database%5d.)