МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Кафедра информационных технологий

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

И. о. заведующего кафедрой
канд. физ.-мат. наук
_____ В.В. Подколзин
(подпись)
_____ 2018 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

ПОДСИСТЕМА МОДЕЛИРОВАНИЯ ДВИЖЕНИЯ ТРАНСПОРТА ПО ДОРОГЕ С ПРЕПЯТСТВИЯМИ

Работу выполнила ___А.И. Аликумова
(подпись, дата)

Факультет компьютерных технологий и прикладной математики

Направление 01.03.02 – «Прикладная математика и информатика»

Научный руководитель
доц., канд. тех. наук, доц. _____ А.А. Полупанов
(подпись, дата)

Нормоконтролер ст. преп. _____ А.В. Харченко
(подпись, дата)

Краснодар 2018

РЕФЕРАТ

Выпускная квалификационная работа 55 с., 3 ч., 31 рис., 15 источников.

СЕГМЕНТАЦИЯ ДОРОГИ, КОМПЬЮТЕРНОЕ ЗРЕНИЕ, ОБРАБОТКА ИЗОБРАЖЕНИЯ, АЛГОРИТМЫ, МЕТОДЫ, ДЕТЕКТОРЫ.

Объектом исследования являются системы детектирования, базирующиеся на алгоритмах компьютерного зрения.

Цель работы - разработка программной подсистемы моделирования движения транспорта по дороге с препятствиями.

В процессе работы проводился анализ основных алгоритмов компьютерного зрения, использующихся для сегментации и распознавания образов на изображениях или видеоряде.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

На протяжении всей своей истории человечество старалось облегчить и автоматизировать свои действия. Начиная с изобретения колеса и заканчивая механическими устройствами, способными поднимать грузы, во много раз превосходящие вес человека. Это является актуальной задачей сейчас точно так же, как и сотни лет назад. Человек во всем старается сделать свою жизнь более комфортной и безопасной, включая собственные дома и средства передвижения. Поэтому желание пользоваться автомобилем, в котором не будет требоваться постоянного управления со стороны человека, не кажется чем-то выходящем на границы разумного.

Автоматическое управление на реальных дорогах – это давнее стремление не только водителей, но и всей автомобильной промышленности в целом. Это имеет большое значение для всего мира, ведь когда этот день наступит, на дорогах станет намного безопаснее. На сегодняшний день большинство происшествий, которые случаются на дороге, связаны в той или иной степени с тем, что водитель отвлекся или не успел среагировать в сложной дорожной ситуации (оживленное движение, плохие погодные условия, непредвиденные ситуации, недостаточно опыта).

Будущее беспилотного управления неизбежно потребует всестороннего и надежного восприятия дороги и дорожного окружения. Эта задача может быть очень сложной из-за огромного количества вариантов предметов и условий (дорожное полотно, транспортные средства, различные препятствия, освещение, погода и т.д.). Дорожное полотно и другие транспортные средства являются наиболее актуальными темами в данной задаче, поэтому большое исследований посвящено методам обработки подобного рода информации. Это так, поскольку невозможно разрабатывать какую-либо систему без учета контекста условий, в которой она будет применятся.

Компьютерное зрение – это область информатики, которая учит компьютеры видеть. Это способ, с помощью которого компьютеры собирают

и интерпретируют визуальную информацию, получаемую из окружающей среды. Однако то, что человек делает мгновенно и часто бессознательно, для компьютера является достаточно сложной задачей.

В каждой сфере нашей деятельности присутствует различная фото и видеоаппаратура, начиная от камер мобильных телефонов и заканчивая спутниковыми камерами. С их помощью мы получаем большое количество данных, внедряем различные системы видеонаблюдения и мониторинга, медицинского зрения и идентификации личности и другие, которые призваны максимально автоматизировать различные процессы производства и контроля.

Цифровая информация весьма разнородна не только по качеству, которое зависит от оборудования, но и по содержанию, поэтому не существует универсального подхода для решения всех задач, связанных с фото или видеообработкой. Поэтому процесс разработки систем обработки и анализа цифровых изображений является достаточно трудным. В каждом случае появляются уникальные особенности и подзадачи, которые требуют особого внимания и подхода.

Именно большое количество данных, практическая польза и нетривиальность задач положило начало такой научной дисциплине, как компьютерное зрение, которая позволяет решать достаточно широкий спектр задач, затрачивая относительно небольшое количество ресурсов.

1 Анализ систем компьютерного зрения

Как научная дисциплина, компьютерное зрение относится к созданию искусственных систем, способных к обнаружению, отслеживанию и классификации объектов. Оно преследует двойную цель. С точки зрения биологии, его цель заключается в создании вычислительной модели зрительной системы человека. С технической точки зрения, компьютерное зрение нацелено на создание автономных систем, способных выполнять поставленные задачи на уровне близком к человеческому восприятию. Эти два подхода тесно связаны между собой. С одной стороны, свойства и характеристики человеческой визуальной системы часто помогают инженерам, которые разрабатывают системы компьютерного зрения. С другой стороны, алгоритмы компьютерного зрения дают более глубокое понимание работы нашей зрительной системы.

1.1 История развития

В конце 60-х годов прошлого века компьютерное зрение зародилось в университетах, которые были первопроходцами в сфере искусственного интеллекта. Оно было предназначено для имитации зрительной системы человека и являлось трамплином для создания роботов с интеллектуальным поведением. В 1966 считалось, что этого можно достичь с помощью присоединения камеры к компьютеру и описания того, что она видит.

Компьютерное зрение отличает от преобладающей области цифровой обработки изображений желание извлечь трехмерную структуру из изображения с целью достижения полного понимания картины. Исследования 1970 года заложили фундамент для многих алгоритмов компьютерного зрения, существующих на сегодняшний день, включая извлечение краев из изображений, маркировки линий, без многогранного и

многогранного моделирования, представление объектов как взаимосвязей небольших структур, оптического потока, и оценка движения.

В следующее десятилетие исследования базировались на более строгом математическом анализе и количественных аспектах компьютерного зрения. Они включили в себя понятие масштаба пространства, получение формы объекта с помощью различных «подсказок» таких, как затемнение, текстуры и фокус, и контурные модели, известные как змеи. Исследователи также осознали, что большинство из этих математических понятий можно рассматривать в пределах таких же оптимизационных рамок, как регуляризация и Марковское случайное поле (MRF).

К 1990-ым годам некоторые из предыдущих тем исследований стали развиваться активнее остальных. Исследования в области проектирования 3D реконструкций привело к более глубокому пониманию о калибровке камеры. С появлением методов оптимизации для калибровки камеры, стало ясно, что многие идеи уже были изучены в теории авто-калибровки из области фотограмметрии. Это привело к возможности создания 3D реконструкции картины по нескольким изображениям. Прогресс был достигнут в соответствующей технологии стерео-зрения. В то же время были использованы вариации разрезов графа для решения проблемы сегментации изображения. Это десятилетие ознаменовалось первым применением статистических методов обучения в задаче распознавания лиц на изображении. К концу 1990-ых годов произошли значительные изменения из-за увеличения взаимодействия между компьютерной графикой и компьютерным зрением, которые включали изображения на основе рендеринга, морфинг изображения, панорамные изображения и т.д.

Последние работы рассматривают возрождение методов, использующих машинное обучение и сложных механизмов оптимизации.

1.2 Решаемые задачи

Применение компьютеров для обработки цифровых данных достаточно велико. К основным задачам компьютерного зрения относят:

- распознавание объектов, текстов;
- идентификация лиц, биометрия;
- система автоматического управления машинами;
- контроль медицинских изображений;
- системы видеонаблюдения.

К последнему пункту относится довольно широкий спектр задач. Видеокамеры предоставляют возможность осуществлять:

- автоматический контроль воздушной обстановки;
- контроль доступа на охраняемые объекты;
- мониторинг мест большого скопления людей (торговых центров, аэропортов, банков и так далее);
- анализ дорожного полотна.

Естественно, это далеко не полный список проблем, которые могут быть решены с помощью компьютерного зрения. Несмотря на столь широкий спектр применений, успех системы компьютерного зрения достигается весьма непросто. В большинстве случаев разработчикам приходится накладывать ряд ограничений, связанных с окружающей средой. К примеру, может появиться необходимость в искусственном освещении, механической изоляции или позиционирование объекта. Поскольку мы находимся в постоянно изменяющемся мире, то возникает необходимость во все более совершенных компьютерных алгоритмах для вычисления инвариантных признаков объектов. Главной проблемой для систем компьютерного зрения является изменчивость объектов: вид в зависимости от освещенности сцены, присутствия других объектов, большая изменчивость внутри одного класса и так далее. Загораживание объектов является серьезной проблемой для

распознавания трехмерных объектов, так как происходит их деформация на изображении [1].

1.3 Подходы к решению задач

Главные шаги, которые используются в процессе извлечения знаний по изображению, заключаются в получении, обработке, анализе изображения и преобразование его в числовую или символьную форму. Данный подход используется в качестве аналога человеческого зрения.

Изображение при обработке трансформируется в зависимости от проводимых операций, таких как повышение контрастности, выделение края, удаление шума или геометрического преобразования. На этапе обработки содержание изображения никак не интерпретируется, это происходит лишь во время анализа полученной информации.

Существует множество методов, с помощью которых это можно осуществить. Некоторые их них базируются на выделении инвариантных признаков, которые будут давать об объекте достаточно полную информацию, или очертания объектов и последующего исследования на наличие углов, связности и т.д. Другие используют искусственные нейронные сети для распознавания образов. Также существуют методы, выделяющие на изображении такие признаки как цветовую гистограмму, направление и величину градиентов, и затем сравнивают новые данные с уже полученными данными.

1.4 Постановка задачи

Целью данной выпускной работы является исследование возможных способов сегментации дорожного полотна, основных алгоритмов, позволяющих осуществить выделение полосы движения, определение препятствий и анализ возможных путей движения. Рассмотрение способов

нормализации, предобработки, а также классификации данных. Изучение таких методов как: преобразование Хафа, оператор Кэнни, гистограмма направленных градиентов и SVM-классификатор, а также реализация подсистемы моделирования движения транспорта, применяя изученные алгоритмы, и разработка графического интерфейса. Выявление преимуществ и недостатков данных подходов, а также исследование особенностей работы при различных входных данных и рассмотрение предпочтительных условий использования для каждого их них.

2 Применение методов компьютерного зрения

Существуют различные методы компьютерного зрения, которые позволяют решать задачи сегментирования дорожного полотна и поиска препятствий. Далее рассмотрены алгоритмы, применяемые для программной реализации.

2.1 Гистограмма направленных градиентов

Гистограмма направленных градиентов (HOG) – это способ обнаружения точек, в которых изображение имеет особенности, для поиска необходимого объекта. Этот метод предполагает подсчет количества вхождений направленных градиентов в рассматриваемых областях на изображении. Особенностью данного метода является то, что вычисления проводятся на плотной сетке, с равномерно расположенными ячейками и для повышения точности используют нормализацию перекрывающегося контраста.

Алгоритм был впервые описан Биллом Триггсом и Навнитом Далалом в 2005 году. Изначально они использовали его для распознавания пешеходов на статическом изображении, немного позже в видеоряде, а затем усовершенствовали для обнаружения других объектов [2].

Дескриптор особенностей является таким представлением изображения, которое упрощает его, то есть извлекает полезную информацию, игнорируя непригодную для дальнейшего использования. В большинстве случаев дескриптор преобразует картинку, которая имеет свои размеры (высоту, ширину, количество каналов), в вектор или массив особенностей, имеющий определенную длину N. К примеру, если размер входного изображения составляет 64 x 128 x 3, то полученный вектор особенностей будет иметь длину 3780. Следует обговорить, какой смысл вкладывается в слово «полезный» и для каких целей оно является таковым.

Очевидно, что дескриптор особенностей будет бесполезен для простого просмотра изображения, однако он оказывается невероятно полезным в задаче обнаружения и распознавания объектов.

В дескрипторе особенностей HOG в качестве признаков используется распределение направления градиентов. Градиент, вычисленный в произвольной точке изображения, является информативной характеристикой, поскольку его величина велика в области краев и углов, как известно, участки, где происходит резкое изменение интенсивности цвета предоставляют намного больше информации по сравнению в однотонными участками.

После того, как на вход было передано изображение, алгоритм выполняет следующие этапы:

- гамма-коррекция и нормализация цвета;
- вычисление градиентов;
- группировка направлений;
- вычисление гистограммы градиентов;
- нормализация;
- svm-классификатор.

После этого мы получаем ответ на вопрос, что находится на изображении [4].

2.1.1 Вычисление градиентов

Авторы выяснили, что в данном алгоритме первый шаг, то есть гамма-коррекцию и нормализацию цвета, можно пропустить, потому что нормировка, которая будет проводиться в дальнейшем, даст тот же результат. Поэтому сразу переходят к вычислению градиентов.

Для того чтобы в итоге получить гистограмму градиентов, необходимо сначала рассчитать горизонтальные и вертикальные градиенты. Самым

распространенным и простым способом является фильтрация изображения с использованием следующих ядер, представленных формулой (1):

$$[0 \quad 1 \quad 0] \text{ и } \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \tag{1}$$

Дескриптор довольно чувствителен к способу вычисления градиента, однако данный способ оказался наиболее эффективным. Триггс и Далал также пробовали использовать другие методы нахождения градиента яркости на изображении, например, оператор Собеля, но отказались от таких подходов из-за более низкой производительности.

Градиент изображения позволяет отсеять много несущественной информации, например, однотонный цветной фон, но оставляет для дальнейшего изучения выделенные контуры.

2.1.2 Группировка направлений

Далее необходимо вычислить гистограммы ячеек. Все пиксели в ячейки принимают участие во взвешенном голосовании для каналов ориентированных гистограмм, которые базируются на значении градиентов. Форма ячеек может быть круглой и прямоугольной, а в зависимости от того, какой градиент вычисляется «знаковый» или «беззнаковый», каналы гистограммы могут равномерно распределяться от 0 до 180 или же от 0 до 360 градусов. Вес пикселя при взвешенном голосовании может задаваться абсолютным значением градиента или какой-то функцией, зависящей от нее; на реальных тестах лучшие результаты показало абсолютное значение градиента.

2.1.3 Вычисление гистограммы градиентов

На этом этапе, изображение делится на клетки размером 8 × 8 и затем вычисляется гистограмма градиентов. Одной из важнейших причин, по которой необходимо использовать дескриптор для описания части изображения, является то, что он позволяет компактно представить информацию о нем. Изначально информация о данном участке хранится в 192 значений пикселей (8×8×3), а градиент той же области содержит уже 128 чисел (8×8×2 - последний параметр хранит информацию о величине и направлении градиента). Затем эти числа преобразуются в 9-битную гистограмму, которая может храниться в виде массива, состоящего из девяти цифр. Кроме компактности хранимой информации, такое представление делает ее более устойчивой к шуму. Гистограммы, по существу, это вектор или массив из 9 элементов, соответствующий углам 0, 20, 40, … 160. На рисунке 1 представлен процесс создания гистограммы градиентов.

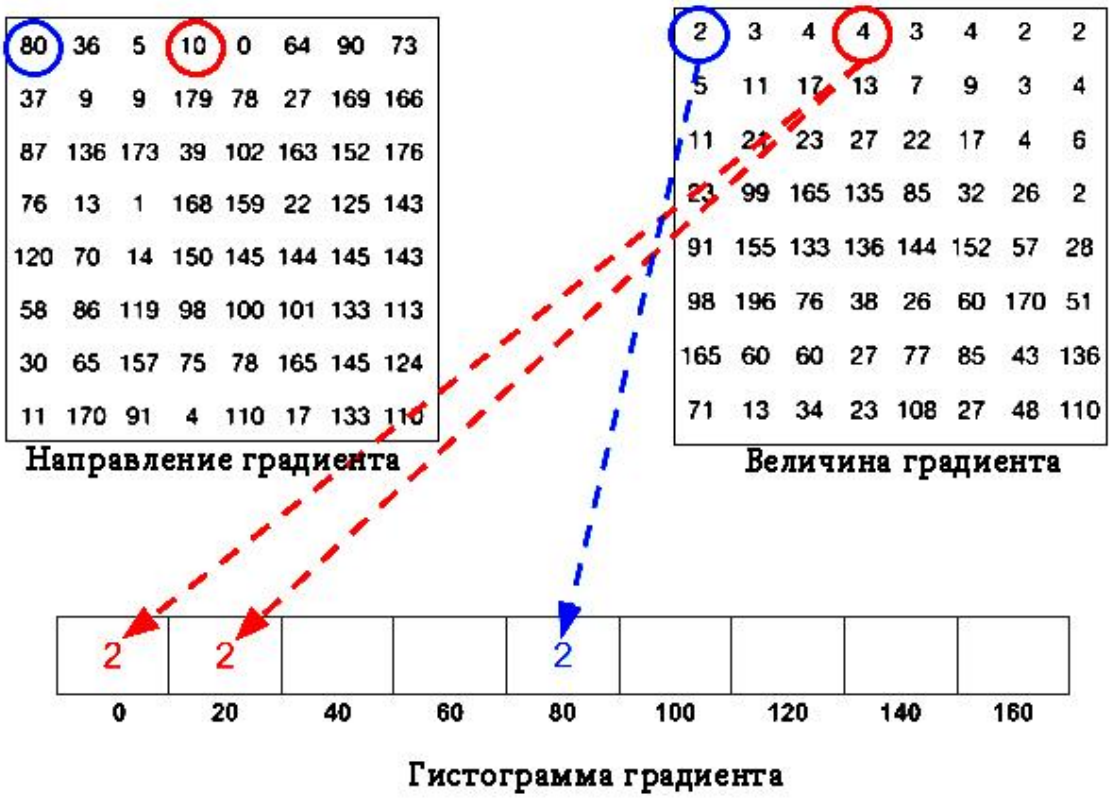

Рисунок 1 – Создание гистограммы градиентов

Рассматриваются величина и направление градиента в рассматриваемой области. Ячейка в массиве выбирается в зависимости от направления, а голос (значение, которое добавляется в ячейку) выбирается на основе значения величины. Если рассмотрим левый верхний пиксель, то увидим, что его голос, равный 2, запишется в пятую ячейку массива. А пиксель правее него имеет направление в 10 градусов и величину 4. Поскольку 10 градусов лежит на полпути между 0 и 20, голосование пикселя равномерно распределяется между первой и второй ячейкой. Однако у данного процесса есть особенность, которая представлена на рисунке 2.

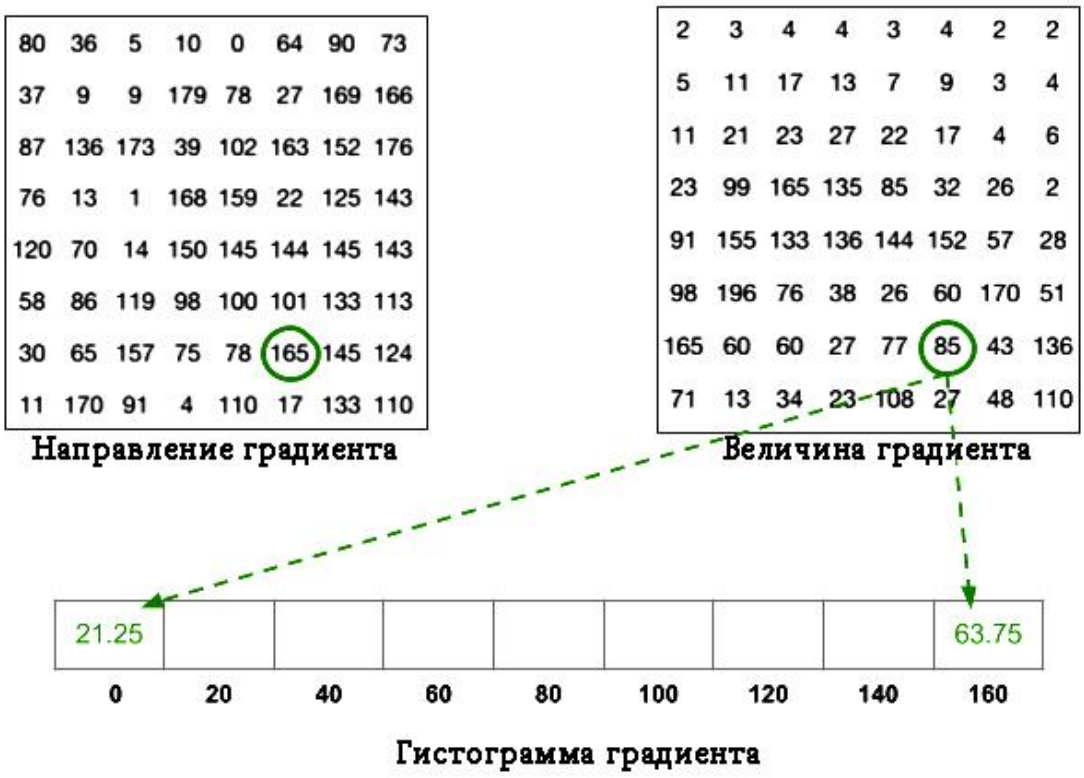

Рисунок 2 – Создание гистограммы градиентов

Если угол больше чем 160, то есть находится в интервале от 160 до 180, то используется соображение, что 0 и 180 градусов эквивалентны. Таким образом, в приведенном примере, пиксель, с углом равным 165 градусов, пропорционально распределяет значение между первой и последней ячейкой

массива. Вклады всех пикселей в рассматриваемой области суммируются, и строится 9-разрядная гистограмма, приведенная на рисунке 3.

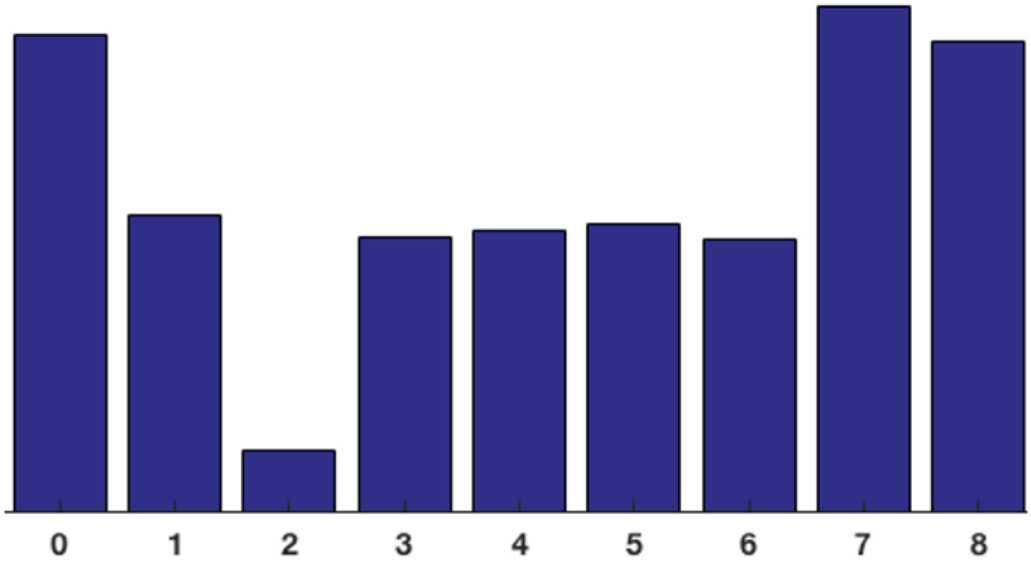

Рисунок 3 – Гистограмма градиентов

2.1.4 Нормализация

Градиенты изображения довольно чувствительны к освещению. Если сделать изображение более темным, с помощью разделения всех пикселей на 2, величина градиента, следовательно, и значение гистограммы, также изменятся в два раза. Чтобы свести к минимуму влияние освещения необходимо нормализовать гистограммы для каждого из блоков.

Есть несколько методов, которые можно использовать для нормализации блоков. Предположим, что v это ненормированный вектор, который содержит все гистограммы из рассматриваемого блока. Введем норму этого вектора: $||v||_n$, где $n = 1,2$ и e - малая величина, значение которой не влияет на конечный результат. Тогда нормировочный коэффициент можно вычислить с помощью одной из следующих формул (2) – (4):

$$L2 - norm: f = \frac{v}{\sqrt{||v||_2^2 + e^2}} \quad (2)$$

$$L1 - norm: f = \frac{v}{||v||_1 + e} \quad (3)$$

$$L1 - sqrt: f = \sqrt{\frac{v}{||v||_1 + e}} \quad (4)$$

Было установлено, что повышения точности полученных результатов, следует пользоваться формулами (2) и (4).

2.2 SVM-классификатор

Метод опорных векторов (Support vector machine - SVM) – это алгоритм машинного обучения, который используется в задачах регрессии и классификации, однако в основном его применяют для второй задачи. Относится к методу обучения с учителем. По входному набору данных, каждый объект которого имеет метку о принадлежности к одной из двух категорий, строит модель, которая каждый новый объект будет относить либо к одной категории, либо к другой. Метод опорных векторов позволяет построить оптимальную разделяющую гиперплоскость. Под оптимальной гиперплоскостью будем понимать гиперплоскость, которая перпендикулярна кратчайшему отрезку, соединяющему выпуклые оболочки разных классов, и проходит через середину этого отрезка. Помимо линейной классификации, SVM эффективно справляется с задачей нелинейной классификации.

Если начальные данные не имеют маркировок, то задача переходит в разряд кластерного анализа, то есть обучение без учителя. В данном случае данные разбираются на кластеры по своей структуре таким образом, что объекты из одной группы очень схожи между собой, а из разных групп

значительно отличаются. Тогда новые данные относятся к одному из сформированных кластеров. Такой метод носит название метод кластеризации опорных векторов и часто применяется в промышленности, когда данные не помечены или метки стоят лишь для предварительной обработки классификатора [5].

Задача классификации является общей задачей в области машинного обучения. Предположим, что заданы точки данных, каждая из которых принадлежит к одному из двух классов, а цель заключается в том, чтобы решить, к какому классу будет принадлежать новая точка данных. В случае метода опорных векторов, объекты, они же точки, данных рассматривается как p мерный вектор, и задача состоит в том, сможем ли мы разделить точки с помощью $(p-1)$-мерной гиперплоскости. Это называется линейным классификатором. Существует много гиперплоскостей, способных классифицировать данные. Однако необходимо стремиться к наилучшему варианту, то есть к варианту, когда зазор между классами является максимальным.

В реальном мире SVM применяется для решения различных задач:

- классификация изображений;
- распознавание рукописных символов;
- гипертекстовая категоризация;
- в биологии и других науках.

Для построения оптимальной гиперплоскости, SVM использует итеративный алгоритм обучения, который применяется для минимизации функции ошибок. В соответствии с формой функции ошибок, SVM модели могут быть разделены на четыре группы [6]:

- классификация svm первого типа (c-svm классификация);
- классификация svm второго типа (nu-svm классификация);
- регрессия svm первого типа (epsilon-svm регрессия);
- регрессия svm первого типа (nu-svm регрессия).

Ниже приводится краткое описание каждой модели.

2.2.1 Классификация SVM первого типа

Для классификации SVM первого типа, обучение включает в себя минимизацию функции ошибок, представленной формулой (5):

$$\frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i, \tag{5}$$

при условии соблюдения ограничений, как в формуле (6):

$$y_i(w^T\phi(x_i) + b) \geq 1 - \xi_i, \xi_i \geq 0, i = 1, \dots, N, \tag{6}$$

где C, b – константы, w – вектор коэффициентов, ξ_i представляет собой параметры для обработки неразделимых данных, $y \in \pm 1$ представляет метки класса, а x_i – независимые переменные. Ядро ϕ используется для преобразования входных данных в пространство признаков. Следует отметить, что чем больше C, тем больше штрафуется ошибка, поэтому коэффициент выбирать с осторожностью, чтобы избежать чрезмерной подгонки.

2.2.2 Классификация SVM второго типа

Классификация SVM второго типа в отличие от классификации первого типа минимизирует функцию ошибок, представлена в формуле (7):

$$\frac{1}{2}w^T w - \nu\rho + \frac{1}{N}\sum_{i=1}^{N}\xi_i, \tag{7}$$

при условии соблюдения ограничений, как в формуле (8):

$$y_i(w^T\phi(x_i) + b) \geq \rho - \xi_i, \xi_i \geq 0, i = 1, \dots, N, \rho \geq 0. \quad (8)$$

В SVM регрессии необходимо оценить функциональную зависимость зависимой переменной y на множестве независимых переменных x. Это предполагает, что отношение между независимыми и зависимыми переменными задается детерминированной функцией с добавлением некоторого аддитивного шума, представленной формулой (9):

$$y = F(x) + noise. \quad (9)$$

Задача состоит в том, чтобы найти функциональную форму для F, которая сможет правильно предсказывать результат для новых данных, которые не были представлены ранее. Это можно достичь путем обучения модели SVM на множестве образцов, то есть обучающего множества. Данный процесс включает в себя как классификацию, так и последовательную оптимизацию функции ошибки. В зависимости от определения функции, можно рассматривать два типа модели SVM.

2.2.3 Регрессия SVM первого типа

Для регрессии SVM первого типа функция ошибки имеет следующий вид, как в формуле (10):

$$\frac{1}{2}w^T w + C\sum_{i=1}^{N}\xi_i + C\sum_{i=1}^{N}\xi_i^*, \quad (10)$$

который сводится к минимуму при условиях, обозначенных формулами (11) –(13):

$$w^T\phi(x_i) + b - y_i \leq \varepsilon + \xi_i^{\cdot}, \tag{11}$$

$$y_i - w^T\phi(x_i) - b_i \leq \varepsilon + \xi_i, \tag{12}$$

$$\xi_i, \xi_i^{\cdot} \geq 0, i = 1, \dots, N. \tag{13}$$

2.2.4 Регрессия SVM второго типа

Для регрессии SVM второго типа функция ошибки определяется по формуле (14):

$$\frac{1}{2}w^T - C(\nu\varepsilon + \frac{1}{N}\sum_{i=1}^{N}(\xi_i^{\cdot} + \xi_i)), \tag{14}$$

который сводится к минимуму при условиях, выраженных формулами (15) – (17):

$$(w^T\phi(x_i) + b) - y_i \leq \varepsilon + \ \xi_i, \tag{15}$$

$$y_i - (w^T\phi(x_i) - b_i) \leq \varepsilon + \xi_i^{\cdot}, \tag{16}$$

$$\xi_i, \xi_i^{\cdot} \geq 0, i = 1, \dots, N, \varepsilon \geq 0. \tag{17}$$

Есть несколько видов ядер, которые могут быть использованы в моделях опорных векторов, представленных в формуле (18).

$$K(X_i, X_j) = \begin{Bmatrix} X_i * X_j \text{ линейное} \\ (\gamma X_i * X_j + C)^d \text{ полиоминальное} \\ \exp\left(-\gamma|X_i - X_j|^2\right) RBF \\ \tanh(\gamma X_i * X_j + C) \text{ сигмовидное} \end{Bmatrix}, \tag{18}$$

где $K(X_i, X_j) = \phi(X_i)\phi(X_j)$, то есть функция ядра представляет собой скалярное произведение входных точек данных, отображенное в признаковом пространстве более высокой размерности с помощью трансформации ϕ. γ является регулируемым параметром некоторых функций ядра. Тип ядра RBF на сегодняшний день является наиболее популярным выбором, который используется в методе опорным векторов. Это происходит главным образом из-за его локализованных и конечных ответов на всей протяженности действительной оси x.

Также существуют и другие классификаторы, которые широко применяются для решения задач машинного обучения и классификации объектов по признакам, описывающих их.

2.2.5 Логистическая регрессия

Логистическая регрессия – это метод машинного обучения, который позволяет отнести объект к одному из двух заданных классов на основе одного или нескольких описывающих его признаков. Она была разработана в 1958 году Дэвидом Коксом. В отличие от линейной регрессии, в данном алгоритме не производится предсказание значения числовой переменной исходя из выборки исходных значений. Вместо этого, значением функции является вероятность того, что данное начальное значение принадлежит к определенному классу. Главной идеей данного способа классификации является то, что пространство исходных значений может быть разделено линейной границей (т.е. прямой) на две соответствующих классам области. В данном случае решается задача нахождения исхода события, то есть к какому из двух классов будет принадлежать исследуемый объект. Данную задачу можно сформулировать иначе, если вместо предсказания бинарной переменной будем предсказывать непрерывную переменную со значениями

на отрезке [0,1] при любых значениях независимых переменных [7]. Это достигается применением следующего регрессионного уравнения (19):

$$p = \frac{1}{1+e^{-y}}, \tag{19}$$

где p – это вероятность наступления события, e – основание натурального логарифма, y – стандартное уравнение регрессии. Зависимость, которая связывает вероятность события и величину y, показана на рисунке 4:

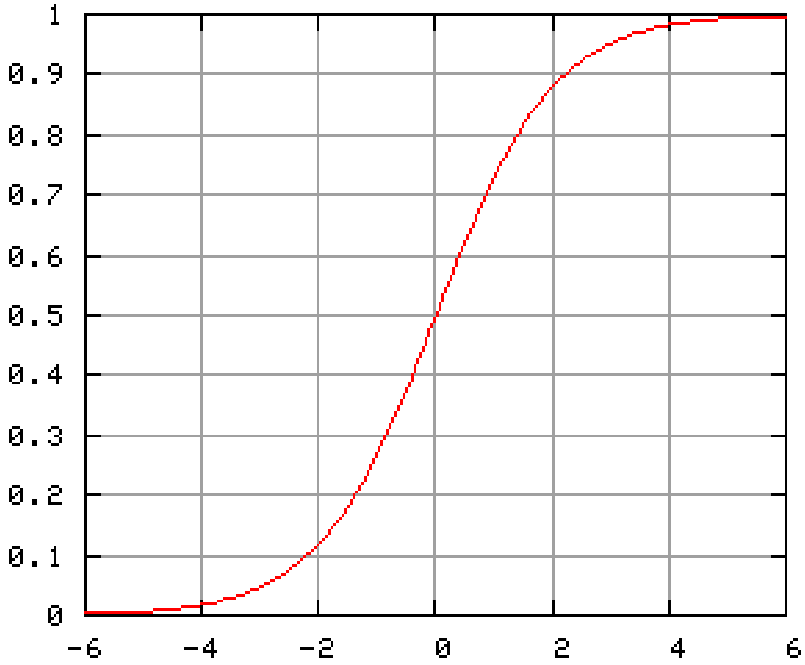

Рисунок 4 – График логистической функции

Данный метод широко применяется для исследования медицинских показателей и расчета кредитной платежеспособности. С логистической регрессией, как и с другими классификаторами, также тесно связаны разные метрики качества обучения алгоритма, такие как ROC и AUC-кривые, F-мера качества, но в рамках данной работы они рассматриваться не будут.

2.3 Оператор Кэнни

Оператор Кэнни является оператором обнаружения границ, который использует многоступенчатый алгоритм для обнаружения ребер на изображении. Он был разработан в 1986 году Джоном Ф.Канни [8].

Данный алгоритм может быть разбит на следующие пять этапов:

- сглаживание изображения для удаления шума с помощью применения фильтра гаусса;
- поиск градиентов интенсивности изображения;
- подавление не максимальных точек;
- применение двойного порога для определения потенциальных ребер.

2.3.1 Сглаживание изображения

Шум на изображении существенно влияет на результаты обнаружения ребер, поэтому необходимо его отфильтровать. Для сглаживания применяется фильтр Гаусса для свертки с изображением. Уравнение для ядра фильтра Гаусса имеет размерность $(2k+1) * (2k+1)$ и задается следующей формулой (20):

$$H_{ij} = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(i-(k+1))^2 + (j-(k+1))^2}{2\sigma^2}\right); 1 \leq i,j \leq (2k+1). \quad (20)$$

Важно понимать, что выбор размера ядра Гаусса имеет значительное влияние на работу детектора. Чем больше размер ядра, тем ниже чувствительность детектора к шуму. Кроме того, ошибка локализации обнаружения края будет немного увеличиваться с увеличением размера ядра фильтра. В большинстве случаев размер 5x5 является оптимальным, однако это зависит от конкретной ситуации.

2.3.2