

Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский Томский политехнический университет»

На правах рукописи



Друки Алексей Алексеевич

**АЛГОРИТМЫ НЕЙРОСЕТЕВОГО ДЕТЕКТИРОВАНИЯ
И РАСПОЗНАВАНИЯ СИМВОЛОВ НА СЛОЖНОМ ФОНЕ**

05.13.01 – Системный анализ, управление и обработка информации
(в отраслях информатики, вычислительной техники и автоматизации)

Диссертация
на соискание ученой степени
кандидата технических наук

Научный руководитель:
доктор технических наук, профессор
Спицын Владимир Григорьевич

Томск – 2015

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	5
ГЛАВА 1. АНАЛИТИЧЕСКИЙ ОБЗОР МЕТОДОВ РАСПОЗНАВАНИЯ СИМВОЛОВ НА СЛОЖНОМ ФОНЕ	14
1.1 Основные особенности процесса распознавания объектов на сложном фоне	14
1.2 Методы детектирования области расположения символов на изображениях.....	16
1.2.1 Методы эмпирического анализа.....	16
1.2.2 Методы, основанные на построении модели изображения объекта	21
1.3 Методы нормализации изображений и выделения отдельных символов	34
1.4 Методы распознавания символов на изображениях	39
1.4.1 Выделение признаков	42
1.5 Технические характеристики систем распознавания автомобильных номерных знаков	50
1.6 Цель и задачи исследования	55
1.7 Основные результаты и выводы по главе 1.....	56
ГЛАВА 2. РАЗРАБОТКА АЛГОРИТМОВ РАСПОЗНАВАНИЯ СИМВОЛОВ НА СЛОЖНОМ ФОНЕ	58
2.1 Разработка алгоритма детектирования области расположения символов на изображениях.....	58
2.1.1 Сверточная нейронная сеть для предварительной классификации.	59
2.1.2 Сверточная нейронная сеть для итоговой классификации.....	65
2.1.3 Алгоритм детектирования области расположения символов	69
2.2 Разработка алгоритма нормализации изображений символов.....	74
2.3 Разработка алгоритма распознавания символов.....	80
2.3.1 Сверточная нейронная сеть для распознавания символов	80
2.3.2 Алгоритм распознавания символов	84
2.4 Выбор активационной функции для сверточных нейронных сетей.....	87

2.5 Нормирование входных значений	90
2.6 Выбор исходных значений синаптических коэффициентов	91
2.7 Выбор алгоритма обучения для сверточных нейронных сетей	92
2.7.1 Результаты применения алгоритмов обучения.....	101
2.8 Основные результаты и выводы по главе 2.....	103
ГЛАВА 3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ РАСПОЗНАВАНИЯ СИМВОЛОВ НА СЛОЖНОМ ФОНЕ	104
3.1 Общие требования к разрабатываемому программному обеспечению .	104
3.2 Выбор средств разработки программного обеспечения	105
3.3 Разработка программной системы	108
3.3.1 Классы для реализации сверточных нейронных сетей	110
3.3.2 Классы для работы с изображениями	116
3.3.3 Классы для нормализации изображений.....	118
3.4 Логические схемы разработанных алгоритмов	121
3.5 Пользовательский интерфейс программной системы.....	131
3.6 Основные результаты и выводы по главе 3.....	142
ГЛАВА 4. ТЕСТИРОВАНИЕ РАЗРАБОТАННЫХ АЛГОРИТМОВ И ПРОГРАММНЫХ СРЕДСТВ	143
4.1 Обучение и тестирование сверточных нейронных сетей	143
4.1.1 Обучение и тестирование сверточных нейронных сетей для детектирования области расположения символов	143
4.1.2 Обучение и тестирование сверточной нейронной сети для распознавания символов	154
4.2 Оценка качества работы программной системы.....	162
4.2.1 Оценка качества распознавания автомобильных номерных знаков при различном уровне освещения.....	163
4.2.2 Оценка качества распознавания автомобильных номерных знаков при различных углах отклонения, относительно регистрирующего устройства.....	166

4.2.3 Определение предельных значений углов отклонения номерных знаков, при которых точность распознавания символов остается на максимально высоком уровне	167
4.3 Сравнение технических характеристик разработанных алгоритмов и программных средств с существующими аналогами.....	175
4.4. Основные результаты и выводы по главе 4.....	184
ЗАКЛЮЧЕНИЕ	186
СПИСОК СОКРАЩЕНИЙ.....	188
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ.....	189
ПЕРЕЧЕНЬ ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ	209
ПРИЛОЖЕНИЕ 1	213

ВВЕДЕНИЕ

Актуальность работы. На сегодняшний день одним из самых интенсивно развивающихся научных и технологических направлений является обработка и анализ изображений. Задача обработки и анализа изображений является относительно новой и получила свое развитие во второй половине 20 века, параллельно с развитием компьютерных технологий [5, 9, 22]. Данная задача приобрела свою актуальность в условиях, когда человек не способен выполнять требуемые функции по причине информационной перегрузки и большого объема информации. В связи с этим возникла необходимость в автоматизации определенного рода процессов, таким образом, данная проблема оказалась в области междисциплинарных исследований [44, 51].

Обработка и анализ изображений находит широкое применение в различных сферах, таких как: контроль доступа к информации по идентификации личности [31, 42], дактилоскопия, распознавание текста [6, 45], улучшение качества изображений [84], обработка спутниковых снимков, обработка рентгеновских снимков в медицине и т.д.

Среди особо важных и интересных задач в данной предметной области можно выделить задачу распознавания образов на изображениях. В развитие теории и практики распознавания образов большой вклад внесли многие зарубежные ученые, среди которых можно отметить: *Y. LeCun* [137–139], *Y. Bengio* [138], которые работают в данной области уже более 15 лет, а так же *S. Lawrence* [134], *P. Viola* и *M.J. Jones* [181–183], *D.G. Lowe* [141, 142], *H. Bay* [92], *R. Bradski* [91], *Y. Freund* [112], *R.W. Hamming* [81] и другие. Большой вклад в данную научную область внесли российские ученые: *Ю.И. Журавлев* [32, 33], *В.А. Сойфер* [72, 73], *В.Н. Ванник* и *А.Я. Червоненкис* [9], *Н.Г. Загоруйко* [29, 30], *Б.А. Алпатов* [3], *Л.М. Местецкий* [40], *А.Л. Горелик* [21], *В.С. Симанков* [52], *М.В. Гашиников* [17], *С.Ю. Желтов* [25], *В.П. Вежневцев* [10], *А.И. Пахирка* [43], *Б.В. Анисимов* [5], *Ю.Р. Цой* [75, 76] и другие.

Методы, применяемые для решения задачи распознавания объектов на изображениях, должны обеспечивать высокую точность и быстродействие.

Процесс распознавания объектов можно разделить на несколько основных этапов: обнаружение объекта на изображении и определение координат его расположения (детектирование); извлечение характерных признаков объекта; классификация объекта по извлеченным признакам (распознавание) [35].

Распознавание объектов на изображениях имеет большую значимость в таких областях как: государственная оборона, обеспечение правопорядка, предотвращение несанкционированного доступа к секретным объектам и объектам военного назначения [18, 19]. Так же к данным областям можно отнести задачу распознавания символов на сложном фоне, а именно маркировок различных объектов и технических изделий, номеров зданий, номеров транспортных средств, номеров железнодорожных вагонов и т.д. За последние годы был представлен ряд методов и алгоритмов, применяемых на различных стадиях процесса распознавания символов и маркировок на изображениях различных объектов. Среди данных методов можно выделить следующие: метод главных компонент; метод опорных векторов; искусственные нейронные сети; дескрипторы локальных особенностей; алгоритмы, основанные на AdaBoost классификаторе [111, 112] и т.д.

Однако применение данных методов недостаточно эффективно в реальных условиях, которые характеризуются наличием сложного фона (сложной фоновой структуры) на изображениях, различной степенью освещенности, наличием шумов, аффинными и проекционными искажениями объектов, возникающими из-за изменения углов регистрации. Особенно ярко эти недостатки стали проявляться при масштабном использовании программных систем, основанных на данных методах. Эффективность работы многих систем становится ниже технологически приемлемого уровня при наличии искажений подобного рода. Таким образом, данная задача до сих пор не решена в полном объеме.

В этой связи, задача разработки методов и алгоритмов распознавания символов на сложном фоне, обладающих высокой степенью инвариантности к различным шумовым, аффинным и проекционным искажениям, остается актуальной и на сегодняшний день. В процессе исследования было выявлено, что

одним из наиболее актуальных и интенсивно развивающихся направлений данной предметной области является распознавание автомобильных номерных знаков (государственных регистрационных знаков транспортных средств). Поэтому было принято решение осуществлять реализацию и апробацию алгоритмов на примере распознавания символов на автомобильных номерных знаках при наличии сложной фоновой структуры на изображениях.

Целью диссертационной работы является разработка алгоритмов и программных средств, позволяющих повысить эффективность распознавания символов на сложном фоне, подверженных шумовым, аффинным и проекционным искажениям.

Для достижения поставленных целей необходимо последовательное решение **следующих задач:**

1. Исследовать методы и алгоритмы, применяемые на различных стадиях процесса распознавания символов на изображениях со сложной фоновой структурой.

2. Разработать алгоритм, обеспечивающий высокую эффективность работы при детектировании символов, подверженных различным шумовым, аффинным и проекционным искажениям.

3. Создать алгоритм нормализации изображений символов, обеспечивающий возможность определения точных границ строки символов, расположенной под различными углами наклона на изображении.

4. Разработать алгоритм, обеспечивающий высокую эффективность работы при распознавании символов, подверженных шумовым, аффинным и проекционным искажениям.

5. Реализовать разработанные алгоритмы в виде программной системы, предназначенной для распознавания символов на изображениях со сложной фоновой структурой, и провести вычислительные эксперименты с целью оценки их качества и эффективности.

Апробация работы. Результаты работы были представлены на следующих конференциях и семинарах: XIX Всероссийский научный семинар

«Нейроинформатика, ее приложения и анализ данных» (Красноярск, 2011 г.); IX Всероссийская научная конференция «Нейрокомпьютеры и их применение» (Москва, 2011 г.); 18 Всероссийская межвузовская научно-техническая конференция студентов и аспирантов «Микроэлектроника и информатика» (Зеленоград, 2012 г.); XII, XIII, XIV Международные научные конференции «Интеллект и наука» (Красноярск, 2012, 2013 гг.); XIV Всероссийская научно-техническая конференция «Нейроинформатика-2012» (Москва, 2012 г.); Всероссийская научно-техническая конференция студентов, аспирантов и молодых ученых «Научная сессия ТУСУР» (Томск, ТУСУР, 2013 г.); XII Всероссийская научно-практическая конференция студентов, аспирантов и молодых ученых «Молодежь и современные информационные технологии» (Томск, ТПУ, 2014 г.); XX Международная научно-практическая конференция студентов, аспирантов и молодых ученых «Современные техника и технологии» (Томск, ТПУ, 2014 г.); XII Всероссийская научно-практическая конференция студентов, аспирантов и молодых ученых «Технологии Microsoft в теории и практике программирования» (Томск, ТПУ, 2015 г.); The 7th International Forum on Strategic technology IFOST (Томск, 2012 г.); The 9th International Forum on Strategic technology IFOST (Cox's Bazar, Bangladesh, 2014 г.) ; The 10th International Forum on Strategic technology IFOST (Bali, Indonesia, 2015 г.); International Siberian Conference on Control and Communications «SIBCON-2015» (Omsk, 2015).

Основное содержание диссертации отражено в 22 работах; из них 5 статей в периодических изданиях из перечня ВАК; 5 публикаций в научных изданиях, индексируемых в базе данных Scopus; 12 докладов на всероссийских и международных конференциях.

Основное содержание работы.

В **первой главе** представлен аналитический обзор алгоритмов и методов, применяемых на различных стадиях процесса распознавания символов и маркировок различных объектов на изображениях со сложной фоновой структурой. Представлены их достоинства и недостатки, на основании чего сделан выбор методов для осуществления их дальнейшей модификации и

разработки алгоритмов. Представлен обзор технических характеристик систем, предназначенных для распознавания автомобильных номерных знаков, на основании которых сформулированы технические требования к разрабатываемым алгоритмам.

Во **второй главе** приводится подробное описание предложенных алгоритмов, применяемых на различных стадиях процесса распознавания символов на изображениях со сложной фоновой структурой. Разработан алгоритм детектирования области расположения символов на изображениях со сложной фоновой структурой, основанный на сверточных нейронных сетях и обеспечивающий высокую степень инвариантности к различным шумовым, аффинным и проекционным искажениям. Предложен алгоритм нормализации изображений символов, обеспечивающий возможность определения точных границ строки символов, расположенной под различными углами наклона на изображении. Представлен алгоритм распознавания символов, основанный на сверточной нейронной сети и обеспечивающий высокую устойчивость к различным искажениям входных данных. Проведен обзор алгоритмов обучения сверточных нейронных сетей, на основании чего был выбран наиболее оптимальный алгоритм обучения.

Третья глава посвящена разработке программного обеспечения, на основе предложенных алгоритмов. Проведен обзор инструментальных библиотек по компьютерному зрению и обработке изображений и приведено обоснование выбора средств для программной реализации алгоритмов. Приведено описание архитектур разработанных библиотек, основных методов и переменных классов. Представлены логические схемы разработанных алгоритмов и пользовательский интерфейс разработанной программной системы.

В **четвертой главе** приведены результаты обучения и тестирования разработанных сверточных нейронных сетей. Представлены результаты апробации разработанных алгоритмов на примере распознавания автомобильных номерных знаков на изображениях. Приведено сравнение полученных

результатов с результатами существующих программных систем, использующихся для решения подобных задач.

Научная новизна результатов диссертационной работы заключается в следующем:

1. Впервые предложен алгоритм детектирования символов на изображениях со сложной фоновой структурой, отличающийся от аналогов применением алгоритмической композиции из двух сверточных нейронных сетей, работающих по принципу последовательной классификации. Алгоритм обеспечивает высокую точность классификации и повышенную устойчивость к шумовым, аффинным и проекционным искажениям входных данных.

2. Предложена модификация алгоритма нормализации изображений символов, основанная на построении гистограмм распределения яркости пикселей, отличающаяся от известных реализаций меньшей вычислительной сложностью и обеспечивающая более высокую скорость работы. Алгоритм обеспечивает обнаружение точных границ строки символов, расположенной под различными углами наклона.

3. Разработан новый алгоритм распознавания символов на изображениях, основанный на применении сверточной нейронной сети и отличающийся от аналогов тем, что позволяет распознавать символы, не используя процесс их сегментации. Алгоритм обеспечивает высокую скорость работы и высокую степень инвариантности к шумовым, аффинным и проекционным искажениям символов на изображениях.

Научную ценность работы представляет вклад в развитие области распознавания объектов на изображениях, заключающийся в предложенном алгоритме детектирования символов, основанном на алгоритмической композиции из двух сверточных нейронных сетей; алгоритме нормализации изображений, основанном на построении гистограмм распределения яркости пикселей; алгоритме распознавания символов, основанном на сверточной нейронной сети. Предложенные алгоритмы обладают высоким быстродействием и обеспечивают возможность распознавания символов на изображениях со

сложным фоном, подверженных различным шумовым, аффинным и проекционным искажениям.

Практическая значимость. Реализованные в диссертационной работе алгоритмы предназначены для решения задачи распознавания символов на изображениях со сложной фоновой структурой. Апробация реализованных алгоритмов осуществлялась на примере распознавания автомобильных номерных знаков на изображениях. Разработанные алгоритмические и программные средства могут найти применение в системах безопасности, видеонаблюдения, видеоконтроля и могут применяться правоохрательными органами, органами ГИБДД и частными лицами для решения задачи распознавания автомобильных номерных знаков, маркировок различных объектов и других подобных задач.

Методы исследования. При решении поставленных задач использованы методы искусственного интеллекта, искусственные нейронные сети, математическое моделирование, алгоритмы цифровой обработки изображений, технология разработки программного обеспечения, а так же методы теории вероятностей и математической статистики для количественной обработки результатов численных экспериментов.

Личный вклад. Представленные в диссертационной работе теоретические и практические результаты, получены лично автором. В работах, опубликованных в соавторстве с сотрудниками научной группы, диссертант принимал непосредственное участие в реализации алгоритмов и экспериментальных исследованиях. Постановка задачи диссертационного исследования осуществлялась автором совместно с научным руководителем, д.т.н., профессором В.Г. Спицыным.

Основные положения, выносимые на защиту:

1. Алгоритм детектирования символов на изображениях со сложной фоновой структурой, отличающийся от аналогов применением алгоритмической композиции из двух сверточных нейронных сетей, работающих по принципу последовательной классификации.

2. Модификация алгоритма нормализации изображений символов, основанная на построении гистограмм распределения яркости пикселей, отличающаяся от известных реализаций меньшей вычислительной сложностью и обеспечивающая более высокую скорость работы.

3. Алгоритм распознавания символов, основанный на применении сверточной нейронной сети и отличающийся от аналогов тем, что позволяет распознавать символы, не используя процесс их сегментации.

4. Разработанная программная система обеспечивает высокую эффективность при решении задач распознавания символов на изображениях со сложной фоновой структурой, подверженных различным шумовым, аффинным и проекционным искажениям.

Автор выражает большую благодарность своему научному руководителю профессору, доктору технических наук В.Г. Спицыну за оказание помощи в написании диссертационной работы, ценные замечания и советы, а так же конструктивную критику. Автор благодарит заведующего кафедрой Вычислительной техники, профессора Н.Г. Маркова за ценные замечания и обсуждение работы. Также выражается благодарность за ценные замечания и всестороннюю помощь кандидатам технических наук, доцентам: Ю.Р. Цою, А.А. Белоусову, Ю.Б. Буркатовской, А.В. Кудинову, Ю.Я. Кацману, Ю.А. Болотовой и всем членам научной группы профессора В.Г. Спицына.

Степень достоверности результатов проведенных экспериментов подтверждается результатами численных экспериментов на тестовых задачах различного вида и согласованностью результатов диссертационной работы с результатами других авторов.

Внедрение работы. Результаты диссертационной работы внедрены в Томском политехническом университете на кафедре Вычислительной техники при подготовке курса «Методы интеллектуальной обработки и анализа изображений» для обучения специалистов по магистерской программе «Компьютерный анализ и интерпретация данных»; в ООО «Сибспецавтоматика» для реализации технологических задач в области видеонаблюдения и обработки

изображений; в Северной клинической больнице Федерального государственного бюджетного учреждения «Сибирский федеральный научно-клинический центр Федерального медико-биологического агентства» в системах видеонаблюдения для распознавания государственных регистрационных знаков транспортных средств на территории данного учреждения.

Реализация результатов работы. Методы, алгоритмы и программные средства, разработанные в диссертационной работе, использовались при выполнении работ по гранту РФФИ № 09-08-00309 «Создание программного комплекса автоматизированной обработки изображений и распознавания образов на основе применения искусственных нейронных сетей, регуляторных сетей и эволюционных алгоритмов» (2009–20011 гг.); при выполнении проекта «Создание комплексных технологий распознавания объектов на изображениях на основе применения моделей зрительного восприятия и методов вычислительного интеллекта», поддержанном грантом РФФИ № 12-08-00296 (2012–2014 гг.); при выполнении проекта от 05.06.2014 г. № 14.578.21.0032 «Разработка экспериментального образца аппаратно-программного комплекса для неинвазивной регистрации микропотенциалов сердца в широкой полосе частот без фильтрации и усреднения в реальном времени с целью раннего выявления признаков внезапной сердечной смерти», в рамках ФЦП «Исследования и разработки по приоритетным направлениям развития научно-технологического комплекса России на 2014 – 2020 годы».

Структура и объем работы. Диссертация включает в себя введение, четыре главы, заключение, список использованной литературы, содержащий 194 наименований. Общий объем диссертационной работы составляет 216 страниц машинописного текста, 84 рисунка и 52 таблицы.

ГЛАВА 1. АНАЛИТИЧЕСКИЙ ОБЗОР МЕТОДОВ РАСПОЗНАВАНИЯ СИМВОЛОВ НА СЛОЖНОМ ФОНЕ

В данной главе рассматриваются основные подходы, применяемые на стадиях процесса распознавания символов и маркировок различных объектов на изображениях со сложной фоновой структурой. Представлены их достоинства и недостатки, на основании чего сделан выбор наиболее оптимальных методов для дальнейшей их модификации и разработки алгоритмов. Проведен обзор технических характеристик существующих систем распознавания автомобильных номерных знаков, на основании которого сформулированы требования к разрабатываемым алгоритмам.

1.1 Основные особенности процесса распознавания объектов на сложном фоне

Объекты на изображениях представляют собой некоторую упорядоченную совокупность признаков. Совокупность объектов, обладающих некоторыми общими характерными признаками, называется классом. Машинное распознавание объектов на изображениях заключается в способности автоматически классифицировать исходные данные и относить их к определенному классу с помощью выделения характерных признаков объекта.

Проблема автоматического распознавания объектов на изображениях является относительно новой и получила свое развитие во второй половине 20 века [5, 9]. В первых системах распознавания использовались простые геометрические модели, и требовалось участие администратора, который производил выделение признаков объекта на изображении. Затем система выполняла численные измерения размеров и расстояний выделенных признаков относительно контрольных точек [10].

В настоящее время распознавание объектов на изображениях осуществляется на основе сложных математических представлений о существующих процессах и является очень востребованной в областях компьютерного зрения, обработки и анализа изображений, биометрии, систем

безопасности и видеоконтроля. Современные методы, решающие задачу распознавания объектов, применяются для решения широкого круга задач: распознавание лиц [39], отпечатков пальцев, сетчатки глаза, печатных символов [78], автомобильных номерных знаков [7], маркировки на поверхностях различных объектов и т.д.

На сегодняшний день достигнуты значительные успехи при решении задач распознавания объектов и символов на изображениях, однако существует ряд сложностей, которые существенно снижают надежность применения современных методов:

- Низкое разрешение изображений. В результате чего, признаки объектов могут быть плохо различимы.

- Наличие сложной фоновой структуры на изображениях. Это подразумевает наличие на изображениях посторонних объектов, которые могут иметь визуальные признаки, схожие с искомым объектом.

- Различные искажения, полученные в процессе регистрации изображений. Регистрация изображений может производиться при неудачных ракурсах, плохих погодных условиях, различных углах и условиях освещения. В результате этого, на изображениях могут быть различные шумовые помехи, символы могут быть подвержены аффинным и проекционным искажениям.

- Количество искомых объектов на изображении заранее не известно.

Все это требует применения различных алгоритмов предварительной обработки, что в свою очередь усложняет процесс распознавания, делает его более громоздким, увеличивает объем и время вычислительных процессов. Поэтому на сегодняшний день до сих пор существует потребность в разработке методов и алгоритмов, решающих вышеперечисленные проблемы.

Процесс распознавания символов на изображениях со сложной фоновой структурой можно разделить на следующие этапы:

1. Детектирование области расположения символов на изображении.
2. Нормализация изображения и выделение отдельных символов.
3. Распознавание символов [28].

Для реализации каждого из этих этапов применяются свои индивидуальные методы и алгоритмы. Поэтому дальше в этой главе приводится обзор методов и алгоритмов по трем категориям применения.

1.2 Методы детектирования области расположения символов на изображениях

В данном разделе проводится анализ подходов, применяемых для решения задачи детектирования области расположения символов на изображениях, рассматриваются алгоритмы, применяемые для распознавания маркировок различных объектов, в том числе и автомобильных номерных знаков.

Детектирование области расположения символов на сложном фоне является первым этапом процесса их распознавания. Данный этап является очень важным и оказывает существенное влияние на производительность системы. Общий смысл данного процесса заключается в классификации изображений по двум классам: наличие или отсутствие в обрабатываемой области изображения искомых символьных образов, соответствующих требуемым критериям. В идеале система должна обнаруживать все области изображения, в которых содержатся искомые объекты, независимо от условий освещения, угла регистрации, масштаба, погодных условий.

Методы детектирования объектов на изображении можно разделить на две категории:

- Методы эмпирического анализа. Данные методы основаны на человеческом опыте об имеющихся характеристиках исследуемого объекта.
- Методы, основанные на построении модели изображения объекта [10].

Данная классификация методов является условной, в некоторых научных работах определенные методы могут относиться к обеим категориям.

1.2.1 Методы эмпирического анализа

Методы эмпирического анализа основываются на определенных правилах, которым должна соответствовать исследуемая область изображения, для того

чтобы определить содержание в ней искомого объекта. Данные правила являются попыткой сформировать эмпирические знания о том, чем руководствуется человек при классификации определенных классов объектов [43]. Правила могут включать в себя различные характеристики, присущие объекту: размер, форма, цвет, области резкого изменения яркости в текстуре объекта, геометрические характеристики.

Методы, основанные на сравнении с шаблоном

Принцип работы данных методов заключается в сравнении каждой области изображения с заданным шаблоном, в результате чего определяется взаимная корреляция. Шаблоны задаются вручную либо определяются функцией и могут соответствовать целому объекту или отдельным компонентам объекта. Если коэффициент корреляции превышает заданный порог, то исследуемая область изображения отмечается как содержащая образ автомобильного номерного знака [164].

Известны различные модификации данного подхода, в которых использовались несколько шаблонов, соответствующих отдельным компонентам объектов, представлявших собой области в виде линий. Работа данных алгоритмов состояла из нескольких этапов, на каждом из которых определялась достоверность найденных областей. Если область изображения соответствует одному из шаблонов, то она отмечается как область, представляющая интерес для более детального анализа. Таким образом, повышается мера сходства областей изображения с заданным шаблоном. Далее выявленные области сравнивались с другими шаблонами, что позволяло определить наличие искомого объекта [99]. Данные методы использовались в основном в ранних работах по классификации и распознаванию объектов на изображениях. Это были первые попытки сформировать признаки изображения объектов.

Существуют современные модификации шаблонных методов, применяемых для обнаружения областей расположения символов, маркировок технических

объектов и автомобильных номерных знаков [114, 132, 159]. Данные методы показывают высокую точность распознавания.

Недостаток данных методов заключается в низкой скорости работы и высокой чувствительности к фотометрическим и геометрическим искажениям объектов на изображениях.

Метод, основанный на преобразовании Хафа

Данный метод был предложен *P. Hough* в 1962 году [125]. Метод заключается в поиске на изображении объектов, принадлежащих определенному классу геометрических фигур: прямые линии, окружности и т.д. Поиск осуществляется с помощью процедуры голосования, применяемой к пространству параметров изображения. Искомый объект представляется в виде параметрического уравнения, параметры которого представляют так называемое фазовое пространство.

Основное утверждение преобразования Хафа заключается в том, что любая точка изображения может принадлежать некоторому набору линий. Прямая линия может задаваться следующим уравнением:

$$r = X \cos \theta + Y \sin \theta, \quad (1.1)$$

где r – длина перпендикуляра, построенного из начала координат на прямую; θ – угол между этим вектором и осью абсцисс.

Синусоидальные кривые для каждой точки являются уникальными. Данные кривые пересекутся в точке (r, θ) только в том случае, если порождающие их точки находятся на прямой, описываемой уравнением (1.1). Функция $A(r, \theta)$ называется аккумуляторной функцией и ее значение в точке (r, θ) будет равно количеству точек, расположенных на соответствующей прямой. Чтобы найти прямые линии на изображении достаточно найти локальные максимумы аккумуляторной функции.

Для усиления пространственных частот на начальном этапе данного метода выполняется улучшение контраста, бинаризация или выделение границ на изображении [1, 20, 73]. Далее выполняется последовательный анализ каждой

точки изображения и ее соседей. При этом вычисляется вес границы в данной точке. Если вес соответствует заданному пороговому значению, то вычисляются параметры прямой и увеличивается значение в соответствующей ячейке массива. Далее выполняется поиск максимальных значений в массиве, за которые проголосовало большее количество пикселей изображения. Таким образом, находятся параметры уравнений искомого объекта [130].

В работах [116, 157, 194] данный метод применялся для детектирования автомобильных номерных знаков. Алгоритм заключается в поиске прямых линий, удовлетворяющих следующим условиям:

1. Прямые линии должны иметь примерно одну длину и совпадающие координаты начала и конца.

2. Линии должны находиться на определенном расстоянии друг от друга.

Таким образом, на изображении находятся пары линий, которые затем сопоставляются. Если линии пересекаются и образуют прямоугольник, то вычисляется соотношение высоты и ширины граней найденного прямоугольника. Если соотношения удовлетворяют условиям, то область помечается на возможное содержание автомобильного номерного знака.

Достоинство данного метода заключается в том, что он является инвариантным к яркости и цвету изображения.

Недостаток данного метода заключается в том, что он восприимчив к любой области изображения, имеющей прямоугольную форму. Это снижает эффективность его применения на изображениях со сложной фоновой структурой.

Методы, основанные на гистограммном анализе изображения

Данные методы основываются на предположении, что частотная характеристика области расположения объекта, содержащего символьные образы, отличается от остальных областей изображения и имеет более высокую интенсивность пикселей.

Для усиления пространственных частот и подавления шумов на начальном этапе данного подхода выполняется улучшение контраста, бинаризация или выделение границ на изображении. В результате этого, границы объектов на изображении становятся более контрастными, а фон затемняется.

Обычно в задачах подобных распознаванию маркировок технических объектов и автомобильных номерных знаков, искомый объект представляет собой совокупность символов и линий темного цвета на более светлом фоне, или же наоборот. Таким образом, данная область изображения будет иметь высокий контраст.

Идея данных методов заключается в следующем: выполняется сканирование изображения и при этом вычисляется среднее значение яркости пикселей в каждой строке изображения (иногда вычисляют по столбцам). В том месте, где расположен искомый объект средняя интенсивность пикселей будет значительно отличаться от остальных областей изображения. Максимальное значение полученной проекции может совпасть с расположением объекта [144] (рисунок 1.1).



Рисунок 1.1 – Результат применения гистограммного анализа: а) исходное изображение; б) построение гистограмм интенсивности пикселей [144]

Данные методы применялись многими исследователями с различными модификациями. В работах [105–107] данный метод применялся с различными видами фильтраций изображения для подавления шумов и более яркого выделения области расположения автомобильного номерного знака. В работе [28] для детектирования автомобильного номерного знака данный метод применялся

совместно с преобразованием Хафа. Данные алгоритмы могут показывать хорошие результаты работы в том случае, если размер изображения автомобиля или другого технического объекта сопоставим с размерами кадра.

Достоинство данных методов заключается в простой реализации и высокой скорости работы.

Недостаток данных методов заключается в чувствительности к любой области изображения, имеющей параметры интенсивности пикселей схожие с автомобильным номерным знаком.

1.2.2 Методы, основанные на построении модели изображения объекта

Методы данной категории не предполагают в явном виде формализацию правил, которыми руководствуется человек при классификации объектов на изображениях. Характеристики объектов вычисляются неявно, в результате применения математической статистики и машинного обучения. Выполняется автоматическая подстройка системы к характеристикам обрабатываемых данных.

Метод главных компонент

Метод главных компонент (Principal Component Analysis, PCA) был предложен *K. Pearson* в 1901 году [153]. Данный метод применяется в различных областях, таких как: обработка и анализ изображений, распознавание образов, сжатие данных и т.д.

Метод PCA направлен на уменьшение размерности данных, при наименьших потерях информации, что бывает очень актуально в области анализа и обработки изображений. В данном методе выполняется линейное ортогональное преобразование входного вектора X , содержащего коррелированные компоненты, в вектор меньшего размера Y , который содержит некоррелированные переменные. Некоррелированные переменные называются главными компонентами. Вектор X состоит из всех примеров обучающего набора изображений. Собственные вектора получаются при решении следующего уравнения:

$$D = \Phi^T K \Phi,$$

где D – диагональная матрица собственных чисел; K – матрица ковариации для вектора X ; Φ – матрица собственных векторов.

Из матрицы собственных векторов Φ выбираются M наибольших собственных чисел, из которых формируется подматрица Φ_M . Таким образом, получим выражение: $y = \Phi_M^T \tilde{x}$, где $\tilde{x} = x - \bar{x}$ – нормализованный вектор с нулевым математическим ожиданием.

При выборе первых M компонент векторное пространство разбивается на собственное пространство, содержащее главные компоненты.

Сначала вычисляются главные компоненты для обучающего набора изображений объектов. Выполняется фильтрация параметров объектов, где отбрасываются менее существенные из них и пространство параметров уменьшается. Таким образом, объекты на изображении могут быть представлены в виде минимального набора параметров. Остальные компоненты являются второстепенными и характеризуют незначительные различия между объектами и шум [124].

При классификации объектов на изображении вычисляется Евклидово расстояние между главными компонентами исследуемого изображения и взвешенной комбинацией главных компонент обучающего набора. Если главные компоненты изображения имеют наиболее близкое соответствие с главными компонентами какого-либо объекта из обучающего набора, то считается, что на изображении содержится данный объект.

В работах [166, 190] данный метод использовался для детектирования автомобильных номерных знаков. Недостаток данного метода заключается в низкой устойчивости к различным шумовым, аффинным и проекционным искажениям объектов на изображениях, возникающим при различных условиях регистрации. Для эффективного применения данного метода изображения объектов должны быть получены при одинаковых условиях регистрации.

Достоинство метода PCA заключается в обеспечении уменьшения размерности данных, что является важным фактором в работе с большими базами изображений.

Линейный дискриминантный анализ

Линейный дискриминантный анализ (Linear Discriminant Analysis, LDA) направлен на решение задачи классификации объектов по нескольким классам. В отличие от PCA в данном методе не ставится цель нахождения пространства признаков наименьшей размерности, эффективно описывающего обучающий набор данных. Основной задачей данного метода является нахождение проекции признаков в пространство, в котором различные классы объектов имеют максимальное отличие друг от друга [111].

Для каждого объекта изображения, принадлежащего определенному классу, находятся характерные признаки, которые имеют общее сходство с признаками других объектов данного класса. Набор таких объектов формирует обучающую выборку, в которой известна принадлежность каждого объекта к определенному классу.

Набор признаков определяется путем проекции пространства признаков R^n в подпространство меньшей размерности R^m . Причем вектора признаков различных классов объектов должны иметь максимально отличные друг от друга параметры, чтобы свести к минимуму их внутриклассовое различие и увеличить межклассовое. Таким образом, формируются максимально компактные кластеры характерных признаков объектов, соответствующие различным классам, которые имеют минимально возможные пересечения друг с другом.

Задача классификации сводится к тому, чтобы выделить характерные признаки входного изображения и провести их корреляцию с имеющимися признаками, сформированными в результате обучения. Таким образом, происходит определение принадлежности рассматриваемого объекта изображения к какому-либо классу.

Метод LDA широко используется при детектировании номеров железнодорожных вагонов, автомобильных номерных знаков [158, 171, 184] и т.д. Данный метод обеспечивает высокую степень инвариантности к аффинным и проекционным искажениям объектов на изображениях, возникающим при различных условиях регистрации.

Недостаток данного метода заключается в снижении эффективности работы в случае если матрица ковариации вырождена, например при большом количестве параметров. В этом случае требуется применение метода главных компонент. Также возможно снижение эффективности применения данного метода в случае одновременного изменения освещения и ракурса регистрации объекта.

Метод опорных векторов

Метод опорных векторов (Support Vector Machine, SVM) был предложен российскими учеными *В. Вапником* и *А. Червоненкисом* [176, 177]. Данный метод является линейным классификатором и применяется для решения задач классификации и регрессионного анализа.

Применение метода SVM к решению задачи классификации объектов на изображениях можно описать следующим образом. Дано множество точек в пространстве, которые сгруппированы по двум классам. Для того, что бы разделить множество точек на классы можно провести разделяющую гиперплоскость. С точки зрения классификации необходимо найти оптимальную гиперплоскость, расстояние от которой до каждого класса будет максимально. Чем больше расстояние от разделяющей гиперплоскости до каждого класса, тем меньше ошибка классификации. Вектора, расположенные в пространстве ближе всего к гиперплоскости называются опорными векторами.

Необходимо найти классифицирующую функцию $f(x)$, которая принимает разные значения для векторов разных классов. Для определения данной функции используется обучающий набор данных $(x_1, y_1), \dots, (x_m, y_m)$. Функция, при которой ожидаемая ошибка классификации принимает минимальное значение, является наиболее оптимальной функцией. Классифицирующая функция имеет вид:

$$f(x) = \text{sign}(\langle w, x \rangle + b),$$

где x – объект, принадлежащий пространству; w – вектор нормали к гиперплоскости; $\langle, \rangle$ – скалярное произведение; b – вспомогательный параметр.

Объекты, для которых функция $f(x)$ принимает значения равные 1 или -1 , относятся к разным классам. Необходимо выбрать параметры w и b таким образом, что бы расстояние до каждого класса было максимальным.

На практике часто возникают случаи, когда данные невозможно разделить линейным способом. Для решения подобной ситуации выполняется преобразование исходного пространства x в пространство более высокой размерности с помощью специального отображения $\varphi: R^n \rightarrow X$. Отображение φ выбирается таким образом, что бы в полученном пространстве множество данных было линейно разделимо. Основная сложность заключается в том, что с увеличением размерности пространства, увеличивается сложность процесса классификации.

Данный метод активно применяется для детектирования различных объектов, в том числе и автомобильных номерных знаков [86, 118, 134]. Недостаток данного метода заключается в том, что в процессе классификации учитывается не все множество данных, а только лишь часть, находящаяся на границах множества.

Достоинство метода SVM заключается в том, что для классификации достаточно небольшого набора данных.

Дескрипторы локальных особенностей изображения

Для выделения признаков объектов на изображениях часто применяются дескрипторы локальных особенностей. Дескриптором называется совокупность параметров, с помощью которых можно описать характеристики изображения, например, такие как цвет, текстуру и т.д.

Начальным этапом обнаружения объектов на изображении с помощью дескрипторов является определение характерных точек. Характерными точками называются точки, обладающие высокой локальной информативностью,

параметры которых не меняются при различных фотометрических и геометрических преобразованиях изображения. Для найденного множества точек на изображении рассчитываются дескрипторы. На рисунке 1.2 представлен пример работы дескрипторов.



Рисунок 1.2 – Пример работы дескрипторов: а) исходное изображение; б) вычисление признаков; в) обнаруженные объекты [142]

Одним из наиболее популярных алгоритмов, включающих дескриптор и детектор характерных точек изображения, является SIFT (Scale Invariant Feature Transform), предложенный *D.G. Lowe* в 1999 году [141, 142]. Данный алгоритм представляет собой локальную гистограмму направлений градиентов изображения. Принцип работы алгоритма SIFT заключается в вычислении свертки исходного изображения с ядром Гаусса при изменяющемся параметре сглаживания. После этого происходит преобразование изображений к одному размеру, и вычисляется их разность. Далее выполняется сравнение каждого пикселя на изображении с восемью соседними пикселями при тех же параметрах и масштабе, с девятью соседними пикселями в большем масштабе и с девятью в меньшем масштабе. Пиксели, в которых локальные экстремумы превосходят заданный порог, выбираются как характерные точки. Для каждой выбранной точки вычисляется определенный локальный дескриптор, который характеризует направление градиентов в данной окрестности пикселей.

В 2005 году исследователи *N. Dalal* и *B. Triggs* представили алгоритм HOG (Histogram of Oriented Gradients) [102]. Принцип работы данного алгоритма заключается в том, что изображение представляется в виде плотной сетки равномерно распределенных ячеек. Для пикселей внутри каждой ячейки

вычисляются гистограммы направлений градиентов. На основе полученных параметров выполняется построение дескриптора. Для увеличения точности в данном алгоритме применяется нормализация перекрывающегося локального контраста гистограмм. Нормализованные дескрипторы обладают повышенной устойчивостью к изменению интенсивности освещения.

В 2006 году *T. Tuytelaars, H. Bay, L. Van Gool* представили алгоритм SURF (Speeded Up Robust Features), который включает в себя дескриптор и детектор характерных точек изображения [92]. В данном алгоритме при вычислении характерных точек используются не гистограммы взвешенных градиентов, а целочисленные прямоугольные фильтры разного масштаба. Это обеспечивает устойчивость к поворотам объекта и к изменению масштаба. Пиксели, в которых локальные экстремумы превосходят заданный порог, выбираются как характерные точки и в них вычисляются локальные дескрипторы. Вокруг точки строится квадратная область и делится на несколько подобластей. В каждой подобласти вычисляются отклики на два типа вейвлетов – горизонтально и вертикально направленные. Полученные отклики взвешиваются Гауссианом и суммируются.

Перечисленные дескрипторы широко используются при детектировании различных объектов, в том числе и автомобильных номерных знаков [93, 97, 120, 123, 151, 154, 165, 189]. Данные дескрипторы обеспечивают высокую степень инвариантности к геометрическим преобразованиям и изменению масштаба изображения.

Недостатком применения дескрипторов является низкая устойчивость работы при различных условиях освещения, при отражающих поверхностях, а также при различных углах регистрации объектов.

Алгоритм AdaBoost

Алгоритм AdaBoost (Adaptive boosting – адаптированное улучшение) был предложен *Y. Freund* и *R. Schapire* в 1999 году [111, 112]. Данный алгоритм

является алгоритмом машинного обучения и успешно применяется при решении задач классификации объектов на изображениях и видеопоследовательностях.

Алгоритм AdaBoost имеет адаптивный принцип работы и представляет собой каскадную структуру из слабых классификаторов, каждый из которых учится на ошибках предыдущего. Каждый слабый классификатор ориентирован на определенный набор характерных признаков и может выдавать два варианта ответов: «верно», либо «ложь» [100]. В каждом каскаде определяется сумма значений слабых классификаторов и сравнивается с заданным порогом. Таким образом, каскадный классификатор убирает из рассмотрения все области изображения, в которых отсутствуют признаки искомого объекта. В итоге остаются только области, имеющие наибольшую вероятность на содержание искомого объекта.

В 2001 году *P. Viola* и *M.J. Jones* предложили алгоритм Виолы-Джонса, основанный на AdaBoost [181–183]. Данный алгоритм позволяет выполнять детектирование объектов на изображениях на основе признаков Хаара. Признаки Хаара представляют собой прямоугольные области, состоящие из нескольких смежных частей. Различные классы объектов обладают индивидуальными признаками, которые можно выразить в виде распределения признаков Хаара (рисунок 1.3).

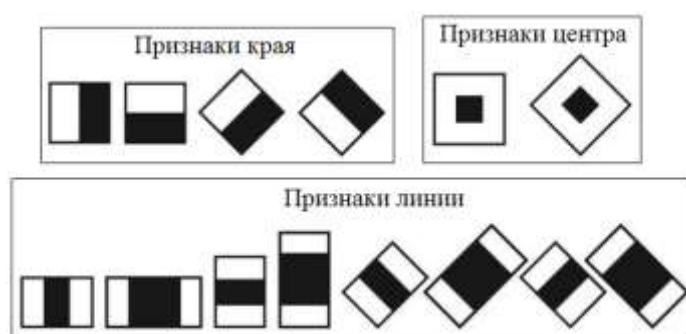


Рисунок 1.3 – Признаки Хаара

В каждой области изображения вычисляется несколько тысяч вариантов расположения признаков, в зависимости от их положения и масштаба.

Происходит суммирование интенсивностей пикселей в черных и белых областях признаков Хаара и затем вычисляется разность между суммами.

Наиболее часто алгоритм Виолы-Джонса используется для детектирования лиц, но также известны его применения для решения задач обнаружения автомобильных номерных знаков [89, 170, 185, 186]. Данный алгоритм показывает достаточно высокую точность детектирования, а так же высокую скорость работы.

Недостаток данного алгоритма заключается в относительно не высокой степени инвариантности к аффинным и проекционным искажениям объектов на изображениях и интенсивности освещения.

Искусственные нейронные сети

Искусственная нейронная сеть (ИНС) представляет собой математическую, программную или аппаратную модель, функционирующую по принципу биологических нервных клеток человеческого организма [109].

ИНС имеют иерархическую структуру и состоят из простых взаимодействующих между собой процессоров, называемых нейронами, которые сгруппированы в слои. Каждый нейрон имеет входные и выходные связи, ведущие к нейронам соседних слоев. Связи между нейронами называются синаптическими связями и характеризуются определенными весовыми коэффициентами, которые в физическом смысле можно сравнить с электрической проводимостью [146].

Текущее значение нейрона определяется как взвешенная сумма его входных значений:

$$S = \sum_{i=1}^N x_i w_{ij},$$

где x_i – текущий выход i -го нейрона; w_{ij} – весовой коэффициент между i -м и j -м нейронами.

Каждый нейрон определяется функцией активации, которая предназначена для определения зависимости сигнала на выходе нейрона от взвешенной суммы

сигналов на его входах [161]. Пример стандартной логистической функции активации:

$$f(x) = \frac{1}{(1 + e^{-s})},$$

где s – взвешенная сумма входных значений нейрона.

Для всех данных из обучающей выборки, поступающих на вход нейронной сети определяются их требуемые значения, которые затем сравниваются с текущими значениями и вычисляется ошибка. Основным принцип обучения ИНС заключается в сведении к минимуму эмпирической ошибки классификации [82]:

$$E = \frac{1}{2} \sum_{i=0}^N (t_i - x_i)^2,$$

где x_i – действительный выход i -го нейрона; t_i – желаемый выход i -го нейрона.

После этого на основе метода градиентного спуска вычисляется величина коррекции синаптических коэффициентов между нейронами [82]:

$$\Delta w_{ij} = -\eta \delta_i x_j,$$

где δ_i – значение ошибки нейрона i ; x_j – текущий выход j -го нейрона; η – коэффициент скорости обучения, $0 < \eta < 1$.

Весовые коэффициенты изменяются до тех пор, пока ошибка не сведется к минимуму и станет меньше заданного порога. Данный тип обучения называется «обучение с учителем» [75]. Таким образом, ИНС представляют собой суперпозицию функций, где каждая функция называется нейронным слоем.

Для решения задач детектирования и распознавания объектов многими исследователями применялись ИНС различной конфигурации [93, 117, 152]. Однако большинство из них обладают низкой эффективностью при решении подобных задач по следующим причинам:

- Большой размер входных изображений приводит к увеличению количества нейронов и синаптических весовых коэффициентов нейронной сети. В результате этого увеличивается вычислительная сложность и время процесса обучения.

- Обладают высокой чувствительностью к искажениям входных данных, шумам, изменению масштаба и углов регистрации объектов.
- Не поддерживают двумерную топологию изображений, что приводит к потере взаимосвязи пространственно зависимых областей изображения.

Все это требует применения дополнительных алгоритмов, тем самым увеличивая вычислительную сложность и время выполнения задачи. Перечисленные недостатки оказались в наименьшей степени присущи так называемым сверточным нейронным сетям.

Сверточные нейронные сети

В 1998 году исследователи *Y. LeCun*, *L. Bottou*, *Y. Bengio* и *P. Haffner* предложили вид нейронных сетей, работающих по принципу зрительной системы человека, которые были названы сверточными нейронными сетями (СНС) [137–139].

СНС представляет собой особый класс многослойного персептрона, который обладает двумерной структурой и хорошо подходит для обработки изображений с высокой степенью инвариантности к смещению, поворотам, масштабированию и другим искажениям входных данных.

Структура СНС представляет собой последовательность из двух типов слоев: сверточные и подвыборочные. Каждый слой состоит из набора плоскостей (карт характеристик), которые в свою очередь состоят из нейронов.

Каждый нейрон сверточного слоя имеет связь с небольшой группой нейронов предыдущего слоя (локальное рецептивное поле). Локальные рецептивные поля нейронов сверточного слоя частично накладываются друг на друга по принципу черепицы. Значения нейронов из локального рецептивного поля умножаются на матрицу синаптических коэффициентов, а результат записывается в соответствующий нейрон сверточного слоя (рисунок 1.4 (а)).

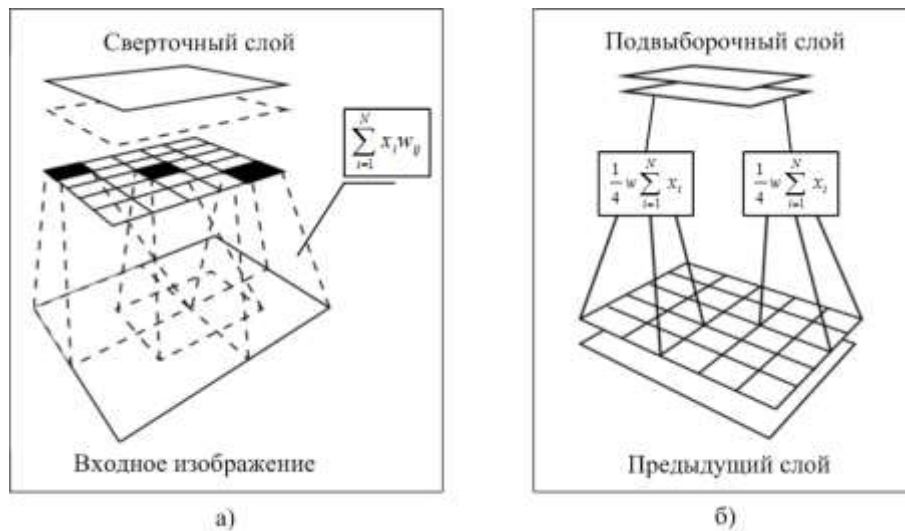


Рисунок 1.4 – Функционирование слоев в СНС

Следом за сверточным слоем располагается подвыборочный слой, который обеспечивает частичную инвариантность нейронной сети к изменению масштаба входного изображения. Количество плоскостей подвыборочного слоя обычно такое же, как и в предыдущем слое. В подвыборочном слое локальные рецептивные поля не пересекаются друг с другом и имеют фиксированный размер 2x2 нейрона. Каждый нейрон данного слоя вычисляет среднее значение своих четырех входов, умножает их на синаптические коэффициенты и полученный результат передает через функцию активации (рисунок 1.4 (б)). Таким образом, подвыборочный слой уменьшает размерность плоскостей предыдущего слоя в два раза [145].

Последовательно чередуясь друг за другом, размеры плоскостей уменьшаются, но их количество увеличивается. Чередование слоев позволяет формировать различные карты характеристик, что наделяет СНС способностью к идентификации более сложных иерархических признаков [126, 160]. Постепенно при прохождении нескольких слоев карта признаков вырождается в вектор. Последние несколько слоев СНС представляют собой классический персептрон, состоят из обычных нейронов и предназначены для классификации выделенных признаков.

Важным свойством СНС является использование общей матрицы весовых коэффициентов для всех нейронов в пределах плоскости. Данный подход позволяет использовать меньшее число настраиваемых параметров (весовых

коэффициентов) при большом количестве связей и повысить скорость процесса обучения. Таким образом, плоскости СНС представляют собой фильтры, каждый из которых осуществляет поиск индивидуальных характерных признаков входного изображения. Это позволяет сверточной нейронной сети запоминать взаимосвязь пространственно зависимых областей изображения [94, 135]. Характерные признаки, извлекаемые той или иной плоскостью, определяются в процессе обучения.

Если входное изображение искажено или смещено, то на выходе плоскости будет аналогично смещенный результат. Благодаря этому свойству обеспечивается устойчивость СНС к искажениям входных данных.

Недостаток использования СНС заключается в сложности настройки оптимальных параметров: количество слоев, плоскостей, нейронов, размер рецептивного поля и т.д.

Преимущества использования СНС для детектирования объектов на изображениях:

- Структура СНС хорошо подходит для обработки двумерных данных.
- Наслаивающиеся друг на друга рецептивные поля обеспечивают взаимосвязь пространственно зависимых областей изображения.
- Повышенная устойчивость к аффинным и проекционным искажениям входных данных, шумам, изменению масштаба.

Благодаря своим преимуществам на сегодняшний день СНС активно используются для обнаружения лиц и других объектов на изображениях и видео последовательностях [39, 115, 119, 135, 163].

Как отмечено во введении, осуществлять реализацию и апробацию алгоритмов было решено на примере распознавания автомобильных номерных знаков. На данный момент СНС не использовались для детектирования автомобильных номерных знаков и ни чего не известно о подобных экспериментах, что создает дополнительный интерес о возможности их применения для решения данной задачи.

На основании проведенного анализа для решения задачи детектирования автомобильных номерных знаков в данной работе было принято решение использовать алгоритмы, основанные на сверточных нейронных сетях.

1.3 Методы нормализации изображений и выделения отдельных символов

В данном разделе проводится анализ подходов, применяемых для решения задачи нормализации изображений и выделения отдельных символов.

Основная сложность данного этапа заключается в том, что объект, содержащий символы, в данном случае автомобильный номерной знак, может быть расположен под неизвестным углом на изображении, соответственно строка символов так же будет расположена под неизвестным углом.

Поэтому данный этап включает в себя операции, которые заключаются в определении точных границ строки символов и ее угла наклона. Так же изображение может быть подвержено шумовым, аффинным и проекционным искажениям, в результате чего, характерные признаки символов могут искажаться, контуры символов могут частично сливаться и т.д. Все это требует применения различных методов и операций предобработки.

Метод, основанный на преобразовании Хафа

Для нахождения верхней и нижней границ строки символов и их угла наклона, возможно применение преобразования Хафа. Принцип работы данного метода описан в разделе 1.2.1.

Для усиления пространственных частот на начальном этапе данного метода обычно применяются операции по улучшению контраста изображения, бинаризация или выделение границ [1, 20, 73].

В работах [116, 144] данный метод использовался для нахождения граней пластин автомобильных номерных знаков. Предполагается, что горизонтальные грани пластины номерного знака представляют собой линии, имеющие максимальную длину на изображении. Поэтому все найденные линии, длина

которых превышает заданный порог, принимаются за горизонтальные грани пластины номерного знака.

Затем осуществляется выравнивание пластины номерного знака относительно угла наклона найденных прямых и обрезка лишних частей изображения. После нахождения верхней и нижней границ строки символов выполняется выделение отдельных символов с применением других алгоритмов.

Недостаток данного подхода заключается в высокой чувствительности к различным шумам и дефектам на изображении, в результате чего прямые линии могут получаться с искажениями и разрывами, что затрудняет их обнаружение (рисунок 1.5).

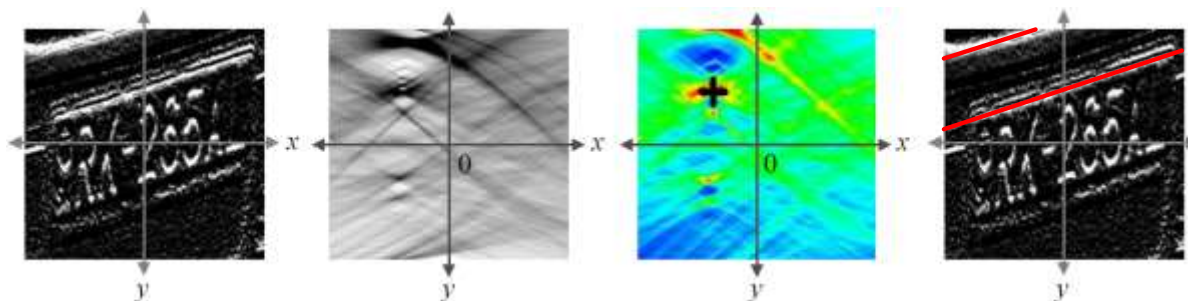


Рисунок 1.5 – Применение преобразования Хафа для выделения граней пластины номерного знака [144]

На рисунке 1.5 видно, что нижняя грань пластины номерного знака не была найдена из-за наличия шумов в данной области изображения.

Метод, основанный на выделении контура

Данный метод используется при решении задач выделения контуров объектов и символов на изображениях. Работа данного метода заключается в исследовании каждого пикселя изображения на принадлежность к отслеживаемому контуру [7, 8].

Для повышения эффективности работы данный метод требует предварительную обработку изображения, которая заключается в бинаризации. Каждый пиксель изображения имеет связь с восемью соседними пикселями, которые являются смежными и примыкают к нему, образуя прямоугольную

область 3x3 пикселей. Изображение сканируется сверху вниз, слева направо до нахождения первого встретившегося пикселя черного цвета. Затем происходит движение вокруг найденного пикселя по восьми соседним примыкающим к нему пикселям, которые анализируются на схожесть по цветовому параметру [28] (рисунок 1.6).

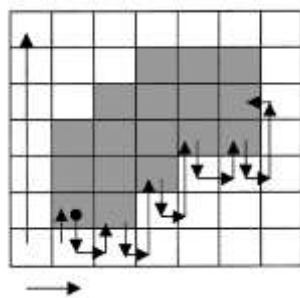


Рисунок 1.6 – Выделение контура на изображении

Движение осуществляется до тех пор, пока не будет найден очередной пиксель черного цвета. Очередной найденный пиксель принимается за центр окрестности и выполняется движение вокруг него. Данный процесс выполняется до тех пор, пока отслеживаемый контур не замкнется. После этого выполняется поиск новых черных пикселей, принадлежащих другим областям изображения.

Достоинство данного метода заключается в простой реализации и относительной независимости от яркости цвета изображения.

Недостаток данного метода заключается в низкой скорости работы и высокой чувствительности к шумовым, аффинным и проекционным искажениям, в результате чего контуры символов могут частично сливаться, что снижает точность их распознавания.

Методы, основанные на выделении связных областей

Данные методы часто применяются при решении задач выделения областей, контуров и символов на изображениях. Связной областью изображения является область, все точки которой обладают близкими значениями яркости, цвета или другими признаками. Такие области изображения можно рассматривать как

самостоятельные семантические единицы, которые обладают индивидуальными геометрическими характеристиками.

Для более эффективного применения данных методов исходное изображение преобразуется в бинарный вид. Идея данных методов заключается в следующем: выполняется сканирование изображения и при этом выполняется поиск пикселей черного цвета, которые вероятно являются частью символа. Первому встретившемуся черному пикселю присваивается порядковый номер. При обнаружении очередного пикселя черного цвета выполняется его проверка на связность с предыдущим найденным пикселем. Если связность пикселей подтверждается, то текущему пикселю присваивается тот же порядковый номер, что и у предыдущего. Иначе текущему пикселю присваивается другой порядковый номер. Порядковые номера, которые присваиваются пикселям до финальной проверки, называются промежуточными. Анализ изображения выполняется до тех пор, пока не будут найдены все связанные пиксели с одинаковыми порядковыми номерами [11] (рисунок 1.7).

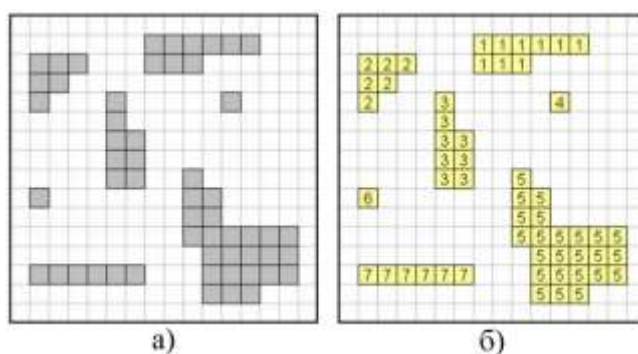


Рисунок 1.7 – Выделение связных областей: а) исходное изображение;
б) выделенные области изображения

В работе [90] данные методы применялись для решения задачи выделения символов на автомобильных номерных знаках. Достоинство методов данной категории заключается в простой реализации и относительной независимости от яркости цвета изображения.

Главный недостаток данных методов заключается в высокой чувствительности к шумовым, аффинным и проекционным искажениям, в

результате чего характерные признаки символов могут искажаться, контуры символов могут частично сливаться, что снижает точность их распознавания.

Метод, основанный на гистограммном анализе изображения

Гистограммный анализ изображения применяется во многих алгоритмах и системах при решении задач выделения символов. При анализе яркостных характеристик изображения можно определить места расположения символов.

Данный подход применялся в работах [7, 28, 144]. На начальном этапе данный метод требует предобработку изображения в виде повышения контраста, бинаризации или выделения границ. Идея данного метода заключается в следующем: выполняется сканирование изображения и при этом вычисляется среднее значение яркости пикселей в каждой строке изображения. Обычно символы на автомобильных номерных знаках и маркировках различных технических объектов изображены черным цветом на более светлом фоне, поэтому в тех местах, где нет символов, среднее значение яркости пикселей будет превышать заданный порог [25, 26]. Таким образом, находятся верхняя и нижняя границы строки символов (рисунок 1.8(а)). Далее выполняется та же операция по столбцам изображения. В промежутках между символами гистограммы показывают максимальные значения яркости пикселей (рисунок 1.8(б)).

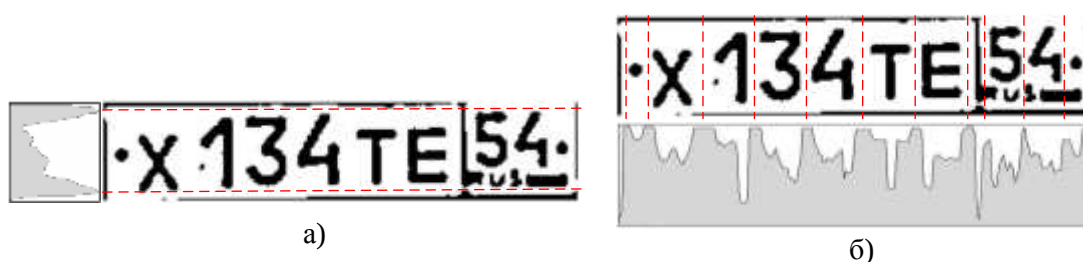


Рисунок 1.8 – Построение гистограмм распределения яркости пикселей:

а) по строкам изображения; б) по столбцам изображения

Таким образом, находятся области прямоугольной формы, которые соответствуют границам символов, что создает определенные удобства для дальнейшего их распознавания [25, 26, 28].

Недостатком данного метода является чувствительность к шумам и дефектам на изображении, особенно в промежутках между символами. В результате чего гистограммы не будут показывать ярко выраженные максимумы. Для подавления мелких шумов и усиления пространственных частот на изображении применяют повышение контраста, бинаризацию и различные виды фильтрации.

В задаче распознавания автомобильных номерных знаков данный метод достаточно эффективно использовать для определения границ пластины номерного знака, расположенной под различными углами наклона. В этом случае осуществляется построение гистограмм распределения яркости пикселей под различными углами, однако это может привести к вычислительной избыточности и снижению скорости работы данного метода.

В процессе реализации описанных методов наилучшие результаты работы были получены при использовании метода гистограммного анализа. Поэтому в данной работе было решено осуществить модификацию метода гистограммного анализа, которая позволила бы значительно снизить его недостатки.

1.4 Методы распознавания символов на изображениях

В данном разделе проводится анализ подходов, применяемых для решения задачи распознавания символов на изображениях.

После этапа выделения отдельных символов, следует последний этап – распознавание символов. Основные сложности данного этапа заключаются в том, что символы на изображении могут быть подвержены шумовым, аффинным и проекционным искажениям. Методы распознавания символов на изображениях можно разделить на три группы: шаблонные, структурные и признаковые.

Шаблонные методы распознавания

В шаблонных методах распознавания символов применяется предварительная обработка изображения, которая заключается в нормализации размера и бинаризации изображения. Принцип работы данных методов

заключается в сравнении изображения символа с множеством шаблонов, которые хранятся в базе системы [2, 34]. При определении меры подобия исследуемого символа к определенному классу шаблонов используется следующая формула:

$$R = \sum_{j=1}^n \left(\ln \frac{P_j}{1 - P_j} \right) x_j + \sum_{j=1}^n \ln(1 - P_j),$$

где R – коэффициент подобия распознаваемого символа; P_{js} – вероятность появления черного цвета в j -ом элементе распознаваемого символа; x_j – параметр интенсивности элемента распознаваемого символа.

Черные элементы изображения символа обозначаются значением $x_j = 1$, белые элементы изображения символа обозначаются значением $x_j = 0$. Исследуемое изображение относится к тому классу символа, с шаблоном которого коэффициент R имеет максимальное сходство [34].

Преимущества шаблонных методов заключается в простоте реализации и устойчивости к дефектам изображения символов.

Основной недостаток шаблонных методов заключается в невозможности распознавания символов, которые подвержены аффинным и проекционным искажениям и отличаются от заложенных в системе шаблонов.

Структурные методы распознавания

Структурные методы предполагают распознавание символов на основе анализа их топологии, которая содержит информацию о взаимном расположении отдельных составных частей символов [81]. Каждый символ можно представить в виде набора характеристик, которые могут быть представлены в виде формы графа. На исследуемом изображении производится анализ данных характеристик, на основе чего выполняется распознавание символа на изображении.

На первом этапе данные методы требуют осуществления предварительной обработки изображения символа, которая заключается в проведении процедуры скелетизации [41, 50, 85] (рисунок 1.9).

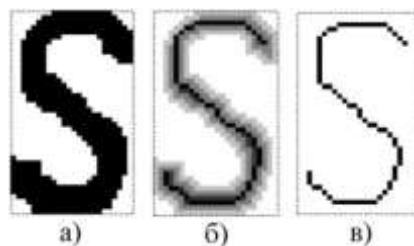


Рисунок 1.9 – Пример скелетизации символа: а) исходное изображение символа; б) удаление нескольких слоев; в) полученное изображение символа

После процедуры скелетизации каждый символ представляется в виде набора особых точек, для которых вычисляются следующие параметры: координаты особых точек; длина ребер графа; нормированное направление входов и выходов особых точек; кривизна дуги, соединяющей особые точки [173].

Так же к структурным методам можно отнести метод, основанный на выделении клики максимального размера в графе соответствия признаков. Кликой графа называется такое подмножество его вершин, в котором любые две вершины соединены ребром [27]. Признаки, выделенные на изображении символа, формируются в граф и сравниваются с эталонным графом для каждого символа. Если признаки сформированного графа соответствуют эталонным признакам, то вершины графа соединяются дугами. После этого в сформированном графе выполняется поиск наибольшей максимальной клики. Символ на исследуемом изображении относят к тому классу, граф которого содержит наибольшее число вершин [48].

Достоинством структурных методов является инвариантность относительно размера и стиля шрифтов.

Недостаток структурных методов заключается в низкой скорости работы и невозможности распознавания символов, подверженных различным искажениям и дефектам (разрывы линий, слияние соседних линий).

Признаковые методы распознавания

Признаковые методы распознавания основываются на представлении изображения символа в виде некоторого набора характеристик, называемых

признаками [23, 36]. Процесс формирования вектора признаков на основе исследуемого изображения символа, называется процессом выделения признаков.

Принцип работы данных методов заключается в сравнении вектора признаков, выделенных на исследуемом изображении, с набором эталонных векторов признаков из обучающей выборки. Выделение признаков, присущих различным символам, происходит в процессе обучения системы на эталонных изображениях. Алгоритмы данного подхода обычно предполагают использование классификаторов, которые вычисляют Евклидово расстояние между текущим вектором признаков и векторами признаков, сформированными на основе обучающей выборки.

Преимуществом признаковых методов является хорошая обобщающая способность и как следствие низкая чувствительность к изменениям масштаба, аффинным и проекционным искажениям символов. Так же данные методы обладают высокой скоростью распознавания.

Недостаток признаковых методов иногда проявляется в чувствительности к шумам и дефектам на изображениях, в результате чего признаковая структура символов может быть искажена и деформирована.

На основании анализа особенностей вышеперечисленных методов, для дальнейшей разработки алгоритмов распознавания символов были выбраны признаковые методы распознавания. Обзор методов, применяемых для выделения признаков объектов на изображениях, представлен в следующем разделе.

1.4.1 Выделение признаков

Выделение признаков является одним из основных и самых сложных процессов присущих признаковым методам распознавания символов на изображениях. Основная сложность заключается в выделении именно тех характерных признаков на изображении, с помощью которых можно эффективно выполнять классификацию символов. Ниже описаны методы, используемые для выделения признаков символов на изображениях.

Морфологический анализ

Данный метод разработан *J. Serra* и *Ю.П. Пытьевым* в 70-е годы XX века [45–47] и применяется для распознавания символов на изображениях, на основе морфологического анализа структурных характеристик каждого символа.

Данный метод предполагает изучение топологии каждого символа и описание их в виде индивидуального набора признаков: заливы, проливы, озера. Озером называется структурная часть символа, которая со всех сторон ограничена контуром. Заливом называется структурная часть символа, которая не ограничена контуром только с одной стороны. Проливом называется часть символа, которая не ограничена контуром с нескольких сторон. Количество данных признаков, их размер и взаимное расположение являются уникальным для каждого символа [77].

В работах [7, 28] данный метод использовался для распознавания символов на автомобильных номерных знаках. Структура каждого символа описывается в виде эталонного шестимерного вектора признаков, который содержит информацию о количестве заливов, проливов и озер. Исследуемое изображение анализируется на выявление признаков и сравнивается с эталонными векторами. Таким образом, происходит классификация символов на изображениях.

Достоинство данного метода заключается в низкой чувствительности к аффинным и проекционным искажениям символов.

Недостаток данного метода заключается в большой вычислительной сложности и высокой чувствительности к шумам на изображении, в результате чего структура символов может быть частично искажена.

Моменты изображений

Моментами изображения называют средневзвешенные параметры интенсивностей пикселей, которые могут быть использованы в качестве признаков для описания объектов на изображениях и для дальнейшего их распознавания. Как правило, выбираются моменты, обладающие высокой

локальной информативностью, параметры которых не меняются при различных геометрических преобразованиях изображения.

Для выделения признаков объектов на изображениях применяются геометрические моменты, описывающие геометрические характеристики объектов на изображениях. Данный метод был предложен *М.К. Ни* в работе [127]. Предложенный им метод обладает высокой устойчивостью к геометрическим преобразованиям исходного изображения. Геометрические моменты являются функциями координат каждого пикселя на изображении:

$$m_{p,q} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} x^p y^q f(x, y),$$

где x, y – координаты пикселя на изображении; $f(x, y)$ – значение интенсивности пикселя; p, q – целые числа от 0 до ∞ ; N, M – размеры изображения по вертикали и горизонтали.

Так же для решения подобных задач применяются моменты Zernike, названные в честь немецкого ученого *F. Zernike* [193]. Моменты Zernike представляют собой проекции интенсивности изображения на ортогональный полином Zernike. Полином Zernike формирует множество сложных полиномов, которые определяются координатами (x, y) и расположены в круге $x^2 + y^2 = 1$ [63].

$$V_{p,q}(x, y) = V_{p,q}(\rho, \theta) = R_{p,q}(\rho) e^{jq\theta},$$

где x, y – координаты пикселя на изображении; p, q – целые числа, определяющие уровень момента; ρ – наименьшее расстояние пикселя до центра круга; θ – угол между вектором ρ и осью X ; R – ортогональный радиальный полином.

При вычислении моментов Zernike координаты центра изображения проецируются на координаты центра круга. Пиксели, которые не входят в область круга, не используются при вычислении. Моменты Zernike вычисляются по следующей формуле:

$$Z_{p,q} = \frac{p+1}{\pi} \sum_x \sum_y f(x, y) V_{p,q}^*(\rho, \theta), \quad x^2 + y^2 \leq 1.$$

Таким образом, при использовании моментов Zernike исходное изображение преобразуется в новое изображение, в котором значение каждого пикселя (x, y) вычисляется по формуле:

$$f_Z(x, y) = \sum_{p=0}^N \sum_{q=-N}^N Z_{p,q} R_{p,q}(\rho) e^{jq\theta},$$

где N – максимальный уровень моментов Zernike.

Метод распознавания символов на основе моментов Zernike был представлен С. Кан и А. Khotanzad в работах [131, 133]. Предложенный метод обладает высокой устойчивостью к геометрическим преобразованиям исходного изображения.

На рисунке 1.10 изображено представление изображения символа «В» при использовании моментов Zernike со следующими максимальными уровнями момента N : 0, 2, 4, 6, 8, 10, 12, 14, 16, 18.

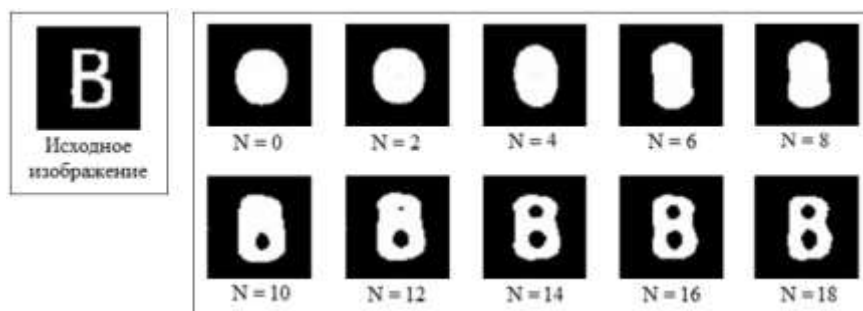


Рисунок 1.10 – Представление символа «В» при использовании моментов Zernike [133]

Так же при решении задач распознавания символов на изображениях применяются моменты Legendre [175]. Моменты Legendre вычисляются по следующей формуле:

$$L_{p,q} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} P_p(x_N) P_q(y_N) f(x, y),$$

где x, y – координаты пикселя на изображении; p, q – целые числа, определяющие уровень момента; N, M – размеры изображения по вертикали и горизонтали. Параметры P_p, P_q вычисляются по следующим формулам:

$$P_p = \frac{1}{2^p} \frac{d^p}{dx^p} (x^2 - 1)^p, \quad P_q = \frac{1}{2^q} \frac{d^q}{dy^q} (y^2 - 1)^q.$$

На рисунке 1.11 изображено представление изображения символов при использовании моментов Legendre. Слева представлены исходные изображения символов (полутонное и бинарное), а справа их представление при использовании моментов Legendre со следующими максимальными уровнями момента M : 20, 40, 60.

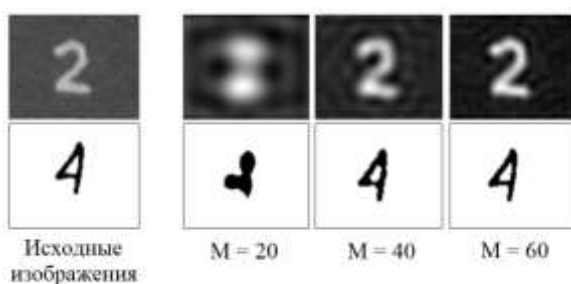


Рисунок 1.11 – Представление изображения символа при использовании моментов Legendre [175]

Достоинство использования моментов изображения для выделения признаков символов заключается в высокой устойчивости к изменению масштаба изображения и другим геометрическим преобразованиям.

Недостаток использования моментов изображения заключается в высокой чувствительности к шумам и дефектам на изображении, в результате чего символы могут неправильно классифицироваться.

Метод зон

В данном методе исходное изображение символа представляется в виде сетки равномерно распределенных ячеек. В каждой ячейке вычисляется вектор зонного описания, который формируется путем деления числа черных пикселей в ячейке на число пикселей данного цвета всего изображения. При обучении векторы зонных описаний вычисляются для каждого эталонного изображения символа. Таким образом, каждый класс символов определяется уникальным параметрическим вектором [4].

В процессе распознавания символов на изображении выполняется вычисление Евклидова расстояния между вектором зонного описания исследуемого изображения и векторами каждого эталонного изображения.

В работе [4] для изображений символов размером 10x15 пикселей выбрано оптимальное количество разделяющих зон равное 5x5 ячеек (рисунок 1.12).

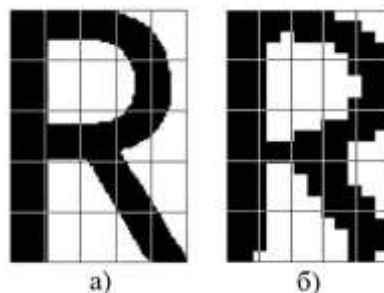


Рисунок 1.12 – Представление символа на основе применения метода зон:

а) эталонное изображение; б) сформированное тестовое изображение [4]

Данное количество разделяющих зон позволило эффективно сформировать вектор зонного описания, что обеспечило устойчивость данного метода к различной толщине контуров символов и изменениям интенсивности освещения на изображении.

Недостатком данного метода является высокая чувствительность к геометрическим, аффинным и проекционным искажениям символов на изображениях.

Вычисление ориентации штрихов

В данном методе изображение символов представляется в виде набора штрихов. Распределение штрихов и их ориентация является уникальными признаками для каждого класса символов и используются при их классификации [191].

Признаки вычисляются путем разбиения штриха по его углу ориентации на фиксированное число сегментов, которые затем используются в качестве признаков. На основе множества признаков формируются гистограммы ориентации штрихов. Гистограммы ориентации штрихов символов можно

определить по-разному: направление сегментов штриха, направление градиента, ориентация скелетного представления символа, сегмент штриха и т.д. [95, 113, 172].

На рисунке 1.13 представлен процесс вычисления ориентации штрихов символа: а) исходное изображение символа; б) контурное представление символа; в) ориентации штрихов символа; г) гистограммы локальной ориентации.



Рисунок 1.13 – Процесс вычисления ориентации штрихов символа [95]

Недостаток данного метода заключается в высокой чувствительности к геометрическим, аффинным и проекционным искажениям символов на изображениях, в результате чего, характеристики символов могут искажаться.

Линейное преобразование

В задачах распознавания символов на изображениях так же применяются методы линейного преобразования пространства признаков.

В работах [97, 101] для распознавания символов на изображениях применяется метод главных компонент (РСА). Данный метод показывает высокую точность при распознавании символов, не подверженных искажениям. Однако, при искажениях символов точность их правильного распознавания резко снижается. На рисунке 1.14 представлены примеры изображений собственных символов, полученных при использовании метода главных компонент.

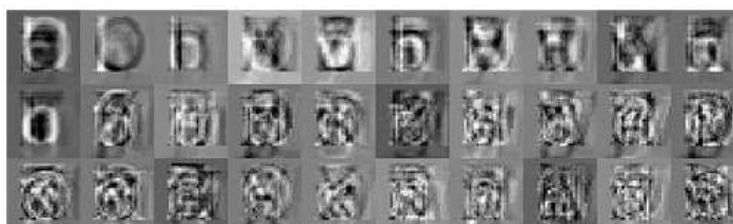


Рисунок 1.14 – Изображения собственных символов, полученных при использовании PCA [98]

В работе [158] для распознавания символов на изображениях применяется линейный дискриминантный анализ (LDA). Данный метод так же показывает высокие результаты работы при распознавании символов, не подверженных искажениям. При искажениях символов точность их правильного распознавания резко снижается. Так же данный метод очень чувствителен к изменению освещения на изображении.

Искусственные нейронные сети

При решении задач распознавания символов на изображениях часто применяются искусственные нейронные сети (ИНС) [78, 138]. Принцип работы нейронных сетей описан в разделе 1.2.2. Большинство видов ИНС обладают высокой чувствительностью к аффинным и проекционным искажениям символов на изображении, а так же не поддерживают двумерную топологию изображений, что приводит к потере взаимосвязи пространственно зависимых областей изображения. Поэтому обычно они применяются для распознавания печатных символов, не подверженных искажениям. Данные недостатки делают большинство видов ИНС не приемлемыми для решения поставленной задачи.

Наименьшую чувствительность к искажениям объектов на изображениях проявляют сверточные нейронные сети (СНС). Принцип работы СНС описан в разделе 1.2.2. Для распознавания символов на изображениях СНС применялись в работах [74, 119], а так же в системе Google Street View [115], разработанной компанией Google. Данная система ориентирована на распознавание номеров домов на изображениях и показывает точность распознавания на уровне 96%.

Важным свойством СНС является их двумерная топология, которая очень хорошо подходит для обработки изображений. В задаче распознавания символов на автомобильных номерных знаках данное свойство позволяет не использовать процесс выделения отдельных символов, а подать изображение обнаруженного

автомобильного номерного знака целиком на вход СНС. В областях расположения символов СНС будет показывать максимальные отклики, соответствующие определенным классам символов. Таким образом, можно значительно снизить вычислительные затраты и повысить время работы системы.

На основании проведенного анализа для решения задачи распознавания символов на автомобильных номерных знаках в данной работе было принято решение использовать алгоритмы, основанные на сверточных нейронных сетях.

1.5 Технические характеристики систем распознавания автомобильных номерных знаков

Современные системы распознавания автомобильных номерных знаков представляют собой специализированные аппаратно-программные комплексы и включают в себя следующие модули (рисунок 1.15):

- регистрирующее устройство;
- аппаратный модуль видеозахвата;
- программный модуль детектирования автомобильного номерного знака;
- программный модуль нормализации изображения автомобильного номерного знака и выделения отдельных символов;
- программный модуль распознавания автомобильного номерного знака;
- внешняя база данных.

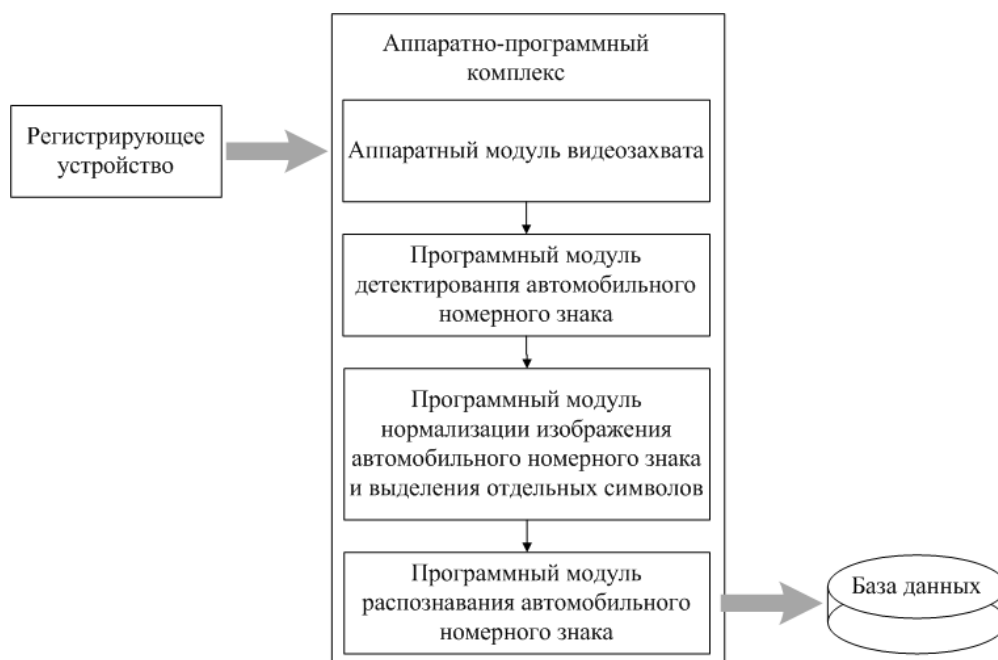


Рисунок 1.15 – Общая структура систем распознавания автомобильных номерных знаков

Регистрирующее устройство устанавливается над контролируемой полосой дорожного движения для осуществления видеонаблюдения. Регистрирующее устройство поддерживает связь с аппаратно-программным комплексом. Аппаратная часть данного комплекса осуществляет захват видеосигнала и его оцифровку, а программная часть данного комплекса осуществляет обработку полученных данных. Программное обеспечение состоит из нескольких программных модулей, осуществляющих детектирование автомобильных номерных знаков, выделение отдельных символов и их распознавание. Далее осуществляется запись распознанных автомобильных номерных знаков в базу данных.

Проведен анализ наиболее популярных и востребованных на российском рынке программных систем, применяемых для распознавания автомобильных номерных знаков. Данные системы являются коммерческими, используемые в них алгоритмы не публикуются в подробной форме. В таблице 1.1 представлены технические характеристики систем распознавания автомобильных номерных знаков. Серым цветом выделены лучшие показатели среди представленных систем.

Таблица 1.1 – Технические характеристики систем распознавания автомобильных номерных знаков

Название системы	Точность распознавания, (%)	Минимальная высота символов, (пиксели)
«Авто-Инспектор» [53]	95	10
«Авто-Интеллект» [54]	90	не указано
«Авто-Номер» [55]	95	14
«Авто-Контроль» [56]	95	14
«Senesys-Avto» [57]	98	14
«Орион-Авто» [58]	95	не указано
«SL-Traffic» [59]	90	25
«Дигнум-Авто» [60]	90	не указано
«CarFlow II» [61]	93-95	не указано
«Интегра-С» [62]	90	не указано
«Auto-TRASSIR» [63]	90	не указано
«SecurOS Auto» [64]	96	не указано
«Milestone XProtect» [65]	не указано	20
«AutoCode» [66]	96	не указано
«Macroscop» [67]	не указано	9
«ТелеВизард-Авто» [68]	90	100
«CVS Авто» [69]	95	не указано
«MegaCar» [70]	95	не указано
«Авто-Ураган» [71]	97	не указано

В таблице 1.2 представлены параметры регистрации, при которых эффективность работы систем сохраняется на высоком уровне. Серым цветом выделены лучшие показатели среди представленных систем.

Таблица 1.2 – Параметры регистрации, при которых эффективность работы систем сохраняется на высоком уровне

Название системы	Вертикальный угол отклонения (градусы)	Горизонтальный угол отклонения (градусы)	Угол отклонения на плоскости (градусы)	Освещенность (лк)

«Авто-Инспектор»	±40	±30	±15	от 50 до 1000
«Авто-Интеллект»	±30	±20	±10	от 50 до 1000
«Авто-Номер»	±30	±30	не указано	от 50 до 1000
«Авто-Контроль»	±30	±30	не указано	от 50 до 1000
«Senesys-Avto»	±30	±30	не указано	от 50 до 1000
«Орион-Авто»	±20	±20	±10	от 50 до 1000
«SL-Traffic»	±60	±30	±10	от 50 до 1000
«Дигнум-Авто»	±40	±40	не указано	от 50 до 1000
«CarFlow II»	±45	±45	±15	от 50 до 1000
«Интегра-С»	±30	±30	±15	от 50 до 1000
«Auto-TRASSIR»	не указано	не указано	не указано	от 50 до 1000
«SecurOS Auto»	±40	±30	не указано	от 50 до 1000
«Milestone XProtect»	±40	±30	±10	от 50 до 1000
«AutoCode»	±40	±30	±10	от 50 до 1000
«Macroscop»	±40	±30	±10	не указано
«ТелеВизард-Авто»	не указано	не указано	±5	от 50 до 1000
«CVS Авто»	±45	±45	±10	от 50 до 1000
«MegaCar»	±30	±30	±10	от 50 до 1000
«Авто-Ураган»	±30	±20	±10	от 50 до 1000

В таблице 1.3 Представлены заявленные минимальные технические требования для систем распознавания автомобильных номерных знаков.

Таблица 1.3 – Минимальные технические требования для систем распознавания автомобильных номерных знаков

Название системы	Минимальные технические требования
Авто-Инспектор	ЦП: IntelCore2 2.0 ГГц; ОЗУ: 512 Мб; Видеоадаптер: 256 Мб.
Дигнум- Авто	ЦП: IntelCore2 3.0 ГГц; ОЗУ: 1 Гб; Видеоадаптер: 1 Гб.
Авто-Номер	ЦП: IntelCore2 3.0 ГГц; ОЗУ: 1 Гб; Видеоадаптер: 256 Мб.
Авто-Контроль	
Senesys-Avto	
Milestone XProtect	ЦП: IntelCore2 3.1 ГГц; ОЗУ: 4 Гб; Видеоадаптер: 256 Мб.

AutoCode	
Орион-Авто	ЦП: IntelCore2 3.1 ГГц; ОЗУ: 2 Гб; Видеоадаптер: 256 Мб.
Авто-Интеллект	не указано
CarFlow II	
Интегра-С	
Auto-TRASSIR	
SecurOS Auto	
Macroscop	
ТелеВизард-Авто	
SL-Traffic	
CVS Авто	
MegaCar	
Авто-Ураган	

Из таблиц 1.1 и 1.2 видно, что технические характеристики большинства современных систем не в полной мере удовлетворяют требованиям к распознаванию автомобильных номерных знаков в реальных условиях эксплуатации. В среднем точность правильного распознавания данных систем находится в диапазоне от 90% до 96%. Один из главных недостатков вышеперечисленных систем заключается в том, что при изменении углов регистрации более 30-40 градусов точность распознавания существенно падает. Данные недостатки снижают эффективность распознавания при следующих условиях:

- при наклонном закреплении пластин автомобильных номерных знаков;
- при осуществлении автомобилем маневрирования;
- при расположении регистрирующего устройства сбоку от полосы движения;
- при использовании системы на объектах с ограниченным выбором места для крепления регистрирующего устройства.

На основании проведенного анализа можно сделать вывод, что на сегодняшний день существует потребность в разработке алгоритмов

распознавания автомобильных номерных знаков, обладающих высокой степенью инвариантности к аффинным и проекционным искажениям.

Из таблиц 1.1 и 1.2 были выделены наилучшие показатели технических характеристик систем, на основе которых составлены минимальные требования для разрабатываемых алгоритмов:

- Точность распознавания: не менее 98%;
- Вертикальный угол отклонения: не менее $\pm 60^\circ$;
- Горизонтальный угол отклонения: не менее $\pm 45^\circ$;
- Угол отклонения на плоскости: не менее $\pm 15^\circ$;
- Освещенность: от 50 до 1000 люкс (лк);
- Возможность сохранения информации в базе данных.

1.6 Цель и задачи исследования

Целью диссертационной работы является разработка алгоритмов и программных средств, позволяющих повысить эффективность распознавания символов на сложном фоне, подверженных шумовым, аффинным и проекционным искажениям.

Для достижения поставленных целей необходимо последовательное решение следующих задач:

1. Исследовать методы и алгоритмы, применяемые на различных стадиях процесса распознавания символов на изображениях со сложной фоновой структурой.

2. Разработать алгоритм, обеспечивающий высокую эффективность работы при детектировании символов, подверженных различным шумовым, аффинным и проекционным искажениям.

3. Создать алгоритм нормализации изображений символов, обеспечивающий возможность определения точных границ строки символов, расположенной под различными углами наклона на изображении.

4. Разработать алгоритм, обеспечивающий высокую эффективность работы при распознавании символов, подверженных шумовым, аффинным и проекционным искажениям.

5. Реализовать разработанные алгоритмы в виде программной системы, предназначенной для распознавания символов на изображениях со сложной фоновой структурой, и провести вычислительные эксперименты с целью оценки их качества и эффективности.

В процессе исследований было принято решение осуществлять реализацию и апробацию алгоритмов на примере распознавания автомобильных номерных знаков, так как данная область применения является одной из наиболее интенсивно развивающихся, актуальной и в полной мере соответствует поставленной задаче. Алгоритмы должны обеспечивать распознавание автомобильных номерных знаков, соответствующих ГОСТ Р 50577 – 93, группе 1, типу 1 и удовлетворяющих требованиям Венской Конвенции о дорожном движении от 8.11.1968 г. [24].

1.7 Основные результаты и выводы по главе 1

1. В данной главе проведен аналитический обзор методов и алгоритмов, применяемых на различных стадиях процесса распознавания символов и маркировок различных объектов на изображениях со сложной фоновой структурой. На основании проведенного анализа можно сделать вывод, что в настоящее время не существует методов и систем, в полной мере удовлетворяющих требуемым характеристикам при детектировании, нормализации и распознавании символов, подверженных шумовым, аффинным и проекционным искажениям. В процессе исследования было принято решение осуществлять реализацию и апробацию алгоритмов на примере распознавания автомобильных номерных знаков.

2. Проведенный анализ позволил сделать вывод о том, что для решения задачи детектирования области расположения символов на изображениях является эффективным применение алгоритмов, основанных на СНС. Этот факт

объясняется тем, что СНС обладают высокой степенью инвариантности к различным шумовым, аффинным и проекционным искажениям.

3. Для осуществления нормализации изображений символов является эффективным применение алгоритмов, основанных на построении гистограмм распределения яркости пикселей. Преимущества данного подхода заключаются в возможности определения точных границ пластины номерного знака, расположенного под различными углами наклона на изображении.

4. Для решения задачи распознавания символов является эффективным применение алгоритмов, основанных на СНС. Данный выбор обусловлен тем, что при распознавании символов на основе СНС отсутствует необходимость в осуществлении предварительной сегментации символов, что повышает скорость работы алгоритма. Так же важным аргументом в пользу выбора СНС является тот факт, что они обладают высокой устойчивостью к различным искажениям входных данных и способны выполнять распознавание символов с частично искаженной визуальной структурой.

ГЛАВА 2. РАЗРАБОТКА АЛГОРИТМОВ РАСПОЗНАВАНИЯ СИМВОЛОВ НА СЛОЖНОМ ФОНЕ

В данной главе приводится подробное описание предложенных алгоритмов на примере их применения для распознавания автомобильных номерных знаков. Предложен алгоритм детектирования области расположения символов на изображениях, основанный на сверточных нейронных сетях. Реализована модификация алгоритма нормализации изображений символов, основанного на построении гистограмм распределения яркости пикселей. Предложен алгоритм распознавания символов, основанный на сверточных нейронных сетях. Проведен обзор алгоритмов обучения сверточных нейронных сетей, в результате их реализации и экспериментальных исследований выбран наиболее оптимальный алгоритм обучения.

2.1 Разработка алгоритма детектирования области расположения символов на изображениях

При решении сложных задач классификации возникает необходимость в реализации алгоритмов, эффективно решающих поставленные задачи. Однако качество реализованных алгоритмов зачастую оставляет желать лучшего. В подобных ситуациях имеет смысл реализовать алгоритмическую композицию (boosting – улучшение), состоящую из нескольких алгоритмов, каждый из которых старается компенсировать ошибки классификации, полученные на предыдущих итерациях [12–14].

Во многих исследованиях наблюдалось, что при наращивании алгоритмических композиций происходит интенсивное уменьшение частоты ошибок на независимых тестовых наборах данных, а так же увеличение скорости работы системы. Данные факты изменили сложившееся представление о необходимости ограничения сложности алгоритмов. В итоге многими исследователями был сделан вывод, что алгоритмические композиции не снижают эффективность работы системы, а лишь сглаживает погрешность базовых алгоритмов [13, 163].

На сегодняшний день построение алгоритмических композиций, в которых различные алгоритмы компенсируют недостатки друг друга, является одним из наиболее перспективных направлений машинного обучения. Достоинства данного подхода заключаются в универсальности, гибкости, высокой обобщающей способности и возможности получения качественного обучения, недостижимого для отдельных базовых алгоритмов.

Детектирование области расположения символов, в данном случае автомобильных номерных знаков является достаточно сложной задачей в связи с наличием на изображениях сложной фоновой структуры и шумовых помех. При решении данной задачи возникает необходимость реализации алгоритма, выполняющего несколько этапов классификации и работающего по принципу последовательной классификации.

На основании этого, было принято решение разработать алгоритм детектирования автомобильных номерных знаков в виде алгоритмической композиции, состоящей из двух СНС различной конфигурации. Алгоритмическая композиция строилась по принципу последовательной классификации, где базовые алгоритмы выполняют свою работу по очереди и каждый следующий алгоритм старается компенсировать ошибки, полученные на предыдущих итерациях. В данном случае первая сверточная нейронная сеть (СНС-1) выполняет предварительную классификацию, вторая сверточная нейронная сеть (СНС-2) выполняет итоговую классификацию, что бы скомпенсировать ошибки классификации, полученные первой нейронной сетью.

Одной из самых сложных задач при реализации нейросетевых алгоритмов является разработка оптимальной структуры нейронной сети и подбор различных параметров: количество слоев и нейронов, тип активационной функции, выбор алгоритма обучения. В процессе исследований были реализованы СНС с различной конфигурацией и параметрами. Наилучшие результаты работы были получены при использовании СНС, описанных в следующих разделах.

2.1.1 Сверточная нейронная сеть для предварительной

классификации

Для предварительной классификации автомобильных номерных знаков на изображениях разработана сеть СНС-1, представленная на рисунке 2.1. Данная нейронная сеть имеет более простую структуру, чем СНС-2: меньшее количество плоскостей, нейронов и связей. Это обеспечивает более высокое быстродействие и снижение количества вычислительных процессов, что является особенно важным на этапе предварительной классификации.

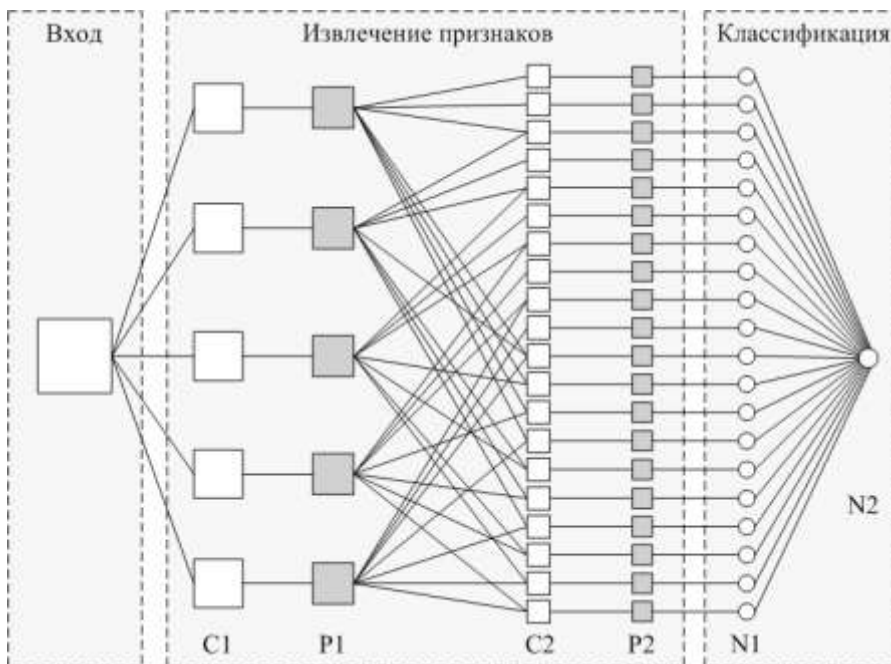


Рисунок 2.1 – Структура СНС-1. C1, C2 – слои свертки; P1, P2 – слои подвыборки; N1, N2 – слои из обычных нейронов

Входной слой имеет размер 28x52 нейрона и предназначен для подачи входного образа в нейронную сеть.

Второй слой C1 является слоем свертки, состоит из 5 сверточных плоскостей размером 24x46 нейронов. Каждая плоскость данного слоя имеет собственную матрицу синаптических коэффициентов плюс нейронное смещение и таким образом выполняет пять сверток входного изображения. Значения нейронов текущего слоя вычисляются при последовательном сканировании входного слоя рецептивным полем и умножении значений входящих в него нейронов на матрицу весовых коэффициентов. Рецептивные поля смежных

нейронов частично пересекаются и в результате этого нейроны извлекают из входного слоя свои характерные признаки, независимо от их расположения, что обеспечивает устойчивость нейронной сети к смещению и повороту входного изображения. Размер локального рецептивного поля равен 5×7 нейронов. Каждой плоскости соответствует $5 \times 7 = 35$ весовых коэффициентов и нейронное смещение. Таким образом, данный слой содержит 5520 нейронов и $(35+1) \times 5 = 180$ весовых коэффициентов.

Размер плоскостей сверточного слоя определяется следующими формулами:

$$s_k = s_p - N + 1, \quad h_k = h_p - M + 1, \quad (2.1)$$

где h_k, s_k – параметры высоты и ширины плоскости сверточного слоя; h_p, s_p – параметры высоты и ширины плоскости предыдущего слоя; M, N – параметры высоты и ширины локального рецептивного поля.

Функционирование нейрона сверточного слоя определяется следующей формулой:

$$y_q^{(i,j)} = a_q + \sum_{s=1}^N \sum_{t=1}^M w_{q,s,t} x_{q-1}^{(i,j)}, \quad (2.2)$$

где, $y_q^{(i,j)}$ – нейрон сверточного слоя, q -ой плоскости; a_q – нейронное смещение q -ой плоскости; N, M – ширина и высота локального рецептивного поля; $x_{q-1}^{(i,j)}$ – выходные значения нейронов предыдущего слоя; $w_{q,s,t}$ – синаптические коэффициенты.

Третий слой P1 является слоем подвыборки состоит из 5 подвыборочных плоскостей. Каждая плоскость имеет связь только с одной соответствующей ей плоскостью предыдущего слоя C1. Данный слой обеспечивает локальное усреднение плоскостей предыдущего слоя, поэтому размер плоскостей данного слоя вдвое меньше, чем в предыдущем слое и равен 12×23 нейрона. Рецептивные поля смежных нейронов не пересекаются и имеют размер 2×2 нейрона, что обеспечивает частичную инвариантность нейронной сети к изменению масштаба входного изображения. Каждой плоскости соответствует один синаптический

коэффициент и нейронное смещение. Каждый нейрон данного слоя вычисляет среднее значение нейронов из соответствующей ему области нейронов предыдущего слоя. Далее полученное значение умножается на синаптический коэффициент и пропускается через функцию активации. Таким образом, данный слой содержит 1380 нейронов и всего 10 синаптических весовых коэффициентов.

После осуществления операции подвыборки повышается инвариантность нейронной сети к смещению, изменению масштаба входного изображения и точное расположение признаков объектов на изображении становится менее важным. Размер плоскостей подвыборочного слоя определяется следующими формулами:

$$s_k = \frac{s_p}{N}, \quad h_k = \frac{h_p}{M}, \quad (2.3)$$

где h_k, s_k – параметры высоты и ширины плоскости подвыборочного слоя; h_p, s_p – параметры высоты и ширины плоскости предыдущего слоя; M, N – параметры высоты и ширины локального рецептивного поля.

Функционирование нейрона подвыборочного слоя определяется следующей формулой:

$$y_q^{(i,j)} = a_q + \frac{1}{NM} w_q \sum_{s=1}^N \sum_{t=1}^M x_{q-1}^{(i,j)}, \quad (2.4)$$

где, $y_q^{(i,j)}$ – нейрон подвыборочного слоя, q -ой плоскости; a_q – нейронное смещение q -ой плоскости; M, N – параметры высоты и ширины локального рецептивного поля; $x_{q-1}^{(i,j)}$ – выходные значения нейронов предыдущего слоя; w_q – синаптический коэффициент.

Четвертый слой С2 является слоем свертки и состоит из 20 сверточных плоскостей размером 10x20 нейронов. Рецептивные поля смежных нейронов частично пересекаются. Для более эффективного извлечения характерных признаков и получения различных наборов данных, плоскости данного слоя перекрестно связаны с плоскостями предыдущего слоя Р1. Первые 11 плоскостей данного слоя формируются по следующему принципу, каждая из трех соседних

плоскостей данного слоя связана с одной соответствующей им плоскостью предыдущего слоя. Таким образом, осуществляется свертка каждой из 5 плоскостей предыдущего слоя тремя различными матрицами синаптических коэффициентов. Другие 10 плоскостей формируются по следующему принципу, каждая из 10 плоскостей имеет связь с двумя плоскостями предыдущего слоя и осуществляет их свертку. Таким образом, данные 10 плоскостей формируются путем суммирования результатов свертки каждой возможной пары плоскостей предыдущего слоя.

Данный слой не является полносвязным по причине того, что плоскости должны извлекать различные (желательно новые) характерные признаки и получать на вход различные наборы данных.

Распределение связей между слоями P1 и C2 в СНС-1 подбиралось экспериментальным путем. В таблице 2.1 представлено распределение связей между слоями P1 и C2 в СНС-1. По горизонтали отображаются плоскости слоя C2, по вертикали плоскости слоя P1.

Таблица 2.1 – Распределение связей между слоями P1 и C2 в СНС-1

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	x	x	x								x	x	x	x						
2			x	x	x						x				x	x	x			
3					x	x	x					x			x			x	x	
4							x	x	x				x			x		x		x
5									x	x	x			x			x		x	x

В результате выполнения этих операций обеспечивается способность нейронной сети объединять различные виды характеристик, которые в меньшей степени зависят от искажений входного изображения.

Размер локального рецептивного поля равен 3×4 нейрона. Каждой плоскости соответствует $3 \times 4 = 12$ весовых коэффициентов и нейронное смещение. Таким образом, данный слой содержит 4000 нейронов и $(12 + 1) \times 20 = 260$ весовых коэффициентов.

Пятый слой P2 является слоем подвыборки, состоит из 20 подвыборочных плоскостей. Размер плоскостей данного слоя в два раза меньше размера плоскостей предыдущего слоя и равен 5x10 нейронов. Каждая плоскость имеет связь только с одной соответствующей ей плоскостью предыдущего слоя. Рецептивные поля смежных нейронов не пересекаются и имеют размер 2x2 нейрона. Каждой плоскости соответствует один синаптический коэффициент и нейронное смещение. Данный слой содержит 1000 нейронов и 40 синаптических весовых коэффициентов.

Пять первых слоев СНС-1 предназначены для извлечения признаков изображения. Следующие два слоя являются слоями классификации.

Шестой слой N1 состоит из 20 нейронов. Каждый нейрон данного слоя имеет связь со всеми нейронами только одной соответствующей ему плоскости предыдущего слоя P2. Каждый нейрон вычисляет взвешенное суммирование 50 соответствующих ему входных параметров, добавляет нейронное смещение и полученный результат передает через функцию активации. Таким образом, данный слой содержит 1020 синаптических весовых коэффициентов.

Седьмой слой N2 является выходным слоем и состоит из одного нейрона. Роль данного слоя заключается в вычислении окончательного результата классификации. Выходные значения этого нейрона находятся в диапазоне от -1 до $+1$, что означает отсутствие или наличие пластины автомобильного номерного знака на изображении. Нейрон данного слоя имеет связи со всеми нейронами предыдущего слоя N1 и производит вычисление Евклидова расстояния между вектором выходных значений и параметрическим вектором:

$$E_k = \frac{1}{2} \sum_{j=0}^N (t_{kj} - x_{kj})^2, \quad (2.5)$$

где, x – вектор входных значений, w – параметрический вектор, y – выход сети.

Благодаря использованию общих синаптических весовых коэффициентов в плоскостях сверточных нейронных сетей достигается уменьшение количества настраиваемых параметров. СНС-1 содержит 13377 нейронов и 1530 синаптических коэффициентов.

В таблице 2.2 представлена конфигурация СНС-1.

Таблица 2.2 – Конфигурация СНС-1

Слой	1	2	3	4	5	6	7
Число плоскостей	1	5	5	20	20	-	-
Размер плоскостей	28x52	24x46	12x23	10x20	5x10	-	-
Количество нейронов в слое	1456	5520	1380	4000	1000	20	1
Размер локального рецептивного поля	-	5x7	2x2	3x4	2x2	-	-
Количество синаптических коэффициентов	-	180	10	260	40	1020	20

При использовании классических нейронных сетей каждый участок изображения рассчитывается отдельно от всех остальных участков, это приводит к большим объемам вычислений и снижению скорости работы системы. В сверточных нейронных сетях данный недостаток устраняется использованием общих синаптических коэффициентов и так называемого метода общих вычислений (рисунок 2.2).



Рисунок 2.2 – Метод общих вычислений

Локальные рецептивные поля сверточных слоев частично наслаиваются друг на друга и имеют общие значения нейронов. Поэтому можно не вычислять значения всех нейронов локального рецептивного поля, а использовать значения, вычисленные на предыдущих итерациях. Использование этого метода позволяет значительно снизить время и объем вычислений.

2.1.2 Сверточная нейронная сеть для итоговой классификации

Для итоговой классификации автомобильных номерных знаков на изображениях разработана сеть СНС-2, представленная на рисунке 2.3. Данная нейронная сеть имеет более сложную структуру: большее количество слоев, нейронов и связей. Это увеличивает количество вычислительных процессов, но при этом обеспечивает более высокую степень обобщающей способности нейронной сети.

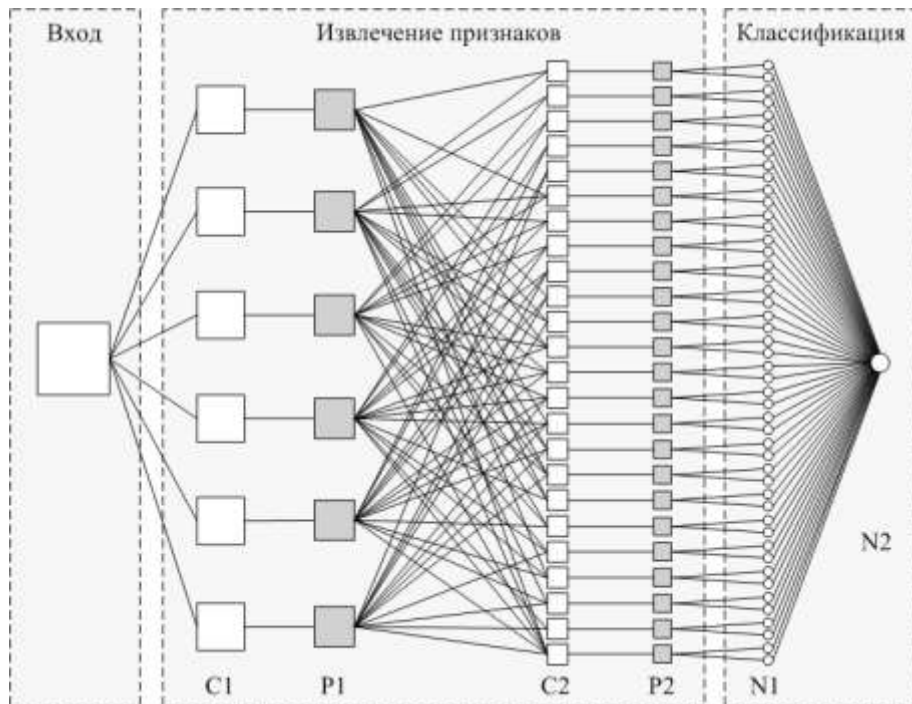


Рисунок 2.3 – Структура СНС-2. C1, C2 – слои свертки; P1, P2 – слои подвыборки; N1, N2 – слои из обычных нейронов

Входной слой имеет размер 28x52 нейрона.

Второй слой C1 является слоем свертки и состоит из 6 сверточных плоскостей. Каждая плоскость данного слоя имеет размер 24x46 нейронов, собственную матрицу синаптических коэффициентов и нейронное смещение. Рецептивные поля нейронов частично пересекаются. Размер локального рецептивного поля равен 5x7 нейронов. Каждой плоскости соответствует $5 \times 7 = 35$ весовых коэффициентов и нейронное смещение. Таким образом, данный слой содержит 6624 нейрона и $(35+1) \times 6 = 216$ весовых коэффициентов.

Размер плоскостей сверточного слоя определяется в соответствии с формулами (2.1).

Функционирование нейрона сверточного слоя определяется формулой (2.2).

Третий слой P1 является слоем подвыборки, состоит из 6 подвыборочных плоскостей. Данный слой обеспечивает локальное усреднение плоскостей предыдущего слоя, поэтому размер плоскостей данного слоя вдвое меньше, чем в предыдущем слое и равен 12x23 нейрона. Каждая плоскость имеет связь только с одной соответствующей ей плоскостью предыдущего слоя C1. Рецептивные поля смежных нейронов не пересекаются и имеют размер 2x2 нейрона. Каждой плоскости соответствует один синаптический коэффициент и нейронное смещение. Таким образом, данный слой содержит 1656 нейронов и всего 12 синаптических весовых коэффициентов.

Размер плоскостей подвыборочного слоя определяется в соответствии с формулами (2.3).

Функционирование нейрона подвыборочного слоя определяется формулой (2.4).

Четвертый слой C2 является слоем свертки и состоит из 24 сверточных плоскостей размером 10x20 нейронов. Рецептивные поля смежных нейронов частично пересекаются. В отличие от СНС-1 данный слой содержит более сложный принцип распределения связей. Первые 5 плоскостей данного слоя связаны со всеми соседними парами плоскостей предыдущего слоя P1. Следующие 6 плоскостей получают данные от всех соседних подмножеств, содержащих по 3 плоскости предыдущего слоя. Затем следуют 6 плоскостей, которые связаны со всеми соседними подмножествами, содержащими по 4 плоскости предыдущего слоя. Следующие 6 плоскостей получают данные от некоторых перемежающихся подмножеств, содержащих по 3 плоскости предыдущего слоя. И последняя плоскость получает данные от всех плоскостей предыдущего слоя.

Распределение связей между слоями P1 и C2 в СНС-2 подбиралось экспериментальным путем. В таблице 2.3 представлено распределение связей

между слоями P1 и C2 в СНС-2. По горизонтали отображаются плоскости слоя C2, по вертикали плоскости слоя P1.

Таблица 2.3 – Распределение связей между слоями P1 и C2 в СНС-2

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	x					x				x	x	x			x	x	x	x			x		x	x
2	x	x				x	x				x	x	x			x	x	x	x			x		x
3		x	x			x	x	x				x	x	x			x		x	x			x	x
4			x	x			x	x	x			x	x	x	x			x		x	x			x
5				x	x			x	x	x			x	x	x	x			x		x	x		x
6					x				x	x	x			x	x	x	x			x		x	x	x

Размер локального рецептивного поля равен 4×6 нейронов. Каждой плоскости соответствует $3 \times 4 = 12$ весовых коэффициентов и нейронное смещение. Таким образом, данный слой содержит 4800 нейронов и $(12 + 1) \times 24 = 312$ весовых коэффициентов.

Пятый слой P2 является слоем подвыборки, состоит из 24 подвыборочных плоскостей. Размер плоскостей данного слоя в два раза меньше размера плоскостей предыдущего слоя и равен 5×10 нейронов. Каждая плоскость имеет связь только с одной соответствующей ей плоскостью предыдущего слоя C2. Рецептивные поля смежных нейронов не пересекаются и имеют размер 2×2 нейрона. Каждой плоскости соответствует один синаптический коэффициент и нейронное смещение. Данный слой содержит 1200 нейронов и 48 синаптических весовых коэффициентов.

Шестой слой N1 состоит из 48 нейронов. В отличие от СНС-1 каждый нейрон данного слоя имеет связи не с одной плоскостью, а с двумя плоскостями предыдущего слоя P2. Каждый нейрон вычисляет взвешенное суммирование 100 соответствующих ему входных параметров, добавляет нейронное смещение и полученный результат передает через функцию активации. Таким образом, данный слой содержит 4848 синаптических весовых коэффициентов.

Седьмой слой N2 является выходным слоем и состоит из одного нейрона. Нейрон данного слоя имеет связи со всеми нейронами предыдущего слоя. Роль

данного слоя заключается в вычислении окончательного результата классификации. Выходные значения этого нейрона находятся в диапазоне от -1 до $+1$, что означает отсутствие или наличие пластины автомобильного номерного знака на изображении.

Значение нейрона выходного слоя вычисляется в соответствии с выражением (2.5).

СНС-2 содержит 15785 нейронов и 5484 синаптических коэффициентов.

В таблице 2.4 представлена конфигурация СНС-2.

Таблица 2.4 – Конфигурация СНС-2

Слой	1	2	3	4	5	6	7
Число плоскостей	1	6	6	24	24	-	-
Размер плоскостей	28x52	24x46	12x23	10x20	5x10	-	-
Количество нейронов в слое	1456	6624	1656	4800	1200	48	1
Размер локального рецептивного поля	-	5x7	2x2	3x4	2x2	-	-
Количество синаптических коэффициентов	-	216	12	312	48	4848	48

2.1.3 Алгоритм детектирования области расположения символов

Алгоритм детектирования области расположения символов (области расположения номерного знака) представляет собой алгоритмическую композицию, состоящую из двух сверточных нейронных сетей: СНС-1 и СНС-2. Сеть СНС-1 выполняет предварительную классификацию автомобильных номерных знаков на изображениях, а СНС-2 выполняет итоговую классификацию. Таким образом, компенсируются ошибки классификации, полученные сетью СНС-1.

При разработке данного алгоритма были проведены исследования на чувствительность разработанных СНС к различным масштабам исходного изображения. Для проведения исследований заранее выполнялось обучение СНС. В качестве исходного изображения использовалось изображение транспортного средства с наличием сложной фоновой структуры (рисунок 2.5). Исходное

изображение было отмасштабировано в десяти различных вариантах с коэффициентами от 0,6 до 1,5. Таким образом, получилось 10 изображений различного масштаба, каждое из которых подавалось на вход нейронных сетей для обработки.

Полученные результаты показали, что при сканировании входного изображения отклики нейронных сетей в местоположениях автомобильных номерных знаков и в некоторой их окрестности образуют максимальные значения. На рисунке 2.4 изображены диаграммы, представляющие собой матрицы откликов нейронной сети для масштабов изображений от 0.6 до 1.3. В тех областях изображения, где отклики нейронной сети приближаются к значению 1.0, расположены признаки соответствующие автомобильному номерному знаку.

Как видно на рисунке 2.4, при различных масштабах изображения отклики нейронных сетей имеют различные параметры. Это связано с тем, что при различных масштабах изображения характерные признаки объектов могут иметь отличающуюся визуальную структуру и нейронная сеть на них может реагировать по-разному. Автомобильный номерной знак на изображении может быть представлен в слишком большом или слишком маленьком масштабе, который не в полной мере подходит для качественной классификации. Процесс масштабирования изображения может скомпенсировать данный недостаток.

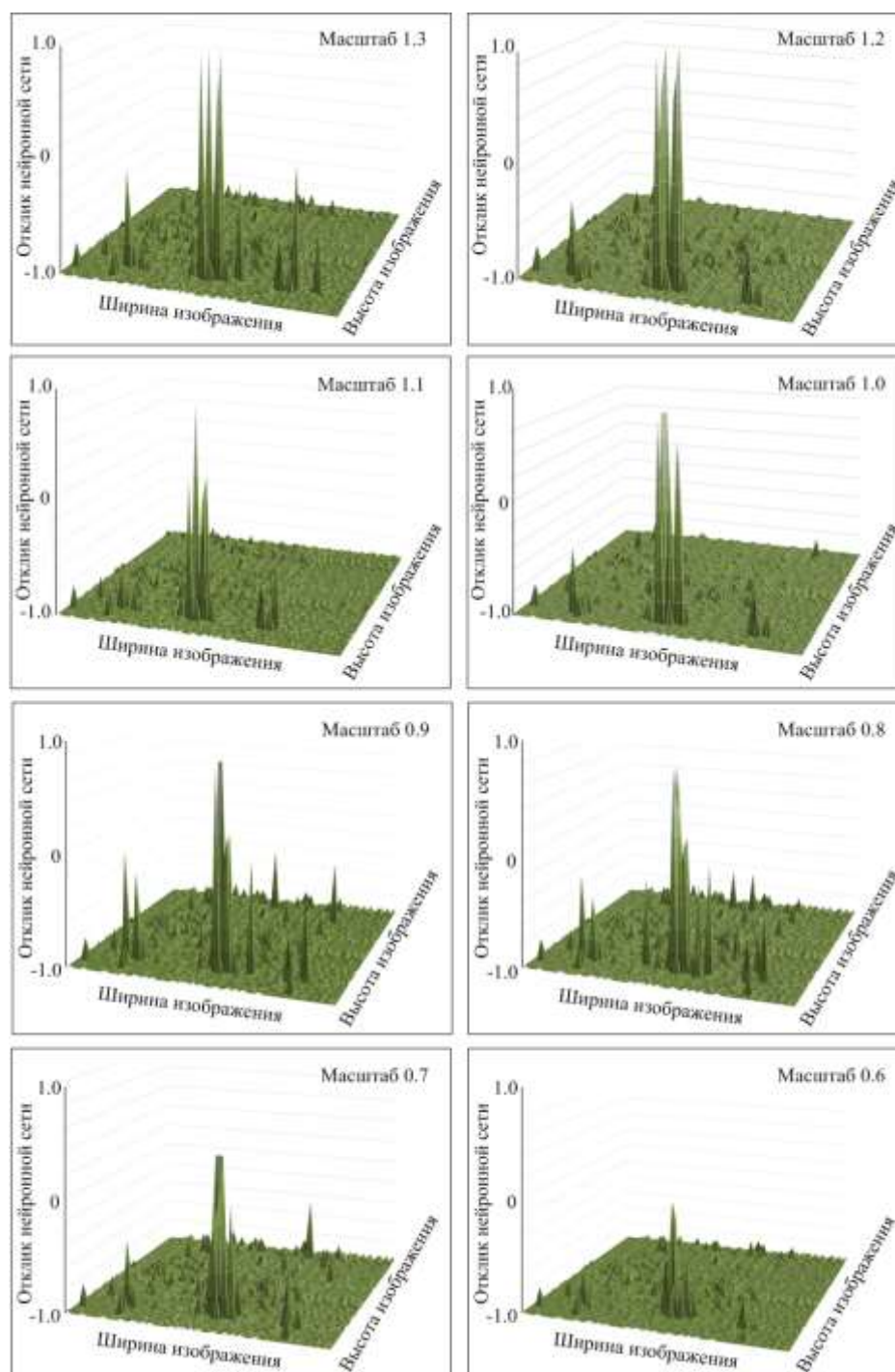


Рисунок 2.4 – Отклики СНС при различных масштабах изображения

Наиболее эффективное распознавание получилось у масштабов 0.8, 0.9, 1.0, 1.1, 1.2 (рисунок 2.5), так как изображение автомобильного номерного знака в этих масштабах не так сильно подвержено искажениям и хорошо соответствует размеру входа нейронной сети.

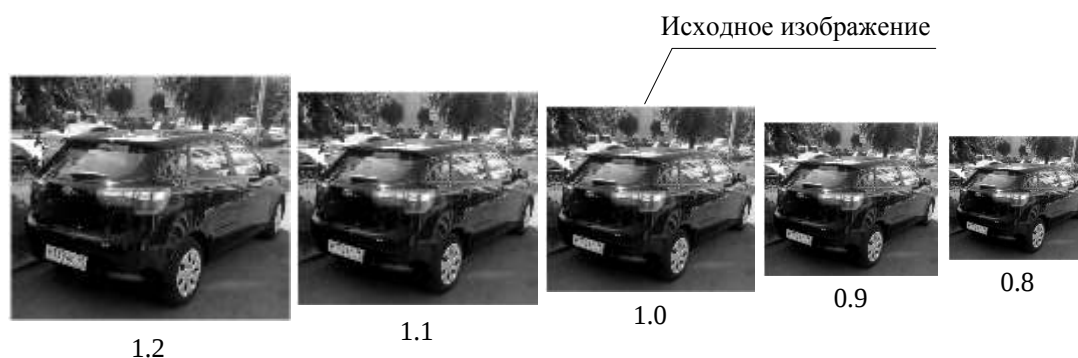


Рисунок 2.5 – Масштабы изображений, используемые в процессе детектирования

На основании этого, для повышения эффективности работы алгоритма детектирования было решено использовать масштабирование изображений, что обеспечило существенное улучшение результатов его работы. Данный алгоритм состоит из следующих шагов:

Шаг 1. Предобработка входного изображения. Исходное изображение преобразуется из цветного пространства RGB (*Red, Green, Blue*) в полутоновое.

Шаг 2. Масштабирование входного изображения. Входное изображение масштабируется с коэффициентами от 0.8 до 1.2. Получается набор из пяти изображений разного масштаба. Количество и размеры масштабов изображений были выбраны экспериментальным путем во избежание вычислительной избыточности и сохранения высокого уровня правильных обнаружений автомобильных номерных знаков.

Шаг 3. Предварительная классификация. СНС-1 производит сканирование изображений. В областях расположения автомобильных номерных знаков и в некоторой их окрестности СНС-1 будет выдавать локальные отклики максимальной величины. Области изображения, в которых локальные отклики превышают заданный порог, отмечаются на возможное содержание автомобильного номерного знака.

На данном этапе является важным уменьшение количества пропущенных обнаружений, снижение вычислительной избыточности и как следствие увеличение скорости работы алгоритма. Шаг сканирования изображений нейронной сетью выбран равным 4-м пикселям. В процессе вычислительных

экспериментов было установлено, что шаг данной величины позволяет повысить скорость работы алгоритма и снизить вычислительную избыточность.

Порог обнаружения (величина отклика сверточной нейронной сети, на основании которой делается вывод о наличии автомобильного номерного знака в исследуемой области изображения) выбирался таким образом, чтобы обеспечить минимальное количество ложных обнаружений и минимальное количество пропущенных автомобильных номерных знаков. На основании анализа откликов нейронной сети, оптимальная величина порога обнаружения для данного шага сканирования (четыре пикселя) была выбрана равной 0.7 (рисунок 2.6).

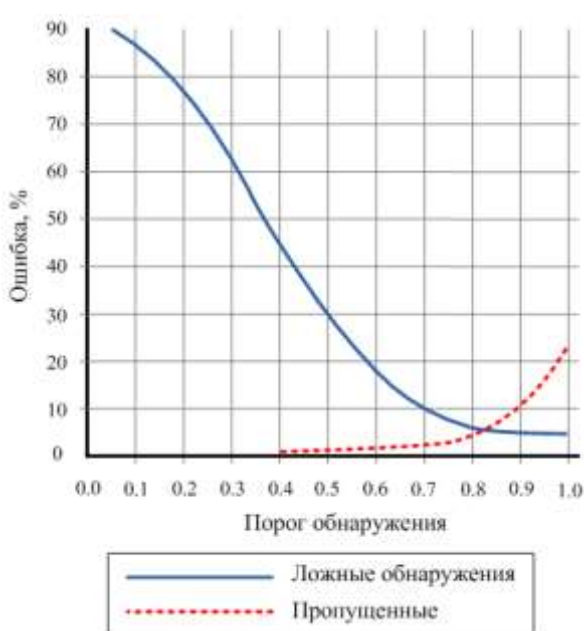


Рисунок 2.6 – Соотношение ложных обнаружений и пропущенных автомобильных номерных знаков

Как видно по рисунку 2.6 при увеличении значения порога обнаружения увеличивается количество пропущенных автомобильных номерных знаков. При уменьшении значения порога увеличивается количество ложных обнаружений. Выбранное значение порога (0.7) обеспечивает достаточно малое количество пропущенных обнаружений, однако количество ложных обнаружений не на достаточно низком уровне. На данном этапе это является допустимым, так как на этапе итоговой классификации ложные обнаружения пройдут более тщательную проверку.

Каждая обнаруженная область изображения характеризуется параметрами: x , y , s , h , r . Где, x , y – координаты центра области, s – ширина области изображения, h – высота области изображения, r – отклик нейронной сети для области изображения.

Шаг 4. Итоговая классификация. СНС-2 производит сканирование изображений в областях, выявленных на шаге 3. Для более тщательного анализа выявленных областей, на данном этапе шаг сканирования равен одному пикселю. Величина порога обнаружения для данного шага сканирования была выбрана равной 0.9. На данном этапе устраняется часть ложных обнаружений.

Шаг 5. Масштабирование изображений относительно исходного размера.

Шаг 6. Сравнение параметров обнаруженных областей: координаты расположения, границы. Перекрывающиеся обнаруженные области группируются в кластеры. Области, имеющие общие параметры принимаются за истинные обнаружения, остальные области принимаются за ложные обнаружения и отбрасываются из рассмотрения. На данном этапе выносятся окончательное решение о наличии автомобильных номерных знаков в исследуемых областях.

Шаг 7. Выделение обнаруженных областей прямоугольной рамкой (рисунок 2.7).



Рисунок 2.7 – Выделение автомобильного номерного знака

2.2 Разработка алгоритма нормализации изображений символов

После выполнения процесса детектирования следует этап нормализации и выделения отдельных символов, чтобы впоследствии их распознать.

Как уже было отмечено в главе 1, для распознавания символов было принято решение использовать алгоритм, основанный на сверточных нейронных сетях. СНС обладают двумерной структурой и функциональными возможностями, которые позволяют не использовать процесс выделения отдельных символов, а подать изображение обнаруженного автомобильного номерного знака целиком на вход сети. Таким образом, можно значительно снизить вычислительные затраты и повысить скорость работы алгоритма. На основании этого, на данном этапе было принято решение использовать только нормализацию изображения автомобильного номерного знака, без выделения отдельных символов.

Нормализация изображения включает в себя операции, обеспечивающие определение точных границ пластины номерного знака и преобразование изображения к виду, наиболее подходящему для эффективного распознавания символов. Основная сложность данного этапа заключается в том, что пластина номерного знака на изображении может быть расположена под неизвестным углом. Как уже было отмечено в главе 1, для решения данной задачи является эффективным применение алгоритма, основанного на построении гистограмм распределения яркости пикселей. Данный подход обеспечивает возможность определения горизонтальных границ номерного знака, расположенного под различными углами наклона [25].

Гистограммы распределения яркости пикселей представляют собой график с осями координат x и y . Ось x соответствует номерам строк изображения, ось y соответствует значениям яркости пикселей в строках. Вдоль горизонтальных граней пластины номерного знака строки изображения соответствуют белому фону, поэтому средняя интенсивность пикселей в данных областях будет превышать заданный порог [25] (рисунок 2.8).



Рисунок 2.8 – Построение горизонтальных гистограмм. Цифрами 1, 2 обозначены области, соответствующие гистограммам с максимальным значением яркости

Однако данный подход является не эффективным в тех случаях, если кузов автомобиля окрашен в светлые тона. Тогда гистограммы будут показывать максимальные значения в областях, соответствующих цвету кузова автомобиля (рисунок 2.9).



Рисунок 2.9 – Построение горизонтальных гистограмм. Цифрами 1 – 4 обозначены области, соответствующие гистограммам с максимальным значением яркости

Чтобы свести влияние фона к минимуму предлагается применять выделение высокочастотных пространственных компонент (выделение границ) на изображении [1, 20, 73]. В результате применения данной операции частично подавляются шумы на изображении и выделяются области резкого перепада яркости соседних пикселей. Изображение детектированного номерного знака содержит только небольшую окрестность номера, с минимальным количеством лишних объектов, поэтому применение операции выделения границ в данном случае более эффективно. Выделенные длинные горизонтальные линии предположительно будут являться горизонтальными гранями пластины номерного знака.

На основании этого был разработан следующий алгоритм нормализации изображений автомобильных номерных знаков:

Шаг 1. Повышение контраста на изображении.

Шаг 2. Выделение границ на изображении. Для выделения границ на изображении используется оператор Собеля, который вычисляет градиент яркости в каждой точке изображения. Точкам с постоянной яркостью будет

соответствовать значению модуля вектора градиента равно нулю. Точкам, лежащим на границе областей с различной яркостью, будет соответствовать значение вектора градиента, пересекающего границу в направлении увеличения яркости. После этого границы пластины номерного знака будут иметь максимальные значения яркости пикселей, что значительно упрощает процесс их последующего обнаружения. Длинные горизонтальные прямые, которые удалось выделить, предположительно будут являться горизонтальными гранями пластины номерного знака (рисунок 2.10).



Рисунок 2.10 – Выделение границ на изображении: а) исходное изображение; б) преобразованное изображение

Шаг 3. Поиск нижней горизонтальной грани пластины номерного знака. Как правило, нижняя часть автомобилей оттенена и имеет более резкий перепад яркости по отношению к пластине номерного знака. Поэтому в результате выделения границ нижняя грань номерного знака выделяется более четко и ее проще определить. В то время как верхняя часть автомобилей часто бывает засвечена и имеет более слабый контраст по отношению к пластине номерного знака. Так же верхняя грань номерного знака может быть частично утоплена, поэтому определить ее сложнее.

На основании этого на данном этапе выполняется построение гистограмм распределения яркости пикселей по строкам в нижней части изображения. Строка, которая соответствует горизонтальной грани пластины номерного знака, будет иметь максимальное значение яркости пикселей (рисунок 2.11).



Рисунок 2.11 – Построение горизонтальных гистограмм в нижней части изображения

Пластина номерного знака на изображении может быть расположена под неизвестным углом, поэтому для нахождения ее граней необходимо строить не одну гистограмму, а n гистограмм под различными углами. Однако данный подход может привести к избыточным вычислительным процессам. Что бы избежать этого предлагается определить параметры соотношения высоты и ширины изображения. Зная эти параметры можно предположить диапазон значений углов наклона автомобильного номерного знака. Как правило, чем больше высота изображения, тем больше угол наклона под которым расположен автомобильный номерной знак (рисунок 2.12).



Рисунок 2.12 – Изменение высоты и ширины изображения при различных углах наклона пластины номерного знака (h и w – высота и ширина изображения соответственно)

Для сокращения избыточных вычислительных процессов предлагается ряд условий:

- Если $h \leq 1/4w$, то выполняется построение горизонтальных гистограмм при следующих параметрах: $g1 = 1/2h$, $a1 = 3$, $a2 = -3$.
- Если $h \leq 1/3w$, то выполняется построение горизонтальных гистограмм при следующих параметрах: $g1 = 1/2h$, $a1 = 10$, $a2 = -10$.
- Если $h \leq 1/2w$, то выполняется построение горизонтальных гистограмм при следующих параметрах: $g1 = 2/3h$, $a1 = 15$, $a2 = -15$.
- Если $h > 1/2w$, то выполняется построение горизонтальных гистограмм при следующих параметрах: $g1 = 2/3h$, $a1 = 20$, $a2 = -20$,

где h – высота изображения, w – ширина изображения, $g1$ – нижняя обрабатываемая область изображения, $a1$ – верхний максимальный угол построения гистограмм, $a2$ – нижний максимальный угол построения гистограмм.

Предельные значения углов построения гистограмм установлены на уровне ± 20 градусов. Это обусловлено тем, что в реальных условиях не возникают ситуации, когда угол отклонения пластины номерного знака на плоскости превышает диапазон ± 20 градусов, за исключением аварийных ситуаций.

Шаг 4. Поиск гистограммы с максимальным значением интенсивности пикселей. Среди всех вычисленных гистограмм выбирается гистограмма, которая содержит наибольшее значение по оси y . Таким образом, определяется расположение нижней грани пластины номерного знака и угол ее наклона.

Шаг 5. Поиск верхней горизонтальной грани пластины номерного знака. Из-за возможных аффинных и проекционных искажений пластины номерного знака, углы наклона ее верхней и нижней грани не обязательно будут равны. Поэтому, на данном шаге производится построение горизонтальных гистограмм в верхней части изображения (рисунок 2.13) при следующих параметрах: $g2 = g1$, $a1 = K+3$, $a2 = K-3$, где h – высота изображения, w – ширина изображения, $g2$ – верхняя обрабатываемая область изображения, K – угол наклона нижней грани пластины номерного знака, $a1$ – верхний максимальный угол построения гистограмм, $a2$ – нижний максимальный угол построения гистограмм.



Рисунок 2.13 – Построение горизонтальных гистограмм в верхней части изображения

Шаг 6. Поиск гистограммы с максимальным значением интенсивности пикселей. Среди вычисленных гистограмм выбирается вторая гистограмма с максимальным значением по оси y , которая принимается за верхнюю грань пластины номерного знака.

По причине сильной зашумленности автомобильного номерного знака верхняя горизонтальная гистограмма может быть не найдена. В этом случае, исходя из геометрических характеристик автомобильных номерных знаков, предполагается, что верхняя грань расположена параллельно нижней грани на расстоянии $1/4w$ от нее.

Шаг 7. Выравнивание изображения. Изображение выравнивается в горизонтальную ориентацию относительно найденных горизонтальных граней пластины номерного знака. Области изображения, выходящие за пределы найденных граней, обрезаются (рисунок 2.14).



Рисунок 2.14 – Пример нормализации изображения автомобильного номерного знака

2.3 Разработка алгоритма распознавания символов

После этапа нормализации изображения автомобильного номерного знака, следует последний этап – распознавание символов. Так как изображение номерного знака имеет не большой размер и не содержит сложную фоновую структуру, то для распознавания символов было принято решение разработать алгоритм, состоящий из одной сверточной нейронной сети (СНС-3).

Согласно ГОСТ Р 50577 – 93 автомобильные номерные знаки соответствующие группе 1, типу 1, содержат следующие символы: А, В, Е, К, М, Н, О, Р, С, Т, Х, У и цифры от 0 до 9. Символы О и 0 имеют идентичную структуру, поэтому было решено объединить их в один класс символов. Следовательно, СНС-3 должна обеспечивать классификацию 21 символа.

В процессе исследований были реализованы СНС с различной конфигурацией и параметрами для выявления наиболее оптимальной и

эффективной конфигурации. Наилучшие результаты работы были получены при использовании СНС, описанной в следующем разделе.

2.3.1 Сверточная нейронная сеть для распознавания символов

Для распознавания символов, изображенных на автомобильном номерном знаке, была разработана СНС-3, представленная на рисунке 2.15.

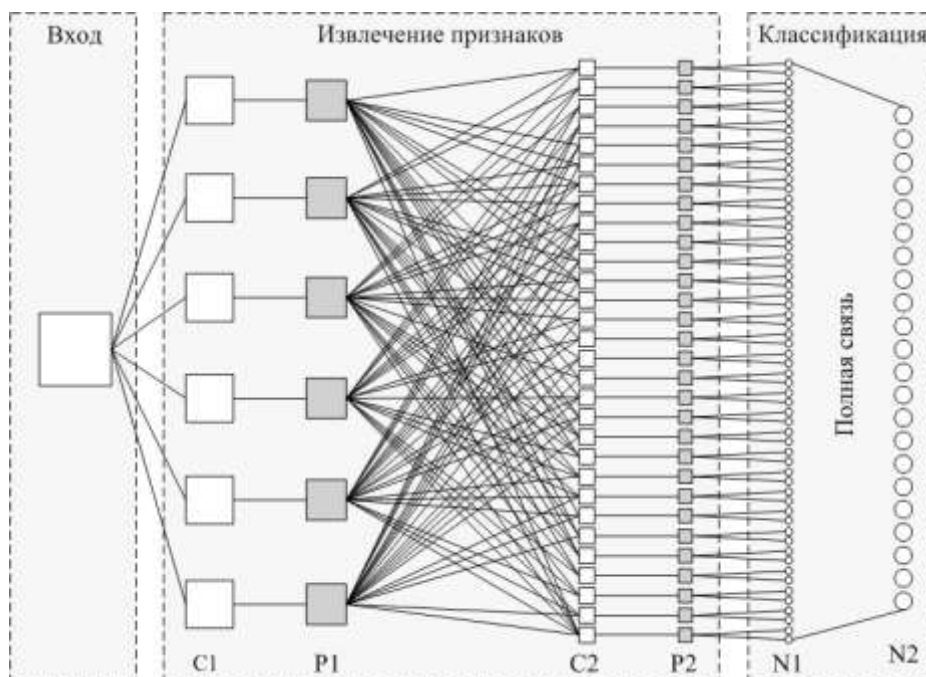


Рисунок 2.15 – Структура СНС-3. C1, C2 – слои свертки; P1, P2 – слои подвыборки; N1, N2 – слои из обычных нейронов

Входной слой размером 29x29 нейронов состоит из 841 нейрона и служит для подачи входного образа в нейронную сеть.

Второй слой C1 является слоем свертки и состоит из 6 сверточных плоскостей размером 26x26 нейронов. Данный размер плоскостей обеспечивает достаточное наложение рецептивных полей друг на друга. Каждая плоскость данного слоя имеет собственную матрицу синаптических коэффициентов и нейронное смещение. Рецептивные поля нейронов частично пересекаются и имеют размер 4x4 нейрона. Каждой плоскости соответствует $4 \times 4 = 16$ весовых коэффициентов и нейронное смещение. Таким образом, данный слой содержит 4056 нейронов и 102 весовых коэффициента.

Размер плоскостей сверточного слоя определяется в соответствии с формулами (2.1).

Функционирование нейрона сверточного слоя определяется формулой (2.2).

Третий слой P1 является слоем подвыборки и состоит из 6 подвыборочных плоскостей размером 13×13 нейронов. Размер плоскостей данного слоя в два раза меньше размера плоскостей предыдущего слоя. Каждая плоскость данного слоя связана только с одной соответствующей ей плоскостью предыдущего слоя, имеет один синаптический коэффициент и нейронное смещение. Рецептивные поля смежных нейронов не пересекаются и имеют размер 2×2 нейрона. Таким образом, данный слой содержит 1014 нейронов и всего 12 синаптических весовых коэффициентов.

Размер плоскостей подвыборочного слоя определяется в соответствии с формулами (2.3).

Функционирование нейрона подвыборочного слоя определяется формулой (2.4).

Четвертый слой C2 является слоем свертки и состоит из 30 сверточных плоскостей размером 10×10 нейронов. Рецептивные поля смежных нейронов частично пересекаются. Плоскости данного слоя перекрестно связаны с плоскостями предыдущего слоя. Принцип распределения связей между слоями P1 и C2 похож на СНС-2, однако отличается, т.к. СНС-3 содержит большее количество плоскостей в слое C2. Плоскости 1–6 и 24–29 данного слоя связаны со всеми соседними подмножествами, содержащими по 3 плоскости предыдущего слоя. Плоскости 7–12 и 18–23 получают данные от всех соседних подмножеств, содержащих по 4 плоскости предыдущего слоя. Плоскости 13–17 получают данные от некоторых перемежающихся подмножеств, содержащих по 3 плоскости предыдущего слоя. И последняя плоскость получает данные от всех плоскостей предыдущего слоя.

Распределение связей между слоями P1 и C2 в СНС-3 подбиралось так же экспериментальным путем. В таблицах 2.5 и 2.6 представлено распределение

связей между слоями P1 и C2 в СНС-3. По горизонтали отображаются плоскости слоя C2, по вертикали плоскости слоя P1.

Таблица 2.5 – Распределение связей между слоями P1 и C2 в СНС-3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	x				x	x	x			x	x	x			x
2	x	x				x	x	x			x	x	x		
3	x	x	x				x	x	x			x	x	x	
4		x	x	x			x	x	x	x				x	x
5			x	x	x			x	x	x	x		x		x
6				x	x	x			x	x	x	x		x	

Таблица 2.6 – Распределение связей между слоями P1 и C2 в СНС-3

	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1		x	x	x	x	x			x	x	x				x
2	x			x	x	x	x			x	x	x			x
3		x			x	x	x	x			x	x	x		x
4			x			x	x	x				x	x	x	x
5	x		x	x			x	x	x				x	x	x
6	x	x	x	x	x			x	x	x				x	x

Размер локального рецептивного поля равен 4×4 нейронов. Каждой плоскости соответствует $4 \times 4 = 16$ весовых коэффициентов и нейронное смещение. Таким образом, данный слой содержит 3000 нейронов и 510 весовых коэффициентов.

Пятый слой P2 является слоем подвыборки, состоит из 30 подвыборочных плоскостей. Размер плоскостей данного слоя в два раза меньше размера плоскостей предыдущего слоя и равен 5×5 нейронов. Каждая плоскость имеет связь только с одной соответствующей ей плоскостью предыдущего слоя. Рецептивные поля смежных нейронов не пересекаются и имеют размер 2×2 нейрона. Каждой плоскости соответствует один синаптический коэффициент и нейронное смещение. Данный слой содержит 750 нейронов и 60 синаптических весовых коэффициентов.

Первые пять слоев предназначены для извлечения признаков изображения. Следующие два слоя являются слоями классификации.

Шестой слой N1 состоит из 60 нейронов. Каждый нейрон данного слоя имеет связи со всеми нейронами двух соответствующих ему плоскостей предыдущего слоя P2. Каждый нейрон вычисляет взвешенное суммирование 50 соответствующих ему входных параметров, добавляет нейронное смещение и полученный результат передает через функцию активации. Таким образом, данный слой содержит 1530 синаптических весовых коэффициентов.

Седьмой слой N2 состоит из 22 нейронов и является выходным слоем. Роль данного слоя заключается в вычислении окончательного результата классификации. Каждый нейрон данного слоя имеет связи со всеми нейронами предыдущего слоя. Нейроны выходного слоя соответствуют 21 классу распознаваемых символов: А, В, Е, К, М, Н, О, Р, С, Т, Х, У и цифры от 0 до 9. 22-ой нейрон данного слоя соответствует фону. Это необходимо для того, что бы ложные обнаружения, которые могут быть выявлены на этапе детектирования, классифицировать как фон. Данный слой содержит 1380 синаптических весовых коэффициентов.

Значения нейронов выходного слоя вычисляются выражением (2.5). На выходе каждого нейрона производится вычисление Евклидова расстояния между вектором выходных значений и параметрическим вектором.

СНС-3 содержит 9743 нейронов и 3594 синаптических коэффициентов.

В таблице 2.7 представлена конфигурация СНС-3.

Таблица 2.7 – Конфигурация СНС-3

Слой	1	2	3	4	5	6	7
Число плоскостей	1	6	6	30	30	-	-
Размер плоскостей	29x29	26x26	13x13	10x10	5x5	-	-
Количество нейронов в слое	841	4056	1014	3000	750	60	22
Размер локального рецептивного поля	-	4x4	2x2	4x4	2x2	-	-
Количество синаптических коэффициентов	-	102	12	510	60	1530	1380

2.3.2 Алгоритм распознавания символов

Алгоритм распознавания символов основан на разработанной СНС-3. В работе данного алгоритма входное изображение представляется в нескольких последовательных масштабах, которые подаются на вход СНС-3. Масштабирование входного изображения предназначено для повышения точности классификации символов. Это обусловлено тем, что при различных масштабах изображения характерные признаки символов могут иметь отличающуюся визуальную структуру и нейронная сеть на них может реагировать по-разному.

В процессе экспериментов были выбраны следующие масштабы: 1.0, 1.1, 1.2 (рисунок 2.16). Данные масштабы были выбраны исходя из того, что изображение автомобильного номерного знака в этих масштабах не теряет своих характерных признаков и наиболее соответствует размеру входа нейронной сети.

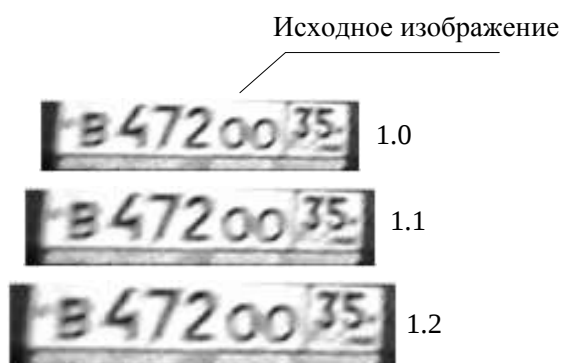


Рисунок 2.16 – Масштабы изображений, используемые в процессе распознавания символов

Данный алгоритм состоит из следующих шагов:

Шаг 1. Масштабирование входного изображения. Входное изображение масштабируется с коэффициентами от 1.0 (40 пикселей по высоте) до 1.2. Получается набор из трех изображений разного масштаба.

Шаг 2. Обработка изображений сверточной нейронной сетью (СНС-3). Шаг сканирования равен одному пикселю. Это обусловлено тем, что изображения автомобильного номерного знака имеют не большой размер, поэтому шаг данной

величины не приведет к вычислительной избыточности и при этом обеспечит более тщательный процесс классификации символов. В областях расположения символов СНС-3 будет выдавать максимальные локальные отклики. Каждый отклик соответствует одному из нейронов в выходном слое нейронной сети и определенному классу символов. На основании анализа откликов нейронной сети оптимальная величина порога классификации была выбрана равной 0.8.

Область расположения символа на изображении характеризуется параметрами: x , y , s , h , r . Где, x , y – координаты центра области, s – ширина области, h – высота области, r – отклик нейронной сети в данной области изображения. На рисунке 2.17 изображена диаграмма, представляющая собой матрицу откликов нейронной сети, соответствующих изображению автомобильного номерного знака.

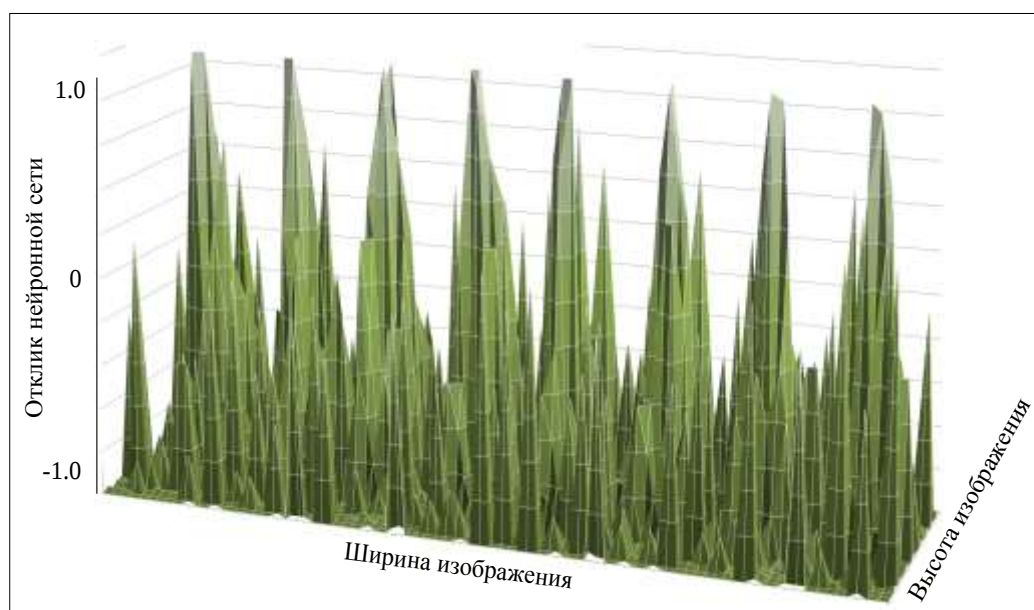


Рисунок 2.17 – Отклики СНС-3 при сканировании изображения автомобильного номерного знака

Шаг 3. Масштабирование изображений относительно исходного размера.

Шаг 4. Сравнение параметров обнаруженных областей: координаты расположения, границы и соответствие определенному классу символов. Области, имеющие общие параметры, принимаются за истинные обнаружения, остальные области принимаются за ложные обнаружения и отбрасываются из рассмотрения.

Так же на данном этапе выполняется проверка параметров найденных областей на соответствие требуемым условиям. На автомобильных номерных знаках могут быть лишние мелкие объекты и символы: флаг страны и символы «rus» (рисунок 2.18). Данные объекты и символы сложно поддаются распознаванию из-за своего маленького размера и могут внести погрешность в работу алгоритма.



Рисунок 2.18 – Указатели страны на автомобильном номерном знаке

Поэтому необходимо отбросить данные объекты из рассмотрения. Для этого параметры найденных областей проверяются на соответствие следующему условию: если $h \leq 0,4h_{max}$, то данная область отбрасывается из рассмотрения. Здесь, h – высота области расположения символа, h_{max} – высота изображения.

Шаг 5. Классифицированные символы выводятся на экран в текстовом виде (рисунок 2.19.).



Рисунок 2.19 – Распознавание символов на автомобильном номерном знаке

2.4 Выбор активационной функции для сверточных нейронных сетей

Активационная функция предназначена для преобразования взвешенной суммы входных сигналов нейрона в выходной сигнал.

При выборе активационной функции для разработанных СНС рассматривались две нелинейные сигмоидальные функции: логистическая и гиперболический тангенс (рисунок 2.20) [75, 139]. Данные функции являются монотонно возрастающими и дифференцируемыми.

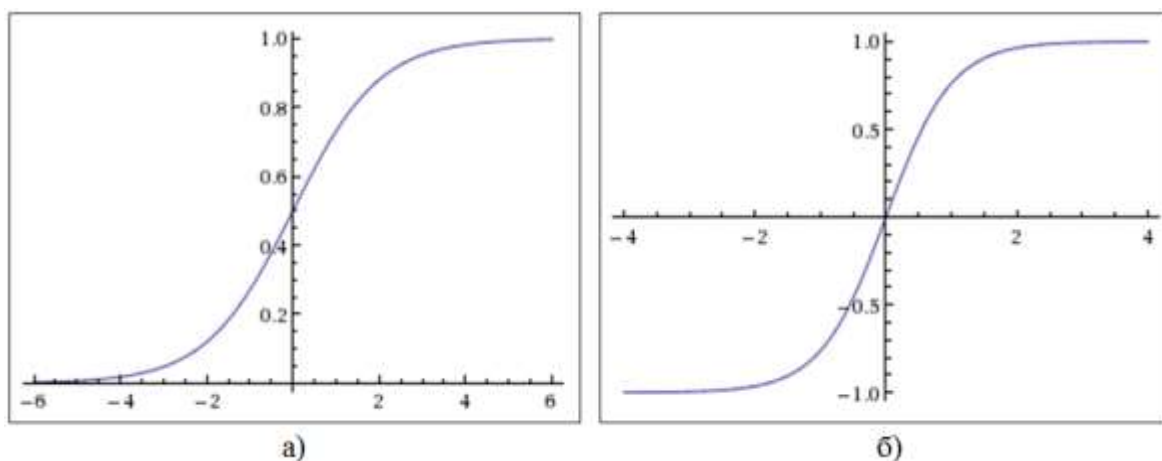


Рисунок 2.20 – Функции активации: а) логистическая функция;
б) гиперболический тангенс

Логистическая функция задается формулой:

$$f(x) = \frac{1}{(1 + e^{-s})},$$

где s – взвешенная сумма входных значений нейрона.

Область значений логистической функции находится в интервале $[0; +1]$.

Функция гиперболический тангенс задается формулой:

$$f(x) = a \tanh(bx),$$

$$\tanh(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}},$$

где a и b – константы.

Функция гиперболический тангенс является симметричной относительно начала координат, а область ее значений находится в интервале $[-1; +1]$.

В данной работе было принято решение использовать активационную функцию гиперболический тангенс. Преимущества данной функции заключаются в следующем:

- функция является симметричной и обеспечивает более быструю сходимость, по сравнению с логистической функцией;
- функция имеет простую производную [137]:

$$\frac{dF}{dy} = \frac{d}{dy} \left(\frac{\sinh(y)}{\cosh(y)} \right) = \frac{\sinh^2 y - \cosh^2 y}{\cosh^2 y} = 1 - \tanh^2 y.$$

Т.е. если $x = \tanh(y)$, то $\frac{dF}{dy} = 1 - x^2$.

- функция имеет максимум второй производной при $a=1$;
- тангенс угла наклона в начале координат принимает значение близкое к единице [137, 139, 169].

Целевые значения

Как правило, в задачах классификации целевые значения являются бинарными и устанавливаются по асимптотам активационной функции $[-1; +1]$. Однако, данный подход может иметь недостатки.

Обучение нейронной сети может привести к не стабильному результату. В процессе обучения выход нейронной сети будет стремиться к целевым значениям, которые находятся на асимптотах активационной функции. Веса синаптических коэффициентов будут увеличиваться и принимать значения, в которых производная функции активации близка к нулю. Слишком большие значения синаптических коэффициентов приводят к увеличению градиента. При умножении больших значений градиента на маленькое значение производной активационной функции получаются слишком маленькие приращения весовых коэффициентов. Таким образом, коррекция синаптических коэффициентов будет не значительной [139].

Так же возможны случаи, когда входные данные нейронной сети могут находиться на границе решения. Однако, большие значения синаптических коэффициентов могут смещать выход нейронной сети к асимптотам активационной функции независимо от класса входных данных. Таким образом, результат классификации может быть не стабилен.

Данные проблемы решаются, если ожидаемый отклик нейронной сети смещается от целевых значений активационной функции в сторону ее внутренней части. В этом случае при обучении нейронной сети можно избежать резкого

увеличения значений синаптических коэффициентов, а так же увеличить скорость процесса обучения. Таким образом, для активационной функции были выбраны следующие параметры:

$$a=1.7159; \quad b=2/3;$$
$$y=1.7159 \tanh\left(\frac{2}{3}x\right).$$

Как было отмечено выше, тангенс угла наклона функции активации в начале координат принимает значение близкое к единице:

$$f'(0) = ab = 1,7159 \times 2 / 3 = 1,1424.$$

Чтобы избежать насыщения активационной функции можно установить целевые значения на грани максимума ее второй производной. В значениях +1 и – 1 функция гиперболический тангенс имеет максимальную величину второй производной. Данные значения соответствуют целевым значениям, которые являются типичными в задачах классификации [39, 137].

2.5 Нормирование входных значений

Входные и выходные данные нейронной сети могут быть совершенно разнородными величинами и измеряться в различных единицах измерения, однако это может негативно сказаться на результатах обучения и классификации. Очевидно, что результаты работы нейронной сети не должны зависеть от единиц измерения входных данных. Что бы нейронная сеть трактовала значения входных и выходных данных единообразно, необходимо эти значения преобразовать к однородным величинам [137–139].

Обычно сходимость процесса обучения нейронной сети происходит быстрее, если среднее значение каждой переменной из входных данных близко к нулю. Если, например все переменные из входных данных будут положительны, то все коррекции синаптических коэффициентов для нейрона в первом слое могут иметь тот же знак. Таким образом, для данного входного значения, веса могут только все вместе увеличиться или же все вместе уменьшиться. В результате этого, для изменения своего направления вектор весовых коэффициентов должен

делать зигзаги, что является не эффективным и снижает скорость процесса обучения.

Таким образом, для оптимизации и ускорения процесса обучения, все входные значения пикселей из интервала от 0 до 255 (т.е. цвет пикселей от черного до белого) нормируются в интервал $[-1; +1]$ по следующей формуле [137]:

$$y_i = \frac{x_i}{128} - 1,$$

где x_i – значение пикселя исходного изображения; y_i – значение пикселя для подачи на вход нейронной сети.

2.6 Выбор исходных значений синаптических коэффициентов

Выбор исходных значений синаптических коэффициентов нейронной сети имеет большое значение в процессе обучения. Так как цель алгоритма обучения заключается в нахождении оптимального значения весовых коэффициентов, то их начальное состояние может повлиять на количество итераций и скорость сходимости алгоритма обучения.

Самый распространенный вариант заключается в случайном выборе значений синаптических коэффициентов из некоторого интервала значений.

Но проблема состоит в том, что значения синаптических коэффициентов могут иметь слишком большую величину, а это может привести к насыщению активационной функции и большим значениям градиента. При умножении больших значений градиента на маленькое значение производной активационной функции получаются слишком маленькие приращения весовых коэффициентов. Это ведет к снижению скорости процесса обучения. Так же при слишком маленьких величинах синаптических коэффициентов может возникать схожая ситуация [139].

Что бы избежать этих недостатков синаптические коэффициенты должны быть выбраны таким образом, что бы попадать в линейную область значений

активационной функции, что положительно повлияет на скорость процесса обучения.

Для реализации оптимального процесса обучения необходимо достичь согласованности между выбранной активационной функцией, начальными значениями синаптических весовых коэффициентов и параметрами входных данных нейронной сети.

В процессе проведенных исследований было решено использовать инициализацию исходных значений синаптических коэффициентов случайным образом на основе равномерного распределения с нулевым средним и стандартным отклонением:

$$\sigma_w = m^{-1/2},$$

где m – число связей входящих в нейрон [137].

2.7 Выбор алгоритма обучения для сверточных нейронных сетей

Способность к обучению на основе данных из внешней среды является одним из самых важных свойств, присущих ИНС. Данная способность придает нейронным сетям гибкость при решении широкого круга задач и делает их более эффективными в сравнении с системами, функционирующими по определенным шаблонным правилам [39].

Процесс обучения ИНС предполагает последовательную корректировку синаптических весовых коэффициентов между нейронами. Значения синаптических коэффициентов играют важную роль в процессе функционирования нейронной сети, поэтому необходимо найти их оптимальные значения [82].

В настоящее время существует множество алгоритмов обучения, основной принцип их работы заключается в сведении к минимуму эмпирической ошибки. Для реализации эффективного процесса обучения необходимо иметь данные из внешней среды, на основе которых будет обучаться нейронная сеть.

Обучение ИНС бывает двух типов: с учителем и без учителя. При обучении с учителем для всех данных из обучающей выборки, поступающих на вход

нейронной сети определяются их желаемые выходные значения. В процессе обучения происходит сравнение текущих выходных значений нейронной сети с желаемыми, вычисляется эмпирическая ошибка и происходит изменение синаптических весовых коэффициентов. Весовые коэффициенты изменяются до тех пор, пока ошибка не сведется к минимуму и станет меньше заданного порога.

Обучение без учителя заключается в последовательной настройке весовых коэффициентов таким образом, что бы на предъявление близких значений входных векторов на выходе получались одинаковые стабильные значения [75, 82].

Что касается самого процесса обучения, то его так же можно разделить на два типа: пакетное и стохастическое. Пакетное обучение предполагает, что на каждой итерации обучающая выборка просматривается полностью, и только после этого осуществляется коррекция синаптических коэффициентов. При стохастическом обучении выбирается лишь часть образов из учебного набора (по определенным критериям) и после этого выполняется коррекция синаптических коэффициентов. Стохастическое обучение является более предпочтительным, так как обеспечивает более быструю сходимость процесса обучения, в отличие от пакетного метода [162, 188].

При выборе наиболее эффективного алгоритма обучения для разработанных СНС было проведено исследование различных алгоритмов, в результате чего были отобраны наиболее эффективные и востребованные алгоритмы. Для обучения разработанных нейронных сетей использовались следующие алгоритмы:

- 1) Алгоритм обратного распространения ошибки (Backpropagation Algorithm) [15, 162, 187].
- 2) Алгоритм Quick Propagation (QuickProp) [108].
- 3) Алгоритм Resilient Propagation (RProp) [155, 156].
- 4) Алгоритм Левенберга-Марквардта (Levenberg-Marquardt Algorithm) [140, 143].
- 5) Генетический алгоритм [121, 122].

Алгоритм обратного распространения ошибки

В 1974 году исследователями *А.И. Галушкиным* и *P.J. Werbos* [15, 16, 187] был предложен итеративный градиентный алгоритм, названный алгоритмом обратного распространения ошибки (Backpropagation Algorithm), который в дальнейшем был доработан *D. Rumelhart*, *G. Hinton* и *R.J. Williams* [162]. Цель данного алгоритма заключается в сведении к минимуму эмпирической ошибки для получения желаемого результата.

В основе данного алгоритма лежит принцип, позволяющий вычислять вектор частных производных (градиент) функции ошибки сети. Функцией ошибки является разница между текущим выходным значением нейронной сети и желаемым, которое необходимо получить. В процессе обучения выполняется последовательная корректировка весовых коэффициентов и ошибка сети постепенно уменьшается [82].

Величина функции ошибки прямо пропорциональна квадрату разности между текущим и желательным значениями выхода сети:

$$E_k = \frac{1}{2} \sum_{j=0}^N (t_{kj} - x_{kj})^2,$$

где E_k – величина функции ошибки для образа k ; t_{kj} – желаемое значение выхода нейрона j для образа k ; x_{kj} – текущее значение выхода нейрона j для образа k .

Коррекция синаптических коэффициентов происходит по формуле:

$$\Delta w_{ij} = -\eta \delta_{kj} x_{kj},$$

где δ_{kj} – значение ошибки нейрона j для образа k ; x_{kj} – текущий выход нейрона j для образа k ; η – коэффициент скорости обучения, $0 < \eta < 1$.

Значение ошибки для нейронов сети определяется по следующей формуле:

$$\delta_i^{(q)} = (f_i^{(q)}(S))' \sum_j w_{ij} \delta_j^{(q+1)},$$

где $\delta_i^{(q)}$ – значение ошибки i -го нейрона в слое q ; $\delta_j^{(q+1)}$ – значение ошибки j -го нейрона в слое $q+1$; w_{ij} – вес связи, соединяющей два нейрона; $(f_i^{(q)}(S))'$ – значение производной активационной функции i -го нейрона в слое q .

Данный алгоритм можно описать в нескольких шагах:

- 1) Инициализация синаптических весовых коэффициентов случайными значениями.
- 2) Подача на вход сети набора данных из обучающей выборки.
- 3) Вычисление выходного сигнала нейронной сети.
- 4) Вычисление величин ошибок нейронов выходного слоя и нейронов скрытых слоев.
- 5) Корректировка весовых коэффициентов сети для минимизации ошибки.
- 6) Шаги 2 – 5 повторяются до тех пор, пока ошибка сети не сведется к минимуму и достигнет приемлемого значения [188, 189].

Недостаток данного алгоритма заключается в очень малых корректировках весов, что ведет к слишком долгому процессу обучения. При этом возникает задача выбора оптимального размера шага. Слишком маленький размер шага приводит к медленной сходимости алгоритма, слишком большой размер шага может привести к потере устойчивости процесса обучения.

Алгоритмы Quick Propagation и Resilient Propagation

Алгоритмы Quick Propagation (QuickProp) и Resilient Propagation (RProp) относятся к эвристическим алгоритмам и разработаны *S. Falman*, *M. Ridmiller* и *H. Braun* [108, 155, 156]. Данные алгоритмы являются модификацией классического алгоритма обратного распространения ошибки и базируются на классическом градиентном алгоритме, также называемом алгоритмом наискорейшего спуска.

Отличие алгоритмов QuickProp и RProp от других алгоритмов, основанных на методе градиентного спуска, заключается в использовании знаков частных производных, а не их значений. Величина шага обновления уникальна для каждого веса и в процессе обучения происходит ее адаптация [108].

В процессе работы методов вычисляется вектор градиента, а весовые коэффициенты изменяются в направлении антиградиента. Вектор градиента вычисляется по формуле:

$$g_i = \frac{\partial \varepsilon(w_i^{(k)})}{\partial w_i^{(k)}} = \frac{\varepsilon(w_i + \Delta w_i^{(k-1)}) - \varepsilon(w_i^{(k)})}{\Delta w_i^{(k-1)}},$$

где, g – элемент вектора градиента, w – весовые коэффициенты [37].

После вычисления вектора градиента выполняется корректировка весовых коэффициентов. В алгоритме QuickProp корректировка весовых коэффициентов осуществляется по следующей формуле:

$$w_i^{(k)} = w_i^{(k-1)} + \Delta w_i^{(k)} = -s(g_i^{(k-1)} + c_w w_i^{(k-1)}) + q_i^{(k)} \Delta w_i^{(k-1)},$$

где, g_i – i -й весовой коэффициент на k -ой итерации; g_i – элемент вектора градиента; s – коэффициент обучения; $g_i^{(k)}$ – коэффициент фактора момента; c_w – параметр, предназначенный для минимизации значений весовых коэффициентов [37].

Алгоритм QuickProp отличается от классических градиентных методов наличием двух слагаемых: параметр минимизации весовых коэффициентов c_w , обычно принимающий значение 10^{-4} ; коэффициент момента q_i , предназначенный для адаптации алгоритма обучения. Каждому весовому коэффициенту соответствует свой уникальный коэффициент момента. Коэффициент момента вычисляется по следующей формуле:

$$q_i^{(k)} = \frac{g_i^{(k-1)}}{g_i^{(k-2)} - g_i^{(k-1)}}.$$

Далее коэффициент момента принимает минимальное значение из $\hat{q}_i$ и $q_{\max}$ [94].

Отличительная особенность алгоритма RProp заключается в том, что значение градиента не учитывается. Что бы вычислить новые значения весовых коэффициентов требуется знать следующие параметры:

- величину коэффициента обучения на предыдущей итерации;
- значения градиента на текущей и предыдущей итерации;

- изменение весового коэффициента на предыдущей итерации.

Для определения величины коррекции весовых коэффициентов используется правило:

$$\Delta_{i,j}^{(k)} = \begin{cases} \eta^+ \Delta_{i,j}^{(k)}, & \frac{\partial E^{(k)}}{\partial w_{i,j}} \frac{\partial E^{(k-1)}}{\partial w_{i,j}} > 0 \\ \eta^- \Delta_{i,j}^{(k)}, & \frac{\partial E^{(k)}}{\partial w_{i,j}} \frac{\partial E^{(k-1)}}{\partial w_{i,j}} < 0 \end{cases}, \quad 0 < \eta^- < 1 < \eta^+. \quad (2.6)$$

Если частная производная весового коэффициента $w_{i,j}^{(t)}$ изменяет свой знак, это значит, что последнее обновление весов было слишком большим и алгоритм пропустил локальный минимум. В данном случае величину изменения необходимо уменьшить на η и вернуться к предыдущему значению весовых коэффициентов:

$$\Delta w_{i,j}^{(k)} = \Delta w_{i,j}^{(k)} - \Delta_{i,j}^{(k-1)}.$$

Если же производная не меняет знак, то для достижения более быстрой сходимости значение коррекции увеличивается на η^+ [128].

Величина коррекции весовых коэффициентов определяется в соответствии со следующим правилом:

$$\Delta w_{i,j}^{(k)} = \begin{cases} -\Delta_{i,j}^{(k)}, & \frac{\partial E^{(k)}}{\partial w_{i,j}} > 0 \\ +\Delta_{i,j}^{(k)}, & \frac{\partial E^{(k)}}{\partial w_{i,j}} < 0 \\ 0, & \frac{\partial E^{(k)}}{\partial w_{i,j}} = 0 \end{cases}. \quad (2.7)$$

В случае если ошибка возрастает, т.е. значение производной положительно, то весовой коэффициент уменьшается на величину коррекции, иначе — увеличивается. Затем осуществляется коррекция весовых коэффициентов [128].

Алгоритм Левенберга-Марквардта

Алгоритм Левенберга-Марквардта был предложен *K. Levenberg* (1944 г.) [140], а в дальнейшем оптимизирован *D. Marquardt* (1963 г.) [143] и является методом оптимизации параметров нелинейных регрессионных моделей. Данный алгоритм является комбинацией метода Гаусса-Ньютона и метода градиентного спуска. Преимущества данного алгоритма заключаются в стабильной работе (позаимствованной от метода градиентного спуска) и в ускоренной сходимости (позаимствованной от метода Гаусса-Ньютона).

Отличие алгоритма Левенберга-Марквардта от других методов заключается в использовании матрицы Гессе (матрица, образованная вторыми частными производными функции), описывающей поведение функции во втором порядке и регулирующего параметра. Параметром оптимизации является среднеквадратичная ошибка нейронной сети на обучающей выборке.

Для оценки значения Δw используется линейное приближение функции:

$$f(w + \Delta w, x) \approx f(w, x) + J \Delta w,$$

где w – вектор весовых коэффициентов; J – якобиан функции $f(w, x_n)$ в точке w [174].

Матрицу Якоби (матрица частных производных) можно представить в виде:

$$J = \begin{bmatrix} \tilde{J}_1 \\ \dots \\ \tilde{J}_N \end{bmatrix}, \quad \tilde{J}_t = \begin{bmatrix} \frac{\partial f(w, x_1)}{\partial w_1} & \frac{\partial f(w, x_1)}{\partial w_2} & \dots & \frac{\partial f(w, x_1)}{\partial w_R} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f(w, x_N)}{\partial w_1} & \frac{\partial f(w, x_N)}{\partial w_2} & \dots & \frac{\partial f(w, x_N)}{\partial w_R} \end{bmatrix}. \quad (2.8)$$

Чтобы найти значение Δw нужно решить систему линейных уравнений:

$$\Delta w = (J^T J)^{-1} J^T (y - f(w)).$$

Так как число обусловленности матрицы $J^T J$ есть квадрат числа обусловленности матрицы J , то матрица $J^T J$ может оказаться существенно вырожденной. Поэтому вводится регулирующий параметр $\lambda \geq 0$:

$$\Delta w = (J^T J + \lambda I)^{-1} J^T (y - f(w)), \quad (2.9)$$

где I – единичная матрица [79].

Регулирующий параметр λ назначается на каждой итерации алгоритма. При выборе оптимального значения параметра λ можно добиться монотонного убывания минимизируемой функции. Если значение ошибки E_k убывает быстро, малое значение λ сводит этот алгоритм к алгоритму Гаусса-Ньютона. Если регулирующий параметр λ будет достаточно велик, то влияние аппроксимированной матрицы Гессе $J^T J$ практически сводится к нулю. Таким образом, обеспечивается увеличение шага вдоль пологих участков функции и уменьшения шага вдоль крутых спусков.

Остановка алгоритма происходит в следующих случаях: если приращение Δw в следующей итерации меньше заданного значения; если вектор весовых коэффициентов сводит ошибку E_k к минимуму; если количество циклов обучения нейронной сети исчерпано. Значение весового коэффициента w на последней итерации считается искомым.

Таким образом, процесс обучения нейронной сети алгоритмом Левенберга-Марквардта можно записать в следующих шагах:

- 1) Вычисление ошибки нейронной сети за одну эпоху.
- 2) Вычисление элементов матрицы Якоби по формуле (2.8).
- 3) Вычисление весовых коэффициентов по формуле (2.9).
- 4) Вычисление ошибки нейронной сети для новых параметров. Если ошибка уменьшилась, то выполняется переход к шагу 6, иначе – переход к шагу 5.
- 5) Вернуться к предыдущим параметрам сети, увеличить регулирующий параметр λ и перейти к шагу 3. Обычно параметр λ увеличивают в 10 раз.
- 6) Принять новые значения параметров сети, уменьшить регулирующий параметр λ и перейти к новой эпохе обучения. Обычно параметр λ уменьшают в 10 раз [79, 174].

Генетический алгоритм

Генетический алгоритм является эвристическим алгоритмом поиска и используется для решения задач оптимизации. Генетические алгоритмы обрели свою популярность благодаря работам *J.H. Holland* в 70-х годах и его аспиранта

K.D. Jong [104, 121, 122]. Принцип работы данных алгоритмов заключается в последовательной обработке множества альтернативных решений, путем случайного комбинирования и поиска требуемых параметров, аналогично работе механизмов естественного отбора в природе.

В данном случае параметром оптимизации является среднеквадратичная ошибка нейронной сети на обучающей выборке.

На начальном этапе генетического алгоритма формируется начальная популяция, состоящая из особей. Каждая особь представляется в виде матрицы весовых коэффициентов нейронной сети w , которые задаются случайным образом из определенного интервала [49, 147].

Далее следует этап селекции. Из популяции размером N выбирается определенное количество особей. Наиболее приспособленные из выбранных особей записываются в промежуточный массив. Для выявления наиболее приспособленных особей выполняется оценивание популяции по формуле:

$$f_i = f(X_i),$$

где $X_i = \{x_{ik} : k=1, 2, \dots, N\}$ – хромосома i -й особи; x_{ik} – значение генов данной особи [76].

Вероятность скрещивания i -ой особи вычисляется по следующей формуле:

$$p_i = \frac{f_i}{\sum_j f_j}.$$

где p_i – вероятность скрещивания i -ой особи; f_i – значение приспособленности i -ой особи.

Отобранные в результате селекции особи скрещиваются. Определяется точка, в которой происходит разделение хромосом на части и обмен частями между особями. Скрещивание выполняется по следующей формуле:

$$h_k = \lambda g_k + (1 - \lambda)g_k, \quad 0 \leq \lambda \leq 1,$$

где g_k – k -й ген родительской особи; h_k – k -й ген потомка [76].

Далее выполняется операция мутации, которая используется для внесения случайных изменений в хромосомы особей и предотвращает попадание

популяции в локальный экстремум. Операция мутации с определенной вероятностью изменяет значение гена в хромосоме на противоположное. Таким образом, создается новая популяция.

После выполнения данного этапа вычисляется значение функции приспособленности для всех хромосом новой популяции. Условие остановки алгоритма может заключаться в достижении желаемого значения функции ошибки, либо при выполнении алгоритмом заданного количества эпох.

Использование генетического алгоритма для обучения нейронных сетей позволяет избежать попадания в локальные минимумы функции, в отличие от градиентных методов.

2.7.1 Результаты применения алгоритмов обучения

Вышеперечисленные алгоритмы были реализованы и применены для обучения разработанных СНС. Критерием остановки алгоритма обучения является незначительное изменение ошибки в течение 10 эпох. В таблицах 2.8, 2.9, 2.10 представлены результаты обучения разработанных СНС.

Таблица 2.8 – Результаты обучения СНС-1

Алгоритм обучения	Ошибка обучения	Количество эпох	Время обучения (часы)
Backpropagation	8.34E-02	34	12.3
Quick Propagation	3.34E-02	15	7.5
Resilient Propagation	1.56E-02	17	8.3
Генетический алгоритм	7.17E-02	19	13.6
Левенберга-Марквардта	5.83E-03	18	17.5

Таблица 2.9 – Результаты обучения СНС-2

Алгоритм обучения	Ошибка обучения	Количество эпох	Время обучения (часы)
Backpropagation	7.95E-02	38	16.3

Quick Propagation	2.58E-02	18	8.2
Resilent Propagation	1.71E-02	19	8.9
Генетический алгоритм	6.76E-02	27	20.4
Левенберга-Марквардта	4.12E-03	19	18.3

Из представленных таблиц видно, что наилучшие результаты обучения для СНС-1 и СНС-2 были получены при использовании алгоритма Левенберга-Марквардта: при обучении СНС-1 ошибка достигла значения 0.00583 (5.83E-03); для СНС-2 ошибка достигла значения 0.00412 (4.12E-03). При этом данный алгоритм обеспечивает сходимость за достаточно не большое количество эпох обучения.

Недостаток данного алгоритма заключается в необходимости преобразования обратной матрицы на каждом шаге, что может быть не приемлемым для его использования в системах слишком большого размера, содержащих десятки и сотни тысяч параметров.

Обучение СНС-3 осуществлялось с помощью следующих алгоритмов: Backpropagation, Quick Propagation и Resilent Propagation (таблица 2.10).

Таблица 2.10 – Результаты обучения СНС-3

Алгоритм обучения	Ошибка обучения	Количество эпох	Время обучения (часы)
Backpropagation	8.28E-02	42	18.1
Quick Propagation	4.84E-02	24	12.4
Resilent Propagation	1.24E-02	22	10.7

Из таблицы 2.10 видно, что наилучшие результаты обучения для СНС-3 были получены при использовании алгоритма Resilent Propagation: ошибка достигла значения 0.0124 (1.24E-02) за меньшее время обучения и меньшее количество эпох.

2.8 Основные результаты и выводы по главе 2

1. Предложена структура и конфигурация сверточных нейронных сетей, обеспечивающая необходимую надежность и высокую степень инвариантности к различным шумовым, аффинным и проекционным искажениям входных данных.

2. Разработан алгоритм детектирования области расположения символов на изображениях со сложной фоновой структурой, основанный на алгоритмической композиции из двух сверточных нейронных сетей, работающих по принципу последовательной классификации.

3. Предложена модификация алгоритма нормализации изображений символов, основанная на построении гистограмм распределения яркости пикселей, отличающаяся от известных реализаций меньшей вычислительной сложностью и обеспечивающая более высокую скорость работы. Алгоритм обеспечивает обнаружение точных границ пластины номерного знака, расположенного под различными углами наклона.

4. Предложен алгоритм распознавания символов, основанный на применении сверточной нейронной сети и позволяющий распознавать символы, не используя процесс их сегментации, что снижает его вычислительную сложность и повышает скорость работы.

5. Проведен обзор различных алгоритмов, использующихся для обучения сверточных нейронных сетей. На основе данных алгоритмов было реализовано обучение разработанных СНС. В процессе экспериментальных исследований наилучшие результаты обучения для СНС-1 и СНС-2 были получены при использовании алгоритма Левенберга-Марквардта. Для СНС-3 наилучшие результаты обучения были получены при использовании алгоритма Resilient Propagation.

ГЛАВА 3. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ РАСПОЗНАВАНИЯ СИМВОЛОВ НА СЛОЖНОМ ФОНЕ

В данной главе представлено описание программной реализации разработанных алгоритмов. Представлен обзор инструментальных библиотек по компьютерному зрению и обработке изображений и приведено обоснование выбора средств для программной реализации алгоритмов. Представлено описание архитектур разработанных библиотек, основные методы и переменные классов, логические схемы разработанных алгоритмов.

3.1 Общие требования к разрабатываемому программному обеспечению

Исходя из современных требований к качеству программного обеспечения, разрабатываемая программная система должна обладать следующими характеристиками:

- Корректная работа программной системы в решении поставленных задач.
- Устойчивость программной системы к различным ошибкам, сбоям и действиям, выходящим за рамки ее функционального предназначения.
- Небольшая вычислительная сложность используемых алгоритмов.
- Возможность расширять, изменять и дополнять функциональность разработанных библиотек. Для обеспечения этого пункта необходимо следовать принципам простоты построения программы и децентрализации.
- Программная система должна обладать модульной архитектурой, где каждый модуль предназначен для решения определенных задач.
- Не противоречивость программных модулей. Заключается в том, что все модули программной системы должны работать в едином слаженном режиме и не противоречить друг другу.
- Возможность повторного использования программных модулей. Обеспечивает сокращение времени разработки программной системы и сокращение размера программного кода.

В соответствии с поставленными задачами к разрабатываемой программной системе предъявляются следующие функциональные требования:

- 1) Выполнение предварительной обработки цифровых изображений для осуществления последующих операций с ними.
- 2) Возможность настройки классификаторов на основе обучающих данных, которые хранятся во внешних файлах.
- 3) Детектирование области расположения символов, в данном случае автомобильных номерных знаков, соответствующих ГОСТ Р 50577 – 93, группе 1, типу 1.
- 4) Осуществление нормализации изображений символов.
- 5) Распознавание символов на изображениях.
- 6) Предоставление данных о результатах работы программной системы:
 - время настройки классификаторов;
 - время выполнения операций классификации.

3.2 Выбор средств разработки программного обеспечения

Одним из самых широко используемых инструментов для реализации нейросетевых алгоритмов является пакет прикладных программ для решения задач технических вычислений Matlab Neural Network Toolbox. Для данного пакета программ существует несколько библиотек, направленных на реализацию сверточных нейронных сетей. Но разработка нейронных сетей в Matlab имеет ряд недостатков, которые заключаются в громоздкой реализации и низкой скорости работы.

Так же существует несколько библиотек для реализации СНС и алгоритмов их обучения, большинство из которых написаны на языке Python. В процессе анализа и тестирования некоторых библиотек было выявлено, что они не всегда корректно устанавливаются и работают. В итоге для решения поставленной задачи был сделан выбор в пользу программной реализации СНС, а так же алгоритмов их обучения.

На сегодняшний день существует множество языков программирования, как с общей, так и с научной направленностью. Один из наиболее востребованных и удобных языков для решения широкого диапазона задач является язык

программирования C++. Удобство использования данного языка программирования заключается в наличии большого количества различных библиотек и оптимизируемых алгоритмов. Так же данный язык программирования обеспечивает следующие достоинства при разработке нейронных сетей:

- Возможность реализации модульной иерархической структуры нейронной сети.
- Возможность создания гибкой и легко перестраиваемой модели нейронной сети.
- Предоставляет наиболее прозрачную реализацию нейронной сети.

Исходя из этого, в качестве языка программирования был выбран язык программирования C++, в качестве среды программирования было выбрано программное обеспечение Microsoft Visual Studio 2010.

Так же для решения задач в области компьютерного зрения и обработки изображений широко используются специализированные библиотеки. Наиболее популярной и распространенной является открытая библиотека **OpenCV** (Open Source Computer Vision Library) [149]. Библиотека содержит алгоритмы интерпретации изображений, устранения оптических искажений, анализа перемещения объектов, сегментации изображений и других численных алгоритмов общего назначения с открытым кодом.

Данная библиотека реализована на языках C, C++, также разрабатывается для Python, Java, Matlab и других языков.

Библиотека содержит следующие модули:

CXCORE – базовые структуры. Содержит: алгоритмы работы с памятью; алгоритмы преобразования типов данных; алгоритмы работы с матрицами; алгоритмы работы с 2D объектами.

CV – модуль, предназначенный для обработки изображений. Содержит: алгоритмы для обработки и анализа изображений; алгоритмы слежения за объектами; алгоритмы распознавания объектов; алгоритмы, предназначенные для калибровки камер.

ML – модуль, предназначенный для машинного обучения. Содержит: алгоритмы, предназначенные для классификации и анализа данных.

HighGUI – модуль, предназначенный для создания пользовательского интерфейса. Поддерживает: создание окон, вывод изображений, захват видео.

CVAUX – модуль, предназначенный для описания пространственного зрения. Содержит: алгоритмы описания черт лица, описания текстур.

CVCAM – модуль, предназначенный для захвата видео с цифровых камер.

Так же существует общедоступная платформа **AForge.NET** [88], предназначенная для решения задач в области искусственного интеллекта и обработки изображений. Данная платформа содержит следующие библиотеки:

AForge.Imaging – библиотека, предназначенная для обработки изображений.

AForge.Fuzzy – библиотека, предназначенная для нечетких вычислений.

AForge.Vision – библиотека, предназначенная для машинного зрения.

AForge.Neuro – библиотека, предназначенная для работы с нейронными сетями.

AForge.Genetic – библиотека, предназначенная для работы с генетическими алгоритмами.

AForge.MachineLearning – библиотека, предназначенная для машинного обучения.

AForge.Math – библиотека, предназначенная для математических операций.

AForge.Fuzzy – библиотека, предназначенная для нечетких вычислений.

AForge.Video – библиотека, предназначенная для обработки видео.

AForge.Robotics – библиотека, предназначенная для робототехнических комплексов.

Так же существует платформа **Accord.NET** [87], которая является расширением платформы AForge.NET с новым инструментарием и библиотеками. Данная платформа предназначена для научных вычислений и содержит следующие библиотеки:

Accord.Math – библиотека для математических операции. Содержит операции над матрицами и численные методы оптимизации.

Accord.Statistics – библиотека для работы со статистическими методами.

Accord.MachineLearning – библиотека для машинного обучения. Содержит деревья решений, сети Байеса, алгоритм К-средних.

Accord.Neuro – библиотека для работы с нейронными сетями. Содержит инструменты для реализации нейронных сетей, алгоритмы обучения нейронных сетей.

Accord.Imaging – библиотека для обработки изображений.

Accord.Audio – библиотека для обработки аудио сигналов. Содержит методы фильтрации и преобразования аудио сигналов.

Accord.Vision – библиотека для работы с видео последовательностями. Содержит методы обнаружения, отслеживания и распознавания объектов на видео последовательностях.

Перечисленные библиотеки были использованы в качестве вспомогательных средств при реализации программной системы.

3.3 Разработка программной системы

В процессе разработки алгоритмов было решено, что программная система должна обладать модульной архитектурой и состоять из нескольких не зависимых модулей, предназначенных для решения определенных задач. Модульная архитектура позволяет обеспечить возможность проведения исследований и тестирования каждого отдельного модуля.

Разработанная программная система имеет название Number Plate Recognition (NPR) и имеет модульную архитектуру. Каждый программный модуль объединяет классы по их функциональному назначению. В таблице 3.1. представлено краткое описание программных модулей.

Таблица 3.1 – Описание основных модулей программной системы

Модуль	Описание
NRP_Main	Основной исполняемый модуль. Модуль программного приложения, который содержит классы компонентов пользовательского интерфейса.

NRP_ImgProc	Модуль, предназначенный для выполнения операций обработки изображений. Данный модуль обеспечивает инкапсуляцию методов обработки изображений и является связующим звеном между клиентским кодом и вызовами функций библиотеки OpenCV. Входными данными модуля являются ненормализованные изображения. Выходными данными являются нормализованные изображения.
NRP_Detect	Модуль, предназначенный для детектирования области расположения символов, в данном случае автомобильных номерных знаков на изображениях, на основе алгоритма, описанного во второй главе. Модуль включает в себя библиотеки классов, позволяющие использовать аппарат сверточных нейронных сетей, алгоритмы работы с данными и алгоритмы обучения классификаторов. Входными данными модуля являются: • изображения со сложной фоновой структурой, на которых предположительно присутствуют образы автомобильных номерных знаков; • файлы с описанием обучающих данных (маркированные примеры) для осуществления настройки классификатора. Выходными данными являются: • сегментированные области изображений, содержащие автомобильные номерные знаки; • информация о времени выполнения детектирования; • информация о времени настройки классификатора.
NRP_Recognize	Модуль, предназначенный для распознавания символов на основе алгоритма, описанного во второй главе. Входными данными модуля являются:

	• нормализованные изображения автомобильных номерных знаков; • файлы с описанием обучающих данных. Выходными данными являются: • результат распознавания номерного знака в виде строки символов; • информация о времени выполнения распознавания; • информация о времени настройки классификатора.
NRP_Sub	Модуль, предназначенный для выполнения вспомогательных операций. Данный модуль содержит классы для выполнения и отладки системы и является доступным для всех остальных модулей.

В таблице 3.2. представлены минимальные системные требования, которые необходимы для работы программной системы.

Таблица 3.2 – Минимальные системные требования

Название параметра	Характеристика
Операционная система (ОС)	Microsoft Windows 7
Центральный процессор (ЦП)	Intel Core 2
Оперативная память (ОЗУ)	512 Мб
Видеоадаптер	256 Мб

3.3.1 Классы для реализации сверточных нейронных сетей

В разработанной программной системе с помощью СНС осуществляется детектирование области расположения символов на изображениях и распознавание символов. Для использования аппарата СНС и алгоритмов обучения классификаторов реализована библиотека классов: ConvNeuralNetwork, CNInputPlane, CNConvolutionPlane, CNSubsamlePlane, CNNeuronLayer, CNSubPlane, CNNeuron,

CNCommonPlane. На рисунке 3.1 представлена диаграмма классов, реализующих сверточную нейронную сеть.

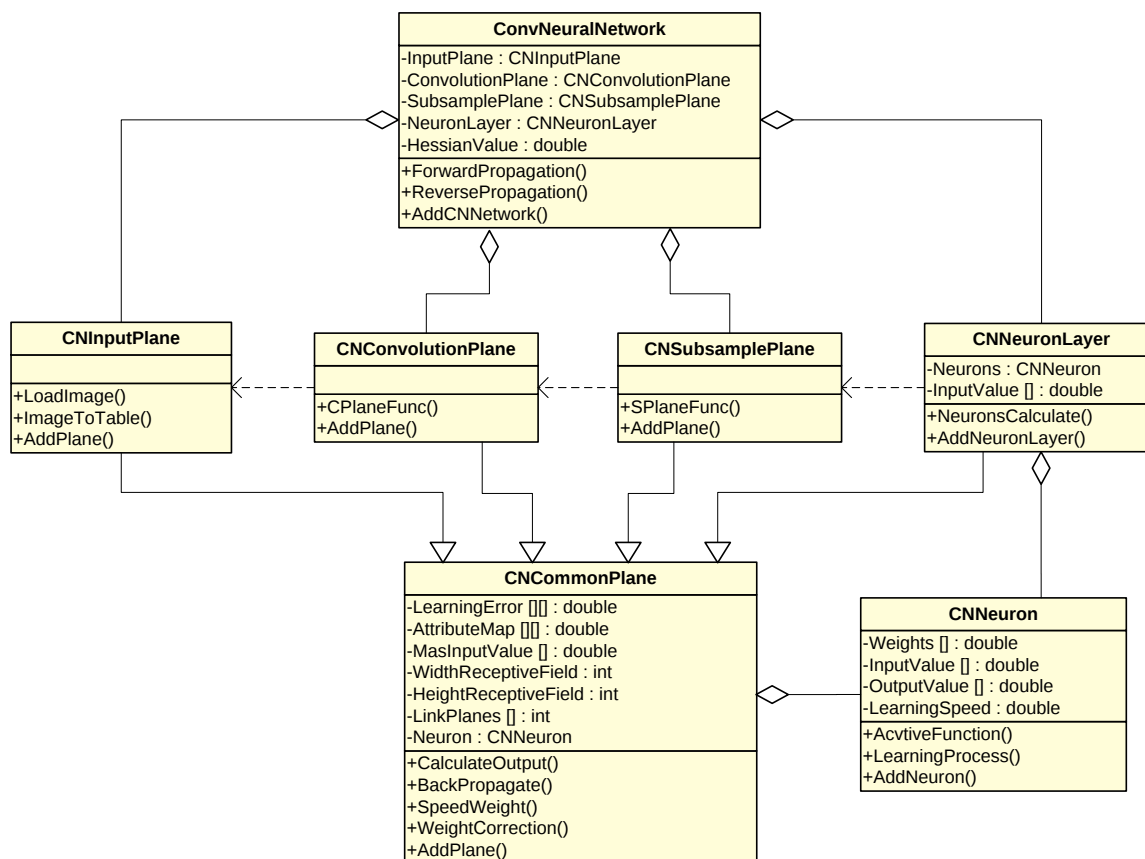


Рисунок 3.1 – Диаграмма классов, реализующих сверточную нейронную сеть

Класс ConvNeuralNetwork

Класс ConvNeuralNetwork предназначен для реализации сверточной нейронной сети и содержит методы, обеспечивающие ее функционирование. Данный класс содержит метод ForwardPropagation, который осуществляет прямой проход по нейронной сети и вычисляет выходные значения нейронов на основе их входных значений. Метод ReversePropagation осуществляет обратный проход по нейронной сети и расчет градиентов для синаптических коэффициентов.

В переменную HessianValue записывается значение Гессе для изображений из обучающей выборки. При обратном проходе по нейронной сети значение HessianValue используется для коррекции скорости обучения для

весовых коэффициентов. В таблице 3.3 представлено описание основных атрибутов и переменных данного класса.

Таблица 3.3 – Описание основных атрибутов и переменных класса `ConvNeuralNetwork`

Атрибут/Переменная	Описание
<code>CNInputPlane</code> <code>InputPlane</code>	Входной слой нейронной сети
<code>CNConvolutionPlane</code> <code>ConvolutionPlane</code>	Сверточная плоскость нейронной сети
<code>CNSubsamplePlane</code> <code>SubsamplePlane</code>	Подвыборочная плоскость нейронной сети
<code>CNNeuronLayer</code> <code>NeuronLayer</code>	Слой нейронов
<code>double HessianValue</code>	Переменная, содержащая значение Гессе

В таблице 3.4 представлено описание основных методов данного класса.

Таблица 3.4 – Описание основных методов класса `ConvNeuralNetwork`

Метод	Описание
<code>ForwardPropagation</code>	Прямой проход по сверточной нейронной сети
<code>ReversePropagation</code>	Обратный проход по сверточной нейронной сети и расчет градиентов
<code>AddCNNNetwork</code>	Создание сверточной нейронной сети

Класс `CNNeuron`

Класс `CNNeuron` предназначен для реализации нейрона и содержит информацию и методы, обеспечивающие его функционирование. Данный класс содержит входные и выходные значения нейронов, значения весовых коэффициентов. Выходное значение нейрона формируется после вызова функции `CalculateOutput` в классе `CNCommonPlane`. Методы `ActiveFunction` и `LearningProcess` предназначены для активации нейрона и реализации

процесса обучения соответственно. В таблице 3.5 представлено описание основных атрибутов и переменных данного класса.

Таблица 3.5 – Описание основных атрибутов и переменных класса `CNNeuron`

Атрибут/Переменная	Описание
<code>double InputValue[]</code>	Входное значение нейрона
<code>double OutputValue[]</code>	Выходное значение нейрона
<code>double Weights[]</code>	Весовые коэффициенты
<code>double LearningSpeed</code>	Параметр скорости обучения

В таблице 3.6 представлено описание основных методов данного класса.

Таблица 3.6 – Описание основных методов класса `CNNeuron`

Метод	Описание
<code>ActiveFunction</code>	Реализация активации нейрона
<code>LearningProcess</code>	Реализация процесса обучения
<code>AddNeuron</code>	Создание нейрона

Класс `CNNCommonPlane`

Класс `CNNCommonPlane` описывает процесс функционирования всех типов плоскостей (входная, подвыборочная, сверточная, слои нейронов) и содержит методы, обеспечивающие процесс обучения. Данный класс содержит массив входных значений, параметры локального рецептивного поля, карты признаков, нейроны. Метод `CalculateOutput` осуществляет вычисление выходных значений нейронов плоскости. Метод `BackPropagate` реализует процесс обратного распространения. Метод `WeightCorrection` выполняет процесс корректировки синаптических коэффициентов. Метод `SpeedWeight` производит расчет скорости обучения для каждого синаптического коэффициента. В таблице 3.7 представлено описание основных атрибутов и переменных данного класса.

Таблица 3.7 – Описание основных атрибутов и переменных класса CNNCommonPlane

Атрибут/Переменная	Описание
double LearningError[]	Ошибка обучения
double AttributeMap[]	Карта признаков
double MasInputValue[]	Массив входных значений
int WidthReceptiveField	Ширина окна локального рецептивного поля
int HeighReceptiveField	Высота окна локального рецептивного поля
int LinkPlanes[]	Связи
CNNNeuron Neuron	Нейроны

В таблице 3.8 представлено описание основных методов данного класса.

Таблица 3.8 – Описание основных методов класса CNNCommonPlane

Метод	Описание
CalculateOutput	Вычисление выходных значений нейронов
BackPropagate	Реализация процесса обратного распространения ошибки
SpeedWeight	Вычисление скорости обучения для синаптических коэффициентов
WeightCorrection	Расчет коррекции и корректировка синаптических коэффициентов
AddPlane	Создание плоскости

Класс CNInputPlane

Класс CNInputPlane предназначен для реализации плоскости входного слоя и осуществляет работу с изображением, т.е. представляет изображение в виде двумерного массива. В таблице 3.9 представлено описание основных элементов данного класса.

Таблица 3.9 – Описание основных элементов класса `CNInputPlane`

Переменная/Метод	Описание
<code>LoadImage</code>	Загрузка изображения
<code>ImageToTable</code>	Представляет изображение в виде двумерного массива
<code>AddPlane</code>	Создание плоскости

Класс `CNConvolutionPlane`

Класс `CNConvolutionPlane` предназначен для реализации сверточной плоскости и процесса ее функционирования. В таблице 3.10 представлено описание основных методов данного класса.

Таблица 3.10 – Описание основных методов класса `CNConvolutionPlane`

Метод	Описание
<code>CPlaneFunc</code>	Реализация функционирования сверточной плоскости
<code>AddPlane</code>	Создание плоскости

Класс `CNSubsamplePlane`

Класс `CNSubsamplePlane` предназначен для реализации подвыборочной плоскости и процесса ее функционирования. В таблице 3.11 представлено описание основных методов данного класса.

Таблица 3.11 – Описание основных методов класса `CNSubsamplePlane`

Метод	Описание
<code>SPlaneFunc</code>	Реализация функционирования подвыборочной плоскости
<code>AddPlane</code>	Создание плоскости

Класс `CNNeuronLayer`

Класс `CNNNeuronLayer` предназначен для реализации слоя, состоящего из нейронов. Данный класс содержит нейроны, массив входных значений и метод `NeuronsCalculate`, вычисляющий выходные значения всех нейронов. В таблице 3.12 представлено описание основных атрибутов и переменных данного класса.

Таблица 3.12 – Описание основных атрибутов и переменных класса `CNNNeuronLayer`

Атрибут/Переменная	Описание
<code>CNNNeuron Neurons</code>	Нейроны
<code>double InputValue[]</code>	Входные значения

В таблице 3.13 представлено описание основных методов данного класса.

Таблица 3.13 – Описание основных методов класса `CNNNeuronLayer`

Метод	Описание
<code>NeuronsCalculate</code>	Вычисление выходных значений всех нейронов
<code>AddNeuronLayer</code>	Создание слоя

3.3.2 Классы для работы с изображениями

Для более удобной работы с изображениями и выполнения необходимых операций были реализованы следующие классы: `ImageConverter` (для осуществления простейших операций с изображениями), `ColorConverter` (преобразование изображения в другое цветовое пространство), `ImageModification` (масштабирование, сглаживание, интерполяция, выделение областей изображения).

Класс `ImageConverter`

Класс `ImageConverter` предназначен для выполнения простых операций с изображениями: загрузка изображения, преобразование изображения в виде

одномерного и двумерного массива. В таблице 3.14 представлено описание основных переменных данного класса.

Таблица 3.14 – Описание основных переменных класса ImageConverter

Переменная	Описание
inhigh_top	Высота изображения
intwidth_top	Ширина изображения
int alignment	Количество выравнивающих байтов

Метод LoadImage предназначен для загрузки изображения. Метод ConvertToVect осуществляет преобразование изображения из матричного вида в одномерный вектор вещественных чисел в виде списка упорядоченных элементов. Метод ConvertToVectB выполняет преобразование изображения из матричного вида в одномерный вектор вещественных чисел с пороговой бинаризацией. Метод ConvertToMatrix предназначен для преобразования векторного представления данных в матричный вид. В таблице 3.15. представлено описание основных методов данного класса.

Таблица 3.15 – Описание основных методов класса ImageProc

Метод	Описание
LoadImage	Загрузка изображения
ConvertToVect	Преобразование изображения из матричного вида в одномерный вектор вещественных чисел
ConvertToVectB	Преобразование изображения из матричного вида в одномерный вектор вещественных чисел (0, 1)
ConvertToMatrix	Преобразование векторного представления данных в матричный вид

Класс ColorConverter

Класс `ColorConverter` предназначен для преобразования изображения из RGB-пространства в оттенки серого и выполнения пороговой бинаризации. В таблице 3.16 представлено описание основных методов данного класса.

Таблица 3.16 – Описание основных методов класса `ColorConverter`

Метод	Описание
<code>AdaptiveThreshold</code>	Преобразование изображения из RGB-пространства в оттенки серого
<code>Threshold</code>	Формирование бинарного изображения, состоящего из значений 0 и 1

Класс `ImageModification`

Класс `ImageModification` предназначен для осуществления необходимых операций с изображениями: масштабирование изображения, сглаживание, интерполяция, выделение областей изображения, сравнение параметров областей изображения. В таблице 3.17 представлено описание основных методов данного класса.

Таблица 3.17 – Описание основных методов класса `ImageModification`

Метод	Описание
<code>ImageResize</code>	Изменение размеров изображения (масштабирование)
<code>Smooth</code>	Устранение шума на изображении (сглаживание)
<code>Interpolation</code>	Интерполяция изображения
<code>MatOperator</code>	Выделение области изображения
<code>CompareMatr</code>	Сравнение параметров областей изображения
<code>GetLineArea</code>	Выделение четырех границ области изображения

3.3.3 Классы для нормализации изображений

После обнаружения на изображении областей, содержащих автомобильные номерные знаки, необходимо выполнить их нормализацию. Данный алгоритм включает в себя следующие операции: повышение контраста и выделение границ

на изображении; нахождение горизонтальных граней пластины автомобильного номерного знака, выравнивание изображения, выделение символов. Данные процессы реализованы на основе следующих классов: `ImageSelectionBorder`, `GetImageProjections`, `ImageRotate`, `SelectCharacters`.

Класс `ImageSelectionBorder`

Класс `ImageSelectionBorder` предназначен для осуществления операций повышение контраста на изображении и выделение границ. Метод `HistogramEqualization` выполняет повышение контраста на изображении. Метод `Sobel` предназначен для выделения границ на изображении оператором Собеля. В таблице 3.18 представлено описание основных методов данного класса.

Таблица 3.18 – Описание основных методов класса `ImageSelectionBorder`

Метод	Описание
<code>HistogramEqualization</code>	Повышение контраста на изображении
<code>Sobel</code>	Выделение границ на изображении

Класс `GetImageProjections`

Класс `GetImageProjections` предназначен для построения гистограмм распределения яркости пикселей по строкам изображения. Данный класс содержит информацию о количестве строк и столбцов изображения; о количестве пикселей с высокой интенсивностью, расположенных в строках изображения; предельные значения углов построения гистограмм и т.д. В таблице 3.19 представлено описание основных переменных данного класса.

Таблица 3.19 – Описание основных переменных класса `GetImageProjections`

Переменная	Описание
<code>int white_pix[]</code>	Количество пикселей с высокой интенсивностью, расположенных в строках изображения
<code>int w_size</code>	Количество строк изображения

int h_size	Количество столбцов изображения
int max_pixels	Максимальное количество белых пикселей
int threshold	Пороговое значение
int bottom_angle	Нижний предел угла построения гистограмм
int top_angle	Верхний предел угла построения гистограмм

Класс `GetImageProjections` содержит метод `CreateHist`, с помощью которого осуществляется построение горизонтальных гистограмм распределения яркости пикселей. С помощью метода `LinesCalculate` выполняется вычисление количества строк с максимальной интенсивностью пикселей. В таблице 3.20 представлено описание основных методов данного класса.

Таблица 3.20 – Описание основных методов класса `HorizontalProjections`

Метод	Описание
<code>CreateHist</code>	Построение горизонтальных гистограмм
<code>LinesCalculate</code>	Вычисление количества строк с максимальной интенсивностью пикселей

Класс `ImageRotate`

Класс `ImageRotate` предназначен для осуществления выравнивания изображения и обрезки его краев. Метод `ImageRotate` выполняет выравнивание изображения на основе информации о проекциях гистограмм по строкам изображения. Метод `ImageCrop` предназначен для обрезки краев изображения до заданного размера. В таблице 3.21 представлено описание основных методов данного класса.

Таблица 3.21 – Описание основных методов класса `ImageRotate`

Переменная/Метод	Описание
<code>ImageRotate</code>	Выравнивание изображения
<code>ImageCrop</code>	Обрезка изображения

Класс SelectCharacters

Класс `SelectCharacters` предназначен для выделения областей изображения, содержащих символы автомобильного номерного знака. Данный класс содержит переменные, в которых содержатся значения параметров областей изображения: количество, координаты границ и т.д. В таблице 3.22 представлено описание основных переменных данного класса.

Таблица 3.22 – Описание основных переменных класса `SelectCharacters`

Переменная/Метод	Описание
<code>int amount_charact</code>	Количество областей
<code>int coord_charact</code>	Координаты обнаруженных областей
<code>int up_charact</code>	Координата верхней границы символа
<code>int bottom_charact</code>	Координата нижней границы символа
<code>int left_charact</code>	Координата левой границы символа
<code>int right_charact</code>	Координата правой границы символа
<code>int rect_charact[]</code>	Список четырех границ области символа

Класс `SelectCharacters` содержит метод `GetLineArea` который выполняет выделение четырех границ области изображения, содержащей символ. В таблице 3.23 представлено описание основных методов данного класса.

Таблица 3.23 – Описание основных методов класса `SelectCharacters`

Метод	Описание
<code>GetLineArea</code>	Выделение четырех границ области

3.4 Логические схемы разработанных алгоритмов

Общая логическая схема работы программной системы

На рисунке 3.2 представлена общая логическая схема работы программной системы.

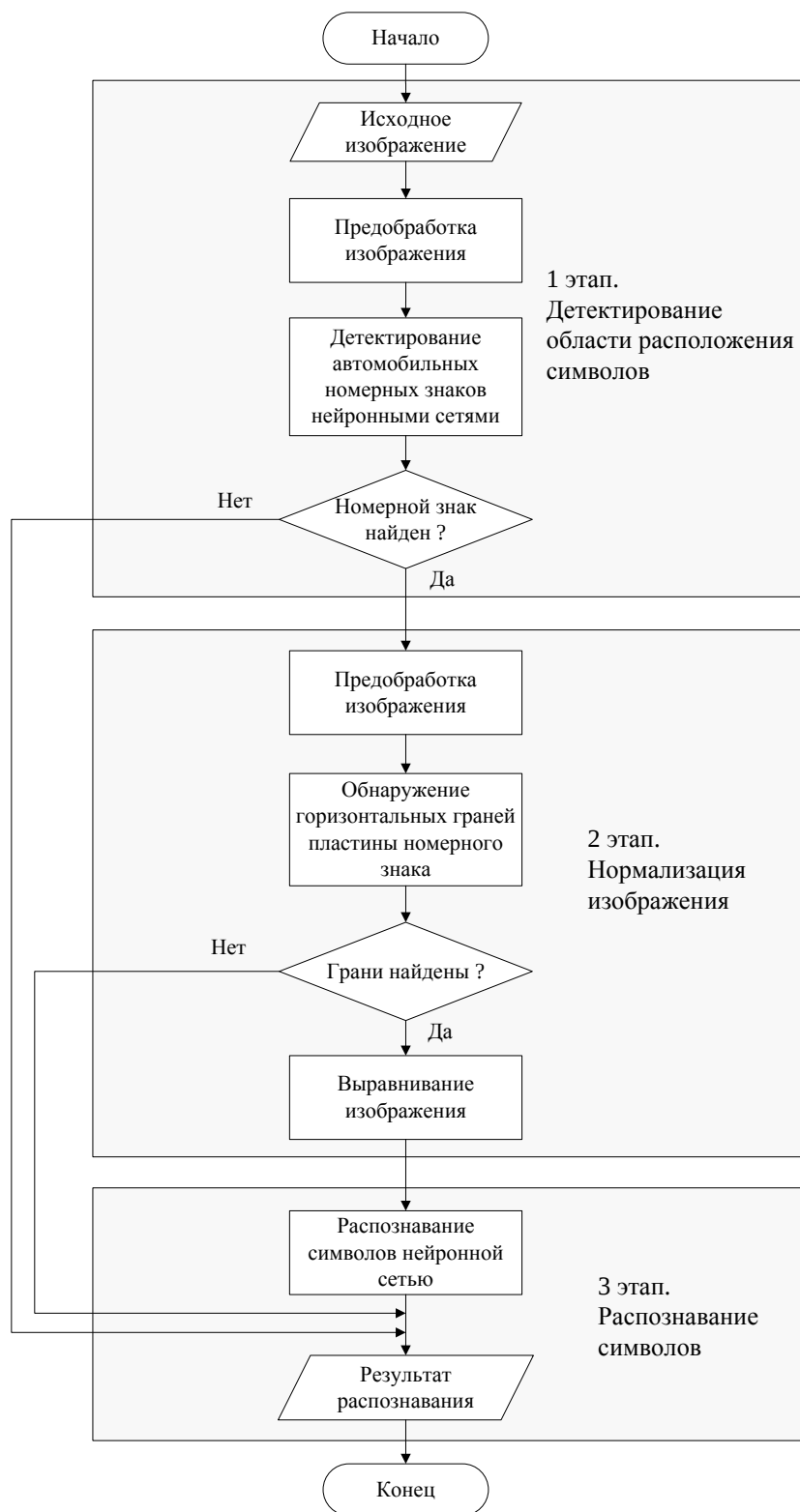


Рисунок 3.2 – Общая логическая схема работы программной системы

Данная логическая схема отображает три этапа процесса распознавания автомобильных номерных знаков на изображениях:

1) Детектирование области расположения символов на изображении. Исходное изображение преобразуется из цветного пространства RGB в полутоновое и поступает на вход программной системы. Предположительно на входном изображении должны присутствовать образы автомобильных номерных знаков, которые необходимо распознать.

Выполняется последовательное сканирование входного изображения разработанными СНС. Согласно алгоритму детектирования, описанному в главе 2, сначала выполняется предварительная классификация сетью СНС-1, затем выполняется итоговая классификация сетью СНС-2. В тех местах изображения, которые по визуальным признакам соответствуют образам из обучающей выборки, отклики нейронных сетей образуют максимальные значения. Далее происходит выделение найденных областей.

2) Нормализация изображения. Области изображения, выделенные на первом этапе, подвергаются процессу предобработки, который заключается в повышении контраста и выделении границ. Далее происходит поиск горизонтальных граней пластины номерного знака. Данный процесс выполняется путем построения гистограмм распределения яркости пикселей под различными углами. После нахождения горизонтальных граней пластины номерного знака, выполняется выравнивание изображения.

3) Распознавание символов. Нормализованное изображение автомобильного номерного знака поступает на вход СНС-3. В области расположения символов на изображении, отклики нейронной сети будут превышать заданный порог. Каждый отклик соответствует одному из нейронов в выходном слое нейронной сети и определенному классу символов. Таким образом, выполняется классификация символов, изображенных на пластине номерного знака.

Логическая схема алгоритма детектирования области расположения символов

На рисунке 3.3 изображена логическая схема алгоритма детектирования области расположения символов на изображении.

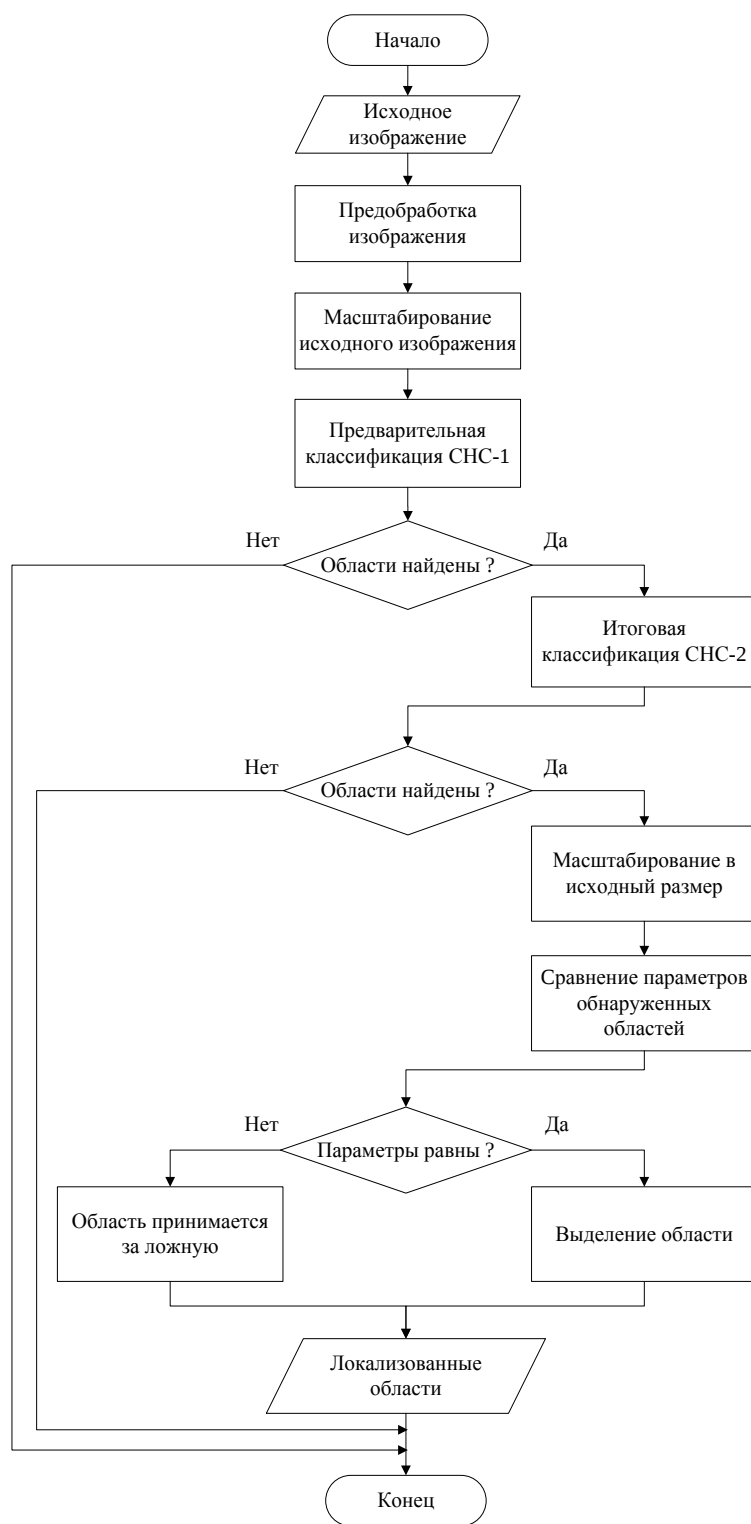


Рисунок 3.3 – Логическая схема алгоритма детектирования области расположения символов

Данная логическая схема отображает следующие этапы:

1) Предобработка входного изображения. Исходное изображение преобразуется из цветного пространства RGB в полутоновое.

2) Масштабирование входного изображения. Изображение последовательно масштабируется с коэффициентами от 0.8 до 1.2. Получается набор из пяти изображений разного масштаба.

3) Предварительная классификация. Полученные изображения пропускаются через СНС-1. На данном этапе возможны ложные обнаружения.

4) Итоговая классификация. СНС-2 производит сканирование областей, выявленных на шаге 3. На данном этапе устраняется часть ложных обнаружений.

5) Масштабирование изображений относительно исходного размера.

6) Сравнение параметров обнаруженных областей: координаты расположения, границы. Перекрывающиеся обнаруженные области группируются в кластеры. Области, имеющие общие параметры принимаются за истинные обнаружения, остальные области принимаются за ложные обнаружения и отбрасываются из рассмотрения.

7) Выделение обнаруженных областей.

Логическая схема алгоритма нормализации изображений

На рисунке 3.4 изображена логическая схема алгоритма нормализации изображений.

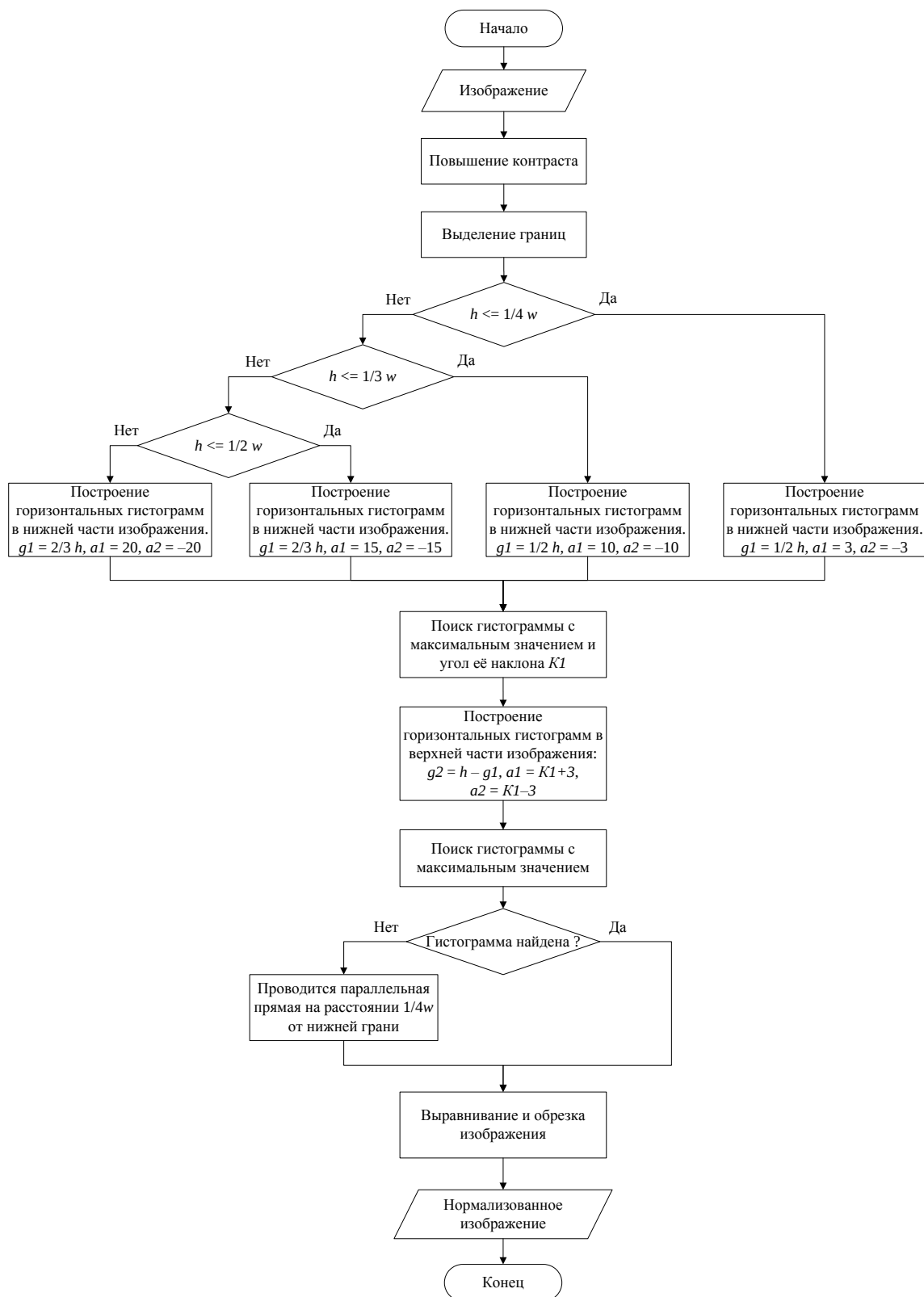


Рисунок 3.4 – Логическая схема алгоритма нормализации изображений

Данная логическая схема отображает следующие этапы:

- 1) Повышение контраста на изображении.

2) Выделение границ на изображении. После этого границы пластины номерного знака будут иметь максимальные значения яркости пикселей.

3) Поиск нижней горизонтальной грани пластины номерного знака. Производится построение горизонтальных гистограмм распределения яркости пикселей в нижней части изображения.

4) Поиск гистограммы с максимальным значением интенсивности пикселей. Среди всех вычисленных гистограмм выбирается гистограмма, которая содержит наибольшее значение по оси y .

5) Поиск верхней горизонтальной грани пластины номерного знака. Производится построение горизонтальных гистограмм распределения яркости пикселей в верхней части изображения.

6) Поиск гистограммы с максимальным значением интенсивности пикселей. Среди всех вычисленных гистограмм выбирается гистограмма, которая содержит наибольшее значение по оси y .

7) Выравнивание изображения. Изображение выравнивается в горизонтальную ориентацию относительно найденных горизонтальных граней пластины номерного знака.

Логическая схема алгоритма распознавания символов

На рисунке 3.5 изображена логическая схема алгоритма распознавания СИМВОЛОВ.

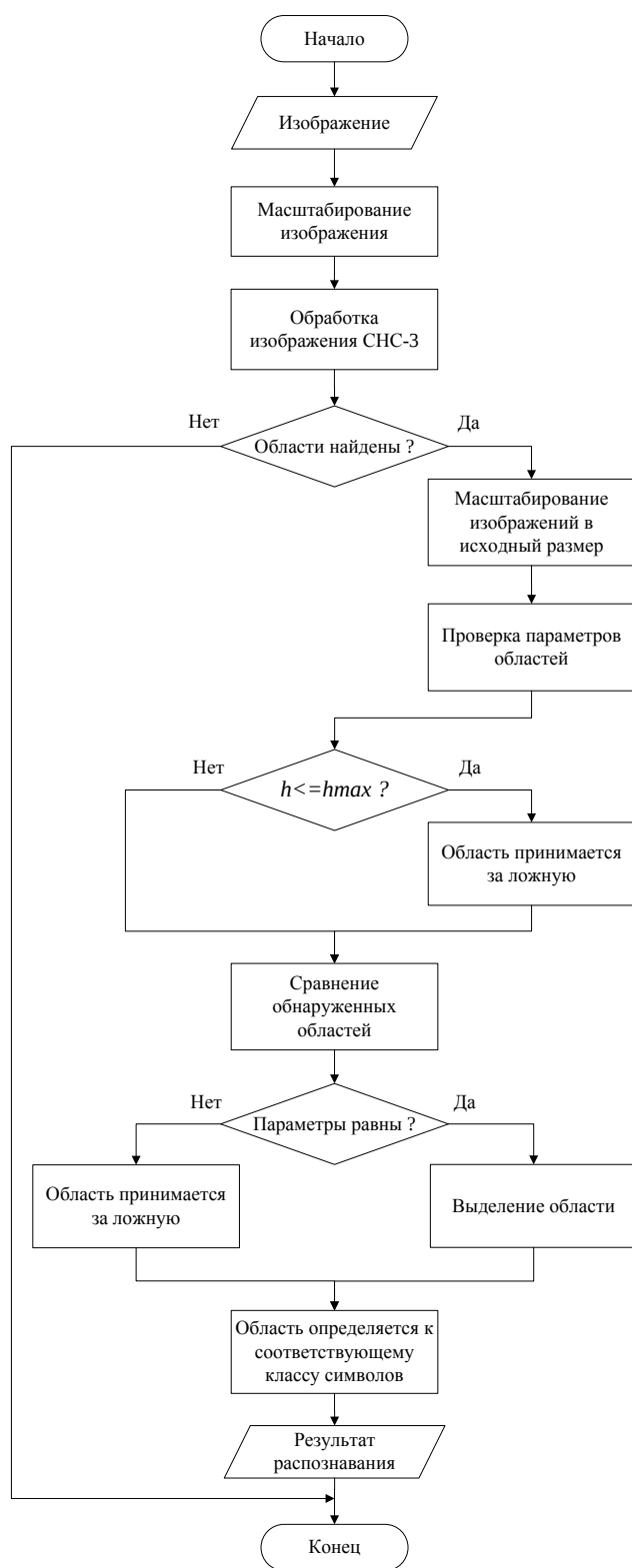


Рисунок 3.5 – Логическая схема алгоритма распознавания символов

Данная логическая схема отображает следующие этапы:

- 1) Масштабирование входного изображения. Изображение последовательно масштабируется с коэффициентами от 1.0 до 1.2.

2) Обработка изображений СНС-3. В области расположения символов на изображении, отклики нейронной сети будут превышать заданный порог.

3) Масштабирование изображений относительно исходного размера.

4) Проверка параметров найденных областей на соответствие требуемым условиям. Если $h \leq 0,3h_{max}$, то данная область отбрасывается из рассмотрения. Где, h – высота области расположения символа, h_{max} – высота изображения.

5) Сравнение параметров найденных областей. Области, имеющие общие параметры принимаются за истинные обнаружения, остальные области принимаются за ложные обнаружения.

6) Найденные области изображения выделяются прямоугольной рамкой. Классифицированные символы, соответствующие найденным областям, выводятся на экран в текстовом виде.

Логическая схема алгоритма обучения сверточных нейронных сетей

На рисунке 3.6 изображена логическая схема алгоритма обучения сверточных нейронных сетей по методу Левенберга-Марквардта.



Рисунок 3.6 – Логическая схема алгоритма обучения сверточных нейронных сетей

Данная логическая схема отображает следующие этапы:

1) Вычисление ошибки нейронной сети за одну эпоху по формуле:

$$f(y) = \sum_{i=1}^K \sum_{j=1}^P (y_{ij} - d_{ij})^2,$$

где y_{ij} – текущее значение j -ого выхода сети для i -ого элемента обучающего множества; d_{ij} – ожидаемое значение j -ого выхода сети для i -ого элемента обучающего множества; K – количество весов сети; P – количество образов в обучающей выборке.

2) Вычисление матрицы Якоби по формуле:

$$\tilde{J}_t = \begin{bmatrix} \frac{\partial f(w, x_1)}{\partial w_1} & \frac{\partial f(w, x_1)}{\partial w_2} & \dots & \frac{\partial f(w, x_1)}{\partial w_R} \\ \frac{\partial f(w, x_2)}{\partial w_1} & \frac{\partial f(w, x_2)}{\partial w_2} & \dots & \frac{\partial f(w, x_2)}{\partial w_R} \\ \dots & \dots & \dots & \dots \\ \frac{\partial f(w, x_N)}{\partial w_1} & \frac{\partial f(w, x_N)}{\partial w_2} & \dots & \frac{\partial f(w, x_N)}{\partial w_R} \end{bmatrix}.$$

3) Вычисление весовых коэффициентов по формуле:

$$\Delta w = (J^T J + \lambda I)^{-1} J^T (y - f(w)), \quad (2.7)$$

где, Δw – приращение весовых коэффициентов, λ – регулирующий параметр.

4) Вычисление ошибки нейронной сети для новых параметров. Если ошибка уменьшилась, то выполняется переход к шагу 6, иначе – переход к шагу 5.

5) Вернуться к предыдущим параметрам сети, увеличить регулирующий параметр λ и перейти к шагу 3.

6) Принять новые значения параметров сети, уменьшить регулирующий параметр λ и перейти к новой эпохе обучения.

3.5 Пользовательский интерфейс программной системы

При запуске программной системы открывается форма главного интерфейса (рисунок 3.7).

На главном интерфейсе программы сверху расположены выпадающие меню «File» (1), «Settings» (2), «Help» (3). Под выпадающими меню на интерфейсе расположены вкладки «Automatic recognition» (4) и «Incremental recognition» (5), обозначающие автоматическое распознавание и пошаговое распознавание. Справа на форме расположены кнопки: «Open image» (6), предназначенная для загрузки изображения; «Save result» (7) для сохранения результатов распознавания в базу данных; «Recognition» (8) для запуска процесса распознавания; «View license plate» (9) для визуального отображения распознанного автомобильного номерного знака.

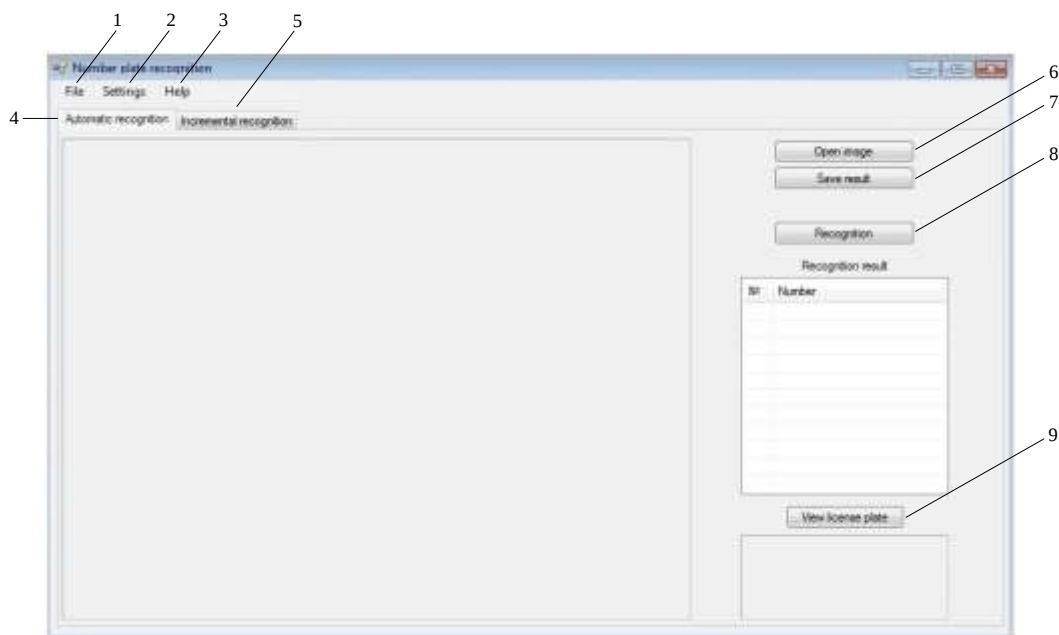


Рисунок 3.7 – Главный интерфейс программы

Меню «*File*» (рисунок 3.8) предназначено для загрузки изображения, сохранения результатов и открытия базы данных номерных знаков.

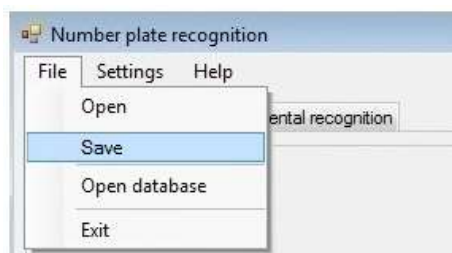


Рисунок 3.8 – Меню «*File*»

Меню «*Settings*» (рисунок 3.9) содержит 4 раздела:

- «*CNN-1 configuration*» – настройка параметров СНС-1.
- «*CNN-2 configuration*» – настройка параметров СНС-2.
- «*CNN-3 configuration*» – настройка параметров СНС-3.
- «*Common settings*» – настройка общих параметров.

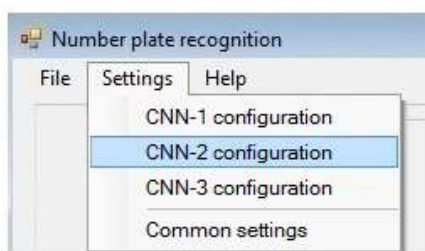


Рисунок 3.9 – Меню «Settings»

При выборе пункта меню «Settings→CNN-1 configuration» откроется форма настройки параметров СНС-1 (рисунок 3.10). Данная форма содержит две вкладки:

- «Training» – настройки обучения сверточной нейронной сети.
- «Testing» – настройки тестирования сверточной нейронной сети.

На вкладке «Training» в группе «Training parameters» содержатся настраиваемые параметры:

- «Data streams» – количество потоков данных.
- «Initial learning rate» – начальная скорость обучения.
- «Min learning rate» – минимальная скорость обучения
- «Mean-square error» – среднеквадратическая ошибка.
- «Number initial image» – номер начального образа.

Группа «Learning algorithm» предназначена для выбора алгоритма обучения нейронной сети и содержит элементы:

- «Algorithm BackProp» – алгоритм обратного распространения ошибки.
- «Algorithm QuickProp» – алгоритм QuickProp.
- «Algorithm RProp» – алгоритм RProp.
- «Algorithm LM» – алгоритм Левенберга-Марквардта.

В поле ввода данных «Training data file» указывается путь к файлу с данными для обучения сверточной нейронной сети. При нажатии кнопки «Start training» происходит запуск процесса обучения. С помощью кнопки «Stop training» можно остановить процесс обучения.

В каждой эпохе обучения образы из обучающей выборки проходят через нейронную сеть в произвольном порядке. Процесс обучения отображается в рабочей области. В самом низу формы отображается количество выполненных эпох обучения и номер обрабатываемого образа.



Рисунок 3.10 – Форма настройки параметров обучения СНС-1

При переходе на вкладку «*Testing*» откроется форма, изображенная на рисунке 3.11. Данная форма содержит настраиваемые параметры:

- «*Data streams*» – количество потоков данных.
- «*Number initial image*» – номер начального образа.

В поле ввода данных «*Testing data file*» указывается путь к файлу с данными для тестирования сверточной нейронной сети. При нажатии кнопки «*Start testing*» происходит запуск процесса тестирования. С помощью кнопки «*Stop testing*» можно остановить процесс тестирования.

По центру расположена панель, отображающая процесс тестирования: номер обрабатываемого образа, тестовое значение образа, реальное значение образа. В самом низу формы отображается номер обрабатываемого образа.

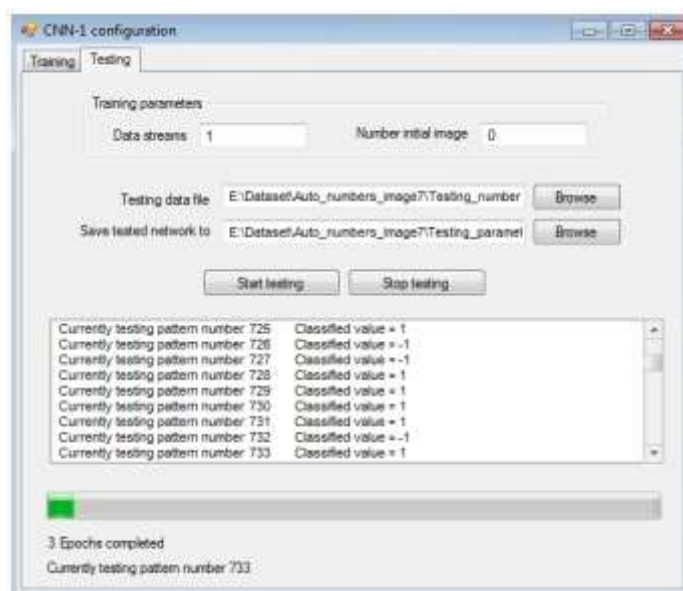


Рисунок 3.11 – Форма настройки параметров тестирования СНС-1

При выборе пункта меню «*Settings—>CNN-2 configuration*» откроется форма настройки параметров СНС-2. Данная форма является идентичной форме «*CNN-1 configuration*».

При выборе пункта меню «*Settings—>CNN-3 configuration*» откроется форма настройки параметров СНС-3 (рисунок 3.12). Данная форма настроек так же содержит вкладки «*Training*» и «*Testing*».

Вкладка «*Training*» (рисунок 3.12) содержит настраиваемые параметры: «*Data streams*», «*Initial learning rate*», «*Min learning rate*», «*Mean-square error*», «*Number initial image*» и выбор алгоритма обучения нейронной сети. Процесс обучения отображается в рабочей области. В самом низу формы отображается количество выполненных эпох обучения и номер обрабатываемого образа.

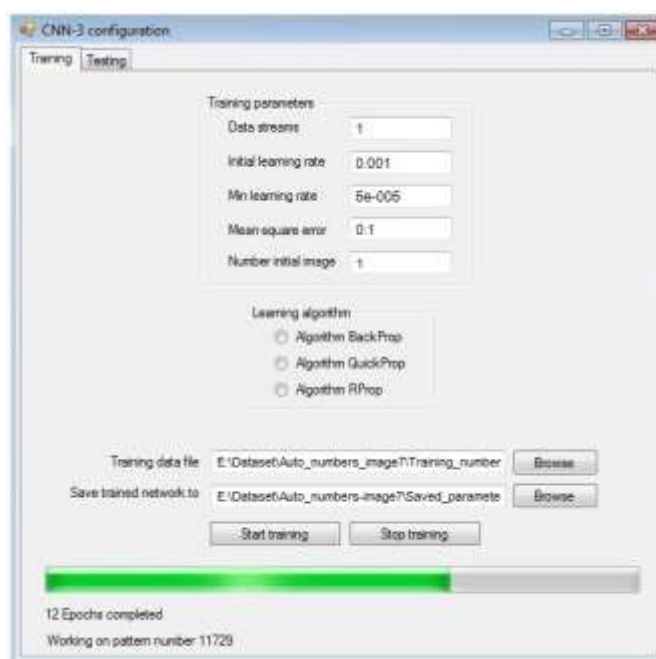


Рисунок 3.12 – Форма настройки параметров обучения СНС-3

Вкладка «*Testing*» (рисунок 3.13) содержит настраиваемые параметры «*Data streams*» и «*Number initial image*», поля ввода данных «*Testing data file*» и «*Save tested network to*», кнопки «*Start testing*» и «*Stop testing*».

По центру расположена панель, отображающая процесс тестирования: номер обрабатываемого образа, тестовое значение образа, реальное значение образа и результат их сравнения. В самом низу формы отображается номер обрабатываемого образа.

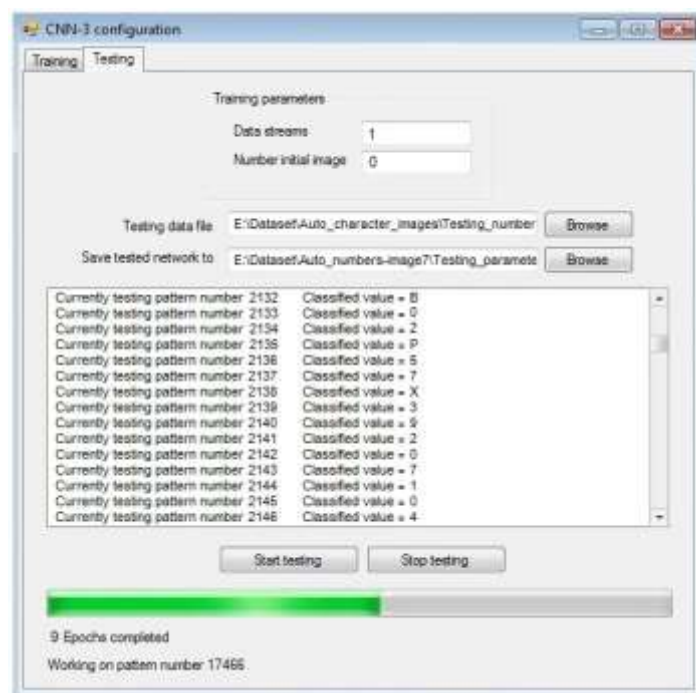


Рисунок 3.13 – Форма настройки параметров тестирования СНС-3

При выборе пункта меню «*Settings*»→*Common settings*» откроется форма для настройки общих параметров процесса распознавания (рисунок 3.14). Данная форма содержит настраиваемые параметры:

- «*Initial step of scanning*» – величина начального шага сканирования изображения нейронной сетью.
- «*Second step of scanning*» – величина повторного шага сканирования изображения нейронной сетью.
- «*1 detection threshold*» – порог обнаружения при предварительной классификации.
- «*2 detection threshold*» – порог обнаружения при итоговой классификации.
- «*Scaling factor*» – коэффициент масштабирования изображения.
- «*Number scales*» – количество масштабов изображения.
- «*Contrast enhancement*» – повышение контраста.
- «*Binarization*» – бинаризация.
- «*Maximum angle of histograms*» – максимальный угол построения гистограмм.

- «*Highlight color*» – цвет выделения автомобильного номерного знака.

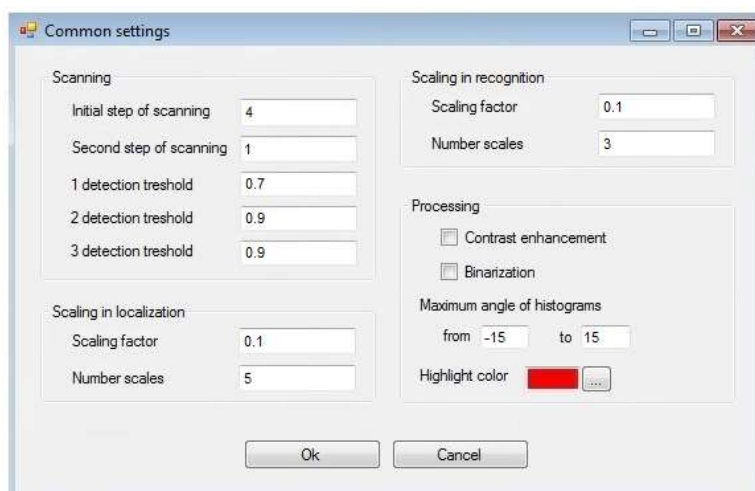


Рисунок 3.14 – Форма настройки общих параметров

На рисунках 3.15, 3.16, 3.17 изображено содержимое вкладки «*Automatic recognition*» (1), отображающей процесс распознавания в автоматическом режиме. При нажатии кнопки «*Open image*» (2) можно загрузить изображение и оно отобразится в центре интерфейса программы. Кнопка «*Recognition*» (4) предназначена для запуска процесса распознавания автомобильных номерных знаков. Если номерной знак найден, то он выделяется рамкой. В поле «*Recognition result*» (5) отображаются распознанные символы в текстовом виде. При нажатии на кнопку «*View license plate*» (6) выводится соответствующее изображение распознанного автомобильного номерного знака. При нажатии кнопки «*Save result*» (3) можно сохранить распознанный номер в базу данных.

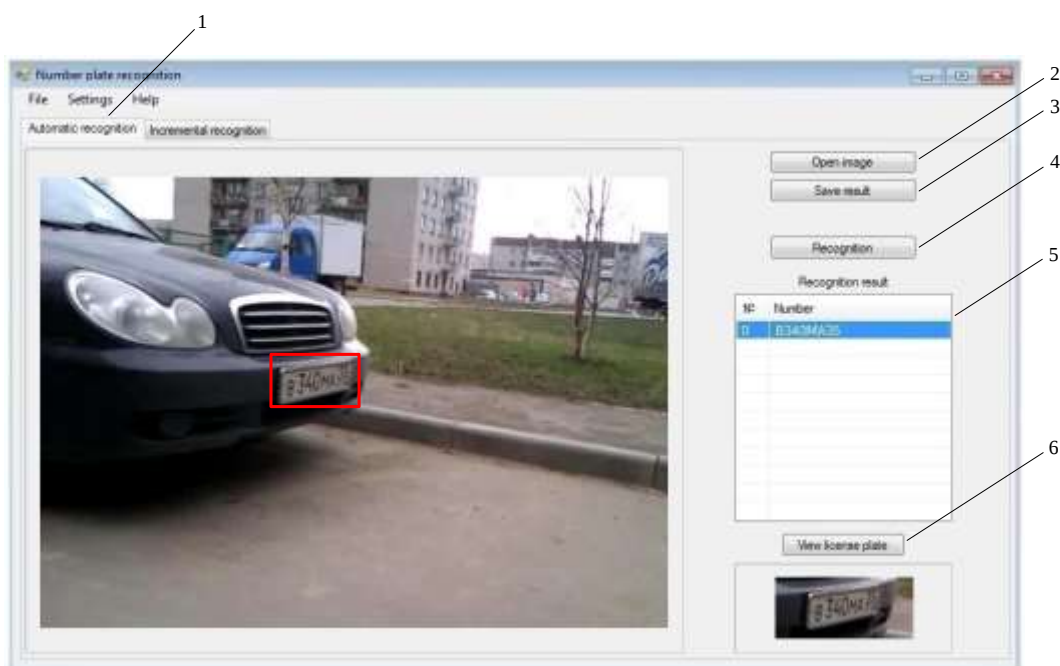


Рисунок 3.15 – Автоматическое распознавание автомобильного номерного знака

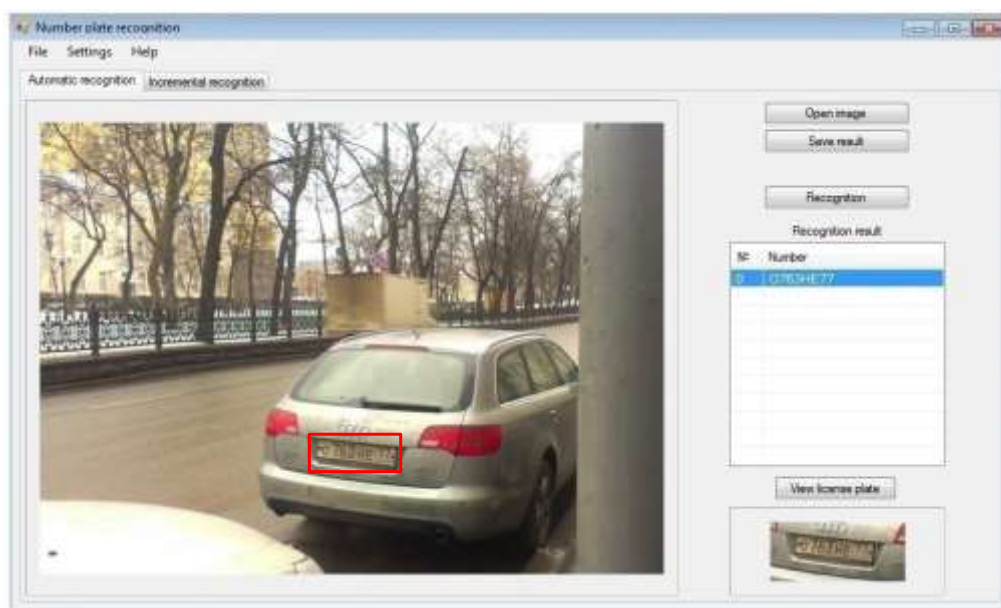


Рисунок 3.16 – Автоматическое распознавание автомобильного номерного знака

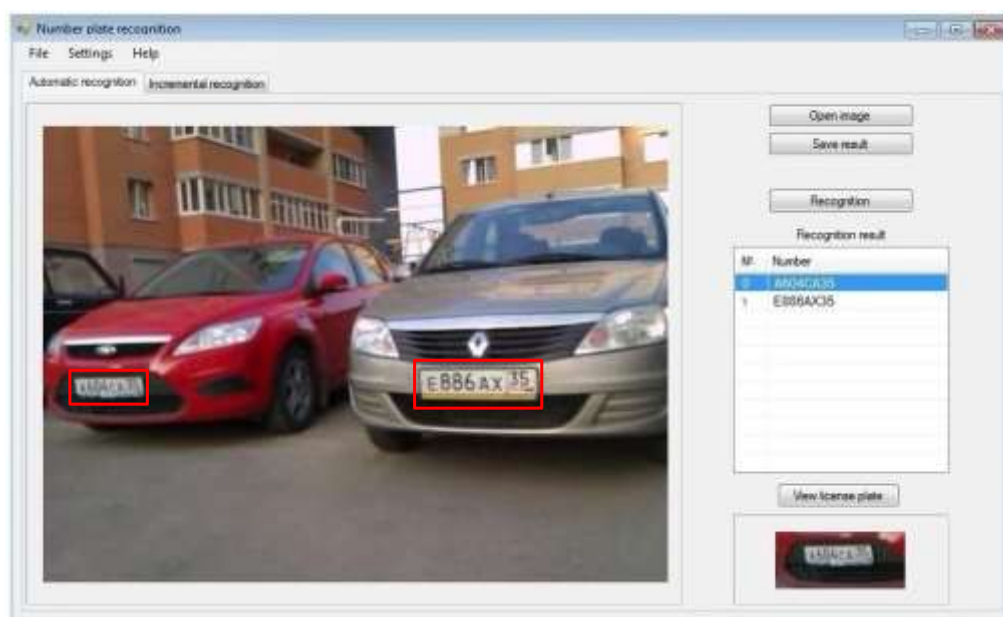


Рисунок 3.17 – Автоматическое распознавание автомобильного номерного знака

На рисунке 3.18 изображено содержимое вкладки «*Incremental recognition*» (1), отображающей процесс распознавания более подробно. При нажатии кнопки «*Open image*» (2) можно загрузить изображение и оно отобразится на панели интерфейса. Кнопка «*Save result*» (3) предназначена для сохранения результата распознавания в базу данных. При нажатии кнопки «*Detect license plate*» (4) осуществляется запуск процесса детектирования автомобильного номерного знака на загруженном изображении. Если номерной знак найден, то он выделяется рамкой. Кнопка «*Recognition*» (5) предназначена для запуска процесса распознавания. В поле «*Recognition result*» (6) отображаются распознанные символы в текстовом виде. При нажатии на кнопку «*View detect plate*» (7) выводится изображение соответствующего распознанного автомобильного номерного знака. Кнопка «*View normalize plate*» (8) предназначена для отображения нормализованного изображения автомобильного номерного знака и выделенных символов на нем.

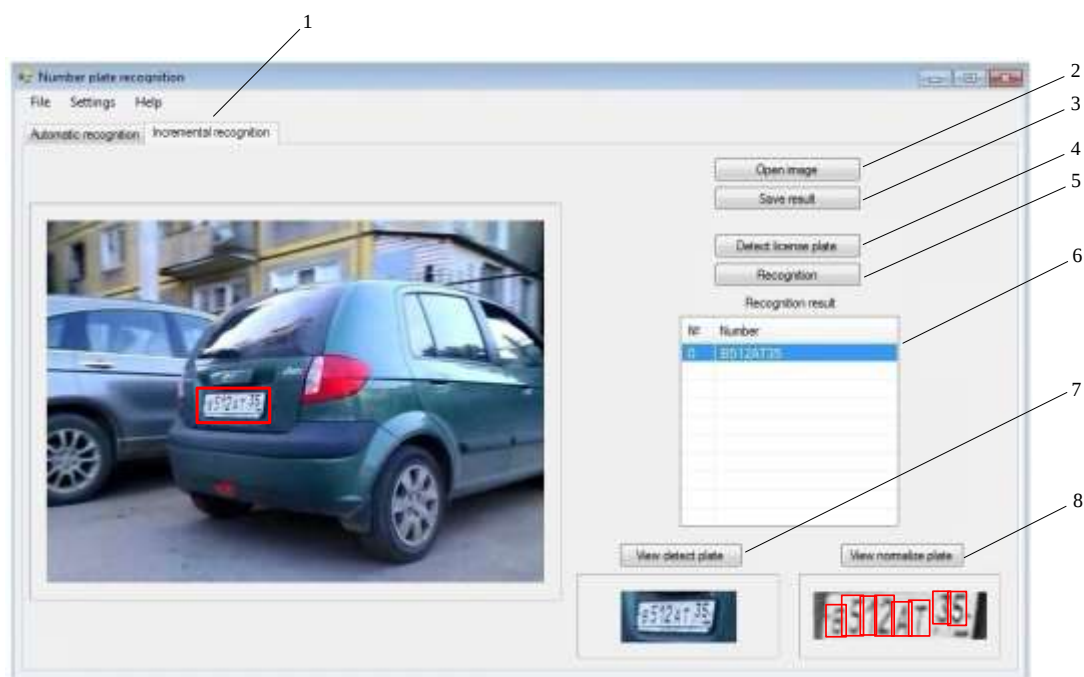


Рисунок 3.18 – Пошаговое распознавание автомобильного номерного знака

При выборе пункта меню «File→Open database» можно посмотреть базу данных распознанных автомобильных номерных знаков (рисунок 3.19).

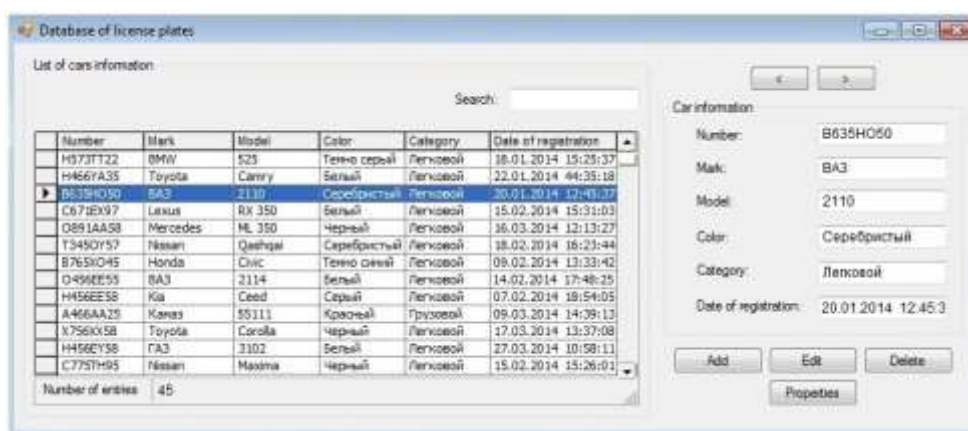


Рисунок 3.19 – База данных распознанных автомобильных номерных знаков

3.6 Основные результаты и выводы по главе 3

1. Сформулированы основные требования к библиотекам, реализующим предложенные алгоритмы. Представлен анализ распространенных библиотек по компьютерному зрению и обработке изображений. На основании этого осуществлен выбор средств для программной реализации алгоритмов.

2. Изложено описание основных библиотек классов, реализующих главные модули программной системы. Представлены логические схемы разработанных алгоритмов.

3. Разработана программная система на основе созданных библиотек. Представлен пользовательский интерфейс программной системы; описан процесс работы с программной системой, включающий этапы настройки, обучения и тестирования сверточных нейронных сетей.

ГЛАВА 4. ТЕСТИРОВАНИЕ РАЗРАБОТАННЫХ АЛГОРИТМОВ И ПРОГРАММНЫХ СРЕДСТВ

В данной главе представлены результаты обучения и тестирования разработанных алгоритмов на примере распознавания автомобильных номерных знаков на изображениях. Представлены результаты тестирования различных функциональных характеристик разработанной программной системы. Представлен сравнительный анализ результатов работы реализованной программной системы с существующими алгоритмами и технологическими системами, решающими задачу распознавания автомобильных номерных знаков.

4.1 Обучение и тестирование сверточных нейронных сетей

Создание набора данных для обучения и тестирования нейронной сети является важным этапом. Для качественного обучения и тестирования нейронных сетей, набор изображений должен обладать следующими характеристиками:

- Репрезентативность – набор изображений должен в полной мере соответствовать поставленной задаче и отображать определенные объекты и их характерные признаки;
- Непротиворечивость – изображения должны быть однородного характера и содержать объекты определенного класса. Изображения, содержащие противоречивые данные могут привести к плохому качеству обучения нейронной сети;
- Объем – набор изображений не должен быть слишком маленьким, т.к. в этом случае нейронная сеть не сможет выполнить полноценное обобщение [75, 82].

Существенное значение имеют количество и качество данных в обучающей выборке, а так же насколько полноценно эти данные описывают предметную область, все это напрямую влияет на качество обучения нейронной сети.

4.1.1 Обучение и тестирование сверточных нейронных сетей для детектирования области расположения символов

Для обучения СНС-1 и СНС-2 была сформирована база данных из 18000 изображений сегментированных автомобильных номерных знаков (размер 28x52 пикселей), соответствующих ГОСТ Р 50577 – 93, группе 1, типу 1.

Основная часть изображений была получена из различных источников: база изображений автомобильных номерных знаков ВМК МГУ [38], база изображений автомобильных номерных знаков «Database VLP number plates» [179], изображения сегментированных не зашумленных номерных знаков [80]. Остальная часть изображений была сформирована самостоятельно. На рисунке 4.1 представлены изображения автомобильных номерных знаков для обучения нейронных сетей.



Рисунок 4.1 – Изображения для обучения нейронных сетей

Сформированная база данных включает в себя: 8000 изображений, содержащих автомобильные номерные знаки, расположенные под различными углами относительно регистрирующего устройства (рисунок 4.2 (а)); 10000 изображений, содержащих автомобильные номерные знаки, расположенные фронтально относительно регистрирующего устройства (рисунок 4.2 (б)).

Изображения автомобильных номерных знаков получены при различных погодных условиях и имеют различную яркость и контраст.

Для обеспечения высокой степени инвариантности СНС к различным искажениям, к некоторым изображениям из обучающего набора применялось изменение освещения, контраста и размытие. На рисунке 4.2 представлены изображения автомобильных номерных знаков из обучающей выборки.

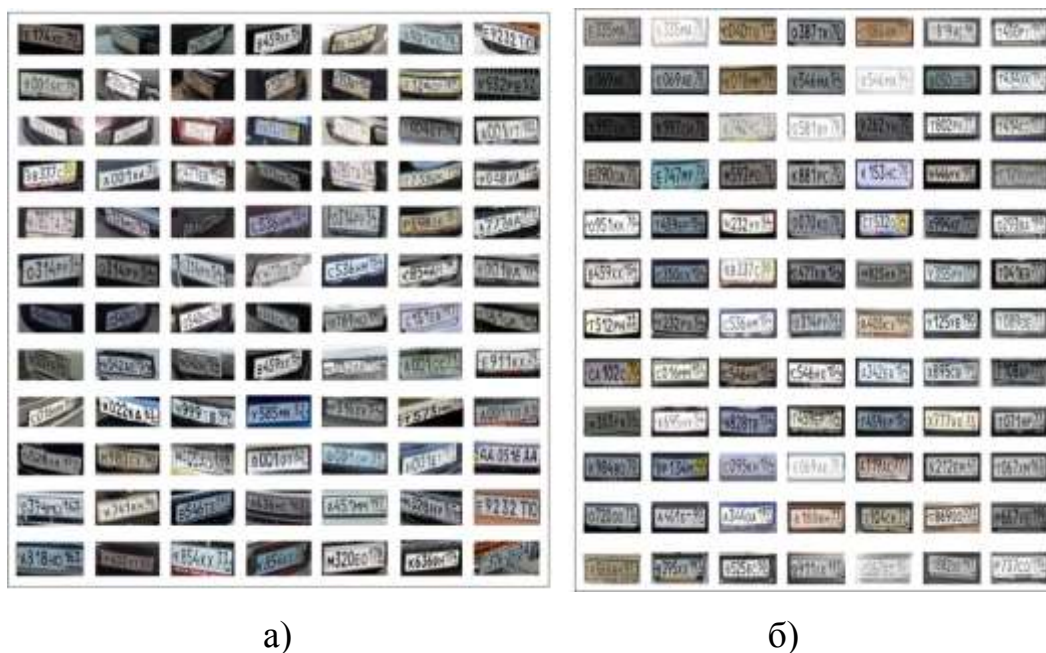


Рисунок 4.2 – Изображения автомобильных номерных знаков, расположенных фронтально относительно регистрирующего устройства

Для реализации обучения нейронных сетей также необходимо создание обучающего набора изображений, соответствующих классу «фон», т.е. не содержащих образы автомобильных номерных знаков. Для решения поставленной задачи был сформирован обучающий набор, состоящий из 20150 изображений, соответствующих фону (рисунок 4.3). В отдельном файле хранятся номера-метки, которые соответствуют каждому изображению и определяют их к различным классам.

Формирование набора фоновых изображений является достаточно сложной задачей, так как практически любое изображение, не содержащее автомобильный номерной знак, может быть отнесено к классу «фон» [38]. Чтобы обеспечить эффективный процесс обучения данный набор должен содержать множество

образов, потенциально являющихся источниками ложных обнаружений, т.е. имеющих визуальные признаки, схожие с автомобильными номерными знаками.

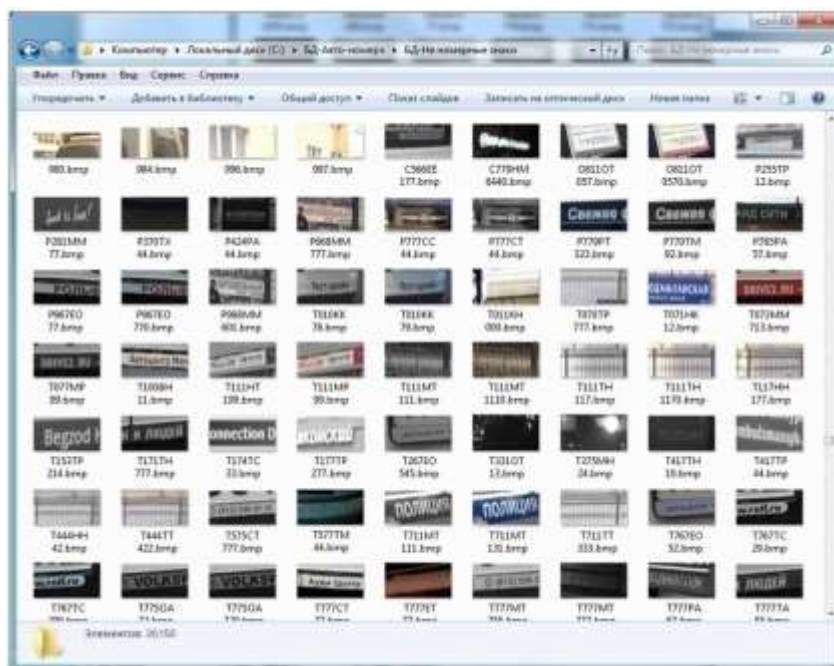


Рисунок 4.3 – Фоновые изображения, не содержащие образы автомобильных номерных знаков

Для решения данной задачи был реализован алгоритм, обеспечивающий эффективный процесс обучения СНС и создание качественного набора фоновых изображений, в полной мере охватывающего множество необходимых образов. Изначально был сформирован набор из 7500 фоновых изображений. Далее осуществлялось многократное переобучение нейронных сетей на данном наборе изображений, и последующее их тестирование. Выявленные в результате тестирования ложные обнаружения добавлялись в обучающий набор фоновых изображений для повторного обучения СНС. Данный алгоритм состоит из следующих шагов:

Шаг 1. Начальная инициализация СНС, синаптических связей и управляющих переменных: IterationNumber – количество итераций; FalseTreshold – величина порога ложных обнаружений (изначально равная 0.8).

Шаг 2. Обучение сверточных нейронных сетей на сформированном наборе изображений.

Шаг 3. Тестирование сверточных нейронных сетей на изображениях со сложной фоновой структурой. Использовались произвольные изображения пейзажа, полученные из различных источников.

Шаг 4. Выявление ложных обнаружений, в которых отклик нейронной сети превышает заданный порог (FalseTreshold). В результате этого, определяются объекты на изображениях, являющиеся источниками ложных обнаружений.

Шаг 5. Выявленные на шаге 4 области изображений масштабируются и добавляются в обучающий набор фоновых изображений.

Шаг 6. Уменьшение величины порога ложных обнаружений (FalseTreshold). Таким образом, требования к классифицирующим способностям СНС постепенно становятся более жесткими. Переход к шагу 2.

Данный процесс выполняется до тех пор, пока не осуществляется его сходимость, т.е. количество ложных обнаружений остается на одном уровне.

В результате выполнения данного алгоритма был сформирован набор из 20150 фоновых изображений, обеспечивающий эффективное обучение разработанных нейронных сетей (рисунок 4.3).

На рисунке 4.4. представлен график, отображающий изменение количества ложных обнаружений в результате выполнения данного алгоритма.

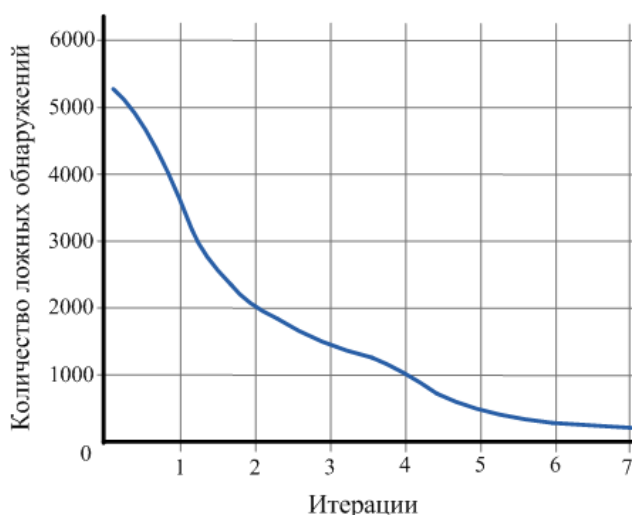


Рисунок 4.4 – Изменение количества ложных обнаружений

По рисунку 4.4. видно, что на первой итерации алгоритма количество ложных обнаружений достаточно велико (около 3500), а после третьей итерации

количество ложных обнаружений снижается до 1500. На шестой итерации наблюдается процесс сходимости и число ложных обнаружений достигает уровня 150.

На рисунке 4.5. изображен график, отображающий постепенное изменение средней величины откликов нейронных сетей, соответствующей разным классам объектов на изображениях из тестовой выборки.

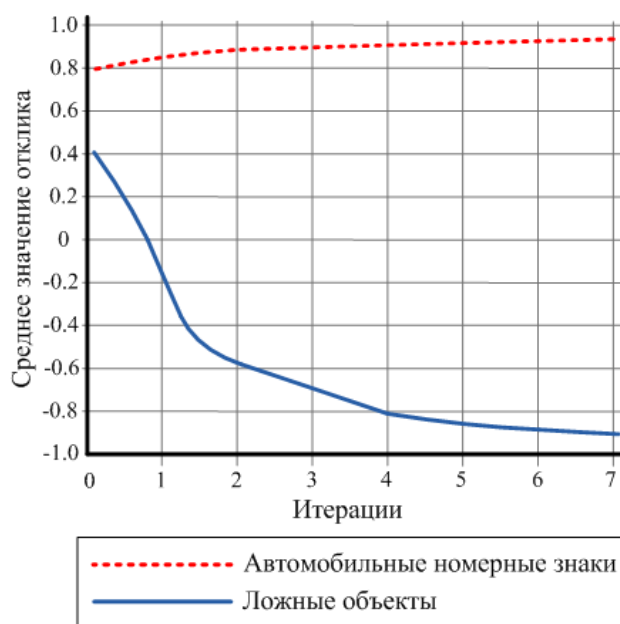


Рисунок 4.5 – Изменение среднего значения откликов нейронных сетей, соответствующих разным классам объектов на изображениях

По рисунку 4.5 видно, что уже на первой итерации среднее значение откликов СНС, соответствующих наличию автомобильного номерного знака на исследуемой области изображения, приближается к 0.9. Это говорит о том, что нейронные сети достаточно быстро запоминают характерные признаки автомобильных номерных знаков. Среднее значение откликов, соответствующих ложным объектам, после каждой итерации постепенно снижается. Таким образом, сверточные нейронные сети обучаются разделять образы на два класса и при этом осуществляется постепенное улучшение их классифицирующих способностей.

Разработанные СНС обучаются достаточно долго, это зависит от их сложной структуры, количества изображений в обучающей выборке и мощности используемого персонального компьютера (ПК). Однако стоит отметить, что

длительное время обучения сверточных нейронных сетей, мало интересует конечного пользователя.

При тестировании нейронных сетей необходимо использовать независимые данные, которые не входили в обучающую выборку, поэтому обучающая и тестовая выборки содержат разные наборы изображений. В процессе тестирования разработанных СНС не выполнялось какой-либо предобработки и нормализации изображений, так как сверточные нейронные сети обладают устойчивостью к искажениям входных данных.

Для тестирования СНС-1 и СНС-2 была сформирована база данных из 10000 изображений транспортных средств.

Основная часть изображений была получена из различных источников: база изображений автомобильных номерных знаков ВМК МГУ [38], база изображений транспортных средств «Database VLP images» [180], изображения транспортных средств [80]. Остальная часть изображений была сформирована самостоятельно. Размер изображений: 650x750 пикселей. Изображения получены при различных погодных условиях и различных углах регистрации с естественными шумовыми помехами. В отличие от обучающей выборки, все изображения из тестового набора содержат сложную фоновую структуру. На рисунке 4.6 представлены изображения из тестового набора.

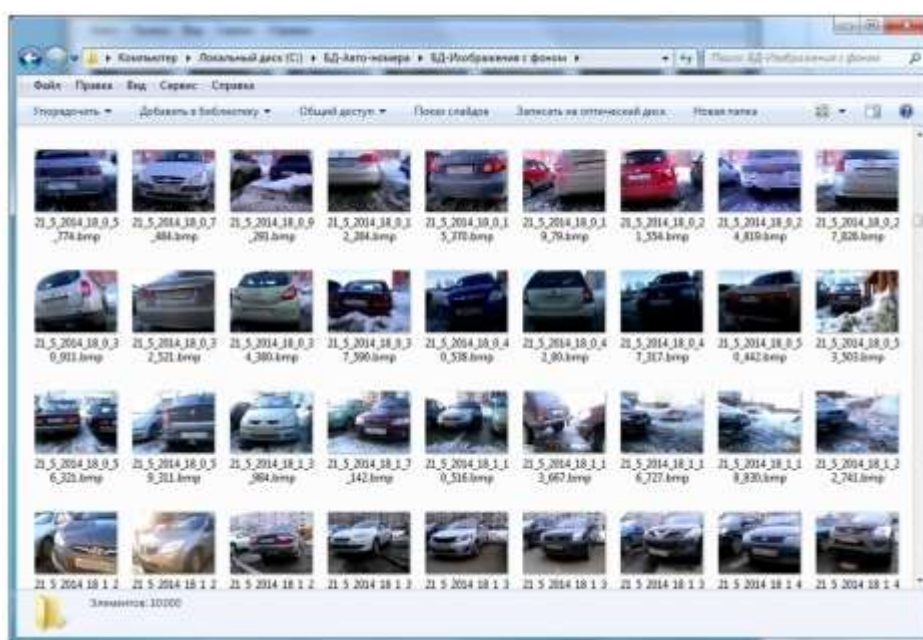


Рисунок 4.6 – Изображения для тестирования нейронной сети

На рисунке 4.7. представлен график, отображающий распределение откликов нейронных сетей, соответствующих разным классам объектов на изображениях из тестовой выборки. Отклики, соответствующие наличию автомобильного номерного знака в исследуемой области изображения, находятся в окрестности 1. Отклики, соответствующие фоновым и ложным объектам, находятся в окрестности -1 .

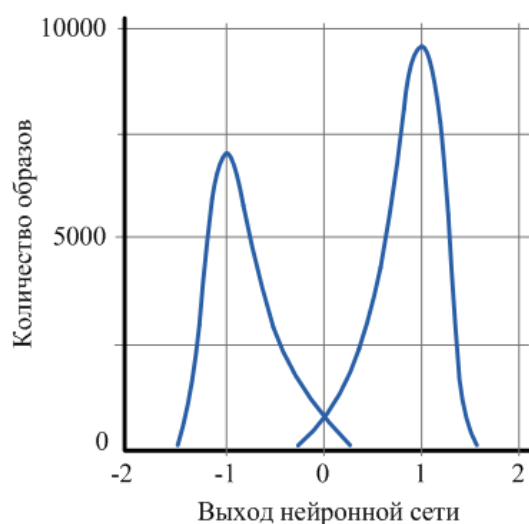


Рисунок 4.7 – Распределение откликов нейронных сетей, соответствующих разным классам объектов на изображениях

Результаты обучения и тестирования СНС-1 и СНС-2

Для сравнения экспериментальных результатов, обучение СНС-1 и СНС-2 производилось несколькими различными алгоритмами. В таблице 4.1 представлены результаты обучения и тестирования СНС-1 при использовании различных алгоритмов обучения.

Таблица 4.1 – Результаты обучения и тестирования СНС-1

Алгоритм обучения	Ошибка обучения	Количество эпох	Время обучения (часы)	Точность обнаружения при тестировании, (%)
Backpropagation	8.34E-02	34	12.3	82.16
Quick Propagation	3.34E-02	15	7.5	95.81
Resilient Propagation	1.56E-02	17	8.3	96.12

Генетический алгоритм	7.17E-02	19	13.6	94.47
Левенберга-Марквардта	5.83E-03	18	17.5	98.31

Как видно из таблицы 4.1 лучшее качество обучения было достигнуто при использовании алгоритма Левенберга-Марквардта (выделено темно-серым цветом):

- ошибка при обучении: 5.83E-03;
- точность обнаружения при тестировании: 98.31%.

Критерием остановки алгоритма обучения является не значительное изменение ошибки в течение 10 эпох. В качестве оценки точности обнаружения выбрано отношение количества правильно обнаруженных автомобильных номерных знаков к их общему количеству.

Количество пропущенных автомобильных номерных знаков на изображениях, т.е. классифицированных как «фон» (ошибки первого рода) составляет 0.3%. Как правило, пропущенные автомобильные номерные знаки являются сильно зашумленными, в связи с этим искажаются их характерные признаки и сверточная нейронная сеть классифицирует их не правильно.

Количество ложных обнаружений, т.е. посторонних объектов на изображениях, которые были классифицированы как автомобильный номерной знак (ошибки второго рода) составляет 1.41%.

Для обучения СНС-2 было принято решение расширить обучающий набор изображений с целью повышения классифицирующих способностей нейронной сети. Выявленные при тестировании СНС-1 ложные обнаружения, а так же некоторые визуально схожие с ними области изображений были добавлены в обучающий набор фоновых изображений, для обучения СНС-2. Так же некоторые зашумленные номерные знаки, классифицированные как «фон», были добавлены в обучающий набор изображений автомобильных номерных знаков. В итоге были получены 500 дополнительных изображений. Таким образом, СНС-2 обучалась на ошибках, полученных при тестировании СНС-1.

В таблице 4.2 представлены результаты обучения и тестирования СНС-2 при использовании различных алгоритмов обучения.

Таблица 4.2 – Результаты обучения и тестирования СНС-2

Алгоритм обучения	Ошибка обучения	Количество эпох	Время обучения (часы)	Точность обнаружения при тестировании, (%)
Backpropagation	7.95E-02	47	16.3	81.26
Quick Propagation	2.58E-02	17	8.2	96.87
Resilient Propagation	1.71E-02	19	8.9	97.04
Генетический алгоритм	6.76E-02	27	20.4	94.19
Левенберга-Марквардта	4.12E-03	23	18.3	98.43

Как видно из таблицы 4.2 лучшее качество обучения было достигнуто при использовании алгоритма Левенберга-Марквардта (выделено темно-серым цветом):

- ошибка при обучении: 4.12E-03;
- точность обнаружения при тестировании: 98.43%.

Количество пропущенных автомобильных номерных знаков на изображениях, т.е. классифицированных как «фон» (ошибки первого рода) составляет 0.28%.

Количество ложных обнаружений, т.е. посторонних объектов на изображениях, которые были классифицированы как автомобильный номерной знак (ошибки второго рода) составляет 1.3%.

На рисунке 4.8 представлены некоторые примеры ложных обнаружений, выявленные в процессе тестирования СНС-1 и СНС-2.



Рисунок 4.8 – Примеры ложных обнаружений, выявленных в процессе тестирования сверточных нейронных сетей

По рисунку 4.8 видно, что большинство не правильно классифицированных объектов имеют визуальные признаки схожие с автомобильными номерными знаками. В основном это некоторые виды дорожных знаков, вывески с названием улиц, рекламные вывески, балконы и т.д.

На рисунке 4.9 представлены некоторые примеры пропущенных автомобильных номерных знаков на изображениях, т.е. классифицированных как «не знак».

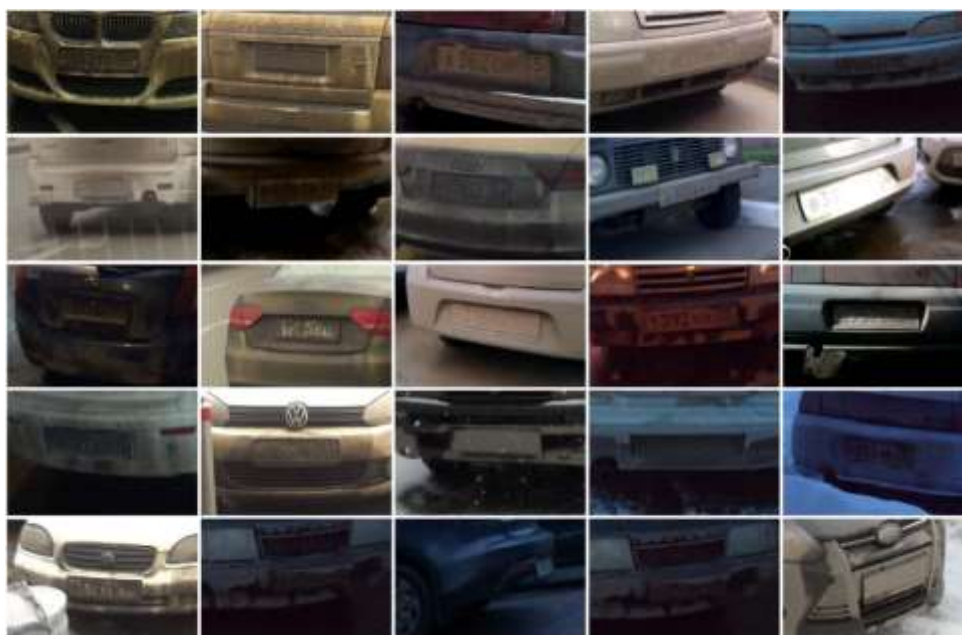


Рисунок 4.9 – Примеры изображений автомобильных номерных знаков, классифицированных как «не знак»

По рисунку 4.9 видно, что автомобильные номерные знаки, классифицированные как «не знак», являются сильно зашумленными и загрязненными. На основе полученных результатов можно сделать вывод, что разработанные сверточные нейронные сети (СНС-1 и СНС-2) обеспечивают высокую эффективность при решении задач детектирования автомобильных номерных знаков.

4.1.2 Обучение и тестирование сверточной нейронной сети для распознавания символов (СНС-3)

В процессе регистрации изображений образы автомобильных номерных знаков и символов на них могут быть подвержены различным аффинным и проекционным искажениям. Данные искажения могут возникать в связи с изменением углов регистрации и шумовых помех, при этом символы на изображениях могут частично менять свою структуру. Сверточные нейронные сети способны распознавать искаженные символы до того момента, пока угол отклонения не становится настолько велик, что символы будут наслаиваться друг на друга и структура символов будет неразличима.

На рисунке 4.10 представлен пример искажения символов при различных углах отклонения автомобильного номерного знака относительно регистрирующего устройства. При угле отклонения более чем на 70 градусов символы меняют структуру, сливаются друг с другом и не поддаются распознаванию. В реальных условиях такое случается крайне редко, в основном при маневрировании или в аварийных ситуациях, когда автомобиль разворачивает более чем на 70 градусов.

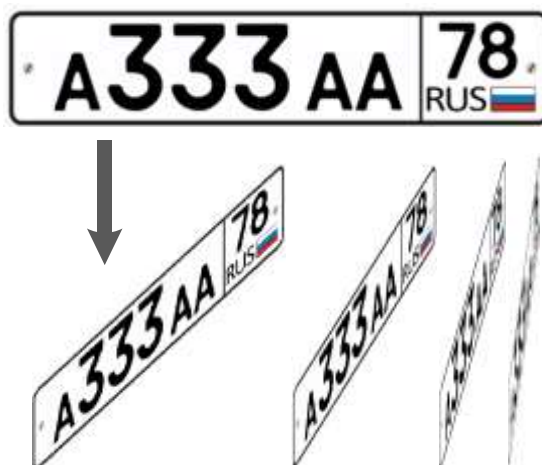


Рисунок 4.10 – Изображение автомобильного номерного знака при различных углах отклонения

В процессе создания обучающего и тестового наборов выполнялась задача обеспечения сверточной нейронной сети (СНС-3) набором изображений символов, представленных в различном стилистическом виде: символы под разными углами наклона, с искажениями, разрывами и шумовыми помехами.

Согласно ГОСТ Р50577 – 93 автомобильные номерные знаки соответствующие группе 1, типу 1, содержат следующие символы: А, В, Е, К, М, Н, О, Р, С, Т, Х, У и цифры от 0 до 9.

На основании этих требований для обучения и тестирования СНС-3 была сформирована база данных, состоящая из 30000 изображений символов, используемых на автомобильных номерных знаках. Основная часть изображений так же была получена из различных источников: база изображений автомобильных номерных знаков ВМК МГУ [38], база изображений транспортных средств «Database VLP characters» [178]. Остальная часть изображений была сформирована самостоятельно. База данных содержит изображения 21 класса символов размером 29x29 пикселей. Символы представлены в различном стилистическом виде, с искажениями, разрывами, размытием и в полной мере отображают разнообразие и богатство естественных данных. Так же база данных содержит 2122 изображения, которые соответствуют

фону. Это необходимо для того, что бы ложные обнаружения, которые могут быть выявлены на этапе детектирования, классифицировать как фон.

Обучающая и тестовая выборки содержат разные наборы изображений и хранятся в отдельных файлах. Размер обучающей выборки состоит из 20000 изображений, размер тестовой выборки состоит из 10000 изображений. Так же в отдельном файле хранятся номера-метки, которые соответствуют каждому изображению и определяют их к различным классам.

На рисунке 4.11 представлены примеры изображений цифр из обучающей и тестовой выборки.

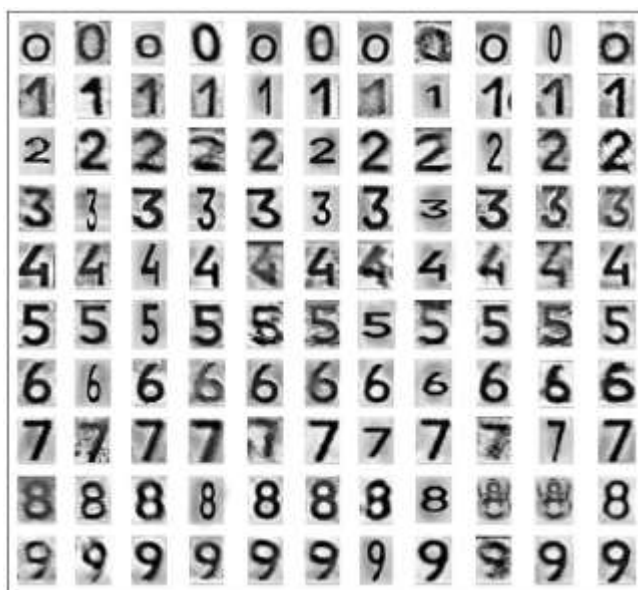


Рисунок 4.11 – Изображения цифр из обучающей и тестовой выборки

На рисунке 4.12 представлены примеры изображений букв из обучающей и тестовой выборки.



Рисунок 4.12 – Изображения букв из обучающей и тестовой выборки

Как видно по рисункам 4.11 и 4.12 символы представлены в различном стилистическом виде с различными искажениями.

На рисунке 4.13 представлены примеры изображений, соответствующих фону. Большая часть данных изображений была получена с помощью алгоритма, описанного в разделе 4.1.1.

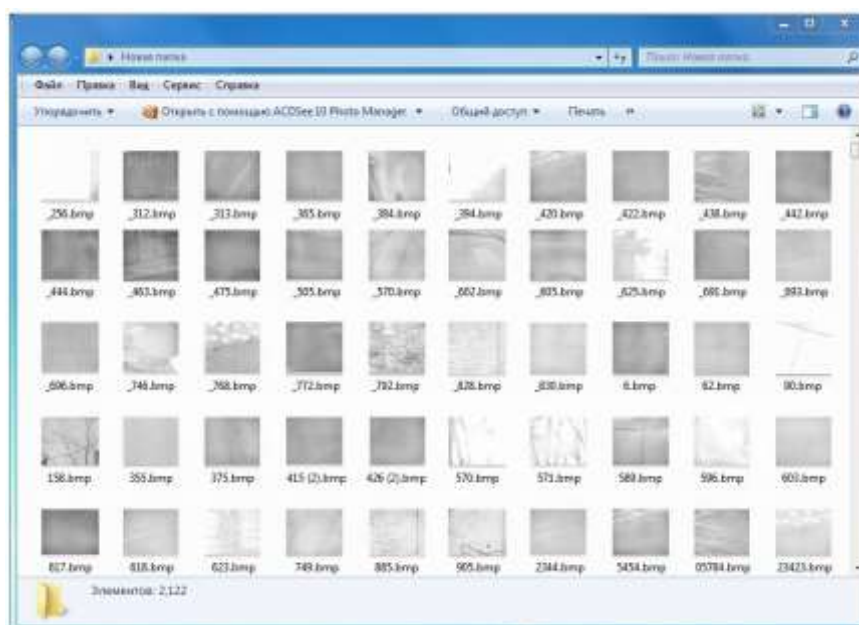


Рисунок 4.13 – Изображения, соответствующие фону

Результаты обучения и тестирования СНС-3

Для сравнения экспериментальных результатов, обучение и тестирование СНС-3 производилось несколькими различными алгоритмами. В таблице 4.4 представлены результаты обучения и тестирования СНС-3.

Таблица 4.4 – Результаты обучения и тестирования СНС-3

Алгоритм обучения	Ошибка обучения	Количество эпох	Время обучения (часы)	Точность при тестировании, (%)
Backpropagation	8.28E-02	42	18.1	96.21
Quick Propagation	4.84E-02	24	12.4	97.93
Resilent Propagation	1.24E-02	22	10.7	98.84

Как видно из таблицы 4.4 лучшее качество обучения было достигнуто при использовании алгоритма Resilent Propagation (выделено темно-серым цветом):

- ошибка при обучении: 1.24E-02;
- точность распознавания при тестировании: 98.84%.

В таблице 4.5 представлены значения точности распознавания для каждого класса символов из тестовой выборки и примеры ошибочных распознаваний. В качестве оценки точности распознавания для каждого класса символа выбрано отношение количества правильных распознаваний к общему количеству изображений символов, относящихся к данному классу.

Таблица 4.5 – Точность распознавания для каждого класса символов из тестовой выборки

Символ	Точность распознавания, (%)	Ошибочные распознавания символа
А	99.16	4, В, О
В	97.27	3, 8, А, Е
Е	98.32	3, В
К	98.11	Х
М	98.95	Н
Н	98.11	А, М

О	100	-
Р	100	-
С	99.37	О
Т	99.79	1
Х	98.95	К
У	100	-
1	98.74	7, Т
2	99.11	3, 8
3	97.69	8, 9, В
4	98.32	9, А
5	98.11	6, 8, О
6	98.53	8, 9
7	98.95	1, 8
8	98.32	3, 6, В
9	98.74	О, 3
Фон	100	-
Среднее значение	98.84	

В таблице 4.6 представлена более подробная информация, отображающая ошибочные распознавания и их количество для каждого класса символов. По столбцам таблицы отображаются ошибки первого рода, по строкам отображаются ошибки второго рода. Темно-серым цветом отмечено количество ошибочных распознаваний, имеющее значение больше нуля.

Таблица 4.6 – Количество ошибочных распознаваний, соответствующих каждому классу символов

	Количество ошибочных распознаваний																					
	А	В	Е	К	М	Н	О	Р	С	Т	Х	У	1	2	3	4	5	6	7	8	9	Ошибки 2 рода
А	-	1	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0	0	0	0	0	4
В	1	-	3	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	5	0	11
Е	0	6	-	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	8
К	0	0		-	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	8
М	0	0	0	0	-	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4
Н	4	0	0	0	3	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7
О	0	0	0	0	0	0	-	0	3	0	0	0	0	0	0	0	0	0	0	0	0	3
Р	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0	0
С	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	0	0	0	0

Т	0	0	0	0	0	0	0	0	0	-	0	0	1	0	0	0	0	0	0	0	1
Х	0	0	0	4	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0	4
У	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	-	0	0	0	0	5	0	0	6
2	0	0	0	0	0	0	0	0	0	0	0	0	-	3	0	0	0	0	2	0	5
3	0	4	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	5	2	11
4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	0	0	4	6
5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	-	5	0	3	9
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-	0	6	7
7	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	-	1	5
8	0	7	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	2	0	-	11
9	0	0	0	0	0	0	2	0	0	0	0	0	0	0	4	0	0	0	0	-	6
Ошибки 1 рода	7	18	3	4	3	4	4	0	3	1	8	0	5	0	13	2	0	7	5	22	7

Из данных, представленных в таблице 4.6 видно, что наибольшие значения ошибок первого рода соответствуют символам: В, Н, 3, 8. Наибольшие значения ошибок второго рода соответствуют символам: В, К, Н, 2, 3, 5, 8.

На рисунке 4.14 представлены примеры неправильно распознанных символов из тестового набора изображений. Под изображениями слева указан правильный ответ, справа – ответ нейронной сети. Как видно по рисунку некоторые изображения не правильно распознанных символов являются не однозначными по причине их сильной искаженности. Однако некоторые символы все же могут быть классифицированы человеком. Дальнейшее увеличение объема данных для обучения может привести к повышению точности работы системы.



Рисунок 4.14 – Изображения неправильно распознанных символов

Так же было проведено тестирование СНС-3 на изображениях автомобильных номерных знаков без использования процесса сегментации

отдельных символов. Для этого использовались 1500 изображений из базы данных сегментированных автомобильных номерных знаков, описанной в разделе 4.1.1. На каждом изображении автомобильного номерного знака содержится 8 – 9 символов. Таким образом, тестовая выборка изображений содержит 12750 символов. Для более эффективного проведения экспериментов была осуществлена предобработка изображений номерных знаков в виде их нормализации, выравнивания и приведения к единому масштабу.

В результате проведенных экспериментов точность распознавания символов на автомобильных номерных знаках составила 98.64%. Скорость распознавания всех символов на сегментированном автомобильном номерном знаке составила 26 мс. В качестве оценки точности распознавания символов используется отношение количества правильных распознаваний символов к общему числу символов на изображениях. В качестве оценки скорости работы алгоритмов представлена средняя скорость распознавания всех символов, изображенных на одном автомобильном номерном знаке совместно с этапом нормализации номерного знака.

На основе проведенных экспериментов можно сделать вывод, что реализованная СНС-3 обладает высокой эффективностью при решении задач распознавания символов на автомобильных номерных знаках.

4.2 Оценка качества работы программной системы

В данном разделе представлены результаты тестирования разработанной программной системы и отдельных алгоритмов, входящих в ее основу. Экспериментальная оценка качества работы программной системы осуществлялась в три этапа:

- 1) Оценка качества распознавания автомобильных номерных знаков при различном уровне освещения.
- 2) Оценка качества распознавания автомобильных номерных знаков при различных углах отклонения, относительно регистрирующего устройства.

3) Определение предельных значений углов отклонения номерных знаков относительно регистрирующего устройства, при которых точность распознавания символов остается на максимально высоком уровне.

4.2.1 Оценка качества распознавания автомобильных номерных знаков при различном уровне освещения

Для оценки данного параметра использовался набор из 400 изображений транспортных средств, расположенных фронтально относительно регистрирующего устройства и содержащих чистые автомобильные номерные знаки. Изображения получены при одинаковых условиях освещения и содержат сложную фоновую структуру. Размер изображений: 650x750 пикселей. Затем освещенность каждого изображения была преобразована в значения 50 лк и 1000 лк (с помощью ПО *ACDSee Photo Manager*) и для полученных изображений вычислялось значение средней интенсивности пикселей. Полученный диапазон изменения средней интенсивности пикселей делился на несколько частей и в соответствии с каждым значением интервала, генерировалось изображение. В результате была получена тестовая выборка, состоящая из 4400 изображений (рисунок 4.15).



Рисунок 4.15 – Изображения с различной интенсивностью освещения

В таблице 4.7 представлены результаты тестирования разработанных алгоритмов в зависимости от изменения интенсивности освещения на изображениях.

Таблица 4.7 – Результаты тестирования разработанных алгоритмов в зависимости от изменения интенсивности освещения на изображениях

Интенсивность освещения, (лк)	Точность предварительного детектирования (СНС-1), (%)	Точность итогового детектирования (СНС-2), (%)	Точность распознавания (СНС-3), (%)
50	95.46	97.79	98.03
145	95.69	98.52	98.28
240	96.15	98.76	98.28
335	96.38	98.76	98.52
430	96.61	99.25	98.52
525	96.61	99.25	98.52
620	96.61	99.25	98.52
715	96.61	99.25	98.28
810	96.15	98.76	98.5
905	95.92	98.52	98.5
1000	95.69	98.03	98.03
Среднее значение	96.15	98.74	98.36

Как видно по таблице 4.7 при снижении интенсивности освещения ниже 335 лк и увеличении выше 715 лк точность детектирования и распознавания не значительно снижается.

Количество пропущенных автомобильных номерных знаков на этапе детектирования, т.е. классифицированных как «не знак» (ошибки первого рода) составляет 0.12%. Количество ложных обнаружений, т.е. посторонних объектов на изображениях, которые были классифицированы как автомобильный номерной

знак (ошибки второго рода) составляет 0.43%. Время распознавания одного номерного знака составляет не более 64 мс.

При предварительном детектировании возможно выявление большого количества ложных обнаружений, поэтому точность детектирования на данном этапе не высокая. Это связано с тем, что на данном этапе шаг сканирования СНС-1 равен четырем пикселям, а порог обнаружения равен 0.7.

На этапе итогового детектирования осуществляется более тщательная обработка областей изображения, шаг сканирования СНС-2 равен одному пикселю, а порог обнаружения равен 0.9, из-за этого снижается количество ложных обнаружений и повышается точность детектирования.

При распознавании символов часть ложных обнаружений устраняется и классифицируется как фон. Таким образом, частично компенсируются ошибочные обнаружения, выявленные на этапе детектирования.

На основе полученных результатов можно сделать вывод, что разработанная программная система обладает высокой степенью инвариантности к изменению освещения в диапазоне от 50 лк до 1000 лк.

4.2.2 Оценка качества распознавания автомобильных номерных знаков при различных углах отклонения, относительно регистрирующего устройства

Для оценки данного параметра использовался набор из 8000 изображений транспортных средств. Изображения имеют различный размер (от 550x650 до 800x900 пикселей), получены при различных углах регистрации, погодных условиях и содержат сложную фоновую структуру (рисунок 4.16).

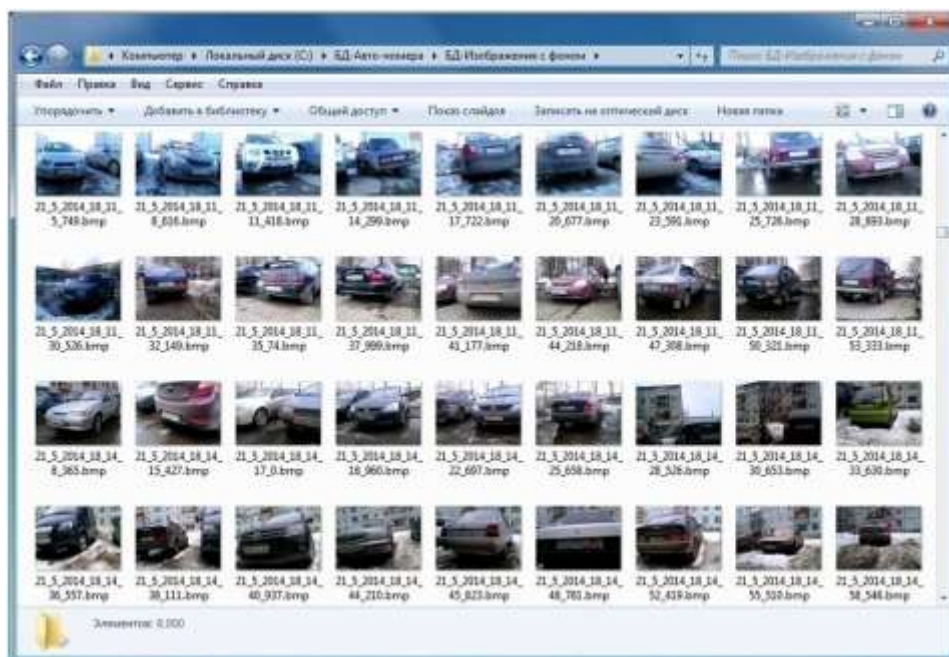


Рисунок 4.16 – Изображения автомобильных номерных знаков, полученные при различных углах регистрации

В таблице 4.8 представлены результаты тестирования разработанных алгоритмов при различных углах отклонения автомобильных номерных знаков, относительно регистрирующего устройства.

Таблица 4.8 – Результаты тестирования разработанных алгоритмов при различных углах отклонения автомобильных номерных знаков, относительно регистрирующего устройства

Точность предварительного детектирования (СНС-1), (%)	Точность итогового детектирования (СНС-2), (%)	Точность распознавания (СНС-3), (%)
95.24	98.52	98.2

Количество пропущенных автомобильных номерных знаков на этапе детектирования составляет 0.36%. Количество ложных обнаружений составляет 1.13%. Время распознавания одного номерного знака составляет не более 64 мс.

На рисунке 4.17 представлены некоторые примеры изображений неправильно распознанных автомобильных номерных знаков.



Рисунок 4.17 – Изображения неправильно распознанных автомобильных номерных знаков

По рисунку 4.18 видно, что символы на изображениях некоторых автомобильных номерных знаков сложно поддаются распознаванию даже человеком, по причине их излишней зашумленности и искаженности.

На основе полученных результатов можно сделать вывод, что разработанная программная система обеспечивает высокую эффективность при решении задач детектирования и распознавания автомобильных номерных знаков, расположенных под различными углами, относительно регистрирующего устройства.

4.2.3 Определение предельных значений углов отклонения номерных знаков, при которых точность распознавания символов остается на максимально высоком уровне

Визуально по изображениям невозможно определить точный угол отклонения автомобильных номерных знаков, относительно регистрирующего устройства. На основании этого было принято решение смоделировать пластины автомобильных номерных знаков, соответствующие ГОСТ Р 50577 – 93 и содержащие все символы, которые применяются на автомобильных номерных

знаках в различной их вариации (рисунок 4.18). В итоге было смоделировано 30 пластин номерных знаков.



Рисунок 4.18 – Смоделированные изображения пластин автомобильных номерных знаков

Далее с помощью программного обеспечения *Autodesk 3ds Max* производилось изменение углов отклонения смоделированных пластин по вертикали, горизонтали, на плоскости и полученные изображения подавались на вход программной системы. Таким образом, определялась точность распознавания символов при каждом угле отклонения пластины номерного знака.

На рисунке 4.19 показано, что при изменении угла отклонения пластины по горизонтали более ± 75 градусов символы визуально сливаются друг с другом и становятся не различимыми.

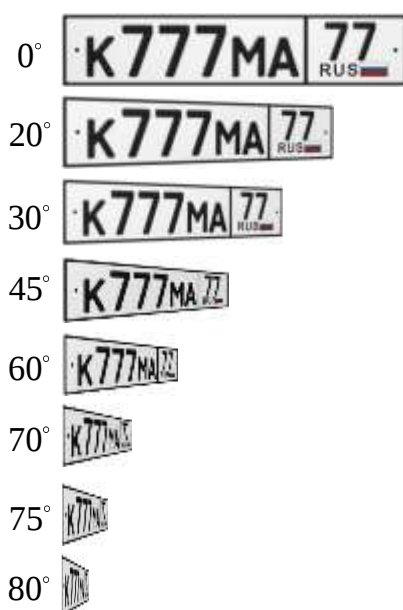


Рисунок 4.19 – Изменение угла отклонения автомобильного номерного знака

по горизонтали

На рисунке 4.20 показано, что при изменении угла отклонения пластины по вертикали более ± 75 градусов символы становятся неразличимыми.

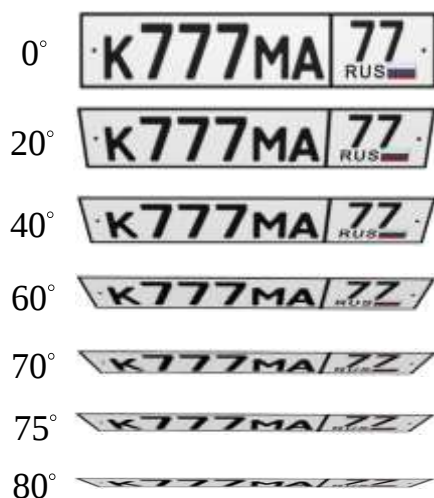


Рисунок 4.20 – Изменение угла отклонения автомобильного номерного знака по вертикали

На рисунке 4.21 изображено изменение угла отклонения автомобильного номерного знака на плоскости.

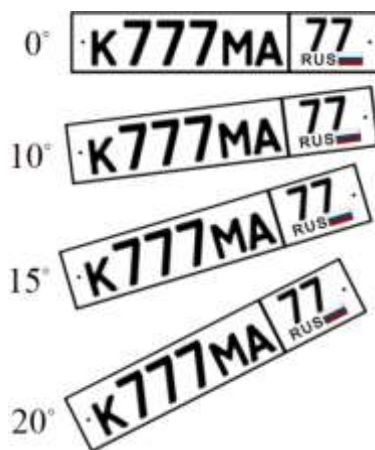


Рисунок 4.21 – Изменение угла отклонения автомобильного номерного знака на плоскости

В таблице 4.9 представлены значения точности распознавания символов на смоделированных автомобильных номерных знаках при изменении угла регистрации по горизонтали.

Таблица 4.9 – Точность распознавания символов на автомобильных номерных знаках при изменении угла регистрации по горизонтали

Угол отклонения пластины по горизонтали, (градусы)	Точность распознавания, (%)
75	78
70	86
65	92
60	97
55	100
50	100
45	100
40	100
35	100
30	100
25	100
20	100
15	100
10	100
5	100
0	100
-5	100
-10	100
-15	100
-20	100
-25	100
-30	100
-35	100
-40	100
-45	100
-50	100
-55	100
-60	97
-65	92
-70	86
-75	78

Из таблицы 4.9 видно, что точность распознавания символов сохраняется на уровне 100% при изменении угла регистрации пластины по горизонтали в диапазоне ± 55 градусов (выделено темно-серым цветом). Выходя за пределы этого диапазона углов, точность распознавания снижается. При углах отклонения, превышающих диапазон ± 75 градусов, символы не поддаются распознаванию.

В таблице 4.10 представлены значения точности распознавания символов на смоделированных автомобильных номерных знаках при изменении угла регистрации по вертикали.

Таблица 4.10 – Точность распознавания символов на автомобильных номерных знаках при изменении угла регистрации по вертикали

Угол отклонения пластины по вертикали, (градусы)	Точность распознавания, (%)
75	87
70	98
65	100
60	100
55	100
50	100
45	100
40	100
35	100
30	100
25	100
20	100
15	100
10	100
5	100
0	100
-5	100
-10	100
-15	100
-20	100
-25	100
-30	100
-35	100
-40	100
-45	100
-50	100
-55	100
-60	100
-65	100
-70	98
-75	87

Из таблицы 4.10 видно, что точность распознавания символов сохраняется на уровне 100% при изменении угла регистрации пластины по вертикали в

диапазоне ± 65 градусов (выделено темно-серым цветом). Выходя за пределы этого диапазона углов, точность распознавания снижается. При углах отклонения, превышающих диапазон ± 75 градусов, символы не поддаются распознаванию.

В таблице 4.11 представлены значения точности распознавания символов на смоделированных автомобильных номерных знаках при изменении угла регистрации на плоскости.

Таблица 4.11 – Точность распознавания символов на автомобильных номерных знаках при изменении угла регистрации на плоскости

Угол отклонения пластины на плоскости, (градусы)	Точность распознавания, (%)
20	100
15	100
10	100
5	100
0	100
-5	100
-10	100
-15	100
-20	100

Из таблицы 4.11 видно, что точность распознавания символов сохраняется на уровне 100% при изменении угла регистрации пластины на плоскости в диапазоне ± 20 градусов. Это обусловлено параметрами разработанных алгоритмов. В реальных условиях не возникают ситуации, когда угол отклонения пластины номерного знака на плоскости превышает диапазон ± 20 градусов, за исключением аварийных ситуаций.

После проведения экспериментов по изменению отдельных углов отклонения смоделированных автомобильных номерных знаков, углы изменялись попарно.

В таблице 4.12 представлены значения точности распознавания символов на смоделированных автомобильных номерных знаках при изменении углов регистрации по вертикали и горизонтали одновременно.

Таблица 4.12 – Точность распознавания символов на автомобильных номерных знаках при изменении углов регистрации по вертикали и горизонтали (значения углов показаны в градусах)

Горизонталь \ Вертикаль	55	45	40	30	20	10	0	-10	-20	-30	-40	-45	-55
65	41	53	66	89	100	100	100	100	100	89	66	53	41
60	51	63	86	100	100	100	100	100	100	100	86	63	51
55	64	83	91	100	100	100	100	100	100	100	91	83	64
50	78	88	100	100	100	100	100	100	100	100	100	88	78
45	86	100	100	100	100	100	100	100	100	100	100	100	86
40	92	100	100	100	100	100	100	100	100	100	100	100	92
35	100	100	100	100	100	100	100	100	100	100	100	100	100
30	100	100	100	100	100	100	100	100	100	100	100	100	100
25	100	100	100	100	100	100	100	100	100	100	100	100	100
20	100	100	100	100	100	100	100	100	100	100	100	100	100
15	100	100	100	100	100	100	100	100	100	100	100	100	100
10	100	100	100	100	100	100	100	100	100	100	100	100	100
5	100	100	100	100	100	100	100	100	100	100	100	100	100
0	100	100	100	100	100	100	100	100	100	100	100	100	100
-5	100	100	100	100	100	100	100	100	100	100	100	100	100
-10	100	100	100	100	100	100	100	100	100	100	100	100	100
-15	100	100	100	100	100	100	100	100	100	100	100	100	100
-20	100	100	100	100	100	100	100	100	100	100	100	100	100
-25	100	100	100	100	100	100	100	100	100	100	100	100	100
-30	100	100	100	100	100	100	100	100	100	100	100	100	100
-35	100	100	100	100	100	100	100	100	100	100	100	100	100
-40	92	100	100	100	100	100	100	100	100	100	100	100	92
-45	86	100	100	100	100	100	100	100	100	100	100	100	86
-50	78	88	100	100	100	100	100	100	100	100	100	88	78
-55	64	83	91	100	100	100	100	100	100	100	91	83	64
-60	51	63	86	100	100	100	100	100	100	100	86	63	51
-65	41	53	66	89	100	100	100	100	100	89	66	53	41

Из таблицы 4.12 видно, что при угле отклонения пластины по горизонтали до ± 55 градусов точность распознавания символов находится на уровне 100%, если при этом вертикальный угол отклонения не превышает диапазон ± 35 градусов (выделено темно-серым цветом).

При угле отклонения пластины по вертикали до ± 65 градусов точность распознавания символов находится на уровне 100%, если при этом

горизонтальный угол отклонения не превышает диапазон ± 20 . Выходя за пределы данных диапазонов, точность распознавания снижается (выделено темно-серым цветом).

В таблице 4.13 представлены значения точности распознавания символов на смоделированных автомобильных номерных знаках при изменении углов регистрации по вертикали и на плоскости.

Таблица 4.13 – Точность распознавания символов на автомобильных номерных знаках при изменении углов регистрации по вертикали и на плоскости (значения углов показаны в градусах)

На плоскости Вертикаль	20	15	10	5	0	-5	-10	-15	-20
65	100	100	100	100	100	100	100	100	100
50	100	100	100	100	100	100	100	100	100
40	100	100	100	100	100	100	100	100	100
30	100	100	100	100	100	100	100	100	100
20	100	100	100	100	100	100	100	100	100
10	100	100	100	100	100	100	100	100	100
0	100	100	100	100	100	100	100	100	100
-10	100	100	100	100	100	100	100	100	100
-20	100	100	100	100	100	100	100	100	100
-30	100	100	100	100	100	100	100	100	100
-40	100	100	100	100	100	100	100	100	100
-50	100	100	100	100	100	100	100	100	100
-65	100	100	100	100	100	100	100	100	100

Из таблицы 4.13 видно, что при угле отклонения пластины на плоскости до ± 20 градусов точность распознавания символов находится на уровне 100%, если при этом вертикальный угол отклонения не превышает диапазон ± 65 градусов.

При изменении углов регистрации по горизонтали и на плоскости значения точности распознавания так же стабильно находится на уровне 100% при угле отклонения пластины на плоскости до ± 20 градусов, если при этом вертикальный угол отклонения не превышает диапазон ± 55 градусов.

В результате тестирования программной системы были получены следующие результаты:

- точность распознавания автомобильных номерных знаков находится на уровне 98,2%;

- время распознавания составляет не более 64 мс;

- минимальная высота символов на изображении составляет 10 пикселей.

Условия регистрации автомобильных номерных знаков, при которых показатели точности распознавания сохраняются на максимально высоком уровне:

- освещенность в диапазоне от 50 лк до 1000 лк;

- максимальный угол отклонения пластины номерного знака по горизонтали до ± 55 градусов;

- максимальный угол отклонения пластины номерного знака по вертикали до ± 65 градусов;

- максимальный угол отклонения пластины номерного знака на плоскости до ± 20 градусов.

В таблице 4.14 представлены технические характеристики персонального компьютера (ПК), который использовался для вычислительных экспериментов.

Таблица 4.14 – Технические характеристики ПК, использованного для вычислительных экспериментов

Название параметра	Характеристика
Оборудование	
Модель	ASUS X200MA
Программное обеспечение (ПО)	
Операционная система (ОС)	Microsoft Windows 8
Тип системы	64-разрядная операционная система, процессор x64
Центральный процессор (ЦП)	
Центральный процессор	Intel Pentium Quad Core
Индекс	N3540
Частота процессора	2.16 ГГц
Оперативная память (ОЗУ)	
Тип оперативной памяти	DDR3
Объем оперативной памяти	2048 МБ

Стандарт памяти	PC3-10600
Частота памяти	1333 МГц
Видеосистема	
Графический процессор	Intel GMA HD
Объем памяти	512 Мб

4.3 Сравнение технических характеристик разработанных алгоритмов и программных средств с существующими аналогами

В данном разделе представлено сравнение результатов тестирования разработанных алгоритмов с другими алгоритмами, применяемыми для решения подобных задач. Так же представлено сравнение технических характеристик разработанной программной системы с существующими коммерческими программными системами, которые применяются для распознавания автомобильных номерных знаков.

Сравнение результатов тестирования алгоритма детектирования автомобильных номерных знаков с существующими аналогами

Результаты работы предложенного алгоритма детектирования автомобильных номерных знаков сопоставлены с результатами работы следующих алгоритмов:

- ED (Edge detection) + HT (Hough transform): алгоритм, основанный на выделении границ и преобразовании Хафа.
- ED + BHA (Brightness histograms analysis) + TM (Template method): алгоритм, основанный на выделении границ, гистограммном анализе изображения и методе сравнения с шаблоном.
- ED + BHA + SIFT (Scale Invariant Feature Transform): алгоритм, основанный на выделении границ, гистограммном анализе изображения и SIFT дескрипторе.

- ED + BHA + HOG (Histogram of Oriented Gradients): алгоритм, основанный на выделении границ, гистограммном анализе изображения и HOG дескрипторе.

- ED + BHA + SURF (Speeded Up Robust Features): алгоритм, основанный на выделении границ, гистограммном анализе изображения и SURF дескрипторе.

- VJA (Viola Jones algorithm): алгоритм Виолы-Джонса.

Для сравнения эффективности работы различных алгоритмов, их обучение и тестирование осуществлялось на одних и тех же данных. На рисунке 4.22 представлено сравнение результатов работы данных алгоритмов с предложенным алгоритмом детектирования.

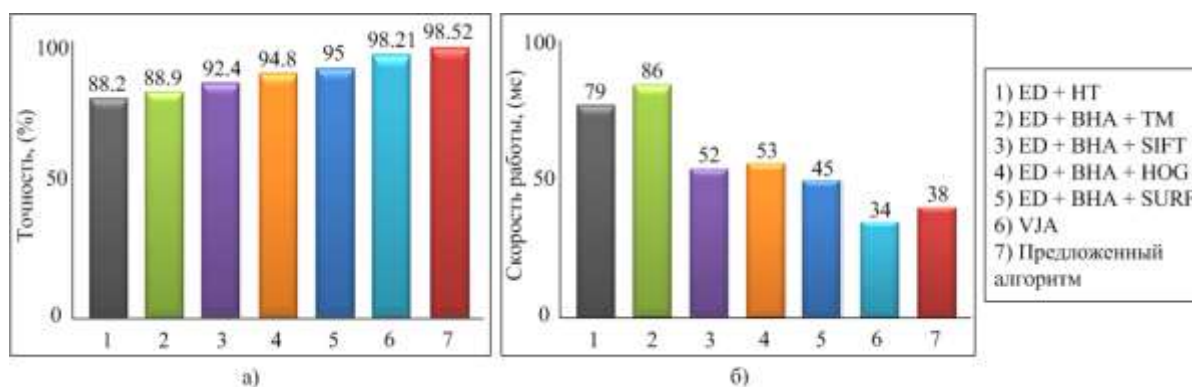


Рисунок 4.22 – Сравнение результатов работы различных алгоритмов при решении задачи детектирования автомобильных номерных знаков: а) точность классификации; б) скорость работы алгоритма

Как видно по рисунку 4.22 лучшие результаты точности классификации были получены с помощью предложенного алгоритма, на уровне 98.52%. Так же высокие результаты по данному показателю были получены с помощью алгоритма VJA (98.21%), практически на уровне с предложенным алгоритмом. Худшие результаты показали алгоритмы ED + HT, ED + BHA + TM.

По показателю «скорость работы алгоритма» лучшие результаты были получены с помощью алгоритма Виолы-Джонса, на уровне 34 мс. Предложенный алгоритм показал время 38 мс, что является вторым результатом среди сравниваемых алгоритмов. Худшие результаты показали алгоритмы ED + HT, ED + BHA + TM.

Сравнение результатов тестирования алгоритма распознавания символов с существующими аналогами

Результаты работы предложенного алгоритма распознавания символов на автомобильных номерных знаках были сравнены со следующими алгоритмами:

- BN (Binarization) + BHA + CCA (Connected components analysis) + TM (Template method): алгоритм, основанный на бинаризации, гистограммном анализе изображения, методе связных областей и методе сравнения с шаблоном.
- ED + BHA + CCA + KNN (K-nearest neighbours): алгоритм, основанный на выделении границ, гистограммном анализе изображения, методе связных областей и методе K-ближайших соседей.
- ED + BHA + CCA + MA (Morphological analysis) [7]: алгоритм, основанный на выделении границ, гистограммном анализе изображения, методе связных областей и морфологическом анализе.
- ED + CCA + HTM (Hierarchical temporal memory) [82]: алгоритм, основанный на выделении границ, методе связных областей и модели иерархической временной памяти.
- BN + BHA + PCA (Principal component analysis) + NN (Neural network) [71]: алгоритм, основанный на выделении границ, гистограммном анализе изображения, методе главных компонент и нейронной сети.
- BN + BHA + CCA + СНС: алгоритм, основанный на бинаризации, гистограммном анализе, методе связных областей и сверточной нейронной сети.

Данные алгоритмы включают операции нормализации изображения, выделения и распознавания символов. Для сравнения эффективности работы различных алгоритмов, их обучение и тестирование осуществлялось на одних и тех же данных (изображения сегментированных автомобильных номерных знаков). На рисунке 4.23 представлено сравнение результатов работы данных алгоритмов с предложенным алгоритмом распознавания символов. В качестве оценки точности распознавания символов используется отношение количества правильных распознаваний символов к общему числу символов на изображениях.

В качестве оценки скорости работы алгоритмов представлена средняя скорость распознавания всех символов, изображенных на одном автомобильном номерном знаке совместно с этапом нормализации номерного знака.

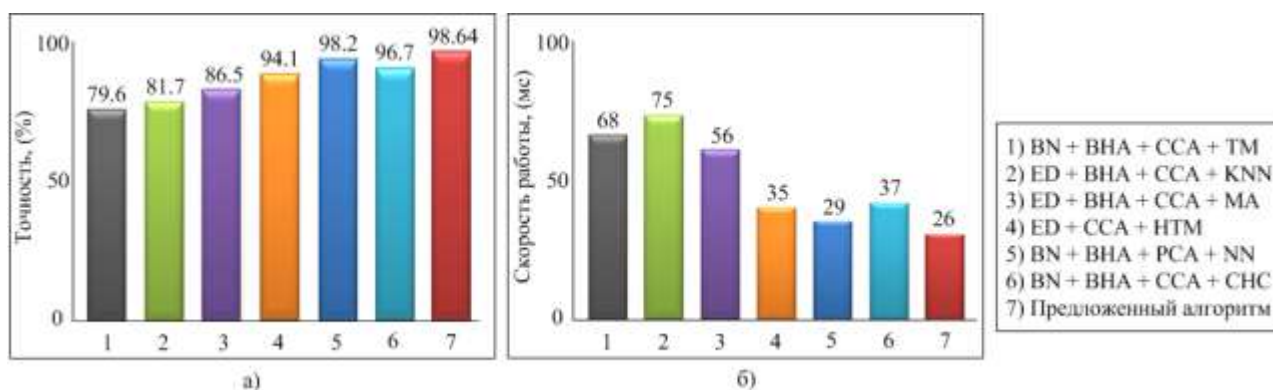


Рисунок 4.23 – Сравнение результатов работы различных алгоритмов при решении задачи распознавания символов на автомобильных номерных знаках: а) точность распознавания; б) скорость работы алгоритма

Из данных, представленных на рисунке 4.23 следует, что лучшие результаты точности распознавания на уровне 98.64% были получены с помощью предложенного алгоритма. Алгоритмы ED + CCA + HTM и BN + BHA + CCA + CHC так же показали хороший результат точности распознавания, на уровне 98.2% и 96.7% соответственно. Худшие результаты показали алгоритмы BN + BHA + CCA + TM и ED + BHA + CCA + KNN.

По показателю «скорость работы алгоритма» лучшие результаты были получены с помощью предложенного алгоритма, на уровне 26 мс. Алгоритмы ED + CCA + HTM и ED + BHA + PCA + NN показали результат на уровне 35 и 29 мс соответственно. Худшие результаты показали алгоритмы BN + BHA + CCA + TM и ED + BHA + CCA + KNN.

Сравнение технических характеристик разработанной программной системы с существующими аналогами

В таблице 4.15 представлено сравнение технических характеристик разработанной программной системы с существующими коммерческими программными системами, которые применяются для решения подобных задач.

Серым цветом выделены лучшие показатели среди аналогов. Темно-серым цветом выделены показатели, соответствующие разработанной программной системе.

Таблица 4.15 – Сравнение технических характеристик систем распознавания автомобильных номерных знаков

Название системы	Точность распознавания, (%)	Минимальная высота символов, (пиксели)
«Авто-Инспектор»	95	10
«Авто-Интеллект»	90	не указано
«Авто-Номер»	95	14
«Авто-Контроль»	95	14
«Senesys-Avto»	98	14
«Орион-Авто»	95	не указано
«SL-Traffic»	90	25
«Дигнум-Авто»	90	не указано
«CarFlow II»	93-95	не указано
«Интегра-С»	90	не указано
«Auto-TRASSIR»	90	не указано
«SecurOS Auto»	96	не указано
«Milestone XProtect»	не указано	20
«AutoCode»	96	не указано
«Macroscop»	не указано	9
«ТелеВизард-Авто»	90	100
«CVS Авто»	95	не указано
«MegaCar»	95	не указано
«Авто-Ураган»	97	не указано
Разработанное ПО	98.2	10

Из данных, представленных в таблице 4.15 видно, что одни из лучших показателей точности распознавания показывают системы «Senesys-Avto» и «Авто-Ураган», на уровне 98% и 97% соответственно (выделено серым цветом). Системы «SecurOS Auto», «AutoCode» показывают точность распознавания на уровне 96%, а остальные системы показывают точность на уровне 90-95%.

Разработанная программная система показывает точность распознавания на уровне 98.2%, что является наилучшим результатом среди сравниваемых систем (выделено темно-серым цветом).

По параметру «минимальная высота символов на изображении» лучшие результаты показывают системы «Авто-Инспектор» и «Macroscop», на уровне 9 и 10 пикселей соответственно. Разработанная программная система так же показывает один из лучших результатов, равный 10 пикселям.

На рисунке 4.24 показатели точности распознавания отображены в виде диаграммы.

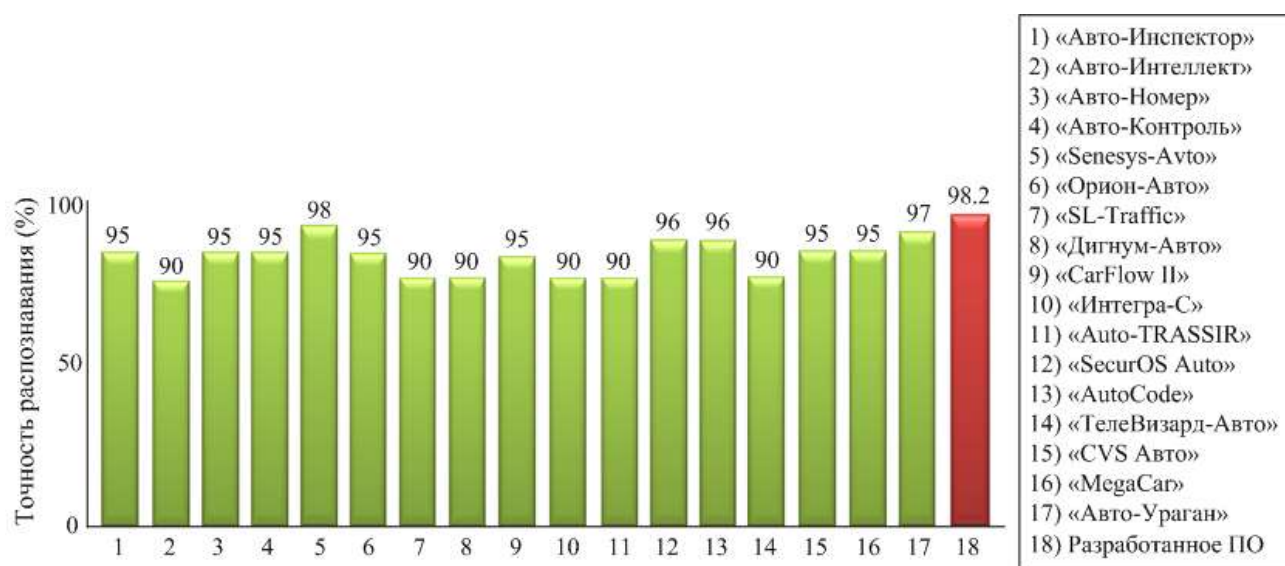


Рисунок 4.24 – Сравнение систем распознавания автомобильных номерных знаков по параметру «точность распознавания»

В таблице 4.16 представлены условия регистрации автомобильных номерных знаков, при которых сравниваемые системы показывают высокие показатели точности распознавания. Серым цветом выделены лучшие показатели среди аналогов. Темно-серым цветом выделены показатели, соответствующие разработанной программной системе.

Таблица 4.16 – Условия регистрации автомобильных номерных знаков, при которых сравниваемые системы показывают высокие показатели точности распознавания

	Вертикальный	Горизонтальный	Угол отклонения	Освещенность,
--	--------------	----------------	-----------------	---------------

	угол отклонения, (градусы)	угол отклонения, (градусы)	на плоскости, (градусы)	(лк)
«Авто- Инспектор»	± 40	± 30	± 15	от 50 до 1000
«Авто- Интеллект»	± 30	± 20	± 10	от 50 до 1000
«Авто-Номер»	± 30	± 30	не указано	от 50 до 1000
«Авто-Контроль»	± 30	± 30	не указано	от 50 до 1000
«Senesys-Avto»	± 30	± 30	не указано	от 50 до 1000
«Орион-Авто»	± 20	± 20	± 10	от 50 до 1000
«SL-Traffic»	± 60	± 30	± 10	от 50 до 1000
«Дигнум-Авто»	± 40	± 40	не указано	от 50 до 1000
«CarFlow II»	± 45	± 45	± 15	от 50 до 1000
«Интегра-С»	± 30	± 30	± 15	от 50 до 1000
«Auto-TRASSIR»	не указано	не указано	не указано	от 50 до 1000
«SecurOS Auto»	± 40	± 30	не указано	от 50 до 1000
«Milestone XProtect»	± 40	± 30	± 10	от 50 до 1000
«AutoCode»	± 40	± 30	± 10	от 50 до 1000
«Macroscop»	± 40	± 30	± 10	не указано
«ТелеВизард- Авто»	не указано	не указано	± 5	от 50 до 1000
«CVS Авто»	± 45	± 45	± 10	от 50 до 1000
«MegaCar»	± 30	± 30	± 10	от 50 до 1000
«Авто-Ураган»	± 30	± 20	± 10	от 50 до 1000
Разработанное ПО	± 65	± 55	± 20	от 50 до 1000

Из таблицы 4.16 видно, что разработанная программная система обеспечивает высокую точность распознавания автомобильных номерных знаков при следующих углах отклонения: по вертикали ± 65 градусов, по горизонтали ± 55 градусов, на плоскости ± 20 . Данные результаты являются наилучшими среди сравниваемых систем (выделено темно-серым цветом). Допустимый уровень освещенности при распознавании для всех систем одинаковый и находится в диапазоне от 50 лк до 1000 лк.

На рисунках 4.25, 4.26, 4.27 данные показатели отображены в виде диаграмм.

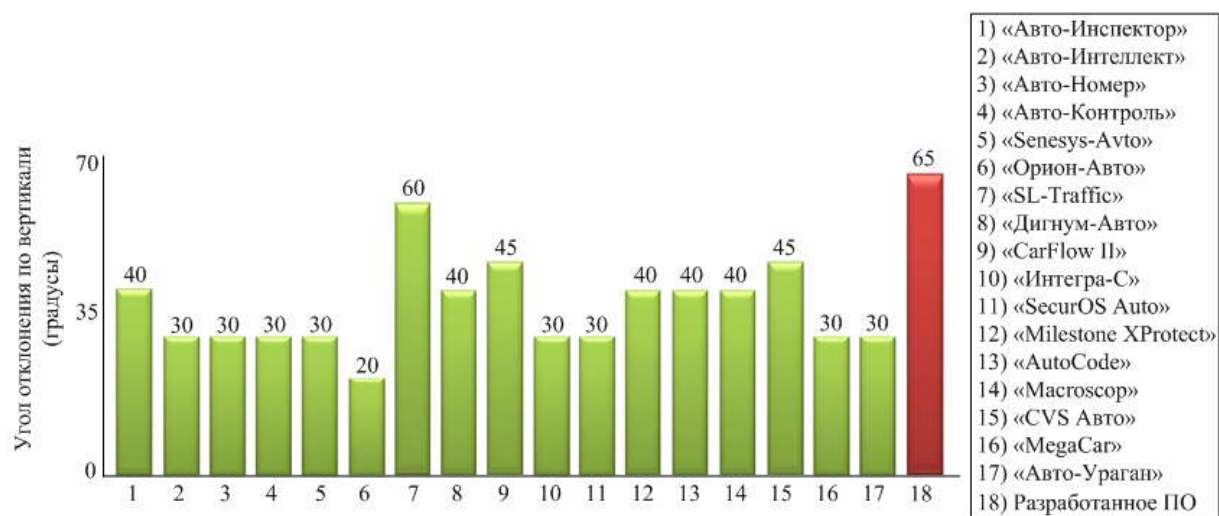


Рисунок 4.25 – Максимальные углы отклонения автомобильных номерных знаков по вертикали

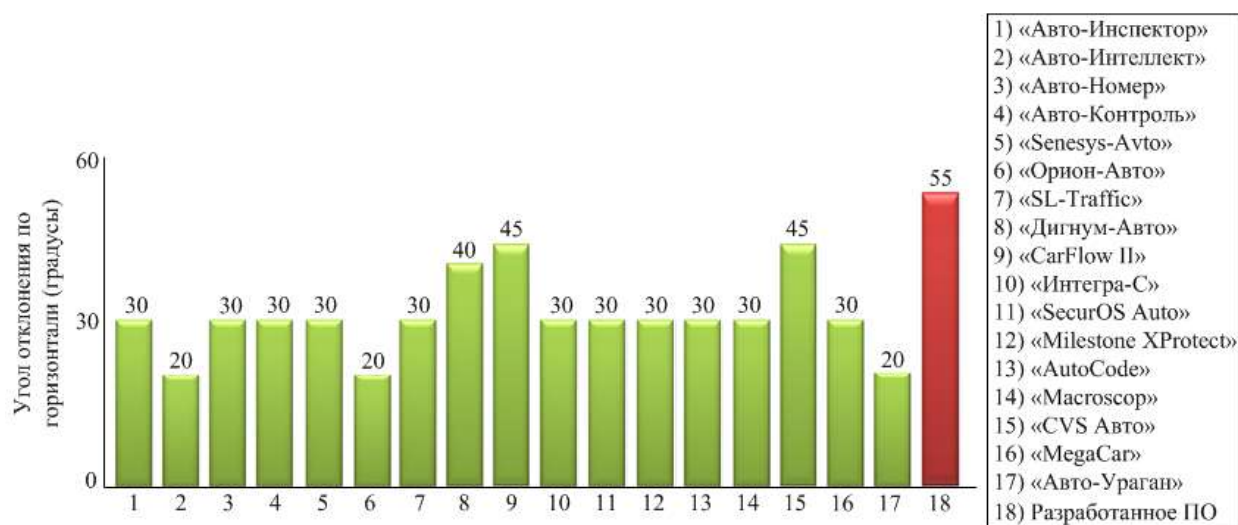


Рисунок 4.26 – Максимальные углы отклонения автомобильных номерных знаков по горизонтали

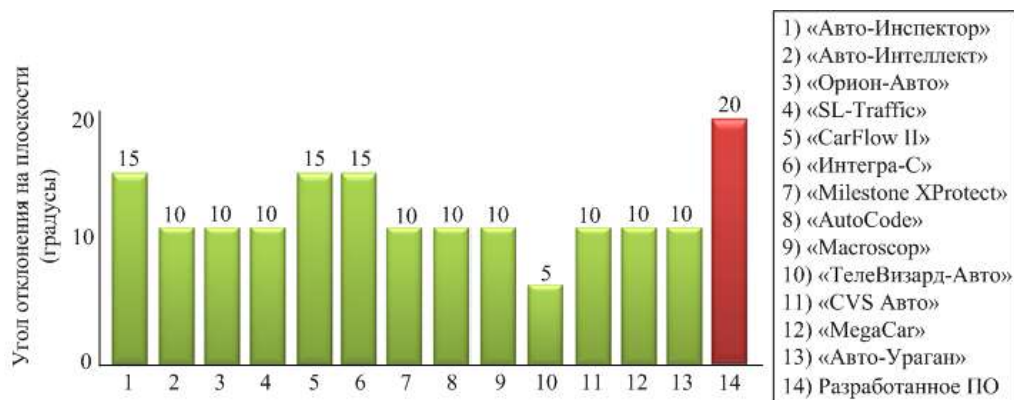


Рисунок 4.27 – Максимальные углы отклонения автомобильных номерных знаков на плоскости

4.4. Основные результаты и выводы по главе 4

1. В данной главе представлены результаты обучения и тестирования сверточных нейронных сетей на примере детектирования и распознавания автомобильных номерных знаков на изображениях. Проведены эксперименты по обучению и тестированию сверточных нейронных сетей с применением различных алгоритмов обучения. Наилучшие результаты обучения СНС-1 и СНС-2 были получены при использовании алгоритма Левенберга-Марквардта. Для СНС-3 наилучшие результаты обучения были получены при использовании алгоритма Resilient Propagation.

2. Представлены результаты тестирования разработанной программной системы по нескольким параметрам: точность распознавания; время распознавания; диапазон освещенности; точность распознавания автомобильных номерных знаков при различных углах отклонения относительно регистрирующего устройства.

3. Результаты проведенных экспериментов показали, что разработанные алгоритмы обеспечивают высокую эффективность при решении задач распознавания автомобильных номерных знаков, а так же обеспечивают высокую степень инвариантности к шумовым, аффинным и проекционным искажениям входных данных.

4. Представлено сравнение технических характеристик разработанной программной системы с существующими системами распознавания

автомобильных номерных знаков. Разработанная программная система по большинству параметров превышает эффективность существующих аналогов.

ЗАКЛЮЧЕНИЕ

Диссертационная работа посвящена разработке алгоритмов распознавания символов на сложном фоне. Реализация и апробация алгоритмов осуществлялась на примере распознавания автомобильных номерных знаков. В результате выполнения работы получены следующие научные и практические результаты:

1. Предложена структура и конфигурация сверточных нейронных сетей, обеспечивающая необходимую надежность и высокую степень инвариантности к различным шумовым, аффинным и проекционным искажениям входных данных.

2. Разработан алгоритм детектирования области расположения символов на изображениях со сложной фоновой структурой, отличающийся от аналогов применением алгоритмической композиции из двух сверточных нейронных сетей, работающих по принципу последовательной классификации. Алгоритм обеспечивает высокую точность классификации и повышенную устойчивость к шумовым, аффинным и проекционным искажениям входных данных.

3. Реализована модификация алгоритма нормализации изображений символов, основанная на построении гистограмм распределения яркости пикселей, отличающаяся от известных реализаций меньшей вычислительной сложностью и обеспечивающая более высокую скорость работы. Алгоритм обеспечивает обнаружение точных границ пластины номерного знака, расположенного под различными углами наклона.

4. Разработан алгоритм распознавания символов, основанный на применении сверточной нейронной сети и отличающийся от аналогов тем, что позволяет распознавать символы, не используя процесс их сегментации. Алгоритм обеспечивает высокую скорость работы и высокую степень инвариантности к шумовым, аффинным и проекционным искажениям символов на изображениях.

5. Разработанные алгоритмы позволили создать программную систему, которая обеспечивает распознавание автомобильных номерных знаков, соответствующих ГОСТ Р 50577 – 93, группе 1, типу 1 и обладает следующими характеристиками:

- точность распознавания автомобильных номерных знаков не менее 98,2%;

- время распознавания не более 64 мс;

- минимальная высота символов на изображении составляет 10 пикселей.

Условия регистрации автомобильных номерных знаков, при которых показатели точности распознавания сохраняются на максимально высоком уровне:

- освещенность в диапазоне от 50 лк до 1000 лк;

- максимальный угол отклонения по горизонтали относительно регистрирующего устройства до ± 55 градусов;

- максимальный угол отклонения по вертикали относительно регистрирующего устройства до ± 65 градусов;

- максимальный угол отклонения на плоскости относительно регистрирующего устройства до ± 20 градусов.

Разработанные алгоритмы и программные средства могут применяться для решения широкого круга аналогичных задач.

СПИСОК СОКРАЩЕНИЙ

ИНС	Искусственная нейронная сеть
ОЗУ	Оперативное запоминающее устройство
ПО	Программное обеспечение
ПК	Персональный компьютер
СНС	Сверточная нейронная сеть
ЦП	Центральный процессор
ВНА	<i>(Brightness histograms analysis)</i> Анализ гистограмм яркости изображения
BN	<i>(Binarization)</i> Бинаризация
ССА	<i>(Connected components analysis)</i> Анализ связных компонент
ED	<i>(Edge detection)</i> Выделение границ
HOG	<i>(Histogram of Oriented Gradients)</i> Гистограмма направленных градиентов
HT	<i>(Hough transform)</i> Преобразование Хафа
HTM	<i>(Hierarchical temporal memory)</i> Модель иерархической временной памяти
KNN	<i>(K-nearest neighbours)</i> Метод k-ближайших соседей
МА	<i>(Morphological analysis)</i> Морфологический анализ
LDA	<i>(Linear Discriminant Analysis)</i> Линейный дискриминантный анализ
РСА	<i>(Principal Component Analysis)</i> Метод главных компонент
ТМ	<i>(Template method)</i> Метод сравнения с шаблоном
VJA	<i>(Viola Jones algorithm)</i> Алгоритм Виолы-Джонса
WH	<i>(Wavelet Haar)</i> Вейвлет Хаара

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Абламейко, С.В. Обработка изображений: технология, методы, применение / С.В. Абламейко, Д.М. Лагуновский. – Минск: Амалфея, 2000. – 304 с.
2. Абрамов, Е.С. Моделирование систем распознавания изображений (на примере печатных текстов): Автореферат диссертации на соискание ученой степени кандидата физико-математических наук / Абрамов, Е.С. – Москва, 2006.
3. Алпатов, Б.А. Методы автоматического обнаружения и сопровождения объектов. Обработка изображений и управление / Б.А. Алпатов, П.В. Бабаян, О.Е. Балашов, А.И. Степашкин. – М.: Радиотехника, 2008. – 176 с.
4. Андреев, С.В. Алгоритмическое обеспечение прототипа устройства считывания паспортов и виз / С.В. Андреев, А.В. Бондаренко, В.И. Горемычкин, А.В. Ермаков, С.Ю. Желтов // Препринты Института прикладной математики им. М. В. Келдыша. – 2003. – № 4. – 32 с.
5. Анисимов, Б. В. Распознавание и цифровая обработка изображений / Б. В. Анисимов, В. Д. Курганов, В. К. Злобин. – М.: Высшая школа, 1983. – 295 с.
6. Антонюк, В. А. Автоматизация визуального контроля изделий микроэлектроники / В. А. Антонюк, Ю. П. Пытьев // Радиотехника и электроника. – М.: МГУ, 1985. – № 12. – С. 2456–2458.
7. Афонасенко, А. В. Программный комплекс поиска и распознавания автомобильных номеров / А.В. Афонасенко // Материалы XIV Международной научной студенческой конференции «Студент и научно-технический прогресс». Новосибирск. – Новосибирск: НГУ, 2007. – С. 102–103.
8. Афонасенко, А. В. Обзор методов распознавания структурированных символов / А.В. Афонасенко, А.И. Елизаров // Доклады Томского государственного университета систем управления и радиоэлектроники. – 2008. Т.2(18). – С. 83–88.
9. Вапник, В.Н. Теория распознавания образов / В.Н. Вапник, А.Я. Червоненкис. – М.: Наука, 1974. – 416 с.

10. Вежневцев, В.П. Алгоритмы анализа изображения лица человека для построения интерфейса человек-компьютер: Автореферат диссертации на соискание ученой степени кандидата физико-математических наук / В.П. Вежневцев. – Москва, 2004.
11. Вежневцев, А. Выделение связных областей в цветных и полутоновых изображениях / А. Вежневцев // Компьютерная графика и мультимедиа. – 2003. – № 1(5). – С. 56–62.
12. Воронцов, К.В. Козволюционный метод обучения алгоритмических композиций / К.В. Воронцов, Д.Ю. Каневский // Таврический вестник информатики и математики. – 2005. – № 2. – С. 51–66.
13. Воронцов, К.В. Лекции по алгоритмическим композициям / К.В. Воронцов // Кафедра математических методов прогнозирования. ВМиК МГУ. – 2007. – 45 с.
14. Воронцов, К.В. О проблемно-ориентированной оптимизации базисов задач распознавания / К.В. Воронцов // ЖВМ и МФ. – 1998. – Т.38. – № 5. – С. 870–880.
15. Галушкин, А.И. Многослойные системы распознавания образов: монография / А.И. Галушкин. –МИЭМ, 1970.
16. Галушкин А.И. Синтез многослойных систем распознавания образов / А.И. Галушкин А.И. – М.: «Энергия», 1974.
17. Гашников, М. В. Методы компьютерной обработки изображений. 2-е издание / М. В. Гашников. М.: ФИЗМАТЛИТ, 2003. – 784 с.
18. Глазунов, А.С. Автоматическое распознавание и идентификация лиц / А.С. Глазунов // Труды Академии Управления МВД РФ. Компьютерные технологии в криминалистике и информационная безопасность. – Москва, 1997. – С. 74–87.
19. Глазунов, А.С. Компьютерное распознавание человеческих лиц / А.С. Глазунов // Журнал Зарубежная Радиоэлектроника. Успехи современной радиоэлектроники. – Москва, 1997. – № 8. – С. 3–14.

20. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М.: Техносфера, 2005. – 1072 с.
21. Горелик, А.Л. Методы распознавания / А.Л. Горелик, В.А. Скрипкин. – М.: Высшая школа, 1977. – 222 с.
22. Горелик, А.Л. Современное состояние проблемы распознавания. Некоторые аспекты / А.Л. Горелик, И.Б. Гуревич, В.А. Скрипкин. – М.: Радио и связь, 1985. – 162 с.
23. Горлов, Д.В. Распознавание изображений на основе признаков, инвариантных к сдвигу, вращению, масштабированию: Автореферат диссертации на соискание ученой степени кандидата физико-математических наук / Д.В. Горлов. – Красноярск, 2002.
24. ГОСТ 50577–1993. Знаки государственные регистрационные транспортных средств типы и основные размеры. Технические требования. Введен 01.01.1994. – М.: Издательство стандартов. – 33 с.
25. Друки А.А. Применение сверточных нейронных сетей для выделения и распознавания автомобильных номерных знаков на изображениях / А.А. Друки // Известия Томского политехнического университета. – Томск: ТПУ, 2014. – Т. 324. – № 5. – С. 85–91.
26. Друки А.А. Распознавание структурированных символов на изображениях с использованием гистограмм средней интенсивности и сверточной нейронной сети / А.А. Друки // Известия Томского политехнического университета. – Томск: ТПУ, 2012. – Т. 320. – № 5. – С. 65–69.
27. Евстигнеев, В.А. Применение теории графов в программировании / В.А. Евстигнеев. – М.: Наука, 1985. – 125 с.
28. Елизаров, А.И. Методика построения систем распознавания автомобильного номера / А.И. Елизаров, А.В. Афонасенко // Известия томского политехнического университета, – Томск: ТПУ, 2006. – Т. 309. – № 8. – С. 118–122.
29. Загоруйко, Н.Г. Методы распознавания и их применение / Н.Г. Загоруйко. – М.: Сов.радио, 1972. – 208 с.

30. Загоруйко, Н.Г. Прикладные методы анализа данных и знаний / Н.Г. Загоруйко. – Новосибирск: ИМ СО РАН, 1999. – 270 с.
31. Желтов, С.Ю. Система распознавания и визуализации характерных черт человеческого лица в реальном времени на персональной ЭВМ с использованием web-камеры / С.Ю. Желтов, Ю.В. Визильтер, М.В. Ососков // Материалы XII Международной конференции по Компьютерной графике и машинному зрению. – Нижний Новгород, 2002. – С. 251–254.
32. Журавлев, Ю.И. Распознавание. Математические методы. Программная система. Практические применения / Ю.И. Журавлев, В.В. Рязанов, О.В. Сенько. – М.: ФАЗИС, 2006. – 176 с.
33. Журавлев, Ю.И. Распознавание образов и распознавание изображений. Распознавание, классификация, прогноз / Ю.И. Журавлев, И.Б. Гуревич. – М.: Наука, 1989. – Т. 2. – 69 с.
34. Ковалевский, В.А. О корреляционном методе распознавания / В.А. Ковалевский // Читающие автоматы (Наукова думка). – Киев, 1965. – С. 46–61.
35. Коробейников, А.П. Методы распознавания образов: Учебное пособие / А.П. Коробейников. – Ростов-на-Дону: Издательский центр ДГГУ, 1999. – 51 с.
36. Котович, Н.В. Проблемы человеко-машинного интерфейса: ввод печатных символов / Н.В. Котович, А.А. Кравченко // Математические методы в распознавании образов и дискретной оптимизации. – Москва, 1990. – С. 84–94.
37. Крючин, О.В. Параллельные алгоритмы обучения искусственной нейронной сети Quickprop и Rprop / О.В. Крючин, А.А. Арзамасцев // Вестник Тамбовского университета. – 2012. – Т. 17. – № 1. – С. 34–41.
38. Лаборатория компьютерной графики ВМК МГУ [Электронный ресурс]. – Режим доступа: <http://courses.graphics.cs.msu.ru>. Дата обращения: 15.11.2014.
39. Макаренко, А.А. Методика локализации изображения лица для систем видеоконтроля на основе нейронной сети / А.А. Макаренко, В.Т. Калайда // Известия Томского политехнического университета. – 2006. – Т. 309. – № 8. – С. 113–118.

40. Местецкий, Л.М. Математические методы распознавания образов / Л.М. Местецкий. – М.: МГУ, ВМиК, 2002. – 85 с.
41. Местецкий, Л. М. Скелетизация многоугольной фигуры на основе обобщенной триангуляции Делоне / Л.М. Местецкий // Программирование. – 1999. – № 3. – С. 16–31.
42. Панканти, Ш. Биометрия: будущее идентификации / Ш. Панканти, Р. М. Болле, Э. Джейн // Открытые Системы. – 2000. – № 3. – С. 46–49.
43. Пахирка, А.И. Компьютерный метод локализации лиц на изображениях в сложных условиях освещения: Автореферат диссертации на соискание ученой степени кандидата технических наук / А.И. Пахирка. – Красноярск, 2011.
44. Прэтт, У. Цифровая обработка изображений / У. Прэтт. Перевод с англ. – М.: Мир, 1982. – Т. 1. – 312 с.
45. Пытьев, Ю. П. Об автоматизации сравнительного морфологического анализа электронно-микроскопических изображений / Ю. П. Пытьев, С. С. Задорожный, Е. Лукьянов // Изв. АН СССР. Сер. Физическая. – 1977. – Т. 41. – № 11. – С. 37–44.
46. Пытьев, Ю. П. Задачи морфологического анализа изображений / Ю.П. Пытьев // Математические методы исследования природных ресурсов Земли из космоса. – М: Наука, 1984. – С. 41–83.
47. Пытьев, Ю. П. Морфологический анализ изображений / Ю.П. Пытьев // Докл. АН СССР. – 1983. – Т.269. – № 5. – С. 1061–1064.
48. Рахманкулов, В.З. Алгоритм распознавания объемных образов на базе модифицированного метода максимальной клики / В.З. Рахманкулов // Сборник трудов Института системного анализа РАН. – М.: РАН, 2002. – С. 85–91.
49. Редько, В.Г. Эволюционная кибернетика / В.Г. Редько. – М.: Наука, 2003. – 156 с.
50. Садыков, С.С. Скелетизация бинарных изображений / СС. Садыков, И.Р. Самандаров // Зарубежная радиоэлектроника. – 1985. – № 11. – С. 30–37.

51. Семенков О.И. Методы и алгоритмы обработки растровой графической информации / О.И. Семенков, С.В. Абламейко. – Минск: Инст. тех. кибернетики АН БССР, 1984. – 116 с.
52. Симанков, В.С. Адаптивное управление сложными системами на основе распознавания образов / В.С. Симанков, Е.В. Луценко. – Краснодар: Техн. ун-т Кубан. гос. технол. ун-та, 1999. – 318 с.
53. Система «Авто-Инспектор» [Электронный ресурс]. – Режим доступа: http://www.terna-opt.ru/html/comp_sys_ISS-AVTO.html. Дата обращения: 01.05.2015.
54. Система «Авто-Интеллект» [Электронный ресурс]. – Режим доступа: <http://www.itv.ru/products/intellect/autointellect>. Дата обращения: 01.05.2015.
55. Система «Авто-Номер» [Электронный ресурс]. – Режим доступа: [элвис-неотек.рф/index.php?id=193](http://elvis-neotek.ru/index.php?id=193). Дата обращения: 01.05.2015.
56. Система «Авто-Контроль» [Электронный ресурс]. – Режим доступа: [автоконтроль.su](http://avtokontrol.su). Дата обращения: 01.05.2015.
57. Система «Senesys-Avto» [Электронный ресурс]. – Режим доступа: <http://elvees.ru/home/index.php?id=39>. Дата обращения: 01.05.2015.
58. Система «Орион-Авто» [Электронный ресурс]. – Режим доступа: http://bolid.ru/production/orion/po-orion/orion_avto.html. Дата обращения: 01.05.2015.
59. Система «SL-Traffic» [Электронный ресурс]. – Режим доступа: <http://www.a383.ru/solution/detektorlic/82.html>. Дата обращения: 01.05.2015.
60. Система «Дигнум-Авто» [Электронный ресурс]. – Режим доступа: <http://www.dignum.ru/gotovyie-resheniya/auto.html>. Дата обращения: 01.05.2015.
61. Система «CarFlow II» [Электронный ресурс]. – Режим доступа: http://mpixel.ru/public_htm/lpr.htm. Дата обращения: 01.05.2015.
62. Система «Интегра-С» [Электронный ресурс]. – Режим доступа: <http://www.integra-s.com/products/integraskd/integra-video-avto.php>. Дата обращения: 01.05.2015.

63. Система «Auto-TRASSIR» [Электронный ресурс]. – Режим доступа: http://www.dssl.ru/products/autotrassir-1-kanal-do-200-km_ch. Дата обращения: 01.05.2015.

64. Система «SecurOS Auto» [Электронный ресурс]. – Режим доступа: <http://www.iss.ru/products/intelligent/auto>. Дата обращения: 01.05.2015.

65. Система «Milestone XProtect» [Электронный ресурс]. – Режим доступа: <http://www.milestone-spb.ru/product.php?ProductID=5692>. Дата обращения: 01.05.2015.

66. Система «AutoCode» [Электронный ресурс]. – Режим доступа: <http://www.milestone-spb.ru/product.php?ProductID=5782>. Дата обращения: 01.05.2015.

67. Система «Macroscop» [Электронный ресурс]. – Режим доступа: <http://macroscop.com/raspoznavanie-avtomobilnyix-nomerov.html>. Дата обращения: 01.05.2015.

68. Система «ТелеВизард-Авто» [Электронный ресурс]. – Режим доступа: <http://nordavind.ru/node/102>. Дата обращения: 01.05.2015.

69. Система «CVS Авто» [Электронный ресурс]. – Режим доступа: http://www.cvsnt.ru/products_soft_cvsauto.html. Дата обращения: 01.05.2015.

70. Система «MegaCar» [Электронный ресурс]. – Режим доступа: <http://mpixel.ru/cf.htm>. Дата обращения: 01.05.2015.

71. Система «Авто-Ураган» [Электронный ресурс]. – Режим доступа: http://www.petrosk.ru/catalogpsk/gibdd/kompleksy_videofikscacii_stacionarnie/apk_aut_ouragan. Дата обращения: 01.05.2015.

72. Сойфер, В.А. Теоретические основы цифровой обработки изображений: Учебное пособие / В.А. Сойфер, В.В. Сергеев, С.Б. Попов, В.В. Мясников. – Самара: СГАУ, 2000. – 256 с.

73. Сойфер В.А. Компьютерная обработка изображений / В.А. Сойфер // Соровский образовательный журнал. – 1996. – №2. – С.110–121.

74. Солдатов, О.П. Применение сверточной нейронной сети для распознавания рукописных цифр / О.П. Солдатов, А.А. Гаршин // Компьютерная оптика. – 2010. – Т. 34. – № 2. – С. 252–259.
75. Спицын, В.Г. Применение искусственных нейронных сетей для обработки информации / В.Г. Спицын, Ю.Р. Цой. – Томск: ТПУ, 2007. – 32 с.
76. Спицын, В.Г. Применение генетического алгоритма для решения задач оптимизации / В.Г. Спицын, Ю.Р. Цой. – Томск: ТПУ, 2007. – 27 с.
77. Фаворская, М.Н. Морфологическая обработка контурных изображений в системах распознавания текстовых символов / М.Н. Фаворская, А.С. Зотин, А.Н. Горошкин // Вестник СибГАУ. – 2007. – Т. 1(14). – С. 70–75.
78. Фан, Н.Х. Распознавание печатных текстов на основе применения вейвлет-преобразования и метода главных компонент. / Н.Х. Фан, Т.Т.Ч. Буй, В.Г. Спицын // Известия Томского политехнического университета. – Томск: ТПУ, 2012. – Т. 321. – № 5. – С. 154–158.
79. Ферцев, А.А. Ускорение обучения нейронной сети для распознавания изображений с помощью технологии NVIDIA CUDA / А. А. Ферцев // Вестник Самарского государственного технического университета. Физ.-мат. науки. – 2012. – Т. 1(26). – С. 183–191.
80. Фотографии транспортных средств и номерных знаков [Электронный ресурс] – Режим доступа: <http://avto-nomer.ru>. Дата обращения: 07.12.2014.
81. Фу, К. Структурные методы в распознавании образов. / К. Фу. Перевод с англ. – М.: Мир, 1977. – 320 с.
82. Хайкин, С. Нейронные сети: полный курс, 2-е издание / С. Хайкин. – М.: Издательский дом «Вильямс», 2006. – 1103 с.
83. Хемминг Р.В. Цифровые фильтры. / Р.В. Хемминг. Пер. с англ. – М.: Сов.радио, 1980. – 224 с.
84. Цой, Ю.Р. ECWorkshop – инструментальная библиотека классов для эволюционных вычислений / Ю.Р. Цой // Труды международных научно-технических конференций «Интеллектуальные системы (IEEE AIS'07)» и «Интеллектуальные САПР (CAD-2007)». – 2007. – С. 94–101.

85. Щепин, Е.В. К топологическому подходу в анализе изображений / Е. В. Щепин, Г. М. Непомнящий // Межвузовский сборник научных трудов «Геометрия, топология и приложения». – Москва, Мин. высшего и средн. спец. образ. РСФСР, Московский институт приборостроения. – 1990 г. – С. 13–25.
86. Abdullah, S.N.H.S. License Plate Recognition Based on Support Vector Machine / S.N.H.S. Abdullah, K. Omar, S. Sahran, M. Khalid // International Conference on Electrical Engineering and Informatics (ICEEI '09). – 2009. – Vol. 1. – P. 78–82.
87. Accord.NET Framework [Электронный ресурс]. – Режим доступа: <http://code.google.com/p/accord/downloads/list>. Дата обращения: 01.05.2015.
88. AForge.NET Framework [Электронный ресурс]. – Режим доступа: <http://code.google.com/p/aforge/>. Дата обращения: 01.05.2015.
89. Arth, C. Real-Time License Plate Recognition on an Embedded DSP-Platform / C. Arth, F. Limberger, H. Bischof // Proceedings of IEEE Conference of Computer Vision and Pattern Recognition (CVPR). – 2007. – Vol. 15. – P. 1–8.
90. Bolotova, Y.A. License plate recognition with hierarchical temporal memory model / Y.A. Bolotova, A.A. Druki, V.G. Spitsyn // The 9th International Forum on Strategic Technology (IFOST). – Cox's Bazar, Bangladesh, 2014. – P. 136–139.
91. Bradski, G.R. Computer Vision Face Tracking For Use in a Perceptual User Interface / G.R. Bradski // Intel Technology Journal, – 1998. – P. 1–15.
92. Bay, H. SURF: Speeded Up Robust Features / H. Bay, T. Tuytelaars, L.V. Gool // 9th European Conference on Computer Vision. – 2006. – P. 404–417.
93. Chaturvedi, A. Automatic License Plate Recognition System using SURF Features and RBF Neural Network / A. Chaturvedi, N. Sethi // International Journal of Computer Applications. – 2013. – Vol. 70. – № 27. – P. 37–41.
94. Chen, Y.N. The application of a convolution neural network on face and license plate detection / Y.N. Chen, C.C. Han, C.T. Wang // Proceedings of the 18th International Conference on Pattern Recognition. – 2006. – P. 552–555.

95. Cheriet, M. Character Recognition Systems A Guide for Students and Practitioners / M. Cheriet, N. Kharma, C.L. Liu, C.Y. Suen – Wiley, Canada, 2007. – 327 p.
96. Cheung, C.C. Improving the Quickprop Algorithm / C.C. Cheung, S.C. Ng, A.K. Lui // International Joint Conference of Neural Networks (IJCNN). – 2012. – P. 1–6.
97. Choubey, S. License Plate Recognition for High Security Registration Plates. International Journal of Computer Science and Information Technologies (IJCSIT). – 2015. – Vol. 6. – P. 1036–1039.
98. Cope, G. Optical Character Recognition using PCA [Электронный ресурс] / G. Cope – Режим доступа: <http://www.algosome.com/articles/optical-character-recognition-java.html>. Дата обращения: 01.05.2015.
99. Craw, I. Automatic Extraction of Face Features / I. Craw, H. Ellis, J. Lishman // Pattern Recognition Letters, –1987. – Vol. 5. – P. 183–187.
100. Cui, D. Sun. License plate detection algorithm based on Gentle AdaBoost algorithm with a cascade structure / D. Cui, Dong, B. Gu, H. Cai, J // Proceedings of IEEE International Conference on Robotics and Biomimetics. – 2009. – P.1962–1966.
101. Curado, A.F. Using PCA to Recognize Characters in Vehicle License Plates / A.F. Curado, A. Bauchspiess, V.B.F. Curado // Proceedings of the 2006 International Conference on Image Processing, Computer Vision, & Pattern Recognition, – 2006. – Vol. 1. – P. 244–251.
102. Dalal, N. Histograms of Oriented Gradients for Human Detection / N. Dalal, B. Triggs // Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). – 2005. – Vol. 1. – P. 886–893.
103. Darken, C. Note on Learning Rate Schedules for Stochastic Optimization / C. Darken, J. E. Moody //Advances in Neural Information Processing Systems. – 1991. – Vol. 3. – P. 83–90.
104. De Jong, K.A. Introduction to the Second Special Issue on Genetic Algorithms / K.A. De Jong // Machine Learning. – 1990. – Vol. 5(4).– P. 351–353.

105. Deb, K. An Efficient Method of Vehicle License Plate Detection Based on HSI Color Model and Histogram / K. Deb, H. Lim, S.J. Kang. K.H. Jo // Proceedings of 22nd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE), – 2009. – Vol. 32. – P. 66–75.
106. Deb, K. Parallelogram and Histogram based Vehicle License Plate Detection / K. Deb, H. U. Chae, K.H. Jo // Proceedings of the International Conference on Smart Manufacturing Application (ICSMA), – 2008. – Vol. 12. – P. 349–353.
107. Deb, K. Vehicle License Plate Detection Method Based on Sliding Concentric Windows and Histogram / K. Deb, H. U. Chae, K.H. Jo // Journal of Computers, – 2009. – Vol. 4. – № 8. – P. 771–777.
108. Fahlman, S.E. An Empirical Study of Learning Speed in Backpropagation Networks / S.E. Fahlman // Computer Science Department of Carnegie Mellon University. – 1988. – 21 p.
109. Farley, B.G. Simulation of Self-Organizing Systems by Digital Computer. / B.G. Farley, W.A. Clark // IRE Transactions on Information Theory. – 1954. – № 4. – P. 76–84.
110. Fisher, R.A. The use of multiple measurements in taxonomic problems / R.A. Fisher // Annals of Eugenics, – 1936. – Vol. 7. – Issue 2. – P. 179–188.
111. Freund, Y. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting / Y. Freund, R.E. Schapire // Journal of computer and system sciences. – 1997. – P. 119–139.
112. Freund, Y. A Short Introduction to Boosting / Y. Freund, R.E. Schapire // Journal of Japanese Society for Artificial Intelligence. – 1999. – Vol. 14. – P. 771–780.
113. Fujisawa, H. Directional Pattern Matching for Character Recognition revisited / H. Fujisawa, C.L. Liu // Proceedings of the 7th International Conference on Document Analysis and Recognition. – Edinburgh, Scotland, 2003. – P. 794–798.
114. Gilly, D. License Plate Recognition – A Template Matching Method / D. Gilly, K. Raimond // International Journal of Engineering Research and Applications (IJERA), – 2013. – Vol. 2. – P. 1240–1245.

115. Goodfellow, I.J Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks / I.J. Goodfellow, Y. Bulatov, J. Ibarz, S. Arnoud, V. Shet // Computer Vision and Pattern Recognition. – 2014. – P. 235–243.
116. Hadi, S.K. An Evaluation of License Plate Recognition Algorithms / S.K. Hadi // International Journal of Digital Information and Wireless Communications (IJDWC), – 2011. – Vol. 1. – P. 247–253.
117. Han, X. A Novel Parallel Neural Network Classifier For Automatic Vehicle License Plate Recognition / X. Han, H.M. Xie, G.Q. Zhang // Artificial Intelligence and Industrial Applications. – 2015. – Vol. 24. – P. 734–741.
118. Hao, D.Z. Research on License Plate Recognition Algorithm based on Support Vector Machine / D.Z. Hao, F. Xin // Journal of multimedia. – 2014. – Vol. 9. – № 2. – P. 253–260.
119. He, K. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition / K. He, X. Zhang, S. Ren, J. Sun // IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI). – 2015. – Vol. 4. – P. 534–542.
120. Ho, W.T. Twostage license plate detection using gentle Adaboost and SIFT-SVM / W.T. Ho, H.W. Lim, Y.H. Tay // Proceedings of 1st Asian Conference on Intelligent Information and Database System (ACIIDS). – 2009. – P. 109–114.
121. Holland, J. H. Adaptation in Natural and Artificial Systems / J.H. Holland // University of Michigan Press, Ann Arbor. – 1975. – P. 432–439.
122. Holland, J. H. Genetic Algorithms and Adaptation / J.H. Holland // Adaptive Control of Ill-Defined Systems. – 1984. – P. 317–333.
123. Hooi, S.N. Detection and Recognition of Malaysian Special License Plate Based on SIFT Features / S.N. Hooi, H.T. Yong, M.L. Kim, M. Hamam, W.H. Hock // Computer Vision and Pattern Recognition (CVPR). – 2015. – P. 321–329.
124. Hotelling, H. Analysis of a complex of statistical variables into principal components / H. Hotelling // Journal of Educational Psychology, – 1933. – Vol. 24. – P. 417–441.
125. Hough, P.V.C. Method and Means for Recognizing Complex Patterns / P.V.C. Hough. – U.S.A, 1962.

126. Howard, A.G. Some Improvements on Deep Convolutional Neural Network Based Image Classification / A.G. Andrew // International Conference on Learning Representations (ICLR). – 2014. – Vol. 10. – № 4. – P. 652–659.
127. Hu. M.K. Visual Pattern Recognition by Moment Invariants / M.K. Hu // IRE Trans. Info. Theory. – 1962. Vol. 8. – P. 179–187.
128. Igel, C. Empirical Evaluation of the Improved Rprop Learning Algorithm / C. Igel, M. Husken // Neurocomputing. – 2003. – Vol. 50. – P. 105–123.
129. Jacobs, R.A. Increased Rates of Convergence Through Learning Rate Adaptation / R.A. Jacobs // Neural Networks. – 1988. – Vol. 12. – P. 295–307.
130. Kamat, V. An Efficient Implementation of the Hough Transform for Detecting Vehicle License Plates Using DSP'S / V. Kamat, S. Ganesan // Proc. of IEEE 31-st Annual 1997 International Carnahan Conference, – 1997. – P. 209–218.
131. Kan, C. Invariant Character Recognition with Zernike and Orthogonal Fourier-Mellin Moments / C. Kan, M.D. Srinath // Pattern Recognition. – 2000. – Vol. 35. – P. 143–154.
132. Khalil, M.I. Car Plate Recognition Using the Template Matching Method / M.I. Khalil // International Journal of Computer Theory and Engineering, – 2010. – Vol. 2. – № 5. – P. 241–248.
133. Khotanzad, A. Invariant Image Recognition by Zernike Moments. / A. Khotanzad, Y.H. Hong // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 1990. – Vol. 12. – № 5. – P. 489–497.
134. Kim, J.B. MSER and SVM-Based Vehicle License Plate Detection and Recognition System / J.B. Kim // Proceeding of International Conference on Convergence and Hybrid Information Technology (ICHIT). – 2012. – Vol. 310. – P. 529–535.
135. Krizhevsky, A. ImageNet Classification with Deep Convolutional Neural Networks / A. Krizhevsky, I. Sutskever, G. Hinton // Conference on Neural Information Processing Systems (NIPS). – 2012. – 27 p.
136. Lawrence, S. Face Recognition: A Convolutional Neural Network Approach / S. Lawrence, C. L. Giles, A. C. Tsoi, A. D. Back // IEEE Transactions on Neural

Networks, Special Issue on Neural Networks and Pattern Recognition. – 1997. – P. 1–24.

137. LeCun, Y. Efficient BackProp / Y. LeCun, L. Bottou, G.B. Orr, K.R. Muller // Neural Networks: Tricks of the trade. –1998. P. 5–50.

138. LeCun, Y. Gradient-Based Learning Applied to Document Recognition / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // Proceedings of the IEEE. – 1998. – Vol. 86. – № 11. – P. 2278–2324.

139. LeCun, Y. Generalization and Network Design Strategies / Y. LeCun // Proceedings of the International Conference Connectionism in Perspective. –1988. – Vol. 11. – P. 523.

140. Levenberg, K. A Method for the Solution of Certain Problems in Least Squares / K. Levenberg // Quarterly of Applied Mathematics. – 1944. – Vol. 2. – P. 164–168.

141. Lowe, D.G. Distinctive Image Features from Scale-Invariant Keypoints / D.G. Lowe // International Journal of Computer Vision. – 2004. – Vol. 60(2). – P. 91–110.

142. Lowe, D.G. Object Recognition from Local Scale-Invariant Features / D.G. Lowe // Proceedings of the International Conference on Computer Vision. – 1999. – V. 2. – P. 1150–1157.

143. Marquardt, D. An Algorithm for Least-Squares Estimation of Nonlinear Parameters / D. Marquardt // SIAM Journal on Applied Mathematics. – 1963. – Vol. 11. – P. 431–441.

144. Martinsky, O. Algorithmic and mathematical Principles of automatic number plate Recognition systems: Thesis / O. Martinsky.–Faculty of information technology. Department of intelligent systems. – 2007. – 83 p.

145. Massa, F. Convolutional Neural Networks for joint object detection and pose estimation: A comparative study / F. Massa, M. Aubry, R. Marlet // Computer Vision and Pattern Recognition(CVPR). – 2015. – Vol. 4. – P. 231–239.

146. McCulloch, W. A Logical Calculus of Ideas Immanent in Nervous Activity / W. McCulloch, W. Pitts // Bulletin of Mathematical Biophysics. –1943. – Vol. 5. – № 4. – P. 115–133.
147. Mitchell, M. Genetic Algorithms and Artificial Life / M. Mitchell, S. Forrest // Artificial Life. – 1994. – P. 267–289.
148. Murata, N. Adaptive on-line learning in changing environments / N. Murata, K.R. Müller, A. Ziehe, S. Amari // Advances in Neural Information Processing Systems. The MIT Press. – 1997. – Vol. 9. – P. 599–607.
149. OpenCV Library [Электронный ресурс]. – Режим доступа: <http://opencv.org/downloads.html>. Дата обращения: 01.05.2015.
150. Ouyang, W. DeepId-Net: Multi-Stage and Deformable Deep Convolutional Neural Networks for Object Detection / W. Ouyang, P. Luo, X. Zeng, S. Qiu, Y. Tian, H. Li, S. Yang, Z. Wang, Y. Xiong, C. Qian, Z. Zhu, R. Wang, C. Loy, X. Wang, X. Tang // Computer Vision and Pattern Recognition(CVPR). – 2014. – Vol. 22. –P. 734–741.
151. Panchal, P.M. A Comparison of SIFT and SURF / P. M. Panchal, S. R. Panchal, S. K. Shah // International Journal of Innovative Research in Computer and Communication Engineering. – 2013. – Vol. 1. – Issue 2. – P. 323–329.
152. Patel, S.G. Vehicle license plate recognition using morphology and neural network / S.G. Patel // International Journal on Cybernetics & Informatics (IJCI). – 2013. – Vol. 2. – № 1. – P. 363–370.
153. Pearson, K. On Lines and Planes of Closest Fit to Systems of Points in Space / K. Pearson // Philosophical Magazine, – 1901. – Vol. 2(11). – P. 559–572.
154. Prates, R.C. An Adaptive Vehicle License Plate Detection at Higher Matching Degree / R.C. Prates, G. Camara-Chavez, W.R. Schwartz, D. Menotti // 19th Iberoamerican Congress on Pattern Recognition. – 2014. – P. 454–461.
155. Riedmiller, M. A Direct Adaptive Method for Faster Backpropagation Learning: the RPROP Algorithm / M. Riedmiller, H. Braun // International Conference of Neural Networks. – 1993. – Vol. 1. – P. 586–591.

156. Riedmiller, M. Rprop – A Fast Adaptive Learning Algorithm / M. Riedmiller, H. Braun // Proceedings of the VII International Symposium on Computer and Information Science. – 1992. – Vol. 5. – P. 472 – 480.
157. Raseena A. Automatic Skew Detection and Localisation of Vehicle License Plate Using Hough Transform / A. Raseena, S. Muhammad // International Journal of Scientific & Engineering Research, – 2013. – Vol. 4(8). – P. 121–128.
158. Rashed, B. Automatic Number Plate Recognition / B. Rashed, H.A. Kader, H. Rafaat, M. Sharaf, M. Shehata // International Journal of Scientific and Research Publications, – 2013. – Vol. 3. – Issue 12. – P. 231–238.
159. Rasheed, S. Automated Number Plate Recognition Using Hough Lines and Template Matching / S. Rasheed, A. Naeem, O. Ishaq // Proceedings of the World Congress on Engineering and Computer Science (WCECS), – 2012. – Vol. 1. – P. 345–349.
160. Razavian, A.S. CNN Features off-the-Shelf: An Astounding Baseline for Recognition / A.S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson // Computer Vision and Pattern Recognition (CVPR). – 2014. – Vol. 21. – 12 p.
161. Rosenblatt, F. The Perceptron: A Probabilistic Model For Information Storage And Organization In The Brain / F. Rosenblatt // Psychological Review. – 1958. – Vol. 65. – № 6. – P. 386–408.
162. Rumelhart, D. Learning Representations of Back-Propagation Errors // D. E. Rumelhart, G. E. Hinton, R. J Williams / Nature. – 1986. – № 323. – P. 533–536.
163. Russakovsky, O. ImageNet Large Scale Visual Recognition Challenge / O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei // Computer Vision and Pattern Recognition (CVPR). – 2015. – Vol. 18. – 43 p.
164. Sakai, T. Line Extraction and Pattern Detection in a Photograph / T. Sakai, M. Nagao, S. Fujibayashi // Pattern Recognition, – 1969. – Vol. 1. – P. 233–248.
165. Senthilraja, P. Advanced Vehicle Tracking and Model Recognition in Rural Areas using SURF Method / P. Senthilraja, T. Karthikeyan // International Journal of

Advanced Research in Computer Science & Technology (IJARCST). – 2014. – Vol. 11. – P.235–236.

166. Shaaban, Z. An Intelligent License Plate Recognition System / Z. Shaaban // International Journal of Computer Science and Network Security (IJCSNS), – 2011. – Vol. 11. № 7.– P. 55–61.

167. Simonyan, K. Very Deep Convolutional Networks for Large-Scale Image Recognition / K. Simonyan, A. Zisserman // International Conference on Learning Representations (ICLR). – 2015. – 14 p.

168. Solanki, R. The Automatic License Plate Recognition / R. Solanki, R.K. Rai, T, Raikwa // International Journal of Engineering Research and Applications. – 2013. – Vol. 3. – Issue 6. – P. 304–310.

169. Sompolinsky, H. On-line learning of dichotomies: algorithms and learning curves / H. Sompolinsky, N. Barkai, H.S. Seung // Neural Networks: The Statistical Mechanics Perspective. Singapore. – 1995. – P. 105–130.

170. Song, M.K. Modeling and Implementing Two-Stage AdaBoost for Real-Time Vehicle License Plate Detection / M.K. Song, M.M.K. Sarker // Journal of Applied Mathematics. – 2014. – P. 546–552.

171. Spulak, D. Evaluation of PCA, LDA and Fisherfaces in Appearance-based Object Detection in Thermal Infra-red Images with Incomplete Data / D. Spulak, R. Otrebski, W. Kubinger // 25th International Symposium on Intelligent Manufacturing and Automation, – 2014. – Vol. 35. – P. 1167–1173.

172. Srikanthan, G. Gradient-based Contour Encoder for Character Recognition / G. Srikanthan, S.W. Lam, S.N. Srihari // Pattern Recognition. – 1996. – Vol. 29. – № 7. – P. 1147–1160.

173. Su, C.L. Chinese-seal-print recognition by color image dilating, extraction, and gray scale image geometry comparison / C.L. Su // Journal of Intelligent and Robotic Systems. – 2005. – № 4. – C. 349–359.

174. Transtrum, M.K. Improvements to the Levenberg-Marquardt Algorithm for Nonlinear Least-Squares Minimization / M.K. Transtrum, J.P. Sethna // Preprint Submitted to Journal of Computational Physics. – 2012. – P. 243–249.

175. Vaktin, M. The System of Handwritten Characters Recognition on the Basis of Legendre Moments and Neural Network / M. Vaktin, M. Selinger // The International Workshop on DiscreteEvent System Design (DESDes). – 2001. – P. 521–528.
176. Vapnik, V.N. A Training Algorithm for Optimal Margin Classifiers / V.N. Vapnik, E. Boser, I.M. Guyon // Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory. –1992. – № 18. – P. 144–152.
177. Vapnik, V.N. Support Vector Networks / V. Vapnik, C. Cortes // Machine Learning. – 1995. – № 20(3). – P. 273–297.
178. Vehicle characters database «Database VLP characters» [Электронный ресурс] – Режим доступа: <https://yadi.sk/d/U41QZ8v7cpJ6R>. Дата обращения: 05.10.2014.
179. Vehicle license plates database «Database VLP number plates» [Электронный ресурс] – Режим доступа: <https://yadi.sk/d/EAfnQ947criHW>. Дата обращения: 05.10.2014.
180. Vehicle license plates database «Database VLP images» [Электронный ресурс] – Режим доступа: <https://yadi.sk/d/0H2AipxrcrXqy>. Дата обращения: 05.10.2014.
181. Viola, P. Fast and Robust Classification Using Symmetric AdaBoost and a Detector Cascade / P. Viola, M. Jones // Proceedings of Neural Information Processing Systems. – Vancouver, 2001. – P. 1311–1318.
182. Viola, P. Rapid Object Detection using a Boosted Cascade of Simple Features / P. Viola, M.J. Jones // Proceedings IEEE Conference on Computer Vision and Pattern Recognition (CVPR). – 2001. – Vol. 1. – P. 511–518.
183. Viola, P. Robust real-time face detection / P. Viola, M.J. Jones // International Journal of Computer Vision. – 2004. – Vol. 57. – № 2. – P. 137–145.
184. Vondrick, C. Visualizing Object Detection Features / C. Vondrick // In Partial Fulfillment of the Requirements for the Degree of Master of Science. Massachusetts Institute of Technology. – 2013. – 61 p.

185. Wang, R.M. AdaBoost for Feature Selection, Classification and Its Relation with SVM / R.M. Wang // International Conference on Solid State Devices and Materials Science. –2012. – P. 800 – 807.
186. Wang, R.M. License plate detection using gradient information and cascade detectors / R.M. Wang, N. Sang, R. Huang, Y.H. Wang // International Journal for Light and Electron Optics. – 2014. – Vol. 125. – P. 186–190.
187. Werbos, P.J. Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences: Thesis / P.J. Werbos. – Harvard University. Cambridge, MA. – 1974. – 92 p.
188. Wilson, D.R. The General Inefficiency of Batch Training for Gradient Descent Learning / D.R. Wilson, T.R. Martinez // Neural Networks. – 2003. – Vol. 16. – P. 1429–1451.
189. Wilson, D.R. The Inefficiency of Batch Training for Large Training Sets / D.R. Wilson, T.R. Martinez // Proc.Int. Joint Conference Neural Networks (IJCNN'2000). – 2000. – Vol. 2. – P. 113–117.
190. Wu, G.P. Slant Correction of Vehicle License Plate Based on Feature Point and Principal Component Analysis / G.P. Wu, S. Cheng, H. Lei // International Conference on Computer Science and Software Engineering, –2008. – Vol. 11. – P. 487–490.
191. Yamashita, Y. Classification of handprinted Kanji characters by the structured segment matching method / Y. Yamashita, K. Higuchi, Y. Yamada, Y. Haga // Pattern Recognition Letters. – 1983. – Vol. 1. – P. 475–479.
192. Yoon, J. Car License Plate Detection under Large Variations Using Covariance and HOG Descriptors. / J. Yoon, B. Kang, D. Kim // 8th International Symposium on Visual Computing (ISVC). – 2012. – P. 636–647.
193. Zernike, F. Beugungs theorie des Schneiden verfahrens und Seiner Verbesserten Form, der Phasen contrast methode / F. Zernike // Physica. – 1934. – Vol. 1(8). – P. 689–704.

194. Zhang, Z. Hough Transform and Its Application in Vehicle License Plate Tilt Correction / Z. Zhang, S. Yin // Computer and Information Science.– 2008. –Vol. 1. – № 3. – P. 116–122.

ПЕРЕЧЕНЬ ПУБЛИКАЦИЙ ПО ТЕМЕ ДИССЕРТАЦИИ

Публикации в журналах, которые включены в Перечень рецензируемых научных изданий, рекомендованных Высшей аттестационной комиссией при Министерстве образования и науки Российской Федерации для опубликования основных научных результатов диссертаций:

1. Друки, А. А. Система поиска, выделения и распознавания лиц на изображениях / А.А. Друки, В.Г. Спицын // Известия Томского политехнического университета. – Томск: ТПУ, 2011. – Т. 318 – № 5. – С. 64–70.

2. Друки, А. А. Алгоритмы выделения лиц на статических RGB изображениях и в видеопотоке / А.А. Друки, В.Г. Спицын // Известия Томского политехнического университета. – Томск: ТПУ, 2012. – Т. 320 – № 5 – С. 65–69.

3. Друки, А. А. Распознавание структурированных символов на изображениях с использованием гистограмм средней интенсивности и сверточной нейронной сети / А.А. Друки, В.Г. Спицын // Известия Томского политехнического университета. – Томск: ТПУ, 2013. – Т. 322 – № 5. – С. 120–125.

4. Друки, А. А. Применение сверточных нейронных сетей для выделения и распознавания автомобильных номерных знаков на изображениях со сложным фоном / А.А. Друки, В.Г. Спицын // Известия Томского политехнического университета. – Томск: ТПУ, 2014. – Т. 324 – № 5. – С. 85–91.

5. Друки, А. А. Алгоритмы распознавания рукописных подписей на основе нейронных сетей / А.А. Друки, М.А. Милешин // Фундаментальные исследования. – Москва: Академия естествознания, 2014. – № 11. – С. 1906–1910.

Публикации в научных изданиях, индексируемых в базе данных Scopus:

6. Druki, A. A. Scaling and Clustering Algorithm for the Persons Search on the Images / A.A. Druki, V.G. Spitsyn // Proceedings of 7th International Forum on Strategic Technology (IFOST – 2012). Tomsk: TPU Press, September 18–21, 2012. – Vol. 1 – P. 592–595.

7. Bolotova, Y. A. License plate recognition with hierarchical temporal memory model / J.A. Bolotova, A.A. Druki, V.G. Spitsyn // Proceedings of 9th International Forum on Strategic Technology (IFOST – 2014). – Cox's Bazar, Bangladesh, October 21–23, 2014. – P. 136–139.

8. Druki, A. A. Scaling and Clustering Algorithm for the Persons Search on the Images / A.A. Druki, V.G. Spitsyn, J.A. Bolotova // Proceedings of The 10th International Forum on Strategic technology (IFOST – 2015). – Bali, Indonesia, June 3–5, 2015. – Vol. 5 – P. 423–428.

9. Druki, A. A. Application of Convolutional Neural Networks for Automatic Number Plate Recognition on Complex Background Images / A.A. Druki, J.A. Bolotova, V.G. Spitsyn // Applied Mechanics and Materials. – Trans Tech Publications, Switzerland, 2014 – Vol. 756. – P. 695–703.

10. Druki, A. A. Development of sequential optimizational algorithms for object detection in images / A.A. Druki, V.G. Spitsyn // International Siberian Conference on Control and Communications (SIBCON-2015). – Omsk, 21–32 may, 2015. – P. 172–176.

Публикации в других научных изданиях:

11. Друки, А. А. Алгоритмы выделения человеческих лиц на изображениях / А.А. Друки, В.Г. Спицын // Нейроинформатика, ее приложения и анализ данных: сборник научных трудов XIX Всероссийского научного семинара. – Красноярск: Сибирский федеральный университет, 1 – 3 Октября, 2011. – С. 44–46.

12. Друки, А. А. Разработка нейросетевых алгоритмов для локализации и распознавания лиц на изображениях / А.А. Друки, В.Г. Спицын // Нейрокомпьютеры и их применение: сборник научных трудов IX Всероссийской научной конференции. – Москва, 15 Марта, 2011. – С. 35–37.

13. Друки, А. А. Алгоритм масштабирования и кластеризации в системе обнаружения объектов на изображениях / А.А. Друки, В.Г. Спицын // Проблемы информатики. – Новосибирск: ИВМиМГ, 2011 – № 2(10) – С. 16–25.

14. Друки, А. А. Идентификация объектов на изображениях на основе нейронных сетей и алгоритма кластеризации / А.А. Друки, В.Г. Спицын // Микроэлектроника и информатика: материалы 18 Всероссийской межвузовской научно-технической конференции студентов и аспирантов. – Зеленоград: МИЭТ, 19 – 21 Апреля, 2012. – С. 144.

15. Друки, А. А. Разработка алгоритмов на основе деформируемых эллиптических моделей для обнаружения объектов на изображениях / А.А. Друки // Интеллект и наука: труды XII Международной научной конференции. – Красноярск: СФУ, 25–27 Апреля, 2012. – С. 109–110.

16. Друки, А. А. Разработка алгоритмов, решающих задачу идентификации объектов на изображениях / А.А. Друки // Нейроинформатика-2012: сборник научных трудов XIV Всероссийской научно-технической конференции. – Москва: НИЯУ МИФИ, 23–27 Января, 2012. – С. 42–51.

17. Друки, А. А. Разработка сверточной нейронной сети для распознавания символов на изображениях / А.А. Друки, В.Г. Спицын // Интеллект и наука: труды XIII Международной научной конференции. – Красноярск: СФУ, 16–18 Апреля, 2013. – С. 74–76.

18. Друки, А. А. Применение гибких шаблонов для выделения лиц на изображениях / А.А. Друки, В.Г. Спицын // Научная сессия ТУСУР – 2013: материалы Всероссийской научно-технической конференции студентов, аспирантов и молодых ученых. – Томск: В-Спектр, 15–17 Мая, 2013. – Т. 3. – С. 53–56.

19. Друки, А.А. Применение нейросетевых алгоритмов для распознавания рукописных подписей на изображениях / А.А. Друки, М.А. Милешин // Молодежь и современные информационные технологии: сборник трудов XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых: в 2 томах, – Томск: ТПУ, 12–14 Ноября, 2014. – С. 116–117.

20. Друки, А. А. Разработка алгоритмов выделения и распознавания автомобильных номерных знаков на изображениях со сложным фоном / А.А. Друки // Современные техника и технологии: сборник трудов XX международной

научно-практической конференции студентов, аспирантов и молодых ученых. – Томск: ТПУ, 14–18 Апреля, 2014. – Т. 2 – С. 171–172.

21. Каковкин, П. А. Алгоритмы выделения и идентификации дорожных знаков на изображениях / П.А. Каковкин, А.А. Друки, В.Г. Спицын // Молодежь и современные информационные технологии: сборник трудов XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых. – Томск: ТПУ, 12–14 Ноября, 2014. – Т. 1 – С. 152–153.

22. Друки, А. А. Разработка нейросетевых алгоритмических композиций для классификации объектов на изображениях / А.А. Друки // Технологии Microsoft в теории и практике программирования: сборник трудов XII Всероссийской научно-практической конференции студентов, аспирантов и молодых ученых. – Томск: ТПУ, 25–26 Марта, 2015. – С. 116–118.

ПРИЛОЖЕНИЕ 1



«УТВЕРЖДАЮ»

Проректор по образовательной
деятельности ТПУ

Ю.С.Боровиков

» 2015 г.

АКТ ВНЕДРЕНИЯ

**результатов диссертационной работы Друки Алексея Алексеевича «Алгоритмы
нейросетевого детектирования и распознавания символов на сложном фоне»**

Настоящим подтверждается, что результаты диссертационной работы Друки А. А. «Алгоритмы нейросетевого детектирования и распознавания символов на сложном фоне», представленной на соискание ученой степени кандидата технических наук по специальности 05.13.01 – «Системный анализ, управление и обработка информации», используются в учебном процессе на кафедре вычислительной техники, института кибернетики, Томского политехнического университета при подготовке курса «Методы интеллектуальной обработки и анализа изображений» для магистрантов направления 230100 «Информатика и вычислительная техника» магистерской программы «Компьютерный анализ и интерпретация данных».

В данном курсе осуществляется реализация разработанных Друки А.А. нейросетевых алгоритмов, позволяющих решать задачу обнаружения и распознавания символов на изображениях со сложным фоном, алгоритмы обучения нейронных сетей и алгоритмы предварительной обработки изображений. Цикл лабораторных работ апробирован при обучении магистрантов гр. 8BM31,8BM4A.

Директор ИК, ТПУ

А.А. Захарова

«28» 05 2015 г.

/ Зав. кафедрой ВТ

профессор, д.т.н. Н.Г. Марков

«28» 05 2015 г.

ФЕДЕРАЛЬНОЕ
МЕДИКО-БИОЛОГИЧЕСКОЕ АГЕНТСТВО
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ УЧРЕЖДЕНИЕ
**СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ
НАУЧНО-КЛИНИЧЕСКИЙ ЦЕНТР**
(ФГБУ СибФНКЦ ФМБА России)
636000, Томская обл., ЗАТО Северск,
г. Северск, ул. Мира, д. 4
Тел/факс: (382-3) 54-37-03
E-mail: kb81@med.tomsk.ru
ОГРН 1147024000309,
ИНН/КПП 7024038542/702401001

01.10.2015 № 02/4585
На № _____ от _____

АКТ ВНЕДРЕНИЯ

результатов диссертационной работы Друки Алексея Алексеевича
«АЛГОРИТМЫ НЕЙРОСЕТЕВОГО ДЕТЕКТИРОВАНИЯ И РАСПОЗНАВАНИЯ
СИМВОЛОВ НА СЛОЖНОМ ФОНЕ»

Данный документ удостоверяет, что результаты диссертационной работы Друки А.А. «Алгоритмы нейросетевого детектирования и распознавания символов на сложном фоне» используются в Северской клинической больнице Федерального государственного бюджетного учреждения «Сибирский федеральный научно-клинический центр Федерального медико-биологического агентства» в системах видеонаблюдения для решения задачи распознавания государственных регистрационных знаков транспортных средств на территории данного учреждения и для отслеживания служебного транспорта.

Главный врач
Северской клинической больницы
(ФГБУ СибФНКЦ ФМБА России)



 Д.Ю. Карымов
«30» сентября 2015 г.



Общество с ограниченной ответственностью

"Сибспецавтоматика"

634034 Россия г. Томск, ул. Нахимова, 18, тел./факс 900-474
ИНН 7017069540 КПП 701701001

Р/с 40702810174790000393 Восточно-Сибирский филиал ПАО «РОСБАНК»
к/с 30101810000000000388; БИК 040407388; ОКПО 14402067 ОКЭВД 45.31, 31.62.9

АКТ ВНЕДРЕНИЯ

**результатов диссертационной работы Друки Алексея Алексеевича «Алгоритмы
нейросетевого детектирования и распознавания символов на сложном фоне»**

Настоящим утверждается, что результаты диссертационной работы Друки А.А. «Алгоритмы нейросетевого детектирования и распознавания символов на сложном фоне», выполненной на соискание ученой степени кандидата технических наук по специальности «05.13.01 – Системный анализ, управление и обработка информации», являются актуальными, обладают практической ценностью и были внедрены в ООО «Сибспецавтоматика» для реализации коммерческих и технологических задач в области видеонаблюдения и обработки изображений.

Результаты диссертационной работы включают в себя программную реализацию следующих алгоритмов:

- 1) Алгоритм детектирования символов на изображениях со сложной фоновой структурой, основанный на сверточных нейронных сетях.
- 2) Алгоритм нормализации изображений символов, основанный на построении гистограмм распределения яркости пикселей.
- 3) Алгоритм распознавания символов на изображениях, основанный на сверточной нейронной сети.

Реализованные алгоритмы являются эффективными при решении соответствующих задач в области обработки изображений, классификации и видеонаблюдения.

Директор
ООО «Сибспецавтоматика»
29.09.2015


А.Н. Туляй

