

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Уральский государственный университет путей сообщения»  
(ФГБОУ ВО УрГУПС)

На правах рукописи

Дмитриев Никита Владимирович

**Повышение точности и производительности алгоритмов  
анализа и обработки сложноструктурных изображений**

Специальность 2.3.1

Системный анализ, управление и обработка информации, статистика

**ДИССЕРТАЦИЯ**

на соискание учёной степени  
кандидата технических наук

Научный руководитель:  
кандидат физико-математических наук  
Тарасян Владимир Сергеевич

Екатеринбург – 2024

## Оглавление

Введение.....	4
Глава 1. Современное состояние проблемы анализа и обработки сложноструктурных изображений.....	13
1.1 Классификация сложноструктурных изображений .....	13
1.2 Анализ теории обработки изображений.....	19
1.2.1 Концепции и типичные задачи обработки цифровых изображений .....	19
1.2.2 Концепция и методы искусственного интеллекта в решении задачи распознавания образов.....	25
1.3 Современные методы анализа и обработки цифровых изображений.....	31
1.3.1 Методы предварительной обработки ССИ.....	31
1.3.2 Методы распознавания образов.....	36
1.3.3 Анализ сцены и постобработка.....	40
1.3.4 Оценка алгоритмов.....	42
1.4 Выводы и постановка задач исследования .....	44
Глава 2. Разработка алгоритмов анализа и обработки сложноструктурных изображений.....	48
2.1 Методика анализа и обработки ССИ .....	48
2.2 Алгоритм многометочной сегментации .....	52
2.3 Локализация и определение типов сегментированных объектов.....	59
2.3.1 Алгоритм для локализации точечных и линейных объектов .....	60
2.3.2 Алгоритм для классификации площадных объектов .....	68
2.4 Распознавание образов и их группировка.....	73
2.4.1 Алгоритмы для точечных объектов.....	73
2.4.2 Алгоритмы для линейных объектов.....	83
2.5 Выводы.....	90
Глава 3. Экспериментальная проверка разработанных алгоритмов.....	92
3.1 Материалы и методика экспериментальной проверки разработанных алгоритмов.....	92

3.2 Экспериментальное исследование алгоритмов для обработки точечных объектов .....	97
3.3 Экспериментальное исследование алгоритмов для обработки площадных объектов .....	110
3.4 Выводы.....	117
Глава 4. Разработка и тестирование программного обеспечения автоматизированной обработки топографических карт .....	119
4.1 Особенности изображения топографических карт, как типичного представителя сложноструктурных изображений .....	119
4.2 Алгоритм комплексного анализа данных для ЦТК .....	127
4.3 Алгоритм интерполяции .....	134
4.4 Программный комплекс преобразования изображений топографических карт в цифровую модель рельефа .....	139
4.5 Проверка разработанных алгоритмов для ЦТК.....	142
4.5.1 Исследование алгоритма многометочной сегментации.....	145
4.5.2 Исследование алгоритма локализации сегментов .....	149
4.5.3 Исследование алгоритмов распознавания ТО.....	155
4.5.4 Исследование алгоритмов группировки ЛО .....	165
4.5.5 Исследование алгоритма комплексного анализа .....	168
4.5.6 Исследование алгоритма интерполяции .....	171
4.6 Выводы.....	173
Заключение .....	176
Список сокращений и обозначений .....	178
Список использованной литературы.....	179
Приложение А. Дополнительный материал .....	201
Приложение Б. Свидетельства о регистрации программ для ЭВМ.....	219
Приложение В. Копии актов о внедрении .....	221

## Введение

**Актуальность темы исследования и степень её разработанности.** В настоящее время происходит ускоренное развитие средств обработки и анализа информации вследствие накопления большого числа данных о различных объектах и процессах. Одними из важных составляющих таких средств являются программные комплексы анализа и обработки растровых сложноструктурных изображений, сочетающих в себе числовые и графические данные. Целью таких программных комплексов является получение информации о пространственно-распределённых объектах для дальнейшей обработки. Основными подходами при этом является векторизация и понимание семантических связей между объектами.

Примерами сложноструктурных изображений (ССИ) являются диаграммы, номограммы, чертежи, схемы, архитектурные проекты, карты местностей. В целом данные изображения можно охарактеризовать наличием многих семантических слоёв, наложенных друг на друга в визуальном плане. Например, топографические карты, которые можно назвать типичным представителем ССИ, содержат наиболее полную информацию о местности, поэтому из них можно получить информацию для геоинформационной системы (ГИС), которые можно использовать в качестве поддержки принятия решения на этапе проектирования для решения различных задач, например, оптимального размещения станций мобильной связи, логистических центров, транспортных развязок и дорог различного назначения. Также ГИС являются полезным инструментом анализа для осуществления, например, кадастровых работ, предсказания потенциальных зон пожаров и затоплений. Согласно отчету «*Geographic Information System (GIS) Market*», опубликованному *Allied Market Research*, общемировой рынок ГИС вырастет до 25,5 млрд. долл. к 2030 г., при среднегодовом темпе роста 15,2%.

При этом ручная обработка большого числа ССИ становится экономически неэффективной. Проблема распознавания и анализа образов приобрела выдающееся значение в условиях информационных перегрузок, когда человек не

справляется с пониманием поступающих к нему всё возрастающих объёмов данных. В результате его мозг переключается на режим одновременного восприятия и мышления, которому такое распознавание свойственно, но негативно влияет на качество работы в условиях нежелательности совершения ошибок. Поэтому современный период развития информационных технологий характеризуется широким внедрением интеллектуальных методов в процесс обработки графической информации, а автоматическая обработка и анализ изображений, обладающих сложной структурой, является одним из наиболее перспективных направлений развития.

Решению проблем обработки графических изображений, анализа и распознавания образов посвящены работы многих российских и зарубежных исследователей. Основы теории обработки цифровых изображений, алгоритмы оптической, геометрической и статической обработки, поиска особенностей изображений были исследованы в работах В.Ф. Кравченко, Н.Н. Красильникова, Ю.П. Пытьева, Я.А. Фурмана, Л.П. Ярославского, J. Canny, R. Gonsales, P. Hough, B. Jähne, D. Marr, T. Pavlidis, W. Pratt, J. Serra, P. Soille, R. Woods.

Вопросам машинного обучения и интеллектуальных методов, в т. ч. искусственных нейронных сетей посвящены труды следующих учёных: В.Н. Вапник, А.И. Галушкин, А.Н. Горбань, В.Д. Мазуров, М.Ю. Хачай, А.Я. Червоненкис, S. Haykin, Y. LeCun, T.M. Mitchel, M. Moller, J. Quinlan, F. Rosenblatt, S. Russe, L. Shapiro, G. Stockman, M. Stone.

Вопросам распознавания образов и компьютерного зрения внимание уделено в работах Ю.В. Визильтера, А.Л. Горелика, И.Б. Гуревича, С.Ю. Желтова, Ю.И. Журавлёва, В.В. Сергеева, В.А. Сойфера, С.М. Bishop, R. Duda, D. Forsyth, P. Hurt, E. Patrick, J. Tou, J. Ward.

Применительно к обработкам топографических карт и других ССИ можно выделить исследования С.А. Барталева, Д.Ю. Васина, Ю.Г. Васина, А.В. Дунаевой, В.Т. Калайды, Ю.Н. Кунакова, С.Ю. Мирошниченко, В.П. Пяткина, О.А. Славина, А.В. Никонорова, М.В. Ронкин, В.С. Титова, Р.А. Томаковой, Л.И. Черновой, R. Szendrey, H. Chen, J. Poudroux, S. Leyk, Y.-Y. Chiang, T. Liu.

Существующие программные решения, которые автоматизируют часть операций по обработке цифровых изображений, реализующих научные результаты вышеуказанных исследователей и собственные коммерческие разработки: ГИС «Панорама», *ArcGIS*, *ArcMap*, *Easy Trace*, *MapEdit*, *MapInfo*, *QGIS*, *SpotLight* и другие.

Недостатками существующих программных решений и научных работ является слабое внимание к вопросам точности, производительности и условий применимости создаваемых алгоритмов. И если ранее данная проблема не стояла так остро из-за слаборазвитого рынка компаний, занимающихся цифровой обработкой ССИ, то сейчас это приводит к необходимости закупки дорогого аппаратного обеспечения и формирования штата сотрудников, выполняющих монотонные операции по поиску и распознаванию образов с малой степенью автоматизации. Таким образом, без дальнейшей научной проработки экономический рост может быть затруднён, что свидетельствует об актуальности выбранной темы исследования.

**Целью работы** является повышение точности и производительности алгоритмов анализа и обработки сложноструктурных изображений с помощью объединения различных подходов обработки изображений.

**Задачи**, решение которых необходимо для достижения цели исследования.

1. Анализ современного состояния и развития подходов обработки ССИ с целью выработки методики построения программных систем анализа и обработки ССИ, с помощью которой можно повысить точность и производительность.

2. Разработка обобщённой методики анализа и обработки ССИ и разработка гибридных алгоритмов сегментации, локализации, распознавания и группировки образов различной размерности на ССИ, используя классические и интеллектуальные модели и методы, опираясь на критерии качества и быстродействия.

3. Экспериментальная проверка разработанных алгоритмов на различных классах ССИ с вычислением критериев качества, быстродействия и стабильности, сравнение полученных результатов с современными научными исследованиями.

4. Разработка специализированного алгоритмического и программного обеспечения для анализа и обработки цифровых топографических карт, как типичного

представителя ССИ, содержащего все виды объектов по размерности: точечные, линейные, площадные.

**Объектом исследования** являются методы и алгоритмы анализа и обработки растровых сложноструктурных изображений.

**Предметом исследования** являются показатели качества алгоритмов анализа и обработки сложноструктурных изображений, такие как точность и производительность.

**Методология и методы исследования.** Для решения поставленных в диссертационной работе задач были применены методы интеллектуальной обработки данных, в т. ч. обработки данных с использованием искусственных нейронных сетей, методы обработки изображений, в т. ч. методы математической морфологии, методы геометрической и семантической фильтрации, числовые методы линейной алгебры, методы оптимизации на основе генетических алгоритмов, методы математической статистики, теории графов, моделирования.

**Научная новизна** диссертационной работы определяется следующими новыми научными результатами.

1. На основе проведённого анализа современного состояния и развития подходов обработки цифровых изображений, предложена классификация сложноструктурных изображений, отличающихся использованием явных и неявных правил отображения распределённых образов различной размерности, а также методика построения алгоритмического и программного обеспечения анализа и обработки таких изображений (соответствует п. 7 паспорта специальности).

2. Разработаны алгоритмы сегментации, локализации, распознавания и группировки точечных, линейных и площадных объектов на сложноструктурных изображениях, основанные на комплексном использовании известных методов анализа и обработки изображений (соответствует пп. 4, 5 и 12 паспорта специальности).

3. Разработан алгоритм комплексного анализа распознанных числовых отметок высот и горизонталей цифровых топографических карт, отличающийся представлением контурной карты в виде орграфа и способный устранять ошибки

предшествующего этапа распознавания (соответствует пп. 4, 5, 12 паспорта специальности).

**Теоретическая значимость** диссертационного исследования заключается в разработке классификации ССИ и методики построения алгоритмического и программного обеспечения их анализа и обработки с помощью гибридного, итеративного и комплексного подхода.

**Практическая значимость работы** заключается в экспериментальной проверке разработанной методики обработки ССИ и набора алгоритмов в виде программных модулей. При этом выполняются следующие пункты.

1. Предложенная методика построения программных систем анализа и обработки сложноструктурных изображений позволяет производить обработку изображений с качеством  $200 \text{ dpi}$  с высокой точностью и производительностью, что приводит к снижению требований на аппаратное обеспечение персональных компьютеров и сокращению времени работы пользователей.

2. Разработанные алгоритмы сегментации, локализации, распознавания и группировки точечных объектов ССИ позволяют понизить ошибку в распознавании цифр относительно существующих методов и понизить количество вычислений за счёт сокращения рассматриваемых пикселей.

3. Разработанные алгоритмы локализации, распознавания и группировки линейных объектов ССИ позволяют понизить количество разрывов и повысить производительность вычислений относительно существующих методов.

4. Разработанные алгоритмы сегментации и распознавания площадных объектов ССИ позволяют использовать более простые интеллектуальные модели с последующей обработкой относительно сложных, что понижает требования к аппаратному обеспечению.

5. Предложенный алгоритм комплексного анализа распознанных точечных и линейных объектов цифровой топографической карты позволяет получить пропущенные данные и исправить неправильно распознанные отметки относительно



других существующих методов и программных решений, в которых данные операции выполняются пользователем.

Научные и практические результаты диссертационной работы были реализованы при выполнении научного проекта РФФИ № 17-08-01123 А «Математическое моделирование транспортных сетей в условиях развития высокоскоростного наземного транспорта» и в рамках госбюджетной научно-исследовательской работы «МХТ-8» по теме «Интеллектуальное управление и анализ данных в мехатронных и робототехнических системах».

### **Основные положения, выносимые на защиту:**

1. Предложенная методика построения программных систем анализа и обработки ССИ формализует данный процесс благодаря разделению обрабатываемых объектов по размерностям, а процесса – по этапам, включая этапы внутри- и межсемантического анализа, что позволяет унифицировать проектирование алгоритмического и программного обеспечения для различных классов ССИ, учитывая их особенности.

2. Набор разработанных алгоритмов, в т. ч. этапов сегментации, локализации, распознавания и группировки объектов ССИ различной размерности, позволяет повысить быстродействие и точность указанных этапов обработки за счёт оптимизации получения параметров алгоритмов, последовательностей операций и гиперпараметров интеллектуальных моделей, относительно существующих подходов, в которых данные параметры и модели подбираются вручную.

3. Разработанное программное обеспечение для анализа и обработки ССИ подтверждает применимость предложенной методики и набора алгоритмов на примере смоделированных навигационных морских карт, спутниковых снимков и сканированных изображений топографических карт в разрешении 200 dpi, что на более детальном разборе последних позволяет сократить время автоматической обработки одной карты с целью получения ЦМР до ~10 минут, при необходимой ручной обработке до 20..60 минут с полностью верным восстановлением значений высот горизонталей, по сравнению с использованием существующих программных

средств, где обработка одной топографической карты может занимать до одного рабочего дня с возможностью ошибки, вызванной ручным трудом.

**Степень достоверности** полученных в диссертационной работе результатов и выводов обеспечивается логической обоснованностью используемых подходов к анализу и обработке изображений, их грамотном сочетании, обеспечивающих повышение точности и производительности; подтверждается воспроизводимостью и непротиворечивости данных, полученных в ходе вычислительных экспериментов, получением свидетельств о государственной регистрации программ для ЭВМ, внедрением полученных результатов, апробацией на международных конференциях, наличием публикаций основных результатов работ в рецензируемых российских и зарубежных изданиях.

**Внедрение результатов диссертационного исследования.** Результаты диссертационного исследования прошли апробацию и были внедрены в деятельности АО «Урало-Сибирской Гео-Информационной компании», г. Екатеринбург (акт о внедрении результатов от 23 декабря 2019 г.). Результаты исследования внедрены в учебный процесс и научно-исследовательскую работу Уральского государственного университета путей сообщения при проведении учебных занятий с бакалаврами по направлениям «Мехатроника и робототехника», «Информатика и вычислительная техника» (акт об использовании результатов от 1 декабря 2023 г.). Соответствующие акты прилагаются к диссертации.

**Апробация работы.** Основные результаты диссертации докладывались и обсуждались на следующих конференциях и семинарах: «Математические методы решения исследовательских задач» (г. Екатеринбург, 2014 г., 2015 г.), «Информационные технологии и нанотехнологии» (г. Самара, СГАУ, 2016 г.), «Математическое моделирование и информационные технологии» (г. Екатеринбург, 2016 г.), «Управление и обработка информации в сложных системах» (г. Екатеринбург, 2017 г.), выставка «ИННОПРОМ» (г. Екатеринбург, 2017 г.), «Applied System Analysis and Computer Science» (г. Екатеринбург, УрФУ, 2018 г.), «Научный семинар по информационным технологиям» (г. Челябинск, НИУ ЮУрГУ, 2018) «Современные

проблемы математики и ее приложений» (г. Екатеринбург, ИММ УрО РАН, 2019 г.), «Applications of Mathematics in Engineering and Economics» (г. Созополь, Болгария, 2019 г.), «Инновационные производственные технологии и ресурсосберегающая энергетика» (г. Омск, ОмГУПС, 2021), «TLC2M Transport: Logistics, Construction, Maintenance, Management» (г. Екатеринбург, 2022).

**Публикации.** По теме диссертации опубликовано 15 работ [29-32, 34, 36-41, 45-47, 150], из них 5 статей опубликованы в рецензируемых научных изданиях, определенных ВАК РФ и Аттестационным советом УрФУ [32, 38, 41, 47], включая 1 статью в издании, индексируемом в международной цитатно-аналитической базе Scopus [150]. Получены два свидетельства о государственной регистрации программ для ЭВМ [43, 44].

**Личный вклад.** Выносимые на защиту результаты получены автором диссертации лично и отражены в работах: [36, 43] – автором проанализирована и разработана концепция преобразования изображения топографических карт в объёмную модель местности, спроектирован прототип программного комплекса; [30–32, 38, 45–47] – автором произведена разработка и экспериментальная проверка алгоритмов сегментации, поиска, распознавания и группировки объектов на изображениях, алгоритма восстановления плана горизонталей и интерполяции. Работы [29, 31, 32, 34, 36-39, 45–47] подготовлены в соавторстве с научным руководителем, которому принадлежит постановка задачи исследования цифровых топографических карт. Работы [30, 40, 41, 44, 150] подготовлены диссертантом самостоятельно, все основные положения тезисов докладов конференций также сформулированы лично автором.

**Структура и объём диссертации.** Диссертация состоит из введения, четырёх глав, заключения, библиографии и трёх приложений. Общий объём диссертации 222 страницы, включая 35 рисунков и 40 таблиц. Библиография включает 275 наименований.

**В первой главе** был проведён анализ современного состояния проблемы анализа и преобразования цифровых изображений, представлена классификация ССИ,

рассмотрены оптические и интеллектуальные алгоритмы анализа и преобразования цифровых изображений, описана проблематика используемых подходов.

Во **второй главе** предлагается решение поставленных задач с помощью предложенной методики анализа и обработки ССИ и разработанных алгоритмов – сегментации, локализации, распознавания и группировки точечных, линейных и площадных объектов.

В **третьей главе** описываются проведённые экспериментальные исследования разработанных алгоритмов для ССИ двух классов, в течение которых с помощью настройки параметров алгоритмов и гиперпараметров интеллектуальных моделей оптимизируется качество и быстродействие.

В **четвёртой главе** описывается пример использования разработанной методики для случая анализа и обработки цифровых топографических карт, производится проектирование и тестирование программного обеспечения, реализующего предложенные алгоритмы.

## **Глава 1. Современное состояние проблемы анализа и обработки сложноструктурных изображений**

В данной главе описано современное состояние проблемы анализа и обработки ССИ. Усовершенствована классификация изображений по категории отображаемых объектов и консолидированы признаки ССИ. Принципы обработки цифровых изображений, интеллектуальные системы, методы машинного обучения рассмотрены в контексте обработки ССИ. Проведён анализ современных методов предварительной обработки изображений, распознавания образов, понимания сцен и способов оценки этих методов, на основании которого были выделены их основные достоинства и недостатки, указывающие на ограниченность их применения или необходимость модернизации для работы со ССИ.

### **1.1 Классификация сложноструктурных изображений**

Структурное изображение (как частное от термина «структурные данные») является изображением, имеющим внутреннюю организацию или метаинформацию об отображаемых объектах. Сложность этой организации, выраженную в виде большого числа зависимостей и разнообразном проявлении объектов, приводит к сложности восприятия, анализа и обработки. Так как векторные изображения и элементы на них имеют известное математическое описание и свойства, то их программная обработка может быть выполнена автоматически с использованием формальных методов. В отличие от них, растровые изображения не хранят напрямую всех характеристик объектов, их анализ и обработка являются актуальной научной задачей, и поэтому именно они рассмотрены на настоящей работе.

Понятие «сложноструктурное изображение» в настоящий момент является зонтичным термином, включающим в себя несколько разнородных определений. В [8] под изображением со сложной структурой понимается «изображение, состоящее из ряда пространственно организованных статистически однородных

фрагментов различных классов и детерминированных изображений, расположенных на их фоне». В [14] выделены следующие признаки: сочетание графических (в т. ч. различной размерности) и текстовых элементов, большеформатность, зависимость атрибутов объектов от местоположения.

Классификация изображений, представленная в [73], отталкивается от подхода к представлению цифровых изображений, но не отражению в них структуры, поэтому её нельзя использовать для классификации ССИ, хотя в данной работе при обработке будут получаться изображения указанных классов (цветные изображения, двухуровневые изображения, кривые и линии, точки и многоугольники).

В настоящее время существует эмпирическая классификация цифровых изображений по категории отображаемых объектов [15], которая является более близкой к теме исследования. Проблема данной классификации состоит в том, что она не основана на теоретических концепциях и не предлагает практических методов для решения задач анализа и обработки, является достаточно обобщённой. Поэтому предлагается усовершенствовать и уточнить данную классификацию.

Цифровое изображение воспринимается в двух планах: визуальном и семантическом. *Визуальный план* содержит в себе растровое описание объектов в одном из цветовых пространств. *Семантический план* содержит в себе информацию об этих объектах. Если на изображении существует несколько различных по природе или функциям объектов, то на нём будет существовать несколько семантических планов; разделение же визуальных планов возможно только при существовании возможности разделения в цветовом пространстве, в общем случае можно говорить о гетерархичности визуальных планов. Под гетерархичностью здесь понимается возможность пересечения визуальных объектов и/или невозможность выделения для них пространственных отношений «снаружи», «внутри», «рядом».

Соответствие между семантическими и визуальными планами определяется с помощью правил отображения, которые являются метаинформацией. Нарушение этих правил или их нерегулярность также является признаком сложноструктурности [93]. Нарушение правил, тем не менее, не должно быть слишком частым или

значительным: для ССИ всё-таки предполагается возможность их анализа и обработки.

Также одним из признаков сложноструктурности можно выделить высокую информационную ёмкость. Данную ёмкость можно посчитать, если векторизовать исходное растровое изображение и разделить размер получившегося файла на единицу площади исходного изображения. Сравнить информационные ёмкости можно только при одинаковом разрешении (*dpi*) исходных растровых файлов.

Можно выделить следующие классы изображений: отражающие реальность (класс I), созданные человеком рукотворно (класс II), созданные человеком автоматизированно (класс III). Классы далее представлены вместе с современными научными исследованиями. Данная классификация не рассматривает простейшие изображения (однородные фоны, градиенты, текстуры) и фрактальную графику, так как данные изображения имеют скорее векторную, чем растровую природу. Выведенные подклассы расположены от имеющих более общее применение к более узкоспециализированным: такой характер имеют и методы, используемые для их обработки.

Ia. Проекция общего назначения: персональные фотографии, фотографии с уличных камер, фотографии дорожного движения, аэрофотоснимки [3, 6, 9, 50, 59, 83, 103]. Получаются с помощью восприятия реального мира с определённой точки зрения в видимом излучении или в других спектрах. Сложность таких изображений в общем случае эквивалентна сложности воспринимаемого мира, например, сложности фона относительно распознаваемых объектов [89], и часто является неявной. По ним мы можем осуществить лишь поверхностный анализ, в общем случае семантические поля не воспринимаемы по визуальным, внесение даже незначительных изменений может поменять картину восприятия [252]. Типичные задачи: поиск и распознавание образов с помощью различных интеллектуальных моделей.

Ib. Проекция специального назначения: сканы поверхности керны, рентгенограммы [92], МРТ, гистологические срезы. Получаются аналогично Ia, но нацелены на объекты определённой физической природы, используя специальное излучение,

часто в строгих лабораторных условиях. Обычно используются полутонные изображения. Для таких изображений предполагается полное знание правил отображения: несоответствие правил реальному положению вещей скорее говорит о необходимости корректировки научного аппарата. Сложность определяется большим количеством объектов одного типа, которые могут быть крайне нерегулярными. Нерегулярность приводит к вариативности в восстановлении структуры, поэтому таким изображениям было дано название «сложноструктурируемых» [91]. Отличие понятия ССИ заключается в том, что для них предполагается существование определённой структуры, которую можно получить, используя детерминированные методы. В отличие от класса Ia, объекты не могут быть расположены поверх друг друга. Типичные задачи: поиск и распознавание образов и текстур, кластеризация с помощью искусственного интеллекта и геометрической обработки.

Is. Матрицы прямого отображение физических величин: спектрограммы [275], тепловые карты [204], цифровые матрицы рельефа по спутниковым данным. Сложности понимания не представляют, так как визуальный план напрямую связан с семантическим. Применяются математические и статистические методы.

IIa. Объекты изобразительного искусства: живопись, графика. Сложность таких изображений ограничена творческими способностями автора. Осмысление визуального плана возможно только при понимании авторской задумки; решение данной задачи в автоматическом режиме не видится без существования общего искусственного интеллекта, в настоящее время может быть осуществлено только человеком. Сложность восприятия образов здесь наблюдается вместе с небольшим их количеством, что также препятствует обоснованности автоматизации.

IIb1. Слабо формализованные документы, выполненные разными инструментами для письма: старые рукописные карты, планы и схемы. Их анализ возникает при необходимости выяснения особенностей местности с исторической точки зрения. Примерами являются анализ изменения лесного покрова со временем [109, 120, 172, 218], анализа городской застройки и земельного использования [128, 177, 178, 180, 222], получения текстовой информации [134, 136, 169], сравнение



географических особенностей [153, 239]. Сложность подразумевает отсутствие строгих правил изображения, нерегулярность человеческого письма и полноцветность. Необходимо осуществлять предварительную обработку, так как исторические карты могут иметь достаточное количество дефектов и искажений. Из-за большой плотности информации обработка исторических карт в настоящее время остаётся прерогативой оператора ЭВМ, существующие программные решения могут лишь в малой доле автоматизировать некоторые действия с помощью простейшей статической и геометрической обработки и использования моделей искусственного интеллекта.

Пб2. Слабо формализованные документы, выполненные одним инструментом для письма: рукописные карты и скетчи [124, 139], дневники, расписки, фотографии формул. Отличаются от класса Пб меньшим количеством семантических полей и ограниченным числом используемых цветов (полутоновое и дискретное цветовое пространство), при возможно большем количестве исследуемого материала. Тем не менее, тут в полной мере может быть использован аппарат глубокого обучения и свёрточных нейронных сетей, так как интеллектуальную систему можно научить воспринимать конкретный почерк.

Пс. Сильно формализованные документы и заполненные формы: кодовые штампы письма, чертежи от руки, ноты. Сложность подразумевает наличие большого количества пересечений, в том числе с печатными элементами. Исходя из этого, к задаче распознавания образов добавляется задача их поиска и отделения друг от друга. Количество нерегулярных объектов от руки сводится к минимуму, существуют известные правила и стандарты нанесения элементов изображения, что позволяет использовать геометрическую обработку и оптическое распознавание символов.

Ша. Объекты изобразительного искусства: сгенерированная графика, коллажи. Описание пункта аналогично классу Па с дополнением, что если сгенерированная графика не использует случайных чисел при сопоставлении семантического плана визуальному, то она может быть обработана различными методами, как

формальными, так и интеллектуальными. Примером такой графики могут выступать изображения экрана в видеоиграх [188], распознавание и анализ которых для построения интеллектуальных ботов-соперников начинает становиться одной из актуальных тем.

Шв. Слабо формализованные документы: топографические карты. Данная задача покрывает довольно широкую область применения, в отличие от обработки исторических карт. В данной проблеме можно объединить обработку геологических, топографических и прочих карт. В современной научной литературе можно выделить американскую [132-136, 153, 193, 257-258] и шанхайскую [197-200, 206-207, 268] школы, и отдельных региональных исследователей-специалистов [37, 39, 67, 90, 91, 119, 155, 157, 161, 166, 167, 179, 187, 212, 219, 221, 227, 238, 259], которые занимаются всеми вопросами анализа и обработки карт. Наличие такого числа исследователей объясняется региональными особенностями и стандартами изображения карт, что приводит к тому, что зачастую исследования одного учёного неприменимы в контексте другого региона без дополнительной доработки и модификации. Для решения задач анализа и обработки таких изображений необходимо использовать весь спектр геометрических и интеллектуальных методов, в т. ч. с разработкой гибридных и комплексных методов анализа, так как присутствует очень сильное пересечение визуальных планов, которые тем не менее можно разделить, и зависимость семантических планов между собой.

Шс. Сильно формализованные документы: чертежи, тематические карты и планы, сканы книг, диаграммы состояния сплавов, номограммы водяного пара, энтальпийные диаграммы, схемы. Данная задача возникает при необходимости преобразования электрических схем в форму, удобную для симуляции [112, 141, 142], векторизации сложных инженерных [158, 211] или логических [140] схем, распознавание элементов трубопроводных систем [182, 272]. Для решения данных задач необходимо уметь находить определённые объекты, имеющие заранее известную форму и способ изображения, и прямые соединительные линии часто строго горизонтальные или вертикальные, но имеется значительное число пересечений.

Данные документы зачастую являются чёрно-белыми изображениями с отсутствием семантического перекрытия либо вариант перекрытия строго регламентирован, что не оставляет возможности для двойного прочтения, например, как в случае с пересечением соединительных проводов на электрических схемах. Используется геометрический анализ: вычисление углов, измерения, построение иерархической структуры объектов, если необходимо.

В итоге, ССИ можно определить следующим образом.

*Сложноструктурное изображение* – растровое изображение, которое содержит объекты, относящиеся к различным семантическим планам, и которое обладает набором явных и неявных правил отображения этих объектов на визуальный план с помощью различных техник в виде распределённых образов, подразумевающих возможность однозначного восстановления смысла исходных объектов и их атрибутов.

Неявные правила включают в себя не только недетерминированные, но и описывающие [56] «неполную, разнородную, косвенную, нечёткую, неоднозначную, вероятностную» информацию.

## **1.2 Анализ теории обработки изображений**

### **1.2.1 Концепции и типичные задачи обработки цифровых изображений**

Проблема обработки изображений непосредственно связана с несколькими областями исследований. С одной стороны, это изучение человеческого зрения, как физиологического процесса, и восприятие цвета, форм, текстур и образов, как психологического. С другой стороны, это разработка математического и алгоритмического аппаратов для имитации данных форм человеческой деятельности, создание моделей изображений, как физических объектов материального мира. Общие вопросы компьютерного зрения рассматриваются в [87, 100, 189].

Такие научные области, как компьютерное зрение, машинное зрение и

обработка изображений, тесно связаны между собой. Обработка изображений подразумевает переработку и анализ цифровых изображений без знаний о содержании таких изображений. Для компьютерного зрения некоторые знания уже являются необходимыми, так как происходит процесс не только обработки, но и понимания того, что происходит на изображении. Машинное зрение, применяемое на производстве, полностью связано с изображениями определённого типа, например, робот-манипулятор отслеживает перемещение конкретно выглядящих деталей по конвейеру. Поэтому в задаче распознавания ССИ необходимо пользоваться методами обработки изображений на начальной стадии, когда мы ещё не пользуемся специальными знаниями, а потом переходить в область компьютерного зрения.

Общие вопросы и открытые проблемы компьютерного зрения и обработки изображений рассмотрены в [154, 156]. Так, последовательность этапов обработки является следующей: предобработка, редукция данных, сегментация, распознавания образов и анализ сцен; все этапы должны сопровождаться оптимизацией. Соответственно, проблемами являются получение репрезентативной выборки изображений для машинного обучения, выбор образов для распознавания, адаптация алгоритмов, обобщение и оценка качества выполненных алгоритмов.

Цифровая обработка изображений рассмотрена в [25, 53, 80, 84, 88, 98, 102, 105, 107, 108]. Любое ССИ является непрерывным на плоскости своего отображения, причём цветовые ощущения можно описать с помощью набора цветовых координат. Обычно рассматривают цветовое пространство *RGB*, состоящее из трёх компонент: красного, зелёного и синего цветов. Таким образом, каждый цвет понимается, как вектор в цветовом пространстве.

Для обработки изображения на цифровом устройстве необходимо перевести изображение в цифровой формат, то есть дискретизировать его, что осуществляется с помощью сканирования. При этом происходит, во-первых, квантование пространства, соответствующее разрешению сканирующего устройства (*dpi*), а во-вторых, квантование цветовых координат, соответствующее глубине цвета (обычно используется *TrueColor*, кодирующее цвет по 8 бит на каждую цветовую

компоненту). В итоге сканирования получается двумерная матрица (растр), элементами которой являются цветовые вектора отдельных точек изображения – пикселей.

Полученное цифровое изображение ещё непригодно для такого сложного анализа, как распознавание образов, так как обладает многочисленными искажениями [91, 133], вызванными как физическими изменениями в документе (появление потёртостей, пятен), так и сканированием (например, шумов от фотодетектора, «скачков» цветовых векторов между соседними пикселями из-за дискретизации, засвеченности и т. д.). Чтобы избавиться от этих искажений, необходимо провести улучшение и реставрацию цифрового изображения.

Предобработка осуществляется без участия человека, так как в этом случае ещё не выполнена редукция данных и происходит обработка довольно больших массивов. В основе операций, улучшающих изображение – попиксельные преобразования, например, усиление контрастности, повышение резкости или видоизменение гистограммы за счёт её выравнивания. Также сюда относятся преобразования цветовых пространств, в том числе преобразование изображения в бинарное с помощью пороговой обработки.

Для подавления шумов можно воспользоваться тем, что они не обладают пространственной корреляцией между собой, поэтому могут быть «удалены» за счёт использования статистических фильтров, учитывающих значения цветовых векторов пикселей-соседей, например, усредняющий или медианный фильтр. Также применяются высоко- и низкочастотные фильтры, преобразования Фурье, Уолша, Хаара. Такие фильтры имеют свой недостаток, так как «размывают» зашумлённые области, что недопустимо при обработке изображений с низким разрешением; за шум могут быть восприняты небольшие образы, например, точки в отображении штрихпунктирной линии.

После улучшения изображения его необходимо кластеризовать, то есть разделить все пиксели по кластерам примерно одного цвета, так как обычно правила отображения строятся так, что объекты, принадлежащие к одному семантическому

полю, характеризуются одним цветом в визуальном поле. Самая простейшая кластеризация состоит в переносе векторов всех пикселей карты в одно цветовое пространство и выделение наиболее плотных областей в отдельный кластер. Таким образом, всё многообразие цветов заменяется функцией принадлежности пикселя к определённому кластеру, что означает редукцию по глубине цвета.

Множества пикселей, разбитые по кластерам, ещё не являются образами, это лишь объединения пикселей примерно одного цвета. Образы одного цвета могут отличаться по форме или текстуре, например, реки и озёра изображаются с помощью голубого цвета, но являются образами разных размерностей. Поэтому необходимо провести сегментацию: разбить пиксели, принадлежащие одному кластеру, на локально непрерывные подмножества, обладающие некоторым свойством, например, равенством яркости, отображения контура, внутренней текстуры, формы. Такие сегменты можно называть «образами», так как у них уже можно выделить такие атрибуты, как положение на карте, угол наклона, длина или площадь, но никаких информативных атрибутов (функция, значение) у них ещё нет.

Множества пикселей одного кластера и сегменты могут быть преобразованы с помощью аппарата математической морфологии [18, 81, 241, 247], чтобы нивелировать проявления неисправленных искажений для устранения разрывов и связывания или для разделения сегментов. Аппарат математической морфологии может быть использован для разделения сегментов по своим классам.

Основные результаты по распознаванию образов на цифровых изображениях изложены в [13, 27, 48, 52, 57, 76, 94, 96, 99, 126, 174]. Концепцией распознавания образов является получение соответствия между образом и объектом за счёт анализа дескрипторов образа. Каждый объект также обладает своими дескрипторами, но не все они могут напрямую сравниваться с дескрипторами образов. Нельзя, например, говорить о длине единицы, но можно говорить о длине единицы на конкретном изображении. Таким образом, объект должен быть представлен в виде эталонного образа или совокупности образов. Кроме характерных признаков (форма), некоторые его свойства будут напрямую зависеть от свойств всего изображения в

целом (цветность, разрешение). Поэтому образы перед распознаванием следует нормализовать, например, привести символы к одинаковому размеру.

Существует два основных подхода к распознаванию – оптический и интеллектуальный. Оптический подход более прост в реализации, обладает высокой скоростью выполнения на ЭВМ. Но в отличие от него интеллектуальные методы более точны, обладают более высокими показателями качества.

Оптические методы распознавания рассмотрены в [7, 256], среди них выделяется три подхода: шаблонный, структурный и признаковый. Метод совпадения шаблона основан на непосредственном сравнении изображения с известными шаблонами; такой метод работает хорошо и быстро на чистых изображениях, но показывает плохие результаты при отличии формы изображения, например, при использовании других шрифтов при распознавании символов текста.

Структурные методы используют информацию о топологии изображения, таких как пересечения линий и наличие отверстий. Такой подход более гибок по отношению к изменению шрифта, размера букв или повороте на некоторый угол, но когда элементы изображения могут пересекаться или разрываться, то качество может значительно ухудшиться.

Лучшим оптическим подходом для зашумлённых изображений является признаковый, к нему относятся такие методы, как зонирование, использование инвариантов геометрических моментов, проекционных гистограмм и т. д. В нём изображению ставится в соответствие набор признаков, вычисленных согласно найденному образу. Далее будут рассмотрены методы обработки этих признаков.

Простейшим методом является классификация с помощью расстояния: если воспринимать образ, как вектор дескрипторов в признаковом пространстве, то его принадлежность к некоторому классу может быть определена с помощью расстояния между вектором образа и вектором эталона. Такой эвристический подход работает, если классы обнаруживают тенденцию к кластеризации. Если функция вероятности отклонения образов от эталонов не зависит только от расстояния до эталона в признаковом пространстве, то применяется классификация с помощью

функции правдоподобия или байесовский классификатор.

Так как выделить конкретный эталон или группу эталонов часто оказывается невозможным из-за высокой вариативности атрибутов или высокой зашумлённости, вызывающей сильные искажения атрибутов, приходится принимать за дескрипторы значения цветовых координат пикселей в их связях. Для таких случаев очень хорошие результаты показывают нейронные сети различных топологий [55, 203], для которых с помощью машинного обучения можно в автоматическом режиме отрегулировать внутренние параметры. В основе анализа лежит принцип скользящего окна: анализ проводится не над конкретным образом, определённым своим положением в растре, а над его соседями тоже, «перемещаясь» от одного к другому, пока не будет найден образ, выдающий максимальное значение для какого-либо класса. Данный принцип может значительно ускорить распознавание при использовании многоядерной или многопроцессорной архитектуры ЭВМ.

Особое место также занимает гибридный подход, заключающийся в объединении оптического и интеллектуального подходов [5, 86] или нескольких искусственных нейронных сетей [4, 72, 92], что приводит к повышению качества распознавания и скорости выполнения алгоритмов. Классификаторы объединяются в комитеты посредством бустинга [164, 205, 240], основная идея которого заключается в том, что несколько слабых классификаторов могут проявлять себя сильнее в комплексе.

Оптимизировать классификаторы можно с помощью генетических алгоритмов или алгоритма имитации отжига [85, 249]. В них перебираются правила взаимодействия дескрипторов, пока не будет найден оптимальный набор правил, дающий максимальный функционал качества. Также улучшения (как в качестве, так и в скорости) можно достичь с помощью уменьшения числа дескрипторов [122], так как некоторые из них сами могут являться шумами.



## 1.2.2 Концепция и методы искусственного интеллекта в решении задачи распознавания образов

Концепция искусственного интеллекта (ИИ) в целом рассмотрена в [62, 230]. Интеллектуальные информационные системы и их базы знаний рассмотрены, соответственно, в [63] и [21]. В общем, применение ИИ ставит перед собой проблему автоматизации выполнения интеллектуальной работы человека посредством имитации сложных психических и нейрофизиологических процессов. Интеллектуальная система должна обладать способностями:

- к восприятию информации;
- к обучению на основе опыта;
- к адаптации в новых ситуациях;
- к пониманию и применению абстрактных концепций и знаний.

Для решения задачи распознавания образов эти способности соответствуют процессам считывания информации с ССИ, поиска и распознавания образов, стабильности распознавания при возникновении образов похожих оттенков и форм, использования базы знаний образов. Без использования интеллектуализации алгоритмы становятся восприимчивыми к различным шумам и искажениям в данных, что особенно влияет при обработке изображений.

Теория ИИ использует различные методы исследований, такие как моделирование рассуждений, обработка естественного языка, инженерия знаний, машинное обучение. Все они используются для распознавания изображений.

В методы моделирования рассуждений входит теория принятия решений, использующаяся, когда необходимо на основе некоторых данных и модели правил определить принять решение относительно поставленной проблемы. Такой проблемой при распознавании может быть согласованность данных, например, на некотором расстоянии друг от друга стоят отметки высот, которые распознал алгоритм. Если вычисленный угол наклона будет больше критического значения, значит, что какое-то число или оба были распознаны неверно. Или алгоритм распознавания

образов может выдавать не только результат, но и уверенность в нём, и если степень уверенности низкая, то эта информация также может быть передана оператору для ручного анализа.

Обработка естественного языка нужна при распознавании названий географических объектов, что приведёт к возможности объединения двух образов общей природы через проверку равенства их топонимов, как атрибутивных данных. Инженерия знаний позволяет перенести знания об объекте исследования с интуитивно понимаемых человеком и обрабатываемых автоматически, вне сознания, на формальный язык ЭВМ.

Для настройки параметров интеллектуальных моделей используется машинное обучение, концепции которого рассмотрены в [118, 208, 201, 224, 226]. Существуют различные способы машинного обучения в зависимости от того, что взято за его основу, и какой класса задач необходимо решить. В общем случае входные данные для машинного обучения представляют собой упорядоченное множество  $n$  значений, объединяемые в вектор  $\mathbf{x} = (x_1, \dots, x_n)$ .

Для «обучения с учителем» в алгоритм вводят набор пар значений «ситуация – требуемое решение». Типичной задачей для такого способа является распознавание символов, так как для некоторого множества пикселей исследователь задаёт значение символа, отображённого на этом множестве. В общем случае эта задача называется классификацией. Различают бинарную и многоклассовую классификацию; классы могут быть непересекающимися, пересекающимися и нечёткими.

Для «обучения без учителя» в алгоритм вводят только набор «ситуаций». В задаче анализа изображений такой способ применяется для его кластеризации, то есть разделения множества элементов, обладающих некоторыми признаками, на подмножества в признаковом пространстве на некоторые однородные группы [74]. Для ССИ кластеризация, соответственно, выполняется для разделения цветов пикселей по группам. Данным способом можно понизить размерность и вариативность данных, так как миллионы возможных цветовых оттенков заменяются кластерами небольшого количества цветов. Проблемой кластеризации является необходимость

знания количества кластеров.

Для измерения согласованности вычисленных решений реальности вводится понятие функционала качества алгоритма. Для «обучения с учителем» функционалом является средняя ошибка несоответствия. Если пытаться минимизировать среднюю ошибку по обучающей выборке данных, то возможно получение эффекта «переобучения», когда система становится неспособной к предсказыванию реакций на новые ситуации. Для «обучения без учителя» функционалом может являться, например, среднее внутрикластерное или межкластерное расстояния (для цветов это может быть декартово расстояние между точками в цветовом пространстве центров масс кластеров).

Чтобы избежать переобучения используется кросс-валидация [251]. При кросс-валидации обучающая выборка разделяется на  $k$  частей, при этом обучение проводится на  $k-1$  части, а тестирование – на оставшейся. При этом возникает  $k$  независимых моделей, результат действия которых находят по самому встречающемуся классу.

Одной из актуальных задач машинного обучения является так называемое «проклятье размерности» [224], когда при большом количестве входных данных становится невозможным получение решающей модели с достаточной точностью. Чтобы решить эту проблему необходимо, во-первых, понизить размерность исходных данных (через кластеризацию или отбрасывание малоинформативных признаков), а во-вторых, по возможности увеличить обучающую выборку.

При наличии группы эталонов (особенно близко расположенных) часто возникает ситуация, когда образ принадлежит сразу нескольким классам. Для таких случаев применяются следующие универсальные методы: поиск ближайших  $k$  соседей, метод опорных векторов, метод решающих деревьев [20, 66].

Метод ближайших  $k$  соседей [115] подразумевает присвоение класса для заданного вектора  $x$  наиболее распространённому среди  $k$  ближайших соседей, классы которых уже определены. При этом важным вопросом является определение метрики близости и предобработка значений атрибутов. В качестве метрики

близости обычно используется одна из метрик Минковского порядка  $p$ . Значения атрибутов могут быть нормализованы с помощью минимакс- или  $Z$ -нормализации, значимым атрибутам могут быть заданы определённые веса. Данный метод позволяет легко произвести классификацию, если объекты одного класса имеют близкие значения своих атрибутов. Недостатком данного метода является необходимость больших вычислений при большом значении числа атрибутов  $n$  и количестве существующих эталонов  $m$ : вычислительная сложность такого метода равна  $O(nmk)$ .

В методе опорных векторов (*SVM*, *support vector machine*) производится перевод векторов в пространство более высокой размерности и поиск разделяющих гиперплоскостей с максимальным зазором в этом пространстве. Если обучающая выборка линейно разделима, то переход в пространство большей размерности не является необходимым и можно использовать алгоритм линейной классификации В. Н. Вапника и А. Я. Червоненкиса [13, 121], в которой используется скалярное произведение искомого вектора и опорных векторов (эталонных). Иначе необходимо использовать так называемые ядра свёртки [121]. Метод ориентирован для решения задачи бинарной классификации; в случае нескольких классов необходимо осуществлять сортировку по расстоянию до разделяющей гиперплоскости. Вычислительная сложность равна  $\Theta(nm^3)$  [110], таким образом, для *SVM* важно отсеять эталоны, которые не помогают классификации.

Метод решающих деревьев [123, 226] основывается на представлении принятия решения в виде дерева, на узлах ветвления которого расположены атрибуты, а на ребрах – интервалы значения атрибутов. Движение начинается с корневого узла и переходит по рёбрам, соответствующих значениям атрибутов вектора  $x$ , пока не дойдёт до терминальных узлов, соответствующих классам. Вычислительная сложность равна  $O(h)$ , где  $h$  – высота дерева. Проблемой является построение дерева, так как даже если оно является бинарным, то количество рёбер, а соответственно и сложность по памяти соответствует  $O(2^h)$ , что приводит к задаче регулирования глубины дерева.

Одной из самых хорошо изученных моделей в машинном обучении и теории

распознавания образов в частности, решающей и задачу кластеризации, и задачу классификации, являются искусственные нейронные сети (ИНС), рассмотренные в [10, 22, 26, 64, 72, 82, 165, 175, 229].

В основе ИНС лежит принцип коннективизма – относительно простые элементы, искусственные нейроны, в совокупности могут решать довольно сложные интеллектуальные задачи, такие как распознавание образов. Простейшей моделью искусственного нейрона является модель МакКаллока-Питса, в которой нейрон является вычислителем  $n$ -арной булевой функции:

$$\alpha(x) = \varphi \left( \sum_{j=1}^n (\omega_j * x_j) - \omega_0 \right), \quad (1.1)$$

где  $\varphi(z) = [z \geq 0]$  – ступенчатая функция Хевисайда,  $\omega_j$  – вес для  $j$ -той координаты вектора  $x$ .

Функцией активации может быть также сигмоидальная функция или любая другая (например, нелинейная, как в случае радиальных нейронов), выполняющая роль определения, насколько данное состояние (вектор  $x$ ) соответствует искомой реакции.

Обучение (то есть корректировка весов) происходит по минимизации целевой функции (функционала качества)  $E$ , которая обычно принимает значение:

$$E = \frac{1}{2} * (y_i - d_i)^2, \quad (1.2)$$

где  $y_i$  – вычисленное значение функции активации,  $d_i$  – ожидаемое значение.

Модель МакКаллока-Питса эквивалентна пороговому линейному классификатору, то есть может разделить множество векторов между двумя классами с помощью гиперплоскости, если такое разделение возможно. Если такое разделение невозможно, то есть задача является нелинейно разделимой, то необходимо

использовать более сложную модель персептрона Розенблатта.

Персептрон Розенблатта состоит из 3-х слоёв, содержащих последовательно сенсорные, ассоциативные и реагирующие нейроны. Каждый слой персептрона состоит из одного вида элементов, имеющих входы на предыдущий слой. Веса сенсорных нейронов могут быть только нулями или единицами, веса ассоциативных нейронов – действительные числа. Обычно персептрон Розенблатта используется для решения задач классификации.

Существуют различные алгоритмы обучения нейросетей, основанные на оптимизации значения функционала качества: метод градиентного спуска, алгоритм Левенберга-Марквардта, метод сопряжённых градиентов [210], *Quickprop*. Все они основаны на минимизации целевой функции. Доказано [175], что можно осуществлять поправку весов для последнего слоя и выражать поправку весов низкого слоя через верхние слои, поэтому был создан алгоритм обратного распространения ошибки, позволяющий обучать все слои ИНС.

При наличии  $k$  слоёв и  $n_i$  нейронов на  $i$ -м слое ( $i \in (0, \dots, k)$ ) вычислительная сложность ИНС в прямом распространении равна  $\Theta(\sum_{i=0}^{k-1} n_i * n_{i+1})$ , что можно оценить как  $\Theta(kn^2)$ . В обратном распространении для задачи обучения вычислительная сложность остаётся аналогичной, но домножается на количество эпох обучения, количество которых сложно оценить заранее. Обучение сложных нейросетей типа свёрточных [192] приводит к тому, что время обучения становится порядка нескольких дней даже на сильных компьютерах [136], стабильность распознавания образов при этом может быть нарушена за счёт изменения единичного числа пикселей [252], что типично для ССИ с достаточным уровнем зашумлений и пересечением объектов.

Для решения задач кластеризации применяются нейронные сети Кохонена и самоорганизующиеся карты. В них при подаче входного вектора выбирается нейрон-победитель, вырабатывающий максимальный сигнал, и его топологическая окрестность. В результате обучения меняется вес нейрона-победителя и нейронов, входящих его окрестность, по заданному правилу. «Мёртвые» нейроны, то есть те,

которые не модифицируют свои веса, удаляются из сети. Таким образом, проходит самоорганизация и нейросеть разделить входные вектора по некоторому количеству кластеров.

Основной проблемой при построении ИНС является изначальное определение её структуры: количества слоёв, количества нейронов на слоях, функции активации нейронов и пр. Все эти параметры нелинейно влияют на обучение и качество нейросети: использование большего количества нейронов не всегда приводит к увеличению функционала качества и уменьшению ошибки.

### **1.3 Современные методы анализа и обработки цифровых изображений**

#### **1.3.1 Методы предварительной обработки ССИ**

Начало работы со ССИ подразумевает анализ, какие несовершенства ярче сказываются на их качестве. Это важно, так как низкоуровневые преобразования предварительной обработки очень чувствительны к малейшим вариациям графических свойств [133]. Этот анализ также является основанием для дальнейшей разработки программного обеспечения по обработке цифровых изображений.

Помимо формальных математических способов цветовой коррекции изображений (исправления яркости, контраста, цветности и т. д.) также разработаны методы, применяющие более ориентированные на объекты, а не пиксели подходы, например, анализ спектрально-контурных элементов [69].

Перед началом предварительной обработки мы имеем цифровое изображение, которое нужно преобразовать в набор объектов. Разработан адаптивный низкочастотный фильтр, подавляющий цифровой шум, и помехоустойчивый анализирующий алгоритм, работающий на изображениях, выполненных в условиях недостаточной освещенности [104]. Похожими проблемами локально обладают и топографические карты, но в работе не затронуты вопросы дальнейшего распознавания образов на изображениях.

Информация об объектах прежде всего определяется их положением и границами: даже не зная, какого цвета образ, какую функцию он может нести, мы можем выделить границы нескольких образов, конечно, если изображение уже лишено шумов, так как они могут быть интерпретированы, как края. Для выделения границ принимаются так называемые операторы, самые простейшие: Собеля, Прюитта, Робертса и лапласиана гауссиана. На основе оператора Собеля построен метод выделения контуров объектов с размытыми границами сплайн-функциями [65] и метод формирования линейных контуров на аэрофотоснимках [12].

Более продвинутый детектор, разработанный Дж. Ф. Кэнни [127], позволяет получать границы с лучшим качеством, но размывает мелкие детали и образует «ложные» контуры, поэтому непригоден для топографических карт без дополнительной модернизации. Алгоритм, использующий вейвлет-преобразование [79], наоборот, практически не образует «ложных» контуров, но создаёт разрывы в кривых границ, которые также необходимо удалять специальными алгоритмами.

Часто при обнаружении границ используется преобразование Хафа [25]. С помощью него на изображении можно находить фигуры определённого класса, заданные некоторой моделью, совершая преобразование двумерного цифрового изображения в дискретное пространство параметров, определяемых моделью. Например, если эта модель линия, то пространство параметров также двумерно, параметрами являются угол наклона линии и расстояние от неё до центра координат. Если на изображении присутствует линия с данными параметрами, то в параметрическом пространстве это соответствует большему значению.

Помимо статистических детекторов разработан детектор, использующий ИНС [245]. Данный детектор адекватно работает на изображениях непостоянного контраста и размытых границ, как на топографических картах. Недостатком данного метода является использование большой нейросети, в которой на каждый пиксель изображения приходится по 3 нейрона, что неосуществимо для топографических карт размеров в несколько миллионов пикселей. Решением является использование нейросети на меньшую область и применение метода скользящего окна.



Свёрточные ИНС применяются в основном для распознавания больших по протяжённости и сложных по цвету объектов, например, лиц или классов объектов при осуществлении дорожного движения. Использование их при обработке, например, цветных изображений низкого разрешения нецелесообразно, так как неточности при изображении маленьких объектов оказывают слишком большое влияние.

Также кластеризация может выполняться не с помощью одного вычислителя, а с помощью нескольких отдельных агентов, как в алгоритме пчелиной колонии [220]. Данный алгоритм позволяет легче распараллелить процесс, но является очень требовательным к аппаратному обеспечению.

После выделения границ необходимо выделить образы различной функциональности на топографической с помощью кластеризации цветов. Алгоритм *k-means* позволяет разбить множество цветов карты на группы, уменьшая суммарное квадратичное отклонение точек кластеров от точек центров этих кластеров. Построены алгоритмы, ускоряющие работу стандартного алгоритма Ллойда, посредством введения древовидных структур данных [183] или использованием разделяемых кластеров и медианного фильтра [146].

Разработан фильтр селективного внимания, который выполняет разбивку карт землепользования на слои, выделяя особенности, согласующиеся с результатами деления множества цветов карты, выполненного с помощью метода нечёткой кластеризации *C-means* [237]. Регулируя различные параметры алгоритма, и, учитывая контекст (влияние отдельных слоёв друг на друга), можно осуществить выборку необходимых концептуальных слоёв. Исследуемые в работе карты землепользования ориентированы на анализ застроенного пространства, обладают большим многообразием контрастных цветов, что позволяет использовать упрощённый метод кластеризации, непригодный для топографических карт.

Следующей операцией после кластеризации является сегментация согласно некоторым графическим признакам. Основные техники сегментации рассмотрены в [16, 116]; различают пороговый подход, методы, основанные на сжатии, использовании гистограмм или найденных контуров, методы выращивания областей,

графовые методы, методы водораздела, многомасштабные методы. Кроме этого последнее время развитие получил объектно-ориентированный подход, заключающийся в сегментации за счёт поиска конкретных элементов [173, 199].

Так как пороговые методы достаточно негибкие (на разных картах порогами цветов могут быть различные значения), то был разработан метод, основанный на концепции адаптивного порога, обладающего таким значением, чтобы внутриклассовая дисперсия цвета была минимальной, который был затем модернизирован на случай нескольких классов [196]. Также предложен метод [262], основанный на вычислении порога согласно значению нечёткой энтропии.

Метод водораздела заключён в представлении операции сегментации как процесса разделения воды внутри пикселей-водоразделов, имеющих максимальное значение градиента яркости. Вода, помещённая на любой пиксель внутри общей линии водораздела, течёт вниз к локальному минимуму яркости. Пиксели, от которых вода стекает к общему минимуму, и представляют собой вычисленный сегмент. Данный метод не позволяет контролировать гладкость полученных сегментов, поэтому был усовершенствован в [215].

Метод выращивания областей и многомасштабного анализа рассмотрен в [75]. Суть методов заключается в разрастании областей-сегментов на основе центров кристаллизации и постепенном переходе от локальных областей к глобальным. Они позволяют адекватно работать на изображениях, где существует множество связей, таких как топографические карты, использовать не только яркостную или цветовую, но и текстурную информацию. К недостаткам методов можно отнести высокие требования к ресурсам ЭВМ.

Одним из важнейших аппаратов для работы с изображениями, особенно в целях сегментации, является математическая морфология, рассмотренная в [81]. Морфологический анализ – это анализ геометрической структуры изображения, основанный на теории множеств и топологии. В основе аппарата – выполнение определённых операций над каждым пикселем изображения. Основными операциями являются: перенос, дилатация и эрозия, замыкание и размыкание; они выполняются

с помощью структурных элементов (небольшое изображение), с помощью которого определяется влияние значения пикселя на своих соседей.

Классический способ использования математической морфологии приведён в исследовании, автоматизирующем получение логических цепей из сканированных документов [140]. Здесь морфология с одной стороны используется для того, чтобы избавиться от шумовых пикселей, с другой – чтобы выделить объекты, имеющие определённую ориентацию и форму.

Данный подход также полезен, когда происходит работа с моделями, для которых выделить дескрипторы очень сложно или невозможно, например, как в работе [260], в которой авторы обрабатывают карты для нахождения и анализа лесной территории. Разработанный метод достаточно прост, но позволяет различать леса различного типа между собой.

Одним из терминов математической морфологии является скелет – множество точек, равноудалённых от границ фигуры. Скелеты используются, когда нужно привести кривую к форме, когда её толщина будет состоять всего из одного пикселя. Полученные скелеты [58] нельзя напрямую использовать в распознавании, но они являются адекватными упрощёнными моделями кривых, совпадающих с ними по расположению и форме и отличающиеся только толщиной линий. Волновой алгоритм [61] выполняется итерационно внутри уже определённого сегмента: скелет представляет собой кривую, образованную центром распространяющейся по сегменту сферической волны.

В отличие от предыдущих, методы, рассмотренные в [185, 191, 200], касаются вопросов предварительной обработки именно карт, учитываются особенности и структура таких ССИ, как топографические карты. В [200] авторы предлагают комплексный метод, выполняющий следующие операции: нахождение границ между двух цветов и метода водораздела, строгое слияние полученных кусочков для избегания ложных сращиваний и анализ текстуры для выделения сегментов по разным слоям, основанный на методе опорных векторов. В данной работе в основном рассматривается сегментация по цвету, поэтому образованные суперпиксели

могут содержать в себе объекты разной природы.

В [191] проводится анализ исторических карт, нарисованных вручную, в решении частой задачи нахождения символов деревьев. В основе разработанного метода – модернизированное преобразование Хафа, основанного на параметрической модели изображения дерева. Данный подход обоснован тем, что нарисованные вручную деревья очень непохожи друг на друга, поэтому невозможно провести текстурную сегментацию напрямую.

Метод сегментации, основанный на аппарате математической морфологии, позволяет точно обрабатывать двумерные области, такие как кадастры [185]. В работе также рассмотрено улучшение изображения посредством удаления координатной сетки с последующим восстановлением контуров границ кадастров. В топографических картах также присутствует координатная сетка, поэтому предложенное решение будет актуально и в рамках решаемой задачи.

### 1.3.2 Методы распознавания образов

В работе [49] построен алгоритм, модернизирующий шаблонный подход для распознавания образов, выполняющий нахождение опорных точек (совпадающих по цвету) на образе и шаблонах. При этом алгоритм осуществляет аффинные преобразования при необходимости, что позволяет учесть повороты и изменения размеров изображений. Данный алгоритм подходит для поиска отдельных изображений, но использование его при распознавании топографических карт может привести к значительному увеличению времени выполнения.

В [97] разработан новый метод триплетных признаков, представленный, как обобщение метода моментов, метода признаков интегральной геометрии, метода проекций. Данный метод не распознаёт образы напрямую, но может быть использован для формирования признаков образа и дальнейшей обработке известными методами классификации.

Применяемый для распознавания написанных цифр метод опорных векторов

в [232] позволяет распознавать изображения с помощью «вида» цифры с определённой стороны (слева, сверху и т. д.), что напоминает метод гистограмм. Каждый вид представляет собой график функции, отражающей расстояние до значащего пикселя от плоскости взгляда, вдоль горизонтальной оси взгляда. В качестве признаков используются конечные разности функций вида между соседними пикселями. Вероятность распознавания цифр данным методом составила 91%.

Анализ эффективности объединения метода линейного дискриминантного анализа (ЛДА) и метода главных компонент (МГК) производился в [68] на примере задачи распознавания лиц. МГК позволяет уменьшить количество исследуемых признаков, потеряв наименьшее количество информации. Полученные признаки объединяются в комбинации с помощью ЛДА, ещё сильнее уменьшая количество признаков. Объединение данных двух методов позволяет улучшить распознавание с 60% до 82% (при использовании двух примеров изображения) и с 94% до 98% (при использовании восьми примеров).

Оптическое распознавание применительно к топографическим данным рассматривается в [147]. Поставленная задача распознавания кратеров решается с помощью применения окружного преобразования Хафа (в качестве параметрической модели – окружность) и использования признаков Хаара в каскаде.

Общий недостаток оптических методов распознавания – снижение качества распознавания при работе в зашумлённых и/или небольших областях. Приёмы статистического и интегрального анализа могут адекватно работать только на больших изображениях, на ССИ же часто образами имеют площадь в несколько десятков пикселей. Поэтому логичнее было бы не упрощать модели распознавания, а использовать максимальное количество признаков, например, с помощью нейронных сетей.

В [117] рассматриваются различные модели распознавания символов: одно- и многослойные, радиальные нейросети, описаны алгоритмы их оптимального обучения, оптимизации и обобщения. Построенная с помощью пакета *MatLab* нейросеть распознаёт отдельно стоящие, нормализованные символы [243], но не

может быть использована для распознавания символов под произвольным углом. Данная простейшая нейросеть имеет процент распознавания 85%, что показательно относительно сложности построения метода из [68, 232].

Построенная ИНС в работе [171] позволяет осуществлять текстурную сегментацию (в данном случае текстура рассматривается как отдельный образ, а не признак). На вход нейросети подаётся сигнал с фильтров Габора, использующихся для выделения границ. Данный метод можно применить в распознавании болотистой местности на топографических картах.

Обычно нейросеть, обучаемую для распознавания символов, строят таким образом, чтобы один и тот же нейрон мог реагировать на некоторое множество входов и быть входом для нескольких последующих нейронов. В [24] построена связанная модульная нейросеть, каждый модуль которой ответственен за распознавание отдельной цифры, при этом обучение происходит только у межмодульных связей, а модули имеют некую заранее организованную идентичную структуру. Полученный процент распознавания символов 98.15%.

В [2] рассмотрена нейросеть особой конфигурации «неокогнитрон», являющейся почти полным аналогом нейрофизиологической системы распознавания человека: она состоит из большого числа слоёв простых (*S*-типа), тормозящих (*V*-типа) и сложных (*C*-типа) нейронов. Построенная система нечувствительна к изменению масштаба, поворотам изображения, шуму и прочим искажениям. Обучение таких нейросетей происходит гораздо быстрее, чем обычных. Получен результат в 85% правильного распознавания символов.

Кроме распознавания отдельных символов, объектами также могут быть дороги и дома, не имеющие постоянной структуры. Соответственно, для таких объектов используется структурный подход. В [209] построен алгоритм, выполняющий распознавание домов по сегментированным линиям стен. Если цепь, образованная этими линиями, замыкается и полученный объект соответствует по размеру домам, то полученный объект и будет являться домом; качество распознавания доходит в данном случае до 90%. Также широко применяются свёрточные нейронные

сети [149, 257, 258, 271] и метод решающих деревьев [152, 176], основанные на выделении особенностей изображений зданий, тем не менее, такие подходы в настоящее время ограничены по качеству в 90-95%.

В [78] показано, что использование структурных методов, например, применяя дескрипторы изображений на основе графов можно достичь повышения надёжности распознавания, что также можно приводит к повышению его качества. Кроме того, анализ большого числа пикселей при этом заменяется анализом небольшого количества структурных составляющих.

В [160] построен алгоритм для определения дорог в лесных горных областях по топографическим данным лидара. В алгоритме используются морфологические операции для предварительного нахождения дорог и классификатор случайного леса. После анализа проходило построение графа дорог. Метод показал качество распознавания 80%.

В [181] также проведена векторизация кривых с помощью небольших отрезков, но алгоритм обладает свойством адаптивности. Во-первых, происходит распознавание узловых точек кривой, например, точек изгиба. Во-вторых, происходит распознавание линий между этими узловыми точками. Таким образом, топологически, даже если второй этап выполнится не совсем корректно, структура приблизительно может быть восстановлена с использованием узловых точек. Полученная ошибка распознавания меньше 1%.

Разработанный алгоритм [234] распознавания изолиний топографической карты на основе триангуляции Делоне и диаграммы Вороного позволяет устранить разрывы, вызванные наложением посторонних линий (например, сетки карты) и прочих искажений. Главной проблемой такого распознавания является наличие отметок высот и прочих объектов, одинакового цвета с изолиниями. Чтобы решить её необходимо строить более сложную систему, учитывающую функциональные взаимодействия и производящую автоматический поиск ошибок.

Автоматизированное распознавание топографических карт разрабатывалось в рамках проекта *IRIS*. В [253, 254] отражены основные результаты работы по

распознаванию точечных, линейных и двумерных объектов. Результаты распознавания колеблются от 80% до 99%, что является хорошим результатом, но при обработке используются карты размеров порядка 100 Мп.

### 1.3.3 Анализ сцены и постобработка

Анализ сцен – финальный этап распознавания образов, понимание не только характеристик самих объектов на изображении, но и их отношений в целом. Только здесь возможно наиболее полное понимание совокупности полученной информации и поиск ошибок, совершенных на предыдущих этапах. Никакой метод не даёт абсолютного качества распознавания, но оно возможно при комплексном взаимодействии рассматриваемых образов.

В основе анализа сцен лежит принцип согласованности или использования метаинформации об образах, например, мы знаем, что буквы в словах или цифры в числах расположены рядом, на одной прямой, отображены с помощью одного шрифта, размера и цвета. Так же мы можем сказать и о других связанных образах, например, двумерные объекты гидрографии имеет равную альтитуду по всей поверхности, горизонтали не могут пересекаться и т. д. Нахождение таких неявных правил и использование их при распознавании, то есть использовании группировки распознанных объектов, относящихся к разным семантическим планам, способствует улучшению качества и пониманию изображения в целом.

В [268] разработан алгоритм группировки букв в слова на топографических картах, использующий принцип согласованных ограничений: равенство цвета, равенство размера и сонаправленность букв. Все эти ограничения позволяют отделить различные строковые обозначения друг от друга, но не позволяют работать с названиями, расположенными по кривым, например, названия рек.

Модернизация распознавания топонимов на топографических картах может идти с помощью накладывания дополнительных связей на буквы: геометрических [135] или лексических [223]. В [135] принцип согласованности улучшен за счёт



учёта наклона букв, поэтому слова, расположенные вдоль кривых, могут распознаваться. Дополнительными ограничениями являются частота букв вдоль направления слова, параметры шрифта. В [223] принцип согласованности улучшен за счёт использования словаря: в распознанных словах ищутся ошибки, что приводит к увеличению распознавания до абсолютного уровня.

Алгоритм, построенный в [162] для распознавания рукописных слов в связанный текст основан на обучаемой скрытой марковской модели (*HMM*), и производится сравнение его с алгоритмом динамической трансформации временной шкалы (*DTW*). В разработанном алгоритме не используется группировка букв, а в строки различной длины, и алгоритму неважно понимание используемого алфавита и словаря. Похожий алгоритм, использующий *HMM*, но ориентированный на выделение слов разработан в [191]; точность распознавания составляет 93%. В [265] точность доведена до уровня 99,9% благодаря использованию не только словарей, но и сходству формы у разных букв и учёта вероятности появления букв друг после друга.

Важным вопросом при распознавании образов на ССИ является учёт неоднородности фона: символы и линии могут лежать на нескольких двумерных объектах различного цвета, распознавание при этом резко падает; качество различных методов рассмотрено в [101]. Отмечено, что единственной моделью, адекватно работающей на смешанном фоне, являются нейросети.

Следующие вопросы относятся к анализу сцен, но должны выполняться перед предобработкой, так как изображение топографической карты обычно является частью некоторого большего изображения, содержащего рамку и легенду карты. Интересующая нас информация также может оказаться за пределами изображения топографических карт. Например, классификация документов изображений, рассмотренная в [148], позволяет распознать отдельные составляющие документа и построить его структуру.

Очевидно, что некоторые сущности реального мира не могут быть отображены строго в одном ССИ, а поэтому возникает проблема адекватной их сборки для образования одного семантического плана. При обработке топографических

карт это поможет нивелировать погрешности у края карт. Способы совмещения нескольких изображений рассмотрены в [70, 247]: в них используется контурный анализ для нахождения сходств и аппарат математической морфологии.

Для получения цифровой модели местности необходимо будет интерполировать значения высот в точках, принадлежащим горизонталям, на всю оставшуюся область карты. В геостатистике существуют [23, 246] стандартные методы интерполяции, показывающие хорошие результаты для нормально распределённых данных. Высокое быстродействие и качество показывает линейный метод интерполяции, метод кубических сплайнов и метод естественного приближения Сибсона. Кригинг хоть и даёт наилучшее линейное несмещённое предсказание, но является достаточно затратным с точки зрения времени обработки.

#### 1.3.4 Оценка алгоритмов

Только на конечном этапе обработки цифрового изображения топографической карты мы можем определить качество непосредственно, через отношение правильно распознанных объектов к количеству объектов на карте. Оценивание промежуточных результатов распознавания, таких как сегментация символов и кусочков линий или кластеризация цветов, использующие различные методы, не является корректным, так как большая ошибка при использовании одного метода может нивелироваться в постобработке, а меньшая, при использовании другого, – нет.

Тем не менее, оставлять оценку методов до финального этапа не является удобным, особенно в случаях, когда провести её автоматически полностью не представляется возможным, как в случае со ССИ со слабой формализацией. Проверить работоспособность методов распознавания текстовых документов гораздо проще, ведь кроме документа, как множества пикселей, есть также и документ, как упорядоченное множество символов, которые можно поставить в соответствие распознанным образам.

Способы оценивания для различных моделей машинного обучения

рассмотрены в [145], указаны варианты возможных оптимизаций и снижения показателей ошибок. В [17, 54, 202, 273] также показаны современные способы оценивания качества работы классификаторов, детекторов краёв, процессов сегментации. Главными вопросами тут является адекватность этих критериев и способы оценивания качества процессов постобработки для распознавания топографических карт, так как в данной области они практически не исследованы.

Одним из часто встречающихся общих принципов оценки качества для бинарной классификации (в рамках проверки нулевой  $H_0$  и альтернативной  $H_1$  гипотезы) является доля правильных ответов (*accuracy*), полнота (*recall*), точность (*precision*), оценка по ошибке первого (*false positive*), второго (*false negative*) рода и их комбинация  $F$ -мера:

$$\begin{aligned} accuracy &= \frac{TP + TN}{TP + TN + FP + FN}, & recall &= \frac{TP}{TP + FN}, \\ precision &= \frac{TP}{TP + FP}, & F_\beta &= (1 + \beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \end{aligned} \quad (1.3)$$

где  $TP$  – количество правильно отвергнутой  $H_0$ ,  $TN$  – количество правильно принятой  $H_0$ ,  $FP$  – количество неправильно отвергнутой  $H_0$ ,  $FN$  – количество неправильно принятой  $H_0$ ,  $\beta$  – параметр  $F$ -меры.

К сожалению, данные метрики не отражают устойчивости моделей распознавания, так как их обучение (особенно если они являются интеллектуальными) происходит, во-первых, при случайном выборе данных для обучения, а, во-вторых, порядок данных для обучения тоже выбирается случайно. Например, при запуске обучения ИНС точность распознавания может варьироваться от запуска к запуску в районе 10-15%, поэтому важной метрикой является ещё и мера разброса показателей качества.

Чтобы повысить стабильность интеллектуальных моделей используется метаалгоритм бэггинга (*bootstrap aggregating*) [233, 244]. Его идея заключена в генерации большого числа интеллектуальных моделей. Изначальный тренировочный

набор для каждой модели модифицируется взятием из общей выборки с возвратом.

Важным критерием качества является не столько количество ошибок, но относительное количество ошибок на обрабатываемый объём изображения. Размер ССИ, особенно карт, зачастую измеряется в мегапикселях. Этот параметр важен, так как возможно придётся сравнивать алгоритмы, которые проходили экспериментальную проверку на картах с различной разрешающей способностью, то есть с различным *dpi*. При увеличении *dpi*, увеличивается её размер в мегапикселях, при сохранении физического размера карты. Алгоритмы при этом должны давать меньшую ошибку.

Для определения качества работы алгоритмов, выдающих двумерные области, используется мера Жаккара или *Intersection over Union (IoU)*, понимаемая как отношение мощности множества пересечения верной и предлагаемой области к мощности множества их объединения.

Скорость работы алгоритмов также необходимо считать относительно размера изображений, так как увеличение размера изображения в 2 раза должно приводить к увеличению времени обработки в 2 раза при сохранении аппаратного обеспечения. К сожалению, во многих современных исследованиях вопрос времени выполнения обработки не поставлен; в редких исследованиях указано аппаратное обеспечение, на котором происходила проверка алгоритмов.

#### **1.4 Выводы и постановка задач исследования**

В первой главе проведён анализ существующих моделей и алгоритмов обработки цифровых изображений с точки зрения концепции сложноструктурного изображения. Проведена классификация изображений по категориям отображаемых объектов и выяснены признаки сложноструктурности. Были исследованы основные методы и модели, использующиеся для предварительной обработки цифровых изображений, распознавания образов, понимания изображения и принципы оценивания этих методов. Из материалов, изложенных в данной главе, можно

сделать следующие выводы.

1. В рассмотренных исследованиях представлено большое количество разнообразных методов и моделей для обработки изображений. Использование конкретных методов может давать хороший результат при обработке изображений, имеющих узкую направленность. При этом ССИ тяготеют к консолидации графики различного типа и гетерархичности семантических планов, что приводит к тому, что необходимо использовать гибридные и итеративные алгоритмы, которые были найдены лишь в небольшом числе публикаций, что говорит о слабой разработанности темы.

2. Большинство исследований рассматривает только отдельные шаги обработки ССИ, без комплексного решения задачи, оставляя эту работу оператору ЭВМ. Без решения комплексных задач невозможна дальнейшая автоматизация и автоматический поиск ошибок. Не выявлено работ, предполагающих совместную обработку образов, относящихся к различным семантическим полям. Анализ неявных правил отображения объектов или анализ влияния нарушения этих правил выражен лишь в малой части работ.

3. В большинстве проанализированных исследований не отражены временная сложность алгоритмов, конкретное время выполнения и аппаратное обеспечение. При известности времени оно обычно является достаточно большим из-за использования сложных интеллектуальных моделей и неоптимальных алгоритмов полиномиального времени высокого порядка. Точность и стабильность алгоритмов также являются важными критериями качества, но именно суммарное количество времени работы ЭВМ и оператора является ключевым в плане эффективности.

4. Многие исследования тяготеют к рассмотрению более простых моделей ССИ, которые удобно обрабатываются отдельными, негибридными методами. Реальные ССИ часто могут кроме своей сложности иметь низкое разрешение, высокую зашумлённость, неоднородность фона и факты нарушения правил отображения. Всё это приводит к высокой степени дробления сегментов, что говорит о необходимости создания алгоритмов, которые могут восстанавливать целостность

образов через группировку. Существует лишь небольшое количество исследований, посвящённых этой теме, что говорит о том, что она ещё недостаточно разработана.

5. Качество разработанных интеллектуальных моделей показано в виде конечных значений, что не позволяет оценить их стабильность при многократном запуске из-за стохастичности алгоритмов машинного обучения. Стабильными по определению являются линейный регрессор,  $k$ - $NN$  классификатор и  $SVM$ , для всех типов  $ANN$  анализ стабильности является открытой проблемой.

6. Часто в исследованиях используется простейшая пороговая обработка, многие важные параметры методов и моделей задаются константами, выбор значений размеров скользящих окон и структурных элементов не обоснован. Всё это приводит к слабой гибкости разработанных алгоритмов. Данный подход целесообразен при использовании интеллектуальных моделей с большим количеством решающих элементов (ИНС), которые поэтому и могут быть простыми.

7. Дополнительно стоит отметить качество не только материалов ССИ, которые можно использовать для обучения интеллектуальных моделей и алгоритмов, и которые зачастую либо не имеют размеченных данных, либо имеют слабую разметку вследствие сложности оригинальных изображений, либо представляются в виде кусков изображений, а не целиком, но и недостаточное рассмотрение их исследователями. Почти во всех работах тренировочный набор данных скорее аугментируется с использованием стандартных операций (симметрия, вращение, сдвиг, добавление шума), что приводит к необходимости рассмотрения более сложных интеллектуальных моделей, вместо попытки упрощения данных и нахождения их инвариантов, в принципе невосприимчивых к подобным операциям. Открытым остаётся вопрос: насколько изначальные образы могут быть ухудшены для сохранения необходимого уровня точности распознавания?

**Целью работы** является повышение точности и производительности алгоритмов анализа и обработки сложноструктурных изображений с помощью объединения различных подходов обработки изображений.

Для достижения цели ставятся следующие задачи.

1. Разработка обобщённой методики анализа и обработки ССИ.
2. Разработка гибридных алгоритмов сегментации, классификации и локализации сегментов, поиска, распознавания и группировки образов различной размерности на ССИ, используя различные геометрические и интеллектуальные модели, опираясь на критерии качества и быстродействия.
3. Экспериментальная проверка разработанных алгоритмов на различных классах ССИ с вычислением критериев качества, быстродействия и стабильности, сравнение полученных результатов с современными научными исследованиями.
4. Разработка специализированного алгоритмического и программного обеспечения для анализа и обработки цифровых топографических карт, как типичного представителя ССИ.

## Глава 2. Разработка алгоритмов анализа и обработки сложноструктурных изображений

В данной главе приведён анализ ССИ с точки зрения теории обработки цифровых изображений. Составлена общая методика обработки ССИ и разработан набор алгоритмов, выполняющих отдельные операции данной методики. Разработаны критерии анализа качества реализации алгоритмов с точки зрения скорости и точности.

### 2.1 Методика анализа и обработки ССИ

Информацию в виде цифрового изображения можно представить в виде отображения  $f: E \rightarrow P$ , где  $E$  – некоторое описание внешнего мира,  $P$  – цифровое изображение, заданное в виде матрицы. Таким образом, основной целью анализа и обработки ССИ является разработка обратного отображения  $f^{-1}: P \rightarrow E'$ , где  $E'$  – необходимая часть описания внешнего мира, например, свойства и расположение определённых объектов на изображении.

Исходное отображение  $f$  не является обратимым в общем случае, так как, во-первых, является функцией свёртки, преобразующей входные данные произвольного размера в матрицу установленного. Тем не менее, для дальнейшей работы (например, принятия решений) получение полной и точной информации о внешнем мире зачастую не нужно: достаточно знать её с некоторой погрешностью.

Во-вторых, правила отображения  $f$  могут иметь как явный характер, так и неявный. Под явными правилами подразумевается однозначное отображение объектов внешнего мира и их свойств в визуальный план, например, изображение дома на карте в виде условной пиктограммы или упрощённого кадастрового плана. Под неявными правилами можно подразумевать как многозначные отношения, когда элементы изображения не имеют конкретной привязки в пространстве растра, или выражены в виде неявных функций, связывающих части  $P: R(p_1, \dots, p_k) = 0$ , где  $p_i$



– элемент изображения (ЭИ), или части  $E: R(e_1, \dots, e_k) = 0$ , где  $e_i$  – объект или система объектов внешнего мира, или более сложного отношения  $R(p_1, \dots, p_k, e_1, \dots, e_k) = 0$ .

Многозначные отношения, как показано в [125] на примере выяснения соответствия подписей и пиктограмм городов на средневековых картах, приводит к решению комбинаторной  $NP$ -полной задачи. Неявные отношения могут быть выражены сложными характеристиками, вычисленные на неопределённом числе элементов изображения, например, ограничение на количество меток на  $1 \text{ дм}^2$  карты, или наложение друг на друга нескольких визуальных планов или порождены физическими законами внешнего мира, например, динамические процессы, предшествующие фиксации статичного изображения, могут нести дополнительную информацию. Примером неявного отношения, соединяющего в себе как элементы изображения, так и объекты внешнего мира, можно считать порчу физического носителя изображения, которое сохраняется при сканировании и перевод в цифровой формат [159].

Чем больше отношение явных правил в отображении  $f$  над неявными, тем проще выполнить обратное отображение. Если явные правила обычно выражены в виде определённых стандартов (например, [28, 95]), которые распространяются на широкий класс изображений, то неявные правила могут зависеть от многих вещей. Например, предпочтений оператора, отвечающего за создание изображения  $P$ , или от сложности отображаемого внешнего мира (ср. решение задач идентификации по фото паспорта и идентификации по камерам видеонаблюдения в метрополитене).

Для минимизации неявных правил и упрощения их разрешения необходимо нивелировать влияние ЭИ друг на друга, что достигается уменьшением их рассматриваемых размеров с помощью декомпозиции. Для этого будем использовать восходящий подход к созданию общей методики обработки ССИ: анализ и обработка будет происходить от отдельных пикселей к изображению в целом. Обобщённо методика показана на рис. 2.1. Характеристики разработанных алгоритмов и их схемы представлены в Приложении А.

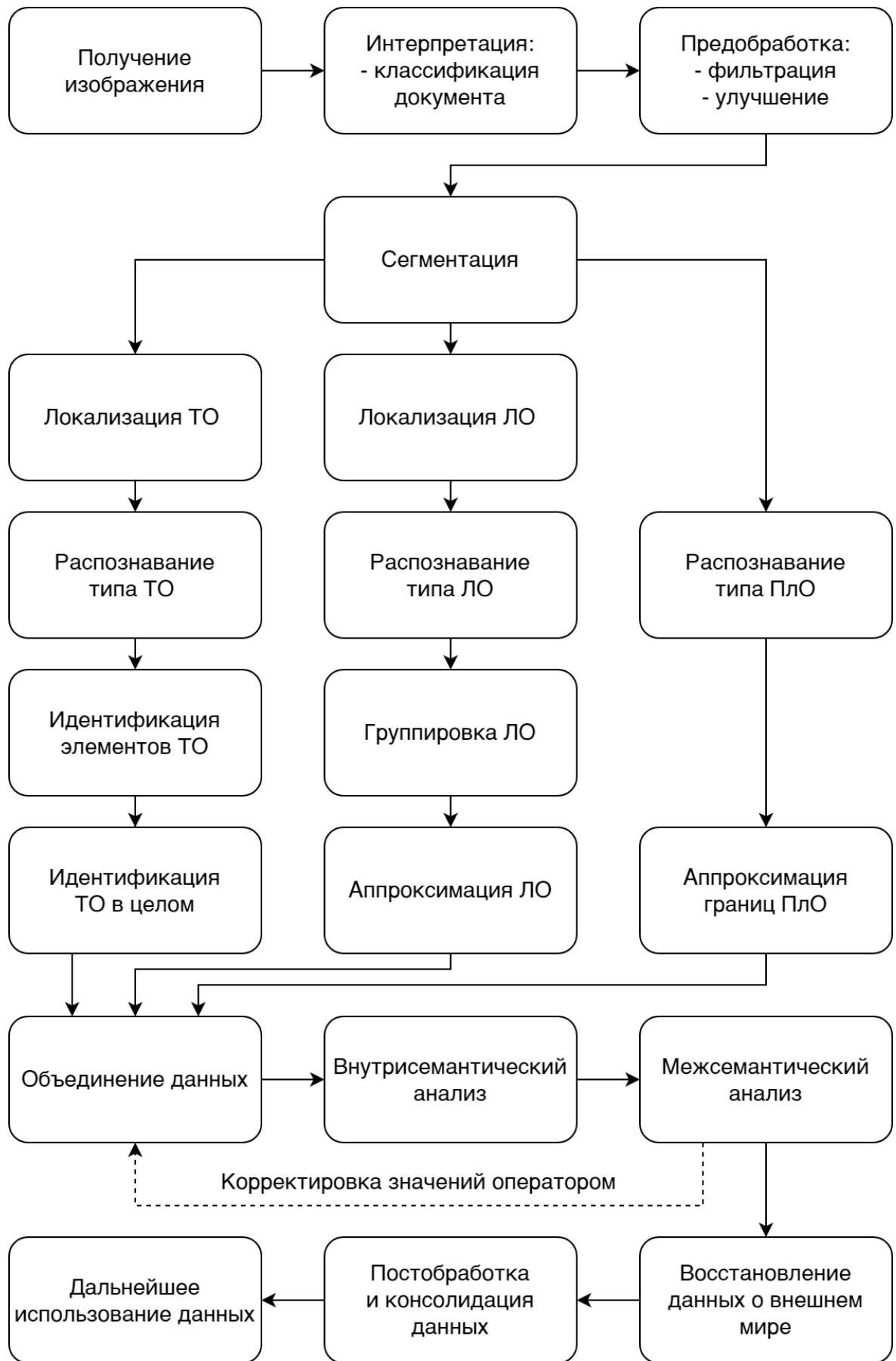


Рисунок 2.1 – Обобщённая методика программного анализа и обработки ССИ

После получения цифрового изображения необходимо классифицировать его, так как специальные документы подразумевают различную дальнейшую обработку. На этом же этапе происходит разделение документа, если это необходимо на поддокументы, например, отделение текстовой части от иллюстраций.

Предобработка в основном заключается в фильтрации от шумов, очистке от макроскопических объектов и цветовом улучшении изображения. Улучшение может заключаться в нормализации контрастности, яркости, выполнения необходимых смещений в цветовом пространстве. Предобработка в основном выполняется попиксельно или с помощью скользящих окон небольшого размера для выполнения свёрточных операций.

Сегментация разделяет множество растра на объекты-сегменты, имеющие точечную (ТО), линейную (ЛО) или площадную (ПЛО) размерность. Все объекты на двумерном растре можно считать ПЛО, поэтому под ТО предполагаем объекты, размер которых в пикселях не изменяется при переходе к изображениям других масштабов (можно говорить о классе  $O(1)$  на затраты памяти на хранение отдельного ТО). Под ЛО предполагаем объекты, размер которых в пикселях изменяется пропорционально изменяемому масштабу (можно говорить о классе  $O(\sqrt{S})$  на затраты памяти на хранение отдельного ЛО, где  $S$  – площадь растра).

Разделение объектов по размерности объясняется тем, что для каждого из них существуют специфичные способы обработки: ТО тяготеют к более интеллектуальным методам, ПЛО – к более статистическим.

После сегментации происходит этап распознавания объектов, каждый из которых специфичен по размерности. При необходимости простые объекты группируются в более сложные и происходит их упрощение. Далее данные собираются по семантическим планам и происходит их комплексная обработка: сначала внутрисемантическая, затем межсемантическая. На этапе комплексного анализа в обработанных данных находятся ошибки и получается отсутствующая информация.

В конце происходит восстановление данных о внешнем мире во всём множестве растра, например, интерполяция, на этапе постобработки происходит

дополнительная привязка и консолидация данных с информацией из других документов. В дальнейшем прикладное программное обеспечение может выполнять визуализацию, статистический и прочий анализ для поддержки принятия решений или преобразования векторизованных данных в другие типы.

Если автоматизировать такую методику в целом невозможно из-за высокой степени влияния объектов друг на друга или особенностей обрабатываемых документов, то необходимо оставить возможность исправления данных оператору (на схеме отображена корректировка только для части, отвечающей за комплексный анализ).

## 2.2 Алгоритм многометочной сегментации

В конечном итоге при обработке ССИ нужно перейти к получению и анализу объектов. Исторически сложились два подхода к реализации такой «объектной» сегментации: экземплярная основана на разделении счётных объектов, семантическая на разделении классов объектов. Ни тот, ни другой подход не может показать достаточного качества при анализе ССИ. Первый опирается на идею обработки объектов-вещей (*things* в [186]), которые могут в значительной степени накладываться друг на друга (нет строгой отделимости). Второй – на идею обработки объектов-регионов (*stuff*), которые на ССИ могут проявляться с большой вариативностью.

В настоящее время активно формируется паноптический подход, который не разделяет два выше указанных понятия и поэтому является более гибким. Основная его идея заключается в использовании простейших вычислителей для формирования непересекающихся суперпикселей [197, 206]. При этом всё равно считается, что отдельный пиксель может принадлежать только одному классу, что не соответствует свойству гетерархичности ССИ, поэтому предлагается подход *многометочной сегментации*, при котором пиксель соответствует классам объектов с некоторой мерой.

В [47] приведена формула (2.1) для описания отдельного пикселя на изображении в случае, когда каждый объект определяется одним цветом. Хотя данный случай и является простейшим, но на ССИ с большим числом пересечений его можно использовать в качестве основы, так как даже сложные объекты зачастую представляются в виде набора одноцветных объектов.

$$\mathbf{p} = \sum_{i=1}^n \omega_i * \mathbf{C}_i + \mathbf{E}, \quad (2.1)$$

где  $\mathbf{p}$  – вектор цвета пикселя в пространстве  $RGB$ , координаты которого принимают дискретные значения, например,  $[0, 255]$ ;  $n$  – количество классов объектов карты;  $\mathbf{C}_i$  – вектор цвета  $i$ -го класса объекта;  $\mathbf{E}$  – вектор зашумлённости (может иметь отрицательные координаты);  $\omega_i$  – весовые коэффициенты при векторах цветов.

Задачей многометочной сегментации является определение весовых коэффициентов  $\omega_i$  для каждого типа объекта. При работе с цветными изображениями данная задача может быть разрешима в общем случае с помощью формальных математических методов не более чем для трёх классов объектов.

Если объект нельзя представить в виде совокупности объектов одного цвета, то можно выдвинуть более общую формулу (2.2), в которой учитывается разница координат между рассматриваемым пикселем и центром объекта ( $\Delta x, \Delta y$ ) и прочие свойства объекта, например, угол наклона  $\alpha$  и масштаб  $s$ . В ней можно учесть и такие изменения ССИ, как изменение яркости или насыщение, с помощью общей функции  $g$ .

$$\mathbf{p} = g \left( \sum_{i=1}^n \omega_i * \mathbf{C}_i(\Delta x_i, \Delta y_i, \alpha_i, s_i) + \mathbf{E} \right) \quad (2.2)$$

Из анализа формулы (2.2) случая можно сделать вывод о невозможности проведения сегментации без знания информации о расположении объектов. Хотя эта информация согласно разработанной методике (рис. 2.1) получается при последующих этапах обработки, но может быть получена без проведения этапа сегментации. Данное положение подтверждается многочисленным использованием алгоритмов *R-CNN* и *YOLO* при обработке фото- и видеоматериалов. Проблема данных алгоритмов заключается в их локальности: они могут быть применимы только для нахождения ТО, но не для ЛО или ПЛО.

Учитывая предыдущие замечания о качестве ССИ, наличии на них объектов определённых типов и исследовании методов сегментации [47], был сделан вывод о необходимости использования интеллектуальных методов для сегментации, представляющую собой классификацию отдельных пикселей. Так как для определения, является ли данный пиксель частью некоторого объекта, нам необходимо знать окружение данного пикселя, то для анализа можно использовать метод скользящего окна. Окно размером  $a \times a$  ( $a \in \mathbb{N}, a > 1$ ) пикселей приведёт к созданию  $3a^2$  значений входных данных, которые необходимо проанализировать с учётом различных направлений. Квадратное окно выбрано исходя из возможности изображения объектов под любым углом.

Так как нам необходимо только определить, принадлежит ли данный пиксель некоторому  $i$ -му типу объектов, то выходной слой интеллектуальной модели должен быть функцией активации *softmax*:

$$\sigma_i = \frac{e^{z_i}}{\sum_{k=1}^K e^{z_k}}, \quad (2.3)$$

где  $K$  – количество классов,  $z$  – вектор размерности  $K$ .

Значение данной функции (которое может лежать в интервале  $[0, 1]$ ) для каждого  $i$  определяет меру того, что пиксель будет принадлежать классу объектов  $i$ . Сумма всех значений  $\sigma_i$  равна единице. Таким образом, изображение после

проведения данной многометочной сегментации преобразуется в  $K$  полутоновых изображений, что предъявляет повышенные требования к памяти системы при наличии большого числа классов. Хранение данных растров можно сократить, если использовать разрежённые матрицы, к которым можно перейти, если переинициализировать низкие значения  $\sigma_i$  нулём, например  $\sigma_i < 0.1$ .

Например, сегментация цифровых топографических карт (ЦТК) для целей определения рельефа подразумевает существование объектов трёх классов. Первый – это объекты отображения рельефа: горизонтالي, бергштрихи, надписи на горизонталях; они изображаются коричневым цветом. Второй – точечные геодезические объекты: отметки высот (в т. ч. командных), геодезические символы; они изображаются чёрным цветом. Третий тип (фоновый) – пиксели, соответствующие другим объектам.

Для начала необходимо обучить интеллектуальную модель с помощью обучающих данных, для этого необходимо непосредственно на изображении разметить пиксели. Из-за большого размера ЦТК провести полностью данную операцию не представляется возможным, поэтому необходимо обеспечить репрезентативность выборки. Она должна отражать все возможные проявления изображения условных знаков: на разном фоне, под разными углами, знаки должны иметь разное значение. Особое внимание следует уделить областям, где пересекаются объекты разных классов.

После составления обучающей выборки необходимо разделить её на три группы: первая группа будет обучающей, вторая валидирующей, а третья тестовой. Валидирующая выборка нужна для вычисления генерализации обучения и для остановки обучения, когда генерализация прекращает улучшаться. Тестовая группа нужна для проверки системы на возможное переобучение: для таких параметров интеллектуальных моделей показатели ошибок будут значительно выше у тестовых выборках.

Алгоритм обучения интеллектуальной модели (на примере искусственной нейронной сети прямого распространения с одним скрытым слоем) для

многометочной сегментации показан в алгоритме 2.2 с помощью псевдокода на основе языка программирования *Matlab*, в котором опущены несущественные для понимания команды. Входными параметрами для алгоритма являются: исходное цифровое изображение, размеченное изображение, вектора цветов для размеченных  $K$  классов, размер скользящего окна. Выходным параметром является обученная интеллектуальная модель.

Вспомогательным для данного алгоритма и многих других алгоритмов далее является функция (алгоритм 2.1), которая по известному множеству точек находит описания всех соседних пикселей в окне размера  $a_x \times a_y$ , повернутом на угол  $\varphi$ . Описание для каждой точки является вектором, в котором сначала идут значения крайнего левого верхнего соседа и т. д. Временная сложность данной функции соответствует  $\Theta(|Points| * a_x * a_y)$ .

---

#### Алгоритм 2.1 – Функция нахождения информации о соседних пикселях

---

```

1:  function result = getNeighborhood(picture, points, a, phi)
2:      depth = numel(picture) / (size(picture, 1) * size(picture, 2))
3:      ax = a(1)
4:      ay = a(2)
5:      result = zeros(depth * ax * ay, size(points, 2))
6:      [dx, dy] = meshgrid(-(ax - 1) / 2 : (ax - 1) / 2,
                          -(ay - 1) / 2 : (ay - 1) / 2)
7:      picture = padarray(picture, [ax, ay, 0], 0)
8:      points(1, :) = points(1, :) + ax;
9:      points(2, :) = points(2, :) + ay;
10:     for i = 1 : size(points, 2)
11:     |   for j = 1 : ax * ay
12:     | |   cx = round(points(1, i) + dx(j) * cos(phi) - dy(j) * sin(phi))
13:     | |   cy = round(points(2, i) + dx(j) * sin(phi) + dy(j) * cos(phi))
14:     | |   if depth == 3
15:     | | |   result(3 * j - 2 : 3 * j, i) = [picture(cx, cy, 1),
                                           picture(cx, cy, 2),
                                           picture(cx, cy, 3)]
16:     | |   elseif depth == 1
17:     | | |   result(j, i) = picture(cx, cy)
18:     | |   end
19:     |   end
20:     end
21: end

```

---

В первой части алгоритма 2.2 (строки 2..12) находятся размеченные пиксели и создаётся разряжённая матрица, которая ставит в соответствие координатам пикселей значение класса. Затем для данных пикселей формируется описание их



соседей. Байтовые значения цветовых координат из интервала  $[0, 255]$  нормализуются к интервалу  $[0, 1]$  для повышения стабильности модели. Временная сложность данной части равна  $O(K * S + a^2 * N_{label})$ , где  $N_{label}$  – количество размеченных пикселей.

---

### Алгоритм 2.2 – Алгоритм обучения интеллектуальной модели для задачи многометочной сегментации

---

```

1:  function net = trainModel (picture, labeledPicture, c, a)
2:      counter = 0
3:      points = zeros(2, sum (picture ~= labeledPicture))
4:      output = zeros (size (c, 1), sum (picture ~= labeledPicture))
5:      for i = 1 : size (c, 1)
6:          | pictureOfIthClass = (labeledPicture(:, :, 1) == c(i, 1))
                                & (labeledPicture(:, :, 2) == c(i, 2))
                                & (labeledPicture(:, :, 3) == c(i, 3))
7:          | [x, y] = find (pictureOfIthClass)
8:          | points(1 : 2, counter : counter + size (x)) = [x, y]
9:          | output (i, counter : counter + size (x)) = 1
10:         | counter = counter + size (x)
11:     end
12:     input = getNeighborhood (picture, points, a, 0) / 255
13:     net = patternnet (hiddenLayerSize)
14:     initParameters (net);
15:     net = train (net, input, output)
16: end

```

---

Во второй части (строки 13..15) происходит непосредственное обучение с помощью сопряжённого градиентного алгоритма обратного распространения ошибок [72], так как при нём происходит более эффективное использование используемой памяти, чем при алгоритмах типа Левенберга-Марквардта или байесовской регуляции. Особенность данного метода – выбор изменения параметров: оно выбирается таким образом, чтобы быть ортогональным к предыдущим направлениям. В качестве критерия оптимизации выбрана перекрёстная энтропия.

Исходное изображение при необходимости аугментируется за счёт добавления шума или выполнения прочих манипуляций с цветом для повышения стабильности работы интеллектуальной модели.

Временная сложность второй части зависит от выбранной интеллектуальной модели. Для выбранного примера она зависит от количества эпох обучения  $n_{\text{эпох}}$  и

количества нейронов на скрытом слое  $H$ :  $O(N * n_{\text{эпох}} * H * (a^2 + K))$ . Совокупная временная сложность:  $O(N * n_{\text{эпох}} * H * (a^2 + K) + K * S)$

В алгоритме 2.3 показан алгоритм проведения многометочной сегментации, который использует обученную ранее модель, на рис. 2.2 показан пример работы алгоритма (здесь и далее в главе приведены примеры для ЦТК, более полных их анализ приведён в главе 4). Входными параметрами являются исходное изображение, размер скользящего окна, интеллектуальная модель. Выходным параметром является результат сегментации, выраженный в виде  $K$  полутоновых изображений.

---

### Алгоритм 2.3 – Алгоритм многометочной сегментации

---

```

1:  function result = segmentation(picture, net, a)
2:      [nx, ny, ~] = size(picture)
3:      input = zeros(3 * a * a, nx * ny)
4:      paddedPicture = padarray(picture, [(a - 1) / 2, (a - 1) / 2])
5:      for i = 1 : a
6:          | for j = 1 : a
7:              | | shiftedPicture = paddedPicture(i : nx + i - 1, j : ny + j - 1, :)
8:              | | reshapedPicture = reshape(shiftedPicture, [3, nx * ny])
9:              | | shift = 3 * (a * i + j - a)
10:             | | input(shift - 2 : shift, :) = reshapedPicture
11:             | end
12:         end
13:         batch = 1 : floor((nx * ny) / (a * a)) : (nx * ny + 1)
14:         batch(numel(batch)) = nx * ny
15:         for i = 1 : numel(batch) - 1
16:             | batchInput = input(1 : 3 * a * a, batch(i) : batch(i + 1)) / 255
17:             | result(:,batch(i) : batch(i + 1)) = sim(net, batchInput)
18:         end
19:         result = reshape(result, [nx, ny, nClasses])
20:         result = result(result > 0.1)
21: end

```

---

Первая часть алгоритма (строки 2..12) отвечает за получения из исходного изображения входных данных для работы интеллектуальной модели. Это производится с выполнением смещения, которое лежит в интервале  $[-\frac{a-1}{2}; \frac{a-1}{2}]$  по обоим осям расширенного изображения (*paddedPicture*). Чтобы пиксели, которые лежат на границе ССИ, с меньшей вероятностью были отнесены к неправильному классу, то расширение стоит производить значениями, находящимися симметрично

границе. Затем данные преобразуются в форму (*reshapedPicture*), аналогичную используемой в алгоритме обучения.

Во второй части алгоритма (строки 13..20) происходит пакетная обработка входных данных, так как они имеют размер  $\Theta(S * a^2)$ , что может привести к переполнению кучи, и восстановление формы результирующих изображений.

Временная сложность алгоритма равна (для нейросети прямого распространения аналогична обучению, но без  $n_{\text{эпох}}$ )  $\Theta(S * a^2 + S * H * (a^2 + K)) = \Theta(S * H * (a^2 + K))$ .

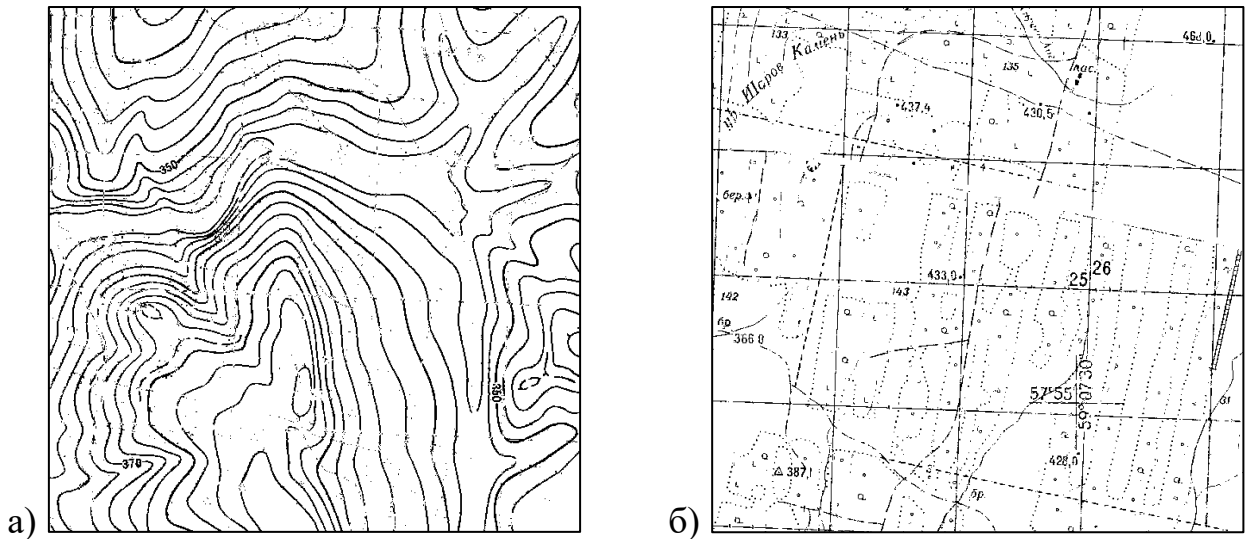


Рисунок 2.2 – Пример многометочной сегментации при обработке ЦТК:  
 а) полутоновое изображение для «коричневых» объектов;  
 б) полутоновое изображение для «чёрных» объектов

### 2.3 Локализация и определение типов сегментированных объектов

После выполнения предыдущих алгоритмов были получены полутоновые изображения, значения в которых означают меру принадлежности пикселя исходного изображения к определённому классу объектов. Необходимо разделить эти изображения на отдельные образы, т. е. произвести их локализацию в пространстве растра, подклассификацию (например, отделение текстовых меток одного размера, несущих одну функцию, от других) и выявить их более абстрактные свойства,

например, пространственный тип объекта (точечный, линейный, площадной), угол наклона (для ТО), толщину линий (для ЛО) и. т. д.

Из-за значительного отличия свойств объектов по пространственному типу невозможно выполнить их совместную обработку одним алгоритмом. Как показано в работе [170] при попытке улучшить качество распознавания ЛО в ТО вносятся искажения. В [225] показана сложность локализации меток на топографических картах из-за высокой степени пересечения объектов и низкого качества ЦТК в целом: точность локализации объектов для разработанного алгоритма составляет 85% с необходимостью большого числа операций, выполняемых оператором.

При этом стоит отметить топологическую связь между ТО и ЛО: зачастую ТО находятся прямо на месте прохождения ЛО (как на диаграммах с изображением нескольких функций с параметром или на картах городских дорог [169]), либо рядом (как на чертежах), и имеют одинаковый цвет (как на ЦТК). ПЛО же обычно выполняют роль фоновых объектов для ТО и ЛО, поэтому не сочетают в себе возможности пространственного пересечения и совпадения цветов с ними. Поэтому можно выделить два алгоритма: для локализации ТО и ЛО и для локализации ПЛО.

### 2.3.1 Алгоритм для локализации точечных и линейных объектов

Существует несколько подходов к отделению ТО и ЛО. Простейшие [67] опираются на контурный анализ: если произвести более строгую сегментацию (с бинарной картой на выходе, а не полутоновой), то ЛО можно отличить от ТО с помощью вычисления длины контура. Такой подход возможен при высоких значениях  $dpi$  и хорошей отделимости объектов в цветовом пространстве.

Гораздо более гибкие результаты получаются при использовании математической морфологии [112]. Минусом указанного исследования является то, что подбор морфологических операций в нём осуществляется непосредственно оператором. А так как на ССИ могут существовать ТО большого количества типов, то подбор операций для повышения автоматизации можно производить с помощью

генетических алгоритмов (ГА) [85]. В [261] реализована локализация и определение простейших типов объектов (квадраты, звёзды, диски) на пустом фоне для распознавания отсканированных нотных партий. В [242] ГА предлагается использовать для улучшения рентгеновских изображений с помощью дерева операций.

Для того, чтобы сформировать ГА, необходимо определить несколько основных понятий для решаемой задачи.

1. Ген – выполняемая операция над текущим изображением с некоторыми параметрами. В качестве операций рассматриваются:

- морфологические: дилатация, эрозия, замыкание, размыкание, скелетонизация – с различными СЭ и связностью, полутоновые и бинарные;

- пороговая – преобразует полутоновое изображение в бинарное через значение порога;

- фильтр по свойству бинарного объекта: площадь, эксцентриситет, размер большей или меньшей полуоси эквивалентного эллипса – данные операции могут выполняться только с бинарными объектами, поэтому им должна предшествовать операция пороговой обработки.

Так как ССИ представляют собой двумерное изображение, то при обработке обычно реализуется два вида связности: 4-связность (ещё называемая окрестностью фон Неймана) и 8-связность (окрестность Мура). 4-связность предполагает, что пиксель имеет 4 соседей: по одному с каждой стороны; 8-связность предполагает наличие к этим ещё и соседей по диагоналям. Под эквивалентным эллипсом [21] понимается эллипс, который имеет те же моменты инерции площади относительно  $x$  и  $y$  осей, как и бинарный объект.

2. Хромосома – упорядоченная последовательность операций-генов, выполняемых над исходным изображением последовательно. Можно рассматривать также более сложную систему, когда входными изображениями будут являться другие изображения, в том числе также полученные с помощью морфологической обработки, а операциями – бинарные (арифметические или логические сложения и умножения). Тогда хромосому можно будет представить в виде дерева операций.

Так как ГА сами по себе являются трудоёмкими в плане вычислительной сложности, то такое решение может ещё сильнее увеличить вариативность при поиске оптимальной хромосомы, поэтому в данном исследовании такую возможность мы не будем рассматривать.

Типичное исполнение ГА содержит генотип (в нашем случае он состоит из одной хромосомы) определённой длины. Для нашей задачи такая интерпретация не подходит, поэтому мы будем предполагать вариативность длины хромосомы.

Учитывая, что морфологические операции могут проводиться как с полутоновыми, так и с бинарными изображениями, а проведение нескольких пороговых операций не имеет смысла, то можно заключить, что в нашем случае хромосома будет иметь такой вид, что сначала идут только морфологические операции (назовём её левой частью хромосомы), одна пороговая (аналог центромеры), потом морфологические операции и фильтры (правая часть).

3. Функция приспособленности – критерий, по которому будет производиться селекция особей.

Для определения функции приспособленности отметим, что является оптимальным результатом выполнения операций по локализации и определению типов сегментов. По полутоновому изображению должно быть получено бинарное, в котором все объекты являются разделёнными в окрестности Мура: никакие пиксели любых двух объектов не имеют общей вершины. Таким образом, *в рамках одного класса* будет исключена возможность наложения или гетерархичности.

Визуальные объекты на изображении можно разметить с помощью областей, в качестве которых можно использовать минимальную выпуклую оболочку или другую, более удобную для разметки фигуру, например прямоугольник. Тогда функцию приспособленности можно выразить в виде суммы  $IoU$  от всех пар объект-фигура. Недостаток такой функции заключается в том, что если какие-то объекты будут потеряны, то максимизация такой суммы всё равно может быть произведена, но это будет только локальный оптимум.

Поэтому предлагается сначала производить селекцию по полноте распознавания, которую будем понимать, как количество найденных объектов, содержащих в себе критическую точку исходных объектов, например, центр размеченной фигуры. Если после изменения хромосомы данный параметр не уменьшается или находится на уровне не ниже 95% от максимального, то тогда данная хромосома может рассчитываться по параметру  $IoU$ . Функция селекции по полноте:

$$RecallGA(P') = [ |P' \wedge P_{crit}| \geq |P'_{\pi} \wedge P_{crit}| \text{ or } |P' \wedge P_{crit}| \geq 0.95 * |P_{crit}| ], \quad (2.4)$$

где  $P'$  – бинарное изображение, полученное в результате выполнения операций над оригинальным в текущей хромосоме,  $P_{crit}$  – бинарное изображение, где единицами отмечены центры размеченных фигур,  $P'_{\pi}$  – бинарное изображение, полученное для предка данной хромосомы (для изначальной популяции можно считать равной нулевому растру),  $[statement]$  – скобка Айверсона,  $|P|$  – мощность ненулевых пикселей бинарного изображения.

Таким образом, совокупную функцию приспособленности можно сформулировать так:

$$FitnessFunction(P') = RecallGA(P') * \frac{|P' \wedge P_{crit}|}{|P_{crit}|} * \frac{|P' \wedge P_{ideal}|}{|P' \vee P_{ideal}|}, \quad (2.5)$$

где  $P_{ideal}$  – размеченное бинарное изображение.

4. Генетические операторы – действия, которые будут производиться с генами для их модификации.

В классических ГА такими операциями являются скрещивание и мутация. Для нашей задачи операция скрещивания с произвольным перемешиванием генов не имеет смысла, так как здесь гены не являются независимыми по своей природе – каждый ген-операция выполняется строго после предыдущего, и функция приспособленности определяется именно для всей последовательности сразу. Поэтому

под скрещиванием понимается такое определение потомка-хромосомы, при котором от одного предка берётся левая часть, а от другого – правая:

$$Chromosome_{new} = [Chromosome_n(1:k_1), Chromosome_m(k_2:end)], \quad (2.6)$$

где  $Chromosome_n(1:k_1)$  – первые  $k_1$  генов  $n$ -й хромосомы-родителя,  $Chromosome_m(k_2:end)$  – гены, идущие после  $k_2$  из  $m$ -й хромосомы-родителя,  $[x, y]$  – оператор *append* для двух хромосом.

Для вычисления функции приспособленности от хромосомы после скрещивания берём предка с наибольшим значением данной функции.

В качестве мутации рассмотрим 3 варианта:

1. вставка гена-операции в середину (может быть также реализована вставка в начало или в конец) хромосомы, при этом выбирается только один родитель:

$$Chromosome_{new} = [Chromosome_n(1:k), Gene, Chromosome_n(k+1:end)]; \quad (2.7)$$

2. модификация случайного гена, например, изменение числовых параметров СЭМО, уровня порога или параметра фильтра;

3. удаление случайного гена, если он не является пороговой операцией.

Операция мутации с изменением СЭМО гена не рассматривается, так как, во-первых, такое изменение скорее уменьшит значение функции приспособленности (например, замена дилатации на эрозию приведет к исчезновению областей, что уменьшит пересечение с изображением критических точек), а, во-вторых, может быть заменена последовательным выполнением двух мутаций: удаления и вставки нового гена.

ГА для получения последовательности выполняемых операций можно записать в виде, представленном в алгоритме 2.4. Входными параметрами для алгоритма является полутоновое изображение после сегментации, бинарные изображения идеальной классификации отдельных сегментов и изображение с размеченными критическими точками. Выходным параметром является хромосома с



операциями, которую затем можно использовать для получения результирующего изображения.

---

### Алгоритм 2.4 – Генетический алгоритм получения последовательности операций для локализации

---

```

1:  function bestChromosome = geneticAlgorithm(picture, pictureIdeal, pictureCrit)
2:      parent = zeros(populationSize, 1)
3:      fitFunc = zeros(populationSize, 1)
4:      for i = 1 : populationSize
5:          | chromosome(i, :) = struct('name', "threshold",
6:                                     'properties', struct('name', "level", 'param', rand()))
7:      end
8:      fitFunc = fitnessFunction(chromosome, 1:populationSize, parent,
9:                               picture, pictureIdeal, pictureCrit)
10:     end
11:     [~, sortIdx] = sort(fitFunc, 'descend')
12:     chromosome = chromosome(sortIdx, :)
13:     fitFunc = fitFunc(sortIdx)
14:     iteration = 0
15:     while iteration < maxIteration
16:         | newChromosome = chromosome(1:ceil(populationSize * elitePart), :)
17:         | for i = ceil(populationSize * elitePart) + 1 : populationSize
18:             | | parent1 = ceil(rand() * populationSize * elitePart)
19:             | | if rand() < probCrossover
20:                 | | | parent2 = ceil(rand() * populationSize * elitePart)
21:                 | | | newChromosome(i, :) = crossover(chromosome(parent1, :),
22:                                                         chromosome(parent2, :))
23:                 | | | parent(i) = (fitFunc(parent1) > fitFunc(parent2)) * ...
24:                                     (parent1 - parent2) + parent2
25:             | | else
26:                 | | | newChromosome(i, :) = mutation(chromosome(parent1, :))
27:                 | | | parent(i) = parent1
28:             | | end
29:         | end
30:         | chromosome = newChromosome
31:         | fitFunc = fitnessFunction(chromosome, 1:populationSize, parent,
32:                                   picture, pictureIdeal, pictureCrit)
33:         | [~, sortIdx] = sort(fitFunc, 'descend')
34:         | chromosome = chromosome(sortIdx, :)
35:         | parent = parent(sortIdx)
36:         | fitFunc = fitFunc(sortIdx)
37:         | if fitFunc(1) > acceptableScore
38:             | | break
39:         | end
40:         | iteration = iteration + 1
41:     end
42:     bestChromosome = chromosome(1, :)
43: end

```

---

Используемые для алгоритма функции (*fitnessFunction*, *crossover*, *mutation*) и функция, которая строит по хромосоме результирующее изображение, и которую можно использовать в дальнейшем для получения оптимального результата, не приведены из-за большого количества команд, заключающихся в различной обработке различных операций ГА.

В начале алгоритма (строки 2..11) выполняется инициализация начальной популяции. Так как пороговая обработка – это единственная обязательная операция, так как полутоновое изображение должно быть преобразовано в бинарное для вычисления функции приспособленности, то изначально хромосомы всех особей состоят из неё, пороговый уровень вычисляется случайно.

Далее (строки 12..37) происходит итеративное выполнение ГА. Используется элитарная селекция (строки 14..25), для которой из всей популяции выбирается часть *elitePart*, которая остаётся на следующую итерацию и только от которой образуются потомки. С вероятностью *probCrossover* у пары родителей создаётся новый потомок, иначе у первого родителя происходит рождение нового потомка с мутацией.

Далее (строки 26..37) происходит вычисление функции приспособленности, хромосомы сортируются в убывающем порядке по ней для будущего формирования новой элиты особей. Если особь с лучшим показателем функции приспособленности преодолела необходимый уровень *acceptableScore*, то ГА останавливается. Если уровень не достигнут, то итерации продолжаются, пока не будет достигнуто максимальное их количество *maxIteration*.

Результат применения оптимальной хромосомы – бинарное изображение (маска локализации), содержащее в себе установленными пиксели, которые принадлежат областям, соответствующим некоторым объектам (пример на рис. 2.3). Таким образом, данный алгоритм позволяет локализовать места, в которых потом можно реализовать алгоритмы распознавания образов.

Определить временную сложность алгоритма для получения результирующих бинарных изображений заранее не представляется возможным, единственное,

что можно дать общую оценку  $\Theta(S * N_{genes}^2)$ , где  $N_{genes}$  – количество генов в оптимальной хромосоме. Время выполнения отдельных операций имеет сложную зависимость, например, для морфологических операций размер структурных элементов при прямом подходе к вычислению влияет на время квадратично  $O(S * M^2)$ , где  $M$  – характеристический размер СЭМО, но при разложении в последовательность нескольких операций начинает влиять линейно  $O(S * M)$  [25].

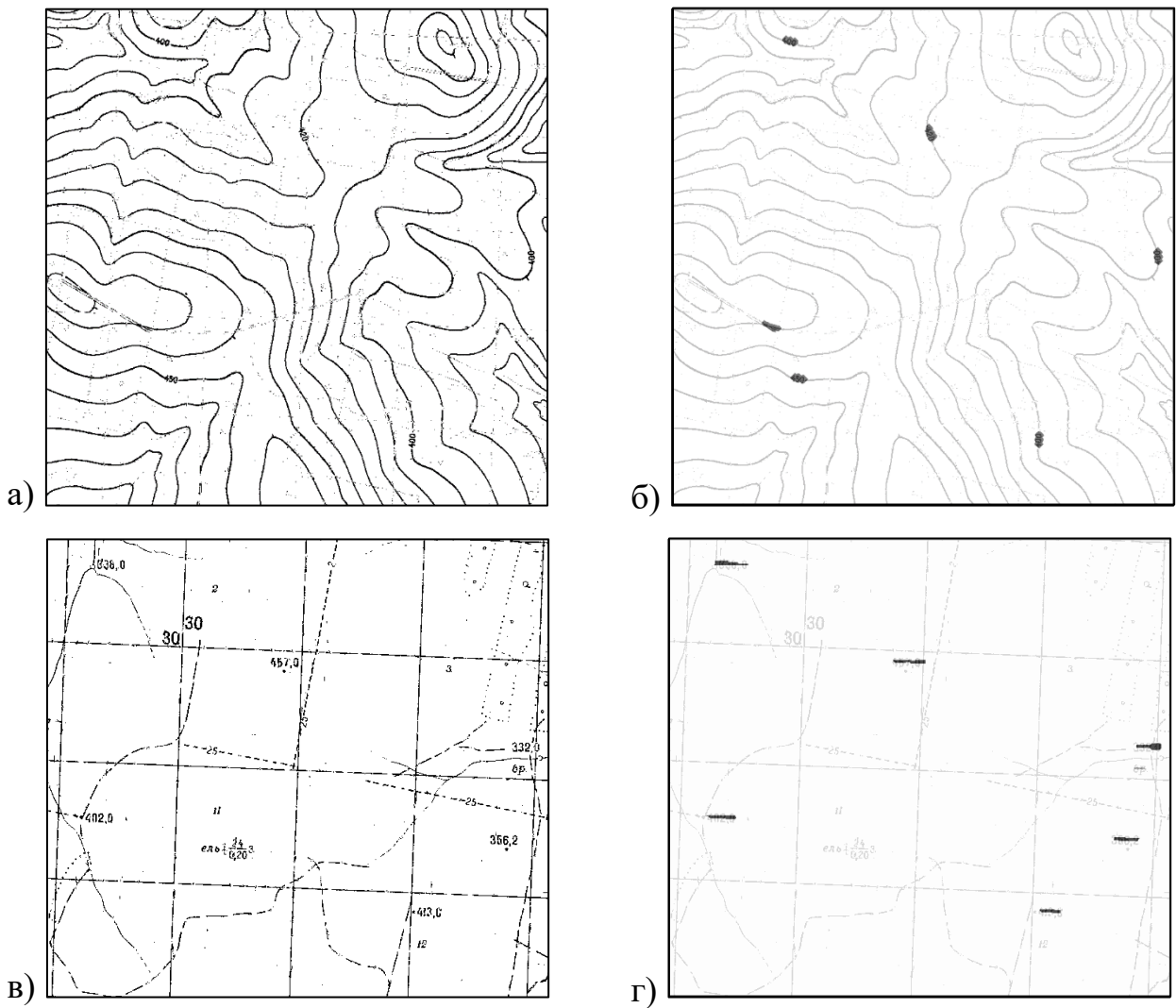


Рисунок 2.3 – Пример работы алгоритма локализации точечных объектов: а) исходное изображение для «коричневых» объектов; б) маска локализации отметок «поверх» исходного для сравнения; в) исходное изображение для «чёрных» объектов; г) маска локализации отметок «поверх» исходного для сравнения

Время работы операций фильтрации зависит от скорее от количества объектов после пороговой обработки, потому что их хранение выполняется с помощью отдельных свойств, поэтому можно сказать, что любая такая операция выполняется за  $O(S)$ , так как можно сказать, что количество объектов на изображении ограничено его площадью. Время для работы самого ГА можно оценить как  $O(S * N_{genes} * P * N_{iter})$ , где  $P$  – размер популяции,  $N_{iter}$  – количество итераций.

### 2.3.2 Алгоритм для классификации площадных объектов

Одним из признаков ССИ является нерегулярность, поэтому распознавание больших ПЛО становится проблемой, так как они могут быть представлены большим разнообразием форм и текстур.

Существует подход к распознаванию площадных объектов как точечных при достаточно больших по размеру обучающих изображениях, используя свёрточные нейронные сети. В [113] *CNN* производит распознавание текстур объектов с аэрофотоснимков в виде бинарного кодирования изображения; результаты при этом значительно ниже для распознавания объектов, обладающих скорее площадной, чем точечной или линейной топологией, например, 83% для парков, 67% для центра города, но 97% для виадуков, 95% для рек, 99% для парковок.

В [270] *CNN* играет роль фильтра при обработке старых топографических карт: со всего изображения выделяются именно изображения домов, что актуально для исторических исследований, так как позволяет автоматизировано изучать изменение городских пространств. *IoU* при этом в зависимости от выбранного метода и размера скользящего окна (от  $64 \times 64$  до  $224 \times 224$ ) варьируется от 56% до 92%.

Главным недостатком использования *CNN* является необходимость длительного обучения и скорость работы уже обученной модели. Видно, что результаты распознавания ПЛО далеки от аналогичных результатов при распознавании ТО. Поэтому можно сделать вывод об использовании других подходов, особенно, когда это касается изображений II и III класса.

Одним из подходов является использование гистограмм: при нём по скользящему окну высчитывается гистограмма (например, яркости для полутонового изображения) и далее делаются выводы о нахождении объекта. Недостатком данного подхода является то, что такие гистограммы по сути являются интегральными характеристиками, и в них могут проявиться свои особенности, если объекты расположены под разными углами. Например, в [184] для распознавания мотоциклов и велосипедов на городских фотографиях строятся отдельные дескрипторы, зависящие от их положения: горизонтального, вертикального или диагонального.

Хотя при отображении ПЛО на ССИ могут получаться значимые шумы, но в целом следует отметить статистическое постоянство в вариативности отклонения цвета и её частоте даже в небольших областях, что позволяет значительно ограничить размер скользящих окон в отличие от, например, *CNN*, для которых размер окон для поточечной классификации зачастую превышает сотни пикселей. Прототип разработанного алгоритма рассмотрен в [45].

Для решения задачи определения ПЛО на ССИ нужно очистить её от объектов, найденных на этапе обработки ТО и ЛО (назовём множество их пикселей  $P_{T\&L}$ ), которые изображаются поверх ПЛО, которые являются для них фоном. Для таких пикселей невозможно в принципе определить, что находилось под образами переднего визуального плана, но можно сделать предположение о некоторой континуальности и протяжённости объектов заднего фона. Данное свойство можно назвать *принципом локальности*, который можно выразить в существовании такой функции  $f_{local}$  (формула 2.8), определяющей класс пикселя по классам пикселей его соседей и удовлетворяющей следующему условию для почти каждого пикселя  $p$ . Под почти каждым пикселем здесь понимается, что данное условие может не выполняться для  $O(1)$  пикселей для каждого площадного объекта.

$$class(p) = f_{local}(\{class(p') \mid d(p, p') \leq d_{max}\}), \quad (2.8)$$

где  $class(p)$  – класс фонового изображения текущего пикселя  $p$ , определяемого через свои координаты растра,  $p'$  – пиксели, для которых выполняется условие соседства,  $d$  – метрика, по которой считается расстояние между двумя пикселями,  $d_{max}$  – максимальное значение данной метрики.

В качестве простейшего примера  $f_{local}$  можно воспользоваться функцией моды классов пикселей в окрестности, расстояние Чебышёва которой до центра не больше определённого значения. Тогда нарушение принципа локальности будет происходить, например, когда размеры ПЛО меньше  $2 * d_{max}$ .

Так как класс ПЛО фонового слоя определяет его визуальное отображение, то для устранения разрывов, образованных удалением  $P_{T\&L}$ , воспользуемся фильтром Гаусса над растром, при этом пиксели, которые не принадлежат  $P_{T\&L}$  сохраняют свой цвет, столько раз, пока изменения не перестанут происходить. Так как ЛО и ТО имеют небольшие размеры, то достаточно конечного количества данных итераций.

Площадные объекты характеризуются скорее своими цветами и простейшей текстурой, то окно вокруг интересующего пикселя можно охарактеризовать набором идеальных цветов в нужной пропорции. Каждому классу ПЛО соответствует 3 гистограммы (по каждой из цветовых координат), что и является признаковым описанием цвета. Данный подход позволяет обрабатывать не только одноцветные объекты, но и многоцветные, например, изображение болот на ЦТК. Тем не менее, данный подход не позволяет учитывать при распознавании относительное расположение пикселей, например, по гистограмме не будет видно, под каким углом выполнена штриховка.

Поточечную классификацию можно производить с использованием различных моделей. При этом может произойти ситуация, что пиксель не отнесётся ни к одному из существующих классов. Вторая ситуация – классификация с достаточно большой выходной зашумлённостью, когда рядом находящиеся пиксели будут отнесены к различным классам, хотя семантически относятся к одному объекту.

Пропущенные пиксели необходимо обработать, используя итеративный процесс наращивания уже классифицированных пикселей, находя моду среди окрестности пикселей. Если частота встречи моды больше критического значения, то пиксель классифицируем, если нет, то пропускаем для обработки в следующей итерации. Критическое значение уменьшаем с увеличением номера итерации.

Разработанный алгоритм представлен на алгоритме 2.5, пример представлен на рис. 2.4. Входными параметрами являются исходной цветное изображение, бинарное изображение  $P_{T\&L}$ , гистограммы классов ПЛО с весовыми коэффициентами по столбцам и размер скользящего окна. Выходным параметром является изображение, в котором каждому пикселю поставлен в соответствие один класс ПЛО.

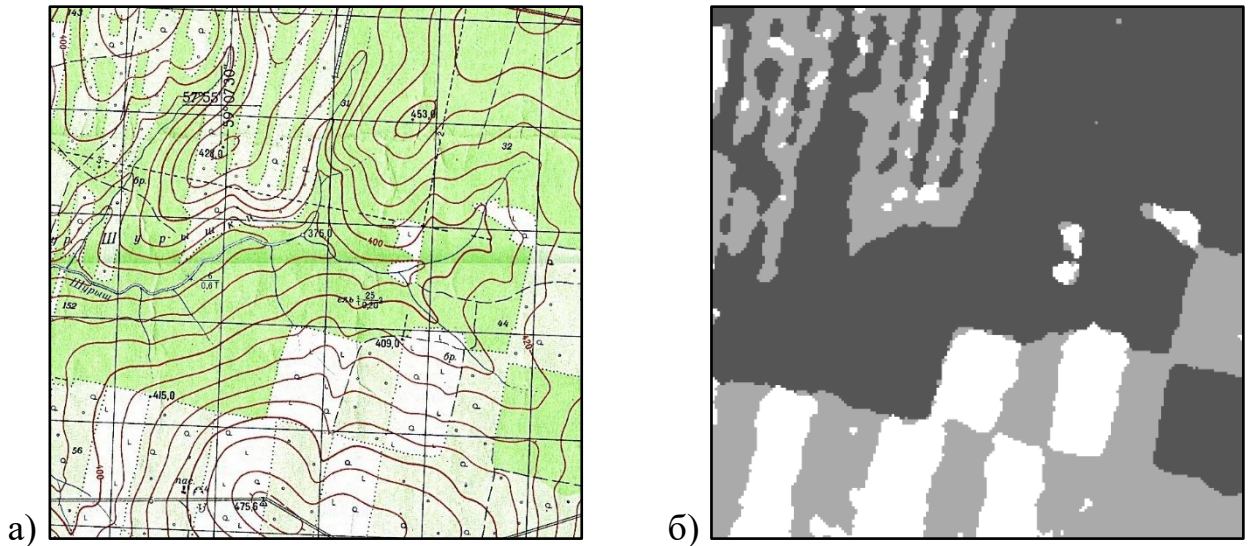


Рисунок 2.4 – Пример классификации площадных объектов:  
 а) исходное цветное изображение; в) изображение с тремя классами площадных объектов: «лес», «низкорослая растительность», «редколесье»

В первой части алгоритма (строки 2..14) происходит первичная классификация площадных объектов, используя цвет пикселей. Её временная сложность равна  $O(S * (G * N_{recon} + a^2 + bins * N_{cl}))$ , где  $G$  – размер фильтра Гаусса,  $N_{recon}$  – количество итераций размытия в функции восстановления,  $N_{cl}$  – количество классов ПЛО. Временная сложность данной части зависит от используемой интеллектуальной модели.

Во второй части алгоритма (строки 15..28) происходит подобные операции, но в качестве «изображения» используется найденное на предыдущем этапе изображение с размеченными классами. Далее процесс запускается итеративно, пока все точки не будут классифицированы модальным значением классов соседних пикселей. Каждый раз в процессе участвуют только те пиксели, которые ещё не были классифицированы, поэтому можно сказать, что временная сложность этого этапа ограничена  $O(S' * (a^2 + N_{cl}) * N_{iter})$ , где  $S'$  – количество неклассифицированных пикселей после первого этапа алгоритма,  $N_{iter}$  – число итераций второго этапа. Так как окончательно классифицируются пиксели, находящиеся на граничном слое нераспознанных областей, то на каждой итерации количество рассматриваемых пикселей сокращается на  $O(a^2 * \sqrt{S'})$ , поэтому число итераций тоже можно оценить как  $N_{iter} = O(\sqrt{S'})$ .

---

#### Алгоритм 2.5 – Алгоритм классификации площадных объектов

---

```

1:   Function result = classAreal (picture, picturePAndL, model, a)
2:     picture = backGroundReconstruction (picture, picturePAndL)
3:     nx = size (picture, 1)
4:     ny = size (picture, 2)
5:     result = zeros (nx, ny)
6:     x = reshape ( repmat (1 : nx, [1, ny]), [1, nx * ny])
7:     y = reshape ( repmat (1 : ny, [nx, 1]), [1, nx * ny])
8:     points = [x; y]
9:     nbhd = getNeighborhood (picture, points, a)
10:    edges = 0 : binwidth : 256
11:    rhist = histc (nbhd (1 : 3 : 3 * a * a, :), edges) / (a * a)
12:    ghist = histc (nbhd (2 : 3 : 3 * a * a, :), edges) / (a * a)
13:    bhist = histc (nbhd (3 : 3 : 3 * a * a, :), edges) / (a * a)
14:    result = predict (model, [rhist; ghist; bhist])
15:    iteration = 1
16:    while numel (find (result == 0)) > 0
17:      | points = find (result == 0)
18:      | nbhd = getNeighborhood (result, points, a)
19:      | histogram = histc (nbhd, 0 : size (hClass, 3))
20:      | histogram = histogram (2 : end, :)
21:      | for i = 1 : size (points, 2)
22:        | | [counter, class] = max (mistogram (:, i))
23:        | | if counter > (1 - iteration / 10) * a * a
24:        | | | result (points (1, i), points (2, i)) = class
25:        | | end
26:        | end
27:      | iteration = iteration + 1
28:    end
29:  end

```

---



## 2.4 Распознавание образов и их группировка

Данный этап обработки ССИ характеризуется большим влиянием специфичности, чем предыдущие. Если предыдущие разработанные алгоритмы можно считать альтернативами для существующих алгоритмов обработки изображений в целом, то уже здесь характеристики каждого отдельного типа ССИ могут оказывать достаточное влияние, что появляется необходимость для разработки отдельных алгоритмов или их вариаций.

Тем не менее предполагается, что возможно создание алгоритмов, которые будут давать хорошие результаты по качеству для широкого класса ССИ, если учесть большее число их проявлений.

### 2.4.1 Алгоритмы для точечных объектов

Более формальные и жёсткие подходы к распознаванию образов на ССИ характерны для более простых образов. В [207] используется обобщённое преобразование Хафа; минусом данного исследования является то, что даже для очень маленьких изображений (порядка 10 Кп) время обработки составляет порядка секунд. В [166] рассчитываются дескрипторы функции автокорреляции, входом которой является массив изменений координат границ символа; даже для документов с разрешением 300 *dpi* такой подход обеспечивает точность распознавания от 80% для отдельных типов символов. В [231] распознавание символов сводится к определению круга, оптимально покрывающего символ; хотя данный алгоритм годится только для распознавания объектов, представляющих собой небольшие выпуклые символы, в исследовании также приведён пример их группировки для восстановления структуры сетки расположения этих объектов (например, изображения кустов на ЦТК, содержащих сады).

Более интеллектуальные методы, использующие готовые *OCR* решения (например, *Google Tesseract*, в т. ч. встроенный в *Computer Vision Toolbox* пакета

*MatLab*) не всегда показывают лучший результат. Сам по себе *Tesseract* не является достаточно гибким инструментом для работы с ССИ, так как требует, чтобы символы были минимум 20 пикселей в высоту (что возможно только для документов начиная с 300 *dpi*), не допускаются даже незначительные повороты символа, сдвиги или размытия [255]. Его использование без дополнительной предобработки приводит к качеству порядка 77-80% для ССИ, имеющих либо пересечения ТО с другими объектами [213], либо некоторые вариации шрифтов [228].

Использование свёрточных нейронных сетей и глубокого обучения тоже не является оптимальным решением. Использование предобученных нейросетей, например, *ResNet-50* в [114] или *VGGNet* для локализации и *Tesseract* для распознавания в [195] также дают спорные результаты. В первом случае это 65-70% для ТО, которые могут пересекаться с другими объектами, во втором случае 87-94% для ТО, которые не пересекаются с другими объектами (из-за особенностей стандартов *USGS US Topo* для изображения ЦТК). В этих случаях данными также являлись документы с разрешением 300 *dpi*.

Прототипы предлагаемого алгоритма распознавания ТО на примере цифр рассмотрены в статьях [30, 31, 34]. Рассмотрение распознавания цифр объясняется как их возможностью внутрифункциональной группировки в числа и невозможностью проведения анализа по словарю, что могло бы облегчить распознавание, например, слов. Алгоритм распознавания без последующей группировки может быть применён для отдельно стоящих символов.

Учитывая, что ИНС прямого распространения с одним скрытым слоем и прочие одноэтапные интеллектуальные модели не могут показать достаточных результатов распознавания на топографических картах, а многослойные (в т. ч. свёрточные) нейросети показывают значительное уменьшение быстродействия, будем ориентироваться на двухкаскадные модели.

Центры распознаваемых образов можно получить из найденных ранее бинарных изображений, на которых имеются области локализации объектов. Данные области не соответствуют напрямую объектам: с одной стороны, они могут не

включать в себя цифр целиком, а с другой стороны могут пересекаться с несколькими цифрами (и другими объектами) одновременно. Морфологическая обработка была осуществлена так, что большинство центров цифр попало в эти сегменты.

Используя изображения скользящих окон с центрами во множестве точек этих областей, обучим первый каскад интеллектуальной модели. Размеры скользящих окон в данном случае отличаются от этапа сегментации, так как распознаваемые объекты (например, цифры) могут являться вытянутыми объектами (для цифр отношение высоты к ширине равно примерно 2).

Выходом интеллектуальной модели является мера уверенности в том, что в текущем скользящем окне находится объект одного из  $K$  классов. Таким образом, мы осуществили операцию, аналогичную многометочной сегментации, проведённой ранее, и одно бинарное изображение было преобразовано в  $K$  полутоновых изображений. Учитывая, что на ССИ может быть использовано большое число классов, то следует реализовать пороговое отсечение и хранить матрицы в разрежённом формате.

Далее необходимо разбить получившиеся области, чтобы они соответствовали разным разрядам числа. Осуществить это можно, проведя дополнительную операцию по фильтрации с использованием функции заполненности по типу:

$$fullness(x_c, y_c, k) = 1 - 4 * \left[ \frac{\sum_{(x,y) \in S(x_c, y_c)} measure(x, y)}{averagefullness(k)} - 1 \right]^2, \quad (2.9)$$

где  $(x_c, y_c)$  – центр скользящего окна,  $k$  – класс объекта,  $S(x_c, y_c)$  – множество точек скользящего окна с центром в указанной точке,  $measure(x, y)$  – значение меры принадлежности к обобщённому классу объекта,  $averagefullness(k)$  – среднее значение заполненности для данного класса объектов.

В качестве значений растра  $measure(x, y)$  можно взять значения из полутонового изображения, полученного после сегментации по обобщённому классу. Итоговое значение  $fullness$  также можно модифицировать, используя

интеллектуальную модель. Данный подход используется для минимизации ложного срабатывания на посторонние объекты, которые точно не относятся к искомому классу. Например, помимо ИНС для распознавания отдельных цифр нужна также и ИНС для получения меры уверенности в том, что в данном скользящем окне цифра вообще есть. Хотя такая информация будет иметь значительную корреляцию, но она позволит отсеять цифроподобные объекты.

Значение функции заполненности должно быть близко к 1 для центра цифры и приближаться к 0 в точках между цифрами. С помощью пороговой обработки можно осуществить разделение и выделить сегменты, которые будут точно соответствовать определённой цифре, что уменьшит влияние расположения объектов рядом для улучшения качества работы второго каскада интеллектуальной модели. Такое разделение позволит разделить цифры лишь на визуальном плане: соседние пиксели могут быть распознаны как центры различных цифр.

Назовём такие сегменты, которые состоят из пикселей, отнесённых к разным классам объектов, *суперсегментами*. Формально суперсегмент можно определить как непрерывное множество пикселей на индексированном изображении, где значение цвета пикселя определяется индексом класса. Например, суперсегменты, соответствующие цифре 3, содержат пиксели, воспринимаемые, как центры двоек, троек и пятёрок. Это происходит потому, что участки некоторых цифр, особенно с учётом искажений, похожи на участки других цифр (например, нижняя часть цифры 3 похожа на верхнюю часть цифры 2 и на нижнюю часть цифры 5), поэтому методы распознавания могли осуществить ошибку в классификации. Данная ошибка возникает в любом случае, особенно на ССИ, поэтому существование дополнительной обработки является необходимым.

Для того чтобы увеличить точность, необходимо выбрать правильное значение из множества значений суперсегментов. Для выбора будем использовать меры уверенности в том, что пиксель является центром цифры, полученные на предыдущем этапе, и расстояние от пикселя до центра суперсегмента. Всего на входе второго каскада будет десять параметров, соответствующие вычисленным мерам, на

выходе будет вероятность того, что данный суперсегмент соответствует одной из 10 цифр. При этом происходит свёртка суперсегмента, состоящего из нескольких точек с множеством значений в одну точку с единым значением. Отметим, что вторым каскадом может быть и не интеллектуальная модель, а статистическая, например функция моды от значений номеров классов в суперсегменте.

Для усиления влияния локальности вход второго каскада откорректируем с использованием расстояния пикселя до центра суперсегмента:

$$input_2(i, n) = \sum_{(x, y) \in S(i, n)} \frac{output_1(x, y)}{1 + k_R * D((x, y), (x_{ci}, y_{ci}))}, \quad (2.10)$$

где  $S(i, n)$  – множество пикселей с координатами  $(x, y)$   $i$ -ного суперсегмента, соответствующих классу  $n$ ,  $output_1(x, y)$  – выходное значение меры класса  $n$  для точки  $(x, y)$ ,  $(x_{ci}, y_{ci})$  – координаты центра  $i$ -го суперсегмента,  $D((x, y), (x_{ci}, y_{ci}))$  – евклидово расстояние между текущей точкой и центром,  $k_R$  – коэффициент учёта расстояния. После вычисления входные параметры нормализуются.

Стоит отметить отличие двухкаскадного алгоритма от ИНС с двумя слоями. В разработанном алгоритме на вход первого каскада идут изображения скользящих окон со всего пространства растра, а на второй каскад уже только данные, получаемые от суперсегментов без скользящих окон. Оба каскада обучаются с использованием одинаковых выходных данных. Общий алгоритм распознавания цифр представлен в алгоритме 2.6, а пример обработки приведён на рис. 2.5.

Входными параметрами алгоритма являются полутонное изображение, полученное после этапа сегментации, бинарное изображение, полученное после этапа локализации, размеры скользящего окна, средние значения заполненности символов в классе, интеллектуальные модели обработки данных. Выходным параметром является множество распознанных образов, заданных четвёркой  $(x, y, \varphi, v)$ , где первые три значения отвечают за положение в пространстве растра и угол поворота ТО, последнее значение является идентификатором класса.

---

**Алгоритм 2.6 – Алгоритм двухкаскадного распознавания образов**


---

```

1:  Function result = recognition(segPicture, locPicture, a, avrFullness)
2:      ax = a(1)
3:      ay = a(2)
4:      label = bwlabel(locPicture)
5:      stats = regionprops(label, 'Orientation')
6:      phi = [stats.Orientation] * pi() / 180
7:      [x, y] = find(locPicture)
8:      nPoints = numel(x)
9:      points = [x, y]'
10:     input2 = zeros(ax * ay, nPoints)
11:     measure = zeros(ax * ay, nPoints)
12:     nClasses = numel(avrFullness)
13:     output = zeros(nClasses, nPoints)
14:     for i = 1 : nPoints
15:         | input(:, i) = getNeighborhood(segPicture, points(:, i),
                                     [ax, ay], phi(label(x(i), y(i))))
16:         | measure(:, i) = sum(input(:, i))
17:     end
18:     fullness = 1 - 4 * (measure ./ avrFullness - 1) .^ 2
19:     filter(1 : nPoints) = sim(filternet, input)
20:     output(:, 1 : nPoints) = sim(firstnet, input)
21:     multiPicture = zeros(size(locPicture, 1), size(locPicture, 2), nClasses)
22:     for i = 1 : nPoints
23:         | multiPicture(x(i), y(i), :) = output(:, i) .* ...
                                     filter(i) .* fullness(:, i)
24:     end
25:     for i = 1 : nClasses
26:         | multiPicture(:, :, i) = bwareaopen(multiPicture(:, :, i) > 0.5, 2)
                                     .* multiPicture(:, :, i)
27:     end
28:     [value, indexedPicture] = max(multiPicture, [], 3)
29:     indexedPicture = indexedPicture .* (value > 0)
30:     indexedPicture = filter(indexedPicture)
31:     [sSegment, nSS] = bwlabel(indexedPicture > 0, 8)
32:     stats = regionprops(sSegment > 0, 'Centroid',
                         'PixelIdxList', 'PixelList')
33:     cntr = [stats.Centroid]
34:     xc = round(cntr(1 : 2 : end))
35:     yc = round(cntr(2 : 2 : end))
36:     input2 = zeros(nClasses, nSS)
37:     for i = 1 : nSS
38:         | idx = stats(i).PixelIdxList
39:         | pt = stats(i).PixelList
40:         | for j = 1 : numel(indexes)
41:             | | input2(indexedPicture(idx(j)), i) =
42:                 | | input2(indexedPicture(idx(j)), i) +
43:                 | | value(idx(j)) / (1+kR * ((xc(i)-pt(j,1))^2 + (yc(i)-pt(j,2))^2)^0.5)
44:         | end
45:         | input2(:, i) = input2(:, i) / sum(input2(:, i))
46:     end
47:     output2 = sim(secondnet, input2)
48:     [~, output2] = max(output2, [], 1)
49:     result = struct('x', num2cell(xc), 'y', num2cell(yc),
                    'phi', num2cell(phi(L(xc, yc))), 'v', num2cell(output2))
50: end

```

---

В строках 7..24 происходит обработка и расчёт первым каскадом. В строках 25..30 происходит формирование индексированного изображения на основе сформированных полутоновых изображений многометочной сегментации и фильтрация для разделения образов. В строках 31..49 происходит обработка суперсегментов вторым каскадом. Временная сложность алгоритма без учёта выбора интеллектуальной модели соответствует  $\Theta(N_{points} * a_x * a_y + S * K)$ .

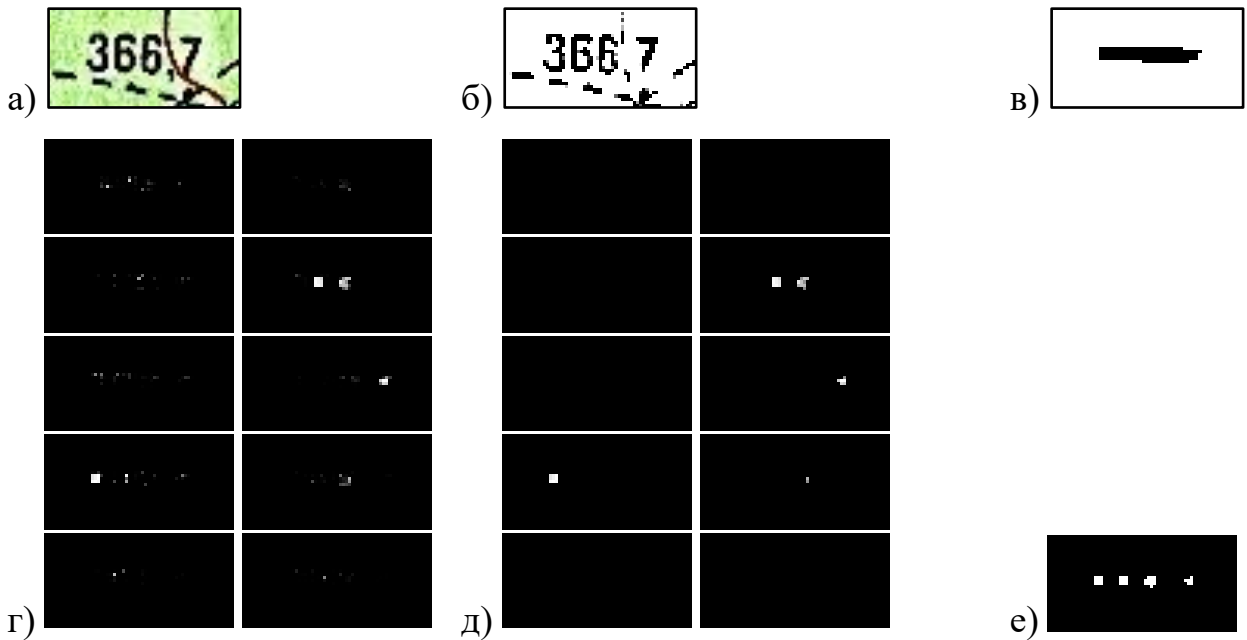


Рисунок 2.5 – Пример распознавания точечных объектов:

- а) исходное изображение; б) изображение после многометочной сегментации для класса «число»; в) бинарное изображение локализации образа;  
 г) выход первого каскада интеллектуальной модели (левая колонка: цифры 0-4, правая: 5-9); д) выход первого каскада интеллектуальной модели с учётом заполненности; е) индексированное изображение суперсегментов

После распознавания отдельных разрядов их необходимо сгруппировать. Учитывая, что может произойти ложное срабатывание на классы других ТО, то необходимо провести группировку, которая отфильтрует ненужные группы объектов. Такую фильтрацию можно произвести с помощью кластерного анализа, исследовав количество цифр в кластере, их взаимное расположение и свойства. Отличие от кластерного анализа в классическом понимании заключается в том, что при группировке символов в слова или цифр в числа становится важен порядок.

В общем случае группировка выполняется агломеративными и дивизионными методами, при которых образы группируются с учётом иерархии и специфики формирования групп ТО. В нашем случае рассмотрим решение задачи группировки последовательно расположенных образов без более сложной внутренней структуры внутри группы.

Сначала осуществим первичную группировку с помощью кластеризации (алгоритм 2.8). Будем называть ось, вдоль которой располагаются цифры, большой, а перпендикулярную ей малой. Основу решения задачи последовательной группировки можно затем использовать при решении более сложной задачи, меняя местами большую и малую ось (если ТО в группе располагаются по двумерной сетке) или используя дополнительные оси (если ТО имеют более сложное расположение в группе, например, радиальное вокруг общего центра или вдоль кривой [135]).

Рассматривая некоторую текущую цифру, будем смотреть, с каким уже сформировавшимся кластером выполняются условия близости вдоль осей. При этом вдоль большой оси можно предполагать большие отклонения (порядка 30-40 пикселей), а вдоль малой оси небольшие (2-3 пикселя). Данные значения, как и указанные ниже в этом параграфе, зависят от разрешения изображения ( $dpi$ ), поэтому при обработке ССИ данный параметр необходимо указывать. Если ни одного кластера с выполнением условий не найдено, то это означает, что эта цифра принадлежит к новому кластеру.

Для решения задачи кластеризации в целом необходимо произвести сравнение всех объектов попарно, поэтому временная сложность таких алгоритмов должна соответствовать  $\Theta(n^2)$ , где  $n$  – число объектов. В алгоритме 2.7 представлен случай, если ТО на ССИ изображены равномерно, что встречается в большинстве случаев. Тогда для ускорения можно разделить изображение на  $N_{bucket}$  горизонтальных полос (геометрическое хеширование), и если хранить информацию о содержимом всех полос, то временную сложность можно сократить до  $\Theta(n^2/N_{bucket})$ . Если осуществить подобную операцию и для вертикальных полос, и искать необходимые кластеры только в пересечении нужных «ячейках» сетки, то



временная сложность зависит от способа выполнения операции пересечения, при хешировании в среднем случае составляет  $O(n^2 / N_{bucket}^2)$ .

---

### Алгоритм 2.7 – Алгоритм первичной кластеризации

---

```

1:  Function group = primaryGrouping(digit, nBucket)
2:      nDigits = size(digit, 2)
3:      maxX = max(digit.x)
4:      bucket = cell(nBucket, 1)
5:      nGroups = 0
6:      for i = 1 : nDigits
7:          | flag = false
8:          | k = ceil(Digit(i).x / maxX * nBucket)
9:          | k = (k==1)*2 + (k > 1 && k < nBucket)*k + (k==nBucket)*(nBucket-1)
10:         | for j = [bucket{k - 1} : k + 1]
11:         | | ro = ((digit(i).x-group(j).x)^2 + (digit(i).y-group(j).y)^2)^0.5
12:         | | psi = atan2(digit(i).y-group(j).y, digit(i).x-group(j).x)
13:         | | if abs(digit(i).phi - group(i).phi) < pi() / 4 &&
14:         | | | (abs(ro * cos(psi - phi)) <= 20) &&
15:         | | | (abs(ro * sin(psi - phi)) <= 5) && ~flag
16:         | | | group(j).childs = group(j).childs + 1
17:         | | | group(j).ids(group(j).childs) = i
18:         | | | group(j).v(group(j).childs) = digit(i).v
19:         | | | group(j).x = mean(digit(group(j).ids).x)
20:         | | | group(j).y = mean(digit(group(j).ids).y)
21:         | | | flag = true
22:         | | | break
23:         | | end
24:         | end
25:         | if ~flag
26:         | | nGroups = nGroups + 1
27:         | | group(nGroups) = struct('childs', 1, 'x', digit(i).x,
28:         | | | 'y', digit(i).y, 'phi', digit(i).phi,
29:         | | | 'ids', [i], 'v', [digit(i).v])
30:         | | end
31:     end
32: end

```

---

После первичной кластеризации можно выполнить ряд операций фильтрации, применимость которых зависит от специфики ССИ. Рассмотрим несколько различных вариантов, которые могут встречаться.

При наличии дополнительных ограничений на количество ТО в кластере можно также произвести фильтрацию небольших кластеров, состоящих менее, чем из минимального числа объектов, так как такие кластеры, скорее всего, являются наборами из ложно определённых объектов. Например, при обработке ЦТК не может существовать подписи горизонтали, состоящей из одной цифры, а при

обработке карт горных районов минимальное количество можно положить равным трём.

Так как некоторые числа могут не объединиться в один кластер, но нужно провести дополнительную операцию агломерации кластеров с помощью похожей операции, которая проводилась для объединения цифр в кластеры, но для центров кластеров, а не цифр. После этой операции можно выполнить фильтрацию слишком больших кластеров, так как они представляют собой различные последовательные образования, не относящиеся к решаемой задаче. Например, в [195] исследователи отказались от использования ограничивающих прямоугольников слов, имеющих отношение длины к высоте большее, чем 5:1, так как иначе при незначительном улучшении точности распознавания, происходило значительное увеличение вычислительной нагрузки.

В основном «ложные» ТО представляют собой различные линии, которые расположены рядом (похожая проблема также отмечена в [213] без предложенного решения), которые распознающий каскад воспринял, как цифры. Обычно это нули или единицы, так как естественные линии и другие объекты представляют собой либо прямые линии, либо небольшие замкнутые кривые, похожие на окружность. Также «ложные» цифры могут возникнуть в областях между соседними настоящими цифрами, у таких цифр обычно очень маленькое значение площади.

Исходя из этого, проведём фильтрацию начальных «нулей» (если не имеется возможность существования начальных нулей в записи числа) и фильтрацию слишком близких цифр друг к другу. В основном, настоящие цифры находятся на расстоянии 8-9 пикселей (или 11-13 в случае разделения запятой) друг от друга. Если две цифры находятся слишком близко друг другу, например, на расстоянии не больше 5 пикселей, то это означает, что как минимум одна из них является ложной.

Определить порядок цифр можно при известном угле поворота числа с помощью сортировки вдоль большой оси. Так как количество цифр в одном числе ограничено сверху небольшой константой, что можно считать, что эта операция для одной группы выполняется за  $O(1)$ .

## 2.4.2 Алгоритмы для линейных объектов

Аналогично распознаванию и группировке ТО, линейные объекты также могут обрабатываться как более формальными, так и более интеллектуальными методами. При этом появляется специфика, которая не свойственна ТО: ЛО могут быть достаточно большими на области ССИ, поэтому могут претерпевать большое количество разрывов, но при этом само их распознавание является достаточно простой задачей: отличить сплошную линию от штриховой, пунктирной или какой-то другой гораздо проще, чем осуществить распознавание различных вариантов ТО, для этого можно воспользоваться алгоритмом, аналогичном рассмотренному в параграфе 2.3.2. Поэтому здесь будем считать, что ЛО в семантическом плане представляют собой сплошные линии.

Существует два различных подхода к восприятию линий: в первом они обладают некоторой двумерной природой (например, имеют ширину или толщину, которая может быть текстурирована), во втором – представляют собой скорее одномерную сущность с единичной шириной. Первый подход не даёт значительного преимущества при обладании большей информации об объекте – зачастую данная информация является скорее зашумлённой, что вынуждает использовать либо очень простые методы, типа пиксельного трекинга [131], либо сложные методы, например, *FCNN* [138]. Минус данного подхода заключен в том, что если рассматривать линейный объект двумерно, то тогда осложняется группировка таких объектов, так как они могут иметь вариативную структуру. Поэтому обычно даже двумерные ЛО вначале подвергают скелетизации, т. е. утончение или выделение остова. При скелитизации по окрестности Мура образуются линии, состоящие из меньшего количества пикселей, чем по окрестности фон Неймана. Хранение и обработка таких изображений занимает меньше времени при незначительном уменьшении точности, поэтому является предпочтительным.

Второй подход гораздо более развит в современной научной литературе, так как позволяет разработать алгоритмы, использующие широкий спектр

математических методов без использования интеллектуальных моделей. Отказ от использования интеллектуальных моделей здесь обосновывается тем, что решение задачи группировки для ЛО не обладает свойством локальности: невозможно по существующему изображению скользящего окна определить именно эти линейные сегменты нужно соединить, или другие.

Основные концепции и проблемы группировки ЛО рассмотрены в [234, 170]. Отмечена возможность ложного соединения ЛО на ССИ, когда линии, которые должны быть разделены в семантическом плане, на визуальном плане «склеиваются». И, наоборот, имеются разрывы типа X, когда в одном месте сходятся две кривые и возникает неразрешимая не только локально, но и глобально задача: какие две пары следует соединить (см. пример на рис. 2.6)?

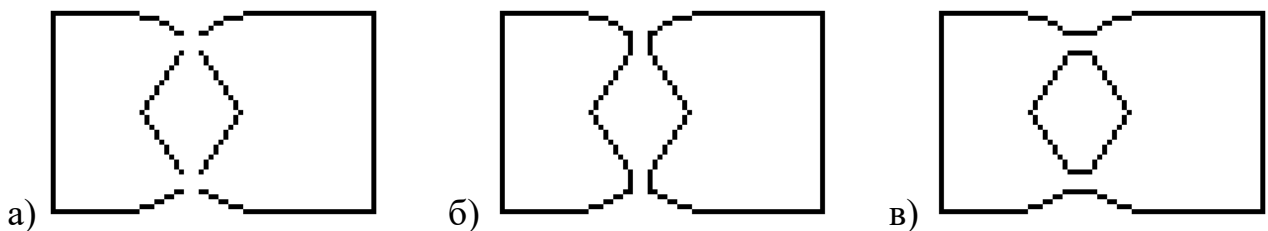


Рисунок 2.6 – Пример неразрешимости задачи устранения разрывов ЛО:

а) исходное изображение;

б) первый вариант соединения; в) второй вариант соединения

Однако, если исключать из рассмотрения подобные проблемы, которые могут быть разрешены только зная специфику ССИ, то разрывы типа X хоть и не устранимы локально, но зачастую могут быть разрешены, если взглянуть на ЛО в целом.

Отсюда можно сделать вывод, что строго геометрический подход, который рассматривает только концевые участки ЛО и их свойства не может решить задачу полностью, даже при значительном количестве правильных соединений на всём изображении. Основные идеи геометрического подхода рассмотрены в [248, 263, 267], они опираются на вычисление расстояний и согласовании углов между соединяемыми участками в скользящем окне. Разработанные алгоритмы носят итеративный характер, что приводит к увеличению времени работы алгоритмов в целом.

Противоположный подход рассматривает линии в целом и применяется для всего изображения, при этом линии воспринимаются как множество точек, которые вообще не обязаны быть связанными. Так, соединять линейные сегменты (под линейными сегментами подразумеваются криволинейные, так как для нас более важна «одномерная» природа этих сегментов) можно с использованием триангуляции Делоне или диаграмм Вороного [170, 234], градиентного векторного потока [269, 267] или расстояния Фреше [194]. К сожалению, такие методы также не решают поставленных проблем точно и адекватно работают только на очищенных изображениях с высоким разрешением.

Предложенные алгоритмы, впервые рассмотренный в статьях [32, 150], отличается использованием как локальной информации о концевых участках ЛО, так и информации об их пересечении с другими ЛО и использовании функции максимального правдоподобия. В данном примере рассматривается класс ЛО, которые в семантическом плане не должны иметь пересечений друг с другом. При возможном наличии пересечений ЛО имеют обычно более простую структуру (например, разметка регулярной сетки) или изображены более широкими линиями для более чёткого отражения структуры (карты или фотографии с изображением дорог). Отдаётся предпочтение менее регулярным кривым относительно прямых отрезков: их обработка также может быть произведена более простыми методами, например преобразованием Хафа.

Концевыми точками (КТ) скелетизированной кривой считаются точки, имеющие только одного соседа. Если кривые не имеют пересечений, то они могут быть воссозданы только из линейных сегментов, имеющих ровно две КТ. В обратном случае должны существовать точки ветвления, имеющие более двух соседей. Если на нашем изображении присутствуют точки ветвления, то их следует удалить. Например, ЛО с тремя КТ и одной точкой ветвления будет преобразован в три ЛО, каждый из которых будет содержать две КТ.

Для каждого линейного сегмента зададим идентификатор. Его введение обусловлено необходимостью не рассматривать в начале возможность соединения КТ одного сегмента, что может возникать на малых сегментах.

Каждая КТ кроме своих непосредственных координат и принадлежности к линейному сегменту может обладать ещё значением угла касательной в этих точках. Определить касательную для скелитизированного сегмента по определению невозможно из-за жёсткой дискретизации кривых: соседи каждой из точек могут касаться только под углом  $\pi n/4$ , где  $n \in \mathbb{N}$ . Для решения этого вопроса можно либо напрямую векторизовать линейные сегменты, либо рассматривать их «как есть» в виде множества точек. Первый подход плох тем, что при параметрической задаче отрезков в виде функций  $x(t), y(t)$ , где  $t \in [0, 1]$  невозможно понять влияние производной этих функций для статического изображения, при этом производная в КТ может принять любое значение.

При рассмотрении линейных сегментов в виде множества точек угол касательной можно определить, используя координаты точек кривой вблизи КТ:

$$\alpha_i = -atan2\left(\frac{\sum_{(x,y) \in S_i} y}{\sum_{(x,y) \in S_i} 1} - y_i, \frac{\sum_{(x,y) \in S_i} x}{\sum_{(x,y) \in S_i} 1} - x_i\right), \quad (2.11)$$

где  $i$  – индекс КТ,  $S_i$  – множество пикселей концевой участка линейного сегмента около данной КТ,  $atan2(y, x)$  – функция четырёхквadrантного арктангенса, получающего на вход координаты вектора, исходящего из центра координат, и вычисляющего угол его наклона в интервале  $(-\pi, \pi]$ .

Учитывая, что линейный сегмент даже в небольшом окне может обладать некоторой кривизной (кривизну дискретной кривой можно определить аналогично определению касательной через несколько соседних точек), то данная формула является примерной, тем не менее, её применение лучше, чем способ, в котором угол наклона касательной к КТ определяется по одной точке, которая лежит на

некотором удалении от КТ [263], или которая является десятой точкой от КТ [248]. При наличии кривизны предложенный алгоритм будет более стабильным.

Также стоит отметить, что формула 2.11 подразумевает, что угол наклона вычисляется от КТ, поэтому вектора направления касательных в двух верно соединённых КТ должны быть противоположно направлены в рамках погрешности.

Учитывая, что мы можем знать только координаты КТ и примерный угол наклона касательной, то соединения КТ можно «безопасно» производить только прямыми отрезками: при допущении о наличии производных у кривой, заданной параметрически, выбор слишком большой нормы вектора производной может привести к пересечению кривой-соединителем других ЛО.

Обозначим множество концевых точек  $P_e$ , его мощность после любого соединения всегда является чётным числом, элементы этого множества обозначим через  $e_i$ , соединение двух точек –  $e_i e_j$ .

Можно обозначить следующие свойства верного соединения КТ:

- 1) локальность: длина соединения  $e_i e_j$  ограничена;
- 2) отсутствие пересечений: соединение  $e_i e_j$  не пересекает другие ЛО;
- 3) взаимное непересечение: любые пары соединений  $e_{i1} e_{j1}$  и  $e_{i2} e_{j2}$  не пересекаются.

Проблема этих свойств заключается в том, что для получения правильного набора соединений необходимо будет перебрать все возможные пары соединений КТ, что приведёт к экспоненциальной временной сложности: если существует  $2n$  КТ, то их можно соединить парами  $(2n - 1)!!$  различными способами. Хотя именно перебор может обеспечить максимально точное соединение, особенно, когда на небольшом участке происходит большое число разрывов, можно предположить, что существуют такие отношения свойств соседних КТ, соединения которых является верным с достаточной вероятностью. Рассмотрим некоторые из таких условий.

I Условие локальности – расстояние между двумя КТ ограничено некоторым пределом, зависящим от номера итерации:  $D(e_i, e_j) \leq R_{max}$ . Можно утверждать, что вероятность верности соединения пары точек имеет обратную зависимость от

расстояния между ними. Недостатком этого отношения является то, что может существовать большое число точек внутри окрестности  $R_{max}$ , что не позволит выделить единственную пару.

II Условие для углов касательных в КТ – угол между двумя концевыми участками должен быть как можно ближе к развёрнутому:  $||\alpha_i - \alpha_j| - \pi| < \varphi_{max}$ . Данное отношение лучше работает для более линейных участков с рассогласованием локальности.

III Условие для углов касательных в КТ с углом наклона возможного соединения:  $|\alpha_i - \gamma_i| < \varphi'_{max}$  &  $|\alpha_j - \gamma_j| < \varphi'_{max}$ . Данное отношение позволит корректно соединить линейные сегменты, для которых существует достаточный разрыв и рассогласование в углах касательных в КТ.

Условия I–III проверяются для всех возможных пар КТ, поэтому временная сложность их проверки составляет  $\Theta(n^2)$ . Вычислительную нагрузку можно сократить, если отсортировать КТ по координатам, разделить их по полосам, как в параграфе 2.4.1, чтобы отсеять точки, которые точно находятся далеко друг от друга.

Нельзя однозначно сказать, что данные соотношения нужно проверять именно в этом порядке. Например, если произошёл значительный разрыв двух близкорасположенных соседних линий, то существует два варианта соединения получившихся четырёх КТ. Верный вариант подразумевает согласованность по II и III отношению, но I не выполняется; для неверного варианта I отношение выполняется, а II и III нет.

Но при этом нельзя сказать, что нельзя учитывать расстояние между точками, так как, как уже было сказано ранее, угол касательных может быть вычислен достаточно неточно. Поэтому данную геометрическую группировку необходимо выполнять в несколько итераций, соединяя сначала более близкие пары КТ. При этом линейные сегменты, которые не могли быть соединены ранее, могут быть соединены позднее при увеличении максимальных параметров рассогласования.

IV Условие доминирования – расстояние между двумя данными КТ значительно меньше, чем при соединении их с другими КТ:



$$\exists i, j \in P_e, \forall k \in P_e \setminus \{i, j\}: \\ D(e_i, e_j) < D(e_i, e_k) - \Delta_{min}, (e_i, e_j) < D(e_j, e_k) - \Delta_{min}.$$

После проведения данной группировки количество точек уменьшается ещё, что позволяет начинать работу уже не с отдельными точками, а с парами КТ. Здесь можно провести фильтрацию, используя второе необходимое условие верного соединения: если линия, соединяющая пару КТ, пересекает ЛО, то это заведомо ложное соединение.

Под пересечением понимаем наложение матрицы отрезка соединения КТ с уже имеющейся матрицей линейных объектов. Обработка заключается в исключении окрестностей концевых точек и дилатацию скелетизированной матрицы ЛО. Данная обработка необходима чтобы избежать ложного наложения отрезка с ЛО, которые он соединяет, и чтобы предотвратить пропуск пересечения, возникающий из-за того, что отрезки, имеющие толщину в один пиксель из-за дискретности пиксельной матрицы, могут не иметь фактического пересечения.

К настоящему этапу осталось очень мало КТ, которые объединены в небольшие кластеры (если кластер состоит всего из двух КТ, то проверяем их на соединение непосредственно), для которых уже нельзя однозначно определить верное соединение, используя геометрический подход. Получить и проанализировать такие кластеры можно с помощью попарного сравнения.

V Условие максимального правдоподобия – мера для данной пары  $e_i e_j$  является минимальной относительно других пар в кластере  $C_e$ :

$$\exists i, j \in C_e, \forall k \in C_e \setminus \{i, j\}: \\ measure(e_i, e_j) < measure(e_i, e_k), measure(e_i, e_j) < measure(e_j, e_k)$$

В качестве меры используется следующая формула, которая объединяет в себе соотношения I и III:

$$measure(e_i, e_j) = (|\alpha_i - \gamma_i| + |\alpha_j - \gamma_j|) * D(e_i, e_j).$$

Временная сложность соотношений IV–V составляет  $\Theta(N_{pairs})$ , где  $N_{pairs}$  – количество рассматриваемых пар. Стоит отметить, что  $N_{pairs} \ll n^2$ , так как после реализации проверок I–III должно быть соединено подавляющее большинство КТ.

Пример работы алгоритма по использованию разработанных отношений представлен на рис. 2.7.

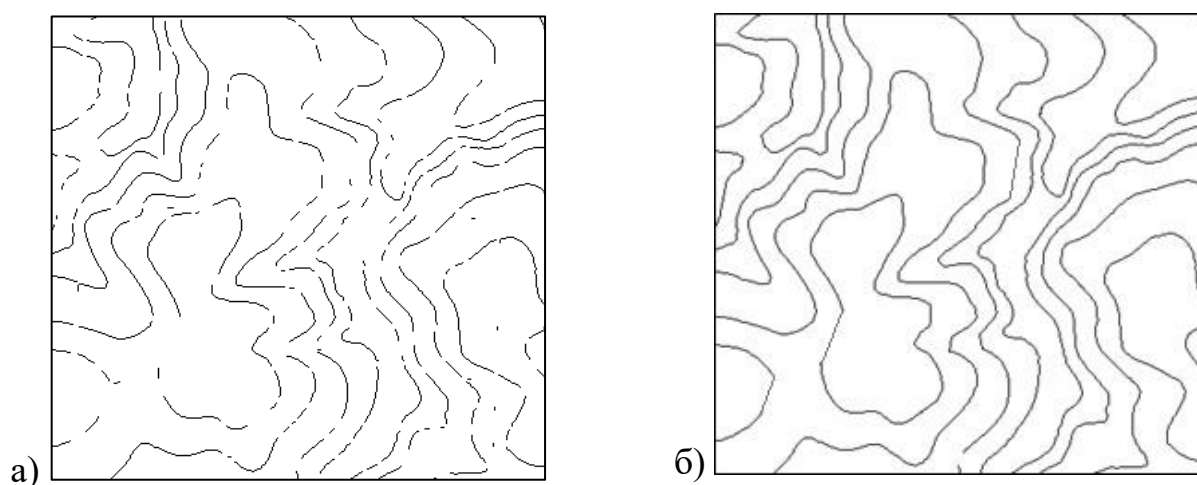


Рисунок 2.7 – Пример работы алгоритма группировки горизонталей:  
а) исходное изображение; б) результат группировки

## 2.5 Выводы

В данной главе был проведён анализ способов распознавания ССИ с учётом их характерных особенностей. Представлена общая методика анализа и обработки ССИ, определены проблемы, возникающие на каждом этапе, и разработан набор общих алгоритмов, которые можно применять к широкому классу ССИ.

Можно выделить следующие результаты, которые согласуются с выводами, представленными в первой главе:

- разработанная общая методика тяготеет к последовательному и комплексному анализу и обработке ССИ, а не концентрируется на отдельных этапах, что

позволяет разработчику алгоритмов глубже задумываться о природе изображаемых объектов;

– разработанные алгоритмы являются гибридными (сочетают в себе несколько подходов к обработке) и/или итеративными, что позволяет сочетать в себе несколько подходов и уходить от узкой специализации конкретных методов;

– для разработанных алгоритмов приведена их временная сложность, использованы подходы, которые могут её уменьшить;

– разработанные алгоритмы содержат как можно меньшее число констант, существующие пороговые значения по возможности зависят от номера итерации алгоритмов, а то, что выбирается в качестве констант (например, размеры скользящих окон), предлагается для оптимизирования в ходе экспериментов;

– разработаны алгоритмы, выполняющие начальную комплексную обработку: внутрифункциональную группировку; для данных алгоритмов были сделаны предположения, которые могут ограничить их использование для различных типов ССИ, тем не менее считаем данные ограничения адекватными.

Недостатком разработанных алгоритмов является то, что они не способны полностью автоматизировать обработку ССИ без ошибок. При этом количество сохранённого времени посредством введения автоматизации, оценивается как более значительное, чем время на обслуживание алгоритмов и корректировку некоторых значений и результатов алгоритмов.

Некоторые разработанные алгоритмы допускают необходимость дополнительных исследований для подбора операций внутри алгоритма (как непосредственный выбор, так и порядок): если для алгоритма локализации точечных объектов удалось построить генетический алгоритм, подбирающий морфологические операции автоматически, то для группировки точечных объектов и линейных объектов подбор выполняемых операций остаётся прерогативой разработчика программного обеспечения.

## Глава 3. Экспериментальная проверка разработанных алгоритмов

В данной главе рассмотрена обработка нескольких классов ССИ и проведён экспериментальный подбор параметров алгоритмов и интеллектуальных моделей, обеспечивающий оптимизацию по критериям качества и быстродействия.

### 3.1 Материалы и методика экспериментальной проверки разработанных алгоритмов

Экспериментальная проверка разработанных алгоритмов выполнялась с помощью пакета прикладных программ *MATLAB R2022a* под управлением 64-разрядной операционной системы *Windows 10* с 8 ГБ оперативной с использованием видеокарты *AMD Radeon 530* (2 ГБ). Ориентированность языка *MATLAB* на выполнение операций с матрицами и векторами, а также наличие значительного количества встроенных функций и средств визуализации позволяет выполнять операции над изображениями в более удобном и наглядном виде.

Из-за того, что ССИ являются узкоспециализированными объектами с большим количеством разнообразных правил отображения, а их обработкой занимается небольшое число исследователей, в научном сообществе исторически пока не сформированы датасеты для проверки работы алгоритмов над ними. Например, с использованием поисковой платформы *Semantic Scholar* было найдено более 200 научных статей по анализу и обработке рукотворных эскизов (скетчей) различных диаграмм: от блок-схем алгоритмов, до принципиальных электронных схем. Датасеты повторялись только в работах, где повторялись соавторы; публично выложенные датасеты содержались лишь в нескольких работах, относящихся к 2020–2023 годам, и над которыми ещё не проводились исследования другими авторами.

Данная особенность не позволяет сравнить результаты различных исследований между собой, в отличие, например, от задачи классификации изображений, для которой существует большое число широко использующихся датасетов, например,



и большим значением дисперсии, для сглаживания неровностей произведём медианную фильтрацию. Полученная матрица нормализовалась, чтобы вода занимала 90% карты. Полученный растр размера  $511 \times 511$  масштабировался с увеличением в 4 раза, используя линейную интерполяцию.

По полученной матрице высот строим изобаты для уровня в 20 метров, внутренние части которых (или прилегающие к берегу) закрашиваем голубым. Расположение отметок выбирается таким образом, чтобы между отметками было достаточное расстояние, но отметки могут располагаться поверх изобат. Кроме отметок глубин были добавлены того же цвета: отметки банок (мелей), скал, потонувших кораблей, координатная сетка и изобаты, подписи банок и грунтов.

После получения итоговых изображений используем гауссов фильтр с  $\sigma = 0,5$  для размытия, добавляем мультиспектральный шум с  $\sigma = 0,01$  и сохраняем их в формате *jpg*, что зачастую соответствует формату существующих изображений.

Всего было смоделировано 20 изображений размера  $2044 \times 2044$  пикселей, половина использовалась для обучения алгоритмов, оставшаяся половина – для тестирования. Все изображения размечались для проверки и настройки алгоритмов сегментации, локализации, распознавания и группировки ТО.

Для повышения обобщающей способности алгоритмов и интеллектуальных методов расстояние между центрами цифр варьируется в рамках нескольких пикселей. Высота цифр в датасете *MNIST* составляет около 20 пикселей, что обычно соответствует разрешению примерно 300 *dpi* при стандартном кегле шрифта 4-5. Для того, чтобы карта соответствовала 200 *dpi*, уменьшим изображения цифр до высоты, равной 13 пикселям.

В качестве моделей для сегментации и распознавания ТО предлагается исследовать логистическую регрессию (*LR*), линейный дискриминантный анализ (*LDA*), гауссов наивный байесовский классификатор (*GNB*), искусственные нейронные сети (*ANN*) с различной внутренней структурой, метод *k* ближайших соседей (*k-NN*), метод опорных векторов (*SVM*), метод классифицирующих

деревьев (*DT*), ансамблевый метод *AdaBoost* с использованием деревьев. Кроме изменения гиперпараметров моделей также исследуем изменение размера скользящего окна для сегментации, последовательность морфологических операций при локализации, различные исходные данные для распознавания и набор фильтров для группировки цифр.

2) Для проверки алгоритмов обработки площадных объектов использовались аэрофотоснимки из соревнования *DeepGlobe 2018 Land Cover Classification* [143]. Данные изображения можно отнести к классу Ia.

В датасете каждому из 803 изображений размера 2448×2448 была поставлена в соответствие маска, пиксели которой принадлежали к одному из семи классов: *urban, agriculture, rangeland, forest, water, barren, unknown* (рис. 3.2). Параметры гистограмм классов ПЛО определялись для тренировочного набора изображений только для тех пикселей, которые не находились на границе нескольких классов.

Сложность обработки таких изображений заключается не только в естественной вариации расположенных на них объектов, но и разметке данных достаточно плохого качества. При анализе данных достаточно часто были встречены изображения, на которых одинаковым по виду пространствам были даны различные метки либо, наоборот, различные по виду пространства отнесены к одному классу.

Так как объекты на изображении могут иметь сильную вариацию в отображении (например, чистота рек имеет большое влияние на их цвет) предлагается не использовать подход «один класс – одна гистограмма», а создать массив тренировочных данных, на основе которых сформировать интеллектуальную модель для дальнейшей классификации.

В качестве критерия качества используется среднеклассовое значение *IoU*:

$$mIoU = \frac{1}{k} * \sum_{j=1}^k IoU_j = \frac{1}{k} * \sum_{j=1}^k \frac{\sum_{i=1}^n TP_{ij}}{\sum_{i=1}^n TP_{ij} + \sum_{i=1}^n FP_{ij} + \sum_{i=1}^n FN_{ij}}, \quad (3.1)$$

где  $IoU_j$  – значение метрики  $IoU$  для  $j$ -го класса по точкам всех  $n$  изображений,  $TP_{ij}$  – количество верно классифицированных точек к классу  $j$  на изображении  $i$ ,  $FP_{ij}$  – количество неверно классифицированных точек к классу  $j$ ,  $FN_{ij}$  – количество неверно классифицированных точек к любому другому классу, кроме  $j$ .

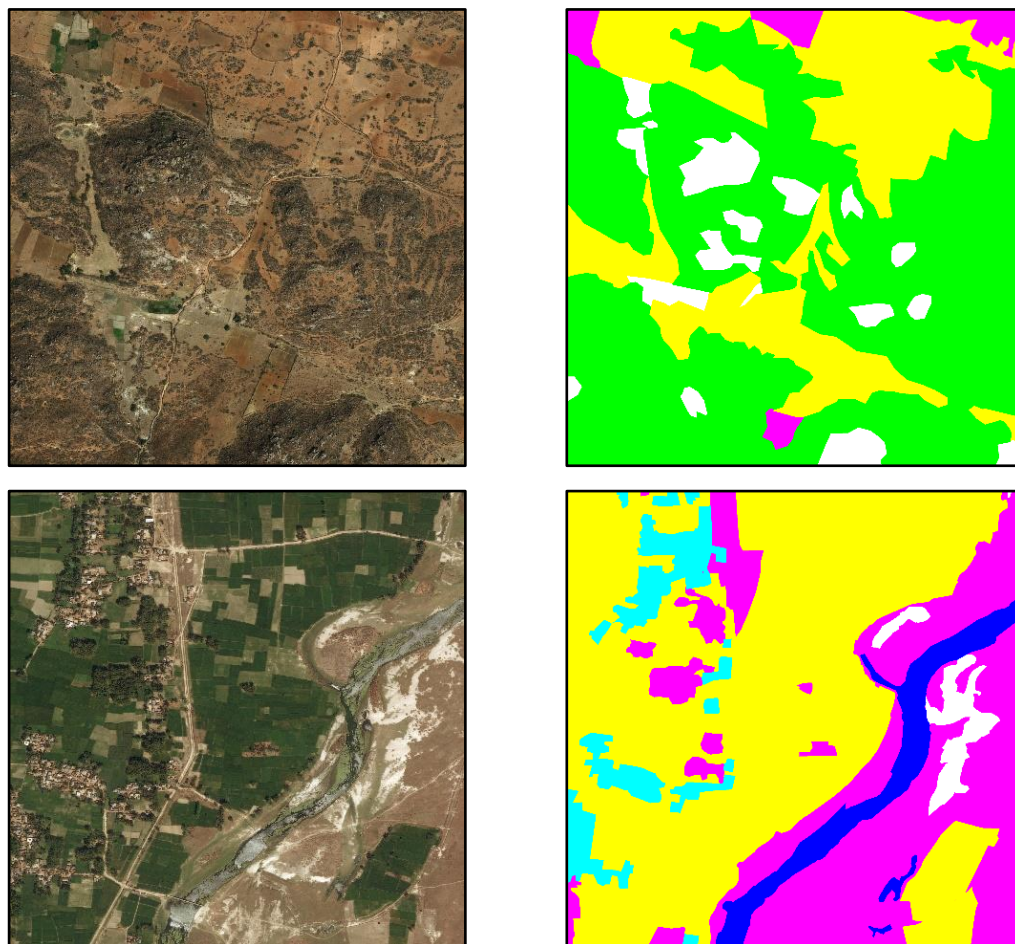


Рисунок 3.2 – Примеры изображений и соответствующих масок соревнования *DeepGlobe* 2018

В дополнение к экспериментальному исследованию всего алгоритма 2.5 в целом производилось тестирование только его второй части, отвечающей за постобработку, используя более мощное оборудование для реализации ресурсозатратной архитектуры DeepLabV3+ [129] в качестве интеллектуальной модели для первичной сегментации.



### 3.2 Экспериментальное исследование алгоритмов для обработки точечных объектов

Критериями качества алгоритмов распознавания ТО является время работы алгоритмов и точность распознавания. При этом, так как исправление ошибок работы алгоритма будет выполнять оператор, то данным ошибкам также можно поставить в соответствие некоторое время. Так как в качестве ТО рассматриваются отметки глубин на смоделированных участках навигационных карт, то ошибками работы алгоритмов с указанием времени их исправления могут быть:

- неверное распознавание, может быть исправлена за время  $T_{\text{испр}} \sim 1$  сек, не зависящее от количества ошибок в числе;
- ложное срабатывание, когда за числовую отметку указывается некоторый другой найденный образ, может быть исправлена удалением за время  $T_{\text{уд}} \sim 1$  сек;
- пропуск отметки, время исправления всех таких отметок в целом зависит не только от их количества, но и от размера карты.

Видно, что ошибки последнего типа являются самыми трудозатратными для оператора ЭВМ, поэтому их количество должно быть минимизировано на уровне сегментации и локализации.

Общая схема обучения и тестирования алгоритмов обработки ТО на морских навигационных картах показана на рис. 3.3. Так как карты моделируются, то нам известна вся разметочная информация: бинарная карта с чёрными пикселями и таблица с данными об отметках: значение числа, координаты его центра  $(x, y)$  и размеры  $H \times W$ , значение отдельных цифр и координаты их центров. Координаты центра числа и его размеры необходимы для этапа локализации, координаты цифр и их значения нужны для этапа распознавания, значение числа нужно для этапа группировки.

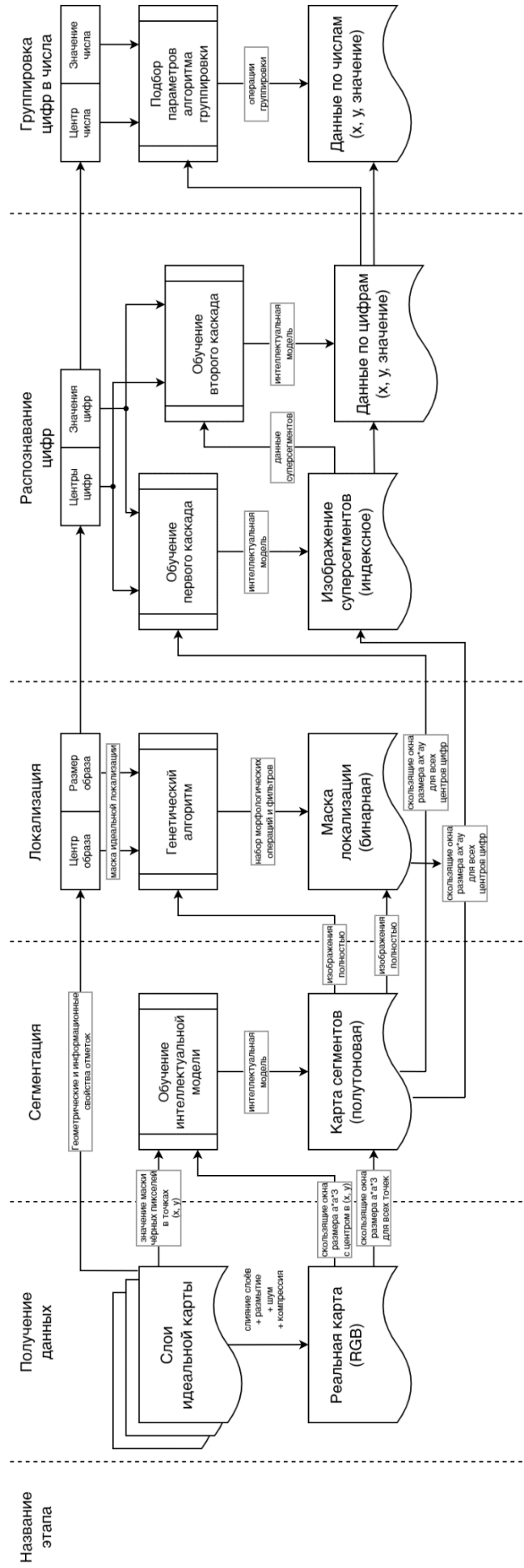


Рисунок 3.3 – Схема обучения и тестирования обработки точечных объектов

Первым этапом исследования является проверка моделей, выполняющих сегментацию. Так как на данном этапе критичен пропуск необходимых пикселей, который может привести к некорректной работе алгоритмов следующих этапов, то в качестве критерия качества будем использовать  $F_\beta$ -меру со значением  $\beta = 2$ .

Для обучения моделей для сегментации было выбрано по 1000 точек на каждом тестовом изображении, относящихся к искомому классу и не относящегося. Данные точки выбирались центрами скользящих окон, из которых производилась выборка признаков.

Результаты по моделям представлены в таблицах 3.1-3.3. Не приведены методы ( $kNN$ , ансамблевый метод и ИНС при  $a > 3$ ), для которых время работы составило больше 10 с/Мп.

Таблица 3.1 – Результаты при использовании формальных методов,  $a = 1$

Формула	$F_\beta, \%$	$T, \text{с/Мп}$
$0 \leq R, G, B \leq threshold = 160$	$96,5 \pm 0,2$	0,16
$0 \leq \text{mean}(R, G, B) \leq threshold = 150$	$96,3 \pm 0,2$	0,17

Таблица 3.2 – Результаты при использовании нейросетевых моделей

Размер скользящего окна, пикс.	Структура сети	$F_\beta, \%$	$T, \text{с/Мп}$
1	3/3/1	$96,9 \pm 0,2$	0,60
	3/6/1	$97,1 \pm 0,2$	0,72
3	27/7/1	$97,0 \pm 0,2$	2,61
	27/13/1	$97,1 \pm 0,2$	2,85
	27/27/1	$97,3 \pm 0,2$	3,17
	27/54/1	$97,2 \pm 0,2$	4,56

Таблица 3.3 – Результаты при использовании прочих моделей,  $a = 3$

Метод	Параметры	$F_\beta, \%$	$T, \text{с/Мп}$
$DT$	7 листов	$95,7 \pm 0,4$	0,99
	35 листов	$95,1 \pm 0,4$	1,08
$SVM$	линейный	$94,3 \pm 0,5$	1,86
$LDA$	–	$96,4 \pm 0,3$	2,24

По результатам видно, что усложнение моделей приводит к увеличению работы алгоритма сегментации, но не всегда приводит к значимому улучшению критерия качества. Это, в том числе, можно объяснить тем, что в данном исследовании важным считается только один класс. При многоклассовой сегментации показатели качества в среднем будут ниже, поэтому там имеет смысл усложнять обрабатываемые модели. В качестве базовой модели для продолжения исследования была выбрана ИНС со структурой 27/13/1.

Следующим этапом является локализация. Для реализации генетического алгоритма были созданы следующие функции, используемые для получения масочных изображений по данным расположения чисел; непосредственного выполнения генетического алгоритма 2.4; вычисления функции приспособленности; реализации генетических операций мутации и скрещивания; вычисления центромеры хромосомы; удаления пустых генов.

Хромосома хранилась в виде массива генов, каждый из которых обладал следующей структурой: имя операции и параметры операции. В качестве параметров операций могли быть названия СЭМО или границы для фильтрации.

Параметры генетического алгоритма:

- размер популяции – 100 хромосом;
- максимальное количество итераций – 100;
- доля элитных хромосом, переходящих в следующее поколение – 30%;
- вероятность скрещивания – 10%;
- вероятность мутации – 90%;
- вероятность вставки нового гена –  $\frac{5}{N_{\text{генов}}}$  (при  $N_{\text{генов}} > 5$ , иначе 1);
- вероятность удаления гена –  $0,05(1 - \frac{1}{N_{\text{генов}}})$  (при  $N_{\text{генов}} > 5$ , иначе 0);
- вероятность модификации гена –  $0,95 - \frac{4,95}{N_{\text{генов}}}$  (при  $N_{\text{генов}} > 5$ , иначе 0);
- вероятность взять ген для мутации из левой части – 50%;
- вероятность взять ген для мутации из правой части – 50%.

ГА для получения наилучшей хромосомы запускался на одном изображении из тренировочных данных, так как на них присутствует достаточное количество отметок глубин: 926, всего цифр 1764. Масочное изображение для идеального преобразования было получено из маски критичных точек с помощью операции дилатации со структурным элементом (СЭ) квадрат размером  $7 \times 7$ ,  $|PictureIdeal| \approx 85$  тыс. точек.

Результаты ГА при получении оптимальной хромосомы на этом изображении продемонстрированы в таблице 3.4 и рис. 3.4. В качестве критерия качества *Recall* использовалось отношение найденных критичных точек к их количеству (средний множитель формулы 2.5).

Таблица 3.4 – Результаты проведённых итераций генетического алгоритма

№	<i>N</i> генов	<i>Recall</i> , %	<i>FF</i> , %	№	<i>N</i> генов	<i>Recall</i> , %	<i>FF</i> , %
1	2	98,9	22,0	50	13	96,8	60,4
5	3	97,6	25,5	60	16	96,6	62,9
10	6	95,2	35,4	70	16	96,4	63,7
20	8	96,3	41,4	80	18	96,3	64,4
30	9	96,2	46,6	90	18	96,8	64,7
40	12	96,8	58,5	100	17	96,8	65,7

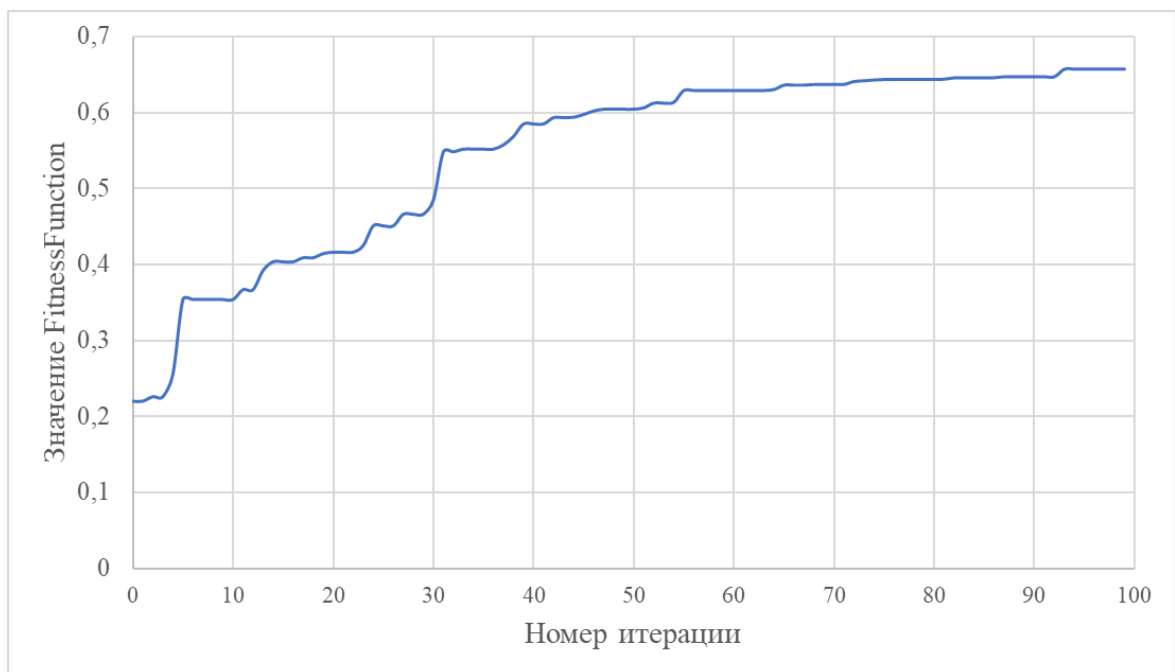


Рисунок 3.4 – График функции приспособленности от номера итерации ГА

В итоге сформированная хромосома состоит из следующих операций:

- 1) закрытие, СЭ – горизонтальная линия длиной 3;
- 2) закрытие, СЭ – вертикальная линия длиной 3;
- 3) открытие, СЭ – квадрат размера  $2 \times 2$ ;
- 4) закрытие, СЭ – диск радиуса 2;
- 5) бинаризация с порогом 0,39;
- 6) закрытие, СЭ – вертикальная линия длиной 6;
- 7) фильтрация по большей полуоси [5,4; 23,2];
- 8) фильтрация по размеру сегмента в интервале [4, 268];
- 9) открытие, СЭ – вертикальная линия длиной 3;
- 10) дилатация, СЭ – горизонтальная линия длиной 5;
- 11) закрытие, СЭ – квадрат размера  $2 \times 2$ ;
- 12) фильтрация по меньшей полуоси в интервале [5,3; 21,1];
- 13) открытие, СЭ – вертикальная линия длиной 4;
- 14) эрозия, СЭ – квадрат размера  $3 \times 3$ ;
- 15) фильтрация по размеру сегмента в интервале [3, 466];
- 16) дилатация, СЭ – горизонтальная линия длиной 2;
- 17) закрытие, СЭ – вертикальная линия длиной 2.

Данная хромосома была запущена на всех 20 изображениях. Скорость обработки 0,16 с/Мп, значение функции приспособленности равно  $64,4\% \pm 1,0\%$ , полнота равна  $96,1\% \pm 0,7\%$ . Данные результаты означают, что из всех отметок глубин только 3,9% будут потеряны, что является достаточно хорошим вариантом, так как ССИ обычно обладают значительной степенью избыточности, поэтому значения в точках этих отметок может быть восстановлено с достаточной точностью. Значение функции приспособленности в 64,4% (учитывая вхождение в этот критерий полноты) означает, что ложноположительных ошибок будет не больше 50% от количества пикселей, принадлежащих *PictureIdeal*.

Разберём в качестве примера обработку первого изображения. На нём после операции сегментации содержится 160 тыс. точек (из ~4 млн. всего), отнесённых к

классу «черные пиксели». Так как напрямую использовать их нельзя из-за того, что центр цифры может находиться внутри сегмента и не относиться к нему, то перед распознаванием необходимо применить МО закрытия с СЭ квадрат размера  $5 \times 5$ , при этом количество пикселей увеличивается до 320 тыс. точек (при этом данная операция будет оптимальной с точки зрения увеличения количества рассматриваемых пикселей, так как она действительно используется для закрытия «отверстий» в изображениях [217]). После же проведения операций, полученных с помощью ГА, количество точек сокращается до 94 тыс. точек (на 10% больше, чем *PictureIdeal*), что приводит к уменьшению времени на этапе распознавания в 3,4 раза; в среднем по всем изображениям в датасете это число равно  $3,7 \pm 0,5$ . Пример запуска операций, полученных с помощью ГА, показан на рис. 3.5 (СМУК): координата *M* отвечает за изображение, полученное с этапа сегментации, *C* – выходное изображение после локализации.

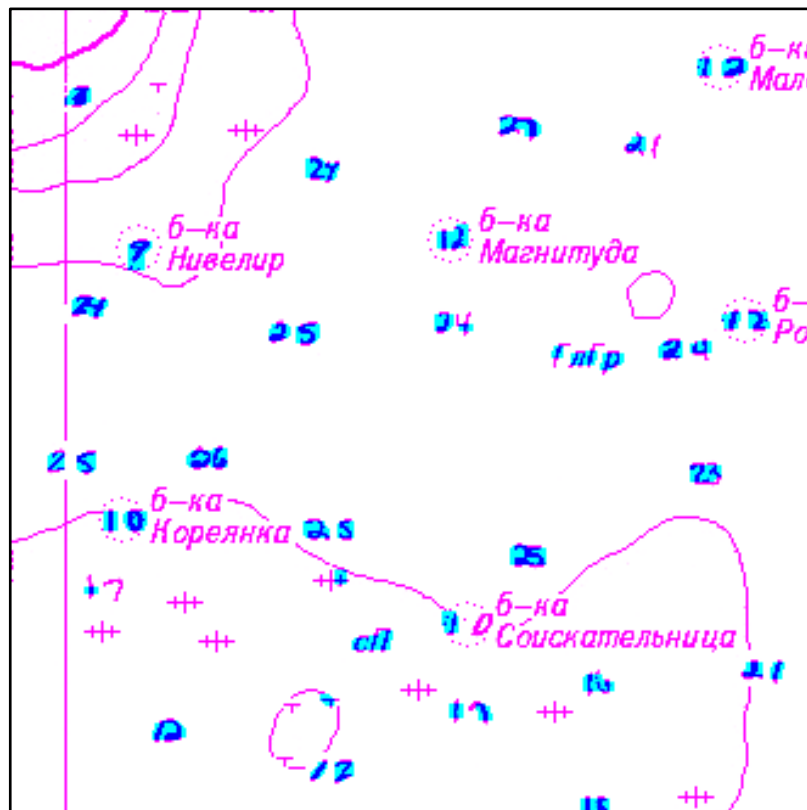


Рисунок 3.5 – Результат работы списка операций, полученных в результате ГА

Следующим этапом является распознавание цифр. На 10 тренировочных изображениях были обучены следующие модели: ИНС прямого распространения с одним скрытым слоем, дерево решений, *kNN*-классификатор, машина опорных векторов. Для *kNN* и *SVM* количество тренировочных данных было сокращено для соответствия времени работы относительно других моделей. Разбивка на тренировочный, валидационный и тестовый датасеты происходила в пропорции 2:1:1.

В качестве обучающих данных выступали изображения цифр размера  $13 \times 11$ . В первом случае в качестве центров образов использовались только центры цифр, во втором случае к ним добавлялись 4 соседних пикселя (всего 5), в третьем случае – ещё 4 соседних по диагонали пикселя (всего 9). Таким способом исследовалось влияние аугментации на точность указанных интеллектуальных моделей.

ИНС обучались с помощью метода сопряжённых градиентов, выходной слой имел функцию *softmax*. Деревья решений ограничивались в 100, 1000 и 10000 ветвлений. *kNN*-классификатор запускался для *k*, равных 1, 3 и 5 соседей. Использовалось линейное ядро для *SVM*.

Для разделения сегмента с числом на несколько сегментов также была обучена нейросеть с выходным бинарным классом «цифра/не цифра»: позитивными примерами класса выбирались пиксели из квадрата  $3 \times 3$  относительно центра, негативными – вершины и середины сторон квадрата  $7 \times 7$  относительно центра.

В табл. 3.5 представлены результаты тестирования первого каскада распознавания. Полнота рассчитывалась относительно центрального квадрата  $5 \times 5$  цифры: если в данной области присутствовал хоть один пиксель, принадлежащий к правильному классу, то это увеличивало *TP*. Точность рассчитывалась относительно всех пикселей, указанных к данному классу.

По анализу табл. 3.5 видно, что ИНС и дерево решений сохраняют достаточно высокий уровень полноты, но точность гораздо выше у нейросетей. Отметим, что такой уровень точности (~30%) будет улучшен после второго каскада и дальнейшей фильтрации на уровне группировки. Полнота при увеличении аугментации значительно росла только у ИНС, точность увеличивалась у всех моделей.



Таблица 3.5 – Результаты тестирования первого каскада распознавания

Метод расознавания	<i>Recall, %</i>			<i>Precision, %</i>			<i>T, с/Мп</i>
	Обучающая выборка			Обучающая выборка			
	1	5	9	1	5	9	
<i>ANN, 143/50/10</i>	77,0	91,3	93,0	18,2	26,3	28,8	0,55±0,02
<i>ANN, 143/100/10</i>	78,7	92,0	93,4	18,3	27,0	29,9	0,64±0,03
<i>ANN, 143/200/10</i>	79,0	92,7	94,0	18,8	27,1	32,8	0,74±0,03
<i>ANN, 143/400/10</i>	78,4	93,4	94,2	18,6	27,2	33,4	0,98±0,05
<i>DT, 100</i>	87,7	90,9	91,5	8,9	9,5	10,0	0,15±0,02
<i>DT, 1000</i>	90,7	93,9	93,6	8,3	10,3	10,6	
<i>DT, 10000</i>	90,7	95,3	95,5	8,3	10,0	11,1	
<i>kNN, 1</i>	90,6	92,1	92,2	9,2	9,7	11,0	1,1±0,1
<i>kNN, 3</i>	86,5	86,3	87,6	9,9	13,2	12,9	
<i>kNN, 5</i>	87,2	87,5	85,5	10,5	12,8	14,2	
<i>SVM</i>	81,3	80,0	75,6	10,7	12,0	12,6	1,8±0,2

На рис. 3.6 показан пример работы первого каскада алгоритма для выбранной ИНС, имеющей архитектуру 143/400/10, обученной с использованием полной аугментации. Рис. 3.7 б) изображён в индексном пространстве: фон изображён белым, найденный образ цифры «1» – оранжевым, «2» – жёлтым, «5» – циановым, «6» – лазурный и т. д. Видно, что обозначение грунта «Гк» также определилось как цифра «1», что можно объяснить схожестью начертания в скользящем окне 13×11. Так как цифры в числе «35» оказались расположены слишком близко друг к другу, то их сегменты не были разделены.

Перед работой второго каскада распознавания были проведены следующие операции над получившимися суперсегментами:

- удаление из них сегментов, состоящих из одного пикселя (в данной операции использовалась окрестность Мура, для остальных – фон Неймана);
- удаление суперсегментов, состоящих меньше, чем из 5 пикселей;

– разбивка сегментов, состоящих больше, чем из 28 пикселей, с помощью разреза посередине.

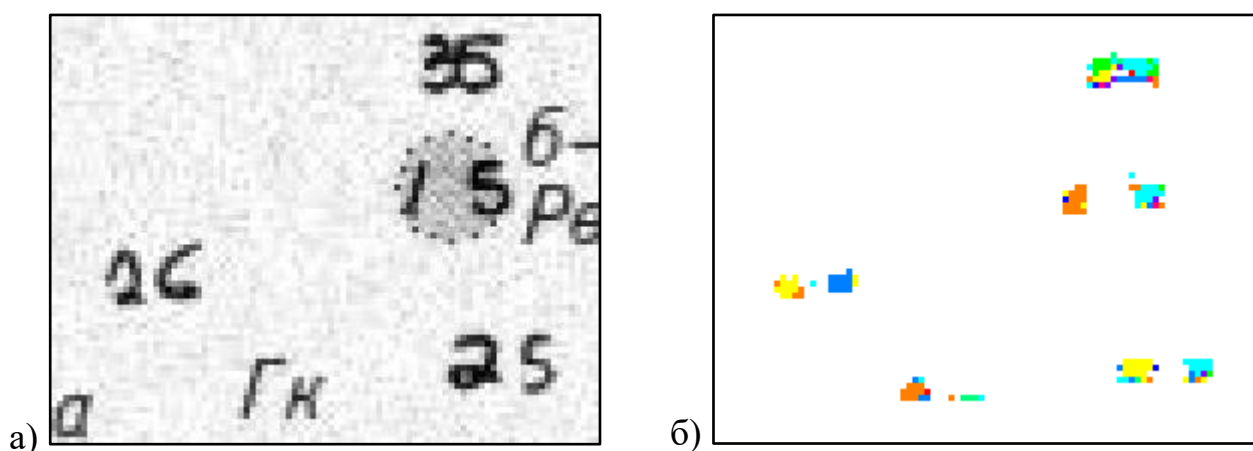


Рисунок 3.6 – Пример работы первого каскада алгоритма:  
а) оригинальное изображение; б) полученные суперсегменты

Обучение интеллектуальных моделей происходило аналогично предыдущему каскаду. В качестве обучающих данных были произведены операции первого каскада распознавания над тренировочными изображениями. Для каждого суперсегмента были найдены доли вхождения в него 10 искомым классов.

Результаты проведенного тестирования показаны в табл. 3.6. Кроме интеллектуальных моделей использовалась также максимизация по доле. Так как на данном этапе обрабатываются более высокоуровневые объекты с небольшой размерностью входных параметров, то время работы всех моделей оказалось примерно одинаковым и равным 0,2 с/Мп.

Отличие *Common Precision* от *Identity Precision* заключается в том, что первая рассчитывается в том числе для случая, когда распознаваемый образ отсутствует, например, вертикальные линии часто распознаются в качестве «1». Вторая же рассчитывается только по множеству образов, которые существуют. Формулы для полноты и двух данных точностей могут быть получены с использованием матрицы неточностей (*confusion matrix*):

$$Recall = \frac{\sum_{i=1}^{10} cm(i, i)}{\sum_{i=1}^{10} \sum_{j=0}^{10} cm(i, j)}, \quad (3.2)$$

$$Common\ Precision = \frac{\sum_{i=1}^{10} cm(i, i)}{\sum_{i=0}^{10} \sum_{j=1}^{10} cm(i, j)}, \quad (3.3)$$

$$Identity\ Precision = \frac{\sum_{i=1}^{10} cm(i, i)}{\sum_{i=1}^{10} \sum_{j=1}^{10} cm(i, j)}, \quad (3.4)$$

где  $cm(i, j)$  – количество объектов, которые были отнесены к классу  $j$ , и на самом деле относятся к классу  $i$ .  $cm(0, j)$  означает, что количество объектов, отнесённых к классу  $j$ , которых не существует,  $cm(i, 0)$  означает количество объектов класса  $i$ , которые пропустили.

Таблица 3.6 – Результаты тестирования второго каскада распознавания

Метод распознавания	<i>Recall</i> , %	<i>Common Precision</i> , %	<i>Identity Precision</i> , %
<i>ANN</i> , 10/10/10	84,4	82,2	92,8
<i>ANN</i> , 10/20/10	84,8	82,6	93,3
<i>ANN</i> , 10/100/10	85,0	82,9	93,6
<i>DT</i> , 10	59,2	57,7	65,1
<i>DT</i> , 20	66,2	64,5	72,8
<i>DT</i> , 100	83,6	81,4	91,9
<i>DT</i> , 1000	83,3	81,2	91,6
<i>kNN</i> , 1	83,2	81,1	91,5
<i>kNN</i> , 3	84,3	82,1	92,7
<i>SVM</i>	82,0	79,9	90,2
<i>max</i>	79,8	77,8	87,8

Последним этапом является группировка цифр в числа. После первичной кластеризации, реализованной по алгоритму 2.8, была также проведена фильтрация образовавшихся кластеров, которая частично отражает также и комплексный анализ данных:

– удаление отметок, имеющих значение меньше 10 (состоят из одной цифры), и не располагающихся на голубом фоне, означающем мелководье;

- удаление отметок со значениями больших 99 (состоят из 3 и более цифр);
- удаление отметок со значениями, отличающимися больше чем на 15 от соседних отметок, найденных с помощью триангуляции Делоне.

В табл. 3.7 показаны итоговые результаты применения фильтраций. Критерии качества показаны для чисел в целом, а не для отдельных цифр. Скорость работы, включая все этапы фильтрации составила 0,14 с/Мп.

Таблица 3.7 – Результаты применения фильтраций

Этап	<i>Recall</i> , %	<i>Common Precision</i> , %	<i>Identity Precision</i> , %
Первичная кластеризация	77,4	70,6	78,4
Удаление отметок из 1 цифры на неголубом фоне	77,4	80,9	86,6
Удаление отметок из больше, чем 2 цифр	77,4	82,9	87,7
Удаление отметок по сравнению со соседними	77,4	85,0	89,1

В результате получается, что из всех отметок с помощью полученного алгоритма можно верно определить 89%, среднее время работы на обработку одной карты составило 18 секунд. Данный результат сравним с качеством распознавания человека для отметок такого размера: в [214] точность равна 82% для отметок высотой до 25 пикселей. Стоит также отметить, что погрешность распознавания и/или пропуска отметок растёт с количеством цифр в них. Стоит учитывать размеры отметок, используемых в данном исследовании: в настоящее время почти все научные работы касаются обработки цифр из базы *MNIST* не масштабируя их и не учитывая пересечения их друг с другом.

На рис. 3.7 представлен иллюстративный материал поэтапной обработки одной из карт, выполненной с помощью программной реализации указанных алгоритмов.

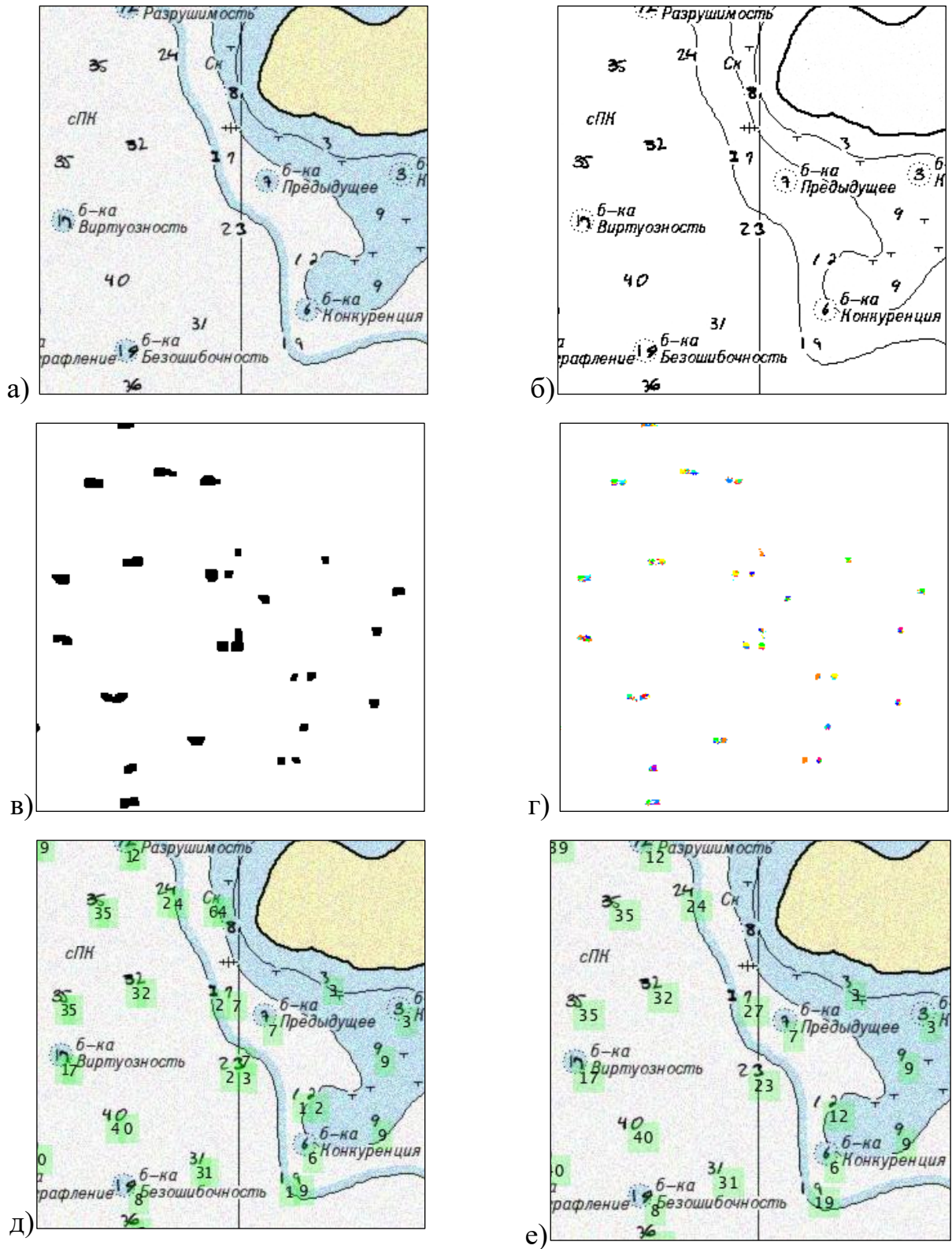


Рис. 3.7 – Поэтапные результаты работы разработанных алгоритмов для случая анализа и обработки смоделированных морских навигационных карт:  
а) участок смоделированного изображения; б) результат сегментации; в) результат локализации; г) результат первого каскада распознавания; д) результат второго каскада распознавания; е) результат группировки

### 3.3 Экспериментальное исследование алгоритмов для обработки площадных объектов

Экспериментальное исследование было основано на получении тренировочных данных, обучении интеллектуальной модели, настройке параметров постобработки и тестировании с определением критерия качества *mIoU*.

Функция *GetData* использовалась для получения тренировочных и валидационных данных. В качестве тренировочных изображений использовались 723 (90%) случайно выбранных из исходного датасета пар, состоящих из снимка и размеченной маски. Для того, чтобы избежать обучения на окнах, пиксели которых могли быть отнесены к разным классам, для каждого бинарного изображения класса маски была произведена эрозия. Среди каждого присутствующего на маске класса случайно выбиралось по 100 точек, которые являлись центрами для скользящих окон. Окрестности были получены с помощью функции *GetNeighborhood*.

Гистограммы строились для каждого из цветовых пространств, разбивая их на 32 интервала, всего 96 признаков на объект с помощью функции *img2hist*. Всего было сформировано 232650 тренировочных и 25850 валидационных объектов. Благодаря особенностям выборки удалось уменьшить сильную несбалансированность исходных данных (таблица 3.8).

Таблица 3.8 – Распределение классов в оригинальном датасете и в тренировочных данных, в %

Датасет	<i>urban</i>	<i>agriculture</i>	<i>rangeland</i>	<i>forest</i>	<i>water</i>	<i>barren</i>	<i>unknown</i>
Общий	9,35	56,76	10,21	13,75	3,74	6,14	0,04
Выборка	21,62	24,41	17,60	6,27	15,59	14,51	-

Пример гистограмм для класса *forest* для различных размеров скользящих окон показан на рис. 3.8. Видно, что гистограммы для различных размеров скользящих окон достаточно похожи, но для маленького размера они являются менее гладкими. Так как признаки для  $a = 11$  и  $a = 25$  примерно похожи, а время,

затраченное на обработку одной точки, зависит от размера окна квадратично, то выбираем  $a = 11$ .

В качестве интеллектуальной модели была выбрана ИНС прямого распространения с одним скрытым слоем (100 нейронов, функция активации сигмоида), выходной слой содержал 6 выходов. В качестве оценки ИНС использовалась перекрёстная энтропия, обучение происходило с помощью метода сопряжённых градиентов. Обучение производилось в течение 2,5 часов, итоговое значение перекрёстной энтропии равно 0,20, что является приемлемым для ССИ. На валидационной выборке (80 изображений из 723) был подсчитан  $mIoU$ , который оказался равен 30,8%.

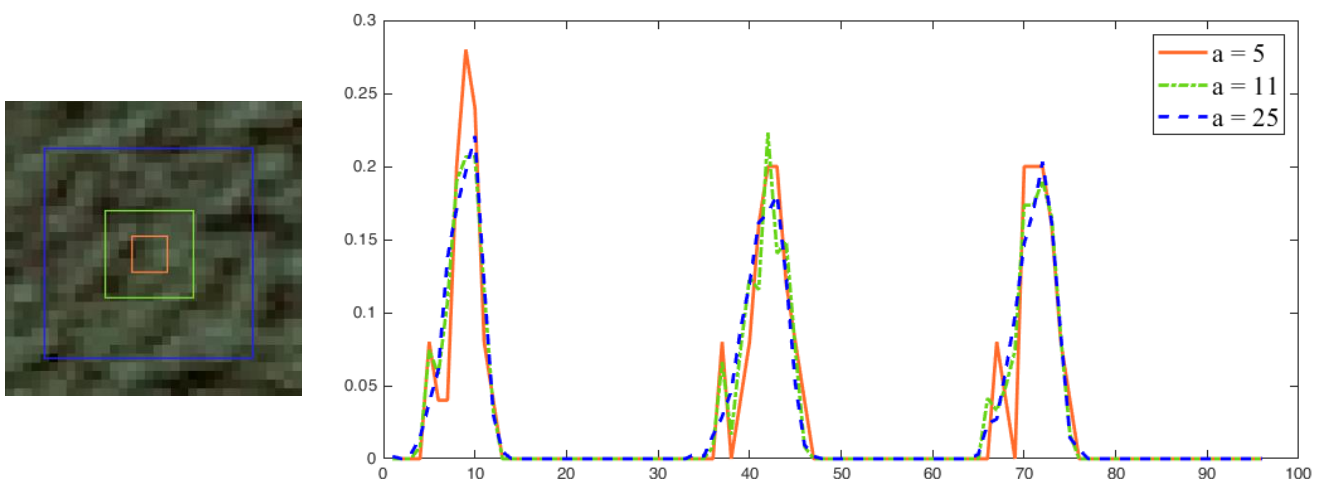


Рисунок 3.8 – Окрестность случайной точки класса *forest* и гистограммы скользящих окон с размерами 5, 11, 25

Так как некоторые классы в датасете встречаются значительно реже и критерий оценки обучения ИНС отличается от критерия качества  $mIoU$ , то тренировочная выборка использовалась в т. ч. для подбора параметров постобработки. Постобработка заключалась в выполнении двух операций:

- коррекция выходных значений ИНС с помощью коэффициентов;
- фильтрация с помощью удаления небольших сегментов и сегментация по алгоритму из подпараграфа 2.3.2.

Коррекция выходных значений ИНС происходила с помощью линейного преобразования:

$$out_1(x, y, class) = out(x, y, class) * c(class), \quad (3.5)$$

где  $out(x, y, class)$  – выходное значение ИНС для выхода, соответствующего классу  $class$ ,  $c(class)$  – значение поправочного коэффициента.

Поправочные коэффициенты подбирались двумя способами: метод наименьших квадратов (МНК) и подбором. Так как после коррекции выхода будет производиться операция выбора класса, имеющего максимальную меру, то можно один из поправочных коэффициентов приравнять к единице: в нашем случае таким был выбран коэффициент для класса *urban*. Для МНК использовалась формула для парной линейной регрессии без свободного члена:

$$c(class) = \frac{|P| * \sum_P out * target - \sum_P out * \sum_P target}{|P| * \sum_P out^2 - (\sum_P out)^2}, \quad (3.6)$$

где  $|P|$  – количество пикселей на изображении,  $target$  – мера принадлежности классу, взятой с маски (для изображённого класса равен 1 на текущей точке, для всех остальных – 0). После расчёта значения коэффициентов были разделены на  $c(urban)$ .

По методу МНК получились следующие коэффициенты:

- *urban*: 1;
- *agriculture*: 2,190;
- *rangeland*: 0,627;
- *forest*: 2,168;
- *water*: 0,748;
- *barren*: 0,963.

При методе подбора коэффициентов в их качестве использовались случайные значения по формуле  $c = 2^{3*rand - 1.5}$ , где  $rand$  – равномерная случайная величина из интервала (0, 1).

Результаты подбора коэффициентов изображены на рис. 3.9 с помощью метода главных компонент (PCA), в качестве осей используются три компонента с



максимальным значением выборочной дисперсии, цветом обозначается значение  $mIoU$ . Треугольником отмечена точка с оригинальным выходом ИНС (коэффициенты равны единицам), квадратом – набор коэффициентов, полученных по методу МНК, окружностью – набор коэффициентов с лучшим значением  $mIoU$ , равным 37,1%. По рисунку можно отметить, что в данной задаче действительно может существовать оптимальное значение поправочных коэффициентов, так как видна тенденция к увеличению качества при определённом направлении в системе координат, заданной главными компонентами.

С помощью подбора были получены следующие коэффициенты:

- *urban*: 1;
- *agriculture*: 1,851;
- *rangeland*: 1,677;
- *forest*: 2,771;
- *water*: 0,506;
- *barren*: 0,744.

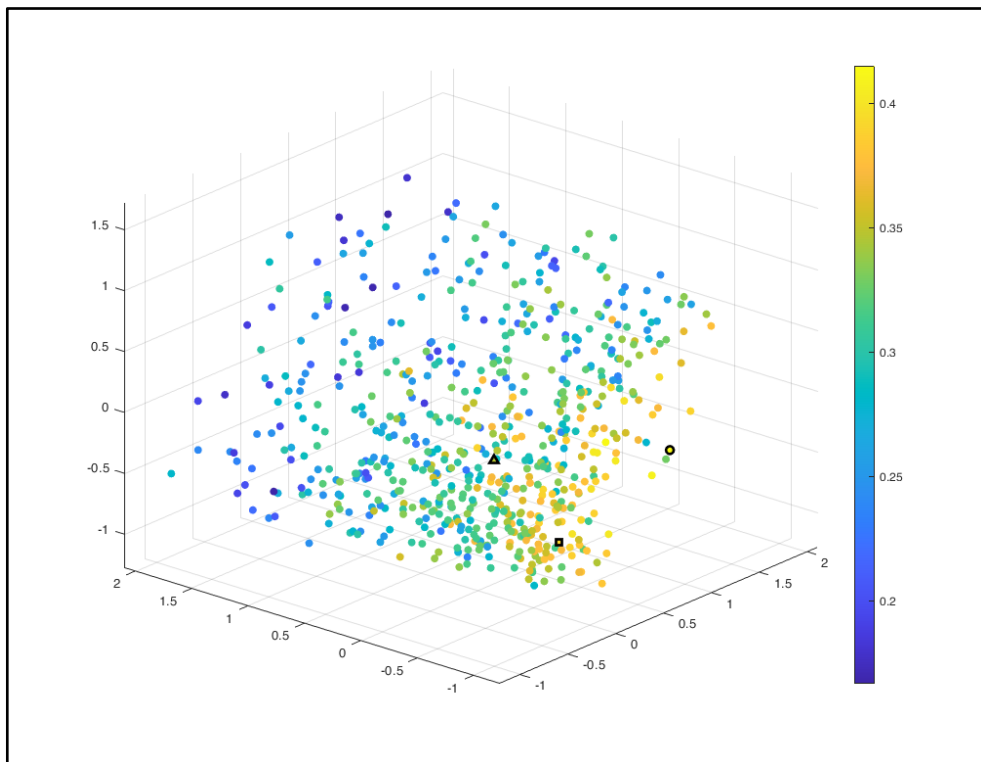


Рисунок 3.9 – Значения  $mIoU$  для разных поправочных коэффициентов ИНС в системе координат, полученной с помощью *PCA*

Дальнейшая постобработка была связана с удалением небольших фрагментов и выполнением операции моды. Параметры данной фильтрации также

определялись с помощью подбора максимального размера сегмента для удаления (значение выбиралось из ряда {500, 1000, 1500, 2000}), и четырёх значений окон для взятия моды (выбирались случайные значения из рядов  $\{[1.2^n] | n \in \{14, 15, \dots, 35\}\}$ ,  $\{[1.5^n] | n \in \{6, 7, \dots, 15\}\}$ ,  $\{[1.6^n] | n \in \{6, 7, \dots, 13\}\}$ ).

Результаты подбора параметров изображены на рис. 3.10 с помощью метода главных компонент, аналогично рис. 3.9. Набор параметров, который дал лучший результат, изображён в окружности; максимальный размер сегмента равен 1000, набор значений окон: {86, 68, 26, 25}, при этом значение *mIoU* равно 55,1%.

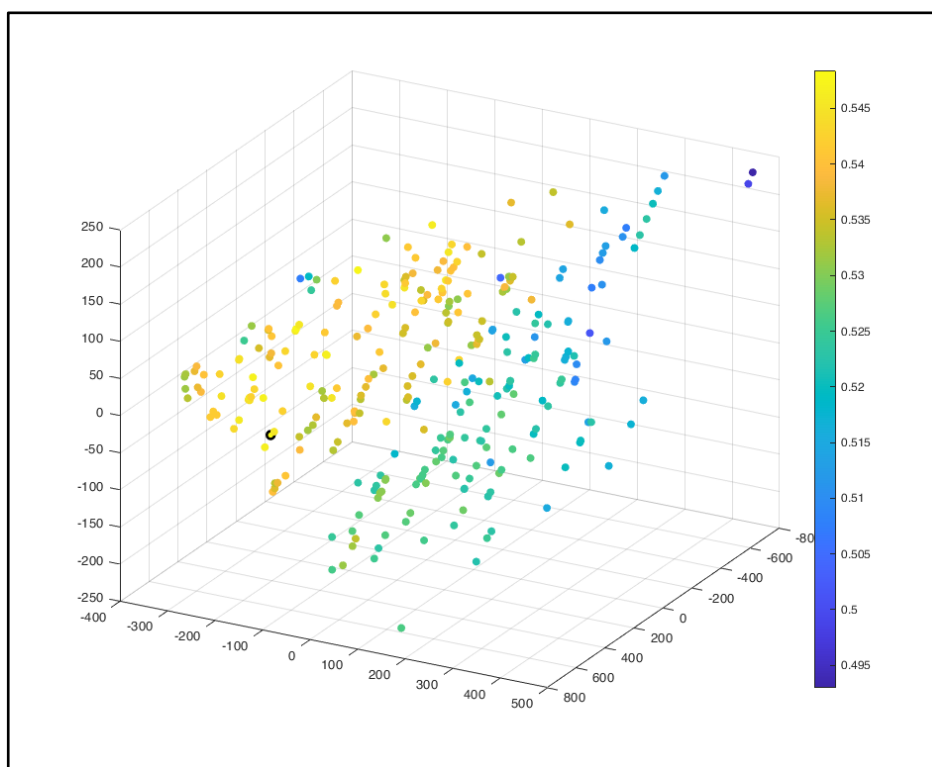


Рисунок 3.10 – Значения *mIoU* для разных значений параметров фильтрации в системе координат, полученной с помощью *PCA*

Иллюстрация работы алгоритма показана на рис. 3.11. Использование размеров окон порядка несколько десятков приводит к исчезновению деталей, но обеспечивает хорошую обобщающую способность. Тем не менее стоит отметить, что указанный подход к постобработке может быть улучшен, если увеличить количество обрабатываемых операций и/или количество подбираемых параметров.

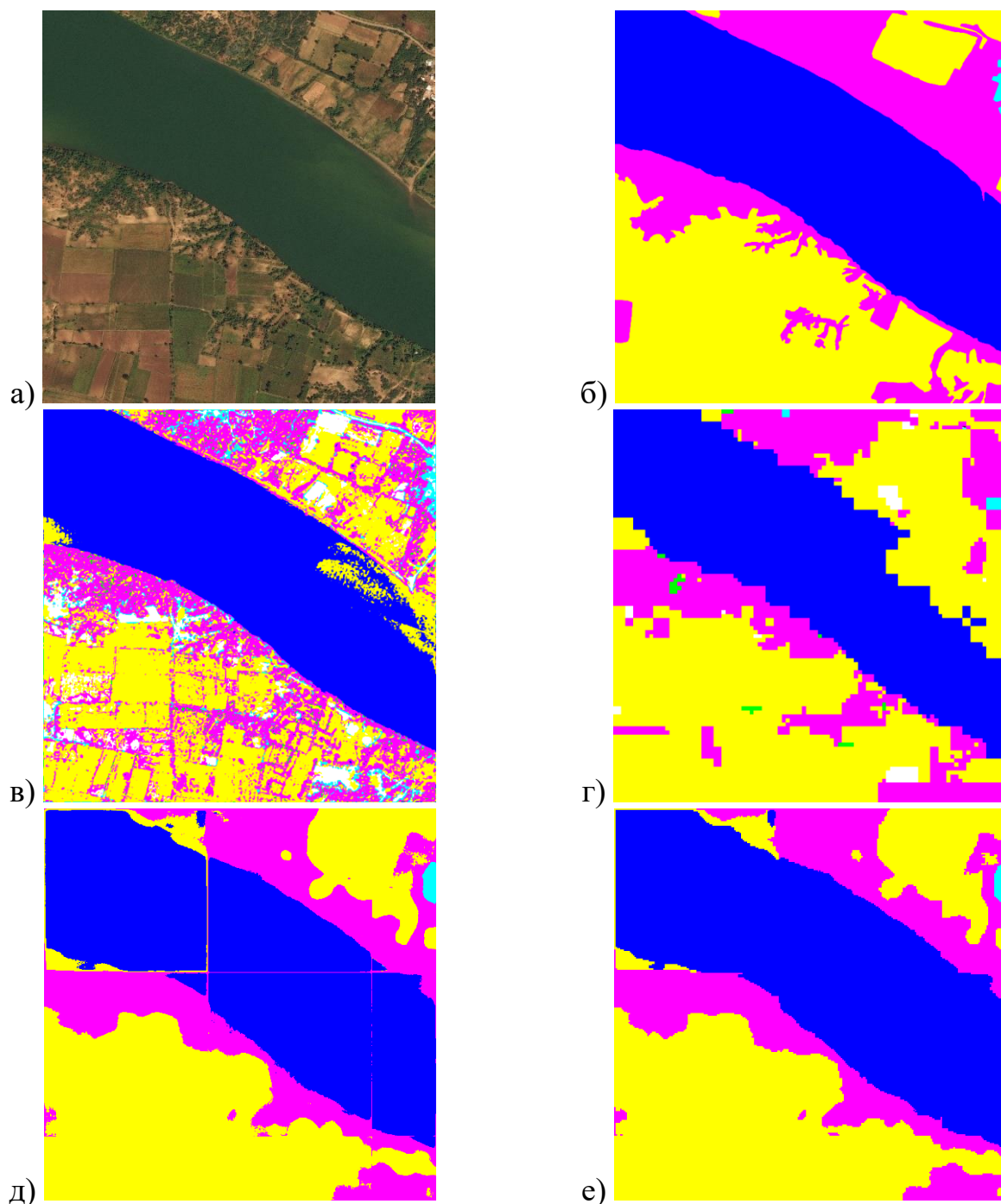


Рисунок 3.11 – Иллюстрация работы построенного алгоритма:  
 а) исходное изображение; б) исходные размеченные данные;  
 в) результат работы ИНС; г) отфильтрованное изображение;  
 д) результаты работы DeepLabv3+; е) отфильтрованное изображение

Тестирование обученной ИНС и операций постобработки проводилось над тестовым набором данных, составляющим 80 (10%) пар изображений. Показатели работы алгоритма продемонстрированы в таблице 3.9 вместе с результатами

других исследователей (звёздочками отмечены алгоритмы с постобработкой, жирным шрифтом выделены исследуемые в параграфе алгоритмы).

Строка 5 отвечает за реализацию подхода с использованием ИНС с одним скрытым слоем, содержащим 100 нейронов, в качестве интеллектуальной модели на имеющемся оборудовании *AMD Radeon 530* в среде *Matlab*. Стоит отметить, что применение указанной постобработки даже с использованием простой интеллектуальной модели показывает уровень качества сопоставимый с применением глубоких нейросетей. Стоит отметить, что количество изначальных обучаемых данных составляло порядка 0,01% от всего датасета, при этом с помощью постобработки качество работы ИНС было поднято с 30,8% до 56,1%. Для сравнения при оптимизации выбора обучающих данных в размере 2% от датасета было получено улучшение с 40,3% для сети *s4GAN*, до 51,1% с помощью постобработки [144].

Строки 6 и 7 отвечают за запуск в виртуальной среде *Kaggle* с использованием использования видеокарты *Nvidia Tesla P100*. В качестве интеллектуальной модели использовалась *DeepLabV3+* [129], использующая *CNN ResNet50* с предобученными весами на основе *ImageNet*. Оригинальный датасет был разделён в пропорции 80:20, тренировочные данные агентировались с помощью команд *RandomCrop*, *HorizontalFlip*, *VerticalFlip*. Данные нарезались на куски размера 1024×1024 пикселей. Обучение происходило с использованием функции потерь *Dice Loss* (обычно используемой для несбалансированных датасетов), а качестве оптимизатора использовался *Adam*, *learning rate*  $8e-5$ . В качестве постобработки использовались коэффициенты с линейной коррекцией. Для подбора использовался описанные ранее алгоритм, полученные коэффициенты равны:

- |                               |                          |
|-------------------------------|--------------------------|
| – <i>urban</i> : 1;           | – <i>forest</i> : 1,477; |
| – <i>agriculture</i> : 2,400; | – <i>water</i> : 0,522;  |
| – <i>rangeland</i> : 7,665;   | – <i>barren</i> : 0,627  |

Приведённые в таблице результаты из статьи, посвящённой разработке *MKANet* [274], подвергаются сомнению: в статье также приведена информация о запуске обучения нейросети *DeepLabv3+* для решения данной задачи при сравнимом

оборудовании. При этом у нас обучение на 5 эпохах заняло 2,5 часа, а в статье обучение 200 эпох на датасете в 4 раза большем произошло за 5,4 часа [264], что сложно объяснить даже лучшими способами оптимизации.

Таблица 3.9 – Показатели работы алгоритмов для обработки датасета конкурса *DeepGlobe 2018 Land Cover Classification*

№	Год	Подход	<i>mIoU</i> , %	Дополнительная информация
1	2018	Demir [143]	43,3	–
2		FCN-32s [190]	45,9	–
3		DeepLabv3 [190]	46,8	–
4		DeepLabv3* [190]	51,0...52,7	Оборудование: GTX 1080Ti Время обучения: 10 часов
5	2022	<b>ИНС*</b>	56,1	Оборудование: AMD Radeon 530 Время обучения: 2.5 часов Время обработки изображения: 160 сек
6	2022	<b>DeepLabv3+</b>	56,2	Оборудование: NVIDIA Tesla P100 Время обучения: 2,5 часа Время обработки изображения: 1,3 сек
7	2022	<b>DeepLabv3+*</b>	62,0	Оборудование: NVIDIA Tesla P100 Время обработки изображения: 1,3 сек
8	2022	МКАNet [274]	70,7...72,0	Оборудование: i5-3470 CPU, NVIDIA RTX 3060 12G GPU Время обработки изображения: 0,1 сек

### 3.4 Выводы

В главе проведено экспериментальное исследование алгоритмов, предложенных в Главе 2, а именно по обработке точечных и площадных объектов. Для исследования алгоритмов обработки ТО были смоделированы участки навигационных карт, на которых был произведён поиск и распознавание отметок глубин, выполненных на основе датасета рукописных цифр *MNIST*. Особенностью являлось то, что изначальные цифры размера  $28 \times 28$  приводились к размеру  $13 \times 11$ . Отметки глубин могут располагаться на разных точках карты, в том числе их могут пересекать другие объекты. Ранее большинство исследователей занимались только

непосредственным распознаванием чистых данных, не используя их в более реальном прикладном применении.

Эксперименты заключались в получении оптимальных структур интеллектуальных моделей и параметров выполняемых операций. При исследовании алгоритма сегментации была получена оптимальная структура ИНС: 27/13/1. При выполнении генетического алгоритма для поиска морфологических операций и операций фильтрации была получена хромосома, состоящая из 17 операций. Данные результаты также являются новыми, так как до этого подобные генетические алгоритмы строили или для локализации простых объектов без возможных пересечений или для оптимальной морфологической обработки площадных объектов.

Построена двухкаскадная система для распознавания цифр, использующая как метод скользящего окна, так и метод анализа суперсегментов, что является новым результатом. В итоге были получены оптимальные структуры нейронных сетей: 143/400/10 для распознавания скользящих окон размера  $13 \times 11$  по 10 классам на первом каскаде, 143/100/1 для распознавания по бинарному классу «цифра/не цифра», 10/100/10 для распознавания суперсегментов по 10 классам для второго каскада. После распознавания производились операции группировки ТО и фильтрации. Итоговая точность составила 89,1% для существующих отметок.

Для исследования алгоритмов обработки ПЛО были использованы материалы конкурса *DeepGlobe 2018 Land Cover Classification*. Эксперимент прежде всего заключался в проверке возможности использования нейросетей простой структуры и последующей постобработки вместо обработки с помощью глубоких нейросетей.

В результате исследования была предобучена ИНС со структурой 96/100/6. С помощью подбора были найдены корректирующие коэффициенты выхода нейросети и размеры скользящих окон фильтров. С помощью постобработки критерий качества *mIoU* был для данной ИНС поднят с 30,8% до 56,1%, что является хорошим результатом относительно других исследователей, занимающихся оптимизацией с помощью постобработки.

## **Глава 4. Разработка и тестирование программного обеспечения автоматизированной обработки топографических карт**

В данной главе приведён анализ и обработка цифровых сканированных топографических карт с точки зрения задачи получения цифровой модели рельефа: исследованы правила получения и понимания ЦТК, проведено абстрагирование от несущественных для построения ЦМР особенностей, выделены основные проблемы для распознавания. В дополнение к разработанным в Главе 2 алгоритмам предложен метод комплексного анализа полученных данных и алгоритм интерполяции, учитывающий специфику распределения данных на изображении ЦТК. На основе предложенных алгоритмов разрабатывается и тестируется программное обеспечение автоматизированной обработки ЦТК.

### **4.1 Особенности изображения топографических карт, как типичного представителя сложноструктурных изображений**

Геологические и топографические карты являются полноцветными изображениями, сочетающими в себе как строгие правила, так и высокую степень нерегулярности. Например, подписи горизонталей присутствуют не на всех линиях, отметки зачастую не имеют конкретного места приложения, а принадлежат некоторым областям, поэтому могут быть расположены в случайном месте этой области и т. д. Такие карты имеют большое число функциональных слоёв, которые могут накладываться друг на друга. Тем не менее, большая информативность позволяет использовать топографические и геологические карты, как основу для данных геоинформационных систем (ГИС). Дальнейшие исследования будут проводиться для случая растровых топографических карт, как типичных представителей ССИ.

Программная система, в которой можно было бы совершать операции по анализу и обработке ЦТК, является частным случаем геоинформационной системы (ГИС). В ГИС могут быть введены геоданные, которые могут пересекаться в

пространстве, поэтому эти данные необходимо разделять по выполняемой функции. Делается это при помощи введения дифференцированных слоёв геоданных, например, на одном слое располагается информация о высоте рельефа, на другом – о характере почв, на другом – о существующих объектах гидрографии. Такое разделение позволяет работать с определённым видом геоданных независимо от других, производить анализ и обработку рационально с точки зрения используемого процессорного времени и памяти.

Например, для поддержки принятия решения при проектировании размещения коммуникаций и логистических объектов ГИС должна обладать следующими слоями геоданных: рельеф, объекты гидрографии, почва и растительность, коммуникации и культурные строения. Данная информация нужна, чтобы определить оптимальное расположение проектируемых объектов, например, траекторию наиболее дешёвого или короткого пути между двумя пунктами. Учёт объектов гидрографии и характера почв позволит рассчитать, нужно ли производить строительство мостов (и где оптимально это можно сделать), осушение болот, бурение тоннелей, возведение насыпей, трамбование. Для проектирования дорог уже существует множество САПР, например, *IndorCAD/Road*, *AutoCAD Civil 3D*, *MXROAD* и т. д. Они используют на входе цифровые карты местности или взаимодействуют с программными комплексами ГИС.

Топографические карты условно отображают местность, поэтому их база образов содержит модели условных знаков, линий и областей. На топографических картах все объекты имеют свои условные изображения, для Российской Федерации они описаны в [95]. Наличие таких правил позволяет создавать систему, в которой могут учитываться все типы реальных объектов. Тем не менее, даже построение такой системы является непростой задачей, поэтому работы в этой области посвящены в основном решению конкретных подзадач распознавания. Полученную ГИС можно использовать вновь для совместной обработки с космическими снимками [59] или топографическими картами [1]. Предложенные алгоритмы и методы используются для решения временных задач (обновления информации о местности,



вызванной строительством новых зданий, изменения течения рек и т. д.).

Актуальность проблемы оптимального анализа местности, например, для проектирования объектов транспортно-логистической инфраструктуры послужила причиной разработки современных автоматизированных систем преобразования топографических карт в ГИС, как в России, так и за рубежом, таких как: *Easy Trace*, *ArcGIS*, *ArcMap*, *SpotLight*, *MapInfo*, ГИС «Панорама», *AutoCAD Map 3D* и т. д. В данных продуктах разработаны свои алгоритмы и методы распознавания образов топографических карт, являющиеся приложением существующих классических методов обработки, и, в связи с коммерческой тайной, не являющихся открытыми, что не позволяет осуществить их сравнение между собой с помощью заданных критериев качества.

Акцент выполняемой обработки таких программ основан на автоматизации предварительной обработки изображения карты, оставляя выполнение как интеллектуальных, так и простых операций за человеком. Например, векторизация горизонталей ТК выполняется либо ручным способом, либо в виде автоматизированной трассировки с полным контролем оператора со скоростью 2-5 пикселей в секунду. Никакие из существующих программных продуктов не осуществляют комплексный анализ карты: каждые отдельные процедуры выполняются независимо от выполнения других, распознавание отдельных символов либо отсутствует, либо воплощено с низким процентом распознавания, что говорит о необходимости разработки алгоритмов компьютерного зрения, с помощью которых можно как можно сильнее повысить процент операций, выполняемых в автоматическом режиме.

Задачей исследования в данной главе было поставлено получение цифровой модели рельефа максимально соответствующей рельефу реальной местности, используя изображения соответствующих ЦТК. Концептуально система анализа и обработки ЦТК рассмотрена в статьях [29, 36, 46, 150]. Под цифровой моделью рельефа (ЦМР) принято понимать средство цифрового представления трехмерных пространственных объектов (поверхностей или рельефов) в виде трёхмерных данных, образующих множество высотных отметок в узлах регулярной или нерегулярной

сети или совокупность записей горизонталей. Рельеф местности можно представить в виде непрерывной функции высоты от двух географических координат:

$$h = \text{height}(\text{latitude}, \text{longitude}), \quad (4.1)$$

где *latitude* – непрерывная координата, соответствующая широте, *longitude* – долготе. Эту функцию можно представить в виде бесконечной суммы степенных функций географических координат, так как она нерегулярна по своей природе.

Так как на ЦТК кроме отображения рельефа необходимо размещать другую информацию, то для уменьшения информационной перегрузки используется редукция. При этом непрерывная функция, определённая на некотором множестве географических координат, представляется набором горизонталей, соответствующих некоторым высотам (то есть происходит дискретизация по высоте), и точек с обозначением их высот. Очевидно, что происходит потеря информации: теперь мы точно можем знать лишь высоты, для которых у нас есть данные, остальные придётся получать с помощью интерполяции, которая не может быть абсолютно точной, таким образом, возникает зависящая от способа интерполяции погрешность:

$$h = \sum_{i=1}^n (\omega_i * [\text{height}_i](\text{latitude}, \text{longitude})) + \text{error}(\text{latitude}, \text{longitude}), \quad (4.2)$$

где  $[\text{height}_i]$  – значения высот от ближайших к данным координатам горизонталей или отметок высот (дискретное значение),  $\omega_i$  – весовой коэффициент данного значения, который зависит от расстояния, *error* – непрерывная функция ошибки.

В процессе преобразования поверхности рельефа в печатную модель с неизбежностью происходит процесс дискретизации по географическим координатам. Такой же процесс происходит и в течение сканирования. Точность печати и

сканирования зависит от их разрешающей способности, измеряемой в  $dpi$  – количестве точек (пикселей) на дюйм. Таким образом, финальная математическая модель рельефа представляет собой формулу:

$$h = \sum_{i=1}^n (\omega_i * [height_i]([latitude], [longitude])) + error([latitude], [longitude]), \quad (4.3)$$

где  $[latitude]$  и  $[longitude]$  – дискретные значения географических координат.

Далее ЦМР можно использовать для решения прикладных задач, например, оптимальной прокладки железнодорожного полотна одновременно в плане и профиле [33, 35, 151]. Получить данные о высоте можно с помощью спутников (например, данные *SRTM* [219]), но эта информация зачастую обладает слишком большой погрешностью (шаг сетки по поверхности 30 м, среднеквадратичная погрешность значения высоты 10 м, абсолютная погрешность до 16 м для 90% точек).

Таким образом, чтобы получить ЦМР, необходимо правильно определить географические координаты каждого пикселя, для каждого пикселя получить значение высот ближайших горизонталей и отметок высот и интерполировать данные.

Географические координаты каждого пикселя можно получить с помощью геопозиционирования, в т. ч. автоматического [168]. На каждой ТК присутствует обозначение широты и долготы либо вершин прямоугольника, ограничивающего карту, либо такая отметка ставится внутри карты. При этом можно объединять ЦМР соседних местностей, полученных из различных ТК в одну матрицу высот без необходимости непосредственного согласования карт между собой.

Основной задачей исследования ставится именно определение значений высот ближайших горизонталей и отметок высот. Для этого необходимо проанализировать правила изображения и понимания ТК, которые даны в правилах изображения условных знаков. К сожалению, существует несколько таких сводов правил, которые так и не стали государственными стандартами, поэтому могут

существовать различные разночтения, аналогичные проблемы возникают и в других странах [216, 219]. При этом издатели ТК в целом следуют некоторому базовому ядру правил, пытаясь обеспечить, прежде всего, правильность и наглядность.

ТК содержит большой объём геораспределённых данных о различных типах объектов, размещённых на местности. Стоит выделить несколько правил, необходимых для решения нашей задачи, чтобы абстрагироваться от элементов, которые не касаются напрямую особенностей рельефа или способов его отображения.

Для возможности идентификации и распознавания функции объектов на ТК их отображения в виде условных знаков имеют отличительные цвета и способы начертания. Так, горизонтали нарисованы коричневым цветом, значение их высот определяет толщину начертания: горизонтали с укрупнённым шагом рисуются утолщённой линией, с полушагом – штриховой. При этом стоит отметить значительную нерегулярность при отображении штриховых горизонталей: они в отличие от обычных могут быть незамкнуты и располагаться достаточно рядом, из-за чего их можно некорректно соединить с обычными горизонталями [235].

Исходя из своей сущности, горизонтали не пересекаются друг с другом и являются гладкими кривыми. Для определения направления ската могут быть использованы бергштрихи, которые обязательно изображаются на вершинах, впадинах и седловинах для быстрого определения характера выпуклости местности. Также определению склонов может помочь направление течения рек и расположение цифровых подписей горизонталей: они должны ориентироваться своим основанием вниз по склону.

Отметки высот изображаются чёрным цветом со строгой горизонтальной ориентацией, в отличие от подписей горизонталей; они отображаются вдоль линий. На вершинах и впадинах в точках локальных экстремумов обязательно должны находиться отметки высот. Чёрный цвет подавляет любые другие цвета на изображении ТК, поэтому может быть легко обнаружен, но именно чёрный цвет является главной причиной разрыва горизонталей.

Указанные правила не всегда строго соблюдаются: различные издательства могут использовать свои способы отображения особенностей рельефа или использовать другие цвета при печати. Под влиянием времени ТК обесцвечиваются и из-за неравномерного способа печати цвета на одной стороне карты могут оказаться светлее или темнее, чем на другой стороне. Поэтому на начальном этапе необходимо провести предобработку, которая могла бы подвести разного вида ЦТК под один вид, с которым можно было бы работать. При этом данная обработка может быть выполнена с использованием простейших методов, регулирующих контрастность, яркость и насыщенность цвета, но которая бы не приводила к тому, что различные объекты имели бы одинаковое начертание или цвет.

Одной из важных особенностей является также то, что значения отметок высот изображаются не на каждой горизонтали. Исходя из этого, нужно доопределять высоты у неуказанных горизонталей. При этом могут возникнуть рассогласования, возникающие из-за неверного распознавания высот на соседних горизонталях.

На некоторых картах неоправданно экономят на дополнительных элементах, позволяющих быстрее распознать особенности рельефа. На них могут отсутствовать бергштрихи, подписи горизонталей, отметки высот на вершинах или впадинах. Горизонтالي особенно на равнинных местностях могут располагаться с очень большим расстоянием друг от друга, что уменьшает качество интерполяции.

Другой общей проблемой при обработке ЦТК является их дискретизация. Толщина горизонталей, согласно стандарту, составляет 0,1 мм для обычных линий и 0,2 мм для утолщённых. При этом печать с разрешением 200-300 *dpi* (типичным для ТК) приводит к тому, что толщина одного пикселя составляет примерно 0.13 мм, то есть горизонтали имеют толщину 1-2 пикселя. Учитывая то, что горизонтали часто прерываются другими объектами карты и могут истираться со временем, получаем, что горизонтали не являются непрерывными объектами, они скорее состоят из сегментов длиной от нескольких миллиметров до нескольких сантиметров. Но, так как сами горизонтали часто имеют длину до нескольких десятков сантиметров, то их можно восстановить из сегментов.

Сходная проблема возникает и в отображении значений высот. Например, для карт масштаба 1:50000 высота символов должна составлять 1.3 мм для горизонталей и 1.5 мм для отметок высот. Таким образом, получаются числа высотой в 10-13 пикселей и толщиной в 7 пикселей, которые ещё могут пересечься с другими объектами. Для объектов такого размера невозможно использование распознавания оптическими методами или методами, использующими поиск особенностей, а только интеллектуальные методы. Но и при этом остаётся довольно высокая вероятность ложного распознавания.

Свою роль в сложности распознавания горизонталей играет и рельеф. В масштабе 1:50000 и 200 *dpi* каждый пиксель соответствует примерно 6 метрам, что приводит к тому, что при изображении крутых склонов производители карт часто допускают ошибку в сращивании нескольких горизонталей, что невозможно в реальном случае (отвесные скалы и склоны с отрицательным уклоном имеют отдельное обозначение). Данные сращивания необходимо разделять и находить вариант правильного соединения горизонталей, обеспечивающего их непересечение.

Последним вопросом является интерполяция распознанных геораспределённых данных. Для начала необходимо выбрать тип представления поверхностей ЦМР. Существует два основных способа: растровый и нерегулярная треугольная сеть (*Triangulated Irregular Network, TIN*). Последний может быть получен посредством триангуляции Делоне. Полученная ЦМР обладает меньшим информационным объёмом, чем растр, и её обработка может занять меньшее время. Тем не менее, такой способ обладает существенными недостатками, в *TIN* могут возникать различные артефакты и объекты с заведомо невозможными геоморфологическими поверхностями, так как интерполяция в *TIN* строго линейна. Поэтому в качестве представления был выбран растр.

После распознавания значений высот всех горизонталей и отметок высот мы получаем матрицу, на которой отмечены высоты всех пикселей, соответствующих горизонталям, и отдельные точки. При интерполировании следует учитывать, что толщина линий горизонталей составляет примерно 1-2 пикселя, а расстояние

между горизонталями может составлять порядка 30-50 пикселей, а то и значительно больше. То есть в этой матрице существуют области как перенасыщенные данными (около горизонталей), так и области разряженных данных.

Соответственно, если использовать данные полностью, то около горизонталей могут быть получены области одинаковой высоты, что не соответствует реальности. Количество пикселей, соответствующих отметкам высот несоизмеримо меньше количества пикселей, соответствующих горизонталям, что также влияет на качество интерполяции. Но если мы будем использовать слишком маленькую долю точек, полученных от горизонталей, то получим ЦМР, которая не будет соответствовать рельефности местности.

Так как существующие исследования по анализу и обработке ЦТК обрабатывают разные по правилам и способам отображения карты (обычно из-за разных локальных стандартов в разных странах), то мы будем сравнивать полученные результаты с другими исследованиями как есть, но будем отмечать особенности разных датасетов, чтобы оставлять возможность интерпретации. При этом одной из критичных характеристик будет являться разрешение и качество сканированных ЦТК.

## **4.2 Алгоритм комплексного анализа данных для ЦТК**

Учитывая разработанные ранее алгоритмы, мы можем считать, что к настоящему моменту получены значения и положение отметок высот, значения высот у горизонталей, у которых есть подписи, и направление ската у горизонталей, у которых есть подписи или бергштрихи. Этого уже может быть достаточно для получения примерной ЦМР, но нас больше интересует точность, поэтому необходимо доопределить остальные горизонтали. Может оказаться, что какие-то из наших значений высот были распознаны неверно, поэтому нельзя полагаться на них полностью. Прототип разработанного здесь алгоритма представлен в [40].

Пользуясь определением горизонталей, и зная, что они могут быть замкнуты только на самих себя или границы карты, можно заключить, что план горизонталей обладает иерархической структурой. Рассмотрим план горизонталей как граф с множеством вершин, хранящих значения высот, рёбрами тогда будет возможность перехода с одной высоты на другую.

Если добавить разницу значений высот, которую нужно преодолеть, переходя из одной вершины в соседнюю, то граф будет взвешенным и ориентированным. Если из вершины  $v_1$  в вершину  $v_2$  идёт ребро  $e_{12}$  с весом  $w$ , то можно сказать, что из вершины  $v_2$  в вершину  $v_1$  идёт ребро  $e_{21}$  с весом  $-w$ . В направлении от узла с большим значением высоты к узлу с меньшим значением (направление ската) разницу будем считать положительным значением. Перейти с одной горизонтали на другую можно и по ребру с нулевым весом, например, в случае нахождения рядом двух соседних вершин.

Построение неориентированного графа соседства по существующему плану горизонталей возможно без знания высот горизонталей, а только по изображению плана. Для того чтобы данный граф стал ориентированным и взвешенным, необходимо получить такую важную информацию, как направление рёбер между соседними значениями, то есть направление ската. Зная эту информацию и хотя бы одно абсолютное значение высоты, в котором мы уверены (это значение может быть получено и от оператора), можно восстановить весь план горизонталей.

Информацию о разнице между соседними значениями высот горизонталей (интервал основного сечения) можно узнать исходя из масштаба карты, или её может ввести оператор.

Информацию о направлении отдельных рёбер можно узнать, по изображению бергштрихов или ориентации надписей у горизонталей. Сложность в автоматическом определении бергштрихов заключается в том, что они имеют достаточно маленькие размеры и могут распознаться как шумовые элементы. Их указание может быть произведено в полуавтоматическом режиме с контролем оператора.



Бергштрихи отображаются, если горизонталь имеет внутри локальный экстремум или при наличии седловин. Можно выделить такое неявно выраженное правило: при отсутствии бергштрихов соседние горизонтали имеют то же направление ската, что и соседние. Если же у соседних горизонталей скат не определён, то нужно смотреть на их соседние горизонтали и т. д. Наличие пары бергштрихов определяет седловины, то есть участки местности, в которых две соседние горизонтали обладают одинаковым значением высоты. Стоит учесть нарушение правил при отображении бергштрихов, поэтому полностью автоматизировать их нахождение не представляется возможным, но можно найти большую их часть и доопределить необходимые горизонтали с помощью оператора.

Для получения полного плана горизонталей построим алгоритм (рис. А.4), основанный на анализе графа соседства горизонталей. Определить, являются ли две горизонтали соседними, то есть, можно ли попасть с одной горизонтали на другую, не пересекая другие горизонтали, можно, используя морфологические операции над пространствами, окружаемыми горизонталями.

Проведём морфологическую операцию дилатации для пространств, получаемых вычитанием из множества пикселей карты множества пикселей, принадлежащих горизонталям. Затем вычтем это пространство и получим множество пикселей, принадлежащих только горизонталям, в каждую из которых можно попасть из другой, не пересекая остальные.

Проинициализируем направление ската горизонталей, у которых присутствуют бергштрихи или у которых есть подпись. Так как наш граф не является деревом в общем случае (циклы могут возникать в областях седловин), то отсутствие необходимых направлений ската приведёт к появлению ошибки в определении высоты горизонтали, которая будет сказываться на всех следующих восстанавливаемых горизонталях.

Так как каждая горизонталь либо замкнута на саму себя, либо на границы карты, то она разбивает плоскость карты на две части, согласно теореме Жордана. Если бы мы не были ограничены рамками карты, то все горизонтали были бы

самозамкнутыми, но так как карты изображаются лишь участки местности, то большое количество горизонталей размыкается границами карты. Поэтому нельзя использовать такие понятия, как внешнее и внутреннее полупространство. Будем считать главной полуплоскостью горизонтали ту, которой принадлежит меньшее число пикселей карты, а другую будем считать фоновой. Условно будем считать, что пиксели, находящиеся на главной полуплоскости, находятся внутри горизонтали, а те, что находятся на фоновой – снаружи. Стоит отметить, что данные понятия строго совпадают только со случаем самозамкнутой горизонтали, а для замкнутых на границы горизонталей всё может быть с точностью до обратного варианта, поэтому необходимо реализовать программу, учитывающую этот момент.

Далее необходимо построить граф, начиная с любой горизонтали (далее назовём её базовой), для которой известно значение ската, её относительную высоту будем считать равной нулю. Используем метод обхода графа в ширину: изначально все вершины, кроме базовой, пометим белым цветом, а базовую пометим серым. Для серых вершин необходимо найти все соседние белые вершины, обработать их (то есть определить высоту и скат) и пометить их серыми; рассмотренные вершины становятся чёрными. Продолжаем итеративный процесс, пока все вершины не станут чёрными.

Обработка белой вершины зависит от того, есть ли на соответствующей ей горизонтали скат или нет. Если нет, то необходимо определить, находится ли соседняя линия внутри или снаружи от рассматриваемой. Если главные полуплоскости двух горизонталей пересекаются, то скат сохраняется, иначе скат меняет значение. Значение высоты белой горизонтали вычисляется исходя из значения высоты серой горизонтали, её ската и того, находится ли белая горизонталь внутри или снаружи серой.

Если значение ската присутствует на соседней линии, то необходимо идентифицировать номер соседних подпространств, находящихся снаружи и внутри от каждой горизонтали. Были получены следующие формулы для определения высоты белой горизонтали:

$$z_j = \begin{cases} z_i + ((N_{in,i} = N_{in,j}) - (N_{out,i} = N_{out,j})) * s_i * step, & s_i = -s_j \\ z_i + ((N_{in,i} = N_{out,j}) - (N_{out,i} = N_{in,j})) * s_i * step, & s_i = +s_j \end{cases}, \quad (4.4)$$

где  $z_i$  – относительная высота  $i$ -ной горизонтали,  $N_{in/out,i}$  – номер соседнего подпространства, находящегося внутри или снаружи  $i$ -ной горизонтали,  $s_i$  – значение ската  $i$ -ной горизонтали,  $step$  – значение основного шага.

Пример орграфа, соответствующего ТК, представлен на рис. 4.1. По рисунку видно, что на изображении карты присутствует достаточно небольшое количество возвышенностей (с них начинается направление движения стрелок слева сверху) и низин (справа снизу), а середина представляет собой достаточно сложную структуру.

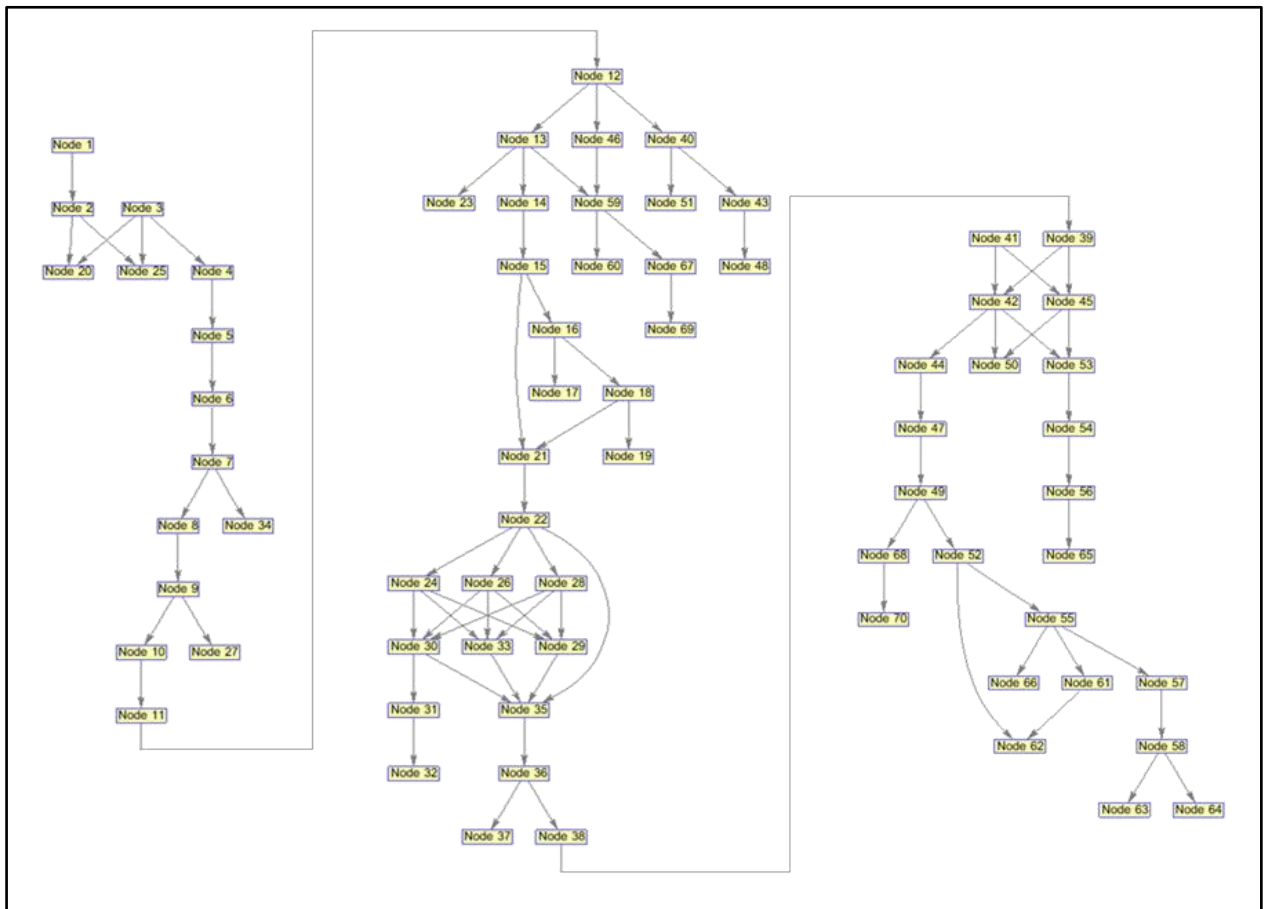


Рисунок 4.1 – Граф, отражающий соседство горизонталей ТК

Восстановление следует проводить одновременно для всех горизонталей с известным значением, так как если делать это только для одной, то в результате ошибки распознавания высоты горизонтали на предыдущем этапе всем остальным горизонталям будут присвоены неправильные значения.

Выполняем данный алгоритм, пока присваивание высот горизонталям не остановится, и выполним анализ по поиску возможных ошибок используя тот факт, что информация о значениях высот горизонталей является избыточной для удобства восприятия.

Так как мы знаем высоты базовых горизонталей (после распознавания) и относительные высоты, то мы можем для каждой базовой горизонтали восстановить значения всех высот. Учтём, что верно распознанные значения горизонталей порождают верные значения высот у других, а неверно распознанные – неверные, и вероятность распознавания чисел в среднем не меньше 50% (фактически точность большая, но для этого алгоритма достаточно и этой).

Если построить таблицу, в которой каждый отдельный столбец будет представлять собой упорядоченный (по идентификатору горизонтали) набор вычисленных значений горизонталей, то тогда для решения задачи нужно найти набор столбцов, количеством не меньший 50% от всех, в котором стоят одинаковые высоты. Следовательно, по этим горизонталям мы определяем остальные высоты горизонталей, в том числе и у тех, которые были распознаны неверно. Для распознавания большинства значений достаточно также и моды. Полное распознавание может быть возможно только в том случае, если нет последовательно неверно распознанных значений с одинаковым уровнем ошибки, вероятность чего является достаточно низкой. Например, вероятность того, что у двух последовательных горизонталей будет произведена одинаковая ошибка в десятках, причем на одну цифру, составляет порядка  $0.01p^2$ , где  $p$  – вероятность ошибки распознавания цифры.

Совокупная сложность алгоритма зависит от разных факторов: размер используемых структурных элементов морфологических операций, количество горизонталей, количество пикселей, занимаемых горизонталями, количество

горизонталей, соседствующие с данной и т. д., но в среднем для реальных карт можно считать, что сложность данного алгоритма составляет порядка  $O(N_{\text{гор}} * S)$ , где  $N_{\text{гор}}$  – количество горизонталей на карте. В худшем случае, при большом числе возвышенностей и низин на примерно одном уровне, размеры кластеров (наподобие изображённых в середине рис. 4.2) могут стать большими, что может привести к классу сложности  $O(N_{\text{гор}}^2 * S)$ .

Пример результирующей таблицы изображён в таблице 4.1. Значения в каждом столбце вычисляются на основе значения базовой линии, которое выделено жирным шрифтом. В значения высот третьей, шестой, седьмой и одиннадцатой горизонталей были внесены ошибки. Алгоритм перестанет работать тогда, когда верные значения горизонталей станут только второй модой, что возникнет (при сохранении случайности при неверном распознавании) только при очень низком качестве распознавания, например, когда верно будут распознаны только 1-2 отметок из всех, присутствующих на карте.

Таблица 4.1 – Пример результата работы алгоритма

	Различные базовые горизонталей										
Вычисленные значения горизонталей	240	240	216	240	240	220	242	240	240	240	239
	<b>250</b>	250	226	250	250	230	252	250	250	250	249
	260	260	236	260	260	240	262	260	260	260	259
	300	<b>300</b>	276	300	300	280	302	300	300	300	299
	310	310	286	310	310	290	312	310	310	310	309
	350	350	<b>326</b>	350	350	330	352	350	350	350	349
	360	360	336	360	360	340	362	360	360	360	359
	350	350	326	<b>350</b>	350	330	352	350	350	350	349
	340	340	316	340	340	320	342	340	340	340	339
	400	400	376	400	<b>400</b>	380	402	400	400	400	399
	300	300	276	300	300	<b>280</b>	302	300	300	300	299
	400	400	376	400	400	380	<b>402</b>	400	400	400	399
	450	450	426	450	450	430	452	450	450	450	449
	400	400	376	400	400	380	402	<b>400</b>	400	400	399
	390	390	366	390	390	370	392	390	390	390	389
	300	300	276	300	300	280	302	300	<b>300</b>	300	299
	450	450	426	450	450	430	452	450	450	<b>450</b>	449
460	460	436	460	460	440	462	460	460	460	<b>459</b>	

### 4.3 Алгоритм интерполяции

Прототип разработанного в данном параграфе алгоритма представлен в работе [38]. Основная идея интерполяции двумерной функции в целом связана с использованием средневзвешенной функции по значению высот  $n$  соседних точек:

$$Z_0 = \sum_{i=1}^n w_{i0} * Z_i, \quad (4.5)$$

где  $Z_0$  – высота определяемой точки,  $Z_i$  – высоты соседних точек,  $w_{i0}$  – весовой коэффициент между текущей точкой и  $i$ -ной соседней.

Интерполяторы отличаются способом выбора соседних точек и функциями весовых коэффициентов. Множество соседних точек можно выбрать с помощью двух подходов: все в определённой окрестности заданного радиуса или ближайšie в заданном количестве. Функция весового коэффициента зависит от расстояния между точками с возможным наличием параметров сглаживания. При сглаживании расчётное значение в точке может не совпадать с известным в ней значением. В случае топографических карт обязательным условием для выбора является наличие соседних точек с разными высотами, так как иначе могут получиться обширные двумерные области с одинаковой высотой. Стандартные методы интерполяции могут порождать систематическую ошибку, так как не ориентированы на специфику ЦТК, поэтому это также следует отследить при реализации алгоритма.

В качестве базового принципа интерполяции используем утверждение: при сохранении точности ЦМР выбираем тот способ интерполяции, в котором лучше передаются особенности рельефа. Это означает, что интерполятор должен как можно меньше сглаживать рельеф, представленный картой горизонталей: например, если построить новую карту горизонталей с меньшим в два раза шагом сечения, то полученные дополнительные линии должны иметь примерно одинаковое расстояние до обеих горизонталей с большим шагом.

Все интерполируемые точки можно разделить на 3 типа: находящиеся между двух горизонталей различной высоты, расположенными внутри замкнутой горизонтали локального экстремума высоты или между границей карты и ближайшей горизонталью. Типизация осуществляется с помощью поиска ближайших точек по всем направлениям: если точки относятся к двум горизонталям, то это первый тип, если к одной, то второй, а если к одной горизонтали и границам, то третий.

Учитывая базовый принцип для интерполяции точек первого типа, выберем одинаковое количество ближайших соседних точек из двух соседних горизонталей, внутри которых находится текущая точка. В качестве весового коэффициента рассмотрим функцию от величины, обратную расстоянию до соседней точки. При этом около первой горизонтали всегда будут выбраны ближайшие точки, что будет способствовать сохранению рельефности, а точки второй горизонтали будут способствовать изменению высоты комплексно.

Если считать, что точка находится вблизи первой горизонтали, то формула средневзвешенной функции будет принимать такие значения (для простейшего случая  $w_{i0} = 1/d_{i,0}$ ):

$$Z_0 \approx \frac{Z_1 * \sum_{i=1}^n \frac{1}{d_{i,0}} + \frac{Z_2 * n}{D_{2,0}}}{\sum_{i=1}^n \frac{1}{d_{i,0}} + \frac{n}{D_{2,0}}} = \frac{Z_1 * \sum_{i=1}^n \frac{D_{2,0}}{n * d_{i,0}} + Z_2}{\sum_{i=1}^n \frac{D_{2,0}}{n * d_{i,0}} + 1} = \frac{\rho * Z_1 + Z_2}{\rho + 1}, \quad (4.6)$$

где  $Z_1, Z_2$  – высоты первой и второй горизонталей,  $d_{i,0}$  – расстояние от  $i$ -ной точки первой горизонтали до текущей,  $D_{2,0}$  – среднее расстояние от текущей точки до второй горизонтали,  $\rho = \sum_{i=1}^n \frac{D_{2,0}}{n * d_{i,0}}$  – среднее отношение расстояния ко второй горизонтали к расстоянию до точек первой.

Полученная формула с одной стороны отражает влияние формы рельефа вблизи горизонталей (при  $d_{i,0} \rightarrow 0: \rho \rightarrow \infty, Z_0 \rightarrow Z_1$ ), а с другой стороны

показывает, что посередине между двумя горизонталями ( $\rho = 1$ ) будет получаться среднее значение высот.

Проблема данного подхода заключается в том, что так как используется такое понятие, как ближайший элемент, то в пространстве ЦМР будут существовать кривые, в которых может происходить резкий пересчёт формулы для новых точек, что приведёт к нарушению непрерывности.

На рис. 4.2 показан пример на сгенерированной поверхности, для которой были получены горизонтали высот и промежуточные точки были проинтерполированы с помощью линейного интерполятора и предложенного: белым показано максимальное значение ошибки. Видно, что непрерывность нарушается в тех точках, расстояние от которых до двух соседних горизонталей совпадает. В данных точках предлагается использовать сглаживающий фильтр, например, гауссовский.

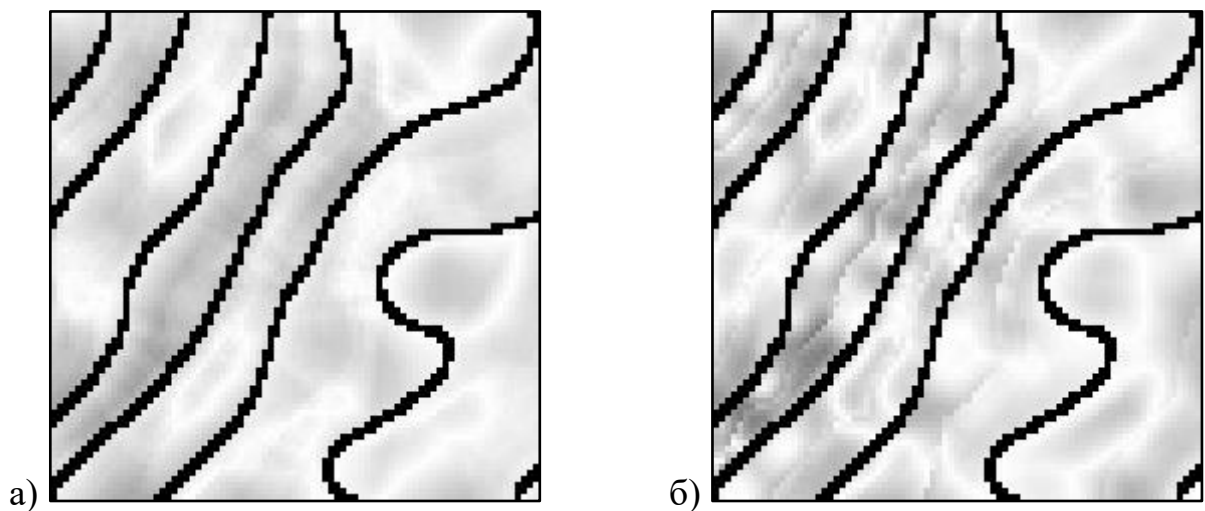


Рисунок 4.2 – Пример работы интерполятора:  
а) линейного; б) предложенного

Реализовать алгоритм для точек первого типа можно с помощью обработки подпространств соседства, ограниченных несколькими горизонталями: такие подпространства всегда лежат между горизонталями двух различных значений высот. Это свойство является отличительным для топографических карт в плане интерполяции. С помощью морфологических операций можно найдены границы этих



подпространств и найти соответствующие значения высоты этих горизонталей. Необходимо случайным образом выбрать точки на этих горизонталях, на основании которых мы будем производить интерполяцию. При выборе слишком большой доли числа точек  $P$  интерполяция будет производиться медленнее, но точнее, а при выборе меньшего числа – быстрее, но менее точно.

Для точек второго типа важно наличие отметки высоты в точке локального экстремума. Остальные точки можно интерполировать, зная крутизну склона к нормалям в точках ограничивающей горизонтали, вычисленные после обработки внешних точек (они принадлежат к точкам первого типа) и факт, что в точке экстремума крутизна равна нулю. При интерполяции данного типа считаем, что крутизна склона изменятся равномерно:

$$\alpha(r_i) = \alpha_0 - \mu * r_i, \quad (4.7)$$

где  $r_i$  – расстояние от точки на горизонтали в сторону точки экстремума,  $\alpha_0$  – значение крутизны склона в точке на горизонтали,  $\mu$  – линейный коэффициент.

Тогда с учётом того, что угол крутизны можно считать малой величиной, относительную высоту можно выразить по формуле:

$$\Delta h = \int_0^{r_{i,c}} \tan \alpha(r_i) dr_i \approx \int_0^{r_{i,c}} \alpha(r_i) * dr_i = \left[ \alpha_0 * x - \frac{\mu * x^2}{2} \right]_0^{r_{i,c}} = \alpha_0 * r_{i,c} - \frac{\mu * r_{i,c}^2}{2}, \quad (4.8)$$

где  $r_{i,c}$  – расстояние от точки на горизонтали до точки экстремума. Из данной формулы можно вывести значение линейного коэффициента.

Фактически данная интерполяция является квадратичной, так как возвышенность или низина приближается полиномом второй степени по направлению от каждой точки горизонтали к точке экстремума. Так как площади таких участков невелики, что получаемую точность считаем достаточной, но её нужно привести в

растровый формат, так как при данном алгоритме сетка получается не прямоугольная, а полярная.

Если отметка высоты отсутствует, то точку локального экстремума можно определить в центроиде ограничивающей горизонтали. Относительную высоту экстремума оцениваем по формуле:

$$\Delta h = \overline{r_{l,c}} * \tan \frac{\bar{\alpha}}{2}, \quad (4.9)$$

где  $\overline{r_{l,c}}$  – среднее расстояние от центроида до точек ограничивающей горизонтали,  $\bar{\alpha}$  – средний угол крутизны склона к нормали в точках горизонтали.

Точки третьего типа можно интерполировать, если рядом находятся отметки высот (алгоритм аналогичен интерполяции по второму типу), иначе возникает задача экстраполяции, для которой ожидаемая точность может сильно упасть к краю карты. Некоторые из этих точек можно отнести к первому типу, если объединить несколько соседних топографических карт: тогда в этом типе останутся только точки, лежащие у границ объединённой карты. Либо можно использовать данные о вычисленных склонах, как в точках второго типа.

В течение экспериментальных исследований будут изменяться параметры функции весового коэффициента и количество ближайших соседей для интерполяции. Полностью алгоритм интерполяции изображён на рис. А.5.

Оценку алгоритма интерполяции можно осуществить автоматически, если у нас есть данные о высоте рельефа. Основным материалом для оценки точности является разность значений высот по ЦМР и в реальности. Считаем, что эта разность состоит из систематической и случайной ошибки:  $\Delta Z = Z_{\text{Интерп}} - Z_{\text{ЦМР}} = \Delta_Z + \delta_Z$ , где  $Z_{\text{Интерп}}$  – вычисленное значение высоты в точке,  $Z_{\text{ЦМР}}$  – реальное значение,  $\Delta_Z$  – значение систематической ошибки,  $\delta_Z$  – значение случайной ошибки.

Оценку систематической ошибки производим по формуле:  $\bar{\Delta Z} = \frac{1}{S} \sum_S \Delta Z$ , где  $S$  – площадь карты в пикселях. После исключения систематической ошибки  $\Delta z = \Delta Z - \bar{\Delta Z}$  можно найти максимальное отклонение  $\Delta z_{max}$  и среднеквадратическое отклонение по формуле  $\sigma_z = \sqrt{\frac{1}{S} \sum_S \Delta^2 z}$ .

Стоит учитывать, что топографические карты не строго соответствуют местности. Часто для акцентирования рельефности горизонтали могут перемещать, из-за чего могут возникать неисправимые в данной работе ошибки в 1/3 от высоты сечения, связанные с неточностью отображения самой карты. Для карт масштаба 1 : 50 000 стандартным значением высоты сечения является 10 м, что говорит о возникновении возможной погрешности порядка 3 м.

Точность полученной ЦМР также зависит от характера рельефа: в холмистых и горных регионах она может быть в несколько раз меньше, чем на равнинных участках. С другой стороны, на рельефной местности возможно более частое отображения горизонталей и отметок высот. В данной работе высота уреза воды принимается равной одной высоте, но фактически топосъемка проводится в разное время года, что приводит к возникновению ошибок определения высоты, связанных с изменением полноводья водоёмов.

#### **4.4 Программный комплекс преобразования изображений топографических карт в цифровую модель рельефа**

Для прикладной реализации разработанных алгоритмов в программной среде *Matlab* был создан программный комплекс [43, 43] для преобразования сканированных изображений топографических карт в цифровую модель рельефа (ПК ПТК). ПК ПТК может быть запущен на ЭВМ, на котором установлены следующие библиотеки программной среды *Matlab* (при создании использовалась версия *R2022a*): *Deep Learning Toolbox* (14.4), *Image Processing Toolbox* (11.5), *Statistics and Machine Learning Toolbox* (12.3), *Computer Vision Toolbox* (10.2), и *Matlab Runtime*.

Автономное приложение также содержит необходимые библиотеки в составе и требует только свободно распространяемого *Matlab Runtime* для запуска инсталлятора.

Использование программной среды *Matlab* объясняется широкими возможностями в обработке изображений, алгоритмы и методы которых достаточно оптимизированы, так как используют современные научные разработки и опираются на векторную парадигму программирования, и широкими возможностями к визуализации, что позволяет достаточно быстро производить отладку кода.

Минимальные системные требования: 2 ГБ места на жёстком диске, 8 ГБ ОЗУ, графический процессор 2 ГБ (возможно дискретный).

Программный комплекс состоит из семи программных блоков (ПБ), реализующих следующие алгоритмы.

1. ПБ сегментации. Необходимыми данными для работы этого ПБ являются файл с метаданными (имена файлов, соответствие цветов размеченных пикселей классам), набор исходных изображений, набор соответствующих им масочных изображений (для оценки качества сегментации), файлы с интеллектуальными моделями для обработки. На выходе ПБ сегментации формируются полутонные многослойные изображения (по одному на каждое исходное), каждый слой которого отвечает за меру соответствия определённого класса.

2. ПБ локализации. Необходимыми данными для работы этого ПБ являются изображения, полученные на предыдущем этапе, масочные изображения (для оценки качества локализации), файлы с содержащимся списком морфологических операций. На выходе ПБ сегментации формируются бинарные многослойные изображения, определяющие локализацию необходимых объектов.

3. ПБ распознавания ТО. Необходимыми данными для работы этого ПБ являются изображения, полученные на этапе 2, масочные изображения (для оценки качества распознавания ТО), файлы с интеллектуальными моделями. На выходе ПБ распознавания формируются файлы, содержащие базу данных о найденных ТО (отметки высот и подписи горизонталей) и их характеристиках.

4. ПБ распознавания ЛО. Необходимыми данными для работы этого ПБ являются изображения, полученные на этапе 2, файлы с последовательностью выполняемых фильтрующих и соединяющих сегменты операций. На выходе ПБ распознавания формируются бинарные файлы, содержащие скелетизированные изображения ЛО и данные о соединении.

5. ПБ распознавания ПЛО. Необходимыми данными для работы этого ПБ являются изображения, полученные на этапе 2, файлы с интеллектуальными моделями и набором операций постобработки. На выходе ПБ распознавания формируются категориальные файлы.

6. ПБ комплексного анализа. Необходимыми данными для работы этого ПБ являются изображения, полученные на этапах 3 и 4. На выходе ПБ комплексного анализа формируется проинициализированная карта высот, на которой пикселям горизонталей и отдельных точек поставлено в соответствие значение высоты.

7. ПБ интерполяции. Необходимыми данными для работы этого ПБ являются изображения, полученные на этапе 6. На выходе ПБ интерполяции формируется цифровая модель рельефа, выполненная в виде двумерной матрицы высот.

Общий алгоритм работы с использованием разработанных функций показан для дата сайентиста (таблица А.13) и конечного пользователя (таблица А.14). Основная цель дата сайентиста: подготовить данные для обучения ИМ, обучить ИМ и настроить параметры алгоритмов обработки и проверить качество и быстродействие алгоритмов. Основная цель конечного пользователя: использовать подготовленные дата сайентистом ИМ и параметры алгоритмов для обработки конкретных ЦТК. Скриншот работы ПО представлен на рис. А.6, свидетельства о государственной регистрации программ для ЭВМ – в приложении Б. Код программных блоков, использованных моделей и пример преобразования ЦТК в ЦМР для одной карты можно посмотреть на хостинге *GitHub* [42], доступ предоставляется по запросу.

#### 4.5 Проверка разработанных алгоритмов для ЦТК

Проверка разработанных алгоритмов и программного обеспечения осуществляется в два этапа. На первом этапе использовалось 20 цветных отсканированных топографических карт (таблица 4.2) для алгоритмов сегментации, классификации, распознавания и группировки. Участки выбранных ТК являются свободно распространяемыми согласно перечню открытых данных [77]. ТК были приведены к 300 *dpi*, карты масштаба 1:25 000 также были шкалированы, чтобы согласовать размеры отметок высот; в таблице показаны итоговые размеры карт. Так как ТК разного года издания были взяты из разных источников, то они отличаются по цветовой гамме, поэтому разбивка карт на тренировочные и тестовые была произведена соответственно; тренировочные материалы отмечены в таблице галочкой.

Таблица 4.2 – Список используемых участков ТК и их характеристики

№	Т	Обозначение ТК	Масштаб	Год ТК	Исходный <i>dpi</i>	Ширина	Высота
1	✓	О-40-083-А	1:50 000	1982	200	3182	4194
2	✓	О-40-083-Б	1:50 000	1982	200	3251	2357
3		О-40-084-А	1:50 000	1982	200	3158	4128
4		О-40-084-В	1:50 000	1982	200	3113	4011
5	✓	О-40-095-Г	1:50 000	1982	200	3366	4167
6	✓	О-39-016-А	1:50 000	1978	300	3232	4308
7	✓	О-39-025-Б	1:50 000	1965	300	3332	3776
8	✓	О-39-028-А	1:50 000	1975	300	3320	4268
9		О-39-029-Б	1:50 000	1965	300	3250	4287
10		О-39-038-А	1:50 000	1988	300	3192	4240
11	✓	О-39-061-В	1:50 000	1965	300	3338	4285
12		О-43-042-В	1:50 000	1967	300	3291	4308
13	✓	О-43-043-Б	1:50 000	1967	300	3286	4300
14	✓	О-40-023-Б-б	1:25 000	1982	250	2453	3363
15		О-40-023-В-в	1:25 000	1982	250	2463	3332
16	✓	О-40-023-В-г	1:25 000	1982	250	2314	3307
17	✓	О-40-024-А-г	1:25 000	1982	250	2517	3378
18		О-40-024-Б-а	1:25 000	1982	250	2538	3408
19		О-40-024-В-б	1:25 000	1982	250	2426	3360
20	✓	О-40-024-В-в	1:25 000	1982	250	2513	3382

Участки представленных ТК представлены в таблице А.10. Для разработанных алгоритмов критично важно, чтобы они работали на разных типах ТК, так как они отличаются не только годом создания, но и имеют вариации следования явным и неявным правилам отображения, либо имеют различные отклонения от них. Примеры нарушений правил представлены в таблице А.11.

Так как отсканированные карты являются большими изображениями, то на них присутствует достаточное количество элементов низкого уровня (пикселей, сегментов, отдельных объектов). Результаты работы текущего проверяемого алгоритма при этом передаются на вход следующего по порядку алгоритма с вычислением критериев качества.

Для разработанного алгоритма многометочной сегментации сравниваются следующие интеллектуальные модели: ИНС прямого распространения (как распознающие все классы, так и отдельные), метод  $k$  ближайших соседей, метод опорных векторов, метод решающих деревьев; ручная разметка используется в качестве сравнения. Алгоритмы сравниваются с помощью  $F$ -меры определения типа пикселей и времени выполнения обработки. Предлагаемый алгоритм двухкаскадного распознавания цифр исследуется с ИМ аналогичными для сегментации.

Для определения оптимальной последовательности морфологических операций для локализации объектов будет запущен генетический алгоритм, результаты которого будут отображены аналогично параграфу 3.3. При выполнении морфологических операций форма сегментов может измениться, но сегменты, соответствующие классу чисел, должны проходить через центры цифр. Дальнейшее распознавание будет происходить с учётом того, что найденные пиксели будут являться серединой цифры, поэтому площадь сегментов должна быть как можно меньшей.

В результате экспериментальных исследования будут получены параметры итераций группировки сегментов горизонталей и геометрические параметры объединения цифр отметок высот в числа. Ошибка в определении числа прямо зависит от ошибки в определении цифры и количества цифр в числе. Так как напрямую нельзя сделать масочные изображения для оценки качества группировки

горизонталей (при их скелетизации они могут сместиться на пару пикселей), то оценка качества группировки сегментов горизонталей проводится путём подсчёта оставшихся концевых точек и с помощью непосредственного визуального анализа.

Так как в результате работы первого этапа появляются сгруппированные объекты высокого уровня, то их количество оказывается недостаточным для проверки алгоритмов комплексного анализа. Входная информация для алгоритма комплексного анализа является достаточно абстрактной, поэтому легче её смоделировать, чем размечать существующие карты без влияния на итоговое качество, а алгоритм интерполяции невозможно проверить напрямую через изображения топографических карт. Поэтому на втором этапе используются сгенерированные модели местности, имитирующие реальный рельеф. Алгоритм комплексного анализа запускается и для тестовых изображений ЦТК, но проверка происходит в ручном режиме.

Для получения моделей с псевдопланетарным рельефом используется алгоритм *diamond-square*, аналогичный параграфу 3.1. Полученная матрица высот нормализовалась к отрезку высот  $[0, 1]$  и происходила медианная фильтрация для сглаживания неровностей. Такое сглаживание необходимо, чтобы горизонталы на получаемой в дальнейшем топографической карте становились более ровными; похожая операция происходит при построении топографических карт для обеспечения большей наглядности и передачи рельефа в угоду точности.

Для проверки алгоритма комплексного анализа в значения некоторых горизонталей и локальных экстремумов вносятся искажения либо эти значения убираются, так как такие ошибки возможны в результате первого этапа. Для проверки предлагаемого алгоритма интерполяции и его сравнения с существующими алгоритмами карта горизонталей и отметки высот остаются без изменений.

Предлагаемый алгоритм интерполяции сравнивается с алгоритмами линейной интерполяции, интерполяции кубическими сплайнами и интерполяцией Сибсона. В качестве критериев оценивания алгоритмов будем использовать максимальное отклонение, среднеквадратическое отклонение (в процентах относительно шага горизонталей) и время обработки.



В течение времени запуска ПБ анализируется время работы ЭВМ и дополнительное время работы оператора, необходимого на исправление ошибок. Экспериментальные исследования направлены как на ускорение работы, так и на сокращение количества ошибок.

#### 4.5.1 Исследование алгоритма многометочной сегментации

Проведём экспериментальное исследование для выявления оптимальной ИМ, максимизирующей  $F$ -меру при сегментировании сканированных топографических карт. Для представленных ТК была проведена классификация отдельных точек, всего для обучения использовались ~240 тыс. точек, 20% из которых использовались для валидации и 20% для тестирования. Так как сегментация является низкоуровневой операцией, то получившиеся ИМ не запускались на тестовых изображениях.

Классификация происходила по 6 классам: подписи высот (I, краска чёрная 2558-01), горизонталы и их отметки (II, коричневая 2558-62), площади низкорослой растительности (III, зеленая расслабленная 2558-45), площади лесов (IV, зеленая 2558-45), объекты гидрографии (V, синяя 2558-38), площади прочей специфики, кроме квартальной застройки (VI, отсутствие краски).

На вход ИМ подавались значения цветовых координат, полученных из окрестностей скользящего окна с центром в определяемой точке, всего рассматривалось 4 варианта размеров окон  $a$ : один, три, пять и семь пикселей.

Выделим параметры ИНС, которые не будем менять в процессе эксперимента. Нейроны скрытого слоя имеют функцию активации сигмоида, на выходном слое нейроны имеют функцию *softmax* (является обобщением логистической функции для количества классов больше двух). Обучение проводилось с использованием алгоритма сопряжённых градиентов.

Рассмотрим два случая реализации ИНС: одна нейросеть с шестью выходными нейронами и две нейросети с одним выходным нейроном, так как для рельефа

нужны только пиксели, относимые к отметкам высот и горизонталям. Обучение запускалось для количества нейронов на скрытом слое  $H$ , где  $H = \alpha * (I + O)$ ,  $I$  – количество нейронов на входном слое ( $I = 3 * a^2$ ),  $O$  – количество нейронов на выходном слое (6 для первого случая, 2 для второго),  $\alpha$  – коэффициент.

Для исследования стабильности обучения ИНС используем подход, разработанный в исследовании [41]. Так как обучение ИНС не является постоянным из-за использующихся стохастических алгоритмов, то после каждого обучения они могут показывать различные результаты. Поэтому кроме средних значений показателей точности и быстродействия используем также и среднеквадратичное отклонение: последний показатель будем считать показателем стабильности метода обучения для данной архитектуры ИНС.

Обучение ИНС запускалось 200 раз для  $a = 1$ , 150 раз для  $a = 3$ , 100 раз для  $a = 5$  и 50 раз для  $a = 7$ . В таблицах 4.3 и 4.4 представлены показатели обучения (время обучения  $T_{об}$ ) и показатели тестирования ( $F$ -мера и время работы  $T$  на 1 Мп). Для времени работы представлено только среднее значение, так как отклонение в данном случае возникает из-за непостоянности загрузки ЦП и не является следствием используемой модели.

Из таблиц видно, что использование двух независимых ИНС даёт незначительно лучшее качество с точки зрения  $F$ -меры, но при этом суммарное время обработки увеличивается в 1,5..1,8 раз для  $a > 1$ , так как при этом нужно запускать работу двух ИНС, а не одной.  $F$ -мера для  $a = 1$  является неприемлемо низкой относительно других размеров окон, а для  $a = 7$  отличается от  $a = 5$  на 0,2..0,4%, поэтому выбирать можно среди ИНС с  $a = 3$  и  $a = 5$ . Оптимальным с точки зрения стабильности является ИНС с  $a = 5$  и  $\alpha = 0,5$ , так как при этом отношение стандартного отклонения к среднему значению является минимальным для времени обучения (табл. 4.3) и для  $F$ -меры (рис. 4.3).

Таблица 4.3 – Показатели обучения и качества для ИНС с шестью выходами

a, пикс	$\alpha$	F-мера, %						$T_{об}$ , с	T, с/Мп
		I	II	III	IV	V	VI		
1	1/2	93,7 ± 0,3	88,8 ± 0,6	69,4 ± 1,3	87,2 ± 0,3	92,7 ± 0,5	94,6 ± 0,5	102 ± 37	0,7
	1	94,1 ± 0,2	89,5 ± 0,3	71,4 ± 0,8	87,8 ± 0,2	93,3 ± 0,3	95,1 ± 0,3	130 ± 44	1,1
	2	94,3 ± 0,1	89,7 ± 0,2	72,2 ± 0,5	88,1 ± 0,2	93,5 ± 0,2	95,2 ± 0,1	168 ± 55	1,2
3	1/2	97,3 ± 0,2	94,2 ± 0,3	85,8 ± 1,4	92,6 ± 0,6	95,5 ± 0,3	97,9 ± 0,1	410 ± 120	2,2
	1	97,7 ± 0,2	94,8 ± 0,3	87,3 ± 1,7	93,4 ± 0,8	96,0 ± 0,3	98,0 ± 0,1	590 ± 200	3,2
	2	98,0 ± 0,2	95,1 ± 0,4	87,6 ± 2,0	93,6 ± 0,9	96,3 ± 0,3	98,0 ± 0,1	790 ± 280	3,9
5	1/2	98,1 ± 0,2	95,9 ± 0,2	89,9 ± 1,4	94,5 ± 0,7	96,7 ± 0,2	98,5 ± 0,1	850 ± 230	5,3
	1	98,3 ± 0,2	96,3 ± 0,2	90,0 ± 1,8	94,6 ± 0,9	96,9 ± 0,3	98,5 ± 0,1	1280 ± 420	6,6
	2	98,4 ± 0,2	96,4 ± 0,2	89,5 ± 1,6	94,5 ± 0,8	97,0 ± 0,3	98,5 ± 0,1	2010 ± 590	9,0
7	1/2	98,6 ± 0,1	96,7 ± 0,2	92,1 ± 1,0	95,6 ± 0,5	97,3 ± 0,2	98,8 ± 0,1	2110 ± 370	11,2
	1	98,7 ± 0,2	96,9 ± 0,2	91,0 ± 1,6	95,1 ± 0,8	97,4 ± 0,3	98,7 ± 0,1	3140 ± 880	13,1
	2	98,6 ± 0,2	96,8 ± 0,2	90,1 ± 1,7	94,8 ± 0,8	97,3 ± 0,3	98,6 ± 0,2	5150 ± 1820	19,6

Таблица 4.4 – Показатели обучения и качества для двух ИНС с одним выходом

a, пикс	$\alpha$	F-мера I, %	$T_{об}$ I, с	F-мера II, %	$T_{об}$ II, с	T, с/Мп
1	1/2	94,0 ± 0,5	17 ± 6,5	89,3 ± 0,3	16 ± 5,5	0,7
	1	94,1 ± 0,1	21 ± 6,9	89,4 ± 0,2	22 ± 8,0	0,8
	2	94,2 ± 0,1	29 ± 9,7	89,5 ± 0,1	28 ± 10	1,1
3	1/2	98,1 ± 0,2	62 ± 15	95,2 ± 0,2	87 ± 18	3,3
	1	98,3 ± 0,1	91 ± 16	95,5 ± 0,3	130 ± 24	4,8
	2	98,4 ± 0,1	130 ± 23	95,6 ± 0,2	170 ± 33	5,8
5	1/2	98,6 ± 0,1	120 ± 15	96,7 ± 0,1	170 ± 28	8,6
	1	98,7 ± 0,1	200 ± 24	96,8 ± 0,2	250 ± 41	10,9
	2	98,8 ± 0,1	380 ± 45	96,9 ± 0,1	450 ± 62	15,1
7	1/2	98,8 ± 0,1	260 ± 25	97,1 ± 0,2	340 ± 43	19,2
	1	99,0 ± 0,1	520 ± 45	97,2 ± 0,1	610 ± 72	22,6
	2	99,0 ± 0,1	1080 ± 110	97,2 ± 0,1	1090 ± 140	34,4

Сравним вычисленные показатели с другими алгоритмами, используемыми для сегментации и которые являются стабильными при сохранении тренировочной выборки. Для алгоритма  $k$ -NN и SVM время обучения и запуска больше зависит от размера скользящего окна (сложность классификации для  $k$ -NN равна  $O(n * a^2)$ ) и количества обучающих данных (сложность обучения для SVM равна  $O(n^3)$ ), поэтому будем рассматривать только меньшие размеры окон и использовать в качестве тренировочных материалов 5% от исходного датасета. Алгоритм обучения CART для DT является стабильным при данных условиях; для DT исследуем

максимальное число ветвлений  $MNS$ . Результаты методов для пикселей первого и второго классов представлены в таблице 4.5.

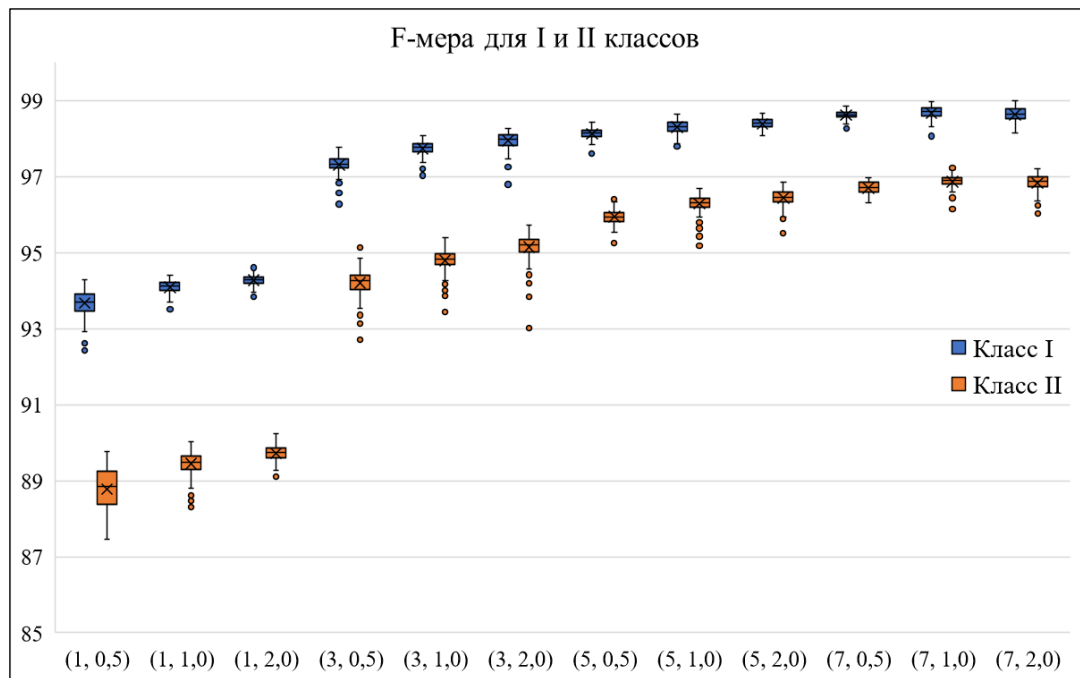


Рисунок 4.3 – Диаграмма «ящик с усами» для значений  $F$ -меры I и II классов пикселей в зависимости от параметров ИНС

Таблица 4.5 – Значения  $F_\beta$ -меры типов сегментов и время обработки в зависимости от метода обработки

Метод обработки	Параметры алгоритмов	$F$ -мера I, %	$F$ -мера II, %	$T_{об}$ , с	$T$ , с/Мп
$k$ -NN	$a = 1, k = 1$	93,4	87,8	0,1	2,0
	$a = 1, k = 3$	94,2	89,6	0,1	2,8
	$a = 3, k = 1$	95,4	92,0	0,1	36,7
SVM, линейное ядро	$a = 1$	89,2	82,8	2,0	2,5
	$a = 3$	90,3	87,0	3,4	6,1
	$a = 5$	91,3	87,8	7,9	13,0
SVM, RBF ядро	$a = 1$	91,0	89,9	1,9	11,2
DT, $MNS = 100$	$a = 1$	93,6	89,2	0,4	0,8
	$a = 3$	95,4	89,9	1,1	1,4
	$a = 5$	95,5	89,9	3,6	2,7
DT, $MNS = 1000$	$a = 1$	94,7	90,9	0,4	0,8
	$a = 3$	96,8	93,1	1,8	1,5
	$a = 5$	96,9	93,3	5,8	2,5

На рисунке 4.4 результаты экспериментальных исследований (с учётом предобработки) представлены в виде графиков: по горизонтальной оси отложено среднее значение  $F_\beta$ -мер I и II типов, по вертикали – время обработки на мегапиксель.

Видно, что на среднем качестве  $F$ -меры, большем 95%, находятся только нейросетевые модели с  $\alpha > 1$ ; для меньшего значения качества лучшие показатели показывает метод классифицирующих деревьев. При этом на данном этапе максимизация качества является приоритетной характеристикой сегментации, так как от этого зависят все последующие этапы обработки, поэтому можно считать, что алгоритм единой нейросети с  $\alpha = 5$  и  $\alpha = 0,5$  является оптимальным для решения задачи сегментации на изображениях топографических карт, обеспечивая достаточное качество при сохранении небольшого значения времени работы.

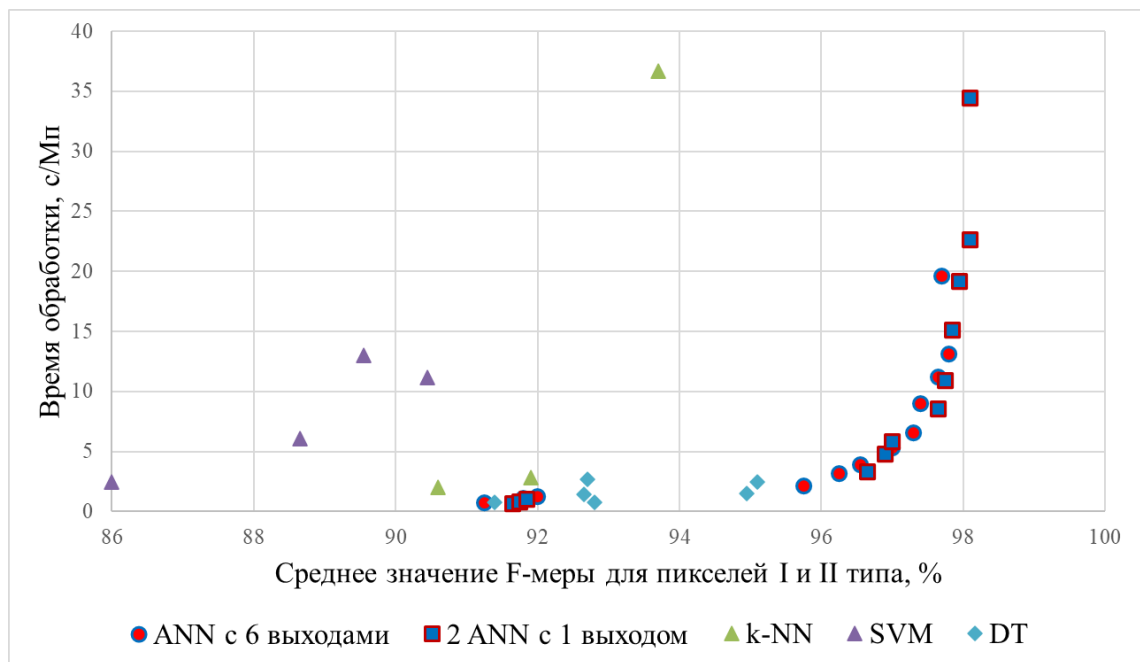


Рисунок 4.4 – Зависимости времени выполнения алгоритмов от среднего значения  $F$ -меры пикселей I и II типов

#### 4.5.2 Исследование алгоритма локализации сегментов

Классификация сегментов I типа подразумевает отделение сегментов, относящихся к классу «отметки высот». Отметки высот располагаются всегда строго

горизонтально и отображаются с большим шрифтом, чем подписи горизонталей. На топографических картах присутствует значительное количество других условных обозначений чёрного цвета, тем не менее, из элементов, похожих на отметки высот можно выделить только подписи топонимов соответствующих отметок, например, наименования гор, или названия поселений.

В качестве критичных изображений взяты изображения, содержащие центры цифр, в качестве идеальных для локализации – критичные изображения, над которыми проведены МО дилатации с СЭ квадрат  $9 \times 9$  и закрытие с СЭ горизонтальная линия длиной 15.

Отличие от проведения ГА в разделе 3.2 заключается в том, что в данной ситуации он запускается для нескольких изображений, поэтому формула 2.5 должна быть модифицирована. Можно предложить два способа модификации: мягкий и строгий. Для мягкого способа  $FF$  считается как среднее значение  $FF$  для всех изображений, при этом  $RecalGA$  для каких-то изображений может оказаться равным нулю:

$$FF = \frac{1}{|P|} * \sum_{P' \in P} \left( RecallGA(P') * \frac{|P' \wedge P_{crit}|}{|P_{crit}|} * \frac{|P' \wedge P_{ideal}|}{|P' \vee P_{ideal}|} \right), \quad (4.10)$$

где  $P$  – множество изображений, на которых происходит обучение.

Строгий способ предполагает зануление  $FF$ , если хотя бы для одного изображения  $RecalGA$  станет равным нулю:

$$FF = \frac{1}{|P|} * \sum_{P' \in P} \left( \frac{|P' \wedge P_{crit}|}{|P_{crit}|} * \frac{|P' \wedge P_{ideal}|}{|P' \vee P_{ideal}|} \right) * \prod_{P' \in P} RecallGA(P'). \quad (4.11)$$

Последовательность выполнения МО для ГА со строгим и мягким вариантом расчёта для пикселей типа I и II представлены в Приложении А. Графики

изменения функции приспособленности и размера хромосомы от номера итерации ГА для обеих модификаций показаны на рис 4.5.

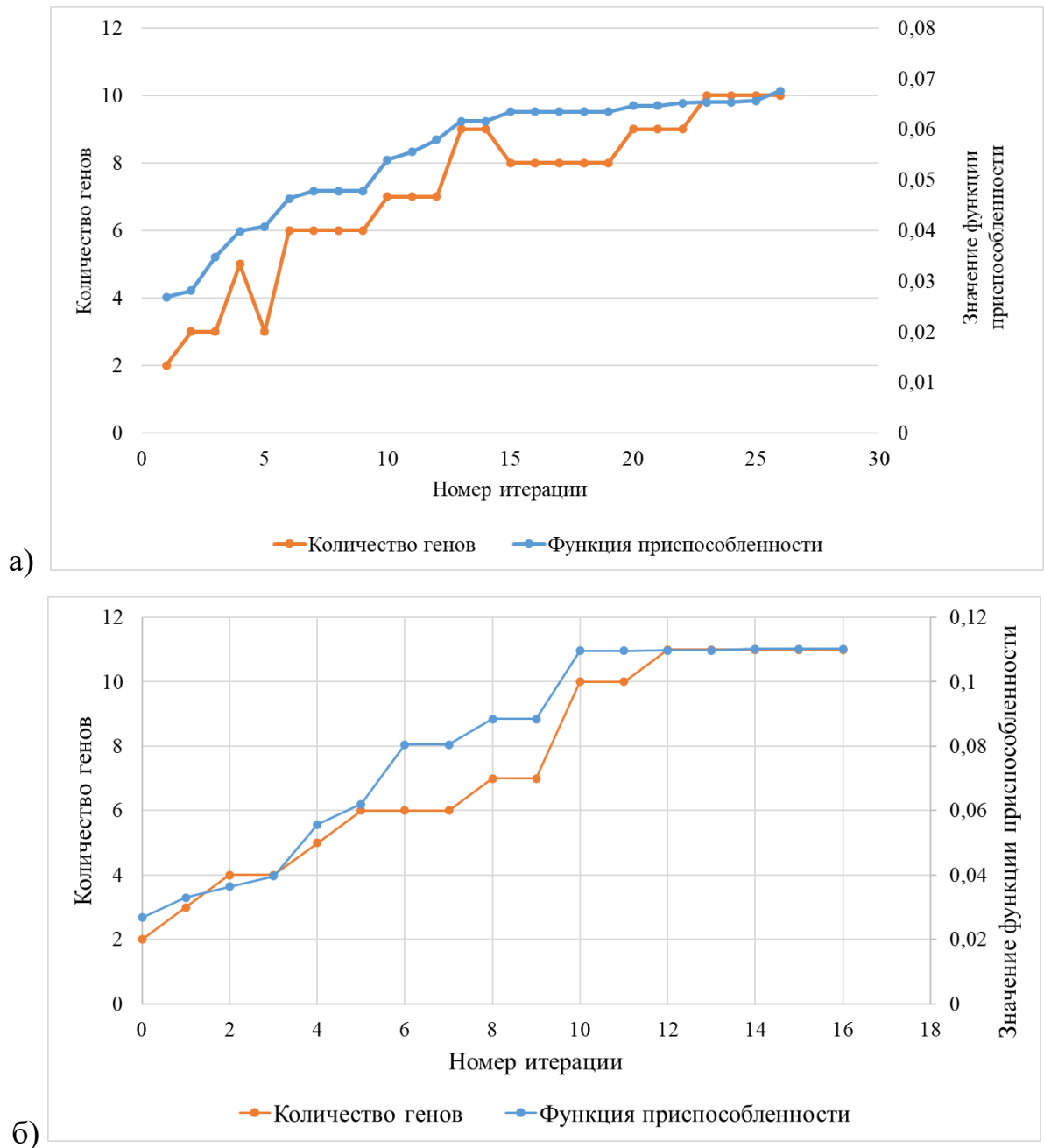


Рисунок 4.5 – Графики изменения функции приспособленности и количества генов в хромосоме в зависимости от номера итерации ГА:  
а) строгая модификация расчёта FF; б) мягкая модификация

Сравнение полученных наборов морфологических операций производилось с базовыми решениями в виде простой бинаризации с порогом 0,5 и с МО закрытия

с СЭ квадрат  $5 \times 5$ . В таблице 4.6 отображены результаты для тестовых изображений, всего на них изображено 1837 цифр.

По результатам можно сделать вывод, что мягкий вариант расчёта функции приспособленности приводит к меньшему количеству локализованных пикселей, но к большему количеству пропущенных цифр относительно строгого варианта. Так как пропуск цифры является более критичным для распознавания (отметка может состоять из 3-4 цифр, поэтому потеря одной цифры приводит к потере всей отметки), то выбираем в качестве основного метода строгий способ расчёта *FF*. 0,1% пропущенных цифр соответствует потере трёх цифр на всех восьми картах, что является небольшой потерей относительно МО закрытия, так как количество локализованных пикселей при разработанном алгоритме уменьшается в 2,6 раз.

Таблица 4.6 – Критерии качества при проведении морфологических операций локализации сегментов I типа на тестовых изображениях

Алгоритм	Кол-во локализованных пикселей, Мп	Кол-во пропущенных цифр, %	Время работы, с/Мп
Изображения полностью	92,2	0	–
Бинаризация с порогом 0,5	5,2	7,4	0,002
Закрытие, СЭ квадрат $5 \times 5$	6,3	0,1	0,02
Мягкий вариант расчёта <i>FF</i>	1,7	0,7	0,18
Строгий вариант расчёта <i>FF</i>	2,4	0,1	0,14
Размеченные изображения	0,2	0	–

Локализация сегментов II типа подразумевает разделение коричневых сегментов на классы «подписи горизонталей», «линии горизонталей» и «бергштрихи». Подписи могут быть расположены на карте под любым углом (ориентирование по скату рельефа) и быть достаточно близки как к собственной горизонтали, так и к соседним. Также высота шрифта не сильно отличается от ширины горизонталей, особенно утолщённых, поэтому имеет смысл использовать МО с как можно



меньшими СЭ, порядка нескольких пикселей. Это может плохо сказаться на точности, например, к отметкам могут быть присоединены участки соседних горизонталей.

Тем не менее, если удастся выполнить локализацию подписей горизонталей, то сами горизонталы и бергштрихи можно получить с помощью простейших МО для очистки, выравнивания, скелетизации и удаление точек ветвлений у линий. Данное удаление необходимо, так как горизонталы, как линии одинакового уровня, должны восприниматься алгоритмом как кривые с двумя конечными точками. В качестве критичных изображений взяты изображения, содержащие центры цифр, в качестве идеальных для локализации – критичные изображения, над которыми проведена МО дилатации с СЭ квадрат  $11 \times 11$ .

Сравнение полученных наборов морфологических операций производилось с базовыми решениями в виде простой бинаризации с порогом 0,5, с МО закрытия с СЭ квадрат  $5 \times 5$ . В таблице 4.7 отображены результаты для тестовых изображений, всего на них изображено 300 цифр.

Графики изменения функции приспособленности и размера хромосомы от номера итерации ГА для обеих модификаций показаны на рис 4.6.

Таблица 4.7 – Критерии качества при проведении морфологических операций локализации сегментов II типа на тестовых изображениях

Алгоритм	Кол-во локализованных пикселей, Мп	Кол-во пропущенных цифр, %	Время работы, с/Мп
Изображения полностью	92,2	0	–
Бинаризация с порогом 0,5	8,2	13,7	~0,001
Закрытие с СЭ квадрат $5 \times 5$	9,2	0	0,02
Мягкий вариант расчёта <i>FF</i>	0,2	4,0	0,12
Строгий вариант расчёта <i>FF</i>	1,5	0	0,11
Размеченные изображения	0,03	0	–

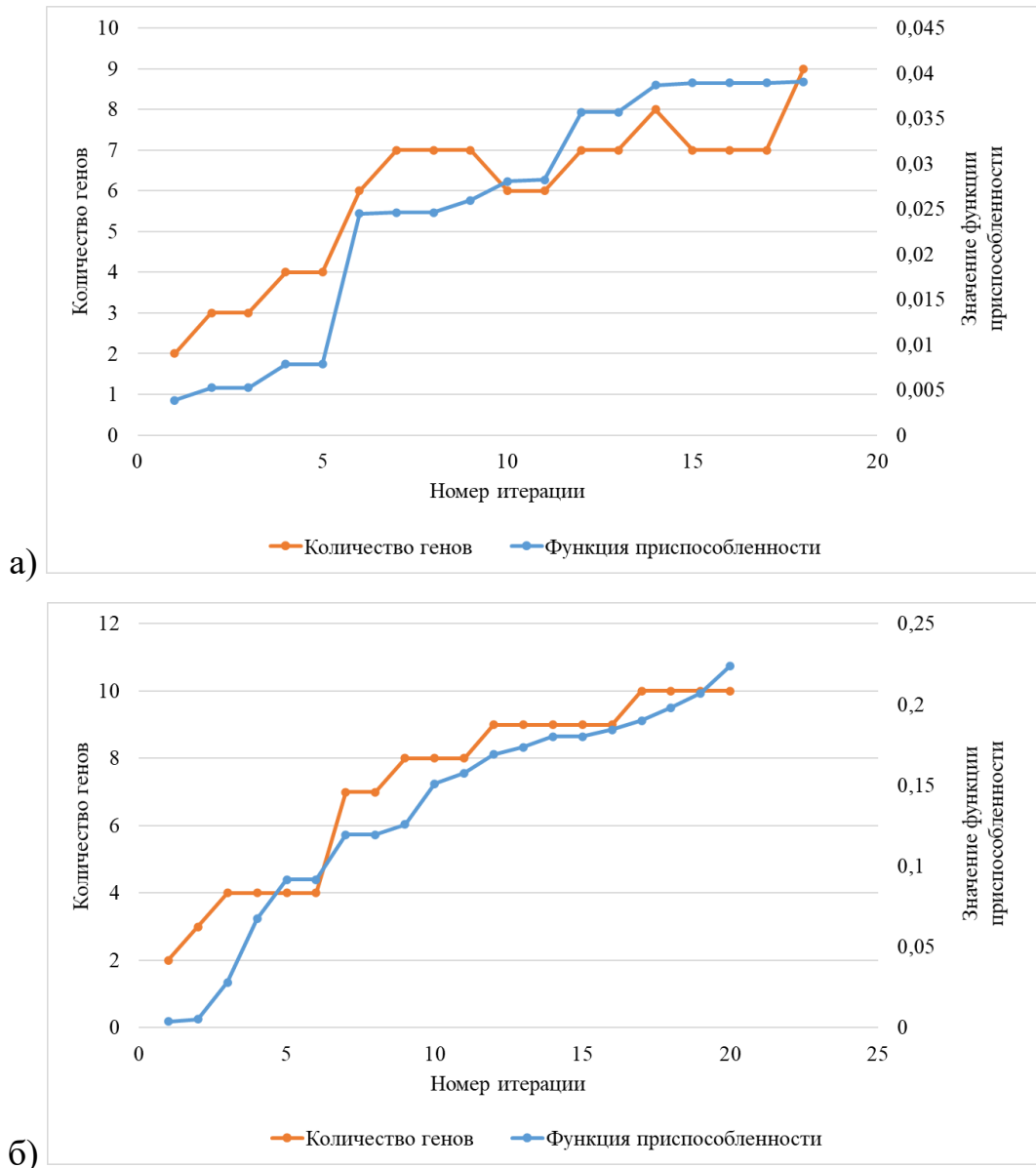


Рисунок 4.6 – Графики изменения функции приспособленности и количества генов в хромосоме в зависимости от номера итерации ГА:

а) строгая модификация расчёта  $FF$ ; б) мягкая модификация

Аналогично эксперименту с чёрными цифрами, выбираем хромосому, полученную при строгом расчёте  $FF$ , так как количество пропущенных цифр при этом отсутствует, а количество локализованных пикселей меньше относительно МО закрытия в 6,1 раз.

Морфологические операции для локализации бергштрихов выполнялись в среднем 0,3 с/Мп, по ним для 80% горизонталей с бергштрихами можно было определить скат. Остальные горизонталы были размечены вручную за 25..50 с/Мп.

### 4.5.3 Исследование алгоритмов распознавания ТО

Методика экспериментального исследования интеллектуальных методов для использования в разработанном алгоритме двухкаскадного распознавания аналогична исследованию методов для сегментации. Сначала рассмотрим распознавание отметок высот.

Для первого каскада объекты представляют собой окна размером  $11 \times 19$  пикселей, полученных с этапа сегментации соответствующего типа. Всего была размечена 4661 цифра, обучение производилось на 2824 цифрах тренировочных изображений.

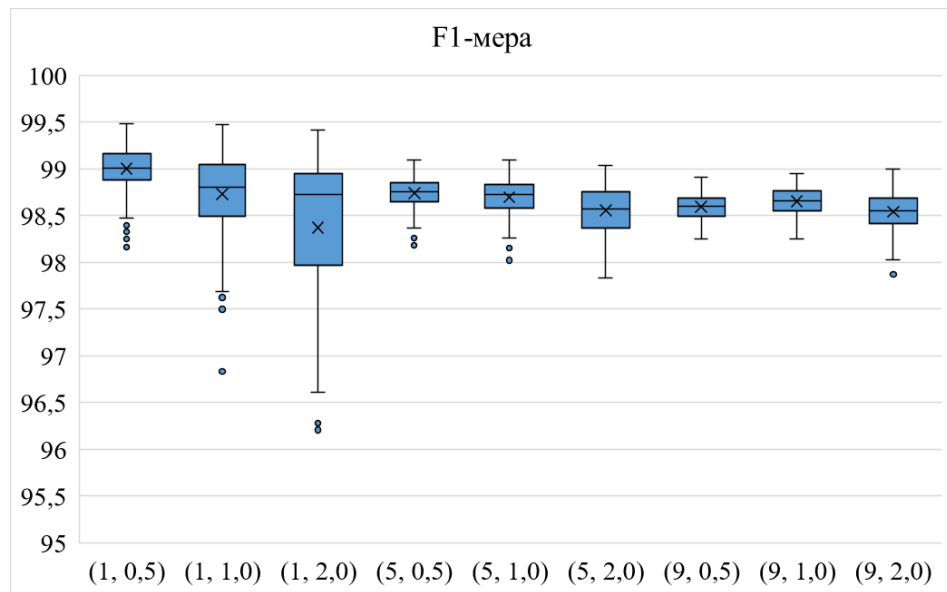
Так как при ручной разметке данных сложно указать точный центр цифры, то обучение и тестирование проводим с помощью трёх подходов: в первом используются только указанные центры, во втором используем окна с центрами, находящимися на расстоянии фон-Неймана не больше 1 (сам центр и 4 соседа), в третьем – находящимся на расстоянии Мура не больше 1 (сам центр и 8 соседей). В качестве критериев качества считаем среднюю  $F_1$ -меру по всем 10 классам цифр (*macroaverage*) и время, затраченное на обучение.

Результаты обучения показаны в таблице 4.8, для нейросетевых моделей также представлена диаграмма «ящик с усами» на рис. 4.7. Всего по каждому способу обучения было обучено по 200 *ANN*.

*ANN* показывает более стабильный результат при рассмотрении соседних точек, что является более предпочтительным, так как координаты центра цифры являются достаточно условным свойством, поэтому относительно высокий результат распознавания в нём (например, для *SVM*) нивелируется тем, что окружающие пиксели будут с большей вероятностью иметь неверные значения. Ярко заметен эффект переобучения сетей на маленьком наборе данных (1 или 5 точек для каждого центра), что приводит к тому, что при  $\alpha = 2$  *ANN* становятся нестабильными и показывают большее значение стандартного отклонения.

Таблица 4.8 – Результаты обучения первого каскада распознавания цифр

Метод распознавания	Обучающая выборка	Параметры метода	$F_1$ -мера, %	$T_{об}$ , с
ANN 209 / 209 $\alpha$ / 10	1 точка	$\alpha = \frac{1}{2}$	99,0 $\pm$ 0,2	4,3 $\pm$ 1,3
		$\alpha = 1$	98,7 $\pm$ 0,4	7,8 $\pm$ 2,6
		$\alpha = 2$	98,4 $\pm$ 1,2	15,8 $\pm$ 5,8
	5 точек	$\alpha = \frac{1}{2}$	98,7 $\pm$ 0,2	22 $\pm$ 3,2
		$\alpha = 1$	98,7 $\pm$ 0,2	40 $\pm$ 6,1
		$\alpha = 2$	98,6 $\pm$ 0,2	83 $\pm$ 18,9
	9 точек	$\alpha = \frac{1}{2}$	98,6 $\pm$ 0,1	45 $\pm$ 5,1
		$\alpha = 1$	98,7 $\pm$ 0,2	82 $\pm$ 9,4
		$\alpha = 2$	98,5 $\pm$ 0,2	176 $\pm$ 28
$k$ -NN	1 точка	$k = 1$	97,2	0,02
		$k = 3$	97,8	
	5 точек	$k = 1$	97,3	
		$k = 3$	97,6	
	9 точек	$k = 1$	97,3	
		$k = 3$	97,8	
SVM	1 точка	линейное ядро	98,4	0,4
	5 точек		95,5	3,4
	9 точек		94,1	18,8
DT	1 точка	$MNS = 100$	83,7	0,05
		$MNS = 1000$	83,7	
	5 точек	$MNS = 100$	84,9	0,2
		$MNS = 1000$	85,0	
	9 точек	$MNS = 100$	81,5	0,5
		$MNS = 1000$	84,3	

Рисунок 4.7 – Диаграмма «ящик с усами» для средних значений  $F$ -меры по 10 классам в зависимости от параметров ANN

Кроме непосредственно распознавания была обучена нейросеть для определения находится ли в данном месте цифра вообще: позитивными примерами класса выбирались пиксели из квадрата  $3 \times 3$  относительно центра точки, негативными – вершины и середины сторон квадрата  $9 \times 9$  относительно центра. Эта нейросеть используется только для разграничения рядом стоящих цифр, для её входа используются аналогичные окна,  $\alpha = \frac{1}{2}$ , средняя  $F$ -мера результата обучения равна 99,4%.

В качестве критериев качества работы первого каскада распознавания используем модификацию понятия полноты и точности. Под полнотой будем считать долю цифр, в окнах относительно центров которых есть хотя бы один пиксель с соответствующим значением:

$$Recall(d) = \frac{\sum_{digit} \cup_{digit.window} [output_1(x, y) = d]}{\sum_{digit} [digit.value = d]}, \quad (4.12)$$

где  $d$  – значение цифры, по которому считаем полноту,  $digit$  – множество размеченных цифр со свойствами  $value$  и множеством пикселей окна  $window$  (в данном случае принимается окно  $3 \times 3$ ),  $output_1$  – индексированное изображение, получившееся на выходе первого каскада.

Под точностью будем считать отношение количества цифр, в окнах относительно центров которых есть хотя бы один пиксель с соответствующим значением, к количеству найденных сегментов с данным значением:

$$Precision(d) = \frac{\sum_{digit} \cup_{digit.window} [output_1(x, y) = d]}{\sum_{segment} [segment.value = d]}. \quad (4.13)$$

Результаты работы первого каскада представлены в таблице 4.9,  $Recall$  и  $Precision$  усреднялись по принципу *microaverage*. Для нейросетей были выбраны модели, показавшие лучшие результаты на этапе обучения. Видно, что на данном этапе точность является достаточно низкой из-за большого количества найденных

посторонних сегментов; они будут отфильтрованы в последующих этапах. В качестве итоговой была выбрана *ANN*, которая обучалась по 9 точкам,  $\alpha = 1/2$ .

Таблица 4.9 – Качество первого каскада распознавания цифр на тестовых изображениях

Метод распознавания	Обучающая выборка	Параметры метода	<i>Recall</i> , %	<i>Precision</i> , %	<i>T</i> , с/Мп
<i>ANN</i> 209 / 209 $\alpha$ / 10	1 точка	$\alpha = 1/2$	99,8	0,9	0,7
		$\alpha = 1$	99,9	1,0	0,9
		$\alpha = 2$	99,8	0,9	1,4
	5 точек	$\alpha = 1/2$	99,8	1,3	0,7
		$\alpha = 1$	99,8	1,3	0,8
		$\alpha = 2$	99,8	1,2	1,4
	9 точек	$\alpha = 1/2$	99,8	1,3	0,8
		$\alpha = 1$	99,7	1,3	0,9
		$\alpha = 2$	99,7	1,3	1,4
<i>k-NN</i>	1 точка	$k = 1$	99,8	1,6	2,2
		$k = 3$	99,7	1,6	2,5
<i>SVM</i>	1 точка	линейное ядро	99,7	1,3	1,0
<i>DT</i>	1 точка	<i>MNS</i> = 100	97,1	0,8	0,4
		<i>MNS</i> = 1000	97,1	0,8	0,4

Обучение второго каскада распознавания производилось на суперсегментах, полученных после работы первого каскада для тренировочных изображений. Суперсегменты, не имеющие соответствия среди цифр (то есть полученные в результате низкой точности), были исключены из обучающей выборки для лучшего обучения. Существует нулевой класс «не цифра», но он подразумевает обширное число вариаций. Обучение было произведено над 2804 суперсегментами. Результаты обучения представлены в таблице 4.10 и рис. 4.8. *ANN* с лучшими результатами были использованы для запуска на тестовых изображениях.

В качестве критериев оценивания методов второго каскада были выбраны полнота, точность с учётом нулевого класса (*Common Precision*) и точность без его учёта (*Identity Precision*), которые считаются по формулам 3.2-3.4. При этом большее значение имеет именно последний критерий, так как на данном этапе

существует большое количество ложно распознанных цифр. Тем не менее, на данном этапе также была произведена следующие операции: фильтрация по размеру сегмента, разрезание слишком длинных сегментов, фильтрация сегментов по наличию соседствующих для удаления одиночных сегментов (параметры операций подобраны на обучающих изображениях).

Таблица 4.10 – Результаты обучения второго каскада распознавания цифр

Метод распознавания	Параметры метода	$F_1$ -мера, %	$T_{об}$ , с
<i>ANN</i> 10 / 10 $\alpha$ / 10 (всего обучено 100 <i>ANN</i> на каждый $\alpha$ )	$\alpha = 1$	$99,2 \pm 0,05$	$0,5 \pm 0,1$
	$\alpha = 2$	$99,2 \pm 0,05$	$0,5 \pm 0,1$
	$\alpha = 10$	$99,2 \pm 0,06$	$0,9 \pm 0,2$
<i>k-NN</i>	$k = 1$	99,4	0,2
	$k = 3$	99,4	
<i>SVM</i>	линейное ядро	99,4	1,2
<i>DT</i>	$MNS = 10$	86,2	0,1
	$MNS = 20$	99,4	
	$MNS = 100$	99,4	
	$MNS = 1000$	99,4	

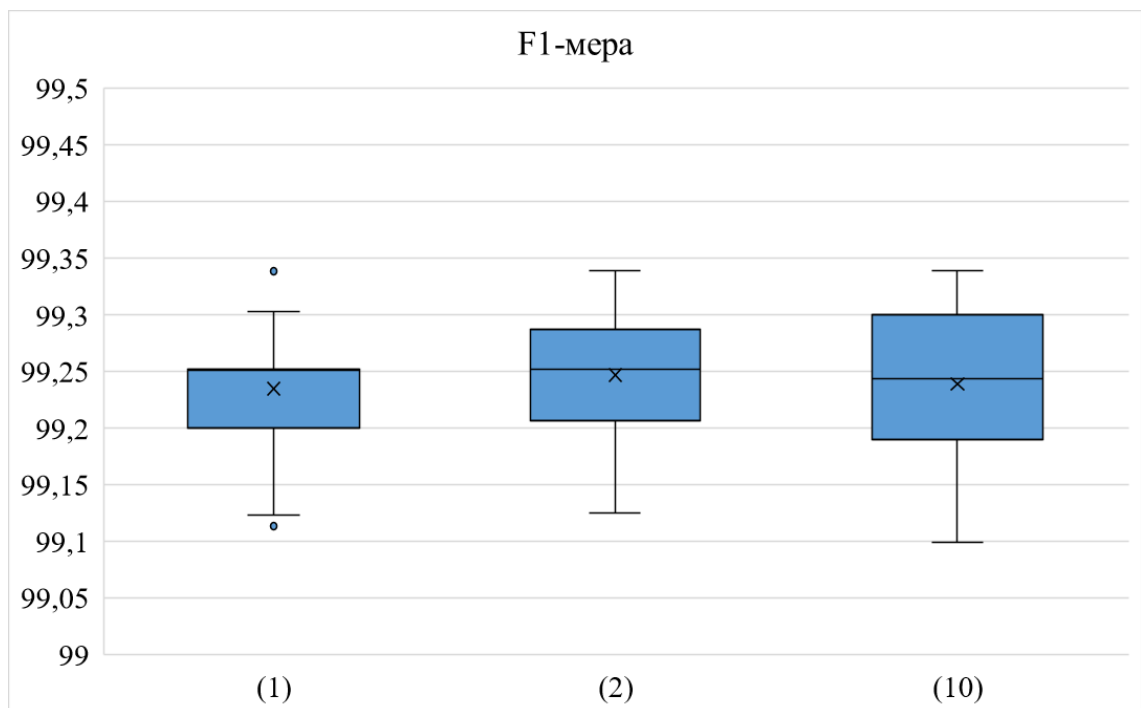


Рисунок 4.8 – Диаграмма «ящик с усами» для средних значений  $F$ -меры по 10 классам в зависимости от параметра *ANN*

Результаты применения второго каскада распознавания с помощью различных методов (в том числе для простого выбора индекса с максимальным значением входного параметра) представлены в таблице 4.11. Всего на тестовых изображениях присутствует 1835 цифр. Из-за небольшого числа обрабатываемых объектов общее время работы для всех моделей на этом этапе составило порядка 0,2 с/Мп.

По таблице видно, что все модели могут выдавать соизмеримые значения критериев качества, поэтому нет сильной разницы в том, какой метод будет выбран, но  $k$ -NN даёт лучший результат по критериям качества *Recall* и *Common Precision*, поэтому выбираем его.

Таблица 4.11 – Качество второго каскада распознавания цифр на тестовых изображениях

Метод распознавания	Параметры метода	<i>Recall</i> , %	<i>Common Precision</i> , %	<i>Identity Precision</i> , %
<i>ANN</i> 10 / 10 $\alpha$ / 10	$\alpha = 1$	99,1	31,8	99,8
	$\alpha = 2$	99,3	31,9	99,8
	$\alpha = 10$	99,3	32,0	99,8
<i>kNN</i>	$k = 1$	99,4	32,1	99,8
	$k = 3$	99,3	32,0	99,8
<i>SVM</i>	линейное ядро	99,3	31,9	99,8
<i>DT</i>	$MNS = 10$	89,6	31,0	99,8
	$MNS = 20$	99,3	32,3	99,8
	$MNS = 100$	99,3	32,3	99,8
	$MNS = 1000$	99,3	32,3	99,8
<i>max</i>	–	98,7	31,5	99,7

Можно отметить, что в процессе тестирования были найдены ошибки разметки данных: было пропущено две отметки высоты, 8 цифр в отметках были размечены с ошибкой. Исходя из этого можно заключить, что точность ручной разметки изначально составляла порядка 99,5%.

Переходим к распознаванию подписей горизонталей. Для первого каскада объекты представляют собой окна размером 11×17 пикселей, полученных с этапа сегментации соответствующего типа. Всего было размечено 750 цифр (250 чисел), тестирование производилось на 297 изображений цифр (99 чисел) из размеченных.



Так как для отметок высот более стабильный результат получался при рассмотрении центров окон, находящимся на расстоянии Мура не больше 1, то используем аналогичный подход к формированию выборки.

Для подписей горизонталей существует сильное разбалансирование классов: около трети всех размеченных цифр были нулями, при этом шестерок (наименьшего по количеству тренировочных экземпляров класса) было около 1%. Для улучшения стабильности распознавания обучающая выборка была дополнена изображениями цифр из отметок высот до равенства количества экземпляров каждого класса несмотря на то, что шрифт отметок высот отличается от шрифта подписей горизонталей: размеры цифр менялись через масштабирование, при это особенности начертания шрифта несколько размылись. Тестовая выборка при этом не изменялась, чтобы сохранить целевое назначение ИМ.

Используем только *ANN* модели для сравнения, так как для распознавания отметок высот данный тип ИМ показал наилучший результат. Результаты обучения показаны в таблице 4.12, всего было обучено по 100 *ANN* для каждой  $\alpha$ . Лучший результат показала нейросеть с  $\alpha = 1$ , обученная на сбалансированной выборке, поэтому выбираем её для дальнейшего использования.

Таблица 4.12 – Результаты обучения первого каскада распознавания цифр

Метод распознавания	Обучающая выборка	Параметры метода	$F_1$ -мера, %	$T_{об}$ , с
<i>ANN</i> 187 / 187 $\alpha$ / 10	9 точек, выборка без балансирования	$\alpha = \frac{1}{2}$	79,1 $\pm$ 1,2	15,8 $\pm$ 1,6
		$\alpha = 1$	79,4 $\pm$ 1,1	25,4 $\pm$ 2,2
		$\alpha = 2$	78,4 $\pm$ 1,6	42,7 $\pm$ 5,6
	9 точек, сбалансированная выборка	$\alpha = \frac{1}{2}$	82,6 $\pm$ 1,1	31,9 $\pm$ 4,8
		$\alpha = 1$	83,2 $\pm$ 1,0	61,7 $\pm$ 9,3
		$\alpha = 2$	82,4 $\pm$ 1,2	125,3 $\pm$ 17,3

Аналогично отметкам высот была обучена нейросеть для определения находится ли в данном месте цифра вообще: позитивными примерами класса выбирались пиксели из квадрата  $3 \times 3$  относительно центра точки, негативными – вершины и середины сторон квадрата  $7 \times 7$ . Эта нейросеть используется только для

разграничения рядом стоящих цифр, для её входа используются аналогичные окна,  $\alpha = 1/2$ , средняя  $F_1$ -мера результата обучения равна 97,1%.

Угол наклона цифр рассчитывался исходя из пролегающей около отметки горизонтали. Поэтому этап распознавания подписей должен выполняться после этапа группировки ЛО. Для получения угла наклона использовалась функция нахождения касательной к дискретной кривой. Для данной функции у кривой брались точки через каждые 10, которые с помощью кубических сплайнов интерполировались и по найденным многочленам производилось дифференцирование.

Результаты работы первого каскада распознавания: *Recall* равен 99,3%, *Precision* – 0,4%, время работы – 1,1 с/Мп. Полнота соответствует значению для отметок высот, а точность получилась примерно в 2-3 раза ниже, так как для подписей горизонталей на данном этапе неизвестно положение числа: для каждой точки на кривой существует два противоположно направленных вектора нормали к ней.

Так как второй каскад распознавания имеет более высокий уровень абстрагирования, то для его использования не являются важными особенности начертания и размеры распознаваемых цифр из-за использования долей классов в суперсегментах, а не абсолютных значений. Благодаря этому разработанную ИМ второго каскада распознавания отметок высот можно использовать для распознавания подписей горизонталей. Результаты работы второго каскада распознавания при этом будут такими: *Recall* – 96,3%, *Common Precision* – 4,1%, *Identity Precision* – 97,6%, время работы – 0,2 с/Мп.

Далее необходимо провести группировку цифр в числа. Всего для этого было проведено четыре операции: первичная кластеризация (по алгоритму 2.8), удаление чисел с небольшим значением (для отметок высот таковыми являются все, меньшие 10), геометрический и семантический анализ. Стоит отметить, что первичная кластеризация без использования идеи геометрического хеширования выполнялась в среднем в 5 раз дольше.

Геометрический анализ представляет собой поиск в кластере подмножеств отметок-цифр, которые больше всего похожи на числа по своим относительным

геометрическим свойствам. Так как при изображении отметок высоты на ТК используется немоноширный шрифт, то была получена матрица средних расстояний между всеми возможными парами цифр на тренировочных изображениях. Например, единицы находятся в среднем на расстоянии 9 пикселей от других цифр по горизонтали, а цифры типа 6 или 8 – на расстоянии 13 пикселей друг от друга. Учитывалось то, что цифра после десятичной запятой находится в среднем на 6 пикселей дальше, от стандартного расстояния.

Так как кластеры состоят из небольшого числа элементов, и мы знаем, из скольких цифр скорее всего будет состоять отметка высот, то мы можем перебрать все варианты, от большего количества элементов в подмножестве, к меньшим. В данном случае сначала рассматривались все возможные четверки, а потом тройки объектов. Далее рассматривалось два свойства: небольшая разница по вертикали каждой цифры относительно среднего значения и небольшое отклонение по горизонтали от необходимого. Если хотя бы один элемент не удовлетворял требованиям, то это подмножество убиралось из рассмотрения. Если требованиям удовлетворяло несколько подмножеств, то в качестве итоговых использовались цифры, которые подходили чаще остальных.

Семантическая фильтрация выполнялась для отсеивания значений, которые меньше всего подходили под правильные. Так как отметки отвечают за реальные значения высоты, которая меняется на ЦТК не так значительно, то можно предположить, что если будут найдены числа, сильно отличающиеся в меньшую или большую стороны, то они с большей вероятностью будут ложными.

Для реализации семантического фильтра отметок высот все полученные значения были отсортированы и найдены отношения большего элемента к меньшему в паре соседей. Если значения отличаются меньше, чем в 1,2 раза, то считаем такое отклонение нормальным. Далее находим самый длинный интервал значений с нормальными отклонениями – все значения вне этого интервала исключаем из итогового рассмотрения.

Для семантического фильтра подписей горизонталей используем информацию об интервале, полученном из семантического фильтра для отметок высот. Удаляем все подписи, которые не могут быть получены исходя из шага, с которым идут горизонтали (10 метров для масштаба 1 : 50 000, 5 для масштаба 1 : 25 000).

Критерии качества при проведении группировки цифр в числа представлены в таблице 4.13 (для отметок высот – столбцы I, для подписей горизонталей – столбцы II). Полученная ошибка в полноте и точности хорошо согласуется между числами и цифрами, учитывая, что число состоит из 3-4 цифр. Время работы алгоритма группировки цифр в числа составляет порядка 0,02 с/Мп.

Таблица 4.13 – Критерии качества при проведении этапов группировки цифр с помощью геометрического анализа

Этап	<i>Recall</i> , %		<i>Common Precision</i> , %		<i>Identity Precision</i> , %	
	I	II	I	II	I	II
Первичная кластеризация	91,3	78,8	22,6	1,1	91,3	79,6
Удаление кластеров, содержащих 1-2 цифры	91,3	78,8	49,1	2,3	91,7	79,6
Геометрическая фильтрация	97,8	82,8	73,4	8,6	98,7	92,1
Семантическая фильтрация	97,6	82,8	92,6	77,4	99,1	98,8

Сравнение полученных результатов с современными исследованиями распознавания ТО на ЦТК представлено в таблице 4.14. Кроме отличия в использовании различного типа ЦТК стоит отметить, что показатели точности и полноты в некоторых исследованиях рассчитывались для отдельно выделенных семплов, а некоторые запускались на изображении всей карты, то есть кроме задачи распознавания ещё решалась задача поиска образа.

По таблице видно, что полученный уровень распознавания чисел для предложенного алгоритма показывает лучшие результаты по точности, учитывая, что в исследовании используются ЦТК разрешения в т. ч. меньше 300 dpi, что доказывает гипотезу о возможности использования более простых интеллектуальных моделей при наличии соответствующей дополнительной обработки.

Таблица 4.14 – Сравнение результатов работы предложенного алгоритма распознавания ТО с другими современными исследованиями

Алгоритм	<i>Recall</i> , %	<i>Identity Precision</i> , %	Примечание
<i>GOOCR</i> [223], 2009	85,0	91,0	<i>French maps</i> , 300 <i>dpi</i> распознавание текста тест на картах
<i>OCR</i> [254], 2011	–	95,0	Неизвестный тип ЦТК хорошего качества
<i>Shape descriptor</i> [166], 2014	–	92,2	<i>Indian topographic maps</i> , 300 <i>dpi</i> , обучение/тест на семплах
<i>SLS-GHT</i> [206], 2017	–	93,3	тест на картах
<i>Faster R-CNN</i> [195], 2018	–	94,8	<i>USGS's US Topo</i> , 300 <i>dpi</i> , распознавание текста с учетом словаря
[225], 2020	–	85,0	<i>Indian topographic maps</i>
<i>Tesseract</i> [168], 2021	75,9	98,6	<i>Hungarian maps</i> , определение <i>ID</i> карты на рамке, тест на картах
<i>Mask TextSpotter V3</i> , <i>ResNet-50</i> [114], 2022	80,0	84,8	<i>USGS's US Topo</i> , 300 <i>dpi</i> ,
тот же, с аугментацией	93,2	98,2	обучение/тест на семплах
Предложенный	97,6	99,1	200-300 <i>dpi</i> , окна 11×19, 11×17, обучение на семплах, тест на картах

#### 4.5.4 Исследование алгоритмов группировки ЛО

Особенностью обработки горизонталей на изображении ЦТК является то, что они являются гораздо более сложными по описанию объектами, чем отметки высот или подписи горизонталей, а поэтому и время, необходимое на их разметку несоизмеримо большее. При этом распознавание типа горизонталей не так важно, как их геометрическое расположение, которое можно получить из изображения, полученного после этапа сегментации.

Так как основной задачей является корректное соединение линейных сегментов, то в качестве критериев качества будем информацией о количестве оставшихся КТ без учёта краевых точек, количестве возможных соединений, количестве ложных соединений, возникших на этапе, время автоматической обработки текущего этапа и время ручной обработки после текущего этапа.

Для получения начального множества КТ необходимо произвести локализацию ЛО, для чего преобразуем полутоновые изображения, полученные с этапа сегментации, в бинарные изображения, содержащие скелетизированные линейные сегменты. Для этого проведём бинаризацию по уровню 0,95, вычтем изображения, полученные после локализации подписей горизонталей, произведем МО закрытия, произведем скелетизацию, удалим небольшие сегменты и точки ветвления. Было выполнено ручное исправление на некоторых картах, так как в некоторых местах координатная сетка была классифицирована ко II типу сегментов или присутствовали другие объекты коричневого цвета (некоторые типы дорог или строений).

Геометрическая обработка заключается в проверке условий I-III для всех пар точек с постепенным соединением их с помощью отрезков. Для данного этапа количество возможных пар соединений считаем по количеству пар КТ, которые находятся на расстоянии не больше 50 пикселей. Номер итерации определяет максимально допустимое расстояние между соединяемыми КТ:  $R_{max} = 10 + 5 * i$ .

Относительная группировка заключается в проверке условия IV: при этом для каждой точки находится ближайшая и рассчитывается расстояние до всех остальных точек. Если разница расстояний больше минимального значения, то данные две КТ соединяем.

Результаты критериев качества реализации алгоритма для группировки горизонталей суммарно по всем тестовым картам представлены в таблице 4.15. Сравнение с современными алгоритмами представлено в таблице 4.16. Видно, что при некотором количестве ложных соединений количество оставшихся КТ значительно меньше, чем в других методах, что в сочетании с хорошим быстродействием свидетельствует о преимуществах предложенного алгоритма.

Таблица 4.15 – Критерии качества при проведении этапов группировки ЛО

Этап	Кол-во КТ	Кол-во возможных соединений	Количество ложных соединений	$T$ , с/Мп	$T_{руч}$ , с/Мп
Возможные соединения считаются по радиусу 50 пикселей					
Локализация ЛО	60 282	271 108	-	0,08	1,0
Геометрическая группировка, 1-я итерация	24 573	53 679	40	0,24	5,5
Геометрическая группировка, 2-я итерация	20 079	39 466		0,12	
Геометрическая группировка, 3-я итерация	17 677	32 981		0,11	
Относительная группировка, 1-я итерация	12 364	25 257	27	0,10	2,2
Относительная группировка, 2-я итерация	11 268	24 229		0,09	
Относительная группировка, 3-я итерация	11 022	24 124		0,09	
Возможные соединения считаются по радиусу 100 пикселей					
Фильтрация соединений по пересечению	11 022	27 390	532	0,02	200
Группировка по единственной паре	10 438	27 089		0,08	
Группировка максимального правдоподобия	2 895	5 300		0,54	

Таблица 4.16 – Сравнение результатов работы предложенного алгоритма группировки ЛО с другими современными исследованиями

Алгоритм	Кол-во КТ на Мп	Количество ложных соединений на Мп	Процент верных соединений	$T$ , с/Мп	Качество карт
[130], 2006	294	6	–	–	среднее
[170], 2010	63	5	83,4	–	высокое
[235], 2012	325	6	96,0	14	среднее
[267], 2016	256	0	–	35	плохое
[194], 2018	46		93,5	71	высокое
Предложенный	31	6	93,2	1,5	плохое

Дальнейшую автоматическую обработку видится невозможным осуществить предложенным подходом, так как либо остаются КТ, которые невозможно соединить прямыми отрезками, либо существует большое число равнозначных по возможности соединений, между которыми нельзя выбрать лучшее. Включение же в алгоритм криволинейных сплайнов для соединения значительно не улучшит качества, так как даже расчёт углов касательных в КТ, являющихся аналогом первой производной, является приближённым из-за сильного влияния дискретной природы раstra.

Стоит отметить сильное влияние рельефности и особенностей отображения карты на качество данного этапа: из 5300 оставшихся возможных соединений более 4400 находились на трёх картах, а оставшиеся 900 – на пяти. Время ручной обработки на рельефных картах повышается до 350 с/Мп, на равнинных – 90 с/Мп.

При большом значении времени ручной обработки относительно автоматической следует принять во внимание, что существующие в современных ПО процедуры оставляют данный этап либо полностью за оператором, либо в виде автоматизированной трассировки. Векторизация горизонталей ЦТК вручную по данным компаний, предоставляющих такую услугу, в зависимости от размера может занять порядка нескольких дней (порядка 2000-5000 с/Мп), поэтому данное сокращение времени ручной обработки можно считать достаточным результатом.

Сокращение времени работы можно также оценить в доле пикселей, отрисованных человеком по отношению ко всем значимым пикселям, не являющихся фоном, в нашем случае это пиксели горизонталей и полугоризонталей. Для ЦТК, используемых для тестирования, эта доля составляет 5,5%.

#### 4.5.5 Исследование алгоритма комплексного анализа

Для тестирования алгоритма комплексного анализа было создано 20 ЦМР размера 4088×4088 пикселей с помощью модифицированного алгоритма *diamond-square*, использующегося в параграфе 3.1. Экстремумы значений высоты



выбирались из интервала [100; 500]. Из ЦМР были получены горизонталы с шагом 10 метров; считаем, что имеют подписи 10% горизонталей. Направление ската для получения орграфы плана горизонталей получаем через обработку ЦМР.

Введём следующие погрешности распознавания, которые были выявлены в подпараграфе 4.5.3: пропуск подписи горизонталей, ложное распознавание цифр в них, ложное нахождение несуществующей подписи. Для исследования зафиксируем несколько конкретных вероятностей пропуска чисел (от 0% до 25% с шагом 5%, для разработанного алгоритма 16%), вероятность ложного распознавания цифр будем задавать от 0% до 50% (2,4% для реальных ЦТК по предложенному методу), ложное нахождение подписи установим в 2%, что соответствует значению для разработанного алгоритма.

Для оценки качества используем прежде всего информацию о подписях горизонталей, так как полугоризонталы не подписываются и могут помочь лишь на этапе интерполяции. Отметки высот также не будем использовать, так как по ним невозможно понять находящаяся рядом горизонталь имеет большее или меньшее значение высоты.

Для каждой пары значений вероятности пропуска чисел и ложного распознавания алгоритм запускался 100 раз для обеспечения вариативности. В качестве показателя качества была использована средняя абсолютная ошибка значений высот всех горизонталей на ЦТК, поделённая на размер шага по высоте (10 м). Отфильтрованные результаты исследования (применялась медианная фильтрация по 9 соседним точкам) приведены на рис. 4.9.

Применение разработанного алгоритма показало корректное восстановление всех значений высот горизонталей при вероятности ложного распознавания цифр до значения 14%. Из полученного можно заключить, что разработанные алгоритмы распознавания и группировки работают с достаточным запасом, что позволит использовать их и для более сложных изображений.

При проведении экспериментальных исследований было доказано, что не важно, какая горизонталь выбирается изначально в качестве базовой: алгоритм

может восстановить план горизонталей полностью только при известной информации о значении высоты одной горизонтали и наличии информации о необходимых скатах.

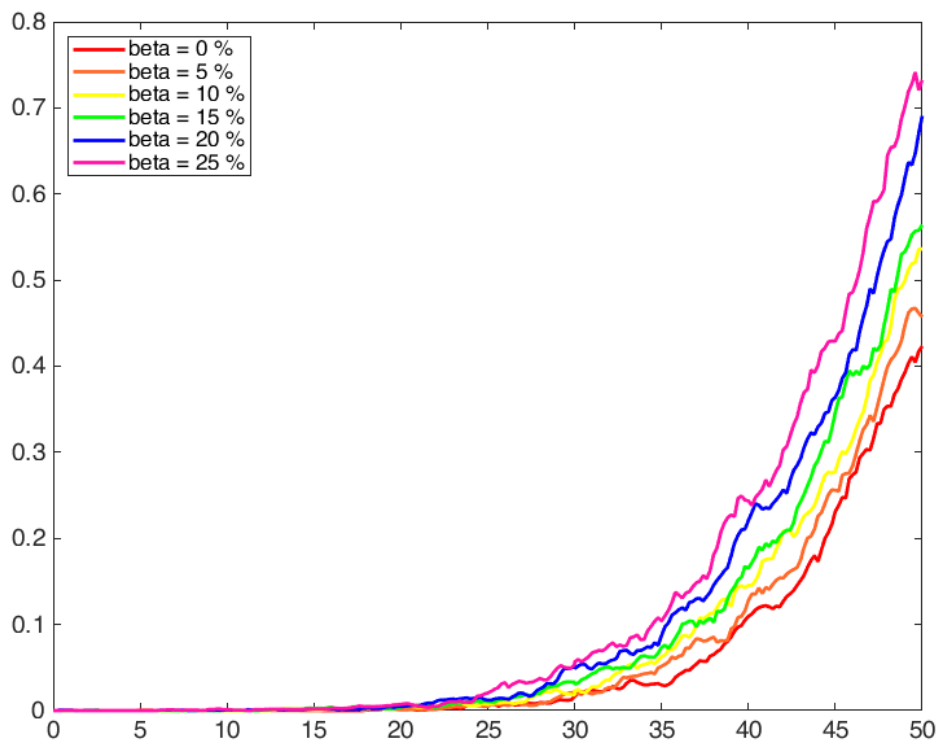


Рисунок 4.9 – Среднее значение доли неверно восстановленных значений подписей горизонталей в зависимости от вероятности ложного распознавания цифры и вероятности пропуска подписей  $\beta$  (показаны разным цветом)

Для запуска алгоритма комплексного анализа на реальных картах необходима информация о горизонталях и полугоризонталях, направление скатов, полученных на этапе нахождения бергштрихов, распознанные отметки высот.

Запущенный алгоритм комплексного анализа полностью смог восстановить значения высот подписей горизонталей, при этом были также удалены ложно распознанные значения, благодаря чему из всех типов ошибок (пропуск, неверное распознавание, ложное нахождение) остался только пропуск числа.

Алгоритм улучшил распознавание отметок высот для тех случаев, где ошибка распознавания была в разряде десятков и сотен метров. На этом же этапе

было произведено уточнение координат отметок высот с помощью нахождения рядом с числами соответствующих точек.

Скорость работы составила 10,6 с/Мп.

#### 4.5.6 Исследование алгоритма интерполяции

В таблице 4.17 представлены критерии качества методов интерполяции. Для анализа были выбраны различные значения доли пикселей  $P$ , используемых для интерполяции.

Таблица 4.17 – Критерии качества алгоритмов интерполяции

Метод	$P$ , %	$\Delta z_{max}$ , %	$\sigma_z$ , %	$T$ , с/Мп
Линейный	10	131	17,2	2,0
	20	117	15,6	2,5
	50	100	15,4	2,6
	100	99	15,4	3,1
Кубическими сплайнами	10	303	15,3	1,9
	20	170	14,1	2,3
	50	119	14,6	2,5
	100	99	14,9	3,4
Сибсона	10	157	16,6	7,3
	20	118	15,1	8,6
	50	100	14,9	12,1
	100	100	14,9	13,8
Предложенный без сглаживающего фильтра	10	84	16,4	8,5
	20	84	13,3	11,0
	50	86	11,5	17,0
Предложенный с фильтром	10	76	14,9	8,7
	20	84	12,6	11,1
	50	86	11,4	17,5

На рисунке 4.10 представлена гистограмма распределения отклонения в шагах для разных методов интерполяции ( $P = 20\%$ ). Предложенный алгоритм при чуть большем количестве значений с ошибкой в  $\pm 10\%$  от шага даёт гораздо меньше значений с ошибкой большей, чем  $50\%$ , что является предпочтительным.

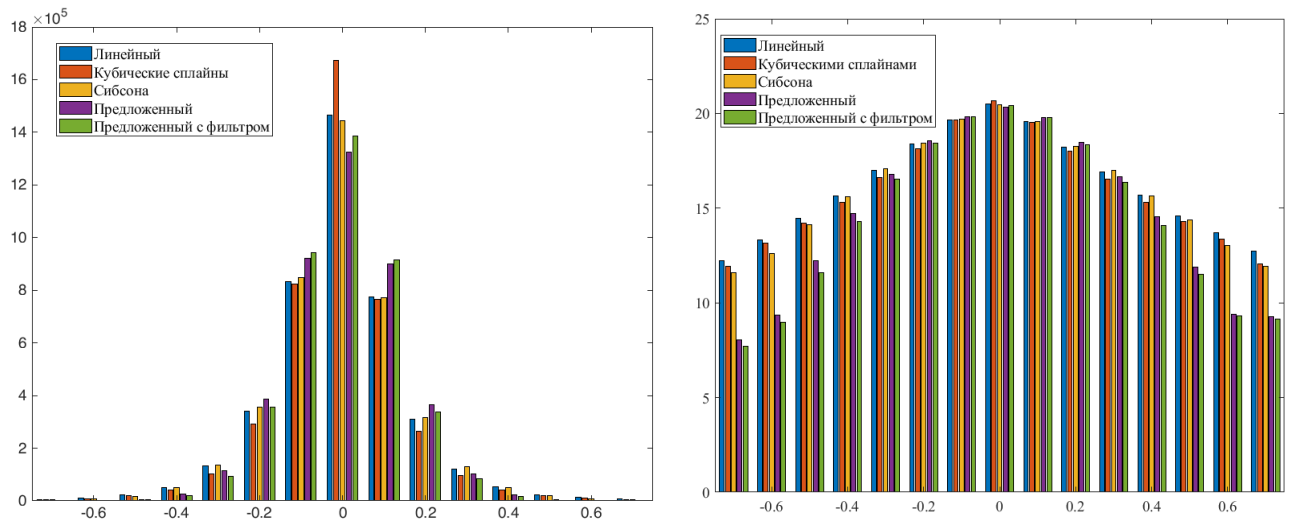


Рисунок 4.10 – Гистограмма распределения отклонений в долях шага для разных методов интерполяции при  $P = 20\%$  (справа с логарифмической шкалой)

Результат запущенного алгоритма интерполяции для участка ЦТК №3 в виде ЦМР показан на рис. 4.11: значения переведены в метры, для наглядности масштаб по вертикальной оси увеличен в 10 раз. Скорость работы интерполяции составила 14,4 с/Мп.

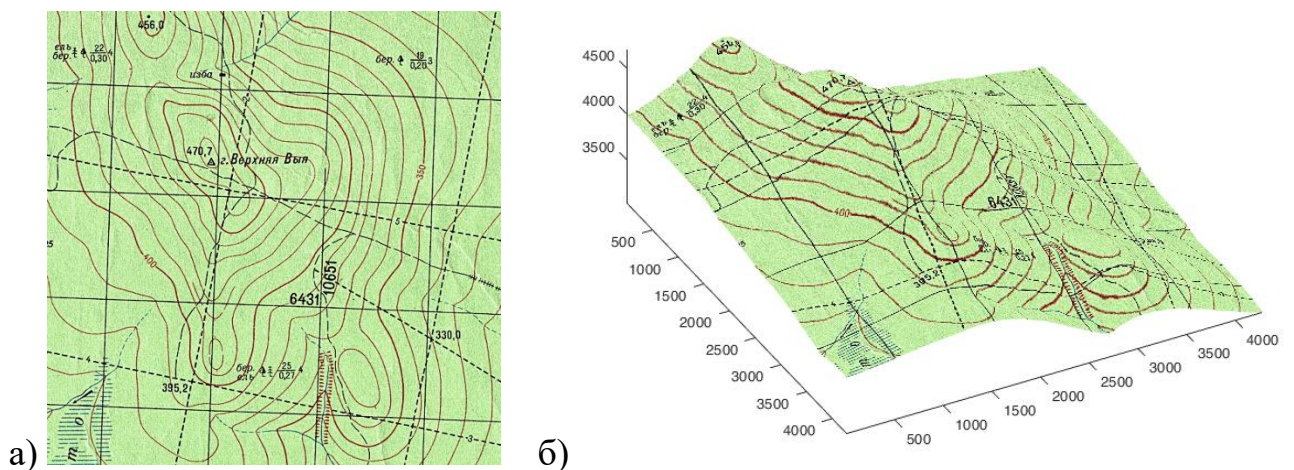


Рисунок 4.11 – Результат работы алгоритмов преобразования ЦТК в ЦМР: а) исходная ЦТК; б) полученная ЦМР

## 4.6 Выводы

Разработан алгоритм комплексного анализа распознанных отметок высот, подписей и линий горизонталей, скатов, позволяющего доопределить значения высот горизонталей и осуществить поиск ошибок распознавания, произошедших на предыдущих этапах. Алгоритм основан на использовании достоверной информации, что позволяет использовать его для ЦТК с недостающим количеством данных.

Разработан алгоритм интерполяции, учитывающий характер распределения данных на топографической карте. Внутренние точки, не лежащие на горизонталях, были классифицированы по трём типам, для каждого из которых был разработан свой алгоритм интерполяции. В качестве базового был принят принцип соответствия ЦМР рельефу местности.

Разработаны способы оценки качества алгоритмов, учитывающие характеристики карты и возможности классификаторов. Собраны данные необходимые для расчёта точности построения ЦМР и скорости работы алгоритмов. Суммарное время автоматической обработки ЦТК можно оценить в 50 с/Мп, необходимое время на ручную обработку – 250 с/Мп.

В данной главе были описаны реализованные блоки программного комплекса для обработки ЦТК, соответствующие различным этапам обработки согласно разработанной методике и алгоритмам. Результаты были сравнены с современными методами, использующимися при интеллектуальной обработке данных.

Разработанный алгоритм многометочной сегментации на основе искусственных нейросетей показал лучшие результаты по  $F_\beta$ -мере по сравнению с алгоритмами  $kNN$ ,  $SVM$  и  $DT$ . Недостатком является пониженное быстродействие, но так как для дальнейшей обработки сегментация является ключевым этапом, то был сделан вывод о преимуществе использования искусственных нейросетей для обработки цветных ССИ в задаче сегментации. Применение двойной нейросети не дало значительного улучшения качества, но в 1,5-1,7 раз увеличило время обработки, поэтому выбор был остановлен на одной нейросети. Отличительной особенностью

предложенной методики являлось то, что выбор архитектуры нейросети основывался не только на средних показателях качества и быстродействия, но и на показателе стабильности, выраженном в величине стандартного отклонения показателей качества и быстродействия.

Применение аппарата математической морфологии позволило создать программный модуль классификации сегментов, осуществляющий быструю обработку с сохранением высокой степени полноты и достаточной точности. Обработка сегментов первого и второго типа отличалась, но включала похожие этапы, различающиеся параметрами, свойственные различным шрифтам. Предложенный метод получения набора морфологических операций отличается от современных исследований использованием генетического алгоритма, предполагающего возможность изменения количества операций-генов в процессе мутации.

Предложенная программная реализация алгоритма распознавания точечных объектов в виде двухкаскадной процедуры с использованием интеллектуальных моделей отличается от современных моделей свёрточных нейросетей тем, что на вход второго каскада передается информация не о текущих пикселях или скользящих окнах, а о суперсегментах, представляющих собой объединение нескольких сегментов объектов, отнесённых к разным классам. На каждом из двух каскадов также рассматривались алгоритмы *kNN*, *SVM* и *DT*, но лучшие показатели относительно полноты и точности продемонстрировали искусственные нейронные сети. Ошибку распознавания удалось уменьшить относительно существующих методов обработки топографических карт в 2,8 раз для полноты, в 2 раза для точности.

Группировка цифр в числа, осуществлённая с помощью кластерного и геометрического, показала высокое быстродействие и качество. Совокупную точность распознавания чисел при сохранении полноты распознавания удалось повысить, благодаря использованию семантической фильтрации, заключающейся в использовании данных о распознанных ранее числах.

Тестирование итерационного алгоритма для геометрической группировки кусочков горизонталей показало высокое быстродействие и качество: уменьшение

количества конечных точек равно 30% по сравнению с современными разработками. Тем не менее, остаётся ещё достаточное количество конечных точек (порядка 5% от изначального количества), автоматизация объединения которых в настоящее время не была реализована, так как они соединяются либо по сложной криволинейной траектории, находясь на большом расстоянии, либо в небольшой области присутствует слишком большое количество участков горизонталей, которые невозможно соединить методом перебора. Такие операции в настоящее время предлагается оставить за оператором, при этом общее время нахождения горизонталей было значительно сокращено относительно работы на современном программном обеспечении.

Разработанный алгоритм комплексного анализа на основе теории графов и морфологической обработки полностью восстанавливает план горизонталей для текущего уровня ошибок распознавания чисел, при этом он может работать и в случае увеличения ошибки распознавания до уровня в 14% и вероятностью пропуска чисел до уровня в 25%. Экспериментально доказана независимость восстановления плана горизонталей из произвольного выбора одной корневой горизонтали.

Разработанный алгоритм интерполяции, учитывающий специфику плана горизонталей топографических карт, сравнивался с методом линейной интерполяции, интерполяции кубическими сплайнами и интерполяцией Сибсона. Предложенный метод с фильтрацией даже при учёте 20% точек горизонталей имеет на 15% меньшее значение максимального абсолютного отклонения и на 15% меньшее значение среднеквадратичной ошибки. Предложенный метод имеет меньшее быстроедействие, но это не является критичным при увеличении качества интерполяции.

Предложенные методики, алгоритмы и программные модули можно использовать в статичном виде при выполнении проектных, строительных и транспортных работ, например, в деятельности компании АО «УСГИК», АО «Уралгеоинформ», других компаний, занимающихся разработкой ГИС, а также в динамике при решении задачи управления мобильными роботами. Акты внедрения результатов работы находятся в Приложении В.

## Заключение

Основные результаты, которые были получены в ходе выполнения теоретических и экспериментальных исследований по теме, состоят в следующем.

– Проведён широкий и критический анализ методов и подходов, используемых при обработке цифровых изображений, анализу и распознаванию образов на них. Выявлены недостатки и преимущества оптического и интеллектуального подходов и было предложено использование гибридного, итерационного и комплексного подхода.

– Разработана классификация сложноструктурных изображений, основанная на способах и особенностях их получения, указаны их основные признаки. Для каждого типа описаны основные методы, использующиеся для их обработки.

– Разработана общая методика обработки сложноструктурных изображений, основанная на представлении их в виде точечных, линейных и площадных объектов, которую можно использовать для изображений различных типов приведённой классификации.

– Разработаны алгоритмы сегментации, локализации, распознавания, группировки и комплексного анализа объектов, находящихся на сложноструктурных изображениях. Для сегментации предложено использовать многометочный подход. Для локализации – поиск морфологических операций и операций фильтрации с помощью генетического алгоритма. Для распознавания – двухкаскадную систему, использующую для первого каскада метод скользящего окна, а для второго каскада – метод анализа суперсегментов. Для группировки точечных и линейных объектов приведены различные подходы к кластеризации, фильтрации и поиске отношений.

– Проведены экспериментальные исследования для нахождения оптимального типа и структуры интеллектуальных моделей, морфологических операций и операций фильтрации. В качестве исследуемых изображений выбраны смоделированные изображения морских навигационных карт и материалы спутниковых снимков. В обоих экспериментах была отработана методика обработки



сложноструктурных изображений и показана возможность применения предлагаемых алгоритмов.

– Сформулированы ключевые проблемы преобразования изображения сканированных топографических карт в объёмную модель местности. Выявлены основные проблемы, возникающие при обработке топографических карт, как цветных сложноструктурных объектов низкой точности и высокой информативной плотности.

– Предложенные алгоритмы применены для сегментации, локализации, распознаванию и группировке объектов на изображениях топографических карт. Создан алгоритм восстановления плана горизонталей, состоящий из алгоритма соединения горизонталей и комплексного анализа распознанных значений на предыдущих этапах. На основе разработанного подхода к классификации интерполяционных областей был создан алгоритм, учитывающий характер интерполяции контурных линий в двумерную матрицу.

– Разработан программный комплекс, состоящий из нескольких модулей, реализующих разработанные алгоритмы, выполняющий преобразование топографической карты в объёмную модель местности с минимальным участием оператора.

Из указанных результатов следует вывод, что цель и задачи диссертационного исследования выполнены в полном объёме.

**Рекомендации и перспективы развития темы исследования** анализа и распознавания сложноструктурных изображений заключается в следующем.

– Дальнейшая разработка алгоритмов, с помощью которых можно производить анализ и обработку над несколькими типами ССИ без изменения параметров и гиперпараметров интеллектуальных моделей и выполняемых операций.

– Разработка алгоритмов, учитывающих современное развитие аппаратного обеспечения, например, параллельных алгоритмов.

– Дальнейшее исследование стабильности указанных алгоритмов и интеллектуальных моделей и их оптимальности при решении задач распознавания.

– Создание онтологии обработки сложноструктурных изображений.

## Список сокращений и обозначений

ГА	–	генетический алгоритм
ИИ	–	искусственный интеллект
ИМ	–	интеллектуальная модель
ИНС	–	искусственная нейронная сеть
КТ	–	концевая точка
ЛДА	–	линейный дискриминантный анализ
ЛО	–	линейный объект
МНК	–	метод наименьших квадратов
МО	–	морфологическая операция
ПБ	–	программный блок
ПЛО	–	площадной объект
ССИ	–	сложноструктурное изображение
СЭМО	–	структурный элемент морфологической операции
ТО	–	точечный объект
ЦМР	–	цифровая модель рельефа
ЦТК	–	цифровая топографическая карта
ЭИ	–	элемент изображения
CNN	–	convolutional neural network
СТ	–	classification tree
DT	–	decision tree
FCNN	–	fast convolutional network
k-NN	–	k-nearest neighbors algorithm
LDA	–	linear discriminant analysis
GNB	–	Gaussian naive Bayes classifier
PCA	–	Principal component analysis
SVM	–	support vector machine

## Список использованной литературы

1. Акинин, М. В. Нейросетевой алгоритм уточнения векторных топографических карт по данным дистанционного зондирования Земли / М. В. Акинин // Цифровая обработка сигналов. – 2014. – №3. – С. 68-72.
2. Аксёнов, С. В. Повышение качества распознавания сцен нейронной сетью «Неокогнитрон» / С. В. Аксёнов, В. Б. Новосельцев // Известия Томского политехнического университета. – 2006. – Т. 309. – № 7. – С. 87-91.
3. Алябьев, А. А. Цифровое трехмерное моделирование местности на основании результатов спутниковой стереоскопической съемки / А. А. Алябьев, А. В. Сосновский // Геодезия и картография. – 2008. – № 8. – С. 23-28.
4. Антощук, С. Г. Гибридные информационные модели в системах обработки изображений / С. Г. Антощук, О. Ю. Бабилунга // Компьютинг. – 2009. – Т. 8(2). – С. 41-49.
5. Арлазаров, В. Л. Адаптивное распознавание / В. Л. Арлазаров, Н. В. Котович, О. А. Славин // Информационные технологии и вычислительные системы. – 2002. – Вып. 4. – С. 11-23.
6. Асмус, В. В. Контролируемая классификация данных дистанционного зондирования Земли / В. В. Асмус, А. А. Бучнев, В. П. Пяткин // Автометрия. – 2008. – Т. 44, № 4. – С. 60-67.
7. Афонасенко, А. В. Обзор методов распознавания структурированных символов / А. В. Афонасенко, А. И. Елизаров // Доклады ТУСУРа. – 2008. – Т. 18(2). – С. 83-88.
8. Базарский, О. В. Радиофизические методы оперативного формирования и анализа многочастотных сигналов со сложной структурой: диссертация на соискание учёной степени доктора физико-математических наук 01.04.03 / О. В. Базарский. – Воронеж: Воронежский государственный университет, 1999. – 236 с.
9. Барталев, В. А. Спутниковое картографирование растительного покрова России / С. А. Барталев, В. А. Егоров, В. О. Жарко [и др.] // Институт космических исследований Российской академии наук (ИКИ РАН). – 2016. – 208 с.
10. Батыршин, И. З. Нечеткие гибридные системы: теория и практика / И. З. Батыршин и др.; под ред. Н. Г. Ярушкиной. – М.: Физматлит, 2007. – 207 с.
11. Блейхут, Р. Быстрые алгоритмы цифровой обработки сигналов / Р. Блейхут: пер. с англ. И. И. Грушко. – М.: Мир, 1989. – 448 с.
12. Бурмистров, А. В. Метод формирования линейных контуров на аэрофотоснимках сельской местности [Электронный ресурс] / А. В. Бурмистров,

- И. И. Сальников // Современные проблемы науки и образования, 2013. – Режим доступа: <http://www.science-education.ru/ru/article/view?id=10674>.
13. Вапник, В. Н. Теория распознавания образов / В. Н. Вапник, А. Я. Червоненкис. – М.: Наука, 1974. – 416 с.
14. Васин, Д. Ю. Автоматизация обработки растровых изображений графических документов со слабо формализованным описанием объектов / В. Ю. Васин, В. П. Громов, С. И. Ротков // СРТ2019, Труды Международной научной конференции. – 2019. – С. 142-151.
15. Васин, Д. Ю. Исследование моделей описания, разработка алгоритмического, программного и технологического обеспечения обработки растровых изображений графических документов: диссертация на соискание учёной степени кандидата технических наук 05.01.01 / Д. Ю. Васин. – Нижний Новгород: НИИ ПМК ННГУ. – 2006. – 193 с.
16. Вежневцев, А. Выделение связанных областей в цветных и полутоновых изображениях [Электронный ресурс] / А. Вежневцев // Компьютерная графика и мультимедиа, 2003. – Вып. 1(5). – Режим доступа: <http://cgm.computergraphics.ru/content/view/53>.
17. Вежневцев, В. Оценка качества работы классификаторов. Компьютерная графика и мультимедиа [Электронный ресурс] // Компьютерная графика и мультимедиа, 2006. – Вып. 4(1). – Режим доступа: <http://cgm.computergraphics.ru/content/view/106>.
18. Визильтер, Ю. В. Использование проективных морфологий в задачах обнаружения и идентификации объектов на изображениях / Ю. В. Визильтер, С. Ю. Желтов // Известия Российской академии наук. Теория и системы управления. – 2009. – № 2. – С. 125-138
19. Вирт, Н. Алгоритмы и структуры данных / Н. Вирт. – М.: ДМК Пресс, 2011. – 272 с.
20. Волошин, Г. Я. Статистические методы решения задач распознавания, основанные на аппроксимационном подходе / Г. Я. Волошин, И. А. Бурлаков, С. Т. Косенкова. – Владивосток: ТОИ ДВО РАН, 1992. – 154 с.
21. Гаврилова, Т. А. Базы знаний интеллектуальных систем / Т. А. Гаврилова, В. Ф. Хорошевский. – СПб: Питер, 2001. – 384 с.
22. Галушкин, А. И. Нейронные сети. Основы теории. Монография / – М.: Горячая линия-Телеком, 2012. – 496 с. – ISBN 978-5-9912-0082-0.
23. Геоestatистика: теория и практика / В. В. Демьянов, Е. А. Савельева; под ред. Р. В. Арутюняна; Ин-т проблем безопасного развития атомной энергетики РАН. – М.: Наука, 2010. – 327 с.
24. Гольцев, А. Д. Модульная нейронная сеть с межмодульными обучающимися

- связями / А. Д. Гольцев, В. И. Гриценко // *Нейроинформатика*. – 2010. – № 2. – С. 214-224.
25. Гонзалес, Р. Цифровая обработка изображений в среде Matlab / Р. Гонзалес, Р. Вудс, С. Эддинс. – М.: Техносфера, 2006. – 616 с.
26. Горбань, А. Н. Нейронные сети на персональном компьютере / А. Н. Горбань, Д. А. Россиев. – Новосибирск : Новосибирский филиал ФГУП "Наука". – 1996. – 276 с. – ISBN 5-02-031196-0.
27. Горелик, А. Л. Методы распознавания: учеб. пособие для вузов / А. Л. Горелик, В. А. Скрипкин. – 4-е изд. – М.: Высшая школа, 2004. – 262 с.
28. ГОСТ Р 50828-95. Геоинформационное картографирование. Пространственные данные, цифровые и электронные карты. Общие требования. – М.: Изд-во стандартов, 1996. – 23 с.
29. Дмитриев, Н. В. Автоматизированная система преобразования топографических карт в геоинформационную модель / Н. В. Дмитриев // *Информационные технологии и нанотехнологии (ИТНТ-2016)*. – 2016. – С. 386-389.
30. Дмитриев, Н. В. Автоматизированная система распознавания символов на топографических картах / В. С. Тарасян, Н. В. Дмитриев // *Научно-методический электронный журнал «Концепт»*. – 2016. – Т. 11. – С. 1876-1880. – Режим доступа: <http://e-koncept.ru/2016/86402.htm>
31. Дмитриев Н. В. Алгоритм автоматического восстановления значений высот горизонталей / Н. В. Дмитриев, В. С. Тарасян // *Молодежь в науке: Новые аргументы: Сборник научных работ V-го Международного молодежного конкурса. Часть I*. – 2016. – с. 176-179.
32. Дмитриев, Н. В. Восстановление плана горизонталей при обработке топографических карт / Н. В. Дмитриев, В. С. Тарасян // *Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки*. – 2017. – № 1. – С. 56-60.
33. Дмитриев, Н. В. Выбор оптимальной стратегии при моделировании участка транспортной сети / Н. В. Дмитриев // *Научно-технический вестник Поволжья*. – 2019. – № 11. – С. 120-123.
34. Дмитриев Н. В. Гибридные процедуры автоматической обработки топографических карт / В. С. Тарасян, Н. В. Дмитриев // *Лучшая научная статья 2017: сборник статей XII Международного научно-практического конкурса*. – Пенза: МЦНС «Наука и Просвещение». – 2017 г. – с. 36-40.
35. Дмитриев, Н. В. Жадный алгоритм для моделирования железнодорожного пути в плане и профиле / Н. В. Дмитриев // *Вестник Ростовского государственного университета путей сообщения*. – 2019. – № 4 (76). – С. 110-119.

36. Дмитриев, Н. В. Интеллектуальная система анализа и преобразования топографических карт / Н. В. Дмитриев, В. С. Тарасян // Современные проблемы науки и образования. – 2015. – № 2-2. – Режим доступа: <http://www.science-education.ru/ru/article/view?id=21614>.
37. Дмитриев, Н. В. Интеллектуальный комплекс 3D-визуализации топографических карт / В. С. Тарасян, Н. В. Дмитриев // Современные проблемы автоматизации и управления в энергетике и машиностроении: сборник трудов международной научно-практической конференции. – Пенза: Пензенский государственный технологический университет. – 2015. – С. 153-161.
38. Дмитриев, Н. В. Интерполяция распределённых данных горизонталей для получения цифровой модели рельефа / Н. В. Дмитриев, В. С. Тарасян // Инженерный вестник Дона. – 2018. – № 1(48). – С. 85.
39. Дмитриев, Н. В. Комплексная система анализа сложноструктурных изображений / В. С. Тарасян, Н. В. Дмитриев // Интеграция образовательной, научной и воспитательной деятельности в организациях общего и профессионального образования: материалы IX Международной научно-практической конференции. – Екатеринбург: УрГУПС, 2017. – Вып. 9(228). – С. 194-198.
40. Дмитриев, Н. В. Метод восстановления значений на топографических картах для создания цифровой модели рельефа / Н. В. Дмитриев // Прикладная информатика. – 2019. – Т. 14. – № 6 (84). – С. 71-79.
41. Дмитриев, Н. В. Методика исследования устойчивости обучения нейронных сетей (на примере задачи классификации образов на сложноструктурных изображениях) / Н. В. Дмитриев // Научно-технический вестник Поволжья. – 2022. – № 12. – С. 108-111.
42. Дмитриев, Н. В. Повышение точности и производительности алгоритмов анализа и обработки сложноструктурных изображений: [Электронный ресурс]. – Режим доступа: <https://github.com/Logiraz/analysis-and-processing-of-images-of-complex-structure>. (Дата обращения 17.06.2024).
43. Дмитриев, Н. В. Построение объемной модели местности по топографической карте / Н. В. Дмитриев, В. С. Тарасян. – Свидетельство о государственной регистрации программы для ЭВМ. Номер и дата поступления заявки: № 2015662517 (18.12.2015). Номер регистрации (свидетельства): RU 2016611844. Дата регистрации: 11.02.2016. Правообладатель: ФГБОУ ВО «Уральский государственный университет путей сообщения».
44. Дмитриев, Н. В. Программный комплекс преобразования изображений топографических карт в цифровую модель рельефа / Н. В. Дмитриев. – Свидетельство о государственной регистрации программы для ЭВМ. Номер и дата поступления заявки: № 2019664218 (11.11.2019). Номер регистрации (свидетельства): RU

2019665260. Дата регистрации: 21.11.2019. Правообладатель: ФГБОУ ВО «Уральский государственный университет путей сообщения».
45. Дмитриев, Н. В. Распознавание площадных элементов на топографических картах / Н. В. Дмитриев, В. С. Тарасян // Научно-методический электронный журнал «Концепт». – 2017. – Т. 39. – С. 3476-3480.
46. Дмитриев Н. В. Система автоматизированного построения виртуальной модели местности по топографической карте / В. С. Тарасян, Н. В. Дмитриев // Молодежь в науке: Новые аргументы: Сборник научных работ II-го Международного молодежного конкурса. Часть I. – 2015. – с. 62-65.
47. Дмитриев, Н. В. Сравнение методов сегментации пикселей топографических карт по типам объектов / Н. В. Дмитриев, В. С. Тарасян // Известия Тульского государственного университета. Технические науки. – 2016. – № 3. – С. 97-106.
48. Дуда, Р. Распознавание образов и анализ сцен / Р. Дуда, П. Харт. – М.: Наука, 1976. – 512 с.
49. Дунаев, А. А. Алгоритмы быстрого поиска фрагментов фотографических изображений / А. А. Дунаев и др. // Сб. ст. Современные проблемы конструирования программ. – Новосибирск: Издательство Института систем информатики, 2002. – С. 88-109.
50. Дунаева, А. В. Обнаружение и локализация строений заданной формы на аэроснимках в инфракрасном диапазоне / А. В. Дунаева, Ф. А. Корнилов // Вестник ЮУрГУ. Серия: Вычислительная математика и информатика. – 2017. – Т. 6(3). – С. 84-100. – doi:10.14529/cmse170306.
51. Дьяконов, В. Обработка сигналов и изображений: специальный справочник / В. Дьяконов, И. Абраменкова. – СПб.: Питер, 2002. – 608 с.
52. Журавлев, Ю. И. Распознавание образов и распознавание изображений / Ю. И. Журавлёв, И. Б. Гуревич // Распознавание, классификация, прогноз. – М.: Наука, 1989. – С. 5-72.
53. Залманзон, Л. А. Преобразование Фурье, Уолша, Хаара и их применение в управлении, связи и других областях / Л. А. Залманзон. – М.: Наука, 1989 год. – 496 с.
54. Захаров, А. В. Критерии оценки качества сегментации изображений / А. В. Захаров и др. // Компьютерная оптика, 2015. – Т. 39. – № 4. – С. 87-99.
55. Ивахненко, А. Г. Перцептроны система распознавания образов / А. Г. Ивахненко. – Киев: Наукова думка, 1975. – 431 с.
56. Искусственный интеллект. – В 3-х кн. Кн. 2. Модели и методы: Справочник / Под ред. Поспелова Д.А. – М.: Радио и связь, 1990. – 304 с.
57. Калайда, В. Т. Модели и методы поддержки решения задач обработки и анализа изображений: диссертация на соискание учёной степени доктора

- технических наук 05.13.18 / В. Т. Калайда. – Томск: Томский государственный университет систем управления и радиоэлектроники. – 2006. – 307 с.
58. Котович, Н. В. Распознавание скелетных образов / Н. В. Котович, О. А. Славин // Сб. трудов ИСА РАН "Методы и средства работы с документами". – М.: Эдиториал УРСС. – 2000. – С. 204-215.
59. Кунаков, Ю. Н. Совместная обработка пространственной информации в виде цифровых карт и космических снимков для планирования противопаводковых мероприятий: диссертация на соискание учёной степени кандидата технических наук ВАК 05.13.01 / Ю. Н. Кунаков. – Уфа: Уфимский государственный авиационный технический университет. – 2009. – 232 с.
60. Ласло, М. Вычислительная геометрия и компьютерная графика на C++ / М. Ласло. – М.: Бином, 1997. – 304 с.
61. Лукашенко, Г. Применение волнового алгоритма для нахождения скелета растрового изображения [Электронный ресурс]. – Режим доступа: [www.ocrai.narod.ru/vectory.html](http://www.ocrai.narod.ru/vectory.html).
62. Люгер, Д. Искусственный интеллект / Д. Люгер. – М.: Мир, 2003. – 690 с.
63. Марлей, В. Е. Интеллектуальные информационные системы: основы теории и практики / В. Е. Марлей, В. Д. Чертовской. – СПб.: Санкт-Петербургский университет водных коммуникаций, 2010. – 173 с.
64. Медведев, В. С. Нейросетевой MATLAB 6 / В. С. Медведев, В. Г. Потёмкин. – М.: Диалог МИФИ, 2002. – 445 с.
65. Медведев, Д. Г. Метод выделения контуров объектов с нечеткими краями на цветных изображениях / Д. Г. Медведев // Цифровая обработка сигналов. – 2014. – №3. – С. 62-67.
66. Мерков, А. Б. Распознавание образов. Введение в методы статистического обучения / А. Б. Мерков. – М.: Едиториал УРСС, 2011. – 256 с.
67. Мирошниченко, С. Ю. Метод локализации, распознавания и определения типов значений высот на топографических картах / С. Ю. Мирошниченко, Е. Н. Дремов // Интеллектуальные информационные системы: тенденции, проблемы, перспективы. Материалы докладов IV региональной заочной научно-практической конференции "ИИС-2016". – 2017. – С. 155-157.
68. Мокеев, А. В. Об эффективности распознавания лиц с помощью линейного дискриминантного анализа и метода главных компонент/ А. В. Мокеев, В. В. Мокеев // Бизнес-информатика. – 2015. – № 3. – С. 44-54.
69. Никоноров, А. В. Коррекция искажений многоспектральных изображений на основе модели спектрально-контурных элементов // Компьютерная оптика. – 2014. – Т. 38, № 2. – С. 304-313. – doi: 10.18287/0134-2452-2014-38-2-304-313.
70. Новиков, А. И. Применение контурного анализа для совмещения



- изображений / А. И. Новиков, В. А. Саблина, Е. О. Горячев // Известия Тульского Государственного Университета. Технические науки. – 2013. – № 9. – С. 260-270.
71. Орлов, А. А. Алгоритм вычисления признаков отрезков на растровых изображениях, основанный на вычислении градиента яркости / А. А. Орлов, М. И. Ткачук // Инфокоммуникационные технологии. – 2008. – Т. 6. – № 3. – С. 36-41.
72. Осовский, С. Нейронные сети для обработки информации / С. Осовский: пер. с польского И. Д. Рудинского. – М.: Финансы и статистика, 2002. – 344 с.
73. Павлидис, Т. Алгоритмы машинной графики и обработки изображений / Т. Павлидис: пер. с англ. – М.: Радио и связь, 1986. – 400 с.
74. Паклин, Н. Алгоритмы кластеризации на службе Data Mining [Электронный ресурс]. – Режим доступа: <https://basegroup.ru/community/articles/datamining>.
75. Паламарь, И. Н. Метод сегментации изображений с применением выращивания областей и многомасштабного анализа / И. Н. Паламарь, П. В. Сизов // Вестник Сибирского гос. аэрокосмического ун-та им. акад. М.Ф. Решетнева, 2009. – № 3(24). – С. 15-20.
76. Патрик, Э. А. Основы теории распознавания образов / Э. А. Патрик: пер. с англ. под ред. Б. Р. Левина. – М.: Сов. радио, 1980. – 407 с.
77. Перечень открытых пространственных данных и материалов федерального фонда: цифровая картографическая продукция // Федеральный научно-технический центр геодезии, картографии и инфраструктуры пространственных данных. – Режим доступа: <https://cgkipd.ru/opendata/kartography> (дата обращения: 09.12.2023).
78. Поиск объектов на изображениях с использованием структурного дескриптора на основе графов / А. А. Захаров, А. Е. Баринов, А. Л. Жизняков, В. С. Титов // Компьютерная оптика. – 2018. – Т. 42, № 2. – С. 283-290. – doi: 10.18287/2412-6179-2018-42-2-283-290.
79. Полякова, М. В. Контурная сегментация изображений с использованием вейвлет-анализа / М. В. Полякова, В. В. Любченко // Сб. науч. тр. Одесского политехнического университета. – 2004. – Вып. 1(21). – С. 1-5.
80. Претт, У. Цифровая обработка изображений: в 2 т. / У. Претт: пер. с англ. под ред. Д. С. Лебедева. – М.: Мир, 1982.
81. Пытьев, Ю. П. Методы морфологического анализа изображений / Ю. П. Пытьев, А. И. Чуличков. – М.: Физматлит, 2010. – 336 с.
82. Редько, В. Г. Эволюционный подход к исследованию естественных и созданию искусственных "биокомпьютеров" / В. Г. Редько // Нейрокомпьютер. – 1994. – № 1(2). – С. 38-49.
83. Решетников, К. И. Исследование подхода «обнаружение объектов» в задаче

- фрагментации горных пород на открытых карьерах / К. И. Решетников, М. В. Ронкин, С. В. Поршнева // Компьютерная оптика. – 2024. – Т. 48, № 2. – С. 272-281. – doi: 10.18287/2412-6179-СО-1382.
84. Роджерс, Д. Математические основы машинной графики / Д. Роджерс, Дж. Адамс. – М.: Мир, 2001. – 604 с.
85. Рутковская, Д. Нейронные сети, генетические алгоритмы и нечеткие системы / Д. Рутковская, М. Пилиньский, Л. Рутковский. Пер. с польск. И. Д. Рудинского. – М.: Горячая линия – Телеком. – 2006. – 452 с.
86. Славин, О. А. Адаптивное распознавание и его применение к системе ввода печатного текста: диссертация на соискание учёной степени доктора технических наук 05.13.01 / О. А. Славин. – Москва: Институт системного анализа РАН. – 2011. – 310 с.
87. Стокман, Дж. Компьютерное зрение = Computer Vision. / Дж. Стокман, Л. Шапиро. – М.: Бином. Лаборатория знаний, 2006. – 752 с.
88. Теоретические основы цифровой обработки изображений / В. А. Соيفер, В. В. Сергеев, С. Б. Попов, В. В. Мясников. – Самара : Самарский национальный исследовательский университет имени академика С.П. Королева, 2000. – 256 с. – ISBN 5-7883-0109-2. – EDN YPFZEB.
89. Теплицкий, Э. Г. Метод обнаружения движущихся объектов на сложном динамическом фоне в оптическом диапазоне / Э. Г. Теплицкий, С. М. Захаров, М. А. Митрохин // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2016. – № 4. – С. 16-26. – doi: 10.21685/2072-3059-2016-4-2.
90. Титов, В. С. Построение цифровых моделей рельефа по растровым топографическим картам / В. С. Титов, С. Ю. Мирошниченко // Математические методы распознавания образов. – 2017. – Т. 18(1). – С. 88-89.
91. Титов, Ю. В. Об искажении символов при сканировании / Ю. В. Титов // Сб. трудов ИСА РАН "Системный подход к управлению информацией". – М.: КомКнига. – 2006. – Т. 23. – С. 260-288.
92. Томакова, Р. А. Проектирование гибридной нейронной сети для анализа сложноструктурированных медицинских изображений / Р. А. Томакова // Системный анализ и управление в биомедицинских системах. – 2011. – Т. 10. – № 4. – С. 916-923.
93. Томакова, Р. А. Гибридные методы и алгоритмы для интеллектуальных систем классификации сложноструктурируемых изображений: диссертация на соискание учёной степени доктора технических наук 05.13.17 / Р. А. Томакова. – Белгород: Белгородский государственный национальный исследовательский университет. – 2013. – 341 с.
94. Ту, Дж. Принципы распознавания образов / Дж. Ту, Р. Гонсалес. – М.: Мир,

1978. – 412 с.

95. Условные знаки для топографических планов масштабов М 1:5000, М 1:2000, М 1:1000, М 1:500, утверждены ГУГК при Совете Министров СССР 25 ноября 1986 г. – 239 с.

96. Файн, В. С. Оpozнaвание изображений / В. С. Файн. – М.: Изд-во «Наука», 1970. – 299 с.

97. Федотов, Н. Г. Теория признаков распознавания образом на основе стохастической геометрии и функционального анализа. / Н. Г. Федотов. – М.: ФИЗМАТЛИТ, 2009. – 304 с.

98. Фисенко, В. Т. Компьютерная обработка и распознавание изображений / В. Т. Фисенко, Т. Ю. Фисенко. – СПб.: СПбГУ ИТМО, 2008. – 192 с.

99. Фомин, Я. А. Статистическая теория распознавания образов / Я. А. Фомин, Г. Р. Тарловский. – М.: Радио и связь, 1986. – 624 с.

100. Форсайт, Д. Компьютерное зрение. Современный подход / Д. Форсайт, Ж. Понс. – М.: Вильямс, 2004. – 466 с.

101. Фраленко, В. П. Локализация текстовых фрагментов на смешанном фоне: краткий научный обзор / В. П. Фраленко // Программные системы: теория и приложения. – 2014. – № 2. – С. 33-45.

102. Фурман, Я. А. Цифровые методы обработки и распознавания бинарных изображений / Я. А. Фурман, А. Н. Юрьев, В. В. Яншин. – Красноярск: Изд-во Краснояр. ун-та, 1992. – 248 с.

103. Хрящев, В. В. Идентификация основных типов кормовых культур и сорняков на изображениях с полей методами искусственного интеллекта / В. В. Хрящев, А. С. Ключников, Н. В. Котов, Р. В. Ларионов // Вестник АПК Верхневолжья. – 2023. – № 3(63). – С. 98-104. – doi: 10.35694/YARCX.2023.63.3.013.

104. Хрящев, Д. А. Предварительная обработка и анализ цифровых изображений, полученных в условиях недостаточной освещенности: диссертация на соискание учёной степени кандидата технических наук 05.13.01 / Д. А. Хрящев. – Астрахань: Астраханский государственный технический университет. – 2013. – 145 с.

105. Цифровая обработка сигналов и изображений в радиофизических приложениях / М. А. Басараб, В. К. Волосюк, В. Ф. Кравченко [и др.]. – Москва : ООО Издательская фирма "Физико-математическая литература". – 2007. – 544 с. – ISBN 978-5-9221-0871-3.

106. Чернова, Л. И. Обработка геопространственной информации при цифровом моделировании топографических задач: диссертация на соискание учёной степени кандидата технических наук 05.13.18 / Л. И. Чернова. – Иркутск: Иркутский государственный университет путей сообщения. – 2006. – 123 с.

107. Яне, Б. Цифровая обработка изображений / Б. Яне. – М.: Техносфера, 2007. –

584 с.

108. Ярославский, Л. П. Введение в цифровую обработку изображений / Л. П. Ярославский. – М.: Сов. радио, 1979. – 312 с.

109. Abadie, J. Forest recovery since 1860 in a Mediterranean region: drivers and implications for land use and land cover spatial distribution / J. Abadie et al. // *Landscape Ecology*. – 2018. – V. 33(2). – Pp. 289-305. – doi:10.1007/s10980-017-0601-0.

110. Abdiansah, A. Time Complexity Analysis of Support Vector Machines (SVM) in LibSVM / A. Abdiansah, W. Retantyo // *International Journal of Computer Applications*. – 2015. – V. 128(3). – Pp. 28-34.

111. Agarwal, S. Recognizing electronic circuits to enrich web documents for electronic simulation / S. Agarwal, M. Agrawal, S. Chaudhury // *International Workshop on Graphics Recognition, GREC 2015*. – 2015. – Pp. 60-74. – doi:10.1007/978-3-319-52159-6\_5.

112. Anquez, P. Automatic correction and simplification of geological maps and cross-sections for numerical simulations / P. Anquez et al. // *Comptes Rendus Geoscience*. – 2019. – V. 351(1). – Pp. 48-58. – doi:10.1016/j.crte.2018.12.001.

113. Anwer, R. M. Binary patterns encoded convolutional neural networks for texture recognition and remote sensing scene classification / R. M. Anwer et al. // *ISPRS Journal of Photogrammetry and Remote Sensing*. – 2018. – V. 138. – Pp. 74-85. – doi:10.1016/j.isprsjprs.2018.01.023.

114. Arundel, S. T. Deep Learning Detection and Recognition of Spot Elevations on Historical Topographic Maps / S. T. Arundel, T. P. Morgan, P. T. Thiem // *Front. Environ. Sci*. – 2022. – 14 p. – doi:10.3389/fenvs.2022.804155.

115. Bailey, T. A note on distance-weighted k-nearest neighbor rules / T. Bailey, A. Jain // *IEEE Transactions on Systems, Man, and Cybernetics*. – 1978. – V 8. – Pp. 311-313.

116. Bhattacharyya, S. A brief survey of color image preprocessing and segmentation techniques / S. Bhattacharyya // *Journal of Pattern Recognition Research*. – 2011. – V. 6(1). – p. 120-129.

117. Bishop, C. M. *Neural networks for pattern recognition* / C. M. Bishop. – Oxford: Clarendon Press, 2015. – 483 p.

118. Bishop, C. M. *Pattern Recognition and Machine Learning* / C. M. Bishop. – N-Y.: Springer-Verlag New York, 2006. – 738 p.

119. Biswas, S. Representation and reconstruction of map regions / S. Biswas, S. Mandal, A. K. Das // *ICDAR*. – 2015. – Pp. 516-520.

120. Bollinger, J. Reconstructing forest-cover change in the Swiss Alps between 1880 and 2010 using ensemble modeling / J. Bollinger et al. // *Regional Environmental Change*. – 2017. – V. 17(8). – Pp. 2265-2277. – doi:10.1007/s10113-016-1090-4.

121. Boser, B. E. A training algorithm for optimal margin classifiers / B. E. Boser, I. M. Guyon, V. N. Vapnik // Proc. of the fifth annual workshop on Computational learning theory. – 1992. – Pp. 144-152. – doi:10.1145/130385.130401.
122. Bow, S.-T. Pattern recognition and Image Preprocessing / S.-T. Bow. – N-Y.: Marcel Dekker, 2002. – 720 p.
123. Breiman, L. Classification and Regression Trees / L. Breiman, J. Friedman, R. Olshen, and C. Stone. – N-Y.: Chapman and Hall, 1984. – 368 p.
124. Broelemann, K. Automatic understanding of sketch maps using context-aware classification / K. Broelemann, X. Jiang, A. Schwering // Expert Systems with Applications. – 2016. – V. 45. – Pp. 195-207. – doi:10.1016/j.eswa.2015.09.037.
125. Budig, B. Matching labels and markers in historical maps: an algorithm with interactive postprocessing / B. Budig, T. C. V. Dijk, A. Wolff // ACM Transactions on Spatial Algorithms and Systems. – 2016. – V. 2(4). – 24 p. – doi:10.1145/2994598.
126. Bunke, H. Handbook of character recognition and document image analysis / H. Bunke, P. S. P. Wang – Singapore: World Scientific, 1997. – 852 p.
127. Canny, J. F. A computational approach to edge detection / J. F. Canny // IEEE Trans. Pattern Recognition and Machine Intelligence. – 1986. – V. 8(6). – Pp. 679-698.
128. Cetinkaya, S. Proximity-based grouping of buildings in urban blocks: a comparison of four algorithms / S. Cetinkaya, M. Basaraner, D. Burghardt // Geocarto International. – 2015. – V. 30(6). – Pp. 618-632. – doi:10.1080/10106049.2014.925002.
129. Chen, L. C. Encoder-decoder with atrous separable convolution for semantic image segmentation / L. C. Chen et al. // Proceedings of the European conference on computer vision (ECCV). – 2018. – Pp. 801-818. – doi:10.48550/arXiv.1802.02611.
130. Chen, Y. Extracting contour lines from common-conditioned topographic maps / Y. Chen, R. Wang, J. Qian // IEEE Trans. Geosci. Remote. Sens. 44. – 2006. – Pp. 1048-1057. – doi:10.1109/TGRS.2005.861478.
131. Chiang, Y.-Y. A general approach for extracting road vector data from raster maps / Y.-Y. Chiang, C. A. Knoblock // International Journal on Document Analysis and Recognition. – 2013. – V. 16(1). – Pp. 55-81. – doi:10.1007/s10032-011-0177-1.
132. Chiang, Y.-Y. A Survey of Digital Map Processing Techniques / Y.-Y. Chiang, S. Leyk, C. A. Knoblock // ACM Computing Surveys. – 2014. – V. 47(1). – Pp. 1-44. – doi:10.1145/2557423.
133. Chiang, Y.-Y. Assessing the impact of graphical quality on automatic text recognition in digital maps / Y.-Y. Chiang et al. // Computers & Geosciences. – 2016. – V. 93. – Pp. 21-35. – doi:10.1016/j.cageo.2016.04.013.
134. Chiang, Y.-Y. Querying historical maps as a unified, structured, and linked spatio-temporal source: vision paper / Y.-Y. Chiang // Proceedings of International Conference

- on *Advances in Geographic Information Systems*. – 2015. – Pp. 1-4. – doi:10.1145/2820783.2820887.
135. Chiang, Y.-Y. Recognizing text in raster maps / Y.-Y. Chiang, C. A. Knoblock // *GeoInformatica*. – 2015. – V. 19(1). – Pp. 1-27.
136. Chiang, Y.-Y. Unlocking textual content from historical maps – Potentials and applications, trends, and outlooks / Y.-Y. Chiang // *1st International Conference on Recent Trends in Image Processing and Pattern Recognition. Communications in Computer and Information Science*. – 2017. – V. 709. – Pp. 111-124.
137. Cofer, R. H. Automated map reading and analysis by computer / R. H. Cofer, J. T. Tou // *AFIPS Conference Proceedings*. – 1972. – V. 1. – Pp. 135-145.
138. Corentin, H. Road Segmentation in SAR Satellite Images With Deep Fully Convolutional Neural Networks / H. Corentin et al. // *IEEE Geoscience and Remote Sensing Letters*. – 2008. – V. 15(12). – Pp. 1867-1871. – doi:10.48550/arXiv.1802.01445.
139. Costagliola, G. Local context-based recognition of sketched diagrams / G. Costagliola, M. De Rosa, V. Fuccella // *Journal of Visual Languages & Computing*. – 2014. – V. 25(6). – Pp. 955-962. – doi:10.1016/j.jvlc.2014.10.021.
140. Datta, R. Automatic Abstraction of Combinational Logic Circuit from Scanned Document Page Images / R. Datta, S. Mandal, S. Biswas // *Pattern Recognition and Image Analysis*. – 2019. – V. 29(1). – Pp. 212-223. – doi:10.1134/S1054661819020068.
141. De, P. Hierarchical vectorization of electrical drawings in document images by connectivity analysis of symbols and super-components / P. De, S. Mandal, P. Bhowmick // *Pattern Recognition and Image Analysis*. – 2017. – V. 27(2). – Pp. 309-325. – doi:10.1134/S1054661817020079.
142. De, P. Robust vectorization method for electrical circuit drawings using component morphology / P. De et al. // *Pattern Analysis and Applications*. – 2019. – V. 22(4). – Pp. 1341-1359. – doi:10.1007/s10044-018-0686-3.
143. Demir, I. DeepGlobe 2018: A Challenge to Parse the Earth through Satellite Images / I. Demir et al. // *2018 IEEE/CVF Conference CVPRW*. – 2018. – 10 p. – doi:10.1109/CVPRW.2018.00031.
144. Desai, S. Active Learning for Improved Semi-Supervised Semantic Segmentation in Satellite Images // *Accepted to Winter Conference on Applications of Computer Vision 2022*. – doi:10.48550/arXiv.2110.07782.
145. Devroye, L. A probabilistic theory of pattern recognition / L. Devroye, L. Györfi, G. Lugosi // *Applications of Mathematics: Stochastic Modelling and Applied Probability*, 1996. – V. 31. – 638 p.
146. Dhanachandra, N. Image segmentation using k-means clustering algorithm and subtractive clustering algorithm / N. Dhanachandra et al. // *Procedia Computer Science*.

– 2015. – V. 54. – Pp. 764-771.

147. Di, K. A machine learning approach to crater detection from topographic data / K. Di et al. // *Advances in Space Research*. – 2014. – V. 54. – Pp. 2419-2429.

148. Diligenti, M. Hidden tree Markov models for document image classification. / M. Diligenti, P. Frasconi, M. Gori // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 2003. – V. 25(4). – Pp. 519-523.

149. Ding, P. A light and faster regional convolutional neural network for object detection in optical remote sensing images / P. Ding et al. // *ISPRS Journal of Photogrammetry and Remote Sensing*. – 2018. – V. 141. – Pp. 208-218.

150. Dmitriev, N. Complex method of reconstruction of contour lines / N. Dmitriev // *AIP Conference Proceedings proceedings of the 45th international conference on application of mathematics in engineering and economics (AMEE'19)*. – 2019. – No. 080005. – 5 p. – doi:10.1063/1.5133563.

151. Dmitriev, N. Heuristic Algorithm for 3D Modelling of a Railway Track Route / N. Dmitriev // *Proceedings of the 1st International Scientific and Practical Conference "Transport: Logistics, Construction, Maintenance, Management"*. – 2022. – Pp. 136-142. – doi:10.5220/0000165000003527.

152. Drapeau, J. Extraction of Ancient Map Contents Using Trees of Connected Components / J. Drapeau et al. // *ICDAR*. – 2017. – Pp. 3-4. – doi:10.1109/ICDAR.2017.249

153. Duan, W. Automatic alignment of geographic features in contemporary vector data and historical maps / W. Duan et al. // *Proc. of the 1st Workshop on GeoAI: AI and Deep Learning for Geographic Knowledge Discovery*. – 2017. – Pp. 45-54. – doi:10.1145/3149808.3149816.

154. Duin, R. P. W. Open Issues in Pattern Recognition / R. P. W. Duin, E. Pekalska // *Computer Recognition Systems: Advances in Soft Computing*. – 2005. – V. 30. – Pp. 27-42.

155. Ebi, N. An image analysis system for automatic data acquisition from colored scanned maps / N. Ebi et al. // *Machine Vision and Applications*. – 1994. – V. 7. – Pp. 148-164. – doi:10.1007/BF01211660.

156. Egmont-Petersen, M. Image processing with neural networks - a review / M. Egmont-Petersen, D. De Ridder, H. Handels // *Pattern Recognition*. – 2002. – V. 35. – Pp. 2279-2301. – doi:10.1016/S0031-3203(01)00178-9.

157. El-Sawy, K. Automated, manual lineaments extraction and geospatial analysis for Cairo-Suez district (Northeastern Cairo-Egypt), using remote sensing and GIS / K. El-Sawy et al. // *IJISSET*. – 2016. – V. 3(5). – Pp. 491-500.

158. Elyan, E. Symbols Classification in Engineering Drawings / E. Elyan, C. M. Garcia, C. Jayne // *International Joint Conference on Neural Networks*. – 2018. – No. 8489087. – 8 p. – doi:10.1109/IJCNN.2018.8489087.

159. Fergani, B. A discrete particle swarm optimisation algorithm for geographical map contour reconstruction / B. Fergani, M.-K. Kholladi // 6th International Conference on Digital Information and Communication Technology and Its Applications. – 2016. – Pp. 142-144. – doi:10.1109/DICTAP.2016.7544016.
160. Ferraz, A. Large-scale road detection in forested mountainous areas using airborne topographic lidar data / A. Ferraz, C. Mallet, N. Chehata // ISPRS Journal of Photogrammetry and Remote Sensing. – 2016. – V. 112. – Pp. 23-36.
161. Fils, S. C. N. Radarsat-1 image processing for regional-scale geological mapping with mining vocation under dense vegetation and equatorial climate environment, South-western Cameroon / S. C. N. Fils et al. // The Egyptian Journal of Remote Sensing and Space Science. – 2018. – V. 21(1). – Pp. 43-54. – doi:10.1016/j.ejrs.2018.05.005.
162. Fischer, A. Lexicon-free handwritten word spotting using character HMMs / A. Fisher et al. // Pattern Recognition Letters. – 2012. – V. 33(7). – Pp. 934-942.
163. Fournier, A. Computer rendering of stochastic models / A. Fournier et al. // Communications of the ACM. – 1982. – V. 25(6). – Pp. 371-384. – doi:10.1145/358523.358553.
164. Fukunaga, K. Introduction to statistical pattern recognition / K. Fukunaga. – N-Y.: Academic Press, 1990. – 591 p.
165. Gámez, J. A. Advances in bayesian networks / J. A. Gamez et al. – Berlin: Springer-Verlag Berlin Heidelberg, 2004. – 328 p.
166. Ganpatrao, N. G. Information extraction from topographic map using colour and shape analysis / N. G. Ganpatrao, J. K. Ghosh // Sadhana. – 2014. – V. 39(5). – Pp. 1095-1117. – doi:10.1007/s12046-014-0270-5.
167. Gao, S. Designing a map legend ontology for searching map content / S. Gao, K. Janowicz, D. Zhang // Advances in Ontology Design and Patterns. – 2017. – V. 32. – Pp. 119-130. – doi:10.3233/978-1-61499-826-6-119.
168. Gede, M. Automatic Georeferencing of Topographic Map Sheets Using OpenCV and Tesseract / M. Gede, L. Varga // Proc. Int. Cartogr. Assoc. – 2021. – V. 4(38). – 4 p. – doi:10.5194/ica-proc-4-38-2021
169. Ghafari-Beranghar, A. Text separation from graphics by analyzing stroke width variety in Persian city maps / A. Ghafari-Beranghar, E. Kabir, K. Kargarloo // International Journal of Advanced Computer Science and Applications. – 2018. – V. 9(6). – Pp. 222-229. – doi:10.14569/IJACSA.2018.090632.
170. Ghircoias, T. Contour lines extraction and reconstruction from topographic maps / T. Ghircoias, R. Brad // UbiCC Journal. – 2009. – V. 6. – Pp. 681-692.
171. Ghosh, J. Neural networks for textured image processing / J. Ghosh, A. C. Bovik // Artificial Neural Networks and Statistical Pattern Recognition. – 1991. – V. 11. – Pp.133-154.



172. Gimmi, U. Assessing accuracy of forest cover information on historical maps / U. Gimmi et al. // *Prace Geograficzne*. – 2016. – V. 146. – Pp. 7-18.
173. Gobbi, S. New tools for the classification and filtering of historical maps / S. Gobbi et al. // *ISPRS International Journal of Geo-Information*. – 2019. – V. 8(10). – No. 455. – 24 p. – doi:10.3390/ijgi8100455.
174. Gurevich, I. B. Comparative analysis and classification of features for image models / I. B. Gurevich, I. V. Koryabkina // *Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications*. – 2006. – Vol. 16, No. 3. – P. 265-297. – doi: 10.1134/S1054661806030023.
175. Haykin, S. *Neural Networks: A Comprehensive Foundation* / S. Haykin. – N-Y.: MacMillan College Publishing Co., 1994. – 696 p.
176. He, X. Recognition of building group patterns in topographic maps based on graph partitioning and random forest / X. He, X. Zhang, Q. Xin // *ISPRS Journal of Photogrammetry and Remote Sensing*. – 2018. – V. 136. – Pp. 26-40.
177. Herold, H. 3D reconstruction of urban history based on old maps / H. Herold, R. Hecht // *5th Conference on Digital Encounters with Cultural Heritage. Communications in Computer and Information Science*. – 2018. – V. 817. – Pp. 63-79.
178. Herold, H. *Geoinformation from the Past: Computational Retrieval and Retrospective Monitoring of Historical Land Use* / H. Herold. – Wiesbaden: Springer Spektrum, 2017. – 192 p. – doi:10.1007/978-3-658-20570-6.
179. Hormann, K.  $C^1$ -continuous Terrain Reconstruction from Sparse Contours / K. Hormann, S. Spinello, P. Schroder // *Proceedings of the Vision, Modeling, and Visualization Conference*. – 2003. – Pp. 289-297.
180. Horodnic, V.-D. Geospatial analysis of land use dynamics using historical maps and GIS techniques. Case study of Rădăuți, Romania / V.-D. Horodnic et al. // *4th International Scientific Conference GEOBALCANICA 2018*. – 2018. – Pp. 577-594. – doi:10.18509/gbp.2018.63.
181. Janssen, R. D. T. Adaptive vectorization of line drawing images / R. D. T. Janssen, A. M. Vossepoel // *Computer Vision and Image Understanding*. – 1997. – V. 65(1). – Pp. 38-56.
182. Kang, S.-O. A digitization and conversion tool for imaged drawings to intelligent piping and instrumentation diagrams / S.-O. Kang, E.-B. Lee, H.-K. Baek // *Energies*. – 2019. – V. 12(13). – No. 2593. – 26 p. – doi:10.3390/en12132593.
183. Kanungo, T. An efficient k-means clustering algorithm: analysis and implementation / T. Kanungo et al. // *Proc. IEEE Conf. Computer Vision and Pattern Recognition*. – 2002. – Pp. 881-892.
184. Kawanami, R. Efficient Object Recognition with Multi-Directional Features in Urban Scenes / R. Kawanami, K. Matsushima // *ICSIIT 2017*. – 2017. – Pp. 39-44. – doi:

10.1109/ICSIIT.2017.56.

185. Kim, N. W. Accurate segmentation of land regions in historical cadastral maps / N. W. Kim et al. // *Journal of Visual Communication and Image Representation*. – 2014. – V. 25(5). – Pp. 1262-1274.

186. Kirillov, A. Panoptic Segmentation / A. Kirillov et al. // *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. – 2019. – Pp. 9396-9405. – doi: 10.1109/CVPR.2019.00963.

187. Kouril, D. Labels on Levels: Labeling of Multi-Scale Multi-Instance and Crowded 3D Biological Environments / D. Kouril et al. // *IEEE Transactions on Visualization and Computer Graphics*. – 2019. – V. 25(1). – Pp. 977-986.

188. Koutnik, J. Evolving Large-Scale Neural Networks for Vision-Based Reinforcement Learning / J. Koutnik et al. // *GECCO 2013, Proc. of the Genetic and Evolutionary Computation Conference*. – 2013. – Pp. 1061-1068. – doi:10.1145/2463372.2463509.

189. Krasil'nikov, N. N. Use of the principles of the optimal observer in modelling the human visual system / N. N. Krasil'nikov, Yu. E. Shelepin, O. I. Krasil'nikova // *Journal of Optical Technology*. – 1999. – V. 66. – № 9. – P. 782-787.

190. Kuo, T. Deep Aggregation Net for Land Cover Classification / *Proc. of 2018 IEEE/CVF Conference CVPRW*. – 2018. – p. 247-2474. – doi: 10.1109/CVPRW.2018.00046.

191. Lavrenko, V. Holistic word recognition for handwritten historical documents / V. Lavrenko, T. M. Rath, R. Manmatha // *Document Image Analysis for Libraries*. – 2004. – Pp. 278-287.

192. LeCun, Y. Gradient-based learning applied to document recognition / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // *Proceedings of the IEEE*. – 1998. – V. 86(11). – Pp. 2278-2323.

193. Leyk, S. Improving feature extraction of composite cartographic information in low-quality maps / S. Leyk, R. Boesch // *Cartography and Geographical Information Science*. – 2009. – 36(1). – Pp. 71-79.

194. Li, C. A reconstruction method for broken contour lines based on similar contours / C. Li et al. // *ISPRS International Journal of Geo-Information*. – 2019. – V. 8(1). – No. 8. – 14 p. – doi:10.3390/ijgi8010008.

195. Li, H. Intelligent Map Reader: A Framework for Topographic Map Understanding With Deep Learning and Gazetteer / H. Li et al. // *IEEE Access*. – 2018. – V. 6. – Pp. 25363-25376. – doi:10.1109/ACCESS.2018.2823501.

196. Liao, P.-S. A fast algorithm for multilevel thresholding / P.-S. Liao, T.-S. Chen, P.-C. Chung // *Journal of Information Science and Engineering*. – 2001. – V. 17. – Pp. 713-727.

197. Liu, T. A contour-line color layer separation algorithm based on fuzzy clustering and region growing / T. Liu et al. // *Computers & Geoscience*. – 2016. – V. 88. – Pp. 41-53. – doi:10.1016/j.cageo.2015.12.017.
198. Liu, T. A review of recent advances in scanned topographic map processing / T. Liu, P. Xu, Sh. Zhang // *Neurocomputing*. – 2019. – V. 328. – Pp. 75-87. – doi:10.1016/j.neucom.2018.02.102.
199. Liu, T. Color topographical map segmentation algorithm based on linear element features / T. Liu et al. // *Multimedia Tools and Applications*. – 2016. – V. 75. – Pp. 5417-5438. – doi:10.1007/s11042-015-2510-z.
200. Liu, T. SCTMS: Superpixel based color topographic map segmentation method / T. Liu et al. // *Journal of Visual Communication and Image Representation*. – 2016. – V. 35. – Pp. 78-90.
201. MacKay, D. J. C. *Information Theory, Inference and Learning Algorithms* / D. J. C. MacKay. – Cambridge: Cambridge University Press, 2005. – 640 p.
202. Maini, R. Study and comparison of various image edge detection techniques / R. Maini, H. Aggarwal // *International Journal of Image Processing*. – 2009. – V. 3(4). – Pp. 1-12.
203. Marinai, S. Artificial neural networks for document analysis and recognition / S. Marinai, M. Gori, G. Soda // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 2005. – V. 27(1). – Pp. 23-35.
204. Mayhua, A. Extracting Visual Encodings from Map Chart Images with Color-Encoded Scalar Values / A. Mayhua et al. // *31st SIBGRAPI Conference on Graphics, Patterns and Images*. – 2018. – Pp. 142-149. – doi:10.1109/SIBGRAPI.2018.00025.
205. Mazurov, V. D. Committee Constructions for Solving Problems of Selection, Diagnostics, and Prediction / V. D. Mazurov, M. Yu. Khachai, A. I. Rybin // *Proceedings of the Steklov Institute of Mathematics*. – 2002. – No. S1. – Pp. 67-101.
206. Miao, Q. Guided superpixel method for topographic map processing / Q. Miao et al. // *IEEE Transactions on Geoscience and Remote Sensing*. – 2016. – V. 54 (11). – Pp. 6265-6279. – doi:10.1109/TGRS.2016.2567481.
207. Miao, Q. The recognition of the point symbols in the scanned topographic maps / Q. Miao et al. // *IEEE Transactions on Image Processing*. – 2017. – V. 26(6). – Pp. 2751-2766. – doi:10.1109/TIP.2016.2613409.
208. Mitchell, T. M. *Machine learning* / T. M. Mitchell. – N-Y.: McGraw-Hill Education, 1997. – 414 p.
209. Miyoshi, T. Automatic extraction of buildings utilizing geometric features of a scanned topographic map / T. Miyoshi et al. // *ICPR*. – 2004. – V. 3. – Pp. 626-629.
210. Møller, M. F. A scaled conjugate gradient algorithm for fast supervised learning / M. F. Møller // *Neural Networks*. – 1993. – V. 6(4). – Pp. 525-533. – doi: 10.1016/S0893-

6080(05)80056-5.

211. Moreno-Garcia, C. F. New trends on digitisation of complex engineering drawings / C. F. Moreno-Garcia, E. Elyan, C. Jayne // *Neural Computing and Applications*. – 2019. – V. 31(6). – Pp. 1695-1712. – doi:10.1007/s00521-018-3583-1.

212. Music, J. Performance of compressive sensing image reconstruction for search and rescue / J. Music et al. // *IEEE Geoscience and Remote Sensing Letters*. – 2016. – V. 13(11). – Pp. 1739-1743. – doi:10.1109/LGRS.2016.2606767.

213. Nazari, N. H. Integrating text recognition for overlapping text detection in maps / N. H. Nazari, T. Tan, Y. Y. Chiang // *Proc. of Document Recognition and Retrieval, International Symposium on Electronic Imaging*. – 2016. – Pp. 1-8. – doi: 10.2352/ISSN.2470-1173.2016.17.DRR-061.

214. Netzer, Y. Reading Digits in Natural Images with Unsupervised Feature Learning / Y. Netzer et al. // *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*. – 2011. – URL: [http://ai.stanford.edu/~twangcat/papers/nips2011\\_house-numbers.pdf](http://ai.stanford.edu/~twangcat/papers/nips2011_house-numbers.pdf) (дата обращения: 09.12.2023)

215. Nguyen, H. T. Watersnakes: Energy-driven watershed segmentation / H. T. Nguyen, M. Worring, R. V. D. Boomgaard // *IEEE Trans. Pattern Analysis and Machine Intelligence*. – 2003. – V. 25. – Pp. 330-342.

216. Oka, S. Vectorization of contour lines from scanned topographic maps / S. Oka, A. Garg, K. Varghese // *Automation in Construction*. – 2012. – V. 22. – Pp. 192-202. – doi:10.1016/j.autcon.2011.06.017.

217. OpenCV Library [Электронный ресурс]. – Режим доступа: <http://opencv.org/>.

218. Ostafin, K. Forest cover mask from historical topographic maps based on image processing / K. Ostafin et al. // *Geoscience Data Journal*. – 2017. – V. 4(1). – Pp. 29-39. – doi:10.1002/gdj3.46.

219. Ozah, A. P. Accuracy assessment of contour interpolation from 1:50,000 topographical maps and SRTM data for 1:25,000 topographical mapping / A. P. Ozah, O. Kufoniyi // *ISPRS Archives*. – 2008. – V. XXXVII-B7. – Pp. 1347-1354.

220. Ozturk, C. Improved clustering criterion for image clustering with artificial bee colony algorithm / C. Ozturk et al. // *Pattern Analysis and Applications*. – 2015. – V. 18(3). – Pp. 587-599. – doi:10.1007/s10044-014-0365-y.

221. Peller, P. From Paper Map to Geospatial Vector Layer / P. Peller // *IASSIST Quarterly*. – 2018. – V. 42(3). – Pp. 1-24. – doi:10.29173/iq914.

222. Pinto, A. T. From archived historical aerial imagery to informative orthophotos: A framework for retrieving the past in long-term socioecological research [Technical Note] / A. T. Pinto et al. // *Remote Sensing*. – 2019. – V. 11(11). – No. 1388. – 15 p. – doi:10.3390/rs11111388.

223. Pouderoux, J. Toponym recognition in scanned color topographic maps /

- J. Poudroux et al. // ICDAR. – 2007. – V. 1. – Pp. 531-535.
224. Powell, W. B. Approximate Dynamic Programming: Solving the Curses of Dimensionality / W. B. Powell. – Hoboken: John Wiley & Sons, 2011. – 656 p.
225. Pradhan, A. Automatic Localization of Elevation Values in a Poor Quality Topographic Map / A. Pradhan, M. P. Pradhan // International Journal of Image and Graphics. – 2021. – V. 21(01). – Pp. 2150009. – doi:10.1142/S0219467821500091.
226. Quinlan, J. R. C4.5: Programs for Machine Learning / J. R. Quinlan. – San Mateo: Morgan Kaufmann Publishers Inc., 1993. – 302 p.
227. Rauch, A. Trace Information Extraction (TIE): A new approach to extract structural information from traces in geological maps / A. Rauch et al. // Journal of Structural Geology. – 2019. – V. 126. – Pp. 286-300. – doi:10.1016/j.jsg.2019.06.007.
228. Rizdania Text detection and recognition using multiple phase method on various product label for visual impaired people / Rizdania, F. Utaminingrum // SIET. – 2017. – Pp. 398-404. – doi:10.1109/siet.2017.8304171.
229. Rosenblatt, F. Analytic techniques for the study of neural nets / F. Rosenblatt // IEEE Trans. on Appl. and Industry. – 1964. – V.83(74). – Pp. 285-292.
230. Russell, S. Artificial Intelligence: A Modern Approach / S. Russell, P. Norvig. – L.: Pearson, 2013. – 1091 p.
231. Rychazhkov, S. Methods for automated vectorization of point objects on cartographic images / S. Rychazhkov, V. Fedoseev, R. Yuzkiv // ITNT. CEUR Workshop Proc. – 2017. – V. 1901. – Pp. 219-225. – doi: 10.18287/1613-0073-2017-1901-219-225.
232. Sadri, J. Application of support vector machines for recognition of handwritten Arabic/Persian digits/ J. Sadri, C. Y. Suen, T. D. Bui // Proceedings of the Second Conference on Machine Vision and Image Processing and Application. – 2003. – V. 1. – Pp. 300-307.
233. Sahu, A. Image denoising with a multi-phase kernel principal component approach and an ensemble version / A. Sahu, G. Runger, D. Apley // IEEE Applied Imagery Pattern Recognition Workshop: Imaging for Decision Making. – 2011. – № 6176339. – 7 p.
234. Salvatore, S. Contour line recognition from scanned topographic maps / S. Salvatore, P. Guitton // Journal of WSCG, 2004. – V. 12 (1-3). – p 419-426.
235. Samet, R. A new approach to the reconstruction of contour lines extracted from topographic maps / R. Samet, E. Hancer // J. Vis. Commun. Image Represent. 23. – 2012. – Pp. 642-647. – doi:10.1016/j.jvcir.2012.02.005.
236. Samsonov, T. Automated placement of supplementary contour lines / T. Samsonov et al. // International Journal of Geographical Information Science. – 2019. – V. 33. – Pp. 2072-2093. – doi:10.1080/13658816.2019.1610965.
237. Santos, R. Filtering and segmentation of digitized land use map images / R. Santos et al. // International journal on document analysis and recognition. – 1998. – V. 1(3). –

Pp. 167-174.

238. Santosh, K. C. Document Image Analysis: Current Trends and Challenges in Graphics Recognition / K. C. Santosh. – Singapore: Springer, 2018. – 174 p.

239. Schaffer, G. Quantifying the completeness of and correspondence between two historical maps: A case study from nineteenth-century Palestine / G. Schaffer, M. Peer, N. Levin // *Cartography and Geographic Information Science*. – 2016. – V. 43(2). – Pp. 154-175. – doi:10.1080/15230406.2015.1029519.

240. Schapire, R. E. The boosting approach to machine learning / R. E. Schapire // *Nonlinear Estimation and Classification*. – 2003. – V. 171. – Pp. 149-171.

241. Serra, J. Image Analysis and Mathematical Morphology / J. Serra. – London: AcademicPress. – 1982. – 610 p.

242. Shahbazpanahi, S. Finding optimal transformation function for image thresholding using genetic programming / S. Shahbazpanahi, S. Rahnamayan // 2014 IEEE Symposium on Computational Intelligence for Multimedia, Signal and Vision Processing (CIMSIVP). – 2014. – Pp. 1-8. – doi:10.1109/CIMSIVP.2014.7013279.

243. Sharma, A. Character recognition using neural networks / A. Sharma, D. R. Chaudhary // *International Journal of Engineering Trends and Technology*. – 2013. – V. 4(4). – Pp. 662-667.

244. Shinde, A. Preimages for variation patterns from kernel PCA and bagging / A. Shinde, A. Sahu, D. Apley, G. Runger // *IIE Transactions (Institute of Industrial Engineers)*. – 2014. – V. 46 (5). – Pp. 429-456.

245. Shoaib, B. M. An improved neural network based edge detection method: technical report / B. M. Shoaib et al. // Nagoya Institute of Technology, 1994. – 6 p.

246. Sibson, R. A brief description of natural neighbor interpolation / R. Sibson // V. Barnett. *Interpolating Multivariate Data*. Chichester: John Wiley, 1981. – pp. 21-36.

247. Soille, P. Morphological image compositing / P. Soille // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. – 2006. – V. 28(5). – Pp. 673-683.

248. Song, J. The reconnection of contour lines from scanned color images of topographical maps based on GPU implementation / J. Song et al. // *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. – 2017. – V. 10(2). – Pp. 400-408. – doi:10.1109/JSTARS.2016.2580903.

249. Sonka, M. Image Processing, Analysis and Machine Vision / M. Sonka, V. Hlavac, R. Boyle. – L.: Chapman & Hall, 1993. – 555 p.

250. Stojanović, M. Analysis of Cartographic Generalization based on PYTHON Programming Language on Digital Topographic Maps / M. Stojanović et al. // *Proceedings of the 6th International Conference on Geographical Information Systems Theory, Applications and Management*. – 2020. – Pp. 191-198. – doi:10.5220/0009396501910198.

251. Stone, M. Cross-Validatory Choice and Assessment of Statistical Predictions / M.

- Stone // *Journal of the Royal Statistical Society. Series B (Methodological)*. – 1974. – V. 36(2). – Pp. 111-147.
252. Su, J. One Pixel Attack for Fooling Deep Neural Networks / J. Su, D. V. Vargas, K. Sakurai // *IEEE Transactions on Evolutionary Computation*. – 2019. – V. 23(5). – Pp. 828-841.
253. Szendrei, R. Texture based recognition of topographic map symbols / R. Szendrei, I. Elek, I. Fekete // *Proceedings of the 2009 International Conference on Artificial Intelligence and Pattern Recognition*. – 2009. – Pp. 7-10.
254. Szendrey, R. A knowledge-based approach to raster-vector conversion of large scale topographic maps / R. Szendrey, I. Elek, M. Marton // *Acta Cybernetica*. – 2011. – V. 20(1). – Pp. 145-165.
255. Tesseract documentation: Improving the quality of the output. – URL: <https://tesseract-ocr.github.io/tessdoc/ImproveQuality.html>
256. Trier, Ø. D. Feature extraction methods for character recognition / Ø. D. Trier, A. K. Jain, T. Taxt // *Pattern Recognition*. – 1996. – V. 29(4). – Pp. 641-662.
257. Uhl, J. H. Automated Extraction of Human Settlement Patterns From Historical Topographic Map Series Using Weakly Supervised Convolutional Neural Networks / J. H. Uhl et al. // *Access IEEE*. – 2020. – V. 8. – Pp. 6978-6996.
258. Uhl, J. H. Extracting Human Settlement Footprint from Historical Topographic Map Series Using Context-Based Machine Learning / J. H. Uhl et al. // *8th International Conference of Pattern Recognition Systems*. – 2017. – Pp. 1-6.
259. Vasin, Yu. G. An Intelligent Information Technology for Symbol-Extraction from Weakly Formalized Graphic Documents / Yu. G. Vasin, D. Yu. Vasin // *Pattern Recognition and Image Analysis. Advances in Mathematical Theory and Applications*. – 2019. – Vol. 29. – No. 1. – P. 51-57. – doi: 10.1134/S1054661818040284.
260. Vatseva, R. Forest Fragmentation Mapping – Case Studies in Bulgaria and Slovakia (Chapter 11) / R. Vatseva et al. // Car, A., G. Griesebner and J. Strobl (eds.), 2009. – Pp. 210-213.
261. Wang, J. A Novel Genetic Programming Algorithm for Designing Morphological Image Analysis Method / J. Wang, Y. Tan // *Advances in Swarm Intelligence, ICSI 2011*. – V. 6728. – Pp. 549-558. – doi:10.1007/978-3-642-21515-5\_65.
262. Wang, Y.-S. A new image threshold segmentation based on fuzzy entropy and improved intelligent optimization algorithm / Y.-S. Wang // *Journal of Multimedia*. – 2014. – V. 9(4). – Pp. 499-505.
263. Wang, F. Broken Contour Line' Reconstruction Based on Compound Iterative Strategy / F. Wang, P. Liu, Y. Yang // *Wuhan Daxue Xuebao (Xinxi Kexue Ban) / Geomatics and Information Science of Wuhan University*. – 2017. – V. 42(5). – Pp. 683-690. – doi:10.13203/j.whugis20150054.

264. Wei, H. DEANet: Dual Encoder with Attention Network for Semantic Segmentation of Remote Sensing Imagery // *Remote Sens.* – 2021. – № 13:3900. – doi:10.3390/rs13193900
265. Weinman, J. J. Scene text recognition using similarity and a lexicon with sparse belief propagation / J. J. Weinman, E. G. Learned-Miller, A. R. Hanson // *PAMI.* – 2009. – V. 31(10). – Pp. 1733-1746.
266. Xin, D. Contour Line Extraction from Paper-based Topographic Maps / D. Xin, X. Zhou, H. Zheng // *Journal of Information and Computing Science.* – 2006. – V. 1. – No. 5. – Pp. 275-283.
267. Xu, B. Identification of Contour Lines from Average-Quality Scanned Topographic Maps / B. Xu, J. Chen, M. Yao // *Mathematical Problems in Engineering.* – 2016. – V. 2. – Pp. 1-14. – doi:10.1155/2016/3089690.
268. Xu, P. Graphic-based character grouping in topographic maps / P. Xu et al. // *Neurocomputing.* – 2016. – V. 189. – Pp. 160-170. – doi:10.1016/j.neucom.2015.12.094.
269. Xu, C. Gradient vector flow: A new external force for snakes / C. Xu, J. L. Prince // *Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* – 1997. – Pp. 66-71.
270. Xydas, C. Buildings Extraction from Historical Topographic Maps via a Deep Convolution Neural Network / C. Xydas et. al // *17th International Conference on Computer Vision Theory and Applications, VISIGRAPP 2022.* – 2022. – V. 5. – Pp. 485-492. – doi:10.5220/0010839700003124.
271. Yan, X. A graph convolutional neural network for classification of building patterns using spatial vector data / X. Yan et al. // *ISPRS Journal of Photogrammetry and Remote Sensing.* – 2019. – V. 150. – Pp. 259-273. – doi:10.1016/j.isprsjprs.2019.02.010.
272. Yu, E.-S. Features recognition from piping and instrumentation diagrams in image format using a deep learning network / E.-S. Yu et al. // *Energies.* – 2019. – V. 12(23). – No. 4425. – 19 p. – doi:10.3390/en12234425.
273. Zhang, H. Image segmentation evaluation: A survey of unsupervised methods / H. Zhang, J. E. Fritts, S. A. Goldman // *Computer Vision and Image Understanding.* – 2008. – V. 110(2). – Pp. 260-280.
274. Zhang, Zh. MKANet: An Efficient Network with Sobel Boundary Loss for Land-Cover Classification of Satellite Remote Sensing Imagery // *Remote Sens.* – 2022. – №. 18:4514. – doi:10.3390/rs14184514
275. Zhu, S. Equivalent circuit model recognition of electrochemical impedance spectroscopy via machine learning / S. Zhu et al. // *Journal of Electroanalytical Chemistry.* – 2019. – V. 855(15). – Pp. 113627. – doi:10.1016/j.jelechem.2019.113627.



## Приложение А. Дополнительный материал

Таблица А.1 – Алгоритм нахождения информации о соседних пикселях

<p><b>Входные параметры:</b>  растр: <math>W \times H \times D</math>, <i>uint8</i>;  массив точек, центров окон: <math>P \times 2</math>, <i>int</i>;  размер окна: <math>w_x, w_y</math>, <i>int</i>;  угол наклона окна: <math>\phi, \theta, P \times 1</math>, <i>double</i>.</p>
<p><b>Выходные параметры:</b>  информация об окрестности точек <math>(w_x \cdot w_y \cdot D) \times P</math>, <i>uint8</i>.</p>
<p><b>Особенность.</b> Для ускорения работы преобразование поворота происходит с округлением до ближайших пикселей, алгоритм оптимизирован под использование матричных операций MATLAB и вариаций входных параметров (полутоновое или цветное изображение, способ задания углов наклона).</p>
<p><b>Показатели качества:</b> время работы.</p>
<p><b>Оценка времени:</b> <math>\Theta(P \cdot w_x \cdot w_y)</math>.</p>

Таблица А.2 – Алгоритм многометочной сегментации

<p><b>Входные параметры:</b>  растр: <math>W \times H \times D</math>, <i>uint8</i>;  интеллектуальная модель для сегментации: ANN, k-NN, Classification Tree, SVM с возможностью определения <math>K</math> классов сегментов.</p>
<p><b>Выходные параметры:</b>  растры мер классификации: <math>W \times H \times K</math>, <i>uint8</i>.</p>
<p><b>Особенность.</b> Использование паноптического подхода сегментации с сохранением информации о мере принадлежности пикселя рассматриваемым классам <math>K</math> вместо строгой попиксельной классификации; для уменьшения использованной памяти используется пакетная обработка.</p>
<p><b>Показатели качества:</b> macro-averaged <math>F_1</math>-score, время обучения, время работы (среднее и стандартное отклонение).</p>
<p><b>Оценка времени:</b>  обучение моделей зависит от типа модели, для ANN:  <math>O(P \cdot n_{\text{эпох}} \cdot n_{\text{hidden layers}} \cdot (w_x \cdot w_y + K) + W \cdot H \cdot K)</math>;  запуск алгоритма зависит от типа модели, для ANN:  <math>\Theta(W \cdot H \cdot n_{\text{hidden layers}} \cdot (w_x \cdot w_y + K))</math>.</p>

Таблица А.3 – Генетический алгоритм получения последовательности операций для локализации

<p><b>Входные параметры:</b>  набор полутоновых растров: <math>P, i \in \{1, \dots, N\}, W_i \times H_i, uint8</math>;  набор бинарных размеченных растров идеальной и критичной локализации: <math>P_{ideal}, P_{crit}, i \in \{1, \dots, N\}, W_i \times H_i, logical</math>;  параметры генетического алгоритма: размер популяции, доля элитной популяции, вероятности генетических операторов, жёсткость расчёта функции приспособленности.</p>
<p><b>Выходные параметры:</b>  хромосома, состоящая из <math>N_{genes}</math> последовательных морфологических и фильтрующих операций: <math>chromosome, N_{genes} \times struct(name, property, value)</math>.</p>
<p><b>Особенность.</b> Используются хромосомы переменной длины. В качестве операций исследовались: морфологические операции, бинаризация, фильтрация по параметрам сегментов.</p>
<p><b>Показатели качества:</b> количество локализованных пикселей, количество пропущенных объектов, время работы.</p>
<p><b>Оценка времени:</b> <math>\Theta(W \cdot H \cdot N_{genes}^2)</math>.</p>

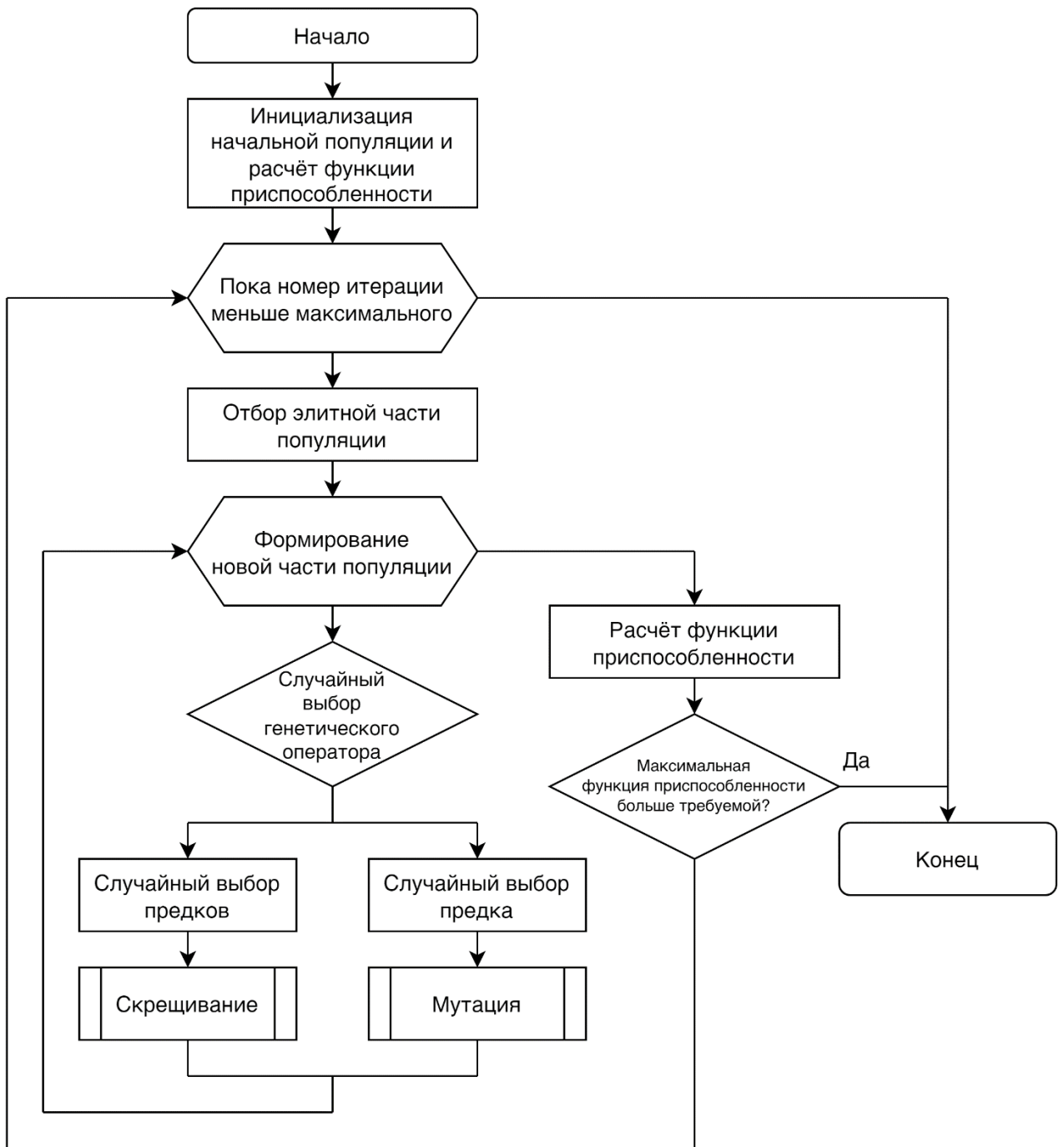


Рис. А.1 – Схема генетического алгоритма получения последовательности операций для локализации

Таблица А.4 – Двухкаскадный алгоритм распознавания точечных объектов

<p><b>Входные параметры:</b>  полутонный растр, полученный после этапа сегментации: <math>W \times H</math>, <i>uint8</i>;  бинарный растр, полученный после этапа локализации: <math>W \times H</math>, <i>logical</i> с <math>P</math> точками;  размер окна: <math>w_x, w_y</math>, <i>int</i>;  интеллектуальные модели для распознавания: классификатор надкласса, классификатор первого каскада, классификатор второго каскада.</p>
<p><b>Выходные параметры:</b>  координаты и классы найденных образов;  индексированный растр суперсегментов.</p>
<p><b>Особенность.</b> Классификатор надкласса и интеллектуальная модель первого каскада запускаются на скользящих окнах. Выход первого каскада представляет собой набор суперсегментов.  <i>Суперсегмент</i> – непрерывное множество пикселей на индексированном изображении, где значение цвета пикселя определяется индексом класса.  Интеллектуальная модель второго каскада запускается на признаках, являющихся долями исходных классов внутри суперсегмента, что позволяет абстрагироваться от размера и угла поворота образов.</p>
<p><b>Показатели качества:</b>  для классификатора надкласса: <i>recall</i> и <i>precision</i>;  для первого каскада: <i>micro-averaged recall</i> и <i>precision</i>;  для второго каскада: <i>micro-averaged recall</i>, <i>common precision</i> и <i>identity precision</i>;  время обучения моделей;  время работы.</p>
<p><b>Оценка времени:</b> <math>\Theta(P \cdot w_x \cdot w_y + W \cdot H \cdot K)</math>.</p>

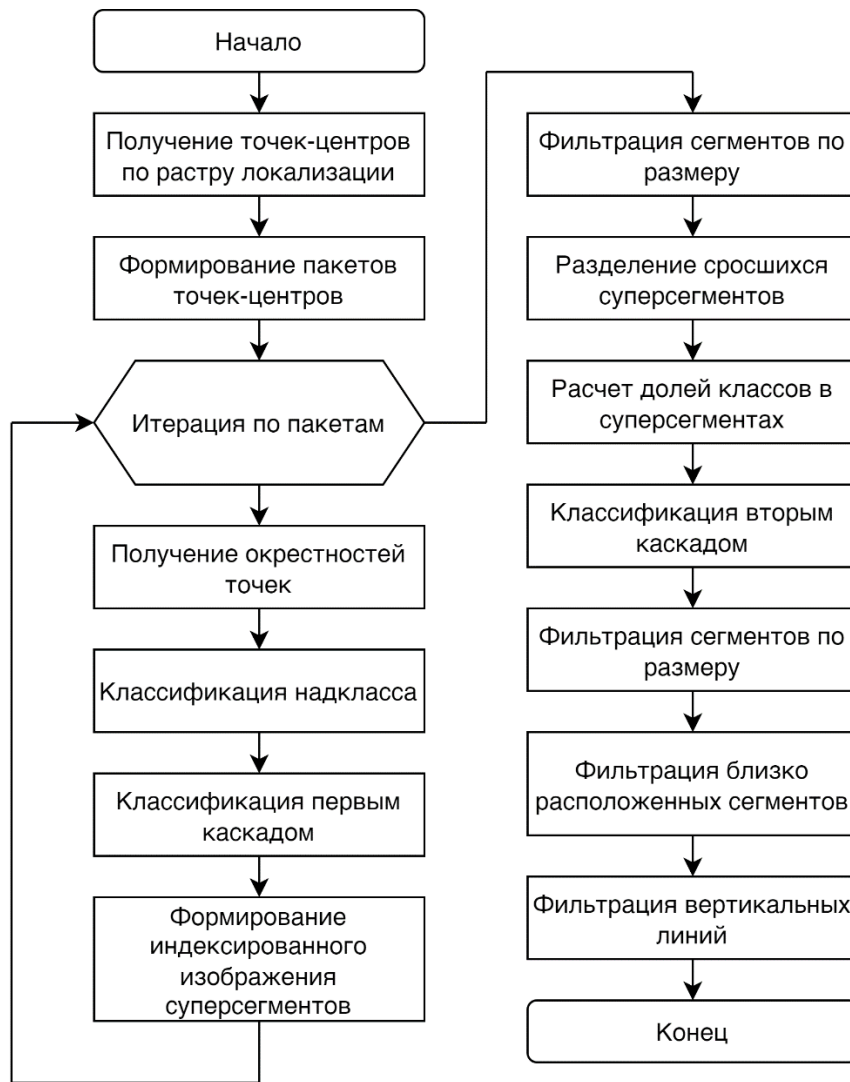


Рис. А.2 – Схема двухкаскадного алгоритма распознавания точечных объектов

Таблица А.5 – Алгоритм группировки точечных объектов

<p><b>Входные параметры:</b>          координаты объектов: <math>P \times 2, int</math>;          класс объектов: <math>P \times 1, int</math>;          параметры группировки: максимальное расстояние между объектами, количество карманов <math>N_{bucket}</math>, параметры фильтров.</p>
<p><b>Выходные параметры:</b>          координаты и значения найденных сгруппированных объектов, указатели на составляющие простые объекты: <math>struct(ind, xx, yy, value)</math>.</p>
<p><b>Особенность.</b> Для группировки используется метод геометрического хеширования и выполнения фильтрующих операций (геометрический и семантический анализ).</p>
<p><b>Показатели качества:</b> micro-averaged recall, common precision и identity precision, время работы.</p>
<p><b>Оценка времени:</b> <math>\Theta(P^2/N_{bucket})</math>.</p>

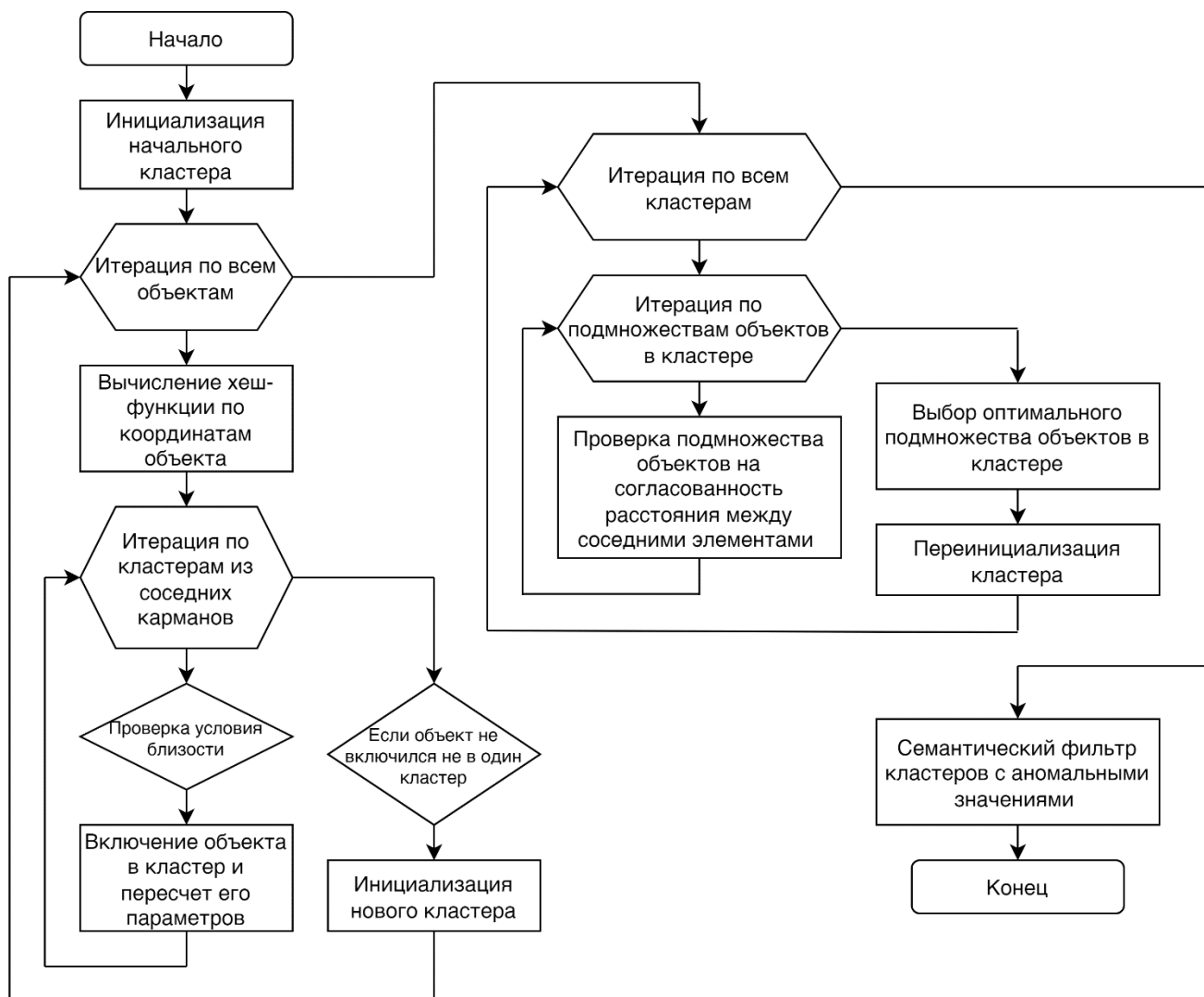


Рис. А.3 – Схема алгоритма группировки точечных объектов

Таблица А.6 – Алгоритм классификации площадных объектов

<p><b>Входные параметры:</b>  растр: <math>W \times H \times D</math>, <i>uint8</i>;  интеллектуальная модель для сегментации <math>K</math> классов;  параметры фильтрации: минимальный размер сегмента, размер окна <math>w</math>, минимальная доля класса в окне.</p>
<p><b>Выходные параметры:</b>  индексированный растр: <math>W \times H</math>, <i>int</i>.</p>
<p><b>Особенность.</b> В качестве первоначального признака для сегментации используются цветовые гистограммы в скользящем окне при фильтре Гаусса размера <math>G</math>. Далее используется функция <math>f_{local}</math>. Для фильтрации в скользящем окне строится гистограмма классов и происходит анализ мод.</p>
<p><b>Показатели качества:</b> macro-averaged IoU.</p>
<p><b>Оценка времени:</b> отличается для различных интеллектуальных моделей, для ANN: <math>O(W \cdot H \cdot (G \cdot N_{recon} + w^2 + N_{bins} \cdot K) + S' \sqrt{S'} \cdot (w^2 + K))</math>, где <math>S'</math> – количество неклассифицированных пикселей после первого этапа алгоритма, <math>N_{recon}</math> – количество итераций фильтрации Гаусса.</p>

Таблица А.7 – Алгоритм группировки линейных объектов

<p><b>Входные параметры:</b>  бинарный растр линейных объектов: <math>W \times H</math>, <i>logical</i>;  массив концевых точек: <math>EP</math>, <i>struct(xx, yy, ind)</i>;  матрица расстояний между концевыми точками: <math>EP \times EP</math>, <i>single</i>;  параметры условий соединения концевых точек: минимальные и максимальные расстояния, углы рассогласования.</p>
<p><b>Выходные параметры:</b>  бинарный растр линейных объектов: <math>W \times H</math>, <i>logical</i>.</p>
<p><b>Особенность.</b> Для проверки возможности соединения используется 5 условий: локальности, согласования углов касательных, согласования касательной с углом наклона линии соединения, доминирования, максимального правдоподобия.</p>
<p><b>Показатели качества:</b> количество возможных соединений, количество ложных соединений, время работы.</p>
<p><b>Оценка времени:</b> <math>\Theta(EP^2)</math>.</p>

Таблица А.8 – Алгоритм комплексного анализа ЦТК

<p><b>Входные параметры:</b>  растр горизонталей: <math>W \times H</math>, <i>logical</i>;  растр полугоризонталей: <math>W \times H</math>, <i>logical</i>;  растр бергштрихов: <math>W \times H</math>, <i>logical</i>;  массив распознанных подписей горизонталей: <math>P</math>, <i>struct</i>;  массив распознанных отметок высот: <math>P</math>, <i>struct</i>.</p>
<p><b>Выходные параметры:</b>  восстановленный план горизонталей: <math>W \times H</math>, <i>single</i></p>
<p><b>Особенность.</b> Представление плана горизонталей в виде орграфа, в котором вершинами являются значения высот горизонталей, рёбрами – возможность перехода с одной горизонтали на другую без пересечения других горизонталей. При данном подходе происходит абстрагирование от отношения внутри/снаружи с помощью детерминированного выбора полупространства, ограниченного кривой и границами карты. Алгоритм корректирует неправильно распознанные значения с прошлых этапов обработки.</p>
<p><b>Показатели качества:</b> recall, precision, время работы.</p>
<p><b>Оценка времени:</b>  в среднем: <math>O(N_{\text{гор}} \cdot W \cdot H)</math>, где <math>N_{\text{гор}}</math> – число горизонталей  в худшем: <math>O(N_{\text{гор}}^2 \cdot W \cdot H)</math>.</p>

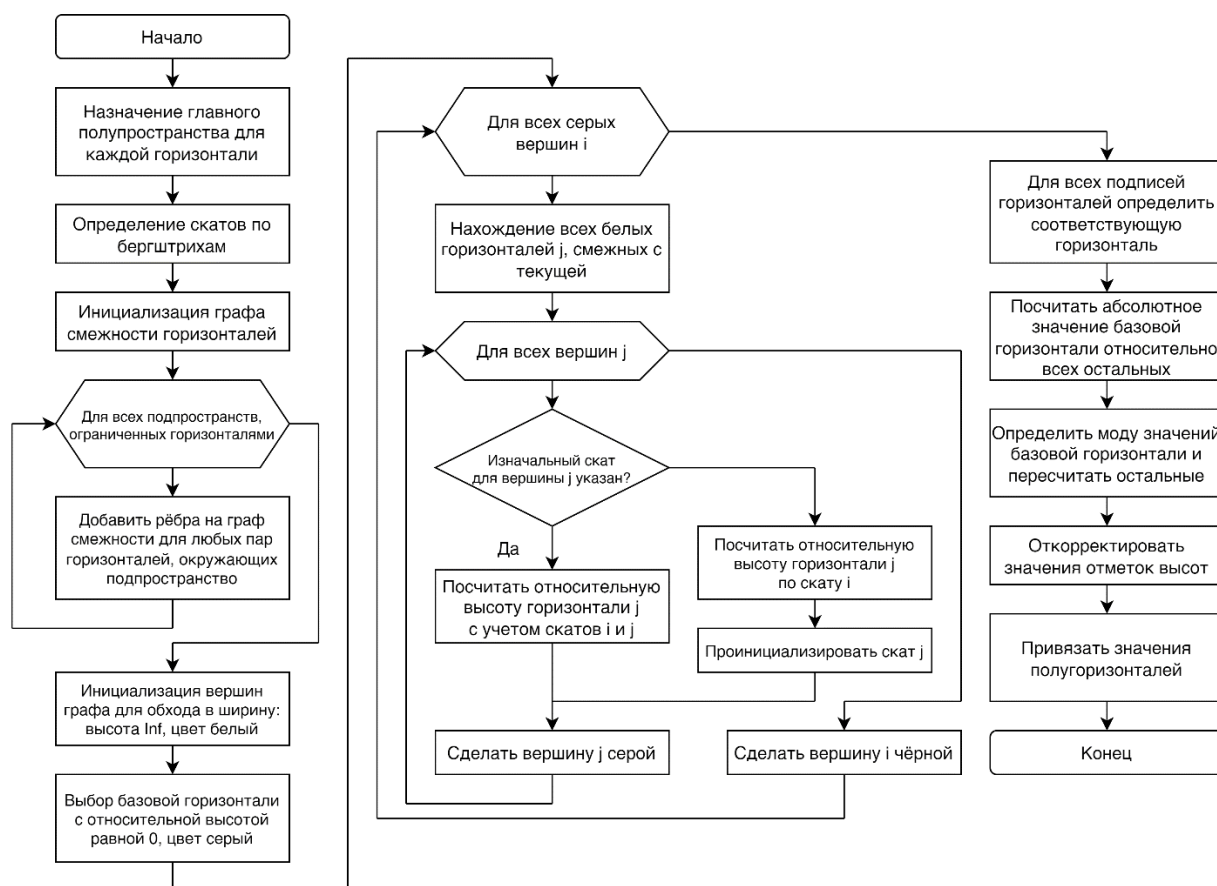


Рис. А.4 – Схема алгоритм комплексного анализа ЦТК



Таблица А.9 – Алгоритм интерполяции, учитывающий специфику ЦТК

<p><b>Входные параметры:</b>          план <math>N</math> горизонталей: <math>W \times H</math>, <i>single</i> из <math>P</math> точек;          параметры интерполяции: доля используемых точек <math>\omega</math>, количество существующих точек для расчета текущей <math>nP</math></p>
<p><b>Выходные параметры:</b>          цифровая модель местности: <math>W \times H</math>, <i>single</i></p>
<p><b>Особенность.</b> Все точки карты разбиваются на три типа: находящиеся между двух горизонталей, расположенные внутри замкнутой горизонтали или между границей карты и ближайшей горизонталью. Точки второго и третьего типа рассчитываются с учётом информации о градиенте, полученной от точек первого типа, формируя локальный экстремум либо внутри горизонтали, либо снаружи соответственно.</p>
<p><b>Показатели качества:</b> максимальное абсолютное отклонение, стандартное отклонение, время работы</p>
<p><b>Оценка времени:</b> <math>\Theta(W \cdot H \cdot P/N \cdot \omega \cdot nP)</math>.</p>

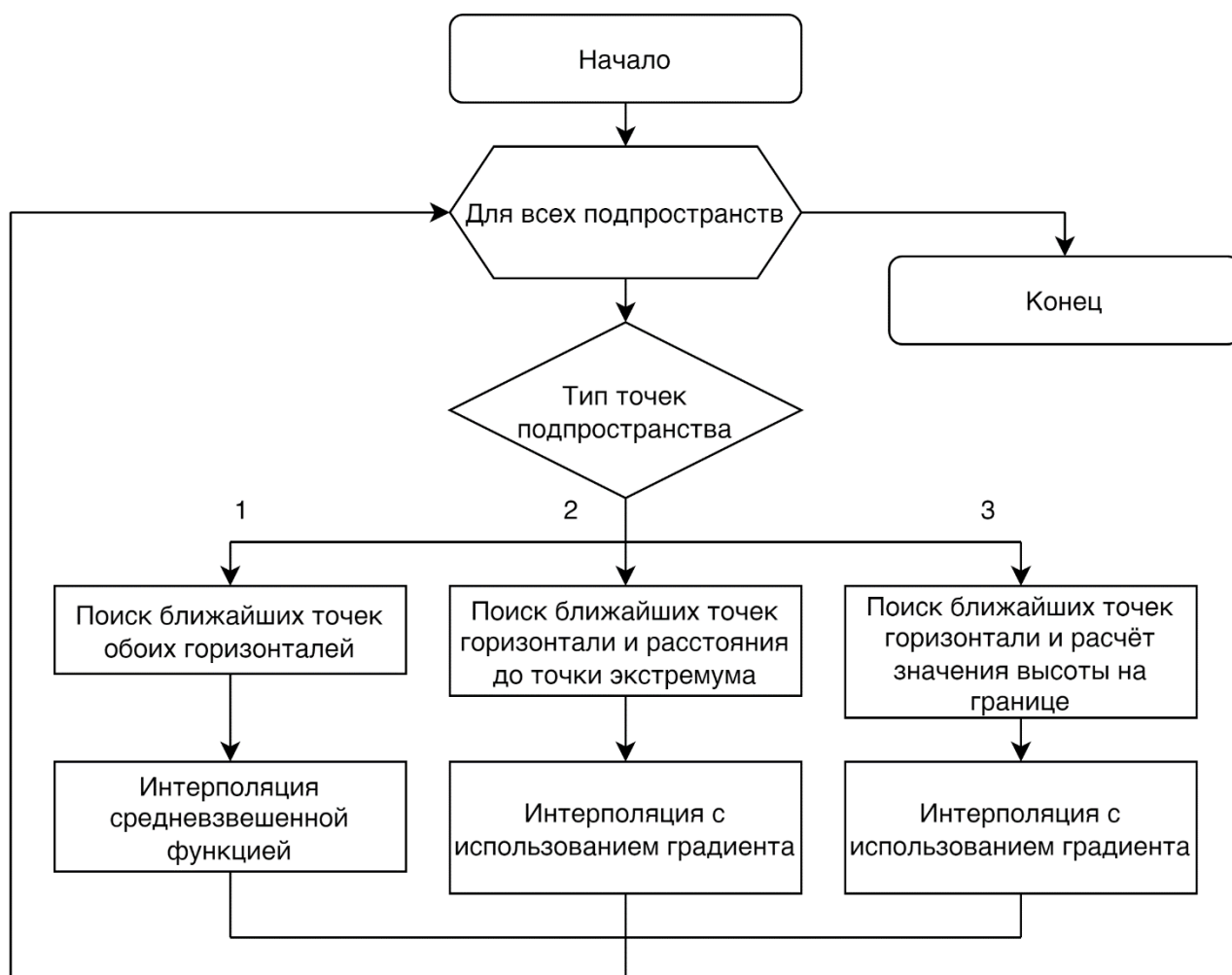
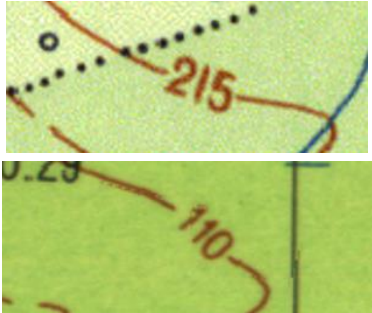
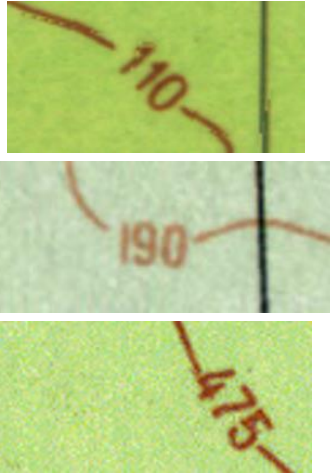

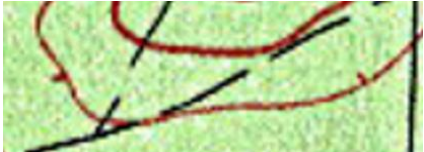
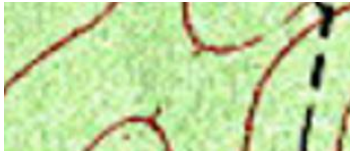
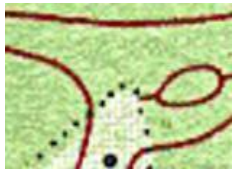
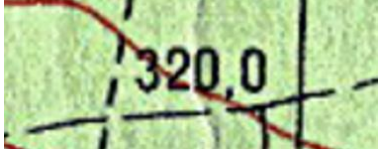


Рис. А.5 – Схема алгоритма интерполяции, учитывающего специфику ЦТК

Таблица А.10 – Примеры участков используемых ЦТК


Таблица А.11 – Примеры нарушений явных и неявных правил изображения ЦТК

Описание нарушения	Пример изображения с нарушением
<b>1. Нарушения правил изображения условных знаков</b>	
Использование различных шрифтов для одного типа элементов	
Непостоянство позиционирования подписей горизонталей внутри разрыва для них	
Наличие разрывов в горизонталях, не являющихся полугоризонталями	
Наличие противоположных бергштрихов на одной горизонтали	
Непостоянство длины бергштриха, нечеткость изображения образов	
Отсутствие необходимых для расшифровки значений высот бергштрихов на седловинах	
Отсутствие точки позиционирования отметок высот	

2. Искажения, вызванные физическим нарушением оригинальной ТК при производстве и в результате использования	
Искажения цветовой гаммы карты в связи с длительным хранением	
Различие в используемых красках на различных картах	
Наличие пятен краски и загрязнений	
Наличие помятостей и прочих деформаций (изменение геометрии и цветности объектов при сохранении топологии)	
Наличие разрывов (нарушение топологии и потеря информации)	
Непостоянство правила о пересечении подписей и линий	

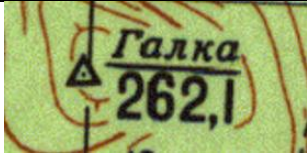


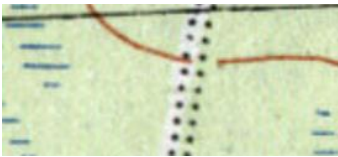
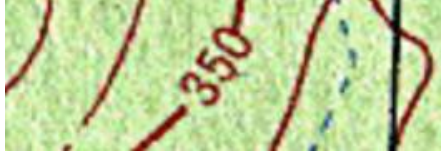

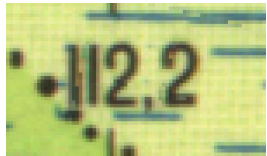
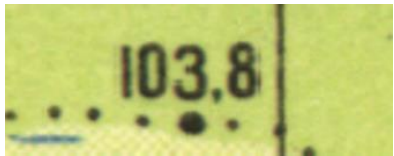
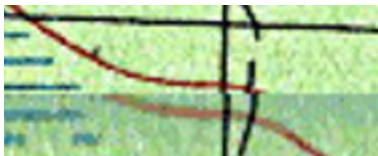
	 
Слияние горизонталей при плотной печати	
Плохое позиционирование печатающих головок	
<b>3. Нарушения при сканировании и обработке после сканирования</b>	
Засвет при сканировании светлых частей карты	 
Цветовой сдвиг: правый и левый края имеют красноватый или синеватый оттенок, локальная размытость изображений	
Рассинхронизация при сканировании	
Некорректное соединение двух изображений, полученных после сканирования	

Таблица А.12 – Список МО для локализации, полученных при реализации ГА

Операция	Структурный элемент (при наличии)	Размер структурного эле- мента (параметр операции)
Локализация пикселей I типа, строгий расчёт функции приспособленности		
закрытие	ромб (♦)	9×9
открытие	квадрат	2×2
закрытие	вертикальная линия	2
открытие	вертикальная линия	3
эрозия	вертикальная линия	5
эрозия	вертикальная линия	2
бинаризация	–	0,4060
эрозия	вертикальная линия	3
закрытие	горизонтальная линия	5
открытие	вертикальная линия	4
Локализация пикселей I типа, мягкий расчёт функции приспособленности		
закрытие	вертикальная линия	3
открытие	вертикальная линия	5
открытие	вертикальная линия	5
закрытие	ромб	13×13
открытие	горизонтальная линия	5
открытие	квадрат	2×2
закрытие	вертикальная линия	5
закрытие	горизонтальная линия	6
бинаризация	–	0,4162
эрозия	горизонтальная линия	2
эрозия	вертикальная линия	6
Локализация пикселей II типа, строгий расчёт функции приспособленности		
закрытие	ромб	7×7
закрытие	ромб	7×7
закрытие	квадрат	3×3
закрытие	вертикальная линия	6
открытие	квадрат	6×6
закрытие	вертикальная линия	5
открытие	квадрат	5×5
бинаризация	–	0,4470
открытие	ромб	7×7
эрозия	вертикальная линия	2
Локализация пикселей II типа, мягкий расчёт функции приспособленности		
закрытие	ромб	5×5
открытие	горизонтальная линия	3
закрытие	квадрат	4×4
эрозия	квадрат	6×6
закрытие	круг	11×11
бинаризация	–	0,3816
фильтрация по меньшей оси	–	(8, 25)
закрытие	ромб	9×9
дилатация	квадрат	5×5
фильтрация по площади	–	[3, 659]

Таблица А.13 – Алгоритм работы для дата сайентиста

№	Операция	Используемые модули
0	Подготовка изображений ЦТК: обрезка краёв и масштабирование для приведения к одному <i>dpi</i>	<i>resizeMaps.m</i>
1.1	Получение тренировочного и тестового датасета для сегментации	<i>getTrainDataSeg.m</i> <i>getNbh.m</i> <i>extractLayers.m</i>
1.2	Обучение стабильных и нейросетевых ИМ на тренировочных данных для сегментации	<i>getStableModelsSeg.m</i> <i>getANNModelsSeg.m</i>
1.3	Определение показателей качества и быстродействия сегментации на тестовых данных	<i>testingSeg.m</i> <i>segmentImage.m</i>
2.1	Получение тренировочного и тестового датасета для локализации отметок высот и подписей горизонталей	<i>getTrainDataLoc.m</i>
2.2	Запуск генетического алгоритма на тренировочных данных для получения оптимальной последовательности операций локализации	<i>getChromosome.m</i> <i>geneticAlgorithm.m</i> <i>fitnessFunction.m</i> <i>crossover.m</i> <i>assignChromosome.m</i> <i>mutation.m</i> <i>clearChromosome.m</i> <i>applyChromosome.m</i> <i>findCentromere.m</i>
2.3	Определение показателей качества и быстродействия локализации на тестовых данных	<i>testingLoc.m</i>
3.1	Получение тренировочного и тестового датасета для распознавания отметок высот и подписей горизонталей для первого каскада распознавания	<i>extractDigits.m</i> <i>getTrainDataIsDigit.m</i> <i>getTrainDataIsDigitB.m</i> <i>getTrainDataRec1.m</i> <i>getTrainDataRecB1.m</i> <i>mergeTrainDataRec.m</i>
3.2	Обучение стабильных и нейросетевых ИМ на тренировочных данных для первого каскада	<i>getStableModelsRec1.m</i> <i>getANNModelsRec1.m</i> <i>getANNModelsRecB.m</i>
3.3	Определение показателей качества и быстродействия первого каскада распознавания на тестовых данных	<i>testingRec1.m</i> <i>testingRecB1.m</i> <i>f1score.m</i> <i>discreteTangent.m</i>
3.4	Получение тренировочного и тестового датасета для распознавания отметок высот и подписей горизонталей для второго каскада распознавания	<i>getTrainDataRec2.m</i> <i>getTrainDataRecB2.m</i>
3.5	Обучение стабильных и нейросетевых ИМ на тренировочных данных для второго каскада	<i>getStableModelsRec2.m</i> <i>getANNModelsRec2.m</i>
3.6	Определение показателей качества и быстродействия второго каскада распознавания на тестовых данных	<i>recognition.m</i> <i>recognitionB.m</i> <i>testingRec2.m</i> <i>testingRecB2.m</i> <i>getPropFromMap.m</i>
3.7	Получение расстояний между цифрами	<i>getRo.m</i> <i>getRoB.m</i>

3.8	Кластеризация цифр в числа, геометрическая и семантическая фильтрация чисел	<i>clusterDigit.m</i> <i>clusterDigitB.m</i> <i>clusterFilter.m</i> <i>clusterFilterB.m</i> <i>groupDigitB.m</i> <i>semanticFilter.m</i>
3.9	Определение показателей качества и быстродействия группировки чисел на тестовых данных	<i>reorderCluster.m</i> <i>testingGroupDigits.m</i> <i>testingGroupDigitsB.m</i>
4.1	Получение элементов ЛО: горизонталей, полугоризонталей, бергштрихов	<i>getContours.m</i> <i>getHalfContours.m</i> <i>getBergstrichs.m</i> <i>testingBergstrichs.m</i>
4.2	Получение концевых точек линий и дискретных касательных к ним	<i>getEndpoints.m</i> <i>discreteTangent.m</i>
4.3	Получение параметров геометрических условий I-III	<i>geometricGrouping.m</i> <i>drawLines.m</i>
4.4	Получение параметров условия доминирования IV	<i>relativeGrouping.m</i>
4.5	Получение всех возможных пар концевых точек и их кластеризация	<i>getPairs.m</i> <i>clusterLines.m</i>
4.6	Получение параметров условия по единственному соединению	<i>oneInCLusterGrouping.m</i> <i>checkLimits.m</i>
4.7	Получение параметров условия максимального правдоподобия V	<i>maxLikelihoodGrouping.m</i>
4.8	Определение оптимальной итеративной последовательности проверок условия I-V и их параметров, определение показателей качества и быстродействия	<i>testingContourRecon.m</i>
5.1	Определение оптимальных параметров фильтрации площадных элементов и тестирование распознавания ПЛО	<i>arealRecon.m</i> <i>seqFilter.m</i> <i>modeFilter.m</i>
6.1	Генерация ЦМР, получение плана горизонталей, точек в локальных экстремумах и скатов для тестирования алгоритма комплексного анализа	<i>genDEM.m</i> <i>genDEMS.m</i> <i>getContour.m</i> <i>getExtremumMap.m</i>
6.2	Получение параметров и проверка алгоритма комплексного анализа на сгенерированных данных со внесёнными ошибками распознавания и потерей данных	<i>getHalfplain.m</i> <i>corruptElevation.m</i> <i>trainComplex.m</i>
6.3	Проверка алгоритма комплексного анализа для реальных ЦТК, уточнение положения отметок высот	<i>numberXYClarification.m</i> <i>testingComplex.m</i>
7.1	Определение параметров разработанного алгоритма интерполяции данных, собственных ЦТК, и определение показателей качества и быстродействия	<i>formalInterpolation.m</i> <i>testingInterpolation.m</i>



Таблица А.14 – Алгоритм работы для конечного пользователя: звёздочкой отмечены этапы с частично ручным исполнением в сторонних приложениях

№	Операция	Используемые модули
0	Подготовка изображений ЦТК: обрезка краёв и масштабирование для приведения к одному <i>dpi</i>	<i>resizeMaps.m</i>
1	Сегментация изображения	<i>segmentImage.m</i> <i>getNbh.m</i> <i>extractLayers.m</i> <i>getSegQuality.m</i>
2	Локализация изображения	<i>localization.m</i> <i>applyChromosome.m</i>
3.1	Распознавание ТО цифр	<i>recognition.m</i> <i>recognitionB.m</i>
3.2	Распознавание ТО чисел	<i>clusterDigit.m</i> <i>clusterDigitB.m</i> <i>clusterFilter.m</i> <i>clusterFilterB.m</i> <i>groupDigitB.m</i> <i>semanticFilter.m</i>
4.1	Получение элементов ЛО: горизонталей, полу-горизонталей, бергштрихов*	<i>getContours.m</i> <i>getHalfContours.m</i> <i>getBergstrichs.m</i> <i>ручной поиск бергштрихов</i>
4.2	Получение концевых точек линий и дискретных касательных к ним	<i>getEndPoints.m</i> <i>discreteTangent.m</i>
4.3	Реализация возможных условий соединения линий*	<i>geometricGrouping.m</i> <i>relativeGrouping.m</i> <i>getPairs.m</i> <i>clusterLines.m</i> <i>oneInClusterGrouping.m</i> <i>maxLikelihoodGrouping.m</i> <i>drawLines.m</i> <i>ручное соединение линий</i>
5	Распознавание Плю	<i>arealRecon.m</i> <i>seqFilter.m</i> <i>modeFilter.m</i>
6	Алгоритм комплексного анализа	<i>getHalfplain.m</i> <i>numberXYClarification.m</i> <i>testingComplex.m</i>
7	Интерполяция и получение ЦМР	<i>interpolation.m</i>

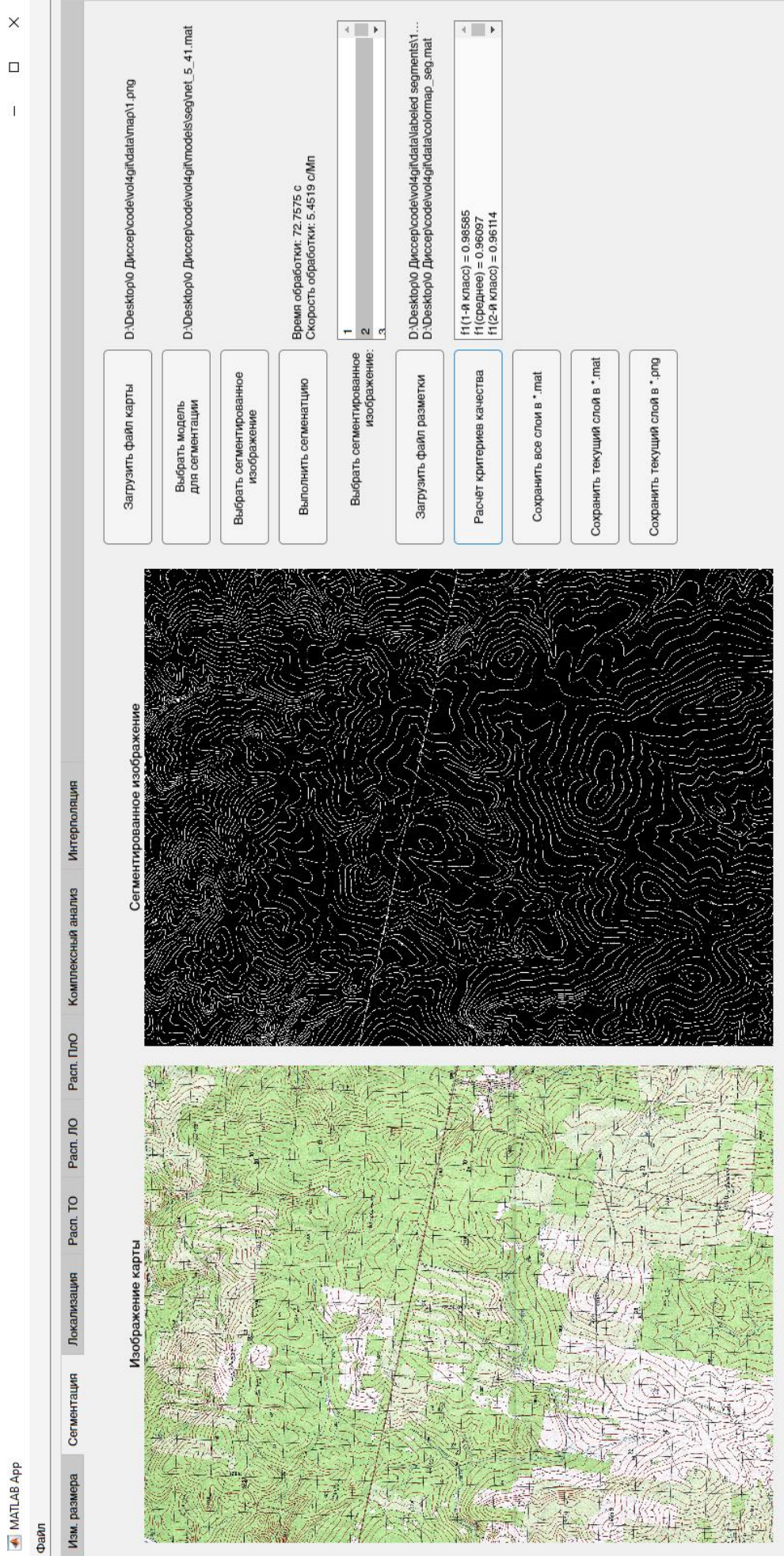


Рисунок А.6 – Скриншот ПО ПТК

# Приложение Б. Свидетельства о регистрации программ для ЭВМ

РОССИЙСКАЯ ФЕДЕРАЦИЯ



**СВИДЕТЕЛЬСТВО**

о государственной регистрации программы для ЭВМ

**№ 2016611844**

**Построение объемной модели местности по топографической карте**

Правообладатель: *Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Уральский государственный университет путей сообщения» (УрГУПС) (RU)*

Авторы: *Тарасян Владимир Сергеевич (RU),  
Дмитриев Никита Владимирович (RU)*

Заявка № **2015662517**  
 Дата поступления **18 декабря 2015 г.**  
 Дата государственной регистрации  
 в Реестре программ для ЭВМ **11 февраля 2016 г.**

Руководитель Федеральной службы  
по интеллектуальной собственности

 **Г.П. Ивлиев**



РОССИЙСКАЯ ФЕДЕРАЦИЯ



## СВИДЕТЕЛЬСТВО

о государственной регистрации программы для ЭВМ

**№ 2019665260**

**Программный комплекс преобразования изображений  
топографических карт в цифровую модель рельефа**

Правообладатель: **Федеральное государственное бюджетное  
образовательное учреждение высшего образования «Уральский  
государственный университет путей сообщения» (RU)**

Автор: **Дмитриев Никита Владимирович (RU)**

Заявка № **2019664218**

Дата поступления **11 ноября 2019 г.**

Дата государственной регистрации

в Реестре программ для ЭВМ **21 ноября 2019 г.**

Руководитель Федеральной службы  
по интеллектуальной собственности

 **Г.П. Ивлиев**



## Приложение В. Копии актов о внедрении

«УТВЕРЖДАЮ»



Директор АО «Урало-Сибирская  
Гео-Информационная компания»

А.А. Алябьев

23 декабря 2019 г.

### АКТ

о внедрении результатов диссертационной работы  
Дмитриева Никиты Владимировича  
«Повышение точности и производительности алгоритмов анализа и обработки  
сложноструктурных изображений»

Экспертная комиссия в составе: директора Алябьева А. А., главного инженера Кобзевой Е.А., заместителя директора по инновационным вопросам Струниной Е.Н. составили настоящий Акт о том, что практические и теоретические результаты диссертационной работы Н. В. Дмитриева прошли апробацию и внедрены в работе АО «УСГИК» при осуществлении деятельности предприятия в области геодезической и картографической деятельности.

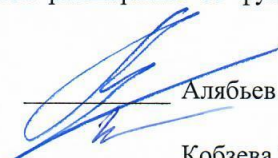


Процесс внедрения происходил 4-29 ноября 2019 года. При этом использованы следующие результаты диссертационной работы Н. В. Дмитриева, представленной на соискание учёной степени кандидата технических наук:

1. Алгоритм и программное обеспечение, выполняющее восстановление плана горизонталей по предоставленным изображениям топографических карт, обладающий по сравнению с аналогичными функциями программы MapInfo лучшим качеством за счёт соединения большего числа разрывов горизонталей, особенно на топографических картах низкого разрешения.
2. Алгоритм и программное обеспечение, выполняющее распознавание отметок высот и подписей горизонталей с точностью не меньшей 75% на представленных топографических картах.
3. Программный комплекс в целом осуществляет построение цифровой модели местности со средней погрешностью 2-3 м, что является достаточным для первичного анализа и принятия решений относительно начальных этапов проектирования.

Результаты диссертационной работы позволяют повысить эффективность работы за счёт быстрого получения цифровой модели местности, сокращения проведения полевых работ, ускорения обработки материалов. По результатам использования программы для ЭВМ «Программный комплекс преобразования изображений топографических карт в цифровую модель рельефа» были выработаны предложения в области расширения её функционала и совершенствования используемых алгоритмов.

Председатель экспертной комиссии

Члены экспертной комиссии

 Алябьев А. А.  
 Кобзева Е.А.  
 Струнина Е.Н.

ФЕДЕРАЛЬНОЕ АГЕНТСТВО  
ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Федеральное государственное бюджетное  
образовательное учреждение  
высшего образования

«Уральский государственный университет  
путей сообщения»

(ФГБОУ ВО УрГУПС)

ул. Колмогорова, д. 66, г. Екатеринбург, 620034

Тел/факс: (343) 221-24-44

E-mail: [rector@usurt.ru](mailto:rector@usurt.ru)

<http://www.usurt.ru>

ОКПО 01116035, ОГРН 1026602950065

ИНН/КПП 6659014366/667801001

УТВЕРЖДАЮ:

Проректор по учебной работе и связям  
с производством ФГБОУ ВО «Уральский  
государственный университет путей  
сообщения»

Сирина Нина Фридриховна

" 01 " декабря 2023 г.

№ \_\_\_\_\_  
на № \_\_\_\_\_ от \_\_\_\_\_

**АКТ**

**об использовании результатов диссертационного исследования**

**Дмитриева Никиты Владимировича**

Мы, нижеподписавшиеся, представители ФГБОУ ВО Уральского государственного университета путей сообщения декан механического факультета (МФ) Андрей Владимирович Архипов и начальник учебно-методического отдела (УМО) Елена Николаевна Морозова составили настоящий акт о том, что результаты диссертационного исследования Дмитриева Н. В. используются при реализации учебного процесса, в том числе:

– алгоритм локализации и распознавания точечных объектов для машинного зрения в рамках дисциплины «Мобильная робототехника» направления подготовки 15.03.06 «Мехатроника и робототехника»;

– алгоритмы сегментации, локализации и распознавания объектов для обработки изображений в рамках дисциплины «Интеллектуальная обработка информации» направления подготовки 09.03.02 «Информационные системы и технологии», направленность «Системы искусственного интеллекта».

Декан МФ

Начальник УМО

А.В. Архипов

Е.Н. Морозова