

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования «Самарский национальный исследовательский
университет имени академика С.П. Королева»
(Самарский университет)

На правах рукописи

Мухин Артем Владимирович

ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА ДЛЯ ХРАНЕНИЯ, ВИЗУАЛИЗАЦИИ
И ОБРАБОТКИ ГИПЕРСПЕКТРАЛЬНЫХ ДАННЫХ

2.3.5. Математическое и программное обеспечение вычислительных систем,
комплексов и компьютерных сетей

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
Куприянов Александр Викторович,
доктор технических наук, доцент

ОГЛАВЛЕНИЕ

Введение.....	5
1 Анализ предметной области и формирование требований к математическому и программному обеспечению обработки гиперспектральных изображений.....	17
1.1 Гиперспектральные изображения, их развитие, особенности и применимость	17
1.2 Программное обеспечение анализа гиперспектральных изображений и формирование требований	24
1.2.1 Критерии сравнения программных систем обработки ГСИ	25
1.2.2 Обзор программных систем анализа гиперспектральных изображений	27
1.2.3 Требования к программному обеспечению.....	39
1.3 Основные задачи анализа гиперспектральных изображений.....	43
1.3.1 Задача попиксельной классификации (сегментации)	43
1.3.2 Задача выбора информативных слоев.....	52
1.3.3 Требования к математическому обеспечению	54
1.4 Выводы и результаты по первому разделу.....	55
2 Разработка архитектуры и расширяемой интеллектуальной системы обработки гиперспектральных изображений.....	57
2.1 Функциональные требования	58
2.2 Нефункциональные требования	71
2.3 Подходы к проектированию архитектур программных систем анализа данных	78
2.4 Архитектура расширяемой интеллектуальной системы обработки гиперспектральных изображений.....	82
2.4.1 Контекстная диаграмма.....	84
2.4.2 Диаграмма контейнеров	85
2.4.3 Диаграмма компонентов	88

2.5	Логическая модель данных интеллектуальной системы	91
2.6	Реализация интеллектуальной системы.....	97
2.6.1	Реализация веб-интерфейса	98
2.6.2	Реализация основных микросервисов системы	99
2.6.3	Реализация вычислительных модулей и программного интерфейса для взаимодействия с ними.....	100
2.6.4	Реализация доступа к серверной части системы	102
2.7	Анализ отказоустойчивости интеллектуальной системы.....	102
2.8	Развертывание системы и окружения	106
2.9	Исследование возможностей разработанной интеллектуальной системы	111
2.9.1	Организация хранения гиперспектральных данных	111
2.9.2	Загрузка гиперспектральных изображений в систему	114
2.9.3	Обработка гиперспектральных изображений	117
2.9.4	Добавление алгоритмов и взаимодействие с ними	122
2.10	Интеллектуальная обработка гиперспектральных изображений.....	127
2.11	Экспериментальное исследование характеристик разработанной интеллектуальной системы	134
2.12	Выводы и результаты по второму разделу	141
3	Разработка алгоритмов и методов хранения, визуализации и обработки гиперспектральных изображений.....	144
3.1	Методика формирования информативного индекса	144
3.2	Алгоритм формирования информативного нормализованного разностного индекса (INDI)	149
3.2.1	Исследование применимости разработанного алгоритма INDI....	152
3.2.2	Результат разработки алгоритма формирования информативного индекса	167
3.3	Алгоритм формирования ансамбля информативных индексов (MINDI)	169

3.3.1 Исследование применимости разработанного алгоритма MINDI	175
3.3.2 Результат разработки алгоритма формирования ансамбля информативных признаков	178
3.4 Алгоритм автоматического построения ансамбля информативных индексов (A-MINDI)	179
3.4.1 Применение метода главных компонент в рамках предложенного алгоритма	180
3.4.2 Метод автоматической разметки гиперспектральных изображений	182
3.4.3 Описание разработанного алгоритма A-MINDI	183
3.4.4 Экспериментальное исследование алгоритма A-MINDI	186
3.4.5 Результат разработки алгоритма автоматического построения ансамбля информативных индексов	191
3.5 Детали программной реализации алгоритмов INDI, MINDI и A-MINDI	192
3.6 Метод анализа с использованием схожести с эталоном	196
3.7 Метод обработки с использованием вегетационных индексов.....	199
3.8 Анализ гиперспектральных изображений методом К-средних	202
3.9 Метод эффективного хранения гиперспектральных изображений	204
3.10 Метод мозаичной обработки данных.....	208
3.11 Методы предварительной нормировки данных.....	210
3.12 Алгоритм цветосинтеза	211
3.13 Выводы и результаты по третьему разделу.....	212
Заключение	217
Список литературы	221
Приложение А. Акты об использовании результатов диссертационной работы.....	241

ВВЕДЕНИЕ

Диссертационная работа посвящена разработке алгоритмов, методов и интеллектуальной системы для решения комплекса задач, направленных на хранение, визуализацию и обработку гиперспектральных изображений.

Актуальность темы

Гиперспектральные изображения являются одним из наиболее информативных источников данных для анализа объектов и процессов, недоступных при использовании классических RGB-изображений. В зависимости от числа спектральных каналов и соотношения пространственного и спектрального разрешения изображения, получаемые в результате спектральной съемки, принято подразделять на следующие типы:

- мультиспектральные данные, обладающие высоким пространственным и сравнительно низким спектральным разрешением, как правило включающие десятки спектральных полос;
- гиперспектральные данные, обладающие высоким спектральным разрешением и включающие сотни спектральных полос;
- ультраспектральные данные, характеризующиеся наибольшим спектральным разрешением и включающие тысячи спектральных полос.

В настоящей работе далее для краткости используется термин **«гиперспектральные изображения» (ГСИ)**, под которым понимаются изображения, получаемые в результате спектральной съемки и включающие мультиспектральные, гиперспектральные и ультраспектральные данные. Такое обобщение обусловлено отсутствием в русскоязычной литературе устойчивого термина, единообразно объединяющего указанные типы изображений, а также распространенной в англоязычных источниках практикой использования термина *Hyperspectral Imaging* (HSI) для обозначения широкого класса методов и данных спектральной съемки без строгого разграничения мультиспектральных, гиперспектральных и ультраспектральных изображений.

Истоки современных методов спектральной съемки восходят к работам Й. Фраунгофера, Р. Бунзена и Г. Кирхгофа [1], заложивших основы спектроскопии и спектрального анализа. Как самостоятельное направление анализа изображений спектральная съемка получила развитие в 1980-х годах в работах А. Гёца, Г. Вэйна, Дж. Соломона и Б. Рока [2], связанных с формированием концепции спектрометрической съемки и созданием первых авиационных спектрометров для дистанционного зондирования.

ГСИ формируются путем спектральной съемки целевых сцен или объектов специальными камерами, захватывающими широкий электромагнитный спектр [3, 4]. Захваченный диапазон света как правило, ограничен видимым спектром (длина волны 0,4 – 0,7 мкм) и инфракрасным спектром (0,7 – 2,4 мкм). В редких случаях такие камеры могут захватывать ультрафиолетовое излучение (0,1 - 0,4 мкм). Изображения, полученные в результате такой съемки, можно представить в виде трехмерного куба, вдоль оси аппликаты которого располагаются спектральные слои (часто называемые – спектральными полосами, каналами), соответствующие разным длинам волн света.

Благодаря высокой информативности ГСИ находят широкое применение в различных прикладных областях. В сельском хозяйстве они используются для мониторинга состояния культур, выявления заболеваний и вредителей, оценки водного стресса растений, оптимизации полива и внесения удобрений. В лесном хозяйстве и экологии гиперспектральные данные применяются для оценки состояния растительности, мониторинга окружающей среды, выявления загрязнений и анализа последствий природных катаклизмов. В медицине гиперспектральные изображения используются при решении задач диагностики заболеваний, включая онкологические и сердечно-сосудистые патологии, что открывает возможности для более раннего обнаружения нарушений и повышения точности анализа биологических тканей.

Приведенные примеры подтверждают значимость гиперспектральных изображений, однако их широкое применение ограничивается рядом существенных проблем. Во-первых, гиперспектральные камеры остаются дорогостоящими и менее распространенными по сравнению с потребительскими фото- и экшен-камерами. Для нивелирования этой проблемы исследователи изучают возможности создания более дешевых и простых образцов камер [5, 6], а коммерческие организации предоставляют доступ к гиперспектральным данным спутниковой съемки [7, 8]. Во-вторых, гиперспектральные камеры обладают повышенной чувствительностью к внешним факторам и условиям съемки [9, 10], включая неравномерность освещения, погодные условия и порывы ветра. Учет данных факторов необходим как при разработке оборудования для съемки, так и при создании алгоритмов анализа гиперспектральных данных.

Дополнительным ограничением является недостаток доступных наборов данных, особенно для разработки, обучения и оценки новых алгоритмов машинного обучения. Большинство открытых наборов содержит ограниченное число гиперспектральных изображений [11–14], а значительная часть данных, собираемых исследователями самостоятельно, не публикуется в открытом доступе. При этом прямое применение к гиперспектральным изображениям алгоритмов, разработанных для монохромных и цветных изображений, сопряжено с существенными трудностями [15]. Такие методы часто требуют адаптации, направленной на оптимизацию вычислений [16–18], снижение размерности данных, учет пространственно-спектральной структуры и борьбу с переобучением [19]. Это обусловлено как большими объемами информации, так и высокой сложностью пространственно-спектральной структуры гиперспектральных изображений.

Особое значение имеет проблема выбора и разработки инструментов хранения, визуализации и обработки гиперспектральных изображений. Гиперспектральные данные имеют сложную пространственно-спектральную

структуру и значительный объем, что предъявляет повышенные требования к программным средствам. Такие средства должны обеспечивать не только загрузку и отображение изображений, но и организацию централизованного хранения, управление метаданными, интерактивную визуализацию спектральных слоев, формирование цветовых композиций, выполнение алгоритмов предварительной обработки, сегментации, кластеризации, классификации и анализа спектральных признаков.

Большинство существующих систем [20, 21] ориентировано на использование в рамках определенных предприятий, исследовательских групп или узких прикладных сценариев. Как правило, такие программные средства распространяются в виде ресурсоемких настольных приложений, имеют лицензионные ограничения, требуют установки и настройки на локальном компьютере, а также предъявляют высокие требования к вычислительным ресурсам пользователя. Кроме того, во многих случаях они не обеспечивают единого подхода к хранению, визуализации и обработке гиперспектральных изображений, не поддерживают централизованное управление наборами данных, затрудняют совместную работу пользователей и ограничивают возможность использования на мобильных или маломощных устройствах.

Важным ограничением существующих решений является недостаточная автоматизация обработки гиперспектральных изображений. Пользователь часто вынужден самостоятельно выбирать алгоритмы обработки, настраивать их параметры, интерпретировать промежуточные результаты и выполнять подготовку данных. Это затрудняет применение спектрального анализа в прикладных задачах, где требуется быстрое получение результата без предварительной разметки данных и без глубокого понимания методов машинного обучения.

Дополнительную актуальность имеет проблема расширяемости программных средств. Область анализа гиперспектральных изображений активно развивается: появляются новые алгоритмы кластеризации, снижения

размерности, классификации, сегментации. Поэтому программное обеспечение должно позволять подключать пользовательские алгоритмы, использовать заранее обученные модели, распределять вычислительные задачи между доступными ресурсами и адаптироваться к новым исследовательским и прикладным сценариям.

Таким образом, актуальной является разработка интеллектуальной системы, объединяющей хранение, визуализацию и обработку гиперспектральных изображений в единой программной среде. Такая среда должна обеспечивать централизованное хранение данных, интерактивную визуализацию ГСИ, автоматизированное выполнение алгоритмов анализа, включая методы обучения без учителя, а также расширение функциональности за счет добавления новых алгоритмов и моделей машинного обучения. Решение данной задачи позволит снизить требования к квалификации пользователя, повысить доступность средств анализа гиперспектральных изображений и обеспечить более эффективное применение таких данных в научных и прикладных исследованиях.

Степень разработанности темы

Развитию аппаратных средств регистрации гиперспектральных данных сопутствовало развитие методов и алгоритмов их обработки, анализа и интерпретации. Вопросы анализа гиперспектральных изображений получили широкое развитие в работах российских и зарубежных исследователей. Существенный вклад в развитие данного научного направления внесли В.А. Зеленцов, Н.Л. Казанский, В.В. Козодеров, Е.А. Лупян, В.В. Мясников, А.В. Никоноров, Н.А. Обухова, И.А. Пестунов, В.Э. Пожар, О.И. Потатуркин, В.П. Пяткин, В.П. Саворский, В.В. Сергеев, В.А. Сойфер, В.Е. Турлапов, В.А. Фурсов, Б.В. Шилин и др. Существенный вклад в исследование вопросов формирования, комплексирования, сегментации и обработки гиперспектральных изображений внесли В.В. Еремеев, А.А. Макаренков, А.Э. Москвитин, А.А. Юдаков, Н.А. Егошкин, С.В. Антонушкина, Г.Н. Мятлов и др.

Среди зарубежных ученых, работы которых оказали значительное влияние на развитие методов анализа гиперспектральных изображений, следует отметить J.A. Benediktsson, J.M. Bioucas-Dias, J.W. Boardman, L. Bruzzone, G. Camps-Valls, C.-I. Chang, J. Chanussot, M. Fauvel, P. Gamba, D. Landgrebe, J.F. Mustard, J.M.P. Nascimento, A. Plaza и др.

Несмотря на значительную степень проработанности методов анализа гиперспектральных изображений, сохраняется потребность в разработке интеллектуальных систем, обеспечивающих не только выполнение отдельных алгоритмов обработки, но и объединение в единой программной среде инструментов решения задач хранения, визуализации, автоматизированной обработки, предварительной разметки данных, анализа и интеграции пользовательских алгоритмов.

Цель и задачи исследований

Целью диссертационной работы является разработка интеллектуальной системы для автоматизации обработки гиперспектральных данных, обеспечивающей формирование информативных спектральных индексов, предварительную разметку данных, хранение, визуализацию, анализ и возможность интеграции пользовательских алгоритмов в единой программной среде.

В соответствии с поставленной целью сформулированы следующие основные задачи диссертации:

1) сформировать систему значимых требований к математическому и программному обеспечению автоматизированной обработки гиперспектральных изображений на основе обзора существующих методов, алгоритмов и систем;

2) разработать архитектуру и функционально расширяемую интеллектуальную систему для хранения, визуализации и обработки гиперспектральных изображений, обеспечивающую интеграцию пользовательских алгоритмов и моделей машинного обучения;

3) разработать методы и алгоритмы автоматизированной обработки гиперспектральных изображений, обеспечивающие построение информативных спектральных индексов и предварительную разметку данных.

Объект исследования

Объектом исследования являются системы обработки гиперспектральных изображений.

Предмет исследования

Предметом исследования являются модели, методы, алгоритмы и архитектурные решения, обеспечивающие хранение, визуализацию и обработку гиперспектральных изображений.

Методология и методы исследований

Методологическую основу исследования составляют методы цифровой обработки изображений, спектрального анализа, машинного обучения, системного анализа и проектирования распределенных программных систем. В работе использованы методы анализа научно-технической литературы, формализации задач формирования информативных спектральных признаков, сегментации, кластеризации, классификации, оценки информативности спектральных каналов и отделимости классов. При разработке интеллектуальной системы применялись методы объектно-ориентированного проектирования, асинхронной обработки вычислительных задач, контейнеризации и организации взаимодействия программных компонентов. Достоверность результатов подтверждалась экспериментальными исследованиями с использованием общепринятых метрик.

Научная новизна

1) Разработана архитектура интеллектуальной системы обработки гиперспектральных данных, отличающаяся выделением независимых компонентов управления исходными гиперспектральными данными, результатами анализа и визуализации, постановки вычислительных задач и выполнения алгоритмов, что обеспечивает горизонтальное масштабирование

компонентов системы и её функциональное расширение за счёт интеграции пользовательских алгоритмов и моделей машинного обучения.

2) Разработана интеллектуальная система для хранения, визуализации и обработки гиперспектральных изображений, отличающаяся объединением в единой программной среде функций централизованного управления гиперспектральными данными, интерактивной визуализации, автоматизированной обработки, интеграции пользовательских алгоритмов и моделей машинного обучения, а также выполнения вычислительных задач с использованием распределённых вычислительных ресурсов.

3) Впервые разработан алгоритм формирования информативного спектрального индекса, отличающийся автоматическим отбором пары спектральных полос для выделения целевых объектов, заданных в рамках пользовательской задачи, что позволяет формировать интерпретируемый спектральный признак без ручного выбора спектральных каналов и упрощает применение алгоритма на различных наборах гиперспектральных данных.

4) Разработан метод автоматической разметки гиперспектральных изображений, отличающийся совместным применением нейросетевой сегментации и информативных спектральных индексов для уточнения предварительных меток, что позволяет уменьшить потребность в ручном формировании эталонной разметки и упростить создание обучающих выборок для последующего применения нейросетевых алгоритмов.

Теоретическая значимость работы

Теоретическая значимость работы заключается в развитии методов автоматизированной обработки гиперспектральных изображений за счёт формализации задач построения информативных спектральных индексов для бинарной и многоклассовой сегментации, разработки алгоритмов INDI и MINDI для автоматического формирования интерпретируемых спектральных признаков, а также разработки алгоритма A-MINDI для формирования

предварительной разметки данных при отсутствии исходной экспертной разметки.

Практическая значимость работы

Практическая значимость работы заключается в возможности применения разработанных алгоритмов и интеллектуальной системы для хранения, визуализации и автоматизации обработки гиперспектральных изображений, а также для формирования информативных индексов, обеспечивающих выделение целевых областей, определяемых условиями прикладной задачи. Разработанные решения позволяют снизить трудоемкость анализа гиперспектральных данных, уменьшить зависимость от ручного выбора спектральных полос и параметров обработки, а также обеспечить расширение функциональных возможностей системы за счет интеграции пользовательских алгоритмов и моделей машинного обучения.

Реализация результатов работы

Результаты диссертации использованы при выполнении ряда госбюджетных работ:

1) проект №20-51-05008 РФФИ (Арм_а) построение вычислительно эффективных процедур для формирования набора информативных признаков согласованных с текстурными свойствами изображений;

2) государственное задание Минобрнауки России № FSSS-2023-0006 «Теоретические основы и программное обеспечение систем интеллектуального анализа оптических изображений и видеопотоков для экспериментального исследования характеристик аналоговых фотонных сопроцессоров с применением графических ускорителей»;

3) государственное задание Минобрнауки России № FSSS-2021-0016 «Дизайн высокоточных спектральных фильтров и интегрированных на чипе спектрометров на основе структур нанофотоники для мониторинга характеристик окружающей среды»;

4) государственное задание Минобрнауки России № 0777-2020-0017 «Разработка теоретических основ, методического и программного обеспечения для решения задач оперативного мониторинга Земли на основе интеллектуальной обработки и тематической классификации гиперспектральных данных с использованием линейки малых космических аппаратов на базе платформы «АИСТ»;

5) научное исследование и разработка в рамках реализации Программы развития передовой инженерной аэрокосмической школы (ПИАШ), проект «Технология интеллектуального анализа данных ДЗЗ на основе цифровой платформы и создание лабораторного макета полнокадрового гиперспектрального комплекса».

Достоверность полученных результатов достигается за счёт применения объективных и общепринятых метрик оценки качества анализа гиперспектральных изображений, воспроизводимости вычислительных экспериментов на различных наборах данных, сравнений с эталонными алгоритмами, а также подтверждения результатов в авторитетных рецензируемых изданиях и на профильных научных конференциях.

Апробация работы

Основные результаты диссертации были представлены на международных научных конференциях: X Международная конференция и молодёжная школа «Информационные технологии и нанотехнологии» (ИТНТ-2024), VIII Международная конференция и молодёжная школа «Информационные технологии и нанотехнологии» (ИТНТ-2022), VII Международная конференция и молодёжная школа «Информационные технологии и нанотехнологии» (ИТНТ-2021); 3-й международный семинар в Американском университете Армении, Колледж науки и техники (23-25 августа, 2022, Армения).

Публикации

По теме диссертационной работы автором опубликовано 8 работ, в т.ч.

2 работы в изданиях, индексируемых в базах данных Scopus и Web of Science (WoS), 3 работы в изданиях из списка ВАК РФ, получено 9 свидетельств о государственной регистрации программы для ЭВМ.

Личный вклад автора

Личный вклад автора состоит в участии на всех этапах процесса исследования, включая постановку задач, разработку комплекса алгоритмических средств решения задач хранения, визуализации и анализа гиперспектральных данных, постановку экспериментов, апробацию результатов исследования, разработку архитектуры и реализацию интеллектуальной системы. Выбор концепции диссертационной работы и направления исследований проводился совместно с научным руководителем.

Соответствие паспорту специальности

Область исследования соответствует пунктам: 4 – Интеллектуальные системы машинного обучения, управления базами данных и знаний, инструментальные средства разработки цифровых продуктов; 7 – Модели, методы, архитектуры, алгоритмы, форматы, протоколы и программные средства человеко-машинных интерфейсов, компьютерной графики, визуализации, обработки изображений и видеоданных, систем виртуальной реальности, многомодального взаимодействия в социкиберфизических системах направлений исследований паспорта научной специальности 2.3.5. Математическое и программное обеспечение вычислительных систем, комплексов и компьютерных сетей (технические науки).

Структура диссертации

Диссертация состоит из введения, трех глав, заключения, списка литературы из 143 наименований и одного приложения. Работа изложена на 245 страницах, содержит 83 рисунка, 24 таблицы.

На защиту выносятся

1) **Архитектура интеллектуальной системы** обработки гиперспектральных изображений, обеспечивающая разделение функций

управления данными, визуализации и обработки, горизонтальное масштабирование компонентов и функциональное расширение системы за счёт интеграции пользовательских алгоритмов и моделей машинного обучения.

2) Интеллектуальная система для хранения, визуализации и обработки гиперспектральных изображений, реализующая централизованную работу с данными, интерактивную визуализацию, асинхронный и автоматизированный анализ гиперспектральных изображений, а также интеграцию пользовательских алгоритмов и моделей машинного обучения.

3) Алгоритм формирования ансамбля информативных индексов, обеспечивающий выбор множества пар спектральных каналов и построение набора нормализованных разностных индексов для классификации заданных классов объектов на гиперспектральном изображении, определяемых условиями пользовательской задачи.

4) Метод автоматического формирования ансамбля информативных индексов, обеспечивающий построение ансамбля индексов без исходной экспертной разметки и применимый как для автоматизированного анализа гиперспектральных изображений, так и для формирования разметки при последующем обучении моделей машинного обучения.

1 Анализ предметной области и формирование требований к математическому и программному обеспечению обработки гиперспектральных изображений

1.1 Гиперспектральные изображения, их развитие, особенности и применимость

За последние десятилетия был достигнут существенный прогресс в разработке новых оптических сенсоров. В частности, были разработаны первые спектральные сенсоры, такие как Airborne Imaging Spectrometer (AVIRIS) [22], работающие в видимом и инфракрасном спектрах. Также были созданы сенсоры [23], способные запечатлеть ультрафиолетовый спектр. Основная цель изобретения таких спектральных оптических сенсоров заключалась в необходимости оценки особых спектральных характеристик объектов, например, их химического состава [24] или для обнаружения невидимых дефектов [25]. Как правило решение данной задачи невозможно с помощью анализа классических трехканальных RGB-изображений.

С помощью спектральных сенсоров формируются мультиспектральные и гиперспектральные изображения. На рисунке 1.1 изображена визуальная демонстрация различий между типами изображений.

Различие между мультиспектральными и гиперспектральными изображениями



Рисунок 1.1 – Визуальная демонстрация различий между мультиспектральными и гиперспектральными изображениями

Разделение спектральных изображений на типы по количеству спектральных полос вносит определённую долю путаницы, которая уже закрепилась в научном сообществе. Основоположники области призывают

отказаться [26] от использования приставок “гипер-“ и “мульти-“ для избегания путаницы. В данной работе будет использоваться термин гиперспектральные изображения (ГСИ), под которым будут подразумеваться все типы спектральных изображений: мульти-, гипер- и ультраспектральные изображения.

Независимо от типа гиперспектральных изображений, их можно представить в виде трёхмерного куба (поэтому гиперспектральные изображения также называют *гиперкубами*), вдоль оси аппликаты которого располагаются спектральные слои (часто называемые спектральными полосами), соответствующие разным длинам волн света. На рисунке 1.2 показано гиперспектральное изображение, представленное в виде трёхмерного куба.

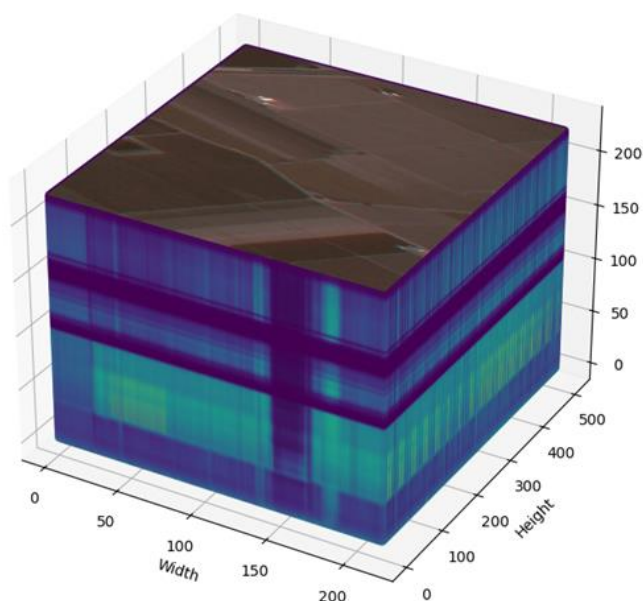


Рисунок 1.2 – Представление гиперспектрального изображения в виде трёхмерного куба

Существуют следующие разновидности спектральных сенсоров, различающихся методами сканирования:

1. Пространственное сканирование (spatial scanning) [27].

Данный тип сканирования формирует ГСИ путём агрегации снимков в узком пространственном и полном спектральном разрешении. Представляя ГСИ в виде куба, этот тип сканирования можно интерпретировать как

постепенное заполнение куба данными вдоль одной из пространственных осей (абсциссы или ординаты).

2. Спектральное сканирование (spectral scanning) [27].

Этот тип сканирования формирует ГСИ путём агрегации снимков отдельных спектральных полос в полном пространственном разрешении. Представляя ГСИ в виде куба, данный тип сканирования можно интерпретировать как постепенное заполнение куба данными вдоль оси аппликаты.

3. Метод мгновенной съемки (snapshot hyperspectral imaging) [28, 29].

Данный метод позволяет сформировать ГСИ целиком без агрегации отдельных пространственных или спектральных слоёв.

На рисунке 1.3 представлено изображение, визуально демонстрирующее вышеописанные различия методов формирования ГСИ.

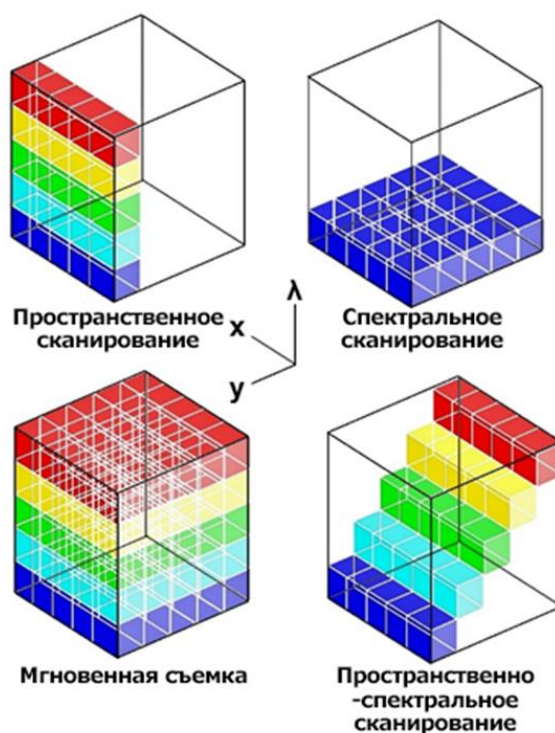


Рисунок 1.3 – Визуализация различных методов съемки ГСИ

В настоящее время спектральные сенсоры до сих пор не получили широкого распространения. Как правило, спектральные сенсоры стоят на порядки дороже, чем обычные камеры. Поэтому особенно актуальны

исследования, направленные на создание более доступных спектральных сенсоров. Из-за использования техник пространственного и спектрального сканирования такие сенсоры оказываются высокочувствительными к внешним факторам: переменчивости освещения, погодным условиям, порывам ветра [30, 31]. В связи с развитием беспилотных летательных аппаратов и появлением новых областей применения гиперспектральных изображений существует также необходимость в разработке компактных спектральных сенсоров [32, 33].

Далее рассмотрим основные области применения гиперспектральных изображений.

Дистанционное зондирование Земли (ДЗЗ)

Гиперспектральные изображения находят широкое применение в области ДЗЗ. Использование ГСИ в ДЗЗ обусловлено рядом факторов:

- ГСИ, благодаря высокому пространственному разрешению, позволяют анализировать большие площади поверхности Земли;
- высокое спектральное разрешение обеспечивает возможность разработки эффективных алгоритмов для анализа состояния растительности и почвы;
- исторический фактор: первые спектральные сенсоры были установлены на искусственных спутниках Земли и использовались для наблюдения за атмосферой и поверхностью планеты.

В области ДЗЗ гиперспектральные изображения применяются наиболее широко. Поэтому далее рассмотрим их использование в специальных областях, таких как сельское и лесное хозяйство, а также геология.

Сельское хозяйство

Развитие применения ГСИ наиболее наглядно прослеживается в области сельского хозяйства. Изначально в данной области широко использовались мультиспектральные данные, полученные с таких спутников, как Landsat [34], Sentinel-2 [35] и других. С помощью этих данных решались задачи, такие как

оценка характеристик культур [36] и почвы [37], определение содержания хлорофилла [38] и биомассы в растениях [39], а также оценка уровня насыщения почвы влагой [40]. Однако применение мультиспектральных изображений показало, что их ограниченное спектральное разрешение не позволяет оценивать характеристики культур и почвы с достаточной степенью детализации для своевременного определения заболеваемости культур [41].

Применение гиперспектральных изображений с их высоким спектральным разрешением, напротив, позволяет с большей эффективностью отслеживать процессы, происходящие в почве и растениях. Многочисленные исследования продемонстрировали значительные преимущества гиперспектральных изображений, которые позволяют более эффективно оценивать характеристики растений [42], почвы [24, 43], определять типы культур [44], оценивать биомассу [45] и содержание азота [46].

Однако, несмотря на продемонстрированные преимущества использования гиперспектральных изображений, они не получили широкого распространения в сельском хозяйстве из-за высокой стоимости спектральных сенсоров и их чувствительности к внешним факторам при съемке.

Позднее были изобретены специальные спектрорадиометры [47], предназначенные для наземных измерений непосредственно на исследуемых участках с целью оценки различных характеристик сельскохозяйственных культур и почвы. Однако их основными недостатками являлись необходимость физического присутствия на целевой территории и ограниченное поле обзора.

С развитием спектральных сенсоров и беспилотных летательных аппаратов появились более доступные и компактные спектральные сенсоры [48]. Эти сенсоры могли быть установлены на беспилотные летательные аппараты, что позволяло проводить анализ больших территорий. Развитие аппаратов для спектральной съёмки способствует повышению доступности анализа гиперспектральных данных.

В результате такого развития спектрального анализа в сельском хозяйстве в настоящее время формируются новые направления его применения: для оценки вегетационных характеристик стресса культур [49], оценки влияния различных условий на процесс роста культур [50], оптимизации процесса лечения культур [51] и других задач.

Лесное хозяйство

Другой обширной областью применения спектральных сенсоров является лесное хозяйство. Применение ГСИ, полученных методами ДЗЗ, позволяет анализировать большие территории для контроля состояния лесов и лесного покрова [52]. С помощью гиперспектральных изображений можно решать задачи, такие как предсказание очагов возникновения лесных пожаров [53], что позволяет своевременно предотвращать их или оперативно организовывать тушение.

Геология

ГСИ также применяются в геологии. Благодаря высокому пространственному и спектральному разрешению, геологи могут определять различные геологические характеристики почвы и минералов. Одной из наиболее изученных задач является литологическое картирование [54], то есть классификация минералов [55], почв [56] и других структур на основе данных ДЗЗ. Согласно исследованиям [57], применение спектральных сенсоров позволяет с высокой степенью детализации различать оксиды и гидроксиды железа, а также обнаруживать различные сульфаты, карбонаты и другие структуры. Это делает возможной геологическую разведку в удаленных от человека районах Земли [58].

ГСИ также помогают решать проблемы, вызванные негативным воздействием человека на природу [59]. Например, заброшенные шахты, особенно остатки токсичных минералов [60], могут загрязнять близлежащие источники воды, почву и растительность [61]. Анализ гиперспектральных изображений позволяет обнаруживать такие геологические отходы [62, 63].

Другим важным классом задач в данной области является предсказание природных катаклизмов. Чаще всего ГСИ используются для прогнозирования возможных оползней и сходов грунта [64].

Здравоохранение

Рассмотрим применение ГСИ в области здравоохранения. Благодаря созданию компактных спектральных сенсоров и развитию методов анализа ГСИ, появился значительный потенциал их применения в медицине и здравоохранении. Наиболее часто ГСИ используются для диагностики различных заболеваний. Например, для классификации степени развития рака шейки матки [65], обнаружения опухолей кожи [66], оценки насыщения крови кислородом [67] и выявления аномалий в структурах кровеносных сосудов и вен [68]. Кроме того, ГСИ позволяют проводить химический анализ различных жидкостей, например, оценивать характеристики крови, такие как содержание нуклеотидов, цитоплазмы, лейкоцитов и других компонентов [69]. Также существуют исследования, направленные на применение гиперспектральных изображений для диагностики кариеса [70] и сегментации ротовой полости [71].

Однако, несмотря на значительный потенциал применения анализа ГСИ в здравоохранении, доступность и распространённость этого подхода для диагностики заболеваний остаются недостаточными.

Производство и контроль качества

ГСИ применяются в различных отраслях промышленности для обеспечения контроля качества выпускаемой продукции. Наиболее часто ГСИ используются для контроля качества пищевых продуктов. Например, с их помощью решаются задачи обнаружения механических повреждений в грибах [72], оценки твёрдости крупы [73], выявления повреждений от холода у персиков [74], обнаружения фальсификации мёда [75], обнаружения насекомых в крупах [76] и другие. Кроме решения задач обнаружения, ГСИ используются для оценки различных характеристик продуктов, например, для

классификации сортов яблок [77] и прогнозирования нежности говядины [78]. Применение ГСИ позволяет проводить постоянную, потоковую и оперативную проверку всей выпускаемой продукции, в отличие от ручного, выборочного анализа отдельных образцов.

Помимо сферы контроля качества продуктов, ГСИ находят применение в различных отраслях, связанных с обработкой металлов и сырой руды. Например, с помощью ГСИ можно осуществлять классификацию железной руды [79] и оценку качества руд на этапе разработки месторождений [80].

Хотя применение анализа ГСИ в данных областях остаётся недостаточно изученным, существующие исследования свидетельствуют о том, что использование ГСИ для контроля качества на различных производствах является перспективным направлением развития.

1.2 Программное обеспечение анализа гиперспектральных изображений и формирование требований

Несмотря на актуальность и широкие возможности применения ГСИ, существует ограниченное количество программных систем, позволяющих решать задачи их хранения, визуализации и обработки в единой программной среде. Большинство существующих решений ориентировано либо на специалистов, обладающих опытом работы с методами обработки ГСИ и машинного обучения, либо на узкие прикладные сценарии, связанные с конкретными типами сенсоров, форматами данных или предметными областями.

Особую проблему представляет недостаточная автоматизация анализа ГСИ для пользователей, не обладающих специальными компетенциями в области машинного обучения, статистического анализа и обработки изображений. Во многих случаях пользователь должен самостоятельно выбирать алгоритмы, задавать параметры обработки, интерпретировать промежуточные результаты и выполнять подготовку данных. Это затрудняет применение ГСИ в прикладных задачах, где требуется быстрое получение

аналитического результата без предварительной разметки данных, например при использовании методов кластеризации, снижения размерности и других подходов обучения без учителя.

Кроме того, существующие программные системы обладают ограниченной гибкостью. Часть решений не позволяет внедрять собственные алгоритмы анализа, подключать пользовательские модели машинного обучения или использовать заранее обученные нейросетевые модели. Это является существенным ограничением для специалистов-исследователей, поскольку область анализа ГСИ быстро развивается, а эффективность обработки данных во многом зависит от возможности оперативно адаптировать систему под новые методы, модели и предметные задачи.

Дополнительным ограничением является преобладание настольных программных средств. Такие решения требуют установки на конкретное рабочее место, зависят от характеристик локального оборудования и не обеспечивают полноценного доступа к данным и инструментам анализа с произвольных устройств. В результате затрудняется совместная работа исследовательских групп, централизованное хранение данных и использование вычислительных ресурсов, вынесенных за пределы пользовательского устройства.

Далее рассмотрены основные программные системы, применяемые для работы с ГСИ, их функциональные возможности и ограничения.

1.2.1 Критерии сравнения программных систем обработки ГСИ

Для сопоставления систем анализа ГСИ были выбраны свойства, отражающие полный цикл работы с данными: от загрузки и хранения изображений до визуализации, анализа и расширения функциональных возможностей системы. Выбранные свойства сгруппированы по функциональным блокам, каждый из которых соответствует отдельному этапу пользовательского сценария.

Первая группа свойств характеризует **доступность и способ эксплуатации программного обеспечения**. В нее включены признаки, связанные с независимостью от операционной системы, возможностью использования на мобильных и маломощных устройствах, а также облачной моделью доступа. Данные свойства важны, поскольку определяют, может ли пользователь работать с ГСИ без установки специализированного программного обеспечения и без привязки к конкретному рабочему месту.

Вторая группа свойств относится к **управлению данными**. Она включает поддержку различных форматов ГСИ, работу с данными от спектральных сенсоров и возможность централизованного хранения. Эти свойства определяют, насколько система применима к данным различного происхождения и может ли она использоваться не только как средство анализа, но и как инфраструктурный компонент для накопления и организации ГСИ.

Третья группа отражает возможности **визуализации ГСИ**. В нее включены интерактивный интерфейс, построение изображений из комбинаций спектральных полос, визуализация спектров пикселей и отображение распределений значений. Эти свойства необходимы для первичного исследования данных, контроля качества изображений и интерпретации результатов анализа.

Четвертая группа описывает **аналитические возможности программных систем**. Она включает методы уменьшения размерности, кластеризации, спектрального анализа, машинного обучения, нейросетевые методы и спектральные индексы. Выбор этих свойств обусловлен тем, что они охватывают основные классы алгоритмов, применяемых при обработке ГСИ: исследование структуры данных, выделение групп объектов без предварительной разметки, построение классификационных и регрессионных моделей, а также расчет прикладных индикаторов на основе спектральных характеристик.

Пятая группа свойств характеризует **расширяемость и адаптируемость системы**. К ней относятся возможность добавления собственных алгоритмов, подключение пользовательских моделей машинного обучения и использование готовых моделей. Эти свойства особенно важны для исследовательских задач, поскольку позволяют не ограничиваться заранее заданным набором инструментов и адаптировать систему к новым методам анализа данных.

Таким образом, выбранный перечень свойств является исчерпывающим в рамках рассматриваемой задачи, поскольку охватывает все ключевые аспекты функционирования системы анализа ГСИ: доступ к системе, работу с данными, визуализацию, аналитическую обработку и расширяемость. Свойства, не влияющие напрямую на указанные этапы, в сравнение не включались, поскольку они не определяют принципиальные возможности системы при решении задач хранения, визуализации и анализа изображений.

1.2.2 Обзор программных систем анализа гиперспектральных изображений

1. *HYPER-Tools*

Данное программное обеспечение [81] является расширением среды разработки MATLAB [82], которая, в свою очередь, представляет собой настольное приложение. Расширение *HYPER-Tools* предоставляет графический интерфейс, позволяющий выполнять анализ и визуализацию гиперспектральных и мультиспектральных изображений. С его помощью нельзя управлять хранением изображений, и единственным поддерживаемым форматом данных является формат MAT-file (.mat). Несмотря на это, *HYPER-Tools* предоставляет широкие функциональные возможности для анализа ГСИ. Ниже перечислены основные категории реализованных алгоритмов:

- 1) алгоритмы пространственной и спектральной предобработки (морфологические операции: наращивание, эрозия, замыкание и др.; удаление некорректных пикселей и всплесков; алгоритмы

- сглаживания и эквализации; стандартное нормальное преобразование (SNV), мультипликативная коррекция рассеяния (MSC) и др.);
- 2) алгоритмы уменьшения размерности (метод главных компонент (PCA), факторный анализ);
 - 3) алгоритмы кластеризации (метод к-средних, метод нечеткой кластеризации);
 - 4) алгоритмы для нахождения корреляции спектральных полос;
 - 5) алгоритмы машинного обучения (регрессия наименьших квадратов (PLSR), регрессия на главных компонентах (PCR), множественная линейная регрессия (MLR), метод опорных векторов (SVM));
 - 6) и другие алгоритмы, используемые для анализа и визуализации гиперспектральных изображений.

Несмотря на широкий набор доступных алгоритмов, данное программное средство имеет значительные ограничения:

- HYPER-Tools является расширением для MATLAB, что требует установки отдельного программного обеспечения и ограничивает возможные сценарии использования;

- программное средство предназначено преимущественно для локальной работы и не обеспечивает облачный доступ к данным и инструментам анализа;

- отсутствуют средства организации централизованного хранения и каталогизации ГСИ;

- система не ориентирована на пользователей без специальных компетенций: выбор алгоритмов, настройка параметров и интерпретация результатов требуют понимания методов обработки ГСИ;

- отсутствуют механизмы автоматизированного анализа изображений без предварительной разметки, ориентированные на прикладного пользователя;

– не предусмотрено полноценное подключение пользовательских алгоритмов машинного обучения и использование заранее обученных моделей;

– в составе системы отсутствуют современные нейросетевые методы анализа гиперспектральных изображений;

– не реализованы методы анализа данных с использованием спектральных индексов.

Таким образом, HYPER-Tools может рассматриваться как вспомогательный инструмент для исследовательского анализа ГСИ в среде MATLAB, однако его возможности ограничены при решении задач облачного доступа, централизованного хранения, автоматизированного анализа и расширения функциональности за счет пользовательских моделей.

На рисунке 1.4 приведен пример, как выглядит графический интерфейс.

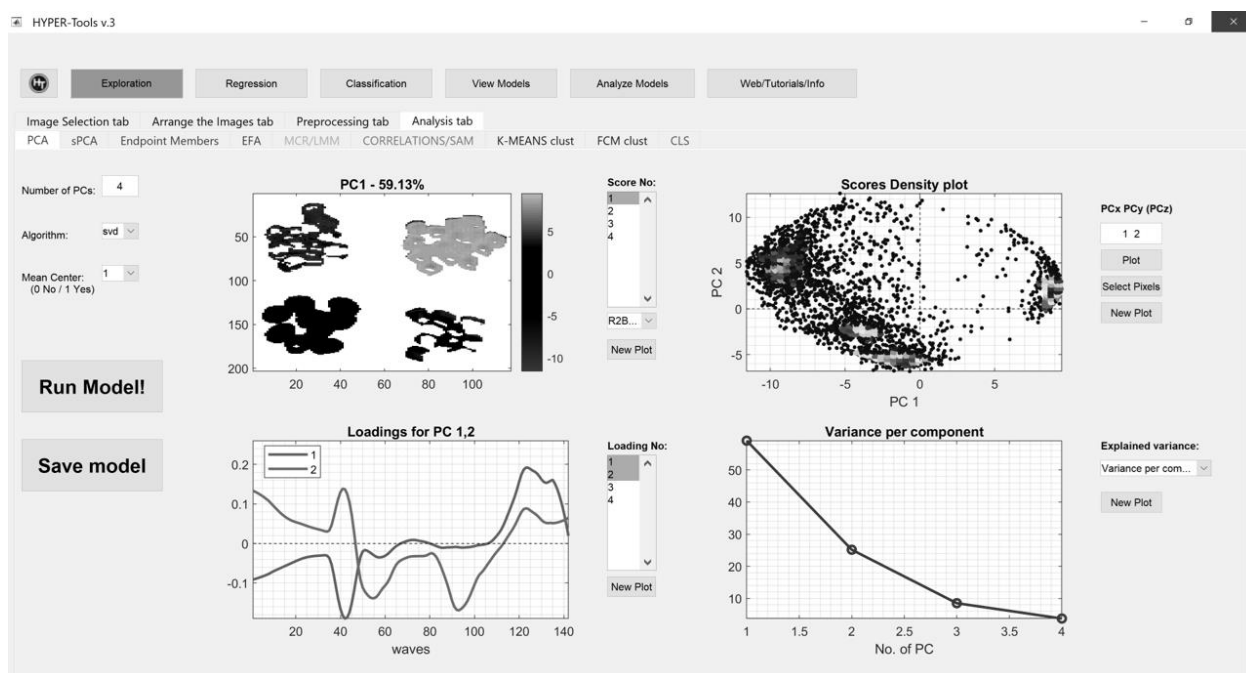


Рисунок 1.4 – Графический интерфейс программы HYPER-Tools

2. ПО Prediktera

Prediktera – это шведская компания [20], предоставляющая набор программ для работы с гиперспектральными и мультиспектральными

изображениями. Компания выделяет несколько целевых областей для применения своих продуктов:

1) **сельское хозяйство**: программные средства *Prediktera* позволяют решать задачи, такие как обнаружение и сортировка посторонних материалов среди зерна или семян, а также выявление заболеваний культур и дефектов продукции;

2) **производство продуктов питания**: применение программного обеспечения компании помогает решать задачи, связанные с детектированием испорченных продуктов, проверкой герметичности упаковки и целостности пломб;

3) **форензика**: программные решения используются для обнаружения и идентификации наркотических веществ и поддельных препаратов;

4) **геология**: продукты компании применяются для задач геологической разведки, включая определение состава пород и минералов;

5) **фармацевтика**: программные средства позволяют контролировать химический состав препаратов и оценивать их качество;

6) **лесная промышленность**: программное обеспечение *Prediktera* используется для определения характеристик сырья и продукции в реальном времени.

Такие широкие возможности применения обусловлены реализацией четырех ключевых программных продуктов: *Evince*, *Breeze*, *Breeze runtime* и *Breeze geo*. В таблице 1.1 приведены отличительные функциональные особенности данных программных средств.

Таблица 1.1 – Сравнительный анализ программных средств компании *Predictera*

	<i>Evince</i>	<i>Breeze</i>	<i>Breeze runtime</i>	<i>Breeze geo</i>
Функционал	Визуализация и анализ ГСИ	Разработка приложений для анализа ГСИ в реальном времени	Средство для интеграции разработанных в Breeze приложений	Визуализация и анализ ГСИ
Краткое описание	Позволяет работать с ГСИ различных форматов, визуализировать их и использовать встроенные алгоритмы для решения задач анализа ГСИ	Позволяет формировать конвейер обработки данных в реальном времени, включающий этапы получения данных (трансляция с камеры или файлы) и анализа данных с применением широкого набора алгоритмов	Упрощает интеграцию конвейера обработки данных в производственные процессы	Предоставляет специализированный набор алгоритмов для анализа ГСИ, предназначенный для проведения геологической разведки и других задач в геологической сфере
Область применения	Любая	Любая	Производственные процессы	Геология

Основным отличием продуктов компании *Prediktera* является различие в доступных алгоритмах анализа ГСИ. В *Evince* доступны следующие алгоритмы: анализ главных компонент (PCA), регрессия наименьших квадратов (PLS), дискриминантный анализ на основе регрессии наименьших квадратов (PLS-DA), метод мягкого независимого моделирования аналогий классов, метод спектрального угла (SAM), а также алгоритмы, используемые для предобработки данных. Основное преимущество *Evince* – это широкие возможности визуализации ГСИ.

Другие программные продукты – *Breeze*, *Breeze Runtime* и *Breeze Geo* – основаны на едином вычислительном ядре и имеют расширенный набор доступных алгоритмов анализа ГСИ. Эти средства поддерживают широкий спектр алгоритмов машинного обучения, включая нейронные сети, деревья решений, метод опорных векторов, случайный лес, логистическую и линейную регрессию и другие. Кроме того, данные программные средства поддерживают использование пользовательских нейросетевых моделей в формате ONNX. Также для реализации конвейеров обработки данных предусмотрена поддержка нейросетевых моделей для сегментации изображений, таких как YOLOv4 [83], YOLOv5, Fast R-CNN [84].

Программное средство *Breeze Geo*, помимо перечисленных возможностей, имеет дополнительный набор алгоритмов (полный перечень которых не приводится на сайте разработчика), специально предназначенных для решения задач в области геологии. На рисунках 1.5 и 1.6 приведены примеры графического интерфейса программ *Evince* и *Breeze* соответственно.

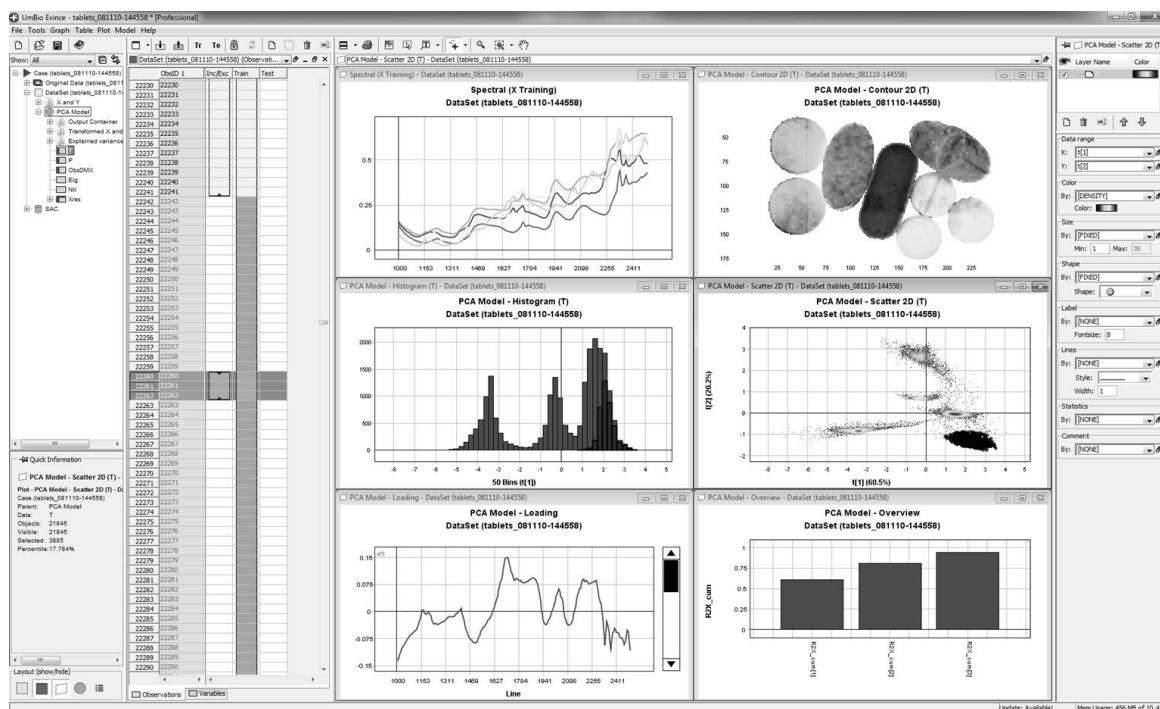


Рисунок 1.5 – Графический интерфейс программы *Evince*

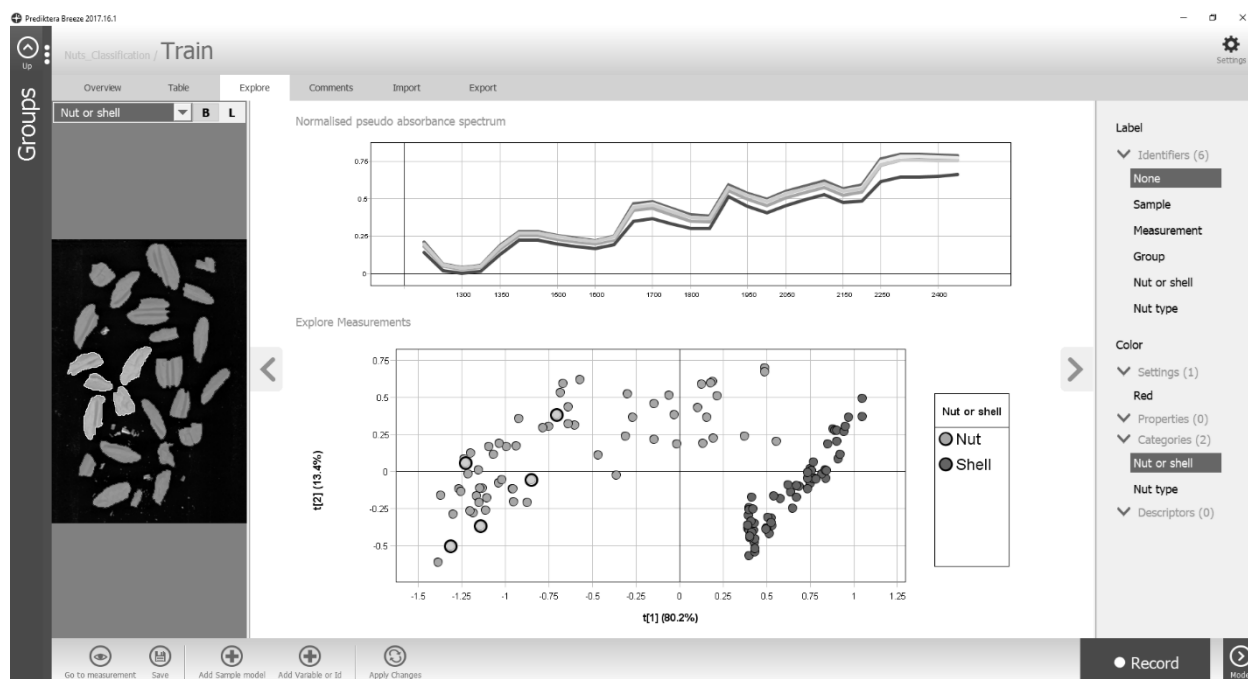


Рисунок 1.6 – Графический интерфейс программы *Breeze*

Несмотря на широкий спектр возможностей, программные средства Prediktera имеют ряд ограничений:

- программные продукты ориентированы преимущественно на настольное использование и требуют установки на локальный компьютер;
- работа с данными и алгоритмами анализа зависит от конкретного продукта линейки Prediktera, что ограничивает универсальность пользовательского сценария;
- поддержка спектральных сенсоров ограничена перечнем совместимых камер и оборудования;
- средства не предназначены для полноценной организации централизованного хранения ГСИ как самостоятельной функции системы;
- несмотря на наличие развитых средств машинного обучения, пользователь должен обладать специальными компетенциями для выбора методов, настройки параметров и интерпретации результатов;
- автоматизированный анализ гиперспектральных изображений без предварительной разметки и без участия специалиста представлен ограниченно;
- возможность внедрения пользовательских моделей реализована частично и зависит от поддерживаемых форматов и сценариев использования.

Таким образом, программные средства Prediktera обладают развитыми инструментами анализа и могут применяться в промышленных и исследовательских задачах. Однако они не полностью решают проблему доступного облачного анализа гиперспектральных изображений, ориентированного как на пользователей без специальных компетенций, так и на исследователей, которым требуется гибкое внедрение собственных алгоритмов и моделей.

3. *SpectraSENS*

SpectraSENS [85] разработан компанией *Clyde HSI*, производителем спектральных сенсоров. Эта система предназначена для решения задач

получения ГСИ, их анализа и визуализации. Основной целью разработки *SpectraSENS* было создание программного средства для работы с ГСИ, полученными с помощью сенсоров собственной разработки компании. *SpectraSENS* распространяется в трех версиях:

- 1) базовой (basic);
- 2) стандартной (standard);
- 3) расширенной (advanced).

Основным различием между версиями является набор доступных алгоритмов, включенных в разные версии систем.

SpectraSENS в расширенной версии обладает широким спектром функций. Программное обеспечение включает ряд специальных методов, позволяющих выполнять предобработку и визуализацию ГСИ, полученных с помощью спектрального сенсора: настройка спектрального разрешения, регулировка количества кадров в секунду, настройка параметров цветокоррекции, освещения, экспозиции и баланса белого.

Во всех версиях программного обеспечения предусмотрены различные способы визуализации ГСИ, включая отображение отдельных спектральных полос, визуализацию значений спектра пикселей и формирование RGB-изображений из комбинаций спектральных полос.

Для анализа ГСИ в *SpectraSENS* предлагается использовать разнообразные алгоритмы. Основные из них включают:

- метод главных компонент;
- метод спектрального угла;
- метод опорных векторов;
- линейное спектральное разложение.

Помимо алгоритмов анализа, разработчики предоставляют обширный набор вспомогательных методов обработки ГСИ, таких как геометрическая коррекция, формирование масок, калибровка значений и другие.

Несмотря на обширные функциональные возможности, SpectraSENS обладает следующими ограничениями:

- программная система ориентирована преимущественно на работу со спектральными сенсорами собственной разработки Clyde HSI;
- применение SpectraSENS к произвольным ГСИ ограничено, что снижает универсальность;
- программное средство не обеспечивает полноценную облачную модель доступа и требует работы в рамках заданной программно-аппаратной среды;
- отсутствуют средства централизованного хранения и каталогизации ГСИ, независимые от конкретного сенсора;
- набор алгоритмов анализа ограничен преимущественно классическими методами обработки ГСИ;
- отсутствуют современные нейросетевые методы анализа изображений;
- не предусмотрена полноценная возможность подключения пользовательских алгоритмов и заранее обученных моделей;
- автоматизированный анализ данных без предварительной разметки представлен ограниченно;
- не поддерживаются методы анализа данных с использованием спектральных индексов.

Таким образом, SpectraSENS является специализированным программным обеспечением для работы с оборудованием конкретного производителя. Его возможности полезны при эксплуатации соответствующих сенсоров, однако программное средство ограничено при решении задач универсального анализа произвольных ГСИ, облачного доступа, расширяемости и автоматизации анализа для пользователей без специальных компетенций.

Далее на рисунке 1.7 представлена демонстрация интерфейса *SpectraSENS*.

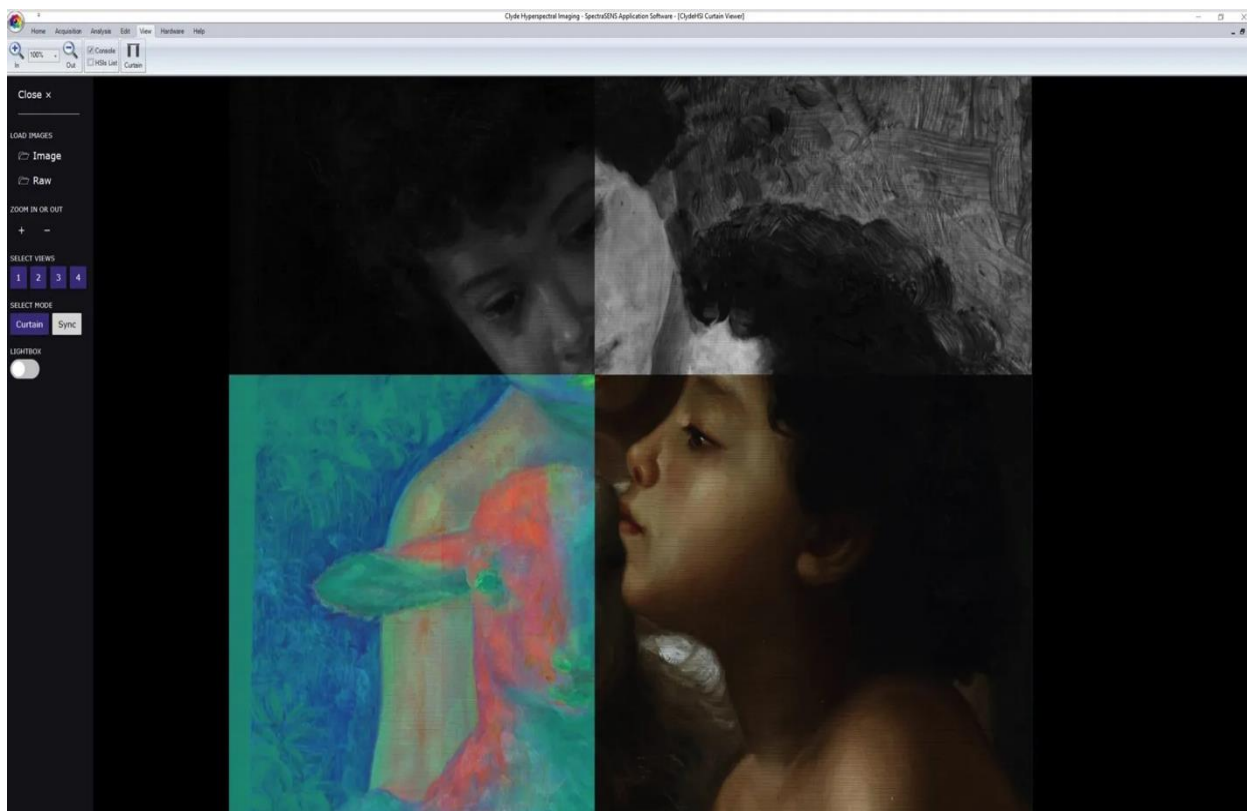


Рисунок 1.7 – Пользовательский интерфейс программы SpectraSENS

4. СПЕКТРА

Цифровая платформа «СПЕКТРА» предназначена для сбора, визуализации, обработки и анализа ГСИ. Платформа включает набор функциональных модулей, обеспечивающих выполнение основных этапов обработки изображений: получение данных, калибровку, геометрическую коррекцию, уменьшение размерности, отбор спектральных признаков, кластеризацию, сегментацию и классификацию.

В составе платформы предусмотрено использование как классических методов анализа гиперспектральных изображений, так и методов машинного обучения. Для уменьшения размерности и отбора признаков рассматриваются методы PCA, MNF и NMF. Для кластеризации и сегментации используются алгоритмы K-means и ISODATA. Для классификации и спектрального анализа

применяются методы SAM, MLC, SVM, Spectral Unmixing, многослойный перцептрон и нейронные сети.

Особенностью платформы является ее связь с аппаратными средствами получения гиперспектральных данных. В описании ПО рассматривается работа с гиперспектральной камерой Specim IQ, включая получение исходных данных, калибровку, преобразование значений в отражательную способность и последующий анализ спектров. Это позволяет рассматривать платформу не только как средство обработки готовых гиперспектральных изображений, но и как элемент программно-аппаратного комплекса, ориентированного на полный цикл работы с гиперспектральными данными.

Основное ограничение платформы состоит в её реализации. Как и рассмотренные ранее решения, для использования «СПЕКТРА» необходима установка соответствующего программного обеспечения. Несмотря на широкий набор алгоритмов, данная платформа не поддерживает расширение добавлением новых алгоритмов или моделей машинного обучения. Для исследовательских задач это существенный критерий, поскольку фиксированный набор алгоритмов ограничивает применимость платформы к решению новых задач.

Далее в таблице 1.2 приведено сравнение функциональных возможностей и особенностей рассмотренных программных систем.

Таблица 1.2 – Сравнительный анализ функциональных возможностей программных систем анализа гиперспектральных изображений

<i>Свойства</i>	<i>Hyper-tools</i>	<i>Predictera</i>	<i>SpectraSENS</i>	<i>СПЕКТРА</i>	<i>Ожидание</i>
Доступность и способ эксплуатации					
Облачная модель доступа	–	–	–	–	+
Возможность совместной работы	–	–	–		+
Управление данными ГСИ					
Работа с произвольными СИ	+	■	–	■	+
Централизованное хранение данных	–	■	■	■	+

Продолжение таблицы 1.2

<i>Свойства</i>	<i>Hyper-tools</i>	<i>Predictera</i>	<i>SpectraSENS</i>	<i>СПЕКТРА</i>	<i>Ожидание</i>
Визуализация данных					
Интерактивный интерфейс	+	+	+	+	+
Составление изображений из полос СИ	—	—	—	■	+
Анализ данных					
Методы уменьшения размерности	+	+	+	+	+
Методы кластеризации	+	+	+	+	+
Спектральные индексы	—	■	■	+	+
Нейросетевые методы	—	+	—	+	+
Классические алгоритмы МО	+	+	—	+	+
Расширяемость					
Добавление пользовательских алгоритмов	—	+	—	—	+
Добавление пользовательских моделей	—	■	—	—	+
Обучение новых моделей	—	—	—	+	+
Сохранение и повторное использование моделей	—	—	—	+	+
«+» - доступно, «■» – доступно условно или неполностью, «—» – недоступно					

В результате сравнения существующих программных систем можно сделать вывод, что каждое из рассмотренных решений обладает отдельными сильными сторонами, однако ни одно из них не обеспечивает полного набора возможностей, необходимых для универсального облачного анализа гиперспектральных изображений.

HYPER-Tools предоставляет широкий набор методов обработки и анализа данных, однако является расширением MATLAB и ориентирован на локальную работу пользователя, обладающего специальными компетенциями. Он не решает задачи централизованного хранения данных, не обеспечивает

облачного доступа и не предусматривает гибкого внедрения пользовательских моделей и алгоритмов.

Программные продукты Prediktera обладают наиболее развитым набором инструментов анализа, включая методы машинного обучения и поддержку отдельных сценариев использования нейросетевых моделей. Вместе с тем данные средства также ориентированы преимущественно на настольное применение и профессиональные сценарии работы. Их использование требует от пользователя понимания методов анализа гиперспектральных данных, а возможности расширения зависят от поддерживаемых форматов, версий продуктов и предусмотренных разработчиком механизмов интеграции.

SpectraSENS предоставляет удобные средства получения, визуализации и анализа гиперспектральных данных, однако его применение в значительной степени связано с оборудованием конкретного производителя. Это ограничивает использование данной системы для анализа произвольных гиперспектральных изображений и снижает его универсальность в исследовательских задачах.

СПЕКТРА ориентирована на полный цикл работы с ГСИ и включает средства сбора, калибровки, геопривязки и анализа ГСИ. Однако ПО требует установки на рабочее место пользователя. Также данный продукт имеет ограниченные возможности по функциональному расширению. Это ограничивает применение платформы СПЕКТРА как универсального средства облачного автоматизированного анализа произвольных гиперспектральных изображений.

1.2.3 Требования к программному обеспечению

Проведённый обзор программных систем анализа ГСИ показывает, что существующие решения позволяют выполнять отдельные операции визуализации, обработки и анализа данных, однако не обеспечивают в полной мере выполнение полного цикла работы с гиперспектральными

изображениями в единой программной среде. Основные ограничения рассмотренных программных средств связаны с преобладанием настольной модели использования, зависимостью от локального оборудования, недостаточной поддержкой централизованного хранения данных, ограниченной автоматизацией анализа и недостаточной гибкостью при подключении новых алгоритмов и моделей машинного обучения.

В связи с этим программное обеспечение автоматизированного анализа ГСИ должно удовлетворять ряду требований, обусловленных особенностями данных, сценариями их использования и ограничениями существующих программных систем.

Программное обеспечение должно обеспечивать доступность работы с ГСИ независимо от операционной системы, характеристик пользовательского устройства и места нахождения пользователя. Для этого целесообразно использовать облачную модель доступа, при которой основные операции хранения, обработки и анализа выполняются на стороне серверной инфраструктуры, а пользователь взаимодействует с системой через веб-интерфейс. Такой подход позволяет снизить требования к локальному оборудованию, обеспечить возможность работы с маломощными и мобильных устройств, а также упростить совместное использование данных и результатов анализа.

Программное обеспечение должно поддерживать централизованное хранение ГСИ и результатов их обработки. ГСИ имеют значительный объём и могут поступать из различных источников, включая спутниковые системы, беспилотные летательные аппараты и специализированные спектральные сенсоры. Поэтому система должна обеспечивать загрузку, хранение и повторное использование ГСИ, а также доступ к ранее полученным результатам обработки. Наличие централизованного хранилища позволяет использовать систему не только как средство анализа отдельных изображений, но и как инфраструктурную платформу для организации коллекций ГСИ.

Программное обеспечение должно поддерживать работу с ГСИ различного происхождения и размерности. Это требование связано с тем, что ГСИ могут отличаться количеством спектральных каналов, пространственным разрешением, форматом хранения и характеристиками сенсора. Следовательно, система должна быть применима к произвольным ГСИ и не должна быть жёстко связан с конкретным производителем оборудования, типом сенсора или узкой предметной областью.

Программное обеспечение должно предоставлять средства интерактивной визуализации ГСИ. Визуализация является необходимым этапом первичного анализа, контроля качества данных и интерпретации результатов обработки. Система должна обеспечивать отображение отдельных спектральных слоёв, формирование изображений из комбинаций спектральных каналов, визуализацию спектров выбранных пикселей и областей, построение распределений значений, а также отображение результатов алгоритмов анализа в виде тепловых карт, бинарных масок и карт сегментации.

Программное обеспечение должно включать средства автоматизированного анализа ГСИ. Пользователь не должен быть вынужден самостоятельно выполнять все этапы обработки, включая выбор алгоритма, подбор параметров, выбор спектральных каналов и интерпретацию промежуточных результатов. Поэтому система должна предоставлять алгоритмы, позволяющие автоматизировать ключевые этапы анализа: снижение размерности, выбор информативных признаков, построение спектральных индексов, кластеризацию, сегментацию и формирование предварительной разметки данных.

Программное обеспечение должно обеспечивать выполнение ресурсоёмких вычислительных операций без блокировки пользовательского интерфейса. Обработка ГСИ может требовать значительных вычислительных ресурсов и времени, особенно при работе с изображениями высокого

пространственного и спектрального разрешения. Поэтому выполнение алгоритмов анализа, загрузка крупных данных, формирование изображений и построение карт разметки должны быть организованы как асинхронные вычислительные задачи с возможностью отслеживания их состояния и получения результата после завершения обработки.

Программное обеспечение должно обеспечивать эффективную работу с большими объёмами ГСИ. Для этого необходимо предусмотреть методы частичного чтения данных и обработки изображений, повторного использования ранее вычисленных результатов и оптимизации хранения. Такие механизмы позволяют снизить нагрузку на оперативную память, уменьшить объём передаваемых данных и повысить отзывчивость пользовательского интерфейса при работе с изображениями большого размера.

Программное обеспечение должно обладать расширяемостью. Область анализа ГСИ активно развивается, поэтому система не должна быть ограничена фиксированным набором встроенных алгоритмов. Необходимо предусмотреть возможность подключения пользовательских алгоритмов, обученных моделей машинного обучения и новых вычислительных модулей без изменения основного ядра системы. Это требование особенно важно для исследовательских задач, в которых требуется оперативная апробация новых методов анализа данных.

Таким образом, программное обеспечение автоматизированного анализа ГСИ должно представлять собой расширяемую облачную программную среду, объединяющую централизованное хранение данных, интерактивную визуализацию, запуск алгоритмов анализа и управление результатами обработки. Такая среда должна обеспечивать доступность обработки ГСИ с различных устройств, поддерживать данные различной размерности и происхождения, автоматизировать основные этапы анализа, эффективно выполнять ресурсоёмкие вычисления и допускать расширение

функциональности за счёт подключения новых алгоритмов и моделей машинного обучения.

Сформулированные требования к программному обеспечению определяют необходимость разработки интеллектуальной системы, архитектура которой должна обеспечивать облачную модель доступа, модульность, масштабируемость, асинхронную обработку данных, централизованное хранение гиперспектральных изображений и унифицированную интеграцию вычислительных модулей.

1.3 Основные задачи анализа гиперспектральных изображений

С появлением спектральных сенсоров возникла область науки, посвященная методам анализа ГСИ, называемая в англоязычной литературе *Hyperspectral Imaging (HSI)*. Основу анализа ГСИ составляют их особенности: высокое пространственное и спектральное разрешение; содержание уникальных наборов характеристик на каждой спектральной полосе. Учет данных особенностей ГСИ позволяет существенно улучшать точность решаемых задач. Рассмотрим далее основные задачи в области анализа ГСИ.

1.3.1 Задача попиксельной классификации (сегментации)

Математически, задача попиксельной классификации ГСИ (далее эта задача также будет называться задачей сегментации) может быть формализована следующим образом.

Пусть $X \in \mathbb{R}^{m \times n \times p}$ – это гиперспектральное изображение, где m и n – определяют пространственное разрешение изображения, его высоту и ширину, а p – количество спектральных полос. В таком случае каждый пиксель ГСИ представляет собой численный вектор с размерностью p .

Пусть $Y \in \mathbb{N}^{m \times n}$ – это матрица с действительными метками пикселей, где $Y(i, j)$ – это метка класса пикселя расположенного в i строке и j столбце.

Таким образом, задача классификации ГСИ формируется как поиск отображения:

$$f: \mathbb{R}^p \rightarrow \mathbb{N}.$$

Оптимальный классификатор f^* определяется как функция, минимизирующая функцию потерь:

$$f^* = \arg \min_f \text{Loss}(f(X), Y),$$

где Loss – измеряющая отклонение предсказаний классификатора f от истинных меток Y .

1. Спектральные индексы

Задача попиксельной классификации ГСИ, заключающаяся в поиске отображения f , может решаться различными методами, одним из которых является использование *спектральных индексов*. Спектральные индексы представляют собой показатели, вычисляемые на основе значений отдельных спектральных полос, и позволяют выделить определенные свойства объектов на изображении.

В интернете существует база данных [86], посвященная сбору известных спектральных индексов и сенсоров. Такие индексы, как правило, проектируются вручную, учитывая физические особенности отражения световых волн от различных поверхностей. Например, выделяется отдельный класс *вегетационных индексов*, которые используются для оценки состояния растений, в частности их фотосинтетической активности.

Примером такого индекса является нормализованный разностный вегетационный индекс (NDVI) [87], который рассчитывается по формуле:

$$NDVI = \frac{NIR - RED}{NIR + RED},$$

где NIR – слой изображения, соответствующий длине волны в диапазоне 0,7-1,0 мкм;

RED – слой изображения, соответствующий длине волны в диапазоне 0,6-0,7 мкм.

Значения NDVI лежат в диапазоне от -1 до 1: отрицательные значения указывают на мертвую растительность, значения от 0 до 0,4 – на больные

растения, а от 0,4 до 1 – на здоровую растительность. Визуализация индекса NDVI представлена на рисунке 1.8.

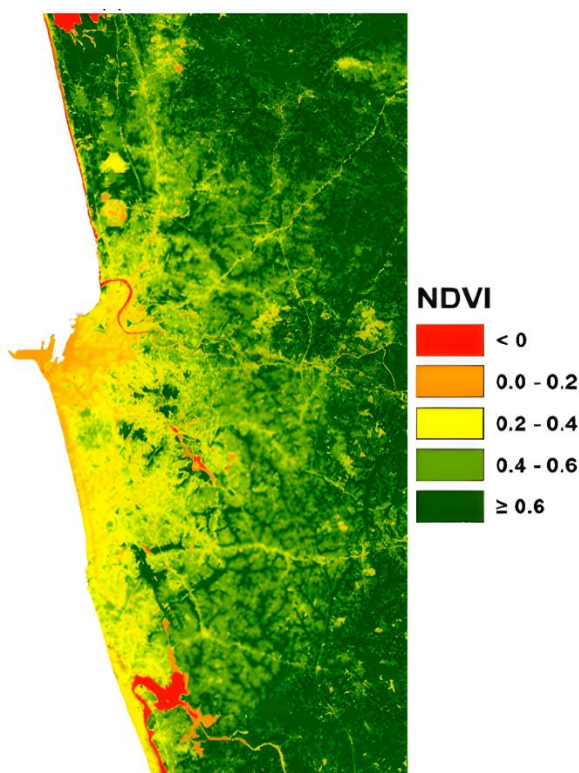


Рисунок 1.8 – Визуализация индекса NDVI

Использование NDVI позволяет быстро оценить состояние растительности: зеленые области соответствуют здоровым растениям, а красные – подверженным неблагоприятным воздействиям, таким как болезни или нехватка влаги.

Однако применение индексов для оценки новых, ранее неизвестных свойств целевых объектов может быть затруднено. Для создания индексов требуется глубокое понимание физических свойств материалов и контроль множества переменных. Во-первых, необходимо определить, какие *спектральные полосы* наиболее информативны для анализа конкретного свойства. Во-вторых, составляемый индекс должен учитывать *влияние внешних факторов*, например облачность, что усложняет выбор формулы индекса. В-третьих, необходимо учитывать возможные *различия в характеристиках* спектральных сенсоров

2. Метод опорных векторов

Метод опорных векторов (SVM) – метод машинного обучения с учителем, который широко применяется для решения задач классификации и регрессии данных с большим числом признаков, таких как ГСИ [88]. Основное преимущество SVM заключается в его способности находить гиперплоскости, которые наилучшим образом разделяют классы в пространстве признаков.

Для иллюстрации рассмотрим линейный SVM на примере задачи бинарной классификации. Пусть имеется следующая обучающая выборка с элементами (x_i, y_i) , где x_i – вектор признак, соответствующий i пикселю ГСИ; а y_i – метка класса, $y_i \in \{1, -1\}$. Тогда гиперплоскость может быть описана следующим выражением

$$w^T x + b = 0,$$

где w – вектор весовых коэффициентов;

b – смещение.

Цель алгоритма SVM нахождение таких значений w и b , которые определяют гиперплоскость, оптимально разделяющую объекты разных классов. Ключевым свойством SVM является *максимизация отступа* между классами, то есть расстояния между гиперплоскостью и ближайшими объектами классов. Ширина полосы, представляющая два отступа, равна $\frac{2}{\|w\|}$. Таким образом, задача SVM сводится к минимизации $\|w\|^2$, или максимизации отступа при условии, что *объекты не должны находиться внутри полосы*. Формально, задача построения оптимальной гиперплоскости выглядит следующим образом:

$$\begin{cases} \|w\|^2 \rightarrow \min \\ y_i(w^T x_i + b) \geq 1, \forall i \end{cases}$$

В случае линейного SVM искомое отображение f можно записать как:

$$f(x) = \text{sign}(w^T x + b),$$

где функция sign возвращает метку класса: 1 или -1 .

Для многоклассового случая, SVM используется для решения частных задач “один против всех” или “один против одного”. В этом случае отображение f определяется как:

$$f(x) = \arg \max_i (w_i^T x + b_i),$$

где i – индекс класса, w_i и b_i параметры гиперплоскости для i -го класса.

Каждая гиперплоскость оценивает принадлежность к одному из классов, и класс с наибольшим значением используется для окончательного решения.

SVM является одним из популярных методов анализа ГСИ по ряду причин:

- SVM хорошо подходит для данных с большим числом признаков, что характерно для ГСИ;
- SVM расширяет возможности классификации за счет применения ядерных функций, которые позволяют эффективно разделять данные в нелинейных пространствах;
- SVM может эффективно работать даже с небольшими наборами данных и устойчив к переобучению.

Кроме того, решения, найденные SVM, демонстрируют устойчивость к зашумленным данным и выбросам, а также обладают хорошей интерпретируемостью, что делает метод практичным для широкого спектра задач классификации в анализе ГСИ.

3. Метод k-ближайших соседей

Для решения задачи поискельной классификации ГСИ одним из простых и интуитивно понятных методов является метод k-ближайших соседей (k-NN). Этот метод предполагает, что элементы пространства признаков группируются в кластеры, и для определения класса пикселя ГСИ достаточно вычислить расстояние до ближайших размеченных пикселей (соседей).

Мера расстояния между двумя точками, x и y , в пространстве признаков может определяться, например, через Евклидово расстояние:

$$d(x, y) = \sqrt{\sum_{i=1}^p (x_i - y_i)^2},$$

где p – размерность признакового пространства.

Или с помощью расстояния Махаланобиса:

$$d_M(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)},$$

где S - ковариационная матрица признаков, учитывающая корреляции между признаками, что делает это расстояние более чувствительным к форме распределения данных.

Пусть x – пиксель, для которого необходимо определить класс. Вычислив расстояния от x до всех размеченных пикселей изображения по выбранной метрике, выбираются k ближайших соседей $\{x_1, x_2, \dots, x_k\}$, с минимальными значениями расстояния. Класс пикселя x определяется по правилу большинства соседей.

Искомое отображение f для метода ближайших соседей можно записать следующим образом:

$$f(x) = \operatorname{argmax}_{j \in \{1, 2, \dots, m\}} \sum_{i=1}^k \delta_{C(x_i)C_j},$$

где $C(x_i)$ – класс i -го ближайшего соседа,

C_j – j класс из набора классов,

$\delta_{C(x_i)C_j}$ - символ Кронекера, равный 1, если $C(x_i) = C_j$, и 0 в противном случае.

Метод k -ближайших соседей часто используется для анализа ГСИ из-за своей простоты и интуитивности. Однако, его применение ограничено в задачах с шумом и выбросами, так как он чувствителен к таким искажениям, что может снижать точность классификации в сложных сценариях.

4. Деревья решений

Для решения задачи попиксельной классификации ГСИ применяются различные методы, основанные на деревьях решений. Деревья решений – это методы машинного обучения, которые разделяют пространство признаков на

области, определяемые сложными, нелинейными правилами. Высокое спектральное разрешение данных, характерное для ГСИ, позволяет деревьям решений формировать более сложные правила классификации, так как наличие большого числа спектральных полос (признаков) даёт возможность учитывать мельчайшие различия в спектральных характеристиках объектов, что улучшает способность алгоритма различать классы. Математически задача классификации деревьями решений сводится к построению отображения, которое можно описать рекурсивно через последовательность условий, разделяющих пространство признаков на подмножества, где:

$$f(x) = \sum_{i=1}^L C_i I(x \in R_i),$$

где L – количество листьев (конечных узлов) дерева,

C_i – предсказание класса для i -го листа,

R_i – область пространства признаков, соответствующая i -му листу,

I – индикаторная функция, равная 1, если x принадлежит области R_i , и 0 иначе.

Благодаря этому подходу деревья решений способны формировать сложные, нелинейные правила, особенно когда используются в ансамблях, таких как случайный лес, который объединяет множество деревьев для повышения общей устойчивости к переобучению.

Случайный лес, который представляет собой ансамбль деревьев решений, улучшает устойчивость к переобучению и повышает точность классификации за счёт случайного подбора признаков и образцов при построении каждого дерева. Этот метод также устойчив к выбросам, так как не все деревья используют все данные, что снижает влияние аномальных значений.

Однако, применение деревьев решений и их ансамблей для анализа ГСИ сопряжено с рядом трудностей:

– деревья решений требуют значительных вычислительных ресурсов для построения, особенно когда количество признаков (спектральных полос) велико, что увеличивает количество возможных разбиений;

– при несбалансированности классов дерево может создавать смещенные модели, отдавая предпочтение доминирующим классам;

– случайные леса и другие ансамбли требуют точной настройки гиперпараметров, таких как количество деревьев, глубина деревьев и критерии разбиения, что может потребовать значительных усилий и ресурсов.

5. Метод *к-средних*

Метод кластеризации *к-средних* часто применяется для анализа ГСИ [89], особенно на этапе подготовки данных [90]. Этот метод позволяет разделить пространство признаков на k кластеров, определяя группы пикселей с похожими спектральными характеристиками. Основными преимуществами метода являются его простота и вычислительная эффективность, что делает его привлекательным для обработки больших объемов данных. Метод *к-средних* относится к методам машинного обучения без учителя, что позволяет использовать его для сегментации неразмеченных ГСИ, помогая выявить структуры в данных.

Данный метод работает путем минимизации суммарного квадратичного отклонения объектов от центроидов их кластеров:

$$\min_{\{C_j\}_{j=1}^k} \sum_{j=1}^k \sum_{x \in R_j} \|x - C_j\|^2,$$

где $C_j \in \mathbb{R}^p$ – центроид j -го класса,

R_j – множество точек, принадлежащих кластеру j ,

$\|\cdot\|$ – Евклидова норма.

Алгоритм начинается с инициализации центроидов C_j , после чего итеративно перераспределяет точки между кластерами, минимизируя

расстояние до ближайших центроидов, и обновляет центроиды на основе среднего положения точек в каждом кластере.

Несмотря на свою простоту и эффективность, метод *k*-средних обладает рядом недостатков:

- *чувствительность к выбросам и шуму*: наличие выбросов может значительно сместить центроиды, что ухудшает качество кластеризации.

- *инициализация центроидов*: начальная позиция центроидов влияет на конечное решение, и плохая инициализация может привести к неоптимальным результатам.

- *выбор числа кластеров k* : определение оптимального числа кластеров является нетривиальной задачей и требует дополнительных методов оценки.

Эти факторы делают метод *k*-средних полезным инструментом для начальной кластеризации данных, но его эффективность сильно зависит от условий и качества входных данных.

6. Нейросетевые алгоритмы

Другим крупным классом алгоритмов машинного обучения, часто применяемых для решения задачи попиксельной классификации и семантической сегментации ГСИ, являются нейросетевые алгоритмы. Описание всех разновидностей нейросетевых алгоритмов и их особенностей выходит за рамки данного исследования, поэтому в данном пункте будут подчеркнуты основные аспекты применения нейронных сетей для анализа ГСИ.

Популярность нейронных сетей обусловлена их высокой точностью, способностью к обобщению и относительной простотой применения. Нейронные сети успешно используются для анализа ГСИ, решая задачи попиксельной классификации и семантической сегментации [91, 92]. Для анализа ГСИ применяются различные архитектуры сверточных нейронных сетей (CNN) [93, 94], включая модели с 3D-свертками [95, 96], которые учитывают пространственно-спектральные взаимосвязи в данных. Также

используются адаптации существующих архитектур, предназначенных для анализа монохромных или RGB изображений [97].

Основные сложности в применении нейронных сетей для анализа ГСИ связаны с высокой размерностью изображений, которая приводит к ряду проблем: нехватке видеопамяти, препятствующая обучению; и риску переобучения, так как ГСИ имеют высокое пространственное и спектральное разрешение. Для решения этих проблем часто используются методы предобработки, такие как уменьшение размерности с помощью метода главных компонент (РСА), который позволяет снизить размерность признакового пространства, сохраняя важные спектральные особенности данных. Согласно исследованиям, применение РСА позволяет успешно обучать нейронные сети для задач семантической сегментации ГСИ [98].

Несмотря на существующие трудности, связанные с применением нейронных сетей для анализа ГСИ, этот подход остается одним из самых перспективных направлений в данной области. Высокий потенциал нейросетевых алгоритмов позволяет достигать высокой точности классификации, сохраняя способность моделей к обобщению, что делает их ключевым инструментом для решения сложных задач анализа ГСИ.

1.3.2 Задача выбора информативных слоев

Задача выбора информативных слоев заключается в определении такого множества спектральных полос, которое представляет наиболее релевантную информацию для решения задачи анализа ГСИ. Решение этой задачи помогает уменьшить спектральную размерность изображения, что особенно важно при применении ресурсоемких вычислительных алгоритмов. Кроме того, выбор информативных слоев может способствовать созданию новых индексов, часто используемых для анализа ГСИ.

Пусть $X \in \mathbb{R}^{m \times n \times p}$ – это ГСИ, где m и n – пространственные размеры, а p – количество спектральных полос. Цель выбора информативных слоев – это получить такое подмножество $B \subseteq \{1, 2, \dots, p\}$, которое позволит сократить

размерность исходного гиперспектрального изображения до X_B , состоящего только из информативных слоев. Важным условием является *минимизация мощности* множества B . Определим также функцию выбора (критерий), которая производит качественную оценку набора слоев $f: 2^{\{1,2,\dots,p\}} \rightarrow \mathbb{R}$, где $2^{\{1,2,\dots,p\}}$ – множество всех подмножеств. Тогда задача выбора оптимального подмножества слоев формулируется как задача оптимизации:

$$B^* = \underset{B \subseteq \{1,2,\dots,p\}, |B| \leq k}{\operatorname{argmax}} f(B),$$

где B^* – найденное оптимальное подмножество информативных слоев;
 k – заданное число необходимых слоев ($k \ll p$).

Функция $f(B)$ обычно выбирается с учетом следующих критериев:

- максимизация количества информации (дисперсии) в выбранных слоях;
- максимизация точности классификатора, обученного на выбранных слоях;
- максимизация спектрального разнообразия, при котором спектральные полосы максимально отличаются друг от друга по своим спектральным характеристикам.

Основной проблемой в решении данной задачи является её комбинаторная сложность, что делает задачу NP-полной в худшем случае. Это требует использования различных эвристических и аппроксимирующих алгоритмов. Также выбор подходящего критерия является важным аспектом решения задачи.

Для решения задачи выбора информативных слоев применяются различные методы, включая те, что используются для классификации и сегментации ГСИ, такие как нейросетевые алгоритмы [99, 100], методы, основанные на k -средних [101], на методе опорных векторов [102], DBSCAN [103] и т.д.

1.3.3 Требования к математическому обеспечению

Проведённый обзор методов анализа ГСИ показывает, что математическое обеспечение программной системы, автоматизирующей анализ должно удовлетворять ряду требований, обусловленных высокой размерностью ГСИ, ограниченностью размеченных выборок, необходимостью снижения участия пользователя в выборе алгоритмов и параметров обработки, а также потребностью в получении интерпретируемых результатов.

Во-первых, математическое обеспечение должно быть применимо к ГСИ с произвольным числом спектральных каналов и учитывать пространственно-спектральную природу данных. Во-вторых, оно должно обеспечивать снижение размерности или выбор информативных спектральных признаков, поскольку использование всех каналов изображения приводит к росту вычислительных затрат и повышает риск переобучения моделей. В-третьих, алгоритмы должны автоматизировать выбор спектральных каналов, индексов и признаков, минимизируя необходимость ручной настройки со стороны пользователя.

Существенным требованием является возможность работы в условиях ограниченной или отсутствующей экспертной разметки. Это связано с высокой трудоёмкостью формирования обучающих выборок для ГСИ. Поэтому математическое обеспечение должно включать методы, способные использовать малое число размеченных наблюдений, а также методы автоматического или предварительного формирования разметки.

Кроме того, алгоритмы должны поддерживать решение как бинарных, так и многоклассовых задач сегментации, использовать формализованные критерии информативности и качества результата, а также формировать интерпретируемые выходные данные: спектральные индексы, тепловые карты, бинарные маски и карты классов. Для применения в интеллектуальной системе алгоритмы должны быть вычислительно эффективными, допускать

векторно-матричную или параллельную реализацию, а также интегрироваться в программную систему через унифицированный интерфейс вычислительных модулей.

1.4 Выводы и результаты по первому разделу

В первом разделе рассмотрены особенности гиперспектральных изображений, области их применения, основные задачи анализа, а также существующие программные системы для работы с гиперспектральными данными.

Показано, что ГСИ обладают высокой информативностью и применяются в дистанционном зондировании Земли, сельском хозяйстве, геологии, медицине, промышленности и других областях. При этом их обработка связана с рядом ограничений: высокой размерностью данных, большим объёмом хранимой информации, чувствительностью к условиям съёмки, недостатком размеченных наборов данных и необходимостью адаптации существующих методов анализа изображений.

Рассмотрены основные задачи анализа ГСИ, включая сегментацию, классификацию, кластеризацию, выбор информативных спектральных слоёв и построение спектральных индексов. Установлено, что существующие методы часто требуют ручного выбора алгоритмов, настройки параметров и наличия экспертной разметки, что ограничивает их применение в автоматизированных программных системах.

На основе проведённого анализа сформулированы требования к математическому обеспечению. Оно должно обеспечивать обработку ГСИ произвольной размерности, выбор информативных признаков, автоматизацию выбора спектральных каналов и индексов, поддержку бинарной и многоклассовой классификации, работу при ограниченной или отсутствующей разметке, интерпретируемость результатов и вычислительную эффективность.

Выполнен обзор программных систем HYPER-Tools, Prediktera и SpectraSENS, СПЕКТРА. Показано, что данные решения обладают отдельными функциональными преимуществами, однако не обеспечивают полного набора возможностей для универсального облачного анализа ГСИ. Их основные ограничения связаны с настольной моделью использования, зависимостью от локального оборудования, недостаточной автоматизацией анализа, ограниченной поддержкой централизованного хранения и слабой расширяемостью.

На основе сравнения программных систем сформулированы требования к программному обеспечению. Оно должно обеспечивать облачный доступ, централизованное хранение данных, интерактивную визуализацию, поддержку гиперспектральных изображений различного происхождения и размерности, асинхронное выполнение ресурсоёмких операций, сохранение результатов анализа и возможность подключения пользовательских алгоритмов и моделей машинного обучения.

Таким образом, результатом первого раздела является система требований к математическому и программному обеспечению автоматизированного анализа ГСИ. Эти требования определяют необходимость разработки алгоритмов анализа, методов хранения и визуализации данных, а также расширяемой облачной интеллектуальной системы, реализующей полный цикл работы с ГСИ.

2 Разработка архитектуры и расширяемой интеллектуальной системы обработки гиперспектральных изображений

Для реализации расширяемой интеллектуальной системы обработки ГСИ необходимо сформировать архитектуру, учитывающую ряд *архитектурно значимых требований (ASRs)*. На основе сформированных на этапе обзора системы требований к программному обеспечению, сформируем ASR следующим образом:

1) система должна предусматривать возможность *функционального расширения* благодаря интеграции вычислительных модулей (плагинов), реализующих алгоритмы анализа ГСИ;

2) система должна предусматривать возможность *горизонтального масштабирования* компонентов системы при росте числа пользователей и объема вычислительных задач;

3) система должна реализовывать *облачную* модель использования, обеспечивающую доступ к системе с устройств, имеющих подключение к сети интернет, без необходимости выполнения ресурсоемких вычислений на стороне пользователя.

Указанные требования определяют необходимость использования архитектурного подхода, обеспечивающего слабую связанность компонентов, независимое масштабирование вычислительных подсистем, асинхронное выполнение длительных операций и возможность контейнеризированного развертывания.

Для удовлетворения ASRs архитектура проектируемой системы в своей основе опирается на микросервисный подход, в рамках которого система представляется как совокупность взаимодействующих компонентов, отвечающих за отдельные функциональные области и допускающих независимую разработку, развертывание и масштабирование. Функциональное расширение системы обеспечивается за счет подключения пользовательских вычислительных модулей, реализующих алгоритмы

анализа гиперспектральных изображений и взаимодействующих с системой через формализованные программные интерфейсы. При таком подходе базовые компоненты системы обеспечивают хранение данных, управление вычислительными задачами, передачу входных параметров, получение результатов обработки и взаимодействие с пользователем, а алгоритмы анализа разрабатываются и разворачиваются как самостоятельные вычислительные компоненты. Это позволяет расширять функциональность системы без изменения существующих сервисов хранения, визуализации и управления задачами.

Представленный далее перечень функциональных и нефункциональных требований является достаточным для проектирования архитектуры системы, поскольку охватывает основные группы требований, определяющие его назначение, способы взаимодействия пользователей с системой, механизмы расширения функциональности, требования к выполнению вычислительных задач, масштабированию, развертыванию, отказоустойчивости и пользовательскому интерфейсу.

2.1 Функциональные требования

К архитектурно значимым требованиям системы относится возможность расширения набора алгоритмов анализа гиперспектральных изображений за счет подключения вычислительных модулей. Данное требование конкретизируется вариантами использования UC12-UC15, описывающими добавление, удаление, получение списка и выполнение алгоритмов. Сценарии UC16-UC18 описывают автоматизированную обработку ГСИ за счет возможности автоматического получения разметки изображения и последующего обучения моделей машинного обучения.

Далее в таблице 2.1 приведено описание участников взаимодействия с системой. В таблице 2.2 представлено детальное описание основных вариантов использования, реализованных в системе.

Таблица 2.1 – Описание акторов системы

№	Имя	Описание
1	Название	A1 - Пользователь
	Тип	Основной актор
	Описание	Загружает гиперспектральные изображения, просматривает и визуализирует данные, запускает алгоритмы обработки, просматривает и анализирует результаты.
2	Название	A2 – Разработчик алгоритмов
	Тип	Основной \ вспомогательный актор
	Описание	Подготавливает и регистрирует пользовательские вычислительные модули, содержащие алгоритмы анализа ГСИ или модели машинного обучения.
3	Название	A3 – Внешний реестр контейнеров
	Тип	Внешняя система
	Описание	Используется для получения контейнерных образов пользовательских алгоритмов и моделей машинного обучения при их подключении к системе.

Таблица 2.2 – Детальное описание вариантов использования

№	Имя	Описание
1	Название	UC1- Добавление ГСИ в систему
	Описание	Система позволяет добавить гиперспектральное изображение, представленное в виде трёхмерного тензора, изображения, видеозаписи или набора файлов поддерживаемого формата. Переданные данные преобразуются во внутреннее представление, используемое в системе, после чего сохраняются данные изображения и связанная с ним метайнформация.
	Цель	Добавить в систему новое ГСИ
	Акторы	
	1	A1 (Пользователь)
	Предусловия	A1 имеет файл или набор файлов с данными гиперспектрального изображения
	Постусловие	Гиперспектральное изображение добавлено в систему и доступно для просмотра, визуализации и обработки
	1	A1 выбирает файл или набор файлов с данными гиперспектрального изображения
	2	A1 заполняет форму, указывая название изображения, описание, параметры нормировки, порядок слоёв, теги и индексы RGB-слоёв

№	Имя	Описание
	3	А1 отправляет форму в систему
	4	Система проверяет корректность переданных данных и поддерживаемость формата
	5	Система выполняет чтение и преобразование данных во внутреннее представление
	6	Система сохраняет данные изображения и связанную метаинформацию.
	7	Система отображает пользователю информацию об успешном добавлении изображения
	Альтернативы	
	А 4.1	Если часть необязательных параметров не указана, система формирует их автоматически или использует значения по умолчанию
	Исключения	
	Е 2.1	При обнаружении ошибок в ходе заполнения формы, для А1 блокируется возможность её отправки и указываются ошибки; после их исправления форма становится вновь доступна
	Е 3.1	В случае ошибок при попытке чтения или некорректности формата переданных данных А1 уведомляется об этом, дальнейшее выполнение останавливается
2	Название	UC2-Изменение параметров и метаинформации ГСИ
	Описание	Система позволяет изменять параметры гиперспектрального изображения и связанную с ним метаинформацию: название, описание, теги, вариант нормировки, порядок слоёв, индексы RGB-слоёв и ориентацию изображения в пространстве
	Цель	Изменить параметры или метаинформацию выбранного ГСИ
	Акторы	
	1	А1 (Пользователь)
	Предусловие	Целевое ГСИ добавлено в систему
	Постусловие	Параметры целевого изображения успешно изменены
	1	А1 выбирает гиперспектральное изображение
	2	А1 изменяет необходимые параметры или метаинформацию через форму редактирования
	3	А1 отправляет форму в систему
	4	Система проверяет корректность переданных данных
	5	Система сохраняет изменённые параметры
	6	Система уведомляет пользователя об успешном изменении данных
	Альтернативы	
	А 5.1	При отсутствии фактических изменений – обновление объекта в базе данных не происходит
	Исключения	
	Е 3.1	При обнаружении ошибок в ходе заполнения формы, для А1 блокируется возможность её отправки и указываются ошибки; после их исправления форма становится доступна
	Название	UC3-Получение списка ГСИ

№	Имя	Описание
3	Описание	Система позволяет получить список гиперспектральных изображений, загруженных в систему, с возможностью фильтрации и сортировки
	Цель	Получить список доступных гиперспектральных изображений
	Акторы	
	1	A1 (Пользователь)
	Предусловие	Пользователь имеет доступ к системе
	Постусловие	Пользователь получает список изображений, удовлетворяющих заданным критериям
	1	A1 формирует запрос к системе указывая или не указывая необходимые фильтры и порядок выдачи
	2	Система формирует список изображений в соответствии с заданными критериями
	3	Система возвращает сформированный список A1
	Альтернативы	
	A 3.1	Если фильтры не заданы, система возвращает список изображений в порядке по умолчанию
	Исключения	
	E 3.1	Если изображения, удовлетворяющие критериям, отсутствуют, система возвращает пустой список
4	Название	UC4-Удаление ГСИ
	Описание	Система позволяет удалить выбранное гиперспектральное изображение и связанные с ним данные
	Цель	Удалить заданное ГСИ из системы
	Акторы	
	1	A1 (Пользователь)
	Предусловие	В системе имеется загруженное гиперспектральное изображение
	Постусловие	Изображение и связанные с ним данные удалены или помечены как удалённые
	1	A1 выбирает гиперспектральное изображение
	2	A1 отправляет запрос на удаление изображения
	3	Система проверяет наличие изображения
	4	Система удаляет запись об изображении и связанные с ним данные либо помечает их как удалённые
	5	Система уведомляет пользователя об успешном удалении
	Альтернативы	
	A 3.1	Если изображение уже удалено, система возвращает пользователю сообщение об отсутствии необходимости повторного удаления
5	Название	UC5-Скачивание ГСИ
	Описание	Система позволяет скачать гиперспектральное изображение в виде архива, содержащего данные изображения и связанную метаинформацию
	Цель	Получить локальную копию гиперспектрального изображения и его метаинформации

№	Имя	Описание
	Акторы	
	1	A1 (Пользователь)
	Предусловие	В системе имеется целевое гиперспектральное изображение
	Постусловие	Пользователь получил архив с данными изображения и метаданной
	1	A1 выбирает гиперспектральное изображение
	2	A1 отправляет запрос на скачивание
	3	Система формирует архив с данными изображения и метаданными
	4	Система передаёт архив пользователю
6	Название	УС6-Вычисление известного спектрального индекса
	Описание	Система позволяет вычислить известный спектральный или вегетационный индекс для выбранного гиперспектрального изображения
	Цель	Получить слой, соответствующий результату вычисления выбранного индекса
	Акторы	
	1	A1 (Пользователь)
	Предусловие	В системе имеется загруженное гиперспектральное изображение; системе известна формула выбранного индекса
	Постусловие	Результат вычисления индекса сформирован и доступен пользователю
	1	A1 выбирает гиперспектральное изображение
	2	A1 выбирает спектральный индекс
	3	A1 задаёт необходимые параметры вычисления
	4	Система формирует результат вычисления ВИ и возвращает его A1
	Альтернативы	
	A 3.1	Если часть параметров не указана, система использует значения по умолчанию
	Исключения	
	E 3.1	Если изображение не найдено, система уведомляет пользователя об ошибке
	E 3.2	Если для изображения отсутствует информация о соответствии слоёв длинам волн, система уведомляет пользователя о невозможности вычисления индекса
7	Название	УС7-Получение списка вычисленных слоёв
	Описание	Система позволяет получить список ранее вычисленных слоёв, связанных с выбранным гиперспектральным изображением
	Цель	Получить список вычисленных слоёв для просмотра, повторного использования или включения в модель представления
	Акторы	
	1	A1 (Пользователь)

№	Имя	Описание
	Предусловие	В системе имеется загруженное гиперспектральное изображение
	Постусловие	Пользователь получил список вычисленных слоёв
	1	A1 выбирает гиперспектральное изображение
	2	A1 отправляет запрос на получение списка вычисленных слоёв
	3	Система формирует список ранее вычисленных слоёв
	4	Система возвращает пользователю сформированный список
	Альтернативы	
	А 3.1	Если вычисленные слои отсутствуют, система возвращает пустой список
8	Название	UC8-Формирование модели представления
	Описание	Система позволяет сформировать визуальное представление гиперспектрального изображения на основе выбранных исходных или вычисленных слоёв с заданными параметрами отображения
	Цель	Получить модель представления для визуального анализа гиперспектрального изображения и результатов его обработки.
	Акторы	
	1	A1 (Пользователь)
	Предусловие	В системе имеется загруженное гиперспектральное изображение; доступны вычисленные слои
	Постусловие	Пользователь получил сформированную модель представления
	1	A1 выбирает гиперспектральное изображение
	2	A1 выбирает исходные или вычисленные слои, которые необходимо включить в модель представления
	3	A1 задаёт параметры отображения слоёв: цвет, диапазон значений, параметры бинаризации, вариант эквализации и порядок наложения
	4	A1 отправляет конфигурацию модели представления в систему
	5	Система проверяет доступность выбранных слоёв
	6	Система формирует модель представления с учётом заданных параметров
	7	Система возвращает пользователю результат визуализации и параметры сформированной модели
	Альтернативы	
	А 6.1	Если отдельный слой недоступен, система исключает его из модели представления и уведомляет пользователя
	А 6.2	Если часть параметров не задана, система использует значения по умолчанию
	Исключения	
	Е 5.1	Если ни один из выбранных слоёв не может быть использован, система уведомляет пользователя о невозможности формирования модели представления

№	Имя	Описание
9	Название	UC9-Экспорт конфигурации модели представления
	Описание	Система позволяет сохранить конфигурацию модели представления в текстовом формате для последующего повторного использования
	Цель	Получить текстовое представление конфигурации модели представления
	Акторы	
	1	A1 (Пользователь)
	Предусловие	Пользователь сформировал конфигурацию модели представления
	Постусловие	Пользователь получил закодированную текстовую строку, содержащую конфигурацию модели представления
	1	A1 формирует конфигурацию модели представления
	2	A1 отправляет запрос на экспорт конфигурации
	3	Система преобразует конфигурацию в текстовое представление
	4	Система возвращает пользователю закодированную конфигурацию
	Исключения	
	Е 3.1	Если конфигурация модели представления некорректна, система уведомляет пользователя о невозможности экспорта
10	Название	UC10-Импорт конфигурации модели представления
	Описание	Система позволяет восстановить конфигурацию модели представления из текстовой строки
	Цель	Получить конфигурацию модели представления на основе ранее экспортированного текстового описания
	Акторы	
	1	A1 (Пользователь)
	Предусловие	Пользователь имеет текстовое описание конфигурации модели представления
	Постусловие	Пользователь получил восстановленную конфигурацию модели представления
	1	A1 передаёт в систему текстовое описание конфигурации модели представления
	2	Система выполняет проверку и декодирование строки
	3	Система восстанавливает конфигурацию модели представления
	4	Система возвращает пользователю восстановленную конфигурацию
	Исключения	
	Е 2.1	Если строка не может быть декодирована или содержит некорректную конфигурацию, система уведомляет пользователя о невозможности импорта
11	Название	UC11-Формирование информативного спектрального индекса

№	Имя	Описание
	Описание	Система позволяет сформировать информативный спектральный индекс, предназначенный для выделения целевых объектов на ГСИ в рамках пользовательской задачи
	Цель	Получить информативный спектральный индекс, различающий заданные классы объектов
	Акторы	
	1	A1 (Пользователь)
	Предусловие	В системе имеется ГСИ; пользователь задал или выбрал данные, необходимые для формирования индекса
	Постусловие	Информативный спектральный индекс сформирован, сохранён и доступен пользователю для анализа
	1	A1 выбирает гиперспектральное изображение
	2	A1 задаёт целевые классы объектов или исходные данные для их выделения
	3	A1 отправляет запрос на формирование информативного индекса
	4	Система проверяет корректность запроса и наличие необходимых данных
	5	Система создаёт вычислительную задачу
	6	Внутренние компоненты системы выполняют формирование информативного индекса
	7	Система сохраняет сформированный индекс как вычисленный слой
	8	Система уведомляет пользователя о завершении вычислений
	Альтернативы	
	A 4.1	Если результат с такими параметрами уже существует, система возвращает ранее сформированный результат
	A 5.1	Если задача с такими параметрами уже выполняется, система уведомляет пользователя о текущем состоянии задачи
	Исключения	
	E 4.1	Если индекс не может быть сформирован из-за некорректных исходных данных или ошибки выполнения, система фиксирует ошибку и уведомляет пользователя
12	Название	UC12-Регистрация пользовательского алгоритма
	Описание	Система позволяет расширить набор доступных алгоритмов анализа гиперспектральных изображений за счёт регистрации пользовательского вычислительного модуля
	Цель	Добавить в систему новый алгоритм анализа ГСИ или модель машинного обучения
	Акторы	
	1	A2 (Разработчик алгоритмов)
		A3 (Внешний реестр контейнеров)
	Предусловие	Разработчик алгоритмов подготовил вычислительный модуль и разместил его контейнерный образ во внешнем реестре контейнеров
	Постусловие	В системе зарегистрирован новый алгоритм, доступный для проверки и последующего выполнения.

№	Имя	Описание
	1	A2 заполняет форму регистрации алгоритма
	2	A2 указывает название, описание, тип алгоритма, параметры запуска и сведения о контейнерном образе
	3	Система проверяет корректность переданного описания алгоритма
	4	Система обращается к A3 для проверки доступности контейнерного образа
	5	Система инициирует развертывание и регистрацию алгоритма
	6	Система отображает A2 результат регистрации алгоритма
	Альтернативы	
	A 3.1	Если алгоритм с такими параметрами уже зарегистрирован, система уведомляет A2 о наличии соответствующей записи
	Исключения	
	B 3.1	Если описание алгоритма некорректно, система возвращает сообщение об ошибке
	B 4.1	Если сведения о контейнерном образе отсутствуют или заданы некорректно, регистрация алгоритма не выполняется
13	Название	UC13-Удаление пользовательского алгоритма
	Описание	Система позволяет удалить или деактивировать зарегистрированный пользовательский алгоритм. Удаление алгоритма не приводит к удалению ранее полученных результатов анализа
	Цель	Исключить алгоритм из списка доступных для выполнения
	Акторы	
	1	A2 (Разработчик алгоритмов)
	Предусловие	В системе зарегистрирован пользовательский алгоритм
	Постусловие	Алгоритм удалён или деактивирован; ранее полученные результаты анализа сохранены
	1	A2 выбирает зарегистрированный алгоритм
	2	A2 отправляет запрос на удаление или деактивацию алгоритма
	3	Система проверяет наличие алгоритма
	4	Система удаляет или деактивирует сведения об алгоритме
	5	Система уведомляет A2 об успешном выполнении операции
	Альтернативы	
	A 3.1	Если алгоритм уже деактивирован, система уведомляет A2 об отсутствии необходимости повторного действия
	Исключения	
	B 3.1	Если алгоритм участвует в активной вычислительной задаче, система отклоняет удаление или выполняет его после завершения задачи
14	Название	UC14-Получение списка доступных алгоритмов
	Описание	Система позволяет получить список зарегистрированных алгоритмов анализа гиперспектральных изображений и сведения об их доступности

№	Имя	Описание
	Цель	Получить список алгоритмов, которые могут быть использованы для обработки данных или администрирования набора вычислительных модулей.
	Акторы	
	1	A1 (Пользователь)
	2	A2 (Разработчик алгоритмов)
	Предусловие	В системе имеются зарегистрированные алгоритмы или пользователь имеет доступ к разделу алгоритмов
	Постусловие	Актор получил список алгоритмов и сведения об их состоянии
	1	Актор отправляет запрос на получение списка алгоритмов
	2	Система формирует список зарегистрированных алгоритмов
	3	Система дополняет список сведениями о доступности алгоритмов и их основных параметрах
	4	Система возвращает сформированный список актору
	Альтернативы	
	A 2.1	Для A1 система может возвращать только алгоритмы, доступные для выполнения
	A 3.1	Для A2 система может возвращать расширенную информацию о зарегистрированных алгоритмах, включая сведения о контейнерных образах и состоянии проверки
	Исключения	
	B 2.1	Если алгоритмы отсутствуют, система возвращает пустой список
15	Название	UC15-Выполнение алгоритма анализа гиперспектрального изображения
	Описание	Система позволяет пользователю выполнить доступный алгоритм обработки или анализа для выбранного гиперспектрального изображения
	Цель	Получить результат обработки ГСИ с использованием выбранного алгоритма
	Акторы	
	1	A1 (Пользователь)
	Предусловие	В системе имеется загруженное гиперспектральное изображение; выбранный алгоритм зарегистрирован и доступен для выполнения
	Постусловие	Результат выполнения алгоритма сохранён в системе и доступен пользователю для просмотра или повторного использования
	1	A1 выбирает гиперспектральное изображение
	2	Система отображает список доступных алгоритмов
	3	A1 выбирает алгоритм обработки или анализа
	4	A1 задаёт параметры выполнения алгоритма
	5	A1 отправляет запрос на выполнение алгоритма
	6	Система проверяет корректность параметров и наличие ранее сформированного результата

№	Имя	Описание
	7	При отсутствии готового результата система создаёт вычислительную задачу
	8	Система уведомляет пользователя о начале выполнения задачи
	9	Внутренние компоненты системы выполняют задачу и сохраняют результат
	10	Система уведомляет А1 о завершении обработки
	11	А1 получает возможность просмотреть результат
	Альтернативы	
	А 6.1	Если результат с заданными параметрами уже существует, система возвращает ранее сформированный результат без повторного вычисления
	А 6.2	Если задача с заданными параметрами уже выполняется, система сообщает пользователю её текущее состояние
	А 7.1	Если часть параметров не указана, система использует значения по умолчанию, если это допустимо для выбранного алгоритма
	Исключения	
	Е 9.1	Если выполнение алгоритма завершилось ошибкой, система фиксирует причину ошибки и уведомляет пользователя
	Е 9.2	Если выполнение задачи было прервано из-за отказа вычислительного модуля, система уведомляет пользователя о невозможности получения результата и предлагает повторно сформировать запрос после восстановления доступности алгоритма
16	Название	UC16-Автоматизированный анализ ГСИ с использованием A-MINDI
	Описание	Система позволяет выполнить автоматизированный анализ гиперспектрального изображения с использованием алгоритма A-MINDI без необходимости предварительной ручной разметки всего изображения
	Цель	Получить результат автоматизированного анализа ГСИ в виде вычисленного слоя, набора слоёв или модели представления
	Акторы	
	1	А1 (Пользователь)
	Предусловие	В системе имеется загруженное гиперспектральное изображение; алгоритм A-MINDI зарегистрирован и доступен для выполнения
	Постусловие	Результат автоматизированного анализа сохранён в системе и доступен пользователю для просмотра
	1	А1 выбирает гиперспектральное изображение для анализа
	2	А1 выбирает алгоритм A-MINDI из списка доступных алгоритмов
	3	А1 задаёт параметры автоматизированного анализа или оставляет значения по умолчанию
	4	А1 отправляет запрос на выполнение анализа
	5	Система проверяет наличие выбранного изображения и доступность алгоритма A-MINDI

№	Имя	Описание
	6	Система проверяет наличие ранее сформированного результата с заданными параметрами
	7	При отсутствии готового результата система создаёт вычислительную задачу
	8	Система уведомляет пользователя о начале выполнения задачи
	9	Внутренние компоненты системы выполняют автоматизированный анализ
	10	Система сохраняет результат анализа
	11	Система уведомляет A1 о завершении задачи
	12	A1 получает возможность просмотреть результат анализа.
	Альтернативы	
	A 6.1	Если результат с заданными параметрами уже существует, система возвращает ранее сформированный результат
	A 6.2	Если задача с заданными параметрами уже выполняется, система сообщает пользователю её текущее состояние
	A 7.1	Если параметры анализа не заданы явно, система использует значения по умолчанию
	Исключения	
	E 5.1	Если алгоритм A-MINDI недоступен, система уведомляет пользователя о невозможности выполнения анализа
	E 9.1	Если выполнение алгоритма завершилось ошибкой, система сохраняет информацию об ошибке и уведомляет пользователя
17	Название	UC17- Формирование обучающих данных с использованием A-MINDI
	Описание	Система позволяет использовать алгоритм A-MINDI для автоматизированного формирования разметки и обучающих данных, применяемых при последующем обучении моделей машинного обучения
	Цель	Получить обучающие данные, сформированные на основе автоматизированного анализа гиперспектрального изображения
	Акторы	
	1	A1 (Пользователь)
	Предусловие	В системе имеется загруженное гиперспектральное изображение; алгоритм A-MINDI зарегистрирован и доступен для выполнения
	Постусловие	Система сформировала и сохранила обучающие данные, которые могут быть использованы для обучения модели машинного обучения
	1	A1 выбирает гиперспектральное изображение
	2	A1 выбирает алгоритм A-MINDI как способ автоматизированного формирования разметки или обучающих данных
	3	A1 задаёт параметры формирования обучающих данных или оставляет значения по умолчанию
	4	A1 отправляет запрос на формирование обучающих данных

№	Имя	Описание
	5	Система проверяет наличие выбранного изображения и доступность алгоритма A-MINDI
	6	Система проверяет корректность параметров
	7	Система создаёт вычислительную задачу
	8	Внутренние компоненты системы выполняют автоматизированное формирование разметки и обучающих данных
	9	Система сохраняет сформированные обучающие данные и связанные параметры
	10	Система уведомляет A1 о завершении задачи
	11	A1 получает возможность просмотреть или использовать сформированные обучающие данные
	Альтернативы	
	A 6.1	Если результат применения A-MINDI с заданными параметрами уже существует, система использует ранее сформированные данные без повторного выполнения алгоритма
	A 6.2	Если задача с заданными параметрами уже выполняется, система уведомляет пользователя о её текущем состоянии
	A 7.1	Если параметры не заданы явно, система использует значения по умолчанию
	Исключения	
	E 5.1	Если алгоритм A-MINDI недоступен, система уведомляет пользователя о невозможности выполнения задачи
	E 8.1	Если формирование обучающих данных завершилось ошибкой, система сохраняет информацию об ошибке и уведомляет пользователя
18	Название	UC18- Обучение модели машинного обучения на основе сформированных обучающих данных
	Описание	Система позволяет выполнить обучение модели машинного обучения с использованием ранее сформированных обучающих данных, включая данные, полученные с применением A-MINDI
	Цель	Получить обученную модель машинного обучения для последующего анализа гиперспектральных изображений
	Актеры	
	1	A1 (Пользователь)
	Предусловие	В системе имеются обучающие данные; выбрана или задана модель машинного обучения; параметры обучения корректно определены
	Постусловие	Обученная модель сохранена в системе и доступна для последующего применения
	1	A1 выбирает ГСИ и источник разметки
	2	A1 выбирает существующую модель машинного обучения или задаёт параметры создаваемой модели
	3	A1 задаёт параметры обучения
	4	A1 отправляет запрос на обучение модели

№	Имя	Описание
	5	Система проверяет наличие обучающих данных и корректность параметров обучения
	6	Система создаёт вычислительную задачу
	7	Система уведомляет пользователя о начале обучения модели
	8	Внутренние компоненты системы выполняют обучение модели
	9	Система сохраняет обученную модель, параметры обучения и связанные результаты
	10	Система уведомляет пользователя о завершении обучения
	11	А1 получает возможность просмотреть сведения об обученной модели и использовать её для дальнейшего анализа.
	Альтернативы	
	А 6.1	Если часть параметров обучения не указана, система использует значения по умолчанию
	Исключения	
	Е 5.1	Если источник разметки не существует, система уведомляет пользователя об ошибке
	Е 5.2	Если параметры обучения заданы некорректно, система возвращает сообщение с описанием причины
	Е 8.1	Если обучение модели завершилось ошибкой, система сохраняет информацию об ошибке и уведомляет пользователя

Анализ функциональных требований показывает, что система должна обеспечивать три основные группы пользовательских сценариев: управление ГСИ, выполнение алгоритмов анализа и управление подключаемыми вычислительными модулями. При этом ряд сценариев предполагает выполнение ресурсоемких вычислительных операций, что обосновывает необходимость асинхронной обработки задач и использования планировщика. Сценарии UC12-UC15 обосновывают необходимость выделения механизма управления вычислительными модулями, обеспечивающего расширяемость программного комплекса.

2.2 Нефункциональные требования

Далее в рамках настоящего пункта перечислены нефункциональные требования, предъявляемые к интеллектуальной системе автоматизированной обработки ГСИ. Представленный перечень требований сформирован с учетом архитектурно значимых требований, определяющих структуру системы,

способы взаимодействия её компонентов, механизмы расширения функциональности, особенности развертывания и выполнения вычислительных задач.

1. Требования к производительности и асинхронной обработке.

1) Система должна выполнять длительные задачи анализа и обработки ГСИ в фоновом режиме, чтобы пользователь мог продолжать взаимодействие с системой после постановки задачи на выполнение.

2) Система должна поддерживать постановку вычислительных задач в очередь с последующим выполнением доступными вычислительными модулями.

3) Система должна обеспечивать возможность одновременного выполнения нескольких вычислительных задач при наличии доступных вычислительных ресурсов.

4) Система должна предотвращать повторное выполнение вычислительной задачи при наличии ранее сформированного результата с идентичными параметрами.

5) Система должна предотвращать повторную постановку вычислительной задачи, если задача с идентичными параметрами уже находится в очереди или выполняется.

6) Система должна сохранять информацию о состоянии вычислительных задач, включая факт постановки в очередь, начало выполнения, успешное завершение или завершение с ошибкой.

7) Система должна обеспечивать возможность уведомления пользователя об изменении состояния вычислительной задачи.

8) Система должна эффективно использовать доступные вычислительные ресурсы при выполнении задач анализа и обработки ГСИ.

2. Требования к масштабируемости.

1) Система должна предусматривать возможность горизонтального масштабирования компонентов, отвечающих за обработку пользовательских запросов.

2) Система должна предусматривать возможность горизонтального масштабирования компонентов, отвечающих за выполнение вычислительных задач.

3) Система должна поддерживать независимое масштабирование отдельных функциональных компонентов без необходимости масштабирования системы как единого монолитного приложения.

4) Система должна обеспечивать возможность увеличения количества экземпляров вычислительных модулей при росте количества вычислительных задач.

5) Система должна обеспечивать распределение вычислительных задач между доступными экземплярами вычислительных модулей.

6) Система должна сохранять работоспособность пользовательского интерфейса и базовых функций управления данными при высокой нагрузке на вычислительные компоненты.

7) Система должна быть спроектирована таким образом, чтобы увеличение количества пользователей не требовало изменения архитектуры системы.

3. Требования к доступности.

1) Система должна быть доступна пользователю через веб-интерфейс с устройств, имеющих доступ к сети Интернет.

2) Система не должна требовать установки специализированного программного обеспечения на пользовательское устройство для выполнения основных функций обработки, визуализации и управления ГСИ.

3) Система должна выполнять ресурсоемкие операции обработки ГСИ на стороне серверной инфраструктуры.

4) Система должна снижать зависимость пользователя от вычислительной мощности клиентского устройства.

5) Система должна обеспечивать возможность удаленного доступа к загруженным ГСИ, результатам анализа и вычисленным слоям.

6) Система должна обеспечивать возможность работы пользователя с ранее полученными результатами анализа без необходимости повторного выполнения соответствующих вычислений.

4. Требования к надежности и отказоустойчивости.

1) Система должна уведомлять пользователя о произошедших ошибках при выполнении пользовательских запросов и вычислительных задач.

2) Система должна сохранять информацию об ошибках, возникающих при выполнении вычислительных задач.

3) Система должна предотвращать цепные реакции сбоев при отказе отдельного компонента.

4) Система должна обеспечивать изоляцию вычислительных модулей от базовых компонентов программного комплекса.

5) Отказ пользовательского вычислительного модуля не должен приводить к отказу базовых функций системы.

6) Система должна обеспечивать возможность восстановления после сбоя отдельного компонента при развертывании в среде управления контейнерами.

7) Система должна корректно обрабатывать ситуацию недоступности вычислительного модуля.

8) Система должна корректно обрабатывать ситуацию недоступности планировщика задач или брокера сообщений.

9) Система должна обеспечивать сохранность метаинформации ГСИ и результатов вычислений при ошибках выполнения вычислительных задач.

10) Система должна обеспечивать целостность данных при операциях добавления, изменения и удаления ГСИ.

5. Требования к расширяемости и модифицируемости.

1) Система должна поддерживать функциональное расширение за счет подключения вычислительных модулей, реализующих алгоритмы анализа ГСИ.

2) Система должна позволять добавлять новые вычислительные модули без изменения исходного кода базовых компонентов программного комплекса.

3) Система должна позволять удалять или отключать вычислительные модули без нарушения работоспособности ранее полученных результатов анализа.

4) Система должна предоставлять формализованный интерфейс взаимодействия с вычислительными модулями.

5) Интерфейс вычислительных модулей должен быть специфицирован в рамках программного пакета на языке программирования Python.

6) Система должна поддерживать подключение вычислительного модуля с использованием общедоступной ссылки на Docker-образ.

7) Система должна обеспечивать возможность получения списка подключенных вычислительных модулей и их текущего состояния.

8) Система должна быть разработана с учетом принципов микросервисной архитектуры, обеспечивающей модульность и независимость компонентов.

6. Требования к взаимодействию компонентов.

1) Внутреннее взаимодействие компонентов системы, связанных с постановкой и выполнением вычислительных задач, должно быть реализовано с использованием очереди сообщений.

2) Для обмена сообщениями между компонентами системы должен использоваться подход RPC.

3) Для взаимодействия компонентов через очередь сообщений должен использоваться протокол AMQP [51].

4) Система должна обеспечивать слабую связанность компонентов, участвующих в выполнении вычислительных задач.

5) Система должна обеспечивать возможность асинхронного взаимодействия между компонентами, планировщиком задач и вычислительными модулями.

6) Система должна обеспечивать передачу в вычислительные модули данных, необходимых для выполнения алгоритмов анализа ГСИ.

7) Система должна обеспечивать прием результатов работы вычислительных модулей и их последующее сохранение.

8) Взаимодействие с внешними участниками должно быть реализовано с использованием REST-подхода.

9) Прикладной интерфейс системы должен быть описан в терминах спецификации OpenAPI.

7. Требования к пользовательскому интерфейсу.

1) Система должна предоставлять пользователю графический веб-интерфейс.

2) Графический интерфейс пользователя должен обеспечивать выполнение основных сценариев работы с ГСИ, вычисленными слоями, моделями представления и вычислительными модулями.

3) Система должна поддерживать обратную связь с пользователем посредством специализированной системы уведомлений.

4) Система должна уведомлять пользователя о постановке вычислительной задачи в очередь.

5) Система должна уведомлять пользователя о завершении вычислительной задачи.

6) Система должна уведомлять пользователя об ошибках, возникших при выполнении вычислительной задачи.

7) Графический интерфейс должен предупреждать пользователя о вводе некорректных данных в формах.

8. Ограничивающие требования.

1) Система не обязана производить содержательную проверку корректности пользовательских вычислительных модулей.

2) Ответственность за корректность реализации пользовательского вычислительного модуля возлагается на лицо, осуществляющее его разработку.

3) Система должна фиксировать факт неуспешного подключения или выполнения пользовательского вычислительного модуля.

4) Неуспешное подключение пользовательского вычислительного модуля не должно приводить к нарушению работоспособности базовых компонентов системы.

5) Удаление вычислительного модуля не должно приводить к удалению ранее полученных результатов анализа, сформированных с его использованием.

9. Требования к развертыванию и эксплуатации.

1) Система должна быть реализована с возможностью развертывания в контейнеризированной среде.

2) Компоненты системы должны быть подготовлены к развертыванию с использованием Docker-контейнеров.

3) Система должна быть реализована таким образом, чтобы ее можно было развернуть в автоматизированной системе управления контейнерами Kubernetes.

4) Система должна быть реализована таким образом, чтобы ее можно было развернуть с использованием утилиты docker-compose.

5) Развертывание посредством docker-compose должно обеспечивать возможность локального запуска системы для отладки, тестирования или эксплуатации в ограниченной инфраструктуре.

6) Развертывание в Kubernetes должно обеспечивать возможность масштабирования компонентов, управления их жизненным циклом и восстановления после сбоев.

7) Архитектура системы должна позволять развертывать вычислительные модули как отдельные контейнеризированные компоненты.

8) Система должна поддерживать возможность независимого обновления отдельных компонентов при условии сохранения совместимости их интерфейсов.

2.3 Подходы к проектированию архитектур программных систем анализа данных

Для реализации всех функциональных и нефункциональных требований особенно важно предугадать архитектурные решения, способствующие скорейшей и верной реализации программной системы. Разработка архитектуры расширяемой интеллектуальной системы обработки ГСИ базируется на наборе архитектурно значимых требований, обеспечивающих не только текущие потребности, но и возможности дальнейшего масштабирования и модернизации.

Современные тенденции проектирования архитектур чаще всего основаны на применении подходов, позволяющих разделять систему на независимые компоненты, которые легко заменять, обновлять и масштабировать. Архитектурами, основанными на применении данного подхода, являются: микросервисная архитектура [104] и сервис-ориентированная архитектура [105] (Service Oriented Architecture, SOA). Основное различие двух данных типов архитектур заключается в их реализации. Программные компоненты в микросервисной архитектуре являются независимыми, тогда как в SOA различные компоненты чаще всего обладают общими ресурсами. В рамках микросервисной архитектуры отдельные сервисы специализируются на различных задачах, взаимодействуя между собой через прикладной интерфейс или очереди сообщений. Кроме

упомянутого подхода, при реализации систем обработки потоковых данных в режиме реального времени часто применяют событийно-ориентированную архитектуру [106] (Event-Driven Architecture, EDA).

При проектировании архитектур анализа данных активно используются современные облачные инфраструктуры и “бессерверные”-вычисления [107] (serverless), что дает возможность масштабировать вычислительные ресурсы в зависимости от потребностей клиентов, высвобождая ресурсы при отсутствии задач. Подобные подходы существенно увеличивают надежность, производительность и удобство управления программными системами анализа данных. А также позволяют экономить средства при использовании облачных провайдеров.

Для реализации программных систем анализа данных часто применяют подход, заключающийся в организации масштабируемых кластеров, основанных на использовании технологий оркестрации контейнеризованных приложений [108]. Стандартом в области для формирования таких кластеров является применение инструмента Kubernetes [109]. Организация кластеров позволяет использовать распределенные вычислительные платформы, такие как Spark [110] и Ray [111]. Эти платформы обычно реализуют модель координатор-исполнитель, где центральный планировщик распределяет задачи на процессы-исполнители, обеспечивая эффективную обработку данных.

Современные тенденции в разработке систем анализа данных включают технологии контейнеризации и оркестровки, такие как Docker [112] и Kubernetes, что позволяет создавать гибкие и масштабируемые решения. Кроме того, активно применяется подход GitOps [113], упрощающий развертывание и управление жизненным циклом приложений, а также автоматическое внедрение новых алгоритмов анализа данных. Данные технологии и подходы способствуют созданию устойчивых, эффективных и легко расширяемых систем анализа данных.

Рассмотрим далее известные архитектуры систем анализа данных на предмет масштабируемости и расширяемости.

1. *Архитектура MOBDA: микросервисно-ориентированная система анализа больших данных для интеллектуальных транспортных систем* [114].

Архитектура **MOBDA** (Microservice-Oriented Big Data Architecture) разработана специально для интеллектуальных транспортных систем в городских условиях. Основная идея подхода заключается в сочетании потоковой и пакетной обработки данных в рамках единой гибридной микросервисной архитектуры.

Архитектура MOBDA отличается высокой горизонтальной масштабируемостью за счёт применения микросервисов, которые можно независимо развертывать и масштабировать в зависимости от нагрузки. Основное внимание авторы работы уделяют способам обработки данных. Потоковая обработка позволяет системе оперативно реагировать на изменения и адаптироваться к динамическому изменению входящих данных, что особенно актуально в условиях пиковых нагрузок на транспортные сети. Пакетная обработка позволяет использовать более сложные и трудоёмкие алгоритмы.

Использование микросервисного подхода может обеспечить легкость добавления новых функций и аналитических модулей без необходимости внесения кардинальных изменений в общую структуру. Авторы работы не описывают интерфейс, который бы позволил внедрять новые модули в систему и не исследуют возможность функционального расширения системы пользователями. Однако, при анализе способа межмодульного взаимодействия через брокер сообщений, становится ясно, что разработанную авторами систему можно функционально масштабировать, подключая новые модули к брокеру.

2. *Архитектура FedMicro-IDA: федерированное обучение на основе микросервисов для анализа данных IoT [115].*

Архитектура FedMicro-IDA представляет собой платформу, использующую федеративное обучение и микросервисы для анализа распределенных данных в IoT-приложениях.

Платформа, реализованная авторами, разделена на две зоны: облачные узлы (Cloud Node) и краевые узлы (Edge Node). Предложенная авторами архитектура обладает высокой масштабируемостью за счёт распределения вычислительных задач по краевым узлам платформы. Федеративное обучение позволяет уменьшить нагрузку на центральные серверы, распределяя обучение по множеству устройств с использованием трансферного обучения, что улучшает производительность и снижает задержки. Это особенно важно в IoT-системах, где данные генерируются в больших объемах и с высокой скоростью.

Микросервисная структура FedMicro-IDA обеспечивает легкую интеграцию новых аналитических сервисов и моделей машинного обучения. Для этого краевые узлы обращаются к облачным узлам для регистрации, откуда получают инструкции работы. Система позволяет быстро добавлять новые модели анализа данных без необходимости полной перестройки всей архитектуры. Благодаря контейнеризации, интеграция новых сервисов происходит просто и без риска нарушить работу существующих компонентов.

3. *Архитектура CLIC: кроссплатформенная система анализа данных с высокой расширяемостью и эффективностью [116].*

CLIC представляет собой платформу для кроссплатформенной обработки данных с помощью графовых сверточных нейронных сетей. Предложенная авторами система обеспечивает разделение задачи анализа данных на отдельные этапы (операторы), каждый из которых может выполняться на наиболее подходящей платформе (например, Apache Spark, PyTorch, TensorFlow). Использование графовых нейронных сетей позволяет

реализовать автоматический выбор оптимальной платформы для каждого оператора на основе структуры задачи. Разработанная система использует механизмы контейнеризации и управления через Kubernetes для развёртывания аналитических операторов на разных платформах, что позволяет интегрировать новые платформы или менять конфигурацию операторов в процессе работы.

Архитектура обеспечивает высокую масштабируемость благодаря применению контейнеризации и оркестрации с помощью Kubernetes. Расширяемость достигается за счёт модульного подхода: новые операторы и платформы легко интегрируются без изменения ядра системы или необходимости её полной перезагрузки, а отдельный микросервис реализует интерфейс, с помощью которого пользователи могут загружать собственные модели и задачи.

2.4 Архитектура расширяемой интеллектуальной системы обработки гиперспектральных изображений

Для удовлетворения ключевых архитектурных требований при разработке интеллектуальной системы обработки гиперспектральных изображений было принято решение использовать микросервисный подход к организации программных компонентов. Данный подход позволяет разделить функции управления исходными гиперспектральными данными, результатами анализа и визуализации, постановки вычислительных задач и выполнения алгоритмов на независимые компоненты, взаимодействующие между собой через формализованные программные интерфейсы.

Микросервисная архитектура позволяет решить задачи горизонтального масштабирования, облачного развёртывания и распределённого выполнения вычислительных задач. Разделение системы на отдельные сервисы, выполняющие независимые функции, обеспечивает гибкость масштабирования, поскольку нагрузка может быть распределена между несколькими серверами или контейнерами, размещёнными в облачной среде.

Такой подход упрощает управление вычислительными ресурсами, повышает устойчивость системы к отказам отдельных компонентов и позволяет обеспечить стабильную работу при увеличении числа пользователей и объёма обрабатываемых данных.

Функциональная расширяемость системы обеспечивается за счёт возможности интеграции пользовательских алгоритмов и моделей машинного обучения в виде самостоятельных вычислительных модулей. Такие модули могут быть подключены к системе через заданные интерфейсы взаимодействия и использовать общие механизмы постановки задач, передачи исходных данных, выполнения вычислений и сохранения результатов анализа. Это позволяет добавлять новые алгоритмы обработки гиперспектральных изображений без изменения принципов организации системы и без нарушения работы существующих компонентов.

Особенность предложенной архитектуры заключается в предметно-ориентированном разделении компонентов интеллектуальной системы обработки гиперспектральных изображений. В системе выделены микросервисы организации хранения ГСИ, микросервис базовой обработки ГСИ и микросервис управления алгоритмами обработки ГСИ. Такое разделение обеспечивает возможность независимого масштабирования компонентов, распределённого выполнения ресурсоёмких алгоритмов и расширения функциональности системы за счёт подключения новых пользовательских алгоритмов и моделей машинного обучения.

Для описания такой архитектуры требуется представить систему на нескольких уровнях детализации: как часть внешнего окружения, как совокупность контейнеров и как набор взаимодействующих компонентов. В связи с этим для последующего архитектурного описания используется модель С4 [117], позволяющая последовательно рассмотреть систему на различных уровнях абстракции: на уровне контекста, контейнеров и компонентов.

Использование C4-нотации в данном разделе обусловлено тем, что она ориентирована на описание архитектуры программных систем с различными уровнями абстракции и позволяет явно отразить границы системы, внешних участников, контейнеры, сервисы, базы данных, очереди сообщений и каналы взаимодействия между ними. В отличие от универсальной UML-нотации, C4 предоставляет более наглядный способ представления высокоуровневой архитектуры системы и лучше согласуется с задачей описания микросервисной и облачной реализации.

2.4.1 Контекстная диаграмма

На рисунке 2.5 изображена контекстная диаграмма, иллюстрирующая внешнее окружение системы и взаимосвязи её компонентов. Как можно видеть, пользователь взаимодействует с системой посредством специального веб-приложения. Данное веб-приложение предоставляет пользователю удобный графический интерфейс, предоставляющий ему возможность решать задачи анализа, обработки, хранения ГСИ. Веб-приложение и серверная часть системы коммуницируют посредством прокси-сервера, который маршрутизирует пользовательские запросы в отдельные части системы, а также выполняет балансировку запросов.

Для упрощения данной диаграммы на ней не показано взаимодействие микросервисов между собой. Подразумевается, что их взаимодействие происходит посредством протокола AMQP через менеджер сообщений. Далее на следующем уровне диаграммы данная связь будет отображена.

Взаимодействие с вычислительными модулями в рамках системы происходит через отдельный планировщик задач, который распределяет нагрузку между модулями и производит их мониторинг.

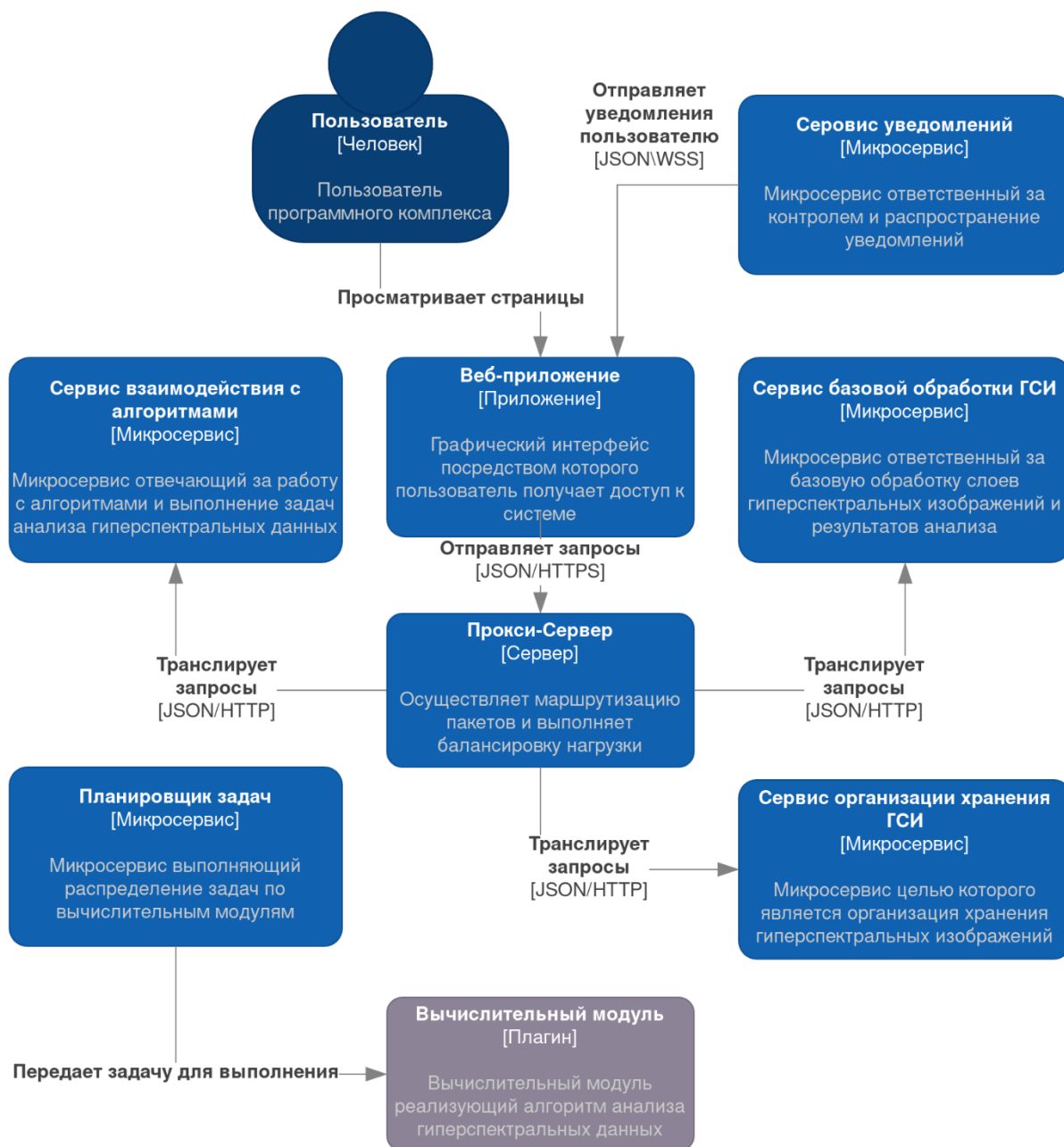


Рисунок 2.5 – Контекстная диаграмма интеллектуальной системы

2.4.2 Диаграмма контейнеров

На данном уровне диаграммы подробнее рассматриваются части системы. На рисунке 2.6 изображена диаграмма контейнеров, описывающая разработанную интеллектуальную систему. На рисунке видно, что основные микросервисы, отвечающие за бизнес-логику приложения содержат два контейнера и базу данных. Использование двух контейнеров объясняется необходимостью реализации взаимодействия между микросервисами. Так, один из контейнеров представляет интерфейс для работы по протоколу HTTP,

что используется для взаимодействия с пользователем. Другой контейнер представляет интерфейс для работы по протоколу AMQP, что используется для связи микросервисов между собой. Собственная база данных внутри микросервиса является стандартом реализации микросервисной архитектуры. В качестве системы управления базами данных в микросервисах системы используется документо-ориентированная СУБД. Данное решение обусловлено особенностями предметной области и архитектуры системы. Обработываемые системой сущности – ГСИ, вычисленные слои, результаты анализа, параметры алгоритмов, конфигурации моделей представления и вычислительные задачи – имеют сложную составную структуру и могут содержать вложенные, необязательные или изменяемые наборы полей. Использование документо-ориентированной модели хранения позволяет представлять такие сущности в виде документов, близких по структуре к объектам прикладной области и форматам обмена данными, используемым в системе.

Для организации взаимодействия между сервисами необходимо использовать какой-либо менеджер сообщений. В данном случае используется реализация RabbitMQ.

Система уведомлений производит публикацию уведомления пользователю посредством протокола WebSocket (WS), что позволяет отправлять уведомления в асинхронном режиме по мере их получения. Задачи анализа и обработки ГСИ производятся в рамках отдельных микросервисов. Все стандартные алгоритмы входящие в состав системы реализованы аналогичным образом, как реализуются пользовательские алгоритмы, расширяющие функциональность системы.

Для обеспечения масштабирования, отказоустойчивости и управления развертыванием используются следующие инфраструктурные компоненты:

– *kube-apiserver* – обеспечивает взаимодействие и управление всеми компонентами Kubernetes, позволяя гибко управлять состоянием и ресурсами кластера;

– *kube-scheduler* – распределяет задачи между доступными узлами кластера на основе текущих нагрузок и доступности ресурсов, обеспечивая равномерную нагрузку и эффективное использование ресурсов;

– *kube-controller-manager* – контролирует состояние системы, обеспечивая восстановление компонентов в случае отказов и поддержание заданного состояния микросервисов;

– *kube-proxy* – отвечает за сетевое взаимодействие внутри кластера, обеспечивая балансировку трафика между узлами и сервисами;

– *FluxCD* – обеспечивает непрерывную интеграцию и развертывание (GitOps), позволяя автоматизировать и ускорить процесс обновления и развертывания системы;

– *Grafana* – инструмент мониторинга состояния системы, предоставляющий визуальную информацию о работе кластера, производительности и возможных проблемах в режиме реального времени.

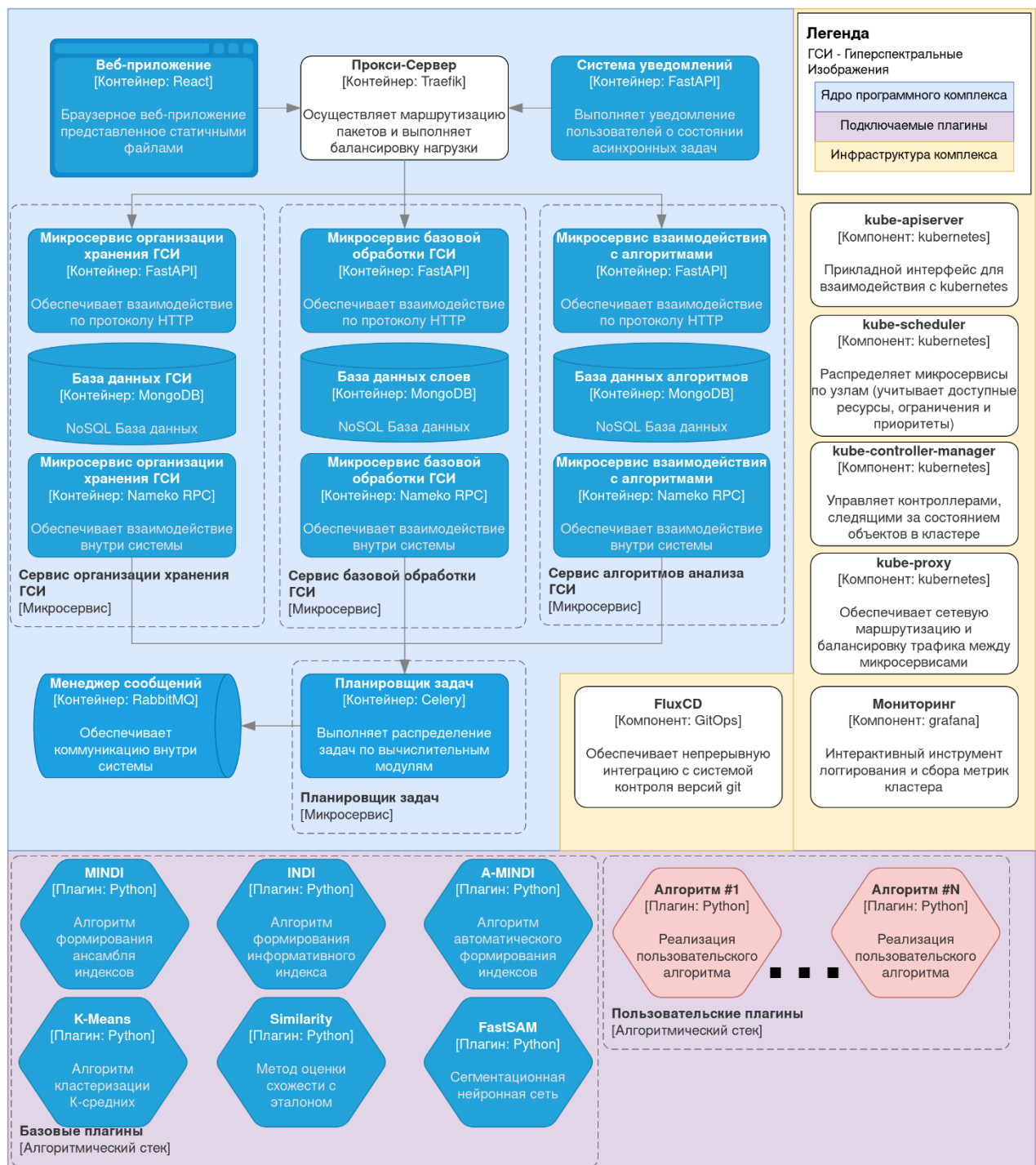


Рисунок 2.6 – Диаграмма контейнеров интеллектуальной системы

2.4.3 Диаграмма компонентов

Далее на рисунке 2.7 показана диаграмма компонентов, иллюстрирующая из каких составных частей состоит микросервис организации хранения ГСИ.

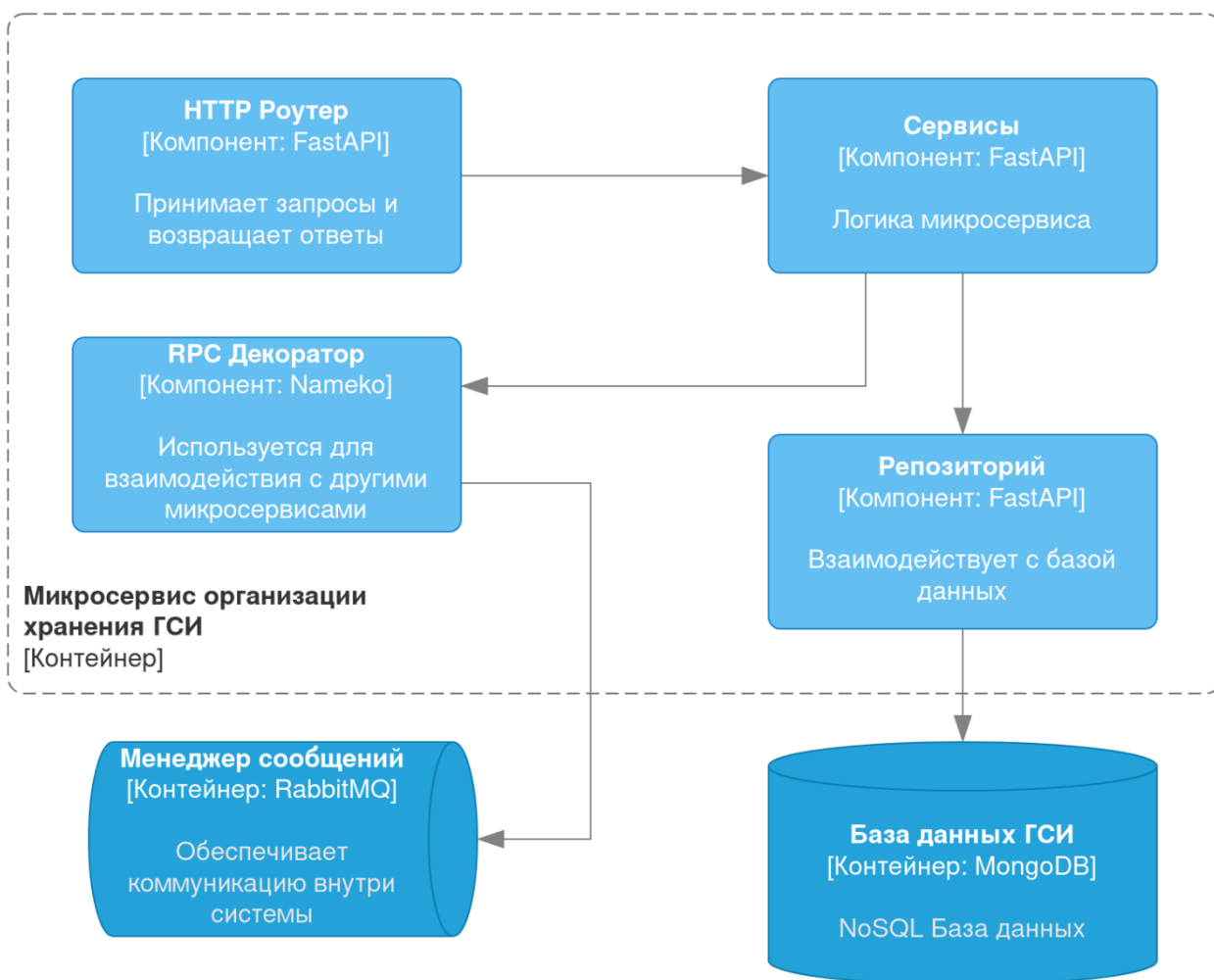


Рисунок 2.7 – Диаграмма компонентов микросервиса организации хранения ГСИ

Как можно видеть, структуру микросервиса можно представить в виде четырех компонент. HTTP роутер используется для реализации интерфейса взаимодействия клиента с системой по протоколу HTTP(S). Сервисы представляют собой бизнес-логику микросервиса. В этом компоненте обрабатываются поступающие запросы, реализуются основные алгоритмы работы с данными, а также формируются вызовы к репозиторию для взаимодействия с базой данных. Компонент репозиторий отвечает за взаимодействие с базой данных: он инкапсулирует операции по чтению и записи информации, что способствует разделению бизнес-логики и доступа к данным. Компонент “RPC Декоратор” реализован с использованием библиотеки Nameko и обеспечивает возможность межсервисного взаимодействия внутри системы. Данный компонент позволяет организовать

обмен данными и вызовы функций между различными микросервисами, что способствует гибкости и масштабируемости архитектуры. Аналогичным образом устроены и другие микросервисы системы.

На рисунке 2.8 иллюстрирована диаграмма компонентов микросервиса базовой обработки ГСИ.

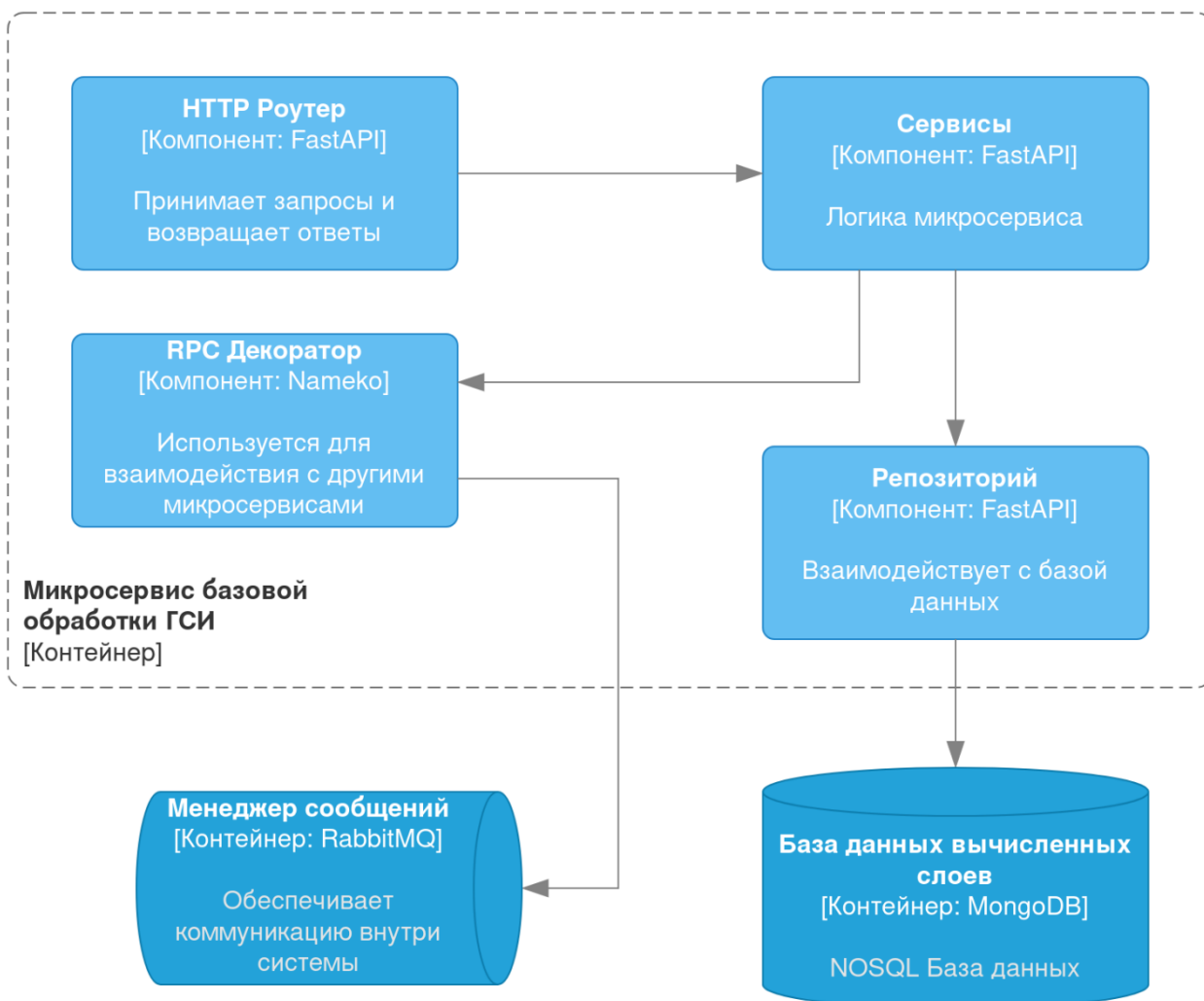


Рисунок 2.8 – Диаграмма компонентов микросервиса базовой обработки ГСИ

На рисунке 2.9 изображена диаграмма компонентов вычислительного модуля.

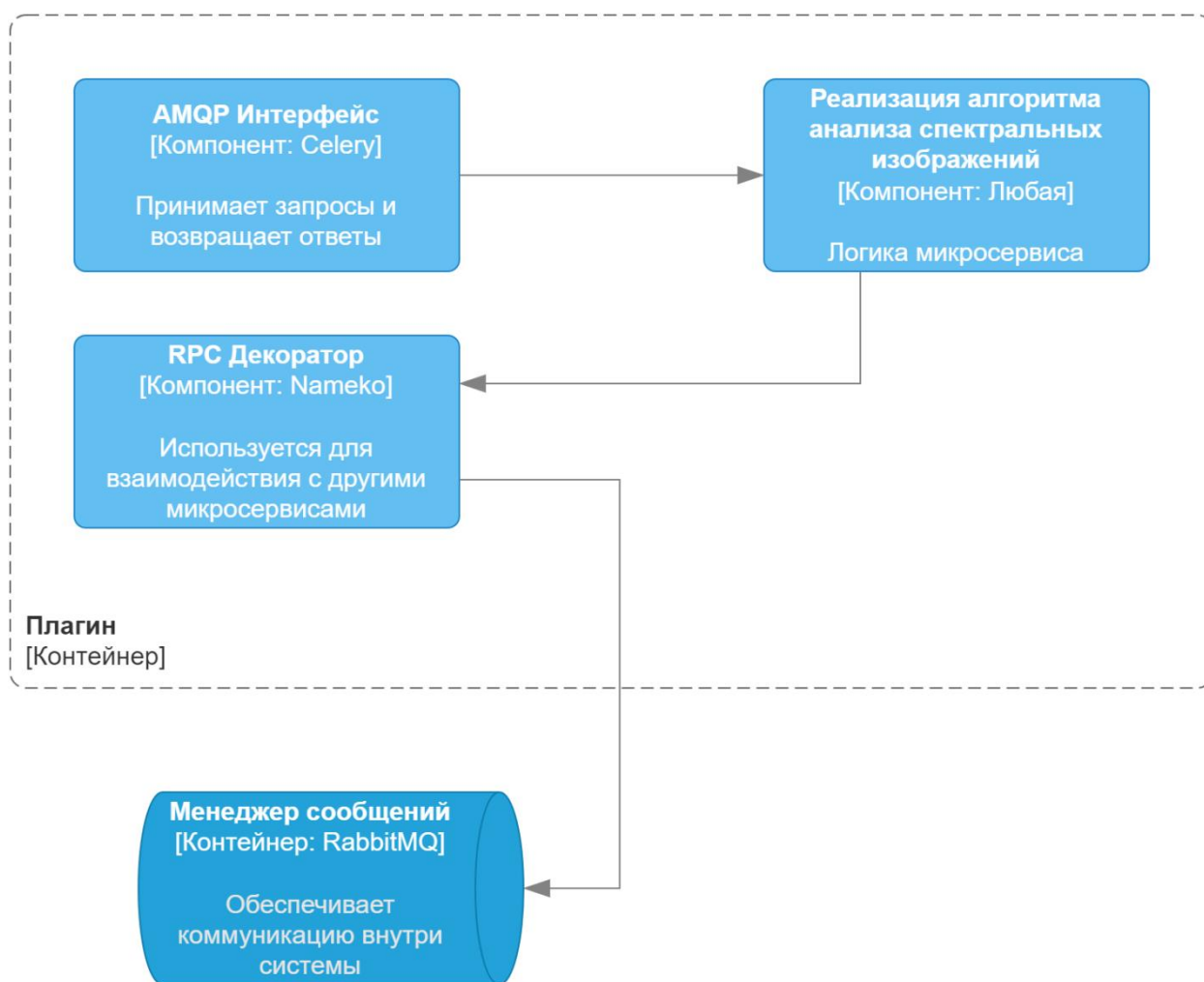


Рисунок 2.9 – Диаграмма компонентов вычислительного модуля

В отличие от основных микросервисов системы, вычислительные модули или плагины не обладают собственной базой данных. Для реализации плагина пользователю необходимо реализовать только компонент плагина ответственный за алгоритм анализа ГСИ. Единственным условием при этом является использование специализированной библиотеки и реализация определенного интерфейса. Данная специализированная библиотека позволит пользователю сосредоточиться только на реализации алгоритма, оставив ответственность за реализацию интерфейса AMQP и RPC специальному декоратору.

2.5 Логическая модель данных интеллектуальной системы

Рассмотрим далее структуру данных, используемых в рамках разработанной системы, и их взаимосвязи. Поскольку в качестве системы

управления базами данных во всех микросервисах используется документо-ориентированная СУБД MongoDB, описание модели хранения данных целесообразно выполнять не в терминах классической ER-модели реляционной базы данных, а в виде логической модели коллекций, документов, вложенных документов и ссылочных связей.

Такое представление позволяет отразить специфику выбранного подхода к хранению данных. В отличие от реляционной модели, в документо-ориентированной модели MongoDB основными единицами хранения являются документы, объединяемые в коллекции. При этом взаимосвязи между данными могут задаваться двумя способами:

- посредством вложенных документов, если связанные данные имеют общий жизненный цикл;
- посредством ссылок на идентификаторы документов, если сущности имеют самостоятельный жизненный цикл или должны использоваться повторно.

Использование MongoDB в рамках системы обусловлено тем, что хранимые данные имеют составную и частично изменяемую структуру. Это относится как к метаданным ГСИ, так и к параметрам алгоритмов, результатам анализа, конфигурациям вычисленных слоев и моделям представления. Документо-ориентированная модель хранения позволяет естественным образом представлять такие сущности в виде вложенных объектов и массивов без необходимости постоянного изменения жестко заданной реляционной схемы при разработке системы.

На рисунке 2.10 представлена логическая модель данных микросервиса организации хранения ГСИ.

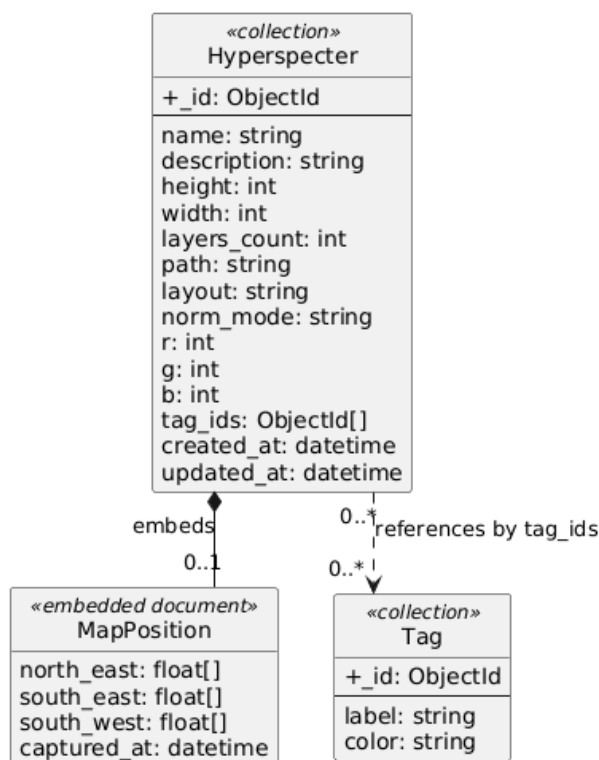


Рисунок 2.10 – Логическая модель данных микросервиса организации хранения ГСИ

Как можно видеть, центральной сущностью данной модели является документ *Hyperspecter*, содержащий основную информацию о ГСИ: его наименование, описание, размеры, число слоев, путь к файловым данным, параметры нормировки, настройки RGB-представления, а также служебные поля времени создания и обновления.

Пространственная информация о положении изображения представлена вложенным документом *MapPosition*. Использование вложенного документа в данном случае обусловлено тем, что данные о географическом положении не имеют самостоятельного жизненного цикла вне соответствующего ГСИ и логически являются его составной частью.

Для описания тегов используется отдельная коллекция *Tag*. Это решение позволяет многократно использовать одни и те же теги для различных изображений, а также централизованно управлять их названиями и цветовыми атрибутами. Связь между документом гиперспектрального изображения и тегами реализуется не через отдельную таблицу-связку, как в реляционной

модели, а посредством хранения массива идентификаторов тегов в документе Hyperspecter.

Таким образом, логическая модель данных данного микросервиса сочетает оба характерных для MongoDB подхода: встраивание зависимых данных во вложенные документы и использование ссылочных связей для повторно используемых сущностей.

На рисунке 2.11 представлена логическая модель данных микросервиса базовой обработки ГСИ.

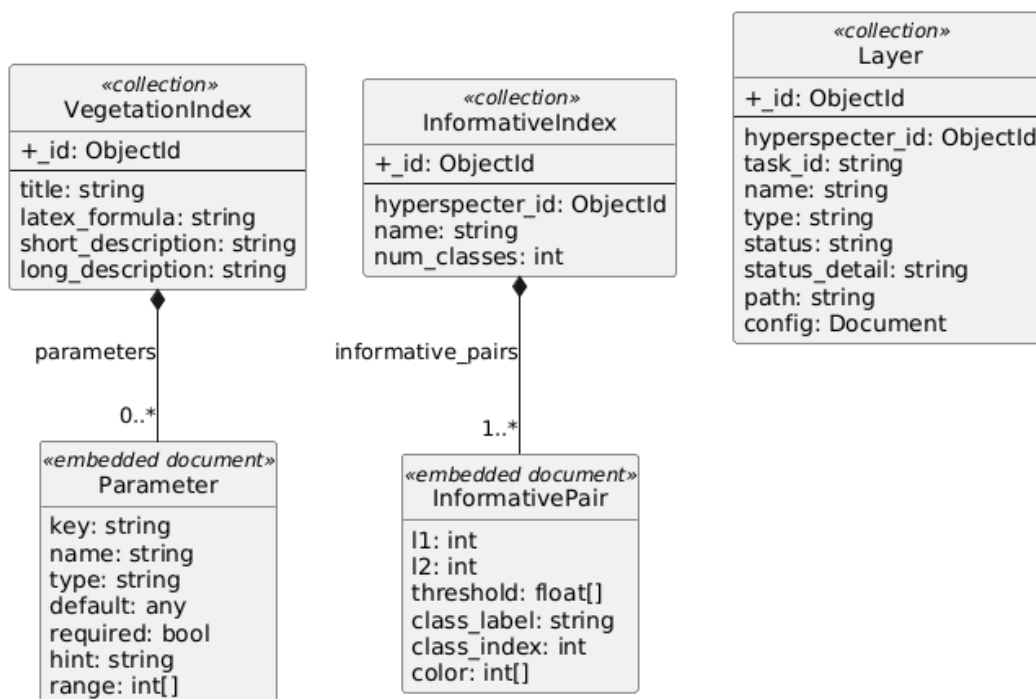


Рисунок 2.11– Логическая модель данных микросервиса базовой обработки гиперспектральных изображений

Данная модель включает коллекции VegetationIndex, InformativeIndex и Layer, предназначенные для хранения описаний вегетационных индексов, информативных индексов и вычисленных слоев соответственно.

Коллекция VegetationIndex хранит описание поддерживаемых вегетационных индексов, включая их наименование, формулу в формате LaTeX, краткое и развернутое текстовое описание. Параметры конкретного индекса представлены в виде массива вложенных документов Parameter.

Такой подход позволяет описывать различные наборы параметров без усложнения общей схемы хранения.

Коллекция `InformativeIndex` предназначена для хранения результатов формирования информативных индексов для заданного ГСИ. Документы данной коллекции содержат ссылку на идентификатор соответствующего изображения, число классов и массив вложенных документов `InformativePair`, описывающих пары информативных слоев и их характеристики.

Коллекция `Layer` используется для хранения вычисленных слоев, полученных в результате анализа ГСИ. Документ слоя содержит ссылку на исходное ГСИ, идентификатор вычислительной задачи, имя и тип слоя, его состояние, дополнительную диагностическую информацию, путь к данным и конфигурацию формирования слоя. Использование поля конфигурации в виде документа позволяет сохранять произвольные параметры, зависящие от конкретного алгоритма или сценария обработки.

Таким образом, логическая модель микросервиса базовой обработки ориентирована на хранение как predetermined описаний индексов, так и результатов вычислений, структура которых может варьироваться в зависимости от применяемого метода анализа.

На рисунке 2.12 представлена логическая модель данных микросервиса взаимодействия с алгоритмами анализа ГСИ.

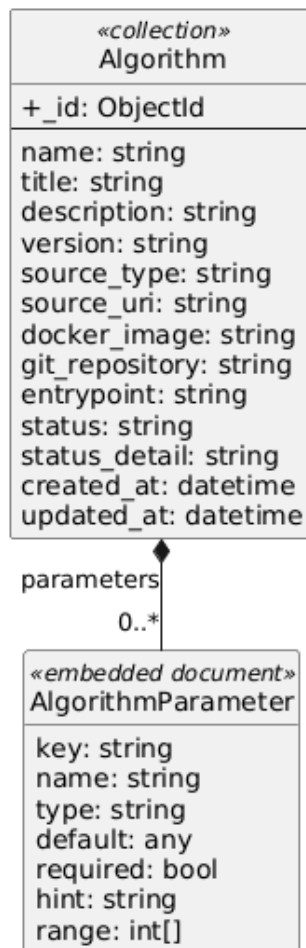


Рисунок 2.12 – Логическая модель данных микросервиса взаимодействия с алгоритмами анализа ГСИ

В рамках данного микросервиса используется коллекция **Algorithm**, предназначенная для хранения сведений о подключаемых вычислительных модулях. Документ алгоритма содержит его уникальное имя, отображаемое наименование, описание, версию, способ подключения, сведения об источнике загрузки, а также текущее состояние алгоритма в составе системы.

Параметры, принимаемые алгоритмом на вход, представлены в виде массива вложенных документов **AlgorithmParameter**. Такое решение позволяет единообразно описывать как встроенные алгоритмы, так и пользовательские вычислительные модули, добавляемые в систему посредством механизма плагинов.

Использование документо-ориентированной модели хранения в данном микросервисе особенно оправдано, поскольку подключаемые алгоритмы

могут различаться по числу и типам параметров, а также по набору метаданных, необходимых для их описания и выполнения. Хранение таких данных в виде документов позволяет добавлять новые алгоритмы без изменения жесткой реляционной схемы базы данных.

2.6 Реализация интеллектуальной системы

Все элементы системы были реализованы в рамках описанной архитектуры и распространяются посредством *Git*-репозитория.

Для реализации системы активно использовались *CD*-подходы (*Continuous Delivery*) для реализации автоматического тестирования, сборки образов и сборки *Python*-пакетов. Сборка *Python* пакетов необходима для публикации библиотеки, на основе которой реализуются пользовательские вычислительные алгоритмы. На рисунке 2.13 представлено изображение, иллюстрирующее интерфейс мониторинга выполнения *CD* конвейеров.

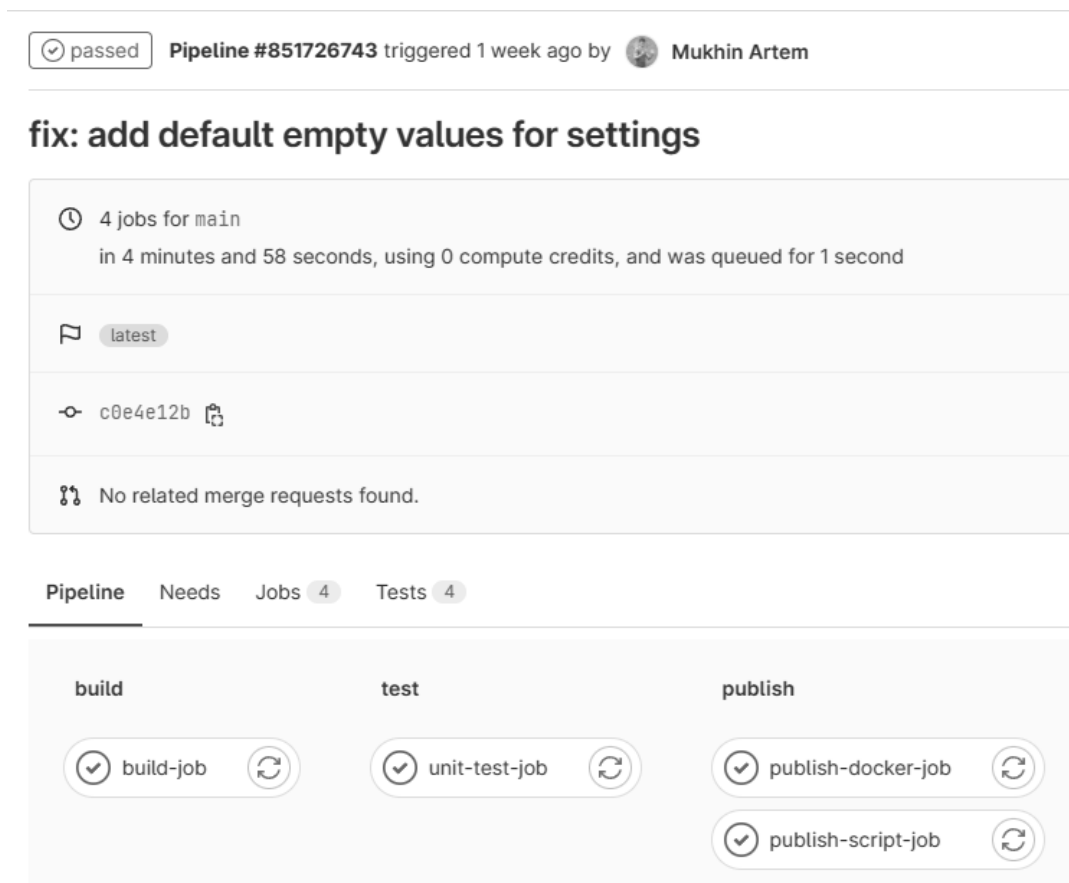
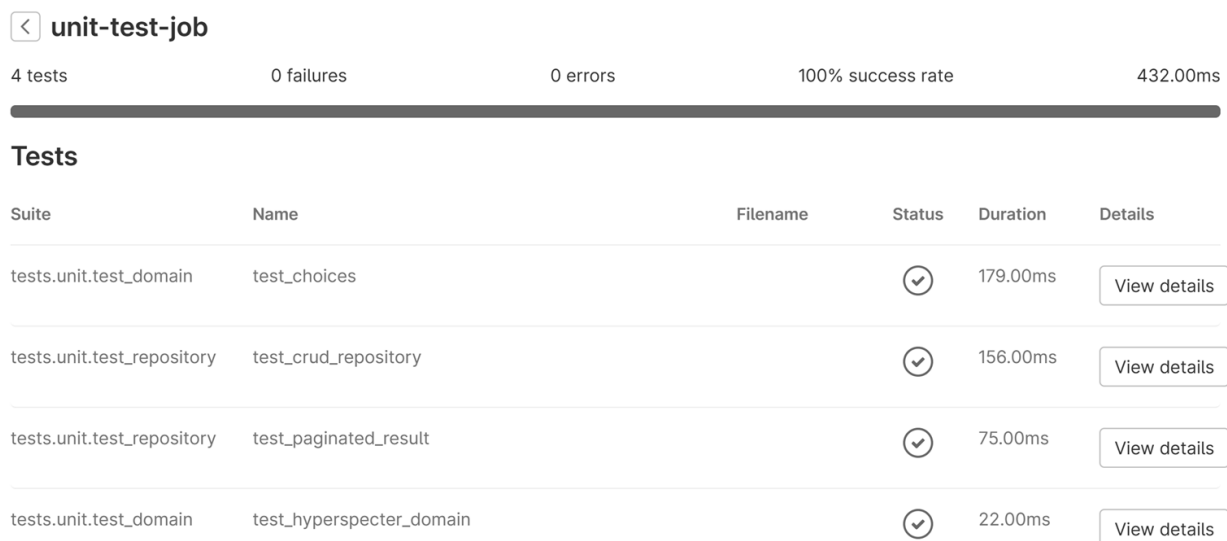


Рисунок 2.13 – Успешно выполненный конвейер, который произвел сборку образа, юнит-тестирование, загрузил итоговый образ в репозиторий и совершил сборку *Python* пакета

На рисунке 2.14 изображен интерфейс, посредством которого можно получить детальную информацию о проведении тестирования.



The screenshot shows a Jest test runner interface. At the top, it displays the test name 'unit-test-job' with a back arrow. Below this, a summary bar shows '4 tests', '0 failures', '0 errors', '100% success rate', and a total duration of '432.00ms'. A progress bar is also visible. The main section is titled 'Tests' and contains a table with the following data:

Suite	Name	Filename	Status	Duration	Details
tests.unit.test_domain	test_choices		✓	179.00ms	View details
tests.unit.test_repository	test_crud_repository		✓	156.00ms	View details
tests.unit.test_repository	test_paginated_result		✓	75.00ms	View details
tests.unit.test_domain	test_hyperspecter_domain		✓	22.00ms	View details

Рисунок 2.14 – Интерфейс, отображающий успешное прохождение всех ЮНИТ-ТЕСТОВ

2.6.1 Реализация веб-интерфейса

Для реализации веб-интерфейса был выбран фреймворк *React*. Данный фреймворк является наиболее популярным для реализации веб-интерфейсов посредством языков *JavaScript* (или *TypeScript*), *HTML*, *CSS*. Одним из наиболее важных факторов при выборе данного фреймворка являлся широчайший набор доступных библиотек. Благодаря *React* была достигнута высокая скорость и удобство разработки. В связке с *React* использовалась библиотека компонентов *PrimeReact*. Её использование позволило реализовать удобный и функциональный интерфейс, а также сократить время на написание общих компонентов разметки, кнопок, полей форм и т.д. Для реализации внутренней логики веб-приложения использовались такие инструменты, как: *formik* – функционал для создания форм, их валидации и прочее; *axios* – для реализации взаимодействия с сервером по протоколу HTTPS, WSS; *mobx* – для реализации внутреннего временного хранилища в браузере пользователя.

2.6.2 Реализация основных микросервисов системы

Для реализации серверной части системы использовался язык программирования *Python*. Данный язык программирования позволяет применять огромное число библиотек, предоставляющих различный функционал, необходимый для реализации практически любых функций. Так, для реализации *http*-сервера использовалась библиотека *fastapi*. Данная библиотека позволяет гибким образом настраивать прослушиваемые маршруты, формировать документацию, основываясь на спецификации *OpenAPI*. На рисунке 2.15 показан пример сгенерированной документации прикладного интерфейса микросервиса организации хранения ГСИ.



Рисунок 2.15 – Документация, сгенерированная по исходному коду инструментами библиотеки *FastAPI*

Кроме того, *fastapi* предназначен для реализации асинхронной логики, что особенно важно в условиях работы с большими объемами данных. Для реализации взаимодействия с базами данных использовалась библиотека *ODMantic*. Благодаря данной библиотеке реализуется ORM подход работы с базами данных.

Для реализации взаимодействия микросервисов посредством протокола AMQP использовалась библиотека *Nameko*. Данная библиотека удобным образом позволяет реализовать взаимодействие с помощью подхода *remote procedure call* (RPC).

Для работы с *data transfer object* (DTO) используется библиотека *pydantic*. Данная библиотека позволяет реализовать валидацию входящих и исходящих данных, а также производить трансформацию объектов из различных форматов.

Также в рамках серверной части системы активно использовались библиотеки *numpy* и *opencv*. Первая из них предоставляет инструмент для работы с многомерными данными, а вторая инструменты для работы с изображениями.

2.6.3 Реализация вычислительных модулей и программного интерфейса для взаимодействия с ними

Одним из требований к разработанной системе является возможность выполнения ресурсоёмких алгоритмов анализа, включая методы машинного обучения и нейросетевые модели. Для этого разработаны методы запуска вычислительных модулей в изолированной среде с возможностью задания требуемых вычислительных ресурсов, в том числе GPU. Такая организация позволяет подключать к системе как классические алгоритмы обработки ГСИ, так и модели глубокого обучения, требующие аппаратного ускорения.

Возможность применения нейросетевых методов к ГСИ ранее была рассмотрена в работах [118, 119] и исследована с применением программных модулей [120, 121]. Данные методы могут быть реализованы в виде отдельных вычислительных модулей системы. Вместе с тем их использование требует подготовки обучающей выборки, выбора архитектуры, настройки гиперпараметров и выполнения процедуры обучения, что повышает требования к квалификации пользователя. Поэтому в настоящей работе основной акцент сделан на алгоритмах автоматизированного анализа на основе информативных индексов, тогда как нейросетевые методы рассматриваются как поддерживаемый расширяемый сценарий для исследовательских и прикладных задач.

Для реализации вычислительных модулей и интерфейса взаимодействия с ними использовалась библиотека *celery*. Данная библиотека позволяет реализовать фоновые вычисления посредством использования протокола AMQP и менеджера сообщений совместно с базой данных типа “ключ-значение”. Так в ходе добавления алгоритма в систему – он добавляется в рассмотренную ранее базу данных алгоритмов, где ему присваивается уникальное имя. Затем в рамках системы производится развертывание контейнера соответствующего алгоритма. Внутри данного контейнера посредством разработанной библиотеки *vidis_algorithm_api* [122] производится регистрация алгоритма в *celery*. После этого алгоритм становится доступным к использованию в рамках системы по его уникальному имени.

Разработанная библиотека *vidis_algorithm_api* распространяется посредством общедоступного репозитория пакетов PyPi. Данная библиотека представляет набор декораторов позволяющих разработчику алгоритмов сконцентрироваться лишь на написании непосредственно алгоритма обработки ГСИ. Данные декораторы скрывают реализацию автоматического логирования прогресса, сохранения результата и получения данных. Также для удобства реализации алгоритмов был разработан базовый docker-образ: *banayaki/vidis-algorithms*. Он содержит ряд необходимых для работы с библиотекой зависимостей и саму библиотеку актуальной версии. На рисунках 2.16 и 2.17 продемонстрировано существование данной библиотеки в рамках репозитория PyPi и образа в репозитории *DockerHub* соответственно.



Рисунок 2.16 – Репозиторий пакетов PyPI

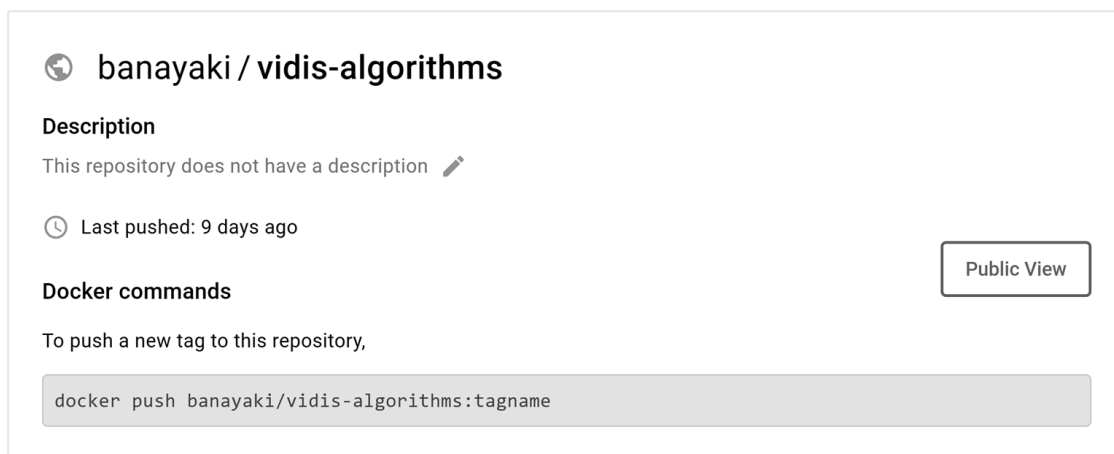


Рисунок 2.17 – Репозиторий docker-образов DockerHub

2.6.4 Реализация доступа к серверной части системы

Для реализации маршрутизации запросов между различными частями системы по протоколам HTTP и WS используется одно из двух средств, в зависимости от способа развертывания. Так в качестве прокси-сервера в рамках окружения для разработки использовался инструмент *Traefik*. Благодаря ему можно детально настраивать маршрутизацию между докер контейнерами с помощью аннотаций. Также удобным является факт динамической маршрутизации. Под этим подразумевается автоматическая конфигурация *Traefik* по мере добавления контейнеров в систему. Для организации прокси-сервера в окружении для конечных пользователей также использовался *Traefik*.

2.7 Анализ отказоустойчивости интеллектуальной системы

Современные системы, построенные на принципах микросервисной архитектуры, предъявляют повышенные требования к обеспечению отказоустойчивости и надёжности. Сбой одного или нескольких микросервисов может привести к существенным перебоям в работе всей системы, потере данных или снижению производительности.

В таблице 2.10 приведена идентификация критических точек отказа различных модулей разработанной системы.

Таблица 2.10 – Идентификация критических точек отказа

Компонент	Возможные причины отказа	Влияние на систему	Механизмы
<i>Прокси-сервер</i>	Перегрузка, сбой	Потеря входного трафика	Горизонтальное масштабирование, настройка нескольких реплик, Liveness/Readiness пробы, перезапуск
<i>Менеджер сообщений</i>	Перегрузка, потеря соединения, сбой	Нарушение коммуникации с алгоритмическим стеком, зависание задач	Конфигурация RabbitMQ кластера, мониторинг состояния, перезапуск
<i>Планировщик задач</i>	Потеря данных, сбой	Остановка обработки задач	Мониторинг состояния, перезапуск, резервная копия состояния
<i>Базы данных</i>	Потеря данных, сбой	Нарушение работы сервисов, потеря данных	Репликация БД, регулярные бэкапы
<i>Микросервисы</i>	Сбой	Недоступность части функционала	Перезапуск, горизонтальное масштабирование, настройка нескольких реплик
<i>Вычислительные модули (алгоритмы)</i>	Длительное выполнение, ошибки в коде, сбой	Задержка/невыполнение вычислений	Контроль времени выполнения, автоматический перезапуск задач, хранение задач в очереди
<i>Ненадежная сеть</i>	Отсутствие сетевой связности между компонентами системы	Недоступность части функциональных возможностей, потеря или невозможность получить результаты	Повторные запросы, сохранение статусов задач

Разработанная архитектура системы включает следующие меры отказоустойчивости.

1) Использование Kubernetes для автоматического перезапуска микросервисов.

В случае любого сбоя – микросервис перезапускается автоматически. Для контроля за состоянием микросервисов реализованы механизмы наблюдения возможности доступа к микросервису.

2) Горизонтальное масштабирование критических сервисов.

Для большинства микросервисов реализовано горизонтальное масштабирование, что позволяет как оптимизировать обработку пользовательских запросов, так и предотвращает проблему “единой точки отказа”.

3) Мониторинг и уведомления о сбоях.

Реализован мониторинг состояния всех компонентов кластера. При обнаружении аномалий или отказов происходит оповещение ответственных лиц и автоматизированный запуск процедур восстановления.

4) Стратегия контролируемой деградации функциональности.

В случае отказа одного из микросервисов система продолжает работать с ограниченным функционалом – например, невозможен анализ ГСИ, но доступен просмотр загруженных в систему данных.

5) Механизмы повторных попыток.

Для внутренних RPC/HTTP запросов используются библиотеки и паттерны повторных попыток.

Кроме вышеописанных возможных причин отказа, существует ряд рисков, связанных с нехваткой тех или иных ресурсов. Полностью автоматизировать механизмы восстановления системы из таких сбоев не предоставляется возможным. Однако время восстановления может быть сведено к минимуму за счет возможности вертикального масштабирования благодаря добавлению нового узла в кластер, в котором развернута система.

Дополнительно к отказам отдельных компонентов необходимо учитывать отказы межсервисного взаимодействия, характерные для распределённых микросервисных систем. В таких системах сбой может быть связан не только с недоступностью конкретного сервиса, но и с нарушением сетевой связности между компонентами, увеличением времени ответа, потерей соединения с менеджером сообщений, невозможностью передачи результатов вычислений или временной недоступностью хранилища данных.

Для разработанной интеллектуальной системы обработки гиперспектральных изображений наиболее значимыми являются следующие сценарии отказов межсервисного взаимодействия: недоступность сервиса управления данными для сервисов визуализации и анализа; потеря соединения между сервисами постановки задач и брокером сообщений; недоступность вычислительного модуля после постановки задачи; невозможность сохранения результата обработки. Такие отказы не всегда приводят к полной остановке системы, однако могут вызывать частичную деградацию функциональности, задержку выполнения задач или необходимость повторного запуска вычислений.

Для снижения влияния указанных отказов в архитектуре предусмотрены механизмы таймаутов и повторных попыток при выполнении межсервисных запросов, сохранение вычислительных задач в очереди сообщений, контроль состояния сервисов средствами Kubernetes, автоматический перезапуск недоступных компонентов, а также разделение функций хранения, визуализации и анализа. Благодаря этому при отказе вычислительного контура пользователь может продолжать работу с ранее загруженными данными и результатами, а при восстановлении связности отложенные задачи могут быть повторно обработаны или перезапущены.

Для обработки отказов вычислительных модулей в системе используется контроллер жизненного цикла вычислительных задач, периодически проверяющий состояние выполняемых операций и наличие признаков активности со стороны вычислительных модулей. При отсутствии подтверждения выполнения в течение заданного интервала задача признаётся некорректно завершённой, а её состояние обновляется в пользовательском интерфейсе. Система не восстанавливает результаты потерянных или частично выполненных вычислений, поскольку их целостность не может быть гарантирована. После восстановления работоспособности вычислительного

модуля пользователь должен повторно сформировать запрос на выполнение алгоритма.

Также важно отметить то, что ответственным за корректную реализацию новых алгоритмов, расширяющих систему, является их автор. В рамках разработанной версии системы не предусмотрено механизмов, позволяющих оценить работоспособность и производительность пользовательского алгоритма. Однако добавление таких механизмов возможно при модификации микросервиса взаимодействия с алгоритмами. Наиболее перспективным способом оценки качества разработанных пользователем алгоритмов является использование LLM-агентов, способных анализировать и исполнять программный код.

2.8 Развертывание системы и окружения

Для удобства разработки, тестирования и развертывания системы было принято решение использовать разные окружения, использующие разные технологии развертывания.

Так при работе использовалось два окружения:

- окружение для разработки;
- окружение для конечных пользователей.

Оба окружения существенно отличаются друг от друга методом развертывания. Так для удобства разработки использовался инструмент *docker-compose*. Благодаря ему становится возможной разработка напрямую в контейнерах, которые обеспечены всеми необходимыми пакетами и связями. Однако *docker-compose* плохо подходит для использования в окружении для конечных пользователей. Одним из наиболее значимых недостатков является отсутствие автоматического балансировщика нагрузки и возможности настройки автоматического развертывания с помощью *CI (Continuous Integration)* конвейеров.

Для окружения конечных пользователей использовался инструмент *Kubernetes* совместно с *Helm* и *FluxCD*. Данный набор инструментов позволяет

настроить систему таким образом, чтобы она автоматически обновлялась по мере обновления соответствующих репозиториях, содержащих исходный код. Система, развернутая в данном окружении, имеет возможность автоматически восстанавливаться и масштабироваться. *Kubernetes* позволяет в полной мере реализовать подход *GitOps*, согласно которому все конфигурационные и инфраструктурные файлы хранятся в *Git*-репозиториях. Далее на рисунке 2.18 представлена общая диаграмма развертывания окружения средствами *Kubernetes*.

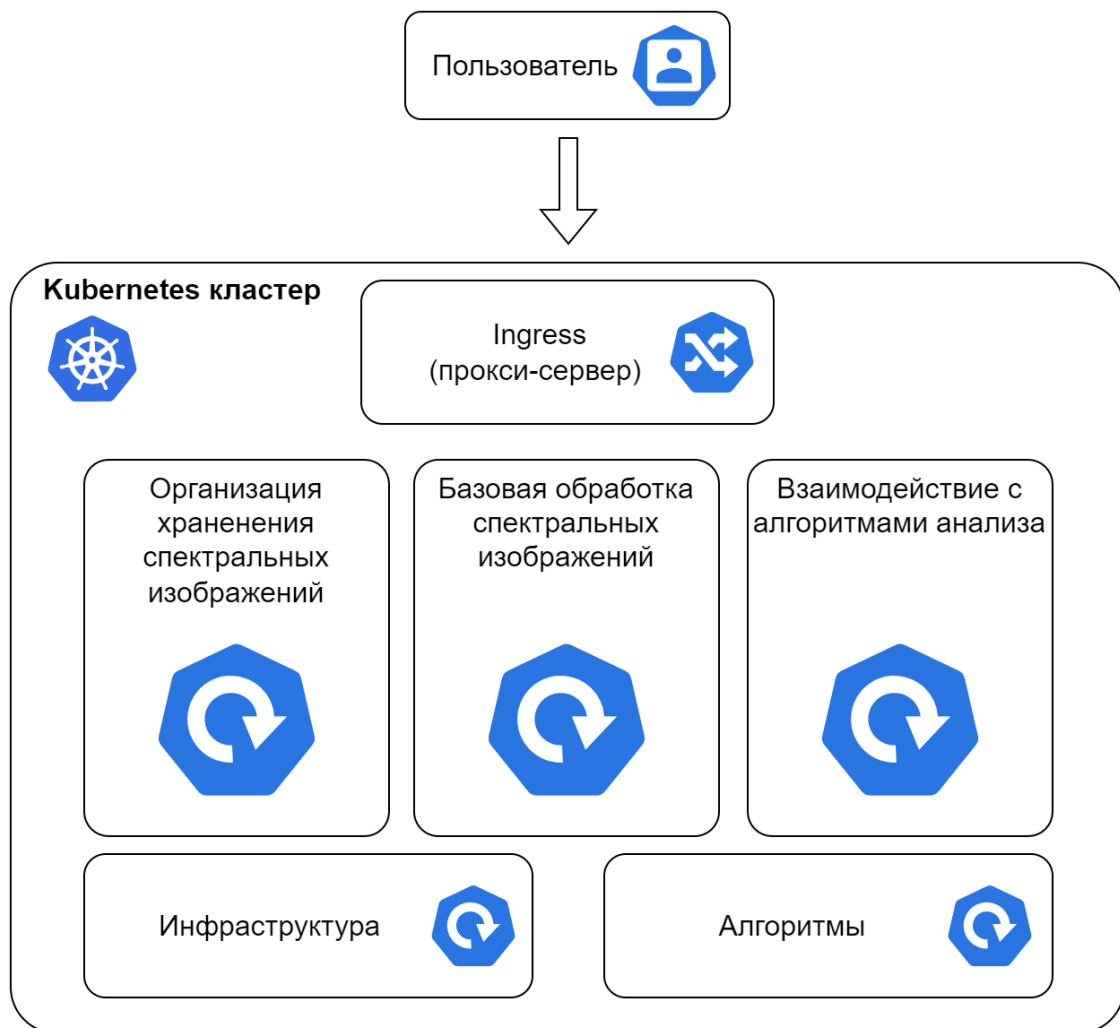


Рисунок 2.18 – Диаграмма, схематично изображающая составные элементы *Kubernetes* кластера

По диаграмме видно, что итоговое окружение состоит из шести частей. Рассмотрим каждую из них подробнее.

Ingress (прокси-сервер) в рамках кластера представляет собой не что иное, как прокси-сервер, ожидающий запросы на портах 80 (HTTP) и 443 (HTTPS). Благодаря *ingress* производится маршрутизация входного трафика в кластере.

Основные микросервисы представлены на диаграмме тремя блоками. Каждый из них в рамках *Kubernetes* кластера является узлом типа *deployment* с сопутствующей им инфраструктурой. Далее на рисунке 2.19 изображена подробная диаграмма внутреннего устройства одного из таких блоков. Другие микросервисы устроены аналогичным образом.

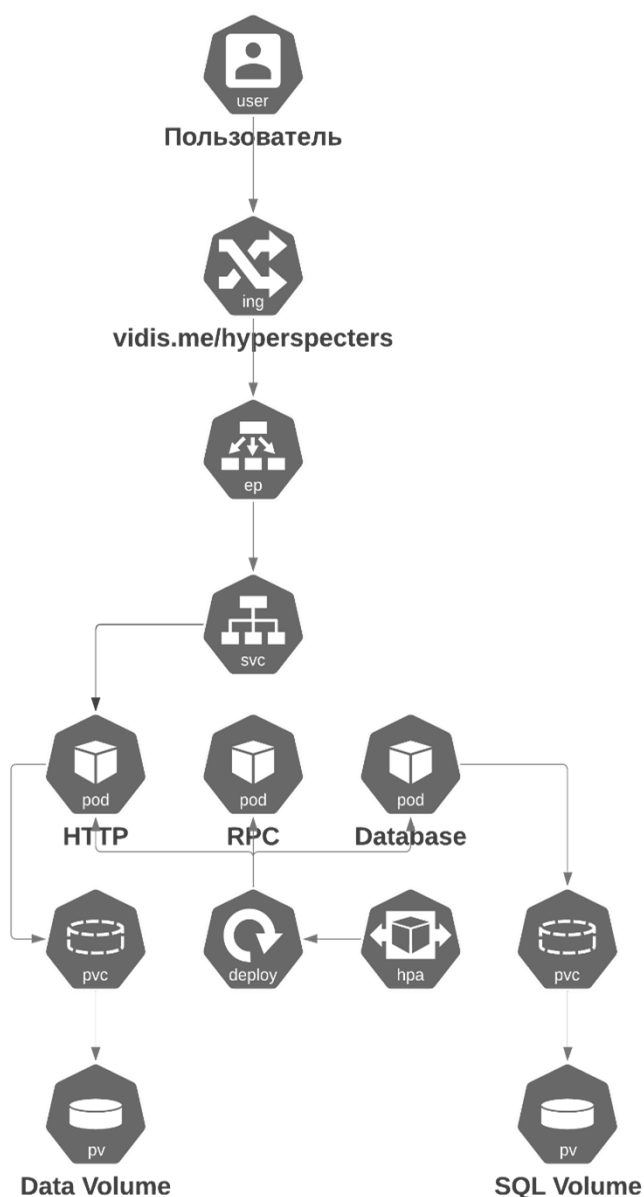


Рисунок 2.19 – Составные части архитектуры *Kubernetes* реализующие микросервис организации хранения ГСИ

На данном изображении можно видеть, что *ingress* контроллер перенаправляет весь трафик по указанному правилу в указанный сервис, откуда он попадает в *pod*, обслуживающий его. Все поды в рамках микросервиса контролируются с помощью такого ресурса как *deployment*. С его помощью настраивается горизонтальное масштабирование микросервиса. Также на диаграмме можно видеть, что *pod*, обслуживающий HTTP трафик (как на самом деле и RPC *pod*) получает доступ к “Data Volume”, который предоставляет место для хранения ГСИ. В настоящей реализации данный *persistent volume* скрывает способ получения доступа к данным. В зависимости от используемого окружения, доступны различные способы, начиная от локального хранения данных (подходит при развертывании в кластере из одного узла), заканчивая распределенными системами хранения типа Longhorn. Также на диаграмме видно, что *pod Database* также получает доступ к хранилищу, где расположены данные для базы данных. Аналогичным образом устроены другие микросервисы, различаться будут лишь используемый образ и правило точки входа трафика для контроллера *ingress*.

К инфраструктурным компонентам системы относятся реализация асинхронного менеджера сообщений *RabbitMQ* и базы данных типа “ключ-значение” *Redis*.

Алгоритмы анализа ГСИ встроены в систему тривиальным образом, посредством ресурса *deployment* с горизонтальным масштабированием. Данный ресурс включает в себя один единственный *pod*, в котором и разворачивается пользовательский образ с алгоритмом.

На рисунках 2.20 и 2.21 представлены изображения со страницы мониторинга состояния развернутой системы на обоих окружениях.

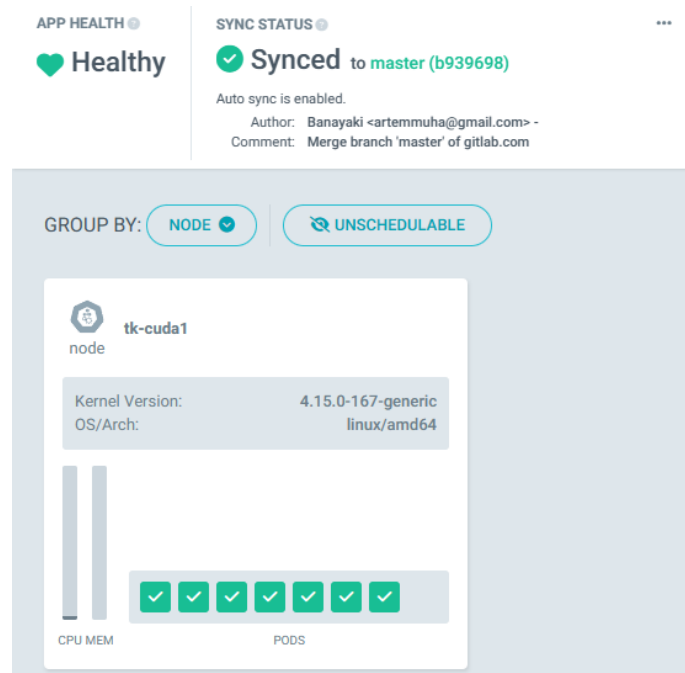


Рисунок 2.20 – Система успешно развернута в пользовательском окружении

☐		vidis	-	Running (4/4)			
☐		flower	d568015af013	mher/flower:latest	Running	8890.8888	37 seconds ag
☐		traefik	537e184cbb3	traefik:v2.9	Running	51123.8080	37 seconds ag
☐		rabbitmq-1	07b4e8a7eca4	rabbitmq:3-management-alpine	Running	15672.15672	37 seconds ag
☐		redis-1	2575074270cf	redis:7-alpine	Running	6379.6379	37 seconds ag
☐		vidis-algorithms	-	Running (3/3)			
☐		app-1	6c5b68989a8c	vidis-algorithms:app:latest	Running		38 seconds ag
☐		mongodb-algorithms-1	fede78085f82	mongo:latest	Running	8085.27017	59 seconds ag
☐		mongodb-algorithms-dashboard	1102d3605671	mongo-express:latest	Running	8086.8081	56 seconds ag
☐		vidis-backend-base-processing	-	Running (4/4)			
☐		app-1	7197882e414d	vidis-backend-base-processing:app	Running		37 seconds ag
☐		app-rpc-1	8b563df495cd	vidis-backend-base-processing:app	Running		37 seconds ag
☐		mongodb-1	370e0325429b	mongo:latest	Running	8084.27017	58 seconds ag
☐		mongodb-dashboard-1	7300c5fcf94a	mongo-express:latest	Running	8083.8081	56 seconds ag
☐		vidis-backend-hyperspecter	-	Running (4/4)			
☐		pgadmin	785fb2e54e4c	dpage/pgadmin4:latest	Running	8081.80	37 seconds ag
☐		app-http-1	2f2c8bdca824	vidis-backend-hyperspecter:app	Running		38 seconds ag
☐		app-rpc-1	1766613d7441	vidis-backend-hyperspecter:app	Running		38 seconds ag
☐		db-1	4d1067e84a3d	postgres:latest	Running	54321.5432	58 seconds ag

Рисунок 2.21 – Система успешно развернута в окружении для разработки

2.9 Исследование возможностей разработанной интеллектуальной системы

Рассмотрим далее функциональные возможности, предоставляемые системой посредством пользовательского веб-интерфейса.

2.9.1 Организация хранения гиперспектральных данных

Перейдя по адресу на веб-сайт, пользователь попадает на страницу, отображающую список доступных в системе ГСИ. На рисунке 2.22 иллюстрирована данная страница.

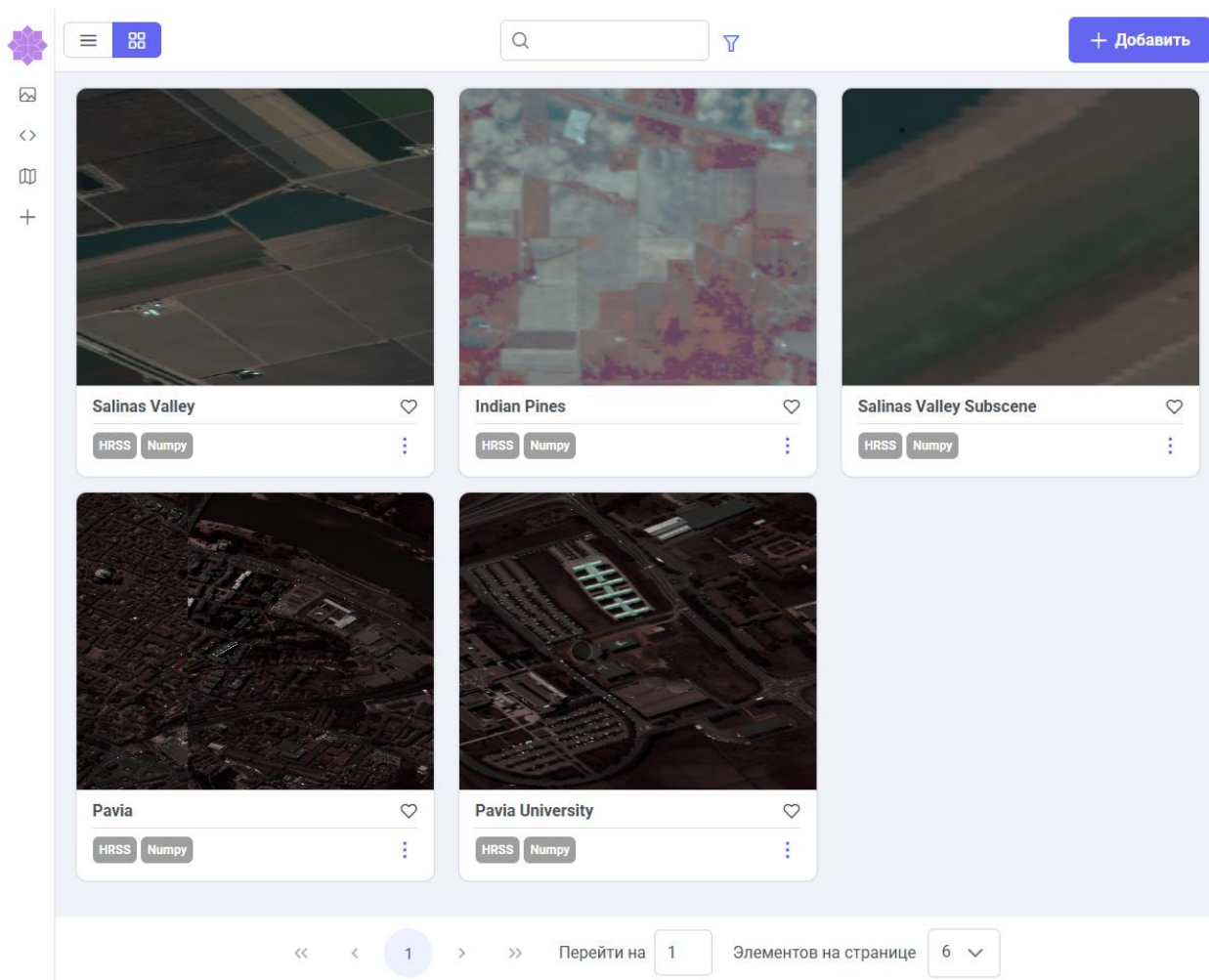


Рисунок 2.22 – Страница со списком загруженных ГСИ

Как можно видеть, данный список предоставляет удобную визуализацию ГСИ с их названиями. Сверху страницы имеется текстовое поле для указания критериев поиска изображений, правее него кнопка, позволяющая настроить фильтры и в конце данной панели кнопка,

используемая для добавления новых ГСИ. Также обратим внимание на нижнюю часть страницы. Список ГСИ разделен на страницы с целью уменьшения нагрузки на пропускной канал благодаря использованию постраничного подхода отображения данных.

Рассмотрим детальнее окно редактирования метаданных ГСИ. На рисунке 2.23 иллюстрирована страница редактирования метаданных ГСИ.

Параметры спектрального изображения

Название
Salinas Valley

Короткое название
salinas

Если поле не будет заполнено, то в качестве значения будет использоваться идентификатор изображения.
Короткое название может быть использовано как псевдоним в адресной строке

Описание изображения
Изображение из набора данных Hyperspectral Remote Sensing Scenes

Метки

HRSS ⊗ Numpy ⊗

☐ В избранном

Индексы RGB

Красный
35

Зеленый
16

Синий
10

Изменить

Рисунок 2.23 – Страница редактирования параметров метаданных ГСИ

На данной странице видны изменяемые параметры метаданных ГСИ. Так пользователь может изменить название изображения; его короткое название, которое можно использовать вместо численного идентификатора; описание; теги и индексы RGB слоев. Помимо этого, на странице визуализирован куб, на сторонах которого изображены срезы ГСИ

относительно соответствующих осей. Такая визуализация позволяет удобным образом трансформировать изображение, если при его загрузке был неверно определен порядок слоев. На рисунке 2.24 изображен процесс поворота ГСИ вдоль одной из осей.

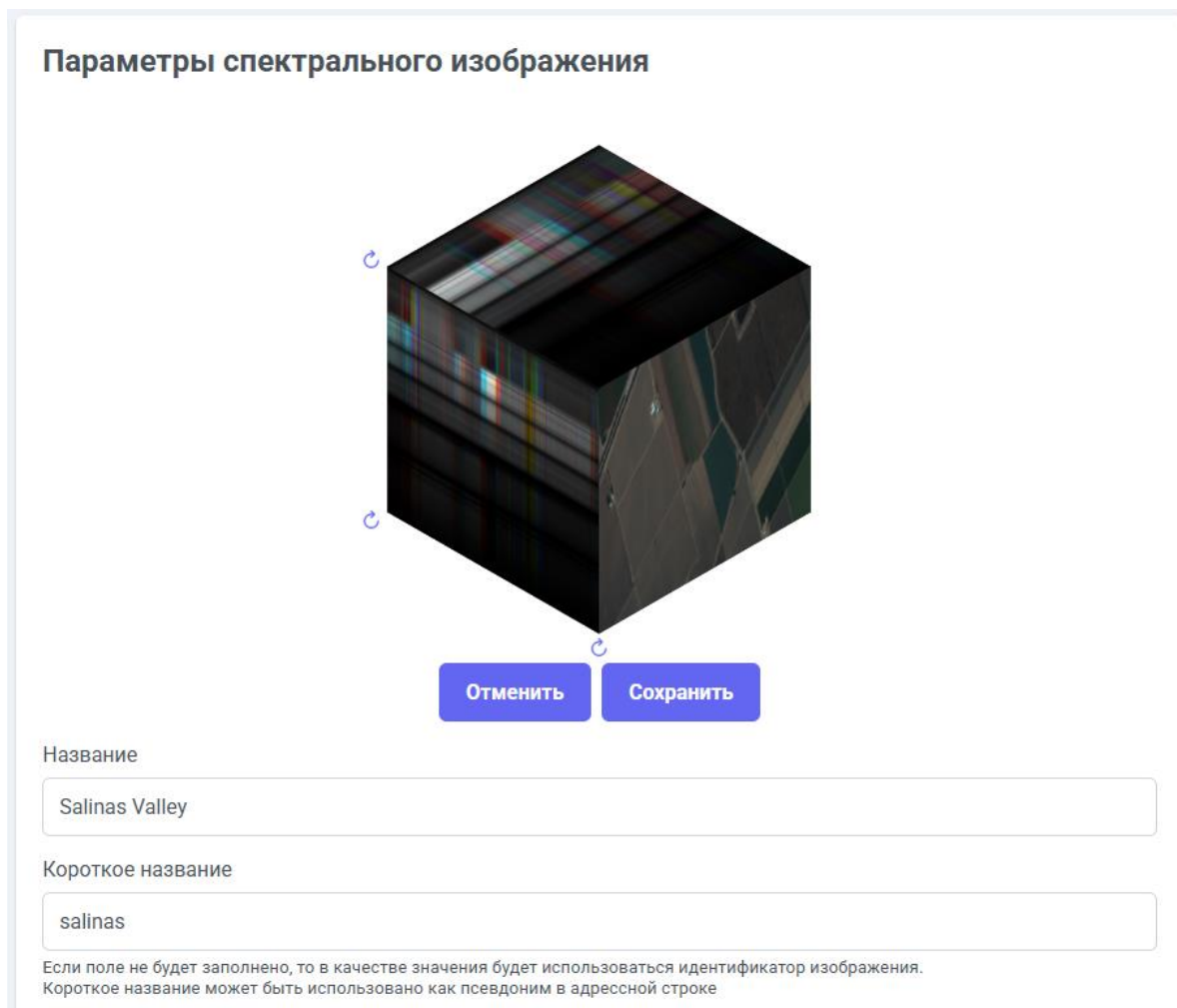


Рисунок 2.24 – Визуализация поворота ГСИ

Изменение ориентации изображения в пространстве сопровождается соответствующей анимацией.

Также на данной странице можно увидеть другие параметры ГСИ и его геопривязку к глобальной карте мира. При условии, если данное местоположение не было задано при загрузке конфигурационного файла, пользователь может самостоятельно задать координаты посредством пользовательского интерфейса. На рисунке 2.25 изображен интерфейс, соответствующий данному функционалу.

будет предложен выбор из двух опций, в зависимости от типа загружаемых данных. От выбора зависит лишь следующий этап, определяющий набор и форматы данных необходимых для загрузки файлов. На рисунке 2.27 продемонстрирована данная страница.

1 Тип данных 2 Файлы 3 Описание

Подготовленные данные

Сформированные спектральные изображения готовы к использованию или файлы содержащие трехмерные данные

Поддерживаемые форматы:

- Numpy массив .npy
- GeoTIFF или BigTIFF .tiff
- Архив изображений .zip
- Обычные изображения .jpeg .png
- RAW формат .raw

Неподготовленные данные

Спектральное изображение которое должно быть сформировано из видеофайла, иначе говоря видео сформированное сканирующим спектральным сенсором

Поддерживаемые форматы

- Видеозаписи .mp4 .avi

Рисунок 2.27 – Страница с выбором формата загружаемых данных. Пользователь может загрузить файл трехмерного массива или видеофайл.

Далее пользователю предоставляются к заполнению два файловых поля. На рисунке 2.28 продемонстрирован соответствующий интерфейс.

Файл данных

Загрузите файл содержащие спектральные данные Выбрать

Файл мета-данных

Загрузите файл содержащие дополнительную мета-информацию об изображении (Опционально) Выбрать

← Назад ✓ Продолжить

Рисунок 2.28 – Интерфейс загрузки файлов

После загрузки файлов, пользователь перейдет на следующий шаг, на котором ему будет предложена форма для заполнения. В этой форме пользователь сможет задать название изображения, его описание, тип нормировки, порядок слоев, индексы RGB слоев. На рисунке 2.29 продемонстрирована данная форма.

Параметры спектрального изображения

Название

Salinas Valley

Короткое название

salinas

Если поле не будет заполнено, то в качестве значения будет использоваться идентификатор изображения.
Короткое название может быть использовано как псевдоним в адресной строке

Описание изображения

Изображение из набора данных Hyperspectral Remote Sensing Scenes

Способ нормализации данных ?

Глобальная нормировка

Порядок слоев в данных ?

C-H-W

Метки

HRSS

Numpy

Индексы RGB

Total layers count of the hyperspecter is unknown. Set RGB values only if you know values exactly

Красный

35

Зеленый

16

Синий

10

← Назад

Сохранить

Рисунок 2.29 – Форма с параметрами загружаемого ГСИ

При успешном заполнении данной формы, пользователь может начать процедуру сохранения данных.

2.9.3 Обработка гиперспектральных изображений

Далее рассмотрим страницу обработки ГСИ, на которую можно попасть, выбрав ГСИ из списка. На рисунках 2.30, 2.31 представлен процесс мозаичной загрузки изображения на странице обработки.

Рассмотрим далее основные панели. На рисунке 2.32 изображена развернутая панель работы со слоями и панель управления.

С помощью панели управления, пользователь может вернуться обратно к списку гиперспектральных изображений, произвести откат операций или повторить отмененные, попасть на страницу, соответствующую выбранному ГСИ и сформировать ссылку с текущей конфигурацией. В панели слоев при входе на страницу анализа, отображаются только три слоя соответствующие красной, зеленой и синей длинам волн. Для любых слоев можно настроить цвет отображения, прозрачность, указать интервал



Рисунок 2.30 – По мере обработки сервером, части итогового изображения загружаются

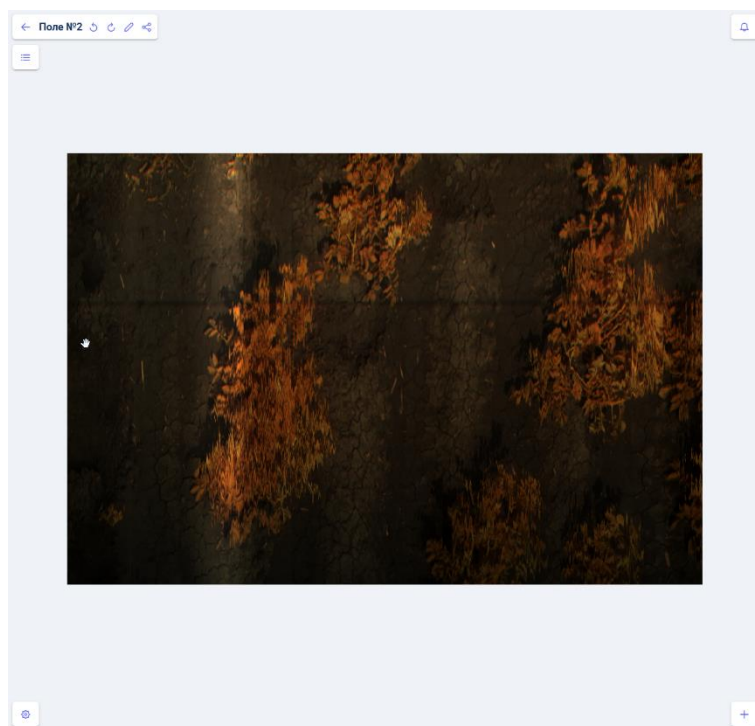


Рисунок 2.31 – Изображение полностью загружено

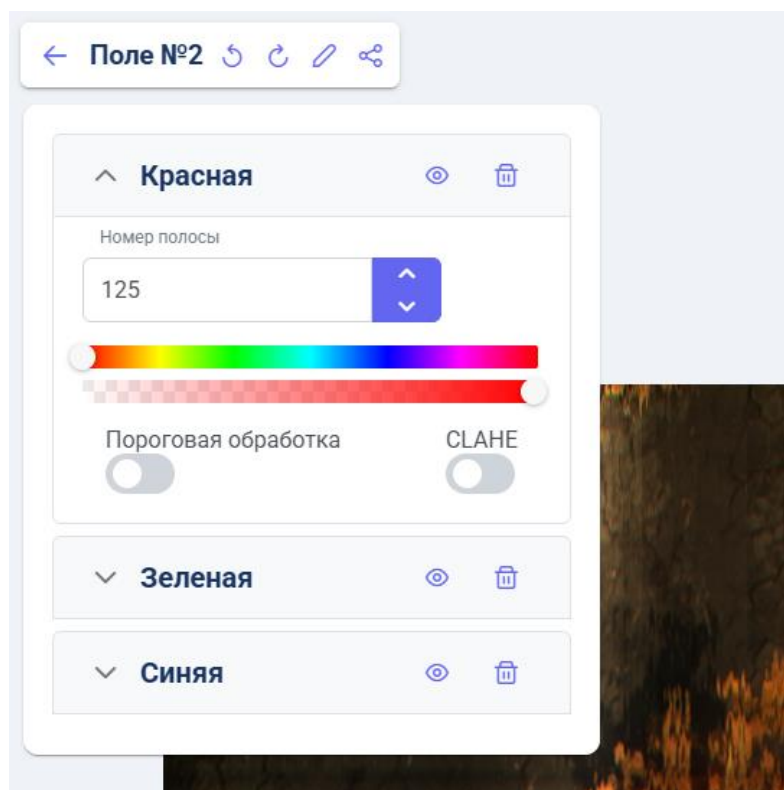


Рисунок 2.32 – Панель управления и панель слоев

пороговой обработки и включить использование эквализации методом *CLANE*. Также слоям можно изменить имя, дважды кликнув по нему, а для слоев, соответствующих ГСИ – можно изменить индекс, или, иначе говоря,

выбрать другую спектральную полосу в качестве слоя модели отображения без перехода в специальное меню добавления нового слоя. Слои можно временно скрывать, или удалять из списка.

В правом нижнем углу страницы располагается кнопка, нажатием на которую можно вызвать панель добавления слоев в конфигурацию. На рисунке 2.33 изображена данная панель.

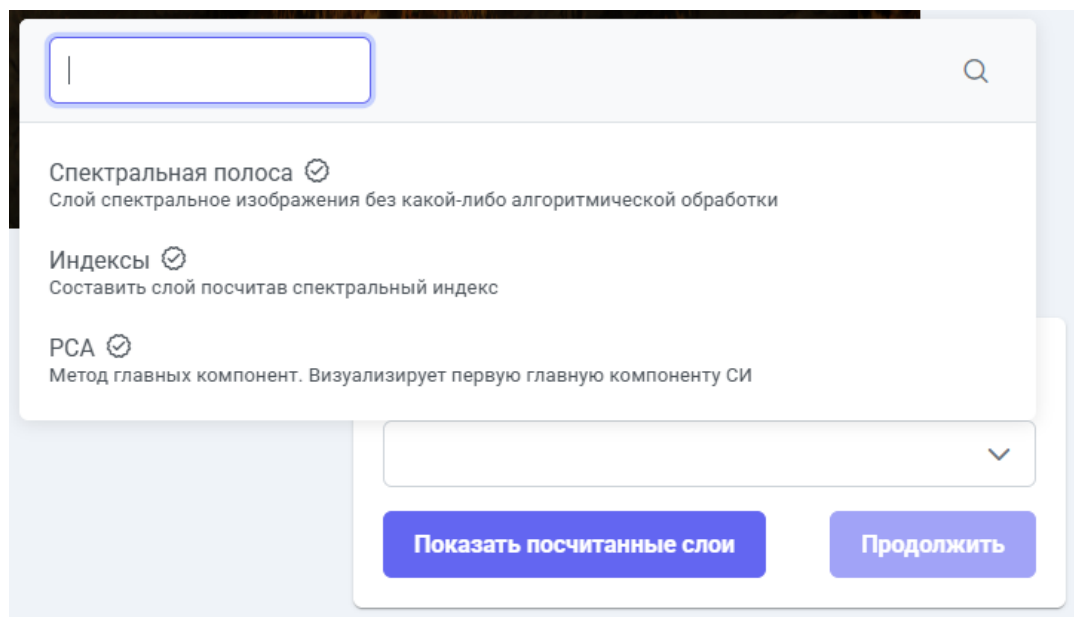


Рисунок 2.33 – Панель добавления слоев: этап выбора типа слоя

При выборе слоя, пользователь сможет перейти на этап настройки параметров слоя. На рисунке 2.34 изображен процесс выбора таких параметров для слоя, представляющего собой вегетационный индекс. Для вегетационного индекса такими параметрами являются – тип индекса, и параметры его вычисления, если таковые имеются.

← **Спектральный индекс**

Название слоя

Индекс SAVI

Почвенный ВИ (Soil Adjusted VI, SAVI) ▼

Относительный индекс; Изовегетационные линии сходятся в квадранте отрицательных значений Red и NIR; Почвенная линия имеет наклон 1 и проходит через точку «0»; Возможные значения: – 1..1.

$$SAVI = \frac{NIR - RED}{NIR + RED + L} \cdot (1 + L)$$

Параметры

L

0.5

L = 0 для наибольшего индекса облиствения, L = 1 для наименьшего, оптимальное значение L = 0.5

Next

Рисунок 2.34 – Панель добавления слоев: этап настройки параметров слоя

Далее пользователю предлагается настроить параметры визуализации слоя. На рисунке 2.35 изображена панель на соответствующем этапе.

← **Параметры визуализации**

Пороговая обработка

CLAHE

Next

Рисунок 2.35 – Панель добавления слоев: этап настройки параметров слоя

Далее пользователю показывается суммарный набор заданных им параметров. На рисунке 2.36 отображен данный этап.

← Сводка

ИНФОРМАЦИЯ О СЛОЕ

Название слоя

Индекс SAVI

Алгоритм

Индексы

Тип слоя

index

ВИЗУАЛИЗАЦИЯ

Цвет

■

rgba(255, 0, 0, 1)

Порог

Отключено

CLANE

Отключено

ПАРАМЕТРЫ

type

L

SAVI

0.5

Назад

✓ Добавить слой

Рисунок 2.36 – Панель добавления слоев: заключительный этап

После этого изображение на странице начнет обновляться, а в списке появится добавленный слой. На рисунке 2.37 отображен данный процесс.

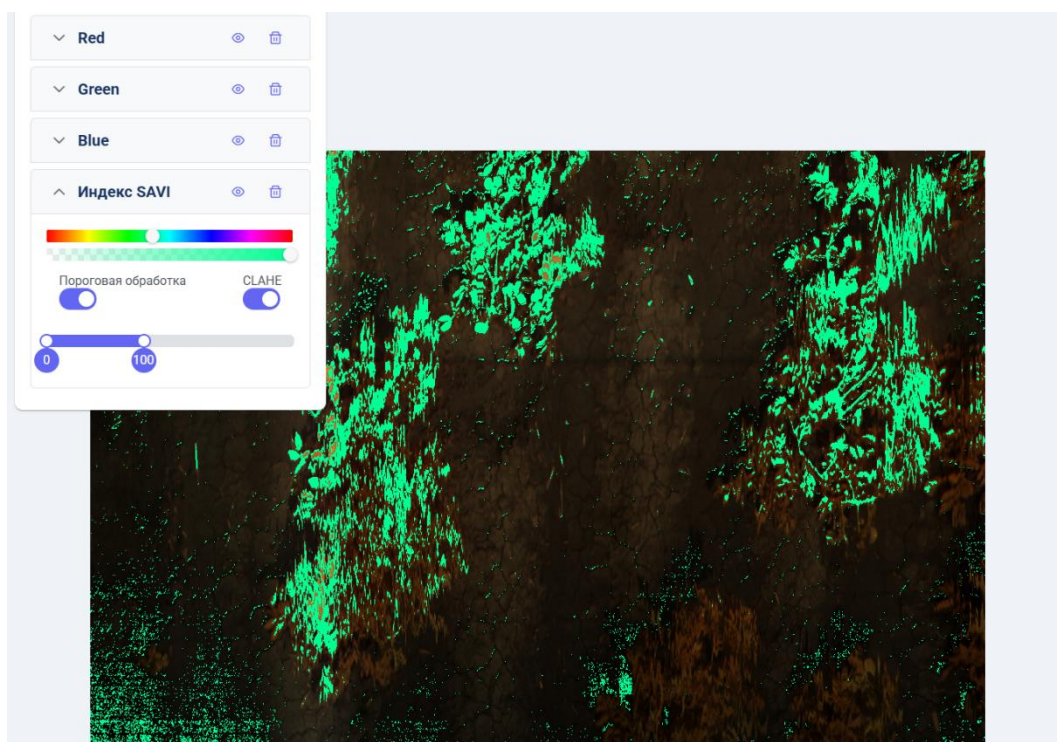
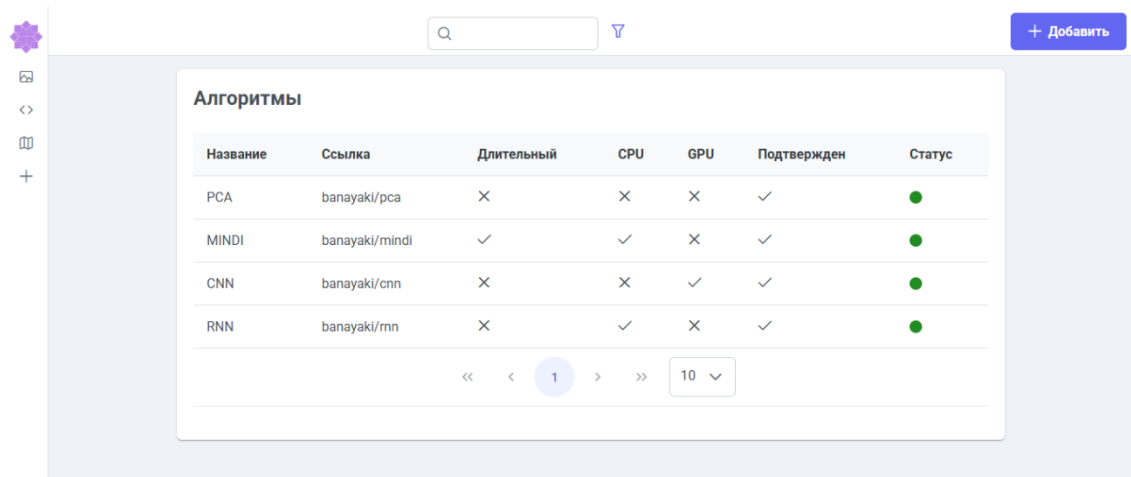


Рисунок 2.37 – Результат добавления слоя

2.9.4 Добавление алгоритмов и взаимодействие с ними

Рассмотрим далее функционал системы, связанный с её функциональным расширением посредством добавления алгоритмов анализа ГСИ. Для этого перейдем на страницу работы с алгоритмами. На рисунке 2.38 представлена страница с таблицей установленных алгоритмов.



Название	Ссылка	Длительный	CPU	GPU	Подтвержден	Статус
PCA	banayaki/pca	×	×	×	✓	●
MINDI	banayaki/mindi	✓	✓	×	✓	●
CNN	banayaki/cnn	×	×	✓	✓	●
RNN	banayaki/rnn	×	✓	×	✓	●

Рисунок 2.38 – Список установленных вычислительных модулей (алгоритмов) в системе

Произведем функциональное расширение системы на примере добавления алгоритма К-средних. Для этого достаточно реализовать один класс, подготовить файл для сборки образа, и описать необходимые для образа зависимости.

Необходимый программный код на языке программирования *python* для реализации алгоритма К-средних представлен далее:

```
1. import numpy as np
2. from celery import Celery
3. from matplotlib import pyplot as plt
4. from spectral import kmeans, imshow
5.
6. from vidis_algorithms_api import Task, CeleryTaskLifecycle
7. from vidis_algorithms_api.core import settings
8.
9.
10. class Kmeans(Task):
11.     name = "kmeans"
12.
13.     def task(self, hyperspecter: np.ndarray, **kwargs) -> np.ndarray:
14.         hyperspecter = np.transpose(hyperspecter, (1, 2, 0)) # To HWC
15.
16.         (m, c) = kmeans(hyperspecter, kwargs.get('clusters'))
17.         yM, xM = np.mgrid[
18.             slice(Hyperspecter.shape[0], 0, -1).
```

```

19.         slice(Hyperspecter.shape[1], 0, -1)
20.     ]
21.     dpi = 80
22.
23.     height, width = xM.shape
24.     figsize = width / float(dpi), height / float(dpi)
25.
26.     fig = plt.figure(frameon=False, figsize=figsize, dpi=dpi)
27.     ax = fig.add_axes([0, 0, 1, 1])
28.     ax.axis('off')
29.
30.     ax.pcolor(xM, yM, m)
31.     fig.canvas.draw()
32.
33.     data = np.fromstring(fig.canvas.tostring_rgb(), dtype=np.uint8, sep='')
34.     data = np.reshape(data, (height, width, 3))[..., 0]
35.     return data
36.
37.
38. celery = Celery(
39.     "algorithms".
40.     backend=settings.CELERY_BACKEND.
41.     broker=settings.CELERY_BROKER
42. )
43.
44.
45. @celery.task(name=Kmeans.name, base=CeleryTaskLifecycle)
46. def your_algorithm(*args, **kwargs):
47.     task = Kmeans()
48.     return task.run(*args, **kwargs)

```

Как видно по вышеприведенному листингу, реализовать расширение достаточно просто, для чего пользователь использует готовый шаблон и реализует логику алгоритма в методе “task”. Далее создадим Docker-образ с данным алгоритмом. Для этого используется следующий код:

```

1. FROM banayaki/vidis-algorithms:1bd7c305
2.
3. WORKDIR /app
4. COPY requirements.txt ./
5.
6. RUN pip install -r requirements.txt --no-cache
7. COPY . ./
8.
9. CMD celery -A algorithm worker --loglevel=INFO

```

Собрав образ и отправив его в доступное хранилище, воспользуемся им посредством разработанной системы. Для этого на странице работы с алгоритмами, необходимо нажать на кнопку “Добавить”, после чего заполним необходимые параметры. На рисунке 2.39 продемонстрирована страница с заполненными параметрами, необходимыми для добавления алгоритма К-средних.

Нажатием на кнопку “Сохранить” инициируем добавление данного алгоритма в систему. После успешного добавления, можем наблюдать новую запись в таблице алгоритмов и уведомление. Данный процесс изображен на рисунке 2.40.

Основные параметры

Название*
KMeans

Ссылка*
banayaki/kmeans

Описание
Реализация алгоритма К-Средних для кластеризации спектрального изображения

Способ отображения параметров ⓘ
Форма с параметрами

☐ Интенсивно использует ресурсы центрального процессора
☐ Интенсивно использует ресурсы графического процессора ☐ Длительно выполняется

Определяемые пользователем параметры алгоритма

Тип: число
Название: Число кластеров
Стандартное значение: 4
Название в программной реализации: clusters

Добавить

Рисунок 2.39 – Заполненные параметры необходимые для добавления алгоритма через веб-интерфейс

Успешно
Алгоритм был успешно добавлен

Алгоритмы

Название	Ссылка	Длительный	CPU	GPU	Подтвержден	Статус
PCA	banayaki/pca	×	×	×	✓	●
MINDI	banayaki/mindi	✓	✓	×	✓	●
CNN	banayaki/cnn	×	×	✓	✓	●
RNN	banayaki/rnn	×	✓	×	✓	●
KMeans	banayaki/kmeans	×	×	×	×	●

Рисунок 2.40 – Алгоритм был успешно добавлен в систему

Далее попробуем применить добавленный алгоритм. Перейдем на страницу анализа ГСИ. На рисунке 2.41 изображена панель добавления нового слоя, в которой можно увидеть добавленный алгоритм.

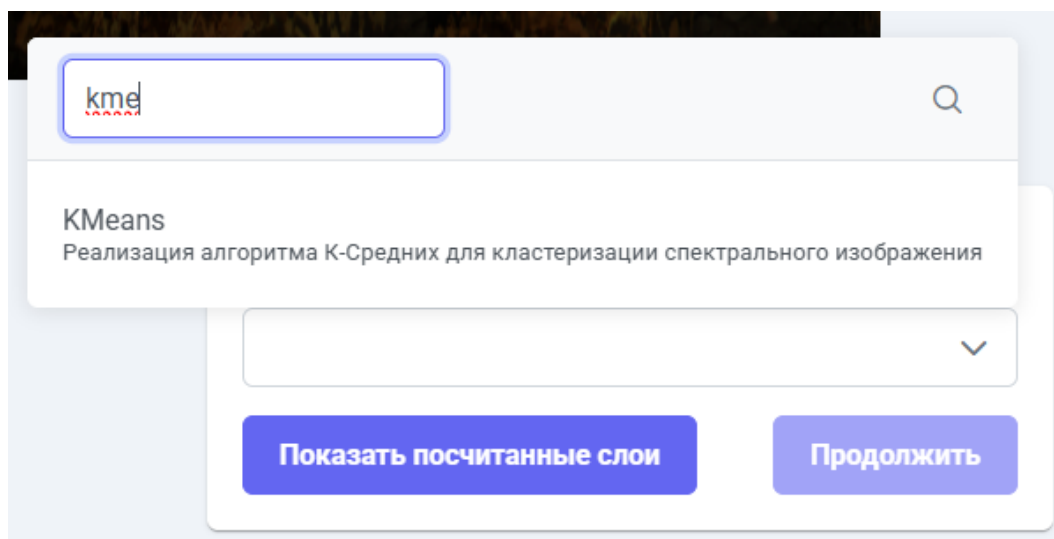


Рисунок 2.41 – На панели добавления слоев появился добавленный алгоритм

Далее, введя параметры слоя, инициируем начало его вычисления. Спустя некоторое количество времени любые пользователи смогут добавить вычисленный по заданным параметрам слой в конфигурацию. Для этого достаточно либо повторить запрос, либо нажатием на кнопку “Показать вычисленные слои” вызвать специальное диалоговое окно, отображающее список вычисленных слоев. На рисунке 2.42 показан результат вывода данного окна.

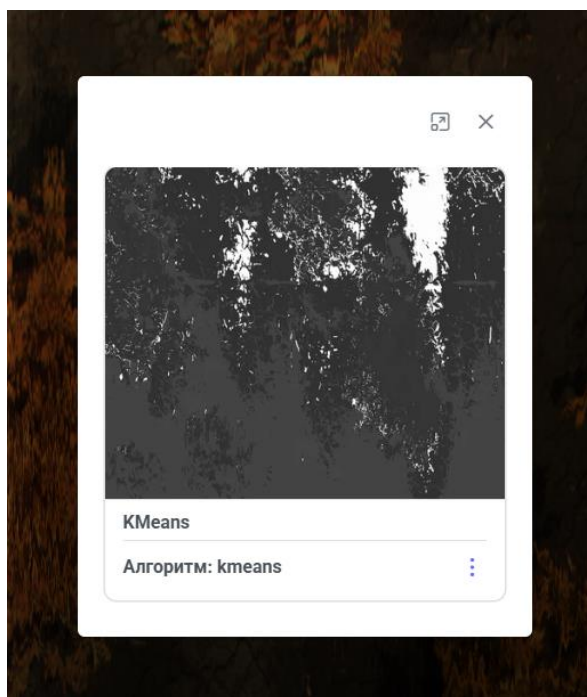


Рисунок 2.42 – Диалоговое окно вычисленных слоев

Выбрав слой и указав параметры его визуализации, изображение на странице обновится, и пользователь сможет увидеть результаты кластеризации, наложенные поверх текущей конфигурации слоев. На рисунке 2.43 показан итоговый результат.

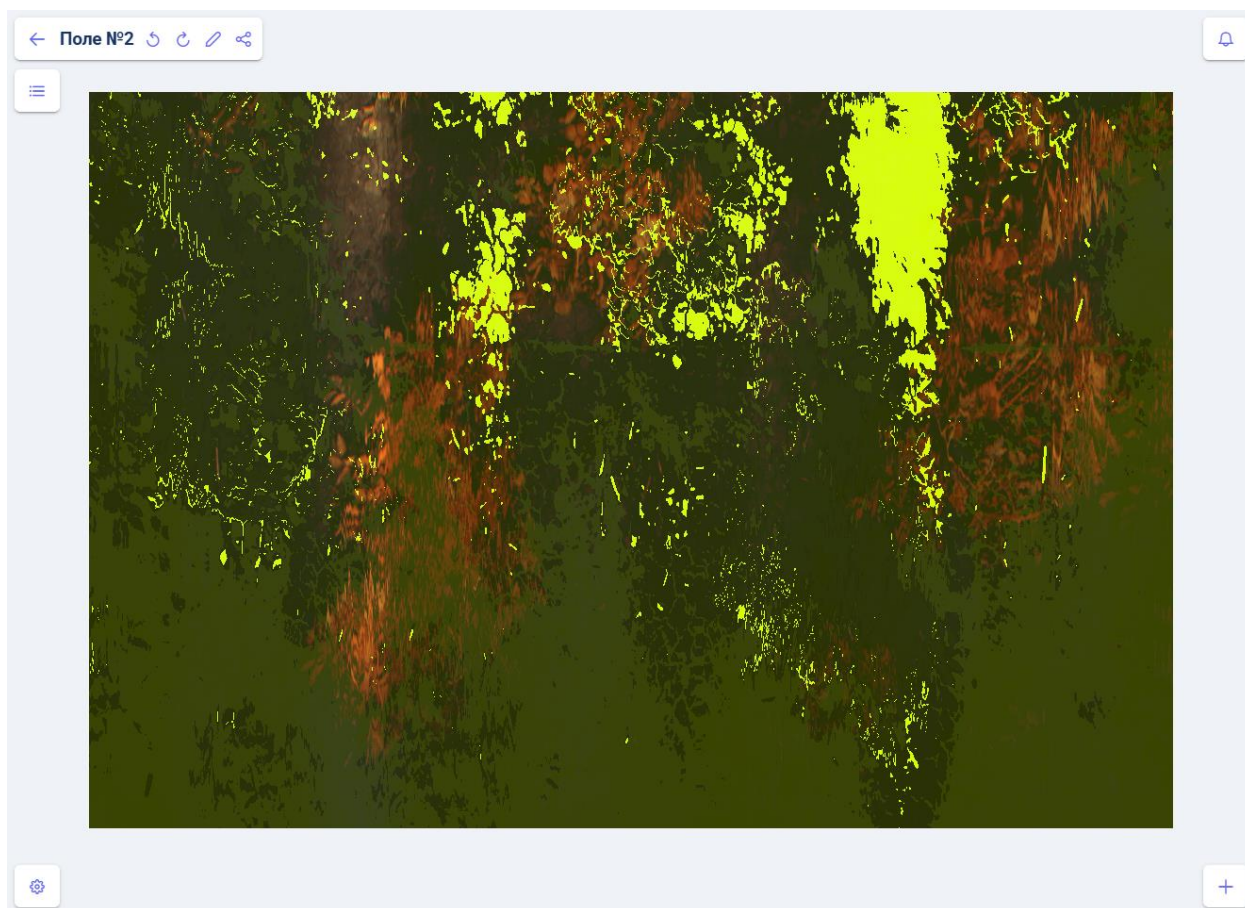


Рисунок 2.43 – Результат визуализации конфигурации слоев, состоящей из RGB слоев и результатов кластеризации методом К-средних

2.10 Интеллектуальная обработка гиперспектральных изображений

Одним из направлений автоматизации анализа гиперспектральных изображений является использование моделей машинного обучения, обучаемых на данных, уже загруженных в систему. Такой подход позволяет пользователю не только выполнить анализ отдельного изображения, но и сформировать модель, пригодную для последующей обработки других гиперспектральных изображений со сходной структурой гиперспектральных данных. В разработанной системе данный сценарий реализован в виде последовательности операций, включающей выбор источника разметки, настройку модели машинного обучения, запуск процедуры обучения, сохранение обученной модели и ее последующее повторное использование.

Рассмотрим сценарий, при котором пользователь решает обучить модель машинного обучения на одном ГСИ, а затем применить полученную модель для анализа другого изображения. Для реализации данного сценария в системе предусмотрен алгоритм обучения моделей на основе заданного источника разметки. Под источником разметки в данном случае понимается результат работы алгоритма или действия пользователя, позволяющие каждому пикселю или группе пикселей исходного изображения сопоставить некоторую метку класса. В качестве такого источника могут использоваться результаты автоматической сегментации, информативные индексы, результаты кластеризации, ручная разметка, а также специализированные нейросетевые методы, выделяющие замкнутые контуры объектов на изображении.

На первом этапе пользователь выбирает ГСИ, перейдя на страницу его обработки, на основе которого требуется сформировать обучающую выборку, и указывает источник разметки. Система поддерживает несколько вариантов получения разметки. Для примера, пусть разметка будет сформирована на основе результата работы нейронной сети, предназначенной для выделения замкнутых контуров на изображении. За счет этого, обучение модели не требует обязательного предварительного создания размеченного набора данных вне разработанной системы и может выполняться непосредственно в рамках пользовательского сценария анализа изображения. На рисунке 2.44 изображен соответствующий пользовательский интерфейс, соответствующий этапу выбора источника разметки для обучения.

Разметка для обучения

Источник разметки

Использовать растр посчитанного слоя (k-means, MINDI, FastSA... ▾

При смене источника список сохранённых регионов очищается.

Растр посчитанного слоя

Значения пикселей из сохранённого результата (.пру) на сервере сопоставляются с метками классов (k-means, MINDI, FastSAM и т.д.).

Обновить

Слой разметки

fastsam (fastsam) ▾

Путь к растру: 30-10-202115-59_19_03_2026__08_46_23_148464/custom_algorithm/fastsam_2026-05-28_04:57:36.972712.npy

Форма 781×1150, тип uint8, значения 0.000–4.000, ~5 уникальных (оценка)

Доступно 4 посчитанных слоёв с растровыми данными.

Выберите растр посчитанного слоя в качестве разметки для обучения.

Рисунок 2.44 – Пользовательский интерфейс выбора разметки

После выбора источника разметки пользователь переходит к настройке модели машинного обучения. На рисунке 2.45 продемонстрирован соответствующий интерфейс. На данном этапе из перечня доступных моделей выбирается требуемый алгоритм, после чего при необходимости задаются или изменяются его гиперпараметры. Такой подход позволяет использовать единый механизм обучения как для базовых моделей машинного обучения, так и для пользовательских алгоритмов. После завершения настройки пользователь инициирует процесс обучения модели.

← Алгоритм: auto-analysis — автоанализ

Название слоя

MLP - V2

Классификатор

Выберите обучаемый классификатор. Каждый пиксель классифицируется по спектральному вектору с помощью полигонов вручную или сохранённого раstra другого посчитанного слоя.

Модель сегментации

MLP (neural network) ▾

Feed-forward network on spectral vectors.

Гиперпараметры модели

Hidden layers (comma-separated widths)

100, 50

Activation

ReLU ▾

L2 penalty

0

Initial learning rate

0.001

Max iterations

200

Random seed

42

Параметры обучения

☒ Разделение train/test перед финальным обучением

Доля теста

0.25

Seed

42

Стратифицированное разбиение

☒

Радиус подвыборки

5

Нормализация

Метод нормализации данных

Стандартная (z-score) ▾

Применяется перед обучением и предсказанием в автоанализе.

Рисунок 2.45 – Пользовательский интерфейс выбора модели машинного обучения

Процедура обучения выполняется в асинхронном режиме. Это позволяет пользователю продолжать работу с системой и не блокирует интерфейс на время выполнения обучения. После завершения обучения система формирует уведомление об успешном окончании. В результате выполнения процедуры пользователь получает два результата: карту предсказаний для изображения, на котором выполнялось обучение, и

обученную модель машинного обучения. Карта предсказаний позволяет визуально оценить результат применения модели к исходному изображению и сопоставить его с выбранным источником разметки. На рисунке 2.46 изображен результат обучения модели.

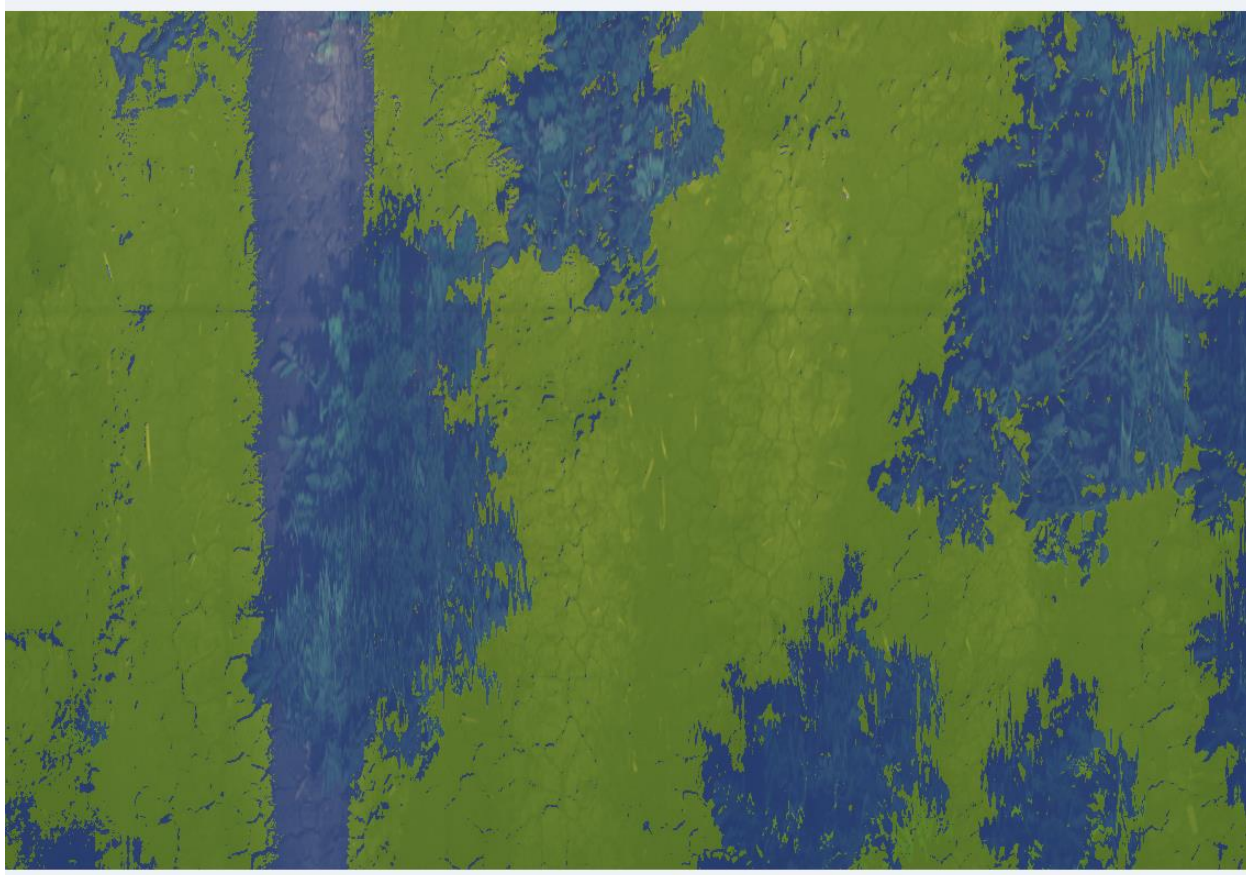
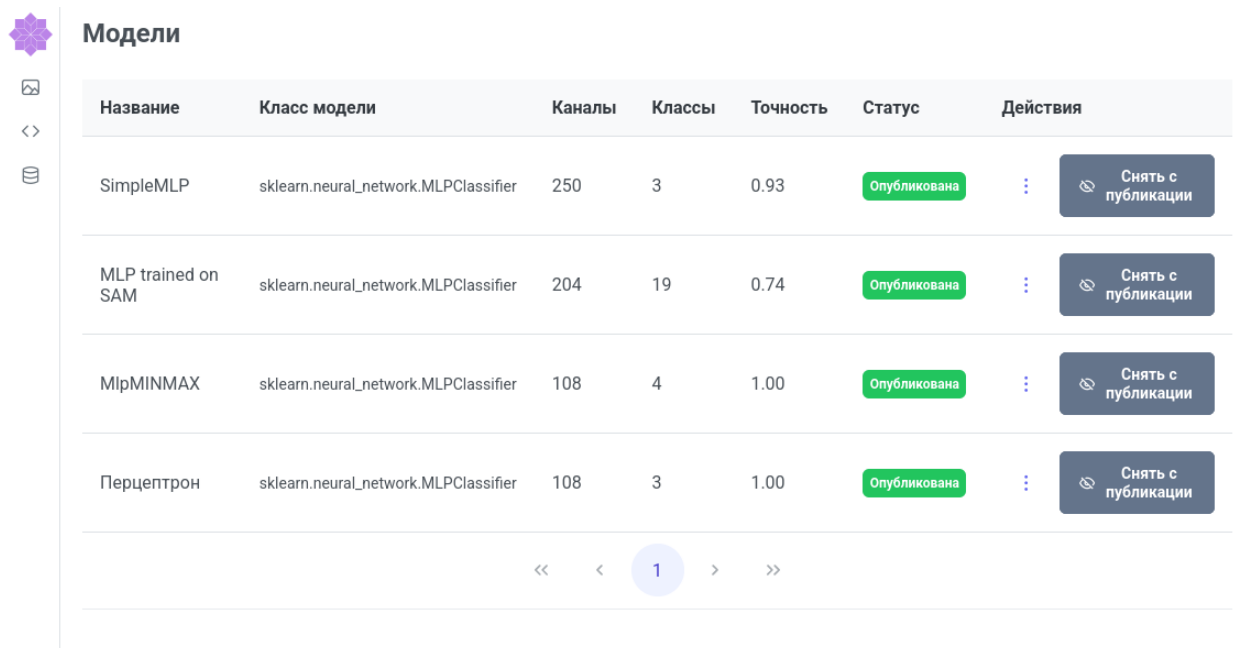


Рисунок 2.46 – Результат обучения модели машинного обучения

Для просмотра сведений об обученной модели пользователь переходит на страницу моделей. На рисунке 2.47 представлен пример содержания данной страницы. На ней отображаются основные характеристики сформированной модели: наименование, число спектральных каналов, использованных при обучении, число выделяемых классов, а также значение метрики качества, полученное в процессе обучения или проверки модели. Эти сведения необходимы для последующего выбора модели и контроля возможности ее применения к другим ГСИ. В частности, число спектральных каналов используется как одно из условий совместимости модели с новыми данными,

поскольку модель может быть применена только к изображениям, имеющим соответствующее число спектральных полос.



Модели

Название	Класс модели	Каналы	Классы	Точность	Статус	Действия
SimpleMLP	sklearn.neural_network.MLPClassifier	250	3	0.93	Опубликована	⋮ Снять с публикации
MLP trained on SAM	sklearn.neural_network.MLPClassifier	204	19	0.74	Опубликована	⋮ Снять с публикации
MlpMINMAX	sklearn.neural_network.MLPClassifier	108	4	1.00	Опубликована	⋮ Снять с публикации
Перцептрон	sklearn.neural_network.MLPClassifier	108	3	1.00	Опубликована	⋮ Снять с публикации

«» < 1 > «»

Рисунок 2.47 – Список обученных моделей

После проверки сведений о модели пользователь может опубликовать ее. Под публикацией понимается перевод модели в состояние, при котором она становится доступной для повторного использования при анализе других ГСИ. Такой механизм позволяет отделить экспериментальные модели, полученные в ходе отдельных запусков обучения, от моделей, которые пользователь считает пригодными для дальнейшего применения. После публикации модель включается в перечень доступных инструментов анализа и может быть выбрана при обработке любого изображения, удовлетворяющего требованиям совместимости по числу спектральных каналов. На рисунке 2.48 представлен пример использования обученной раннее модели.

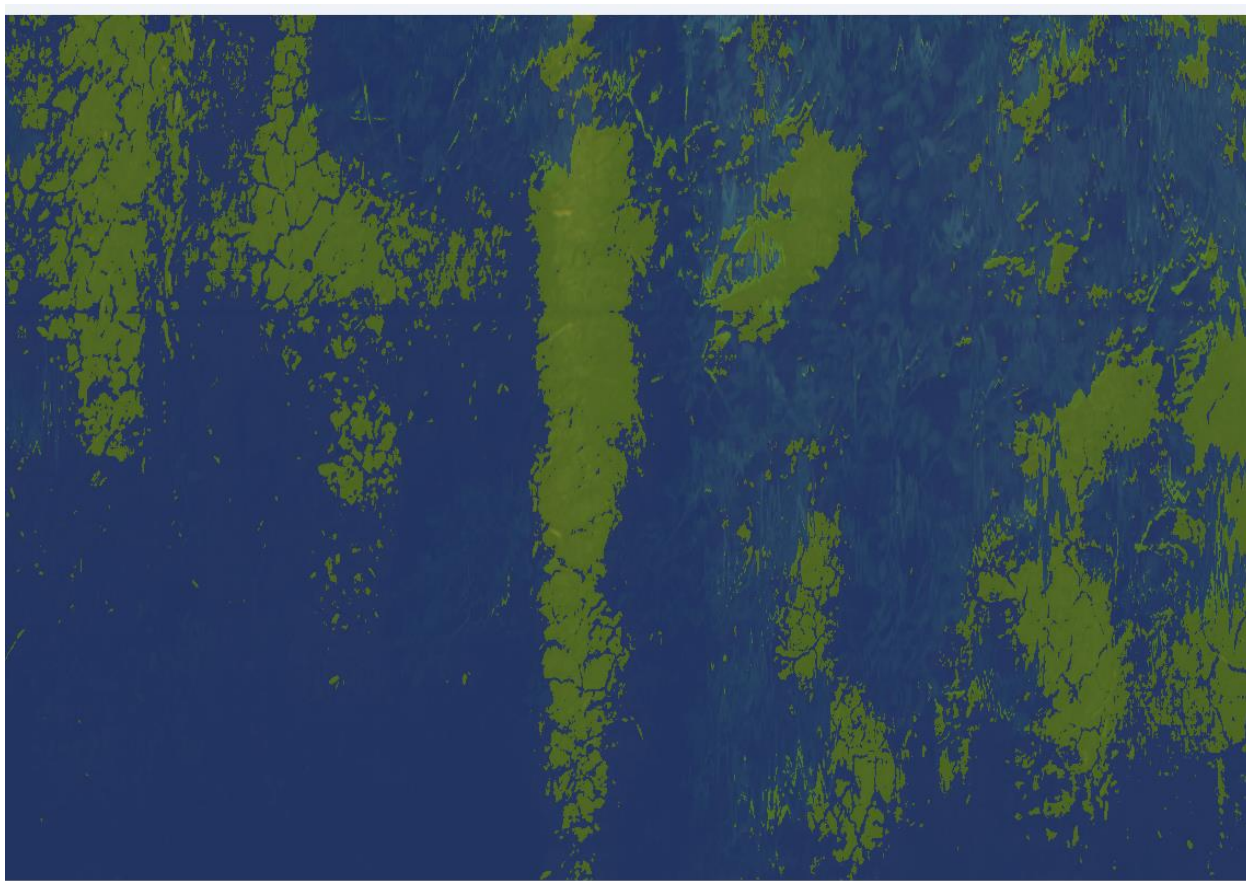


Рисунок 2.48 – Применение обученной модели на другом ГСИ

Таким образом, реализованный сценарий интеллектуальной обработки ГСИ обеспечивает переход от разового анализа отдельного изображения к формированию переиспользуемых моделей. Его особенностью является возможность использовать различные источники разметки, включая результаты других алгоритмов анализа, что снижает зависимость пользователя от заранее подготовленных размеченных наборов данных. При этом система обеспечивает полный цикл работы с моделью: формирование обучающей выборки, настройку алгоритма, запуск обучения, получение карты предсказаний, регистрацию модели, публикацию и последующее применение к другим данным. Это расширяет возможности автоматизированной обработки ГСИ и позволяет использовать результаты анализа одного изображения для обработки последующих изображений.

2.11 Экспериментальное исследование характеристик разработанной интеллектуальной системы

В рамках данного пункта будет произведено количественное исследование разработанной интеллектуальной системы. Предметом исследования является оценка доступности, отказоустойчивости и быстродействия системы. Для этого был произведен ряд нагрузочных тестов и собран набор необходимых метрик.

Нагрузочное тестирование осуществлялось в пользовательском окружении, сконфигурированном следующим образом:

- для развертывания кластера *Kubernetes* использовался легковесный дистрибутив *k3s* [123];
- кластер состоял из двух узлов, размещённых на отдельных серверах, находящихся в одной локальной сети;
- в кластере суммарно было доступно 84 процессорных ядра и 2142 ГБ оперативной памяти;
- для передачи данных между узлами использовался облачный инструмент организации постоянных хранилищ – *Longhorn* [124];
- нагрузочное тестирование проводилось с устройства, находящегося вне локальной сети кластера. Согласно данным утилиты *tracert*, расстояние от этого устройства до кластера составляло 8 сетевых хопов;
- автоматическое горизонтальное масштабирование было настроено по следующему правилу: создание новой реплики происходит при превышении средней нагрузки на выделенные ресурсы процессора порога в 40%.

Конфигурация, используемая для нагрузочного тестирования представлена на рисунке 2.49.

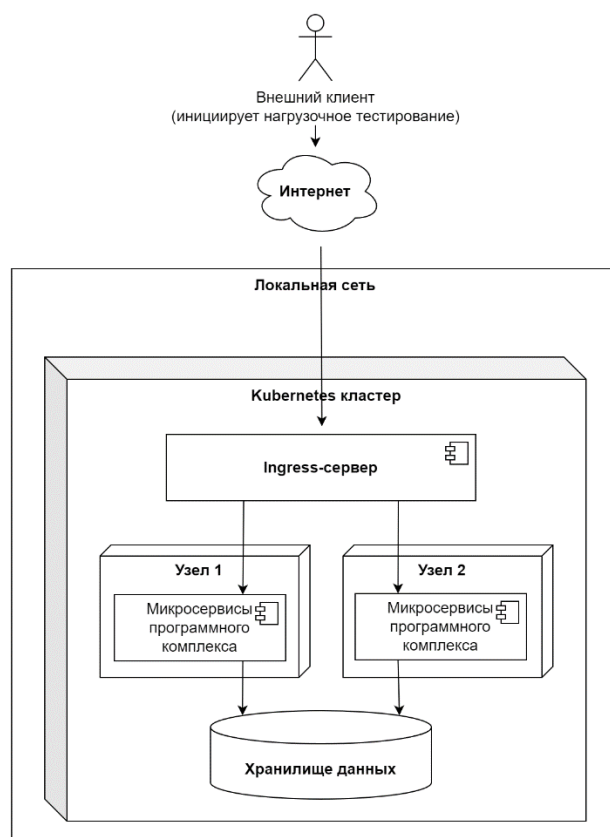


Рисунок 2.49 – Диаграмма, иллюстрирующая используемую для проведения нагрузочного тестирования конфигурацию

Для проведения нагрузочного тестирования использовалась утилита Locust. С её помощью с клиентского устройства отправлялись запросы к базовым микросервисам системы:

- 1) получение списка загруженных в систему ГСИ;
- 2) получение списка загруженных в систему алгоритмов анализа данных;
- 3) обработка фрагмента гиперспектрального изображения с помощью вегетационного индекса NDVI и формирование результата, сочетающего в себе RGB представление гиперспектрального изображения и NDVI индекс с пороговой обработкой.

С помощью данной утилиты имитировался постепенный рост числа активных подключений (пользователей), бесперебойно отправляющих запросы к системе.

Перед тем как перейти к рассмотрению результатов нагрузочного тестирования, оценим функциональную расширяемость системы. Данную оценку можно провести по метрике МТТА (Mean Time To Availability), которая показывает время интеграции нового алгоритма в систему. Среднее значение МТТА по результатам 10 интеграций алгоритмов K-means и PCA составило 1.51 секунды (без учета первичной загрузки образа контейнера). Данный показатель отражает время запуска и интеграции контейнера с системой, являясь минимальным в рамках используемого кластера, так как указанные алгоритмы не требуют предварительной инициализации. Для интеграции новых алгоритмов в систему не требуется изменения исходного кода ядра, что отражается в метриках LoC = 0 (Lines of Code) и времени простоя системы равным нулю (Downtime). Процесс интеграции алгоритмов K-means, PCA, а также уникальных алгоритмов INDI, MINDI, A-MINDI и нейросетевых алгоритмов показал идентичные шаги, подтверждая высокую гибкость и универсальность архитектуры.

Состояние кластера перед началом нагрузочного тестирования и настройки автоматического горизонтального масштабирования (*hpa* – *horizontal pod autoscaler*): заданные пороги использования центрального процессора (ЦП) и максимальное число реплик – представлены в таблице 2.11.

Таблица 2.11 – Изначальное состояние и настройки автоматического горизонтального масштабирования

Название <i>hpa</i>	Пороги ЦП	Минимально реплик	Максимально реплик	Исходное число реплик
Организации хранения ГСИ	0%/40%	1	25	1
Взаимодействия с алгоритмами	0%/40%	1	25	1
Базовой обработки ГСИ	0%/40%	1	25	1

В таблице 2.12 приведены результаты замеров метрики RPS (запросов в секунду), задержек и количества отказов при нагрузочном тестировании

каждого микросервиса по отдельности. В ходе тестирования число одновременных подключений постепенно увеличивалось от 1 до 200.

Таблица 2.12 – Результаты независимого нагрузочного тестирования отдельных компонентов системы

Микросервис	Организации хранения ГСИ	Взаимодействия с алгоритмами	Базовой обработки ГСИ
Отказов	<i>не зафиксировано</i>	<i>не зафиксировано</i>	<i>не зафиксировано</i>
Средняя задержка (мс.)	187,74	146,03	980,36
99% перцентиль (мс.)	1300	360	8000
RPS	1410,4	914,6	142,8
Число реплик	9/25	5/25	25/25
Пороги ЦП	43%/40%	42%/40%	53%/40%

В таблице 2.13 отражены результаты совокупного нагрузочного тестирования, при котором все сервисы тестировались одновременно, а запросы отправлялись равномерно.

Таблица 2.13 – Результаты совокупного нагрузочного тестирования

Микросервис	Организации хранения ГСИ	Взаимодействия с алгоритмами	Базовой обработки ГСИ
Отказов	<i>не зафиксировано</i>	<i>не зафиксировано</i>	<i>не зафиксировано</i>
Средняя задержка (мс.)	44,14	40,91	670,75
99% перцентиль (мс.)	79	73	5700
RPS	120,8	121,2	117,0
Число реплик	7/25	1/25	25/25
Пороги ЦП	39%/40%	36%/40%	60%/40%

В таблице 2.14 представлены результаты тестирования, имитирующего поведение реальных пользователей, при котором интервал между запросами составлял от 1 до 5 секунд.

Таблица 2.14 – Результаты тестирования, имитирующего реальных пользователей

Микросервис	Организации хранения ГСИ	Взаимодействия с алгоритмами	Базовой обработки ГСИ
Отказов	<i>не зафиксировано</i>	<i>не зафиксировано</i>	<i>не зафиксировано</i>
Средняя задержка (мс.)	45,81	40,79	315,94
99% перцентиль (мс.)	44	39	700
RPS	21,0	22,5	22,1
Число реплик	1/25	1/25	5/25
Пороги ЦП	33%/40%	5%/40%	34%/40%

Во всех сценариях тестирования: независимые тесты, совокупная нагрузка, имитация реального пользовательского поведения – не было зафиксировано ни одного отказа среди всех микросервисов. Это свидетельствует о высокой надёжности архитектуры и корректной работе механизмов распределения нагрузки и масштабирования.

При независимом нагрузочном тестировании обнаружено, что микросервис базовой обработки ГСИ имеет наименьшую скорость обработки пользовательских запросов – 142,8 запросов в секунду. Это указывает на ограниченную пропускную способность, связанную с вычислительной сложностью обработки ГСИ.

При совокупном тестировании (одновременной нагрузки на все микросервисы) значения RPS оказались значительно ниже. Однако наблюдалось и существенное уменьшение средних и максимальных задержек. Это объясняется равномерным распределением пользовательских запросов и снижением пиковых нагрузок на отдельные компоненты.

В режиме, имитирующем реальное поведение пользователей, показатели задержек и отказов находятся на минимальных значениях, что подтверждает достаточность текущей архитектуры для типичных пользовательских сценариев.

Механизм автоматического горизонтального масштабирования корректно реагировал на изменение нагрузки: при росте нагрузки число реплик сервисов увеличивалось вплоть до максимальных значений (например, до 25/25 у микросервиса базовой обработки ГСИ). Для сервисов с меньшей нагрузкой рост числа реплик был умеренным, что обусловлено меньшей вычислительной сложностью выполняемых ими операций.

При нагрузке, приближенной к реальной, количество реплик микросервисов работы с данными оставалось минимальным, а для микросервиса базовой обработки данных использовалось 5 реплик, что

свидетельствует об экономичном использовании ресурсов при отсутствии пиковых нагрузок.

Фактическая нагрузка на CPU для всех сервисов оставалась вблизи заданного порога масштабирования (40%), что говорит о сбалансированном распределении нагрузки. Более точная настройка порогов и числа реплик, а также использование альтернативных стратегий масштабирования, возможны благодаря использованию Kubernetes, что позволяет адаптировать разработанную систему для развёртывания практически в любых средах – как на собственных серверах, так и у облачных провайдеров. При этом сохраняется возможность вертикального масштабирования за счёт добавления новых узлов в кластер.

На примере микросервиса организации хранения ГСИ рассмотрим работу горизонтального масштабирования во времени. На рисунке 2.50 представлена динамика результатов нагрузочного тестирования микросервиса.

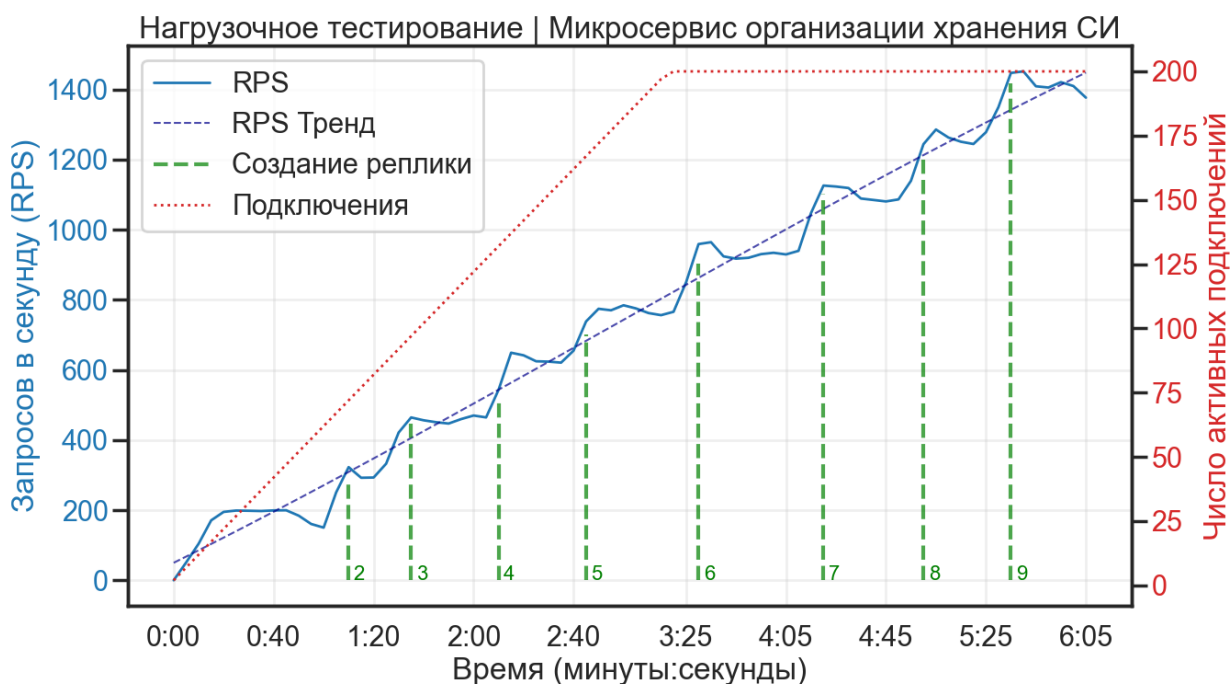


Рисунок 2.50 – Динамика горизонтального масштабирования системы во время нагрузочного тестирования

В начале тестирования значение RPS постепенно увеличивается вслед за ростом числа подключений. По мере достижения определённых пороговых значений нагрузки происходит автоматическое масштабирование: запускаются дополнительные реплики микросервиса, что сопровождается очередными скачками производительности. После каждого увеличения числа реплик наблюдается очередной рост пропускной способности, что подтверждает корректную работу механизма автоматического горизонтального масштабирования.

Представленный график свидетельствует о линейной масштабируемости микросервиса организации хранения СИ в условиях возрастающей нагрузки, а также демонстрирует отсутствие резких спадов в производительности и каких-либо признаков деградации сервиса в процессе тестирования. Это подтверждает эффективность архитектуры и реализованных механизмов масштабирования для поддержания необходимого уровня быстродействия при увеличении количества одновременных пользователей.

Для общей оценки масштаба разработанной системы была выполнена оценка структуры исходного кода, без учета файлов библиотек. В состав системы входит 874 файла, содержащих 64 191 строку, из которых 54 993 строки относятся непосредственно к программному коду, 2 543 строки — к комментариям, а 6 655 строк составляют пустые строки, используемые для структурирования исходных файлов. Совокупная оценка цикломатической сложности составила 3 242. Полученные значения показывают, что разработанная система представляет собой многокомпонентную программную систему, включающую серверную часть, клиентский интерфейс, алгоритмические модули, средства конфигурации, контейнеризации и развертывания.

Наибольший объем исполняемого кода приходится на серверную и алгоритмическую части, реализованные преимущественно на языке Python, а

также на клиентскую часть, реализованную с использованием TypeScript. Значительный объем YAML, Docker и Shell файлов отражает наличие инфраструктурного слоя, обеспечивающего конфигурацию, контейнеризацию и развертывание системы. Таким образом, приведенные показатели позволяют оценить не только объем реализации, но и технологическую неоднородность системы.

2.12 Выводы и результаты по второму разделу

На основе сформулированных требований разработана архитектура интеллектуальной системы обработки гиперспектральных изображений, отличающаяся выделением независимых компонентов управления исходными гиперспектральными данными, результатами анализа и визуализации, постановки вычислительных задач и выполнения алгоритмов. Такое архитектурное решение обеспечивает горизонтальное масштабирование отдельных компонентов системы, распределённое выполнение ресурсоёмких операций и функциональное расширение за счёт интеграции пользовательских алгоритмов и моделей машинного обучения.

Разработаны модели основных компонентов интеллектуальной системы, а также архитектурные представления системы, описывающие взаимодействие микросервисов, организацию хранения данных, выполнение алгоритмов анализа и передачу результатов обработки пользователю.

Реализована интеллектуальная система, включающая веб-интерфейс, серверные компоненты, механизм управления вычислительными задачами, средства взаимодействия с пользовательскими алгоритмами и прокси-сервер. Разработанная система обеспечивает централизованное хранение гиперспектральных изображений, интерактивную визуализацию данных, запуск алгоритмов анализа, асинхронное выполнение вычислительных задач, сохранение и повторное использование ранее полученных результатов.

Показана возможность функционального расширения интеллектуальной системы за счёт подключения алгоритмов анализа гиперспектральных

изображений в виде самостоятельных вычислительных модулей. Среднее значение метрики МТТА при интеграции алгоритмов K-means и PCA по результатам 10 запусков составило 1,51 с без учёта первичной загрузки образа контейнера. При этом для подключения новых алгоритмов не требуется изменение исходного кода существующих компонентов системы: значение LoC составило 0, а время простоя системы — 0. Это подтверждает новизну предложенного архитектурного решения в части организации расширяемой интеллектуальной системы обработки гиперспектральных изображений.

Проведённое нагрузочное тестирование показало, что во всех рассмотренных сценариях отказов компонентов системы не зафиксировано. При независимом тестировании отдельных компонентов максимальная пропускная способность составила 1410,4 запроса в секунду для компонента организации хранения гиперспектральных изображений, 914,6 запроса в секунду для компонента взаимодействия с алгоритмами и 142,8 запроса в секунду для компонента базовой обработки гиперспектральных изображений.

При совокупном нагрузочном тестировании средняя задержка составила 44,14 мс для компонента организации хранения, 40,91 мс для компонента взаимодействия с алгоритмами и 670,75 мс для компонента базовой обработки. В режиме, имитирующем поведение реальных пользователей, средняя задержка составила соответственно 45,81 мс, 40,79 мс и 315,94 мс, при этом 99-й перцентиль задержки не превысил 700 мс для компонента базовой обработки.

В состав реализованной интеллектуальной системы, без учёта файлов библиотек, входит 874 файла, содержащих 64 191 строку, из которых 54 993 строки относятся непосредственно к программному коду. Полученные значения характеризуют разработанную систему как многокомпонентное программное решение, включающее серверную часть, клиентский интерфейс, алгоритмические модули, средства конфигурации, контейнеризации и развёртывания.

Таким образом, разработанная интеллектуальная система обеспечивает полный цикл работы с ГСИ: централизованное хранение, интерактивную визуализацию, запуск алгоритмов анализа, асинхронную обработку вычислительных задач и расширение функциональности за счёт подключения новых модулей. Полученные значения МТТА, LoC, времени простоя, задержек, RPS, а также отсутствие отказов при нагрузочном тестировании подтверждают, что предложенные архитектурные решения обеспечивают работоспособность, расширяемость, отказоустойчивость и масштабируемость программного обеспечения автоматизированной обработки ГСИ.

По материалам данного раздела опубликована работа [125]. Также получены государственные свидетельства о регистрации программы для ЭВМ [126, 127].

3 Разработка алгоритмов и методов хранения, визуализации и обработки гиперспектральных изображений

Основу разработанной интеллектуальной системы составляет совокупность алгоритмов и методов, предназначенных для решения задач хранения, визуализации и обработки ГСИ:

- алгоритм формирования информативного индекса (INDI);
- алгоритм формирования ансамбля информативных индексов (MINDI);
- алгоритм автоматического формирования ансамбля информативных индексов для решения задачи сегментации ГСИ (A-MINDI);
- алгоритм анализа ГСИ методом оценки схожести с эталоном;
- алгоритм анализа ГСИ с помощью вегетационных индексов;
- алгоритм анализа ГСИ методом кластеризации K-средних;
- метод эффективного хранения ГСИ;
- метод мозаичной обработки ГСИ;
- методы предварительной нормализации ГСИ;
- алгоритм цветосинтеза для визуализации ГСИ.

Данная глава посвящена описанию разработанных алгоритмов и методов, обеспечивающих практическую реализацию интеллектуальной системы обработки ГСИ. При их разработке основной акцент был сделан не столько на построении отдельных вычислительных процедур, сколько на формировании ядра системы, позволяющей автоматизировать ключевые этапы обработки гиперспектральных данных.

3.1 Методика формирования информативного индекса

Одним из базовых инструментов анализа ГСИ являются – спектральные индексы. Они представляют собой вычисляемые функции от значений спектральных каналов, позволяющие количественно характеризовать свойства наблюдаемых объектов и тем самым упрощать их различение, обнаружение или классификацию. Несмотря на широкое распространение спектральных индексов, их практическое применение по-прежнему в

значительной степени носит ручной и эвристический характер: пользователь должен самостоятельно выбирать подходящую формулу, определять набор используемых спектральных каналов и оценивать пригодность полученного индекса для решаемой задачи.

Согласно данным сайта IDB [128], в настоящее время известно не менее 519 спектральных индексов. Однако наличие большого числа известных индексов само по себе не решает задачу автоматизации анализа. Во-первых, существующие индексы ориентированы на ограниченный круг типовых объектов и сценариев применения. Во-вторых, отсутствуют общепринятые методы конструирования новых индексов для произвольных прикладных задач. В-третьих, применимость конкретного индекса может существенно зависеть от используемого сенсора, спектрального разрешения и особенностей условий съемки. В результате даже при наличии большого числа опубликованных индексов пользователь по-прежнему вынужден вручную подбирать наиболее подходящий вариант или отказываться от их использования.

Улучшить доступность применения спектральных индексов для анализа данных можно благодаря систематизации существующих индексов [129] или разработке алгоритмов их автоматического формирования.

Решение проблемы формирования спектральных индексов позволит перейти от ручного конструирования индексов к автоматическому поиску информативных спектральных признаков, что является одним из необходимых условий создания программных средств, доступных пользователям без специальной подготовки.

С этой целью была разработана методика формирования информативного индекса для различения заданных пользователем объектов на гиперспектральных данных [130].

Пусть задано гиперспектральное изображение:

$$X = \{x(u) \mid u \in \Omega\},$$

где $\Omega \subset \mathbb{Z}^2$ – множество пикселей изображения,

$x(u) \in \mathbb{R}^B$ – спектральный вектор пикселя u ,

B – число спектральных каналов.

Тогда всё изображение может рассматриваться как матрица или тензор гиперспектральных данных, а все операции анализа сводятся к операциям над векторами признаков, матрицами наблюдений и матрицами статистик. Такое представление является важным, поскольку обеспечивает вычислительную эффективность разрабатываемых алгоритмов, их естественную векторизацию и возможность параллельной реализации.

Пусть также задано множество целевых классов:

$$C = \{c_1, c_2, \dots, c_K\}.$$

Разметка изображения определяется отображением:

$$Y: \Omega \rightarrow C,$$

которое каждому пикселю ставит в соответствие метку класса.

Под **индексом** далее понимается функция:

$$I: \mathbb{R}^B \rightarrow \mathbb{R},$$

которая отображает спектральный вектор пикселя в числовую характеристику, предназначенную для различения объектов или классов объектов на ГСИ.

Применение индекса I к изображению X задает матрицу значений индекса:

$$Z_I: \Omega \rightarrow \mathbb{R}, \quad Z_I(u) = I(x(u))$$

Пусть $\mathfrak{I}$ – множество допустимых индексов. Тогда задача формирования информативного индекса заключается в построении отображения:

$$\Psi_I: (X, Y) \rightarrow I^*, \quad I^* \in \mathfrak{I},$$

которое по гиперспектральному изображению и разметке позволяет определить **информативный индекс** I^* , наиболее подходящий для различения заданных классов объектов.

Фундаментом данной методики является предположение о том, что, задав **формулу индекса** и **критерий оценки его информативности**, можно построить алгоритм, способный автоматически определить такой индекс,

который наилучшим образом разделяет заданные классы объектов. Индекс, найденный в результате такой процедуры, называется *информативным индексом*, а спектральные слои, используемые для его вычисления – *информативными слоями*. Возможность варьирования как формулы индекса, так и критерия его оценки обеспечивает универсальность методики и позволяет адаптировать ее к различным задачам анализа.

Предлагаемая методика описывает процесс формирования информативного индекса на основе полного перебора допустимых комбинаций спектральных слоев. На рисунке 3.1 представлена диаграмма, иллюстрирующая соответствующий алгоритмический процесс в абстрактном виде.

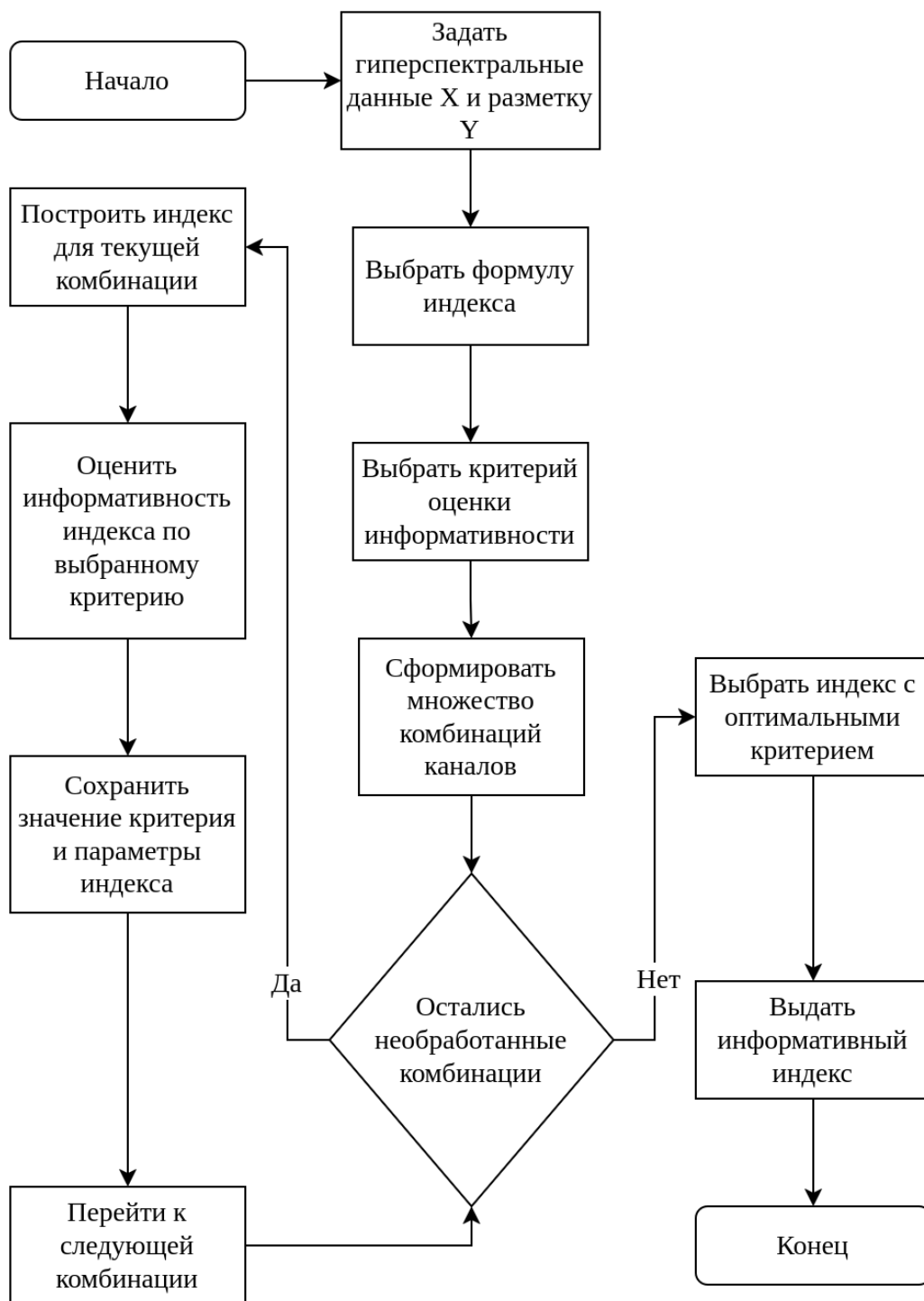


Рисунок 3.1 – Методика формирования информативного индекса

С вычислительной точки зрения основным недостатком данной технологии является использование полного перебора, поскольку при увеличении числа аргументов формулы индекса количество проверяемых комбинаций экспоненциально возрастает. Однако такая организация процесса обладает важным преимуществом: каждая комбинация слоев анализируется

независимо от остальных, что делает возможной эффективную параллельную реализацию. Благодаря этому предложенная методика может быть использована как практическая основа автоматизированного поиска информативных индексов в составе интеллектуальной системы обработки гиперспектральных изображений.

3.2 Алгоритм формирования информативного нормализованного разностного индекса (INDI)

Алгоритм INDI предназначен для решения бинарной задачи классификации объектов на ГСИ. В данной постановке считается, что разметка задает два класса: целевой класс и альтернативный класс. Задача алгоритма состоит в поиске такого нормализованного разностного индекса, который обеспечивает наилучшее разделение этих двух классов.

Пусть для каждого пикселя $u \in \Omega$ задан спектральный вектор $x(u) \in \mathbb{R}^B$, а разметка определяется отображением:

$$Y : \Omega \rightarrow \{c_1, c_2\}.$$

Для пары спектральных каналов (p, q) , вводится нормализованный разностный индекс $I_{p,q}$, по формуле:

$$I_{p,q} = \frac{x_p - x_q}{x_p + x_q}, \quad p, q \in \{1, \dots, B\}, p \neq q.$$

Данный индекс был выбран благодаря его устойчивости к неравномерности яркости и распространенности применения (популярные индексы NDVI, NDWI имеют аналогичную формулу). *Разница* слоев в данной формуле позволяет оценить различия в интенсивности отраженного света. Устойчивость к неравномерной яркости обеспечивается *нормализацией* области значений к диапазону $[-1; 1]$ за счет деления разницы слоев на их сумму. Применение данного индекса к изображению X формирует вектор или матрицу значений индекса на всем множестве пикселей.

Для количественной оценки разделимости классов вводятся матрицы внутригрупповой и межгрупповой дисперсии. Поскольку в алгоритме INDI

для фиксированной пары каналов рассматривается один индекс, соответствующее признаковое пространство является одномерным.

Пусть U_1 и U_2 – множества пикселей, принадлежащих соответственно классам c_1 и c_2 , а $z_u = I_{p,q}(x(u))$ – значение индекса для пикселя u . Тогда для каждого класса определяются выборочные средние:

$$\mu_1 = \frac{1}{|U_1|} \sum_{u \in U_1} z_u, \quad \mu_2 = \frac{1}{|U_2|} \sum_{u \in U_2} z_u,$$

и также общее среднее

$$\mu = \frac{1}{|U_1| + |U_2|} \left(\sum_{u \in U_1} z_u + \sum_{u \in U_2} z_u \right).$$

Внутригрупповая дисперсия характеризует разброс значений индекса каждого класса и в бинарном случае определяется следующим образом:

$$S_w = \sum_{k=1}^2 \sum_{u \in U_k} (z_u - \mu_k)(z_u - \mu_k)^T.$$

Межгрупповая дисперсии характеризует различие между классами по их средним значения и определяется:

$$S_b = \sum_{k=1}^2 |U_k| (\mu_k - \mu)(\mu_k - \mu)^T.$$

Задача поиска информативного индекса сводится к поиску такой пары каналов (p, q) , для которой *межгрупповая дисперсия максимальна, а внутригрупповая минимальна*. Иными словами, требуется найти индекс, обеспечивающий максимальное различие между классами при минимальном разбросе значений внутри каждого класса. Для этого вводится критерий информативности:

$$J(p, q) = \frac{S_b(p, q)}{S_w(p, q) + \epsilon}.$$

Тогда задача поиска индекса записывается как

$$(p^*, q^*) = \arg \max_{p \neq q} J(p, q),$$

а искомый информативный индекс определяется как:

$$I^*(x) = I_{p^*, q^*}(x).$$

Таким образом, алгоритм INDI реализует отображение

$$(X, Y) \rightarrow I^* \rightarrow Z^*,$$

где Z^* – матрица значений найденного информативного индекса на изображении. При дальнейшем переходе к решению задачи бинарной классификации – по данной матрице необходимо сформировать бинарную маску $M : \Omega \rightarrow \{0, 1\}$ с использованием отдельного оператора принятия решений. В таком случае решение задачи бинарной классификации пикселей принимает вид:

$$(X, Y) \rightarrow I^* \rightarrow Z^* \rightarrow M.$$

С вычислительной точки зрения важным свойством алгоритма INDI является то, что все его этапы могут быть реализованы в векторно-матричной форме. Для каждой пары каналов значения индекса вычисляются сразу для всех пикселей как векторная операция. Далее по соответствующим подмножествам значений формируются векторы классовых наблюдений, на основе которых вычисляются выборочные средние, а затем матрицы внутригрупповой и межгрупповой дисперсии. Следовательно, оценка каждого кандидата на роль информативного индекса сводится к последовательности стандартных операций над векторами и матрицами. Такая организация вычислений обеспечивает высокую эффективность реализации, позволяет использовать оптимизированные библиотеки линейной алгебры и делает полный перебор допустимых пар каналов практически осуществимым, в том числе при распараллеливании вычислений. Сложность полного перебора при этом составляет $O(n^2)$.

На рисунке 3.2 разработанный алгоритм представлен в виде блок-схемы.

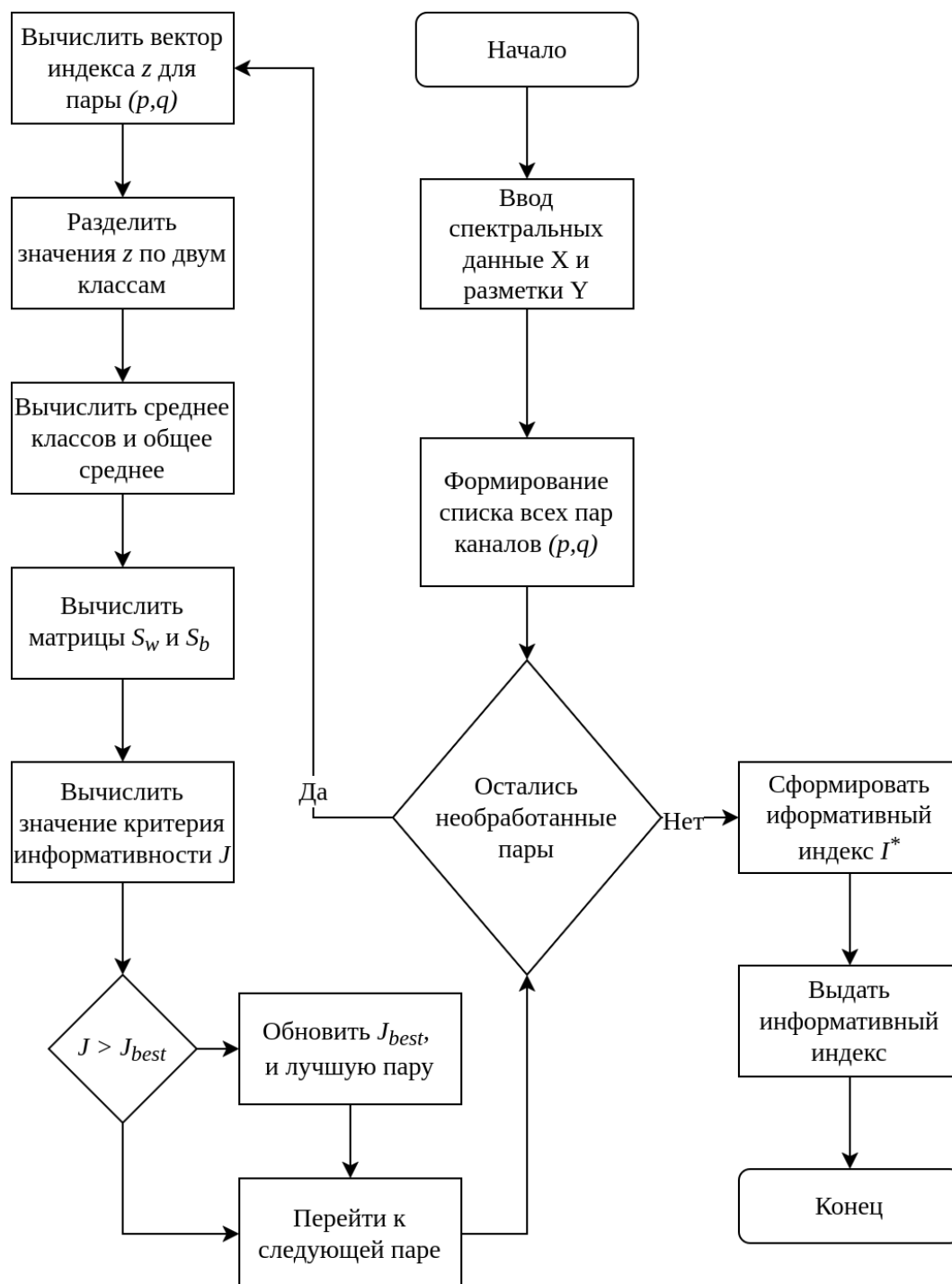


Рисунок 3.2 – Диаграмма разработанного алгоритма формирования информативного индекса INDI

3.2.1 Исследование применимости разработанного алгоритма INDI

На рисунке 3.3 представлен пример использования разработанного алгоритма формирования информативного индекса для различения двух классов объектов: *stubble* (стерня, или засохшие растения) и *celery* (сельдерей) на ГСИ "Salinas Scene" из набора данных "Hyperspectral Remote Sensing

Scenes". Исходное ГСИ в трехканальном представлении изображено на рисунке 3.3(а). На ГСИ, были выбраны две области, соответствующие классам стерни (область, выделенная красным цветом) и сельдерей (область, выделенная зеленым цветом). Разметка классов изображена на рисунке 3.3(б). На рисунках 3.3(в) и 3.3(г) изображены найденный индекс INDI (которому соответствуют информативные слои с длинами волн 668нм и 745нм) и индекс NDVI (которому соответствуют слои с длинами волн 1097нм и 707нм) соответственно.

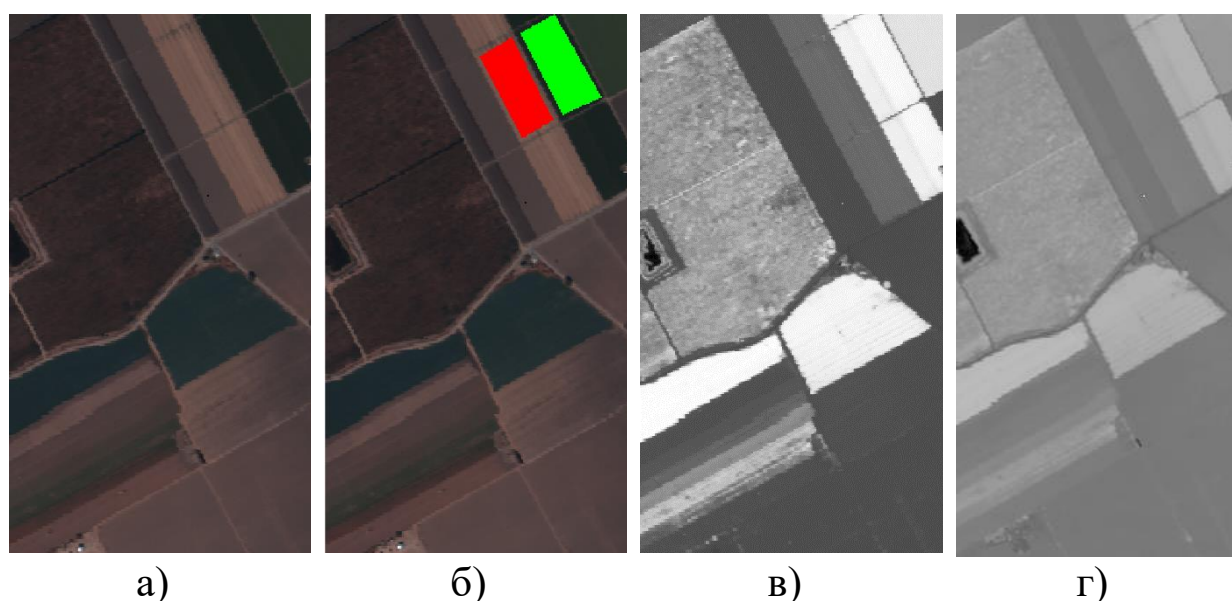


Рисунок 3.3 – Исходные данные и визуализация результатов работы алгоритма: а) визуализация исходного изображения; б) маска классов; в) визуализация INDI; г) визуализация NDVI

Сравнение полученного индекса INDI с индексом NDVI представляет интерес, так-как индекс NDVI является одним из лучших вариантов для оценки фотосинтетической активности растений и, следовательно, должен эффективно разделять классы сельдерей и стерня. На рисунке 3.4 представлена диаграмма распределений значений индексов INDI и NDVI в областях соответствующих маскам классов. Как видно на гистограмме, значения для двух разных классов хорошо разделимы на обоих индексах. В случае INDI это возможно благодаря применению критерия информативности для оценки разделимости классов. Произведем пороговую обработку обоих индексов с

целью формирования бинарных масок для классов стерни и сельдерей. Для этого использовались 20-ый и 80-ый перцентили. Данный способ был выбран эмпирически для демонстрационных целей.

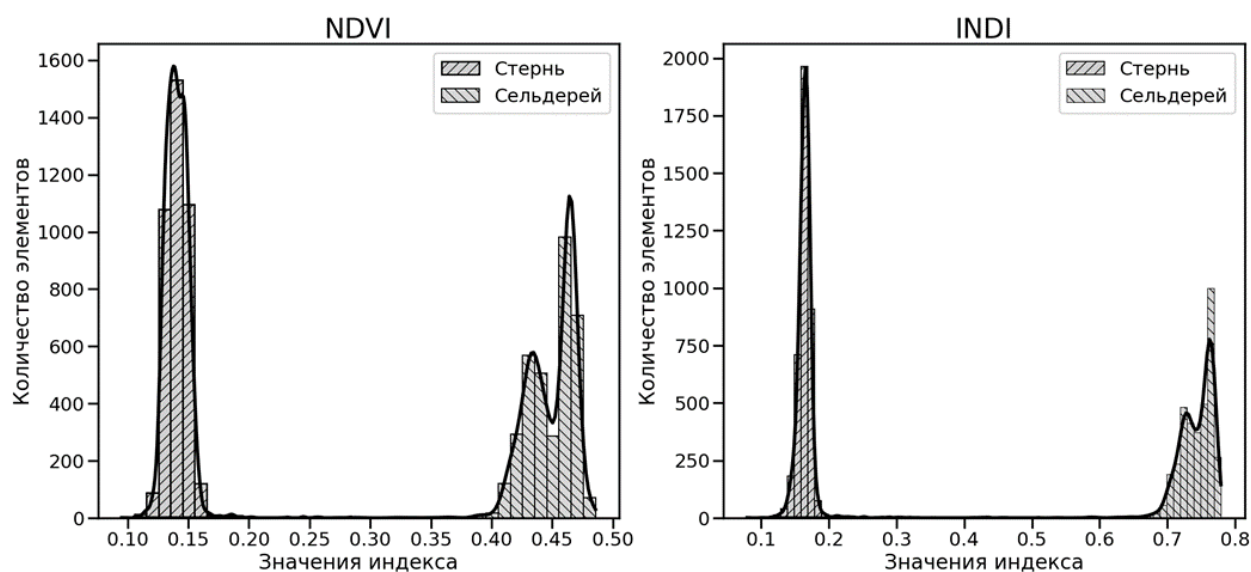


Рисунок 3.4 – Распределение значений индексов INDI и NDVI

Сформированные бинарные маски без постобработки, а также маски классов стерни и сельдерей показаны на рисунках 3.5 и 3.6.

Сравнение индексов для класса "Стернь"

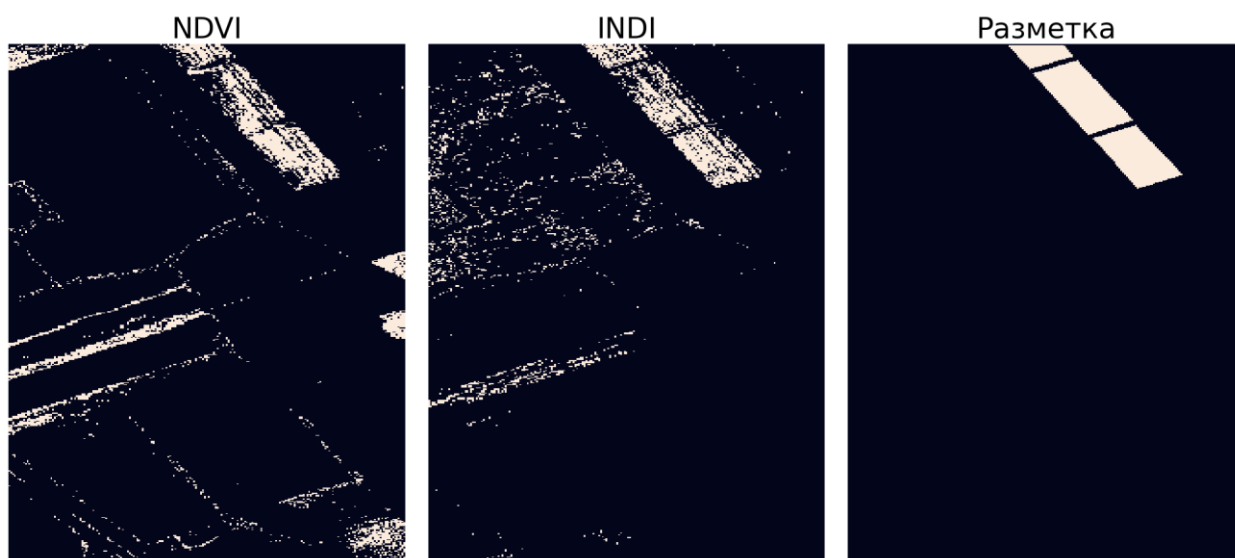


Рисунок 3.5 – Сравнение сформированных масок из индексов INDI и NDVI для класса “Стерня”

Сравнение индексов для класса "Сельдерей"

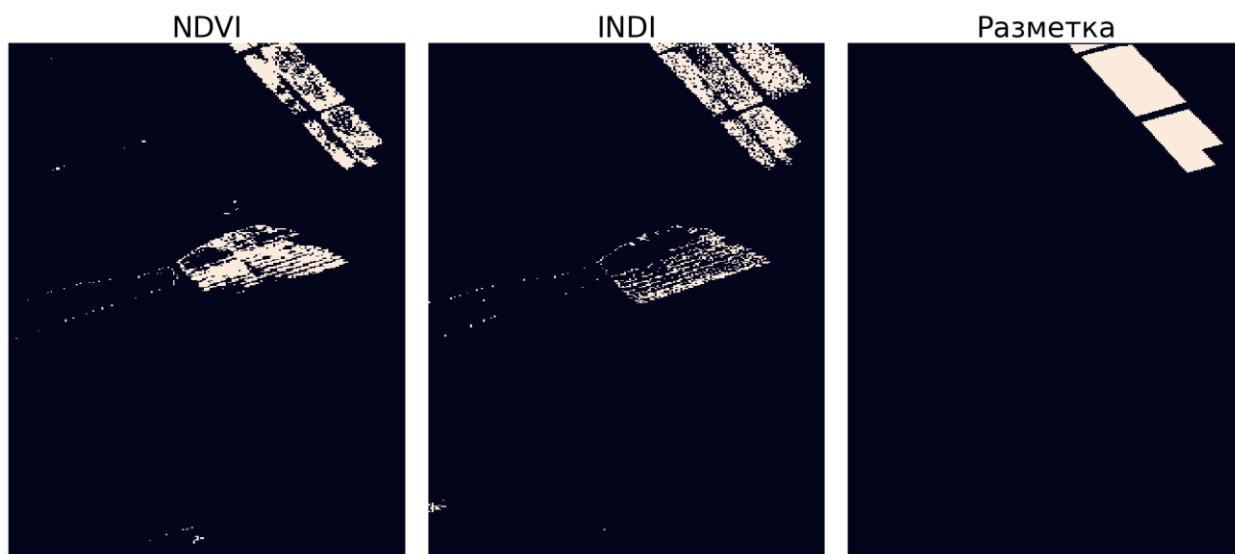


Рисунок 3.6 – Сравнение сформированных масок из индексов INDI и NDVI для класса “Сельдерей”

По данным изображениям видно, что полученные после пороговой обработки маски требуют дополнительной постобработки. Для выбора наилучшей стратегии постобработки масок был проведен сравнительный эксперимент. В рамках данного эксперимента маски были обработаны одним из следующих способов:

- медианный фильтр с ядром размера 5x5;
- медианный фильтр с ядром размера 3x3;
- медианный фильтр с последующим применением морфологических операций открытия и дилатации с размерами ядер 3x3;
- медианный фильтр с последующим применением морфологических операций открытия и двумя итерациями дилатации с размерами ядер 3x3;
- медианный фильтр с последующим применением морфологических операций открытия и тремя итерациями дилатации с размерами ядер 3x3;
- медианный фильтр с последующим применением морфологических операций открытия и дилатации с размерами ядер 5x5;
- медианный фильтр с последующим применением морфологических операций открытия и двумя итерациями дилатации с размерами ядер 5x5;

– медианный фильтр с последующим применением морфологических операций открытия и тремя итерациями дилатации с размерами ядер 5×5 .

Использование медианного фильтра в этих стратегиях позволяет устранить мелкий шум в масках. Морфологическая операция дилатации расширяет положительные области, заполняя пустоты, а операция закрытия, подобно дилатации, заполняет пустоты внутри положительных областей. Операция открытия, наоборот, увеличивает расстояние между областями, удаляя мелкие объекты и шум.

Для оценки точности сегментации определим метрики точности, полноты и F_β -меры по следующим формулам:

– precision (точность), способность модели отличать один класс от другого:

$$Precision = \frac{TP}{TP + FP};$$

– recall (полнота), способность модели обнаруживать класс:

$$Recall = \frac{TP}{TP + FN};$$

– F_β характеризует баланс между способностью модели правильно выделять класс и способностью находить все объекты этого класса:

$$F_\beta = \frac{(1 + \beta^2) \cdot precision \cdot recall}{(\beta^2 \cdot precision) + recall}.$$

Для оценки качества алгоритма в работе используется преимущественно метрика F_β , представляющая собой взвешенное среднее точности и полноты. В отличие от метрики F_1 , одинаково учитывающей точность и полноту, используемая метрика позволяет управлять относительной значимостью ошибок первого и второго рода за счёт параметра β . При $\beta > 1$ больший вес получает полнота, что соответствует ситуациям, в которых пропуск пикселей целевого класса является более нежелательным, чем включение в маску отдельных ложноположительных пикселей.

В настоящей работе используется значение $\beta = 2$, поскольку при анализе гиперспектральных изображений и формировании карт сегментации важным является максимально полное выделение целевых областей, а точное определение классов затруднено методами формирования маски классов. Пропущенные пиксели целевого класса не участвуют в последующих этапах анализа, формирования индексов, построения предварительной разметки и обучения моделей машинного обучения. В то же время часть ложноположительных пикселей может быть устранена на этапах постобработки, фильтрации и морфологической обработки. Таким образом, использование метрики F_2 позволяет учитывать приоритет полноты выделения объектов при сохранении зависимости от точности результата

Для оценки применимости алгоритма INDI была использована постановка "один против остальных". В рамках этой постановки для каждого класса исходной разметки изображения Salinas Valley формировалась бинарная задача, в которой целевой класс противопоставлялся объединению всех остальных классов. Для каждой такой задачи алгоритм INDI выполнял поиск информативного индекса, после чего по значениям найденного индекса строилась бинарная маска целевого класса.

Дополнительно исследовалось влияние постобработки бинарных масок на качество сегментации. Для этого сравнивались результаты без постобработки, после медианной фильтрации и после комбинации медианного фильтра с морфологическими операциями. Результаты представлены в таблице 3.1.

Таблица 3.1 – Сравнительный эксперимент влияния постобработки бинарных масок на качество сегментации классов

<i>Метод пост-обработки</i>	<i>F_2</i>	<i>Precision</i>	<i>Recall</i>
Без пост-обработки	0,465	0,381	0,601
Медианный фильтр (МФ) 3x3	0,536	0,459	0,664
Медианный фильтр 5x5	0,572	0,497	0,695
МФ-Открытие-Дилатация 3x3 1x	0,642	0,383	0,946
МФ-Открытие-Дилатация 3x3 2x	0,648	0,422	0,902
МФ-Открытие-Дилатация 3x3 3x	0,642	0,383	0,946
МФ-Открытие-Дилатация 5x5 1x	0,643	0,473	0,810
МФ-Открытие-Дилатация 5x5 2x	0,651	0,401	0,903
МФ-Открытие-Дилатация 5x5 3x	0,629	0,346	0,945

Полученные результаты демонстрируют, что применение метода совмещающего медианный фильтр и морфологические преобразования, позволяет улучшить совпадение маски с исходной разметкой на 40% по метрике F_2 .

В таблице 3.2 приведены результаты применения алгоритма INDI для всех задач "один против остальных" на изображении Salinas Valley с использованием выбранного способа постобработки.

Таблица 3.2 – Результаты попиксельной классификации с помощью информативного индекса INDI

<i>Класс</i>	<i>INDI</i>	<i>F_2</i>	<i>Precision</i>	<i>Recall</i>
Брокколи сорт 1	(495.8, 563.7)	0,92	0,72	1,00
Брокколи сорт 2	(495.8, 563.7)	0,91	0,79	0,95
Вспаханное поле	(408.5, 2430.0)	0,45	0,14	1,00
Вспаханное плугом	(389.1, 878.5)	0,76	0,51	0,88
Вспаханное гладкое	(973.5, 1087.5)	0,64	0,37	0,78
Стерня	(573.4, 1740.0)	0,91	0,73	0,97
Сельдерей	(2040.0, 2310.0)	0,80	0,58	0,89
Необработанный виноград	(389.1, 878.5)	0,79	0,53	0,90
Почва виноградников	(954.5, 1021.0)	0,63	0,26	1,00
Кукуруза	(398.8, 1670.0)	0,29	0,08	0,86
Листовой салат №1	(457.0, 573.4)	0,24	0,06	0,90
Листовой салат №2	(495.8, 2220.0)	0,67	0,31	0,94
Листовой салат №3	(447.3, 524.9)	0,55	0,22	0,87
Листовой салат №4	(764.5, 1087.5)	0,31	0,09	0,74
Виноградники	(1500.0, 2250.0)	0,75	0,42	0,92
Виноградники с опорами	(755.0, 973.5)	0,79	0,61	0,85
<i>Среднее значение:</i>		0,65	0,48	0,81

Полученные результаты показывают, что алгоритм INDI способен формировать информативные индексы для выделения различных классов

объектов на ГСИ. Среднее значение метрики F_2 по всем классам составило 0,65, средняя точность – 0,48, средняя полнота – 0,81. Высокое значение полноты подтверждает, что найденные индексы позволяют в большинстве случаев выделять значительную часть пикселей целевого класса.

Для ряда классов алгоритм продемонстрировал высокое качество классификации. Например, для классов "Брокколи сорт 1", "Брокколи сорт 2", "Стерня", "Сельдерей", "Необработанный виноград" и "Виноградники с опорами". Это свидетельствует о том, что для данных классов существуют пары спектральных каналов, на основе которых можно построить информативные нормализованные разностные индексы, обеспечивающие устойчивое отделение целевого класса от остальных.

В то же время для некоторых классов, например "Кукуруза", "Листовой салат №1" и "Листовой салат №4", значения F_2 оказались ниже. Это может быть связано со спектральной близостью указанных классов к другим объектам изображения, неоднородностью самих классов, а также с ограничениями одномерного индексного представления. Один нормализованный разностный индекс не всегда способен полностью отделить целевой класс от совокупности всех остальных классов, особенно если множество остальных классов является спектрально неоднородным.

Отдельно следует отметить соотношение точности и полноты. Средняя полнота составила 0,81, тогда как средняя точность – 0,48. Это говорит о том, что алгоритм INDI в рассматриваемой настройке склонен формировать маски с достаточно высоким охватом целевого класса, но при этом может включать в них ложноположительные пиксели. Такой результат согласуется с выбранной метрикой F_2 , в которой приоритет отдан полноте выделения объектов. При необходимости повышения точности результат может быть дополнительно улучшен за счёт более строгих правил пороговой обработки.

Для дополнительной проверки эффективности автоматически сформированного индекса было выполнено сравнение INDI с известным

индексом NDVI на задаче разделения классов "Стерня" и "Сельдерей", соответствующих сухой и живой растительности. Результаты представлены в таблице 3.3.

Таблица 3.3 – Точность разделения классов “сельдерей” и “стерня” индексами INDI и NDVI

<i>Индекс</i>	<i>F_2</i>	<i>Precision</i>	<i>Recall</i>
INDI (668 _{нм} и 745 _{нм})	0,79	0,66	0,93
NDVI (707 _{нм} и 1097 _{нм})	0,73	0,49	0,95

Согласно результатам представленным в таблице 3.3, информативный индекс, сформированный алгоритмом INDI, показал более высокое значение F_2 по сравнению с NDVI: 0,79 против 0,73.

При этом значение точности для INDI составило 0,66, что заметно выше значения 0,49, полученного для NDVI. Полнота у NDVI оказалась немного выше: 0,95 против 0,93 у INDI. Таким образом, NDVI несколько лучше охватывает целевой класс, однако делает это за счёт большего числа ложноположительных пикселей. В свою очередь, INDI обеспечивает более сбалансированный результат, повышая точность при сохранении высокой полноты.

Данный результат показывает, что автоматически сформированный индекс может быть эффективнее классического индекса, выбранного заранее. Это особенно важно для задач, в которых отсутствует заранее известный спектральный индекс, оптимальный для разделения требуемых объектов. Алгоритм INDI позволяет подобрать информативные спектральные каналы непосредственно по данным и разметке, что делает его более гибким инструментом анализа по сравнению с использованием фиксированных известных индексов.

На рисунках 3.7 и 3.8 представлены примеры бинарных масок, сформированных по информативным индексам после применения выбранного метода постобработки.

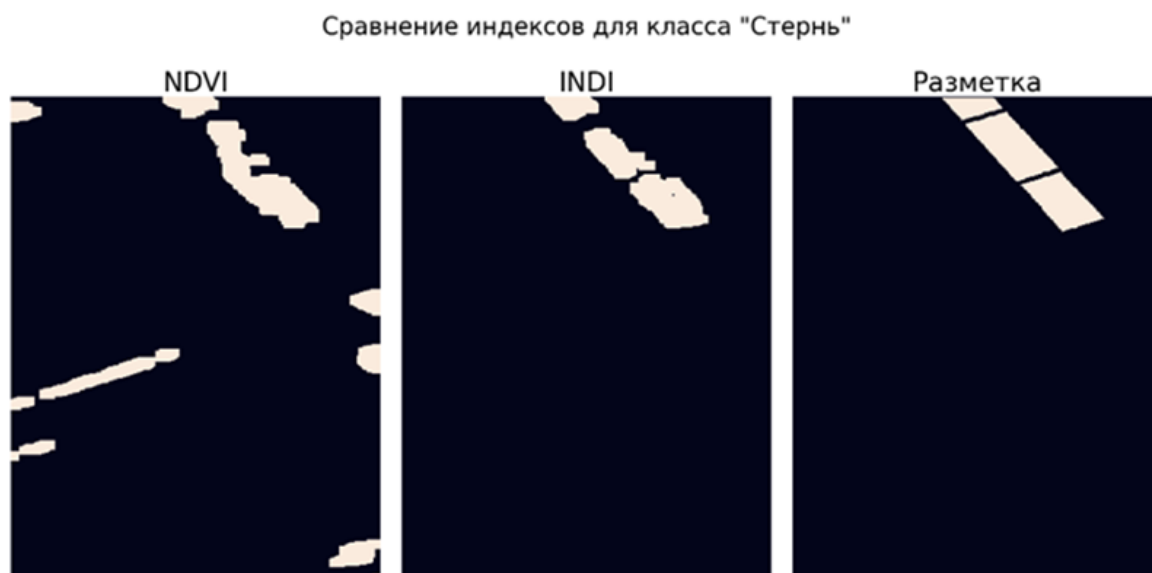


Рисунок 3.7 – Бинарные маски сформированные по индексам после пост-обработки для класса “стерня”

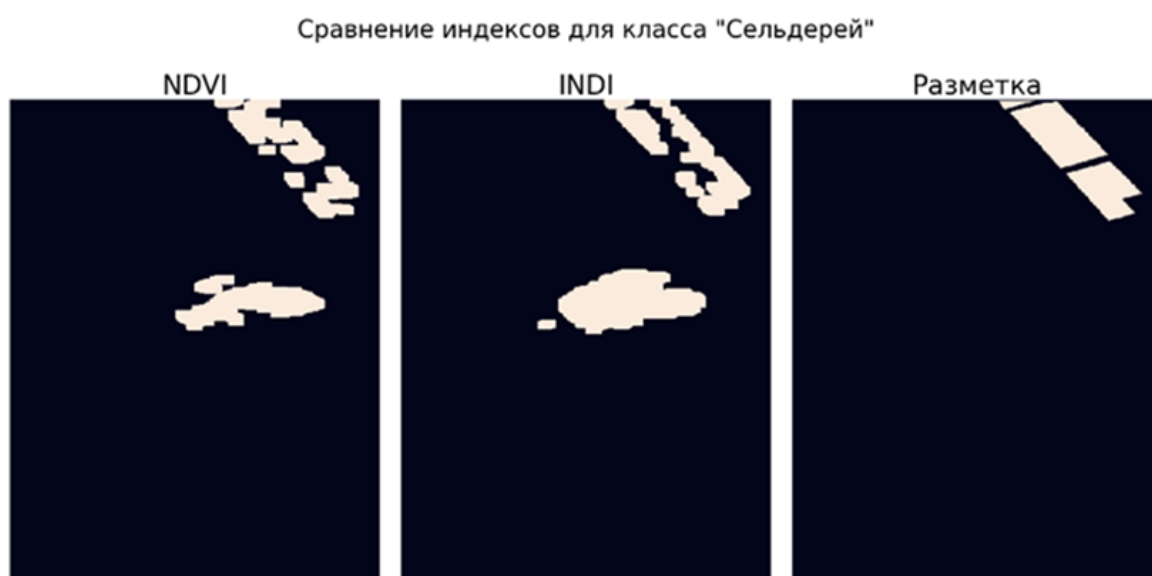


Рисунок 3.8 – Бинарные маски сформированные по индексам после пост-обработки для класса “сельдерей”

Для оценки применимости алгоритма INDI проведён эксперимент по формированию информативного спектрального индекса для обнаружения разливов нефти на гиперспектральном изображении. В качестве исходных данных использован набор Hyperspectral Remote Sensing Oil Spill Benchmark Database (HOSD) [131], включающий 18 гиперспектральных изображений различного пространственного разрешения, полученных с использованием

сенсора AVIRIS. Изображения содержат данные в 224 спектральных каналах, соответствующих диапазону длин волн от 350 до 2500 нм.

В рамках эксперимента использовано изображение *GM1* и соответствующая ему эталонная разметка, определяющая области нефтяного загрязнения. Визуальное представление исходного изображения и эталонной маски приведено на рисунке 3.9.

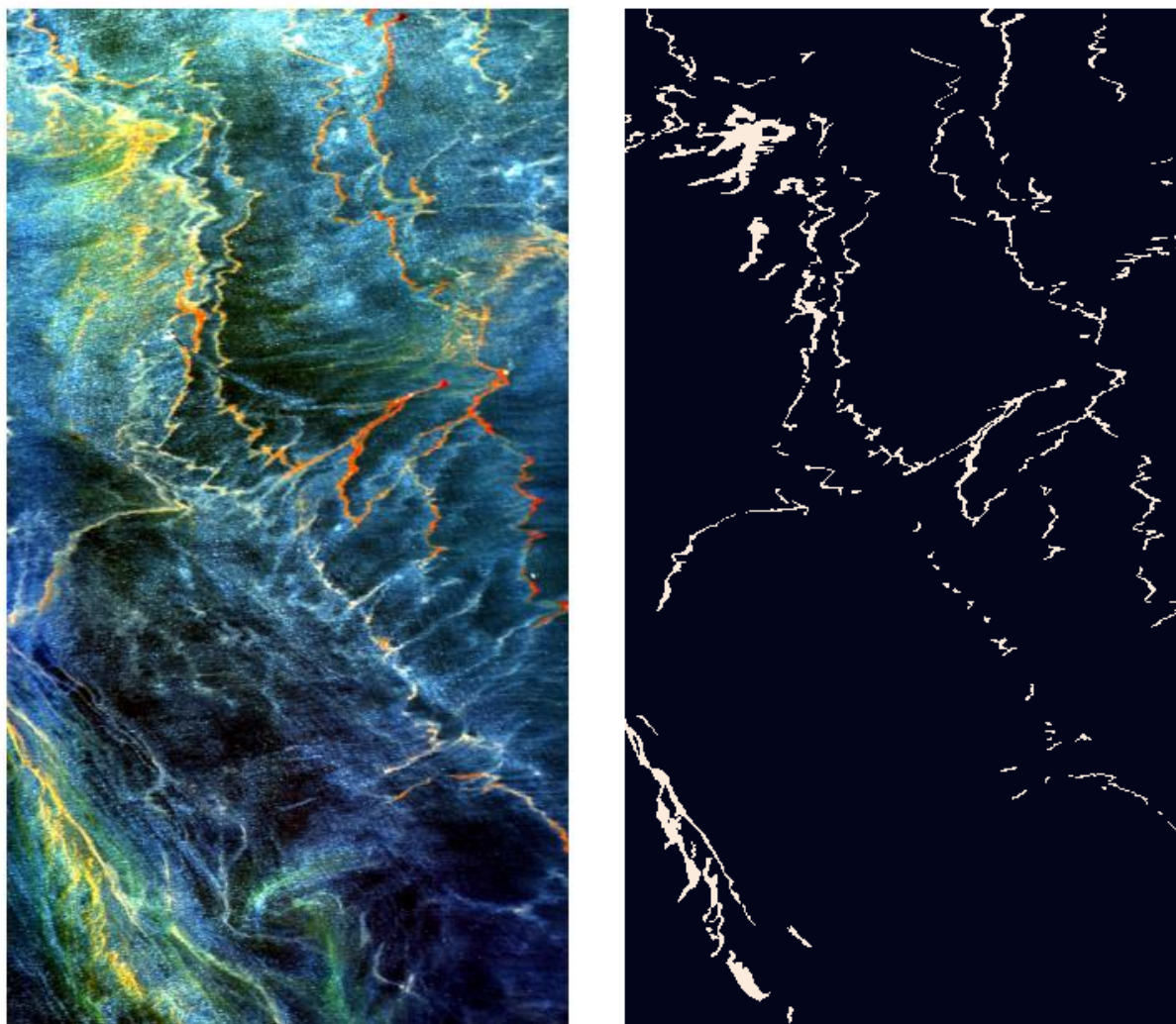


Рисунок 3.9 – Изображение *GM1* из набора данных HOSD и соответствующая ему разметка

На основе исходного гиперспектрального изображения и эталонной разметки с помощью алгоритма INDI была автоматически выбрана пара спектральных каналов (соответствующие длинам волн приблизительно 470нм. и 815нм.) и сформирован нормализованный разностный индекс. На рисунке 3.10 представлен полученный алгоритмом информативный индекс.

Для преобразования непрерывных значений индекса в бинарную маску использована пороговая обработка. Значение порога определялось по ROC-кривой с использованием индекса Юдена, позволяющего выбрать порог, при котором достигается наилучшее соотношение чувствительности и специфичности. ROC-кривая для сформированного информативного индекса представлена на рисунке 3.11.

Для полученной бинарной маски рассчитаны следующие значения метрик: $F_2 = 0,74$, $Precision = 0,38$, $Recall = 0,98$, $AUC = 0,99$.

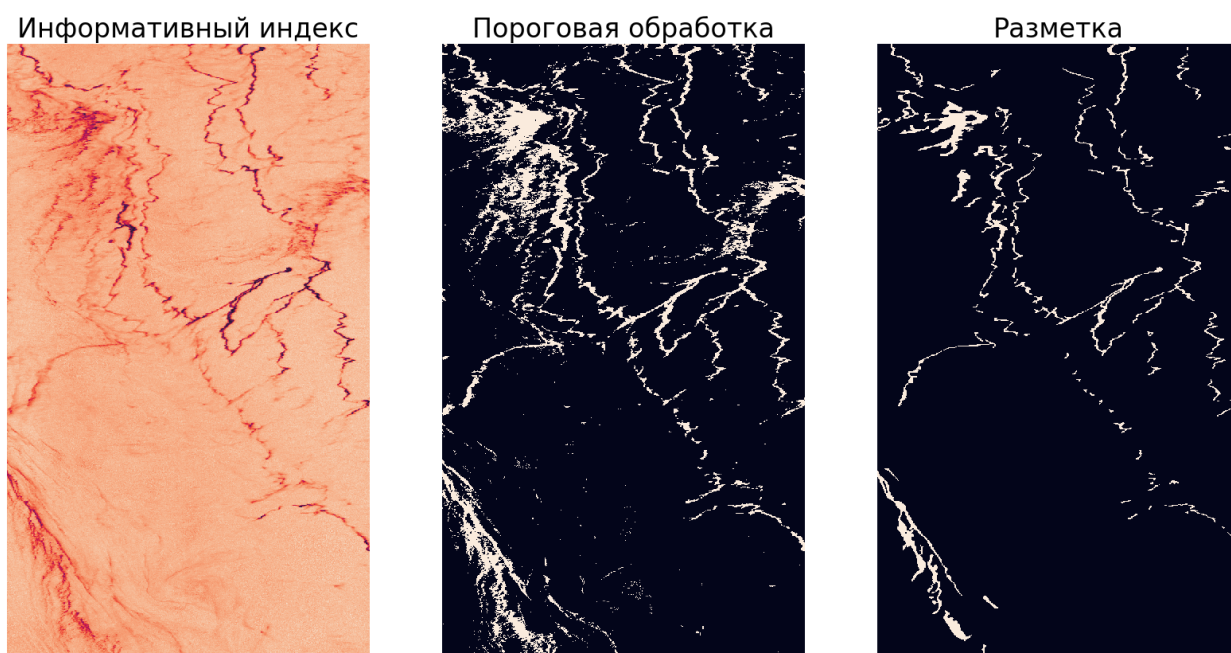


Рисунок 3.10 – Найденный информативный индекс и его сопоставление с исходной разметкой

Значение $recall$ показывает, что сформированный индекс позволил выделить практически все пиксели, отнесённые в эталонной разметке к нефтяному загрязнению. При этом значение $precision$ свидетельствует о наличии ложноположительных срабатываний, то есть часть фоновых пикселей также была отнесена к целевому классу. Полученное значение метрики F_2 отражает выбранный в работе приоритет полноты обнаружения целевых объектов над точностью их выделения.

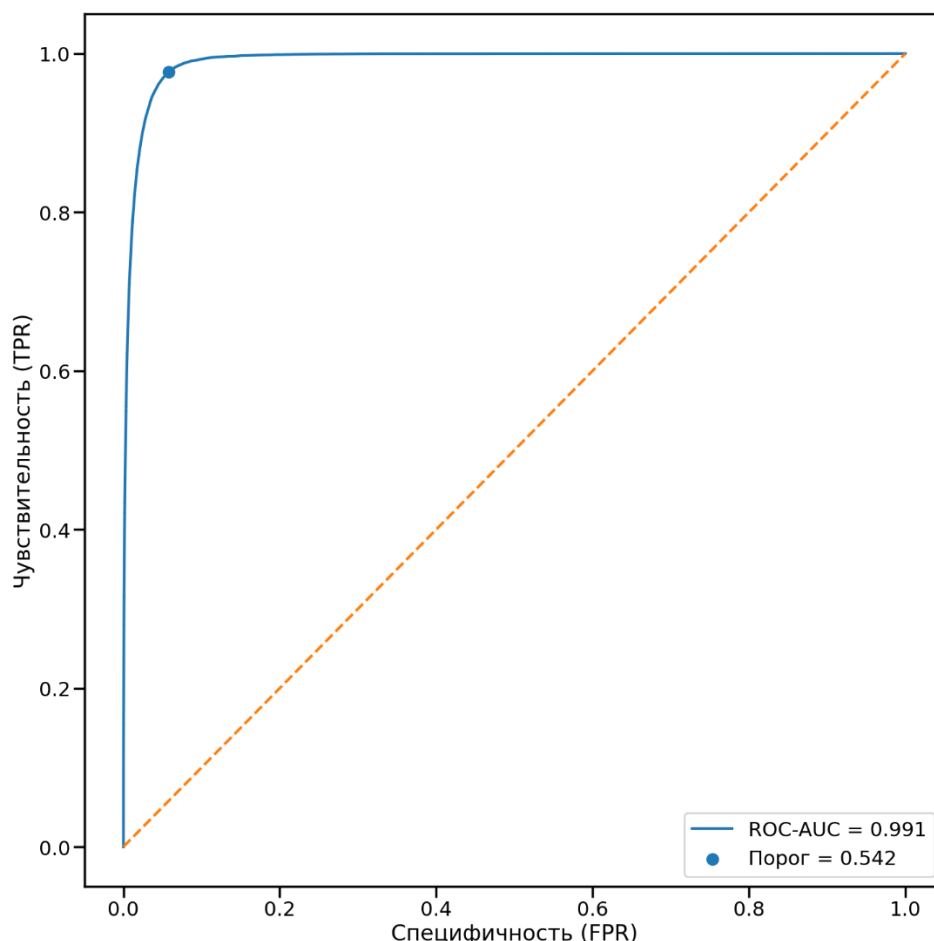


Рисунок 3.11 – ROC-кривая построенная для полученного информативного индекса

Высокое значение метрики AUC свидетельствует о выраженной способности сформированного индекса ранжировать пиксели нефтяного загрязнения выше фоновых пикселей в рассматриваемом изображении. Одновременно различие между высоким значением AUC и сравнительно низким значением $Precision$ показывает, что итоговое качество бинарной маски существенно зависит от выбранного порога и распределения классов на изображении.

Таким образом, проведённый эксперимент подтвердил применимость алгоритма INDI для автоматического формирования спектрального индекса, ориентированного на обнаружение нефтяных загрязнений.

Рассмотрим далее возможность применения алгоритма INDI для решения другой задачи – обнаружения очагов лесных пожаров по

спутниковым гиперспектральным данным. Для этого был проведен эксперимент на мультиспектральном изображении из набора данных *Active Fire Detection in Landsat-8 Imagery* [132]. Изображения данного набора имеют пространственное разрешение 256×256 пикселей и содержат данные в 10 спектральных каналах, соответствующих диапазону длин волн от 430 до 2290 нм. Визуальное представление исходных данных и эталонной маски приведено на рисунке 3.12.

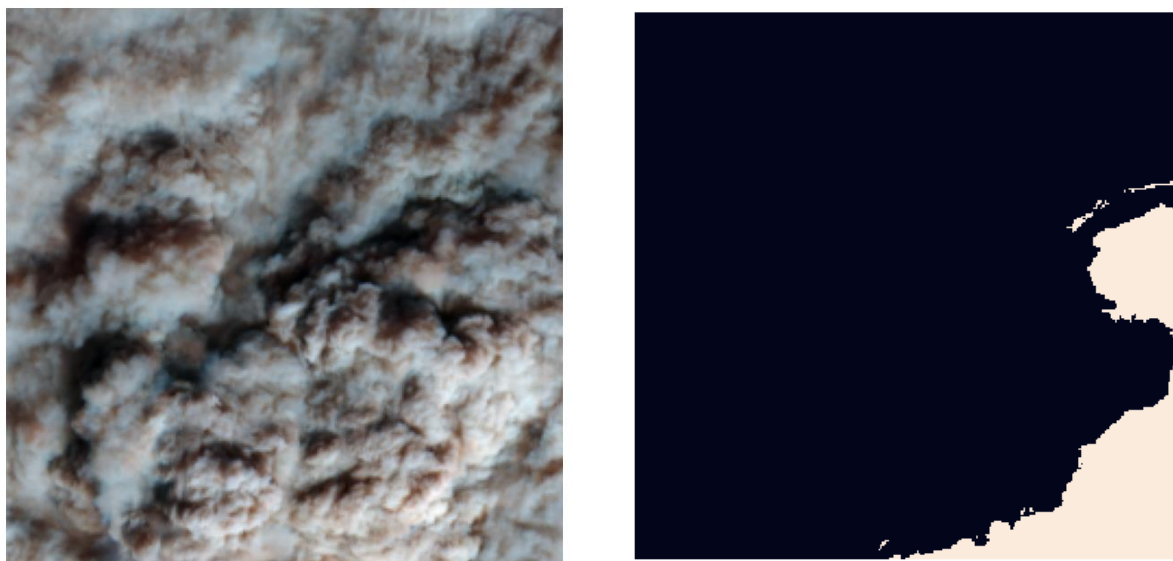


Рисунок 3.12 – Изображение из набора данных Active Fire и соответствующая ему разметка

На основе исходного изображения и эталонной разметки с помощью алгоритма INDI была автоматически выбрана пара спектральных каналов (соответствующие длинам волн приблизительно 2201нм. и 1373нм.) и сформирован нормализованный разностный индекс, ориентированный на разделение областей очагов возгорания и фоновых объектов. Полученный информативный индекс представлен на рисунке 3.13.

По аналогии с предыдущим экспериментом для преобразования индекса в бинарную маску была применена пороговая обработка. На рисунке 3.14 изображена построенная ROC-кривая. Для полученной бинарной маски рассчитаны следующие значения метрик: $F_2 = 0,94$, $Precision = 0,79$, $Recall = 0,99$, $AUC = 0,998$.

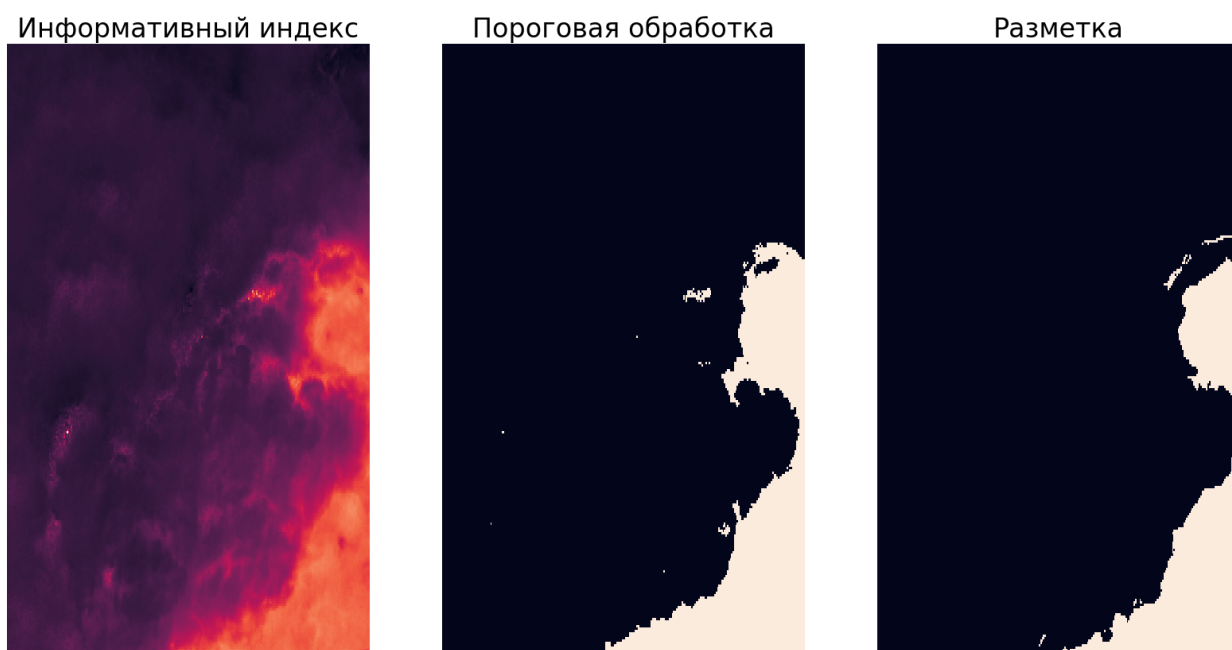


Рисунок 3.13 – Найденный информативный индекс и его сопоставление с исходной разметкой

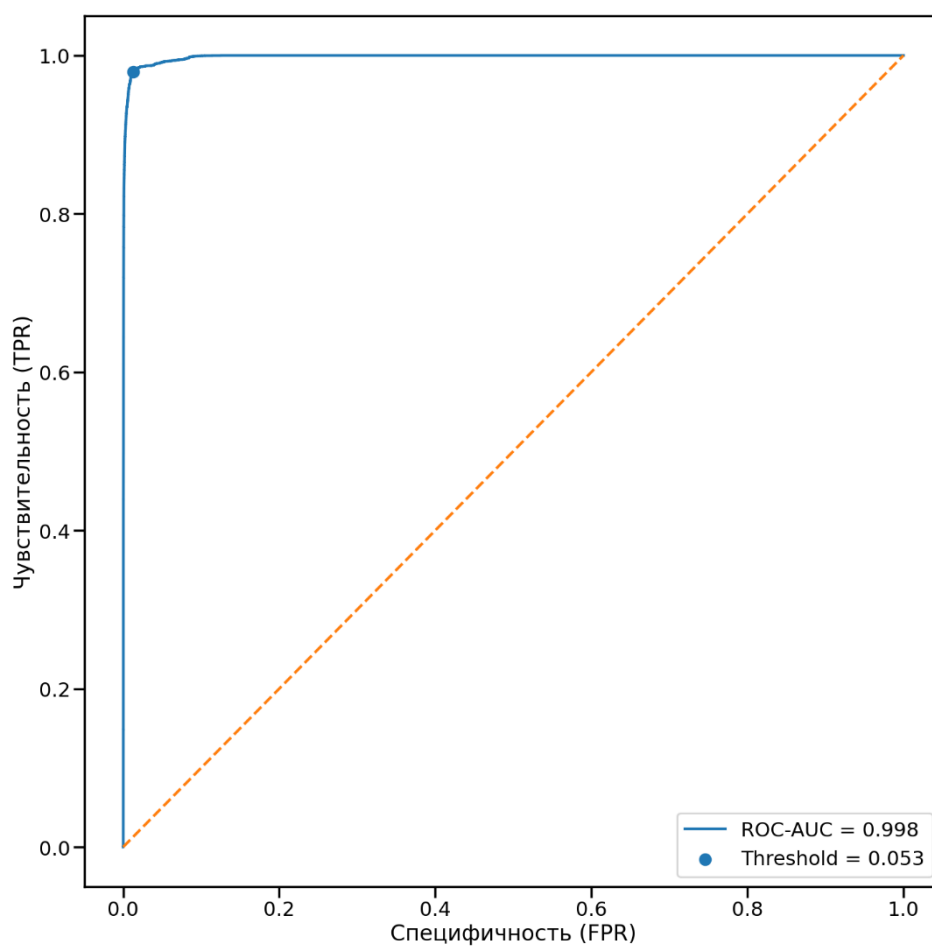


Рисунок 3.14 – ROC-кривая построенная для полученного информативного индекса

Значение *recall* показывает, что сформированный индекс позволил обнаружить практически все пиксели, относящиеся к положительному классу. Значение *precision* свидетельствует о сравнительно небольшом числе ложноположительных срабатываний. Высокое значение целевой метрики F_2 подтверждает качество выделения целевого класса с учётом принятого в работе приоритета полноты обнаружения. Значение *AUC* указывает на высокую способность сформированного спектрального признака разделять пиксели очагов пожара и фоновые области независимо от выбора конкретного порога бинаризации. В совокупности проведенная оценка показывает, что для рассматриваемого изображения сформированный индекс обладает выраженной разделяющей способностью и может использоваться для автоматизированного обнаружения активных очагов горения.

3.2.2 Результат разработки алгоритма формирования информативного индекса

Разработанный алгоритм INDI позволяет автоматически формировать информативный нормализованный разностный индекс для решения задач бинарного сегментации объектов на ГСИ. Алгоритм основан на полном переборе пар спектральных каналов и выборе такой пары, для которой значения индекса обеспечивают наилучшее разделение классов по заданному критерию информативности.

На примере изображения Salinas Valley показано, что INDI может применяться в постановке "один против остальных" для выделения различных классов объектов. Среднее значение F_2 по всем классам составило 0,65, а средняя полнота – 0,81, что подтверждает способность алгоритма формировать индексы, обеспечивающие высокий охват целевых областей. При этом качество результата зависит от спектральной отделимости конкретного класса и от степени неоднородности множества остальных классов.

Эксперимент с различными способами постобработки показал, что формирование бинарной маски по значениям индекса должно дополняться фильтрацией и морфологической обработкой. Наилучший способ постобработки позволил увеличить значение F_2 с 0,465 до 0,651, что соответствует относительному приросту примерно на 40%. Это подтверждает необходимость учитывать не только спектральную, но и пространственную структуру результата при построении масок классов.

Сравнение с классическим индексом NDVI на задаче разделения классов "Сельдерей" и "Стерня" показало, что индекс, сформированный алгоритмом INDI, обеспечивает более высокое значение метрик. Сформированный алгоритмом INDI индекс обеспечил увеличение F_2 с 0,73 до 0,79 и *precision* с 0,49 до 0,66 при незначительном уменьшении *recall* с 0,95 до 0,93. Полученные значения показывают, что автоматический выбор спектральных каналов для конкретной пользовательской задачи позволил сократить число ложноположительных срабатываний и улучшить итоговое качество сегментации по сравнению с использованием заранее заданного индекса NDVI. Это свидетельствует о том, что автоматический подбор спектральных каналов под конкретную задачу может давать более эффективный индекс, чем использование заранее заданного известного индекса.

Применимость алгоритма INDI для решения других задач подтверждена экспериментом по обнаружению нефтяных загрязнений на изображении GM1 из набора HOSE. Для индекса, автоматически сформированного алгоритмом, получены значения: $F_2 = 0,74$, *Precision* = 0,38, *Recall* = 0,98, *AUC* = 0,99. Высокие значения полноты и площади под ROC-кривой свидетельствуют о способности сформированного индекса обнаруживать практически все области нефтяного загрязнения и эффективно ранжировать пиксели целевого и фоновых классов. Вместе с тем значение точности указывает на наличие определенного числа ложноположительных срабатываний и необходимость

дополнительного уточнения результата посредством выбора порога и пространственной постобработки.

В эксперименте по обнаружению очагов лесных пожаров на мультиспектральном изображении из набора Active Fire Detection in Landsat-8 Imagery получены значения $F_2 = 0,94$, $Precision = 0,79$, $Recall = 0,99$, $AUC = 0,998$. Эти результаты показывают высокую полноту обнаружения целевых областей, сравнительно небольшое число ложноположительных срабатываний и выраженную разделяющую способность сформированного индексного признака. Эксперимент также подтвердил возможность применения алгоритма INDI не только к гиперспектральным, но и к мультиспектральным изображениям с меньшим числом спектральных каналов.

Таким образом, алгоритм INDI может быть использован для преодоления ограничения, связанного с отсутствием заранее известных спектральных индексов для решения произвольных задач анализа ГСИ. Он обеспечивает унифицированный способ построения индексных признаков по ГСИ и разметке, снижая необходимость ручного выбора каналов и ручного конструирования индексов. При этом результаты экспериментов показывают, что наиболее устойчивое применение INDI достигается в задачах, где целевой класс обладает выраженными спектральными отличиями от остальных объектов изображения.

По материалам данного раздела опубликованы работы [130]. Также получено государственное свидетельство о регистрации программы для ЭВМ [133, 134].

3.3 Алгоритм формирования ансамбля информативных индексов (MINDI)

Разработанный алгоритм формирования информативного индекса (INDI) в первую очередь предназначен для решения бинарных задач. Несмотря на демонстрацию возможности решения задач в постановке "один против

остальных" для классификации множества классов, данный подход не учитывает специфику многоклассовой задачи. Поэтому далее рассмотрим адаптацию алгоритма INDI для решения задачи многоклассовой попиксельной классификации, представив его в виде упорядоченной процедуры формирования ансамбля информативных индексов. Каждый индекс в этом ансамбле будет решать задачу бинарной классификации, разделяя один класс от всех остальных.

Основной принцип работы алгоритма MINDI заключается в жадном итеративном процессе, который включает следующие шаги:

- выбор наиболее отделимого класса от остальных с использованием *критерия отделимости*;
- нахождение информативного индекса с помощью алгоритма INDI для решения задачи разделения наиболее отделимого класса от остальных.

Пусть для каждого пикселя $u \in \Omega$ задан спектральный вектор $x(u) \in \mathbb{R}^B$, а разметка определяется отображением:

$$Y : \Omega \rightarrow \{c_1, c_2, \dots, c_K\}.$$

Для каждого класса $c_k \in C$ формируется подмножество пикселей

$$U_k = \{u \in \Omega \mid Y(u) = c_k\},$$

а также объединенное множество пикселей всех остальных классов, без классов обработанных на предыдущих итерациях.

$$U_{-k} = \{u \in \Omega \mid Y(u) \neq c_k, Y(u) \notin \{c_1, \dots, c_{k-1}\}\}.$$

Для каждого класса c_k на исходном этапе оценивается его интегральная отделимость от множества остальных классов. С этой целью по исходным спектральным векторам формируются матрицы внутригрупповой и межгрупповой дисперсии. Пусть

$$\mu_k = \frac{1}{|U_k|} \sum_{u \in U_k} x(u), \quad \mu_{-k} = \frac{1}{|U_{-k}|} \sum_{u \in U_{-k}} x(u),$$

а общее среднее определяется как

$$\mu = \frac{1}{|U_k| + |U_{\neg k}|} \left(\sum_{u \in U_k} x(u) + \sum_{u \in U_{\neg k}} x(u) \right).$$

Тогда матрица внутригрупповой дисперсии записывается в виде

$$S_w^{(k)} = \sum_{u \in U_k} (x(u) - \mu_k)(x(u) - \mu_k)^T + \sum_{u \in U_{\neg k}} (x(u) - \mu_{\neg k})(x(u) - \mu_{\neg k})^T,$$

А матрица межгрупповой дисперсии в виде

$$S_b^{(k)} = |U_k|(\mu_k - \mu)(\mu_k - \mu)^T + |U_{\neg k}|(\mu_{\neg k} - \mu)(\mu_{\neg k} - \mu)^T.$$

На основе этих матриц для каждого класса вычисляется *критерий отделимости*:

$$J_k = tr \left(\left(S_w^{(k)} \right)^{-1} S_b^{(k)} \right).$$

Нахождение обратной матрицы в данном критерии используется как многомерный аналог деления на внутриклассовый разброс, а след берётся для получения скалярной оценки общей разделимости классов по всем признакам. Однако, вычисление обратной матрицы является вычислительно неэффективным и численно менее устойчивым решением. Поэтому в реализации используется эквивалентная форма. Вместо поиска обратной матрицы решается матричное уравнение:

$$S_w^{(k)} X^{(k)} = S_b^{(k)},$$

После чего значение критерия определяется как

$$J_k = tr(X^{(k)}).$$

Такой способ вычисления позволяет сохранить математический смысл критерия и одновременно повысить эффективность реализации за счет использования существующих оптимизированных систем решения линейных уравнений.

Упорядочивание классов по отделимости при реализации алгоритма MINDI позволяет улучшить точность решения задачи классификации. Отделимым классом называется тот класс, кластер которого находится

наиболее удаленно от других кластеров. Пример отдельного класса показан на рисунке 3.15.

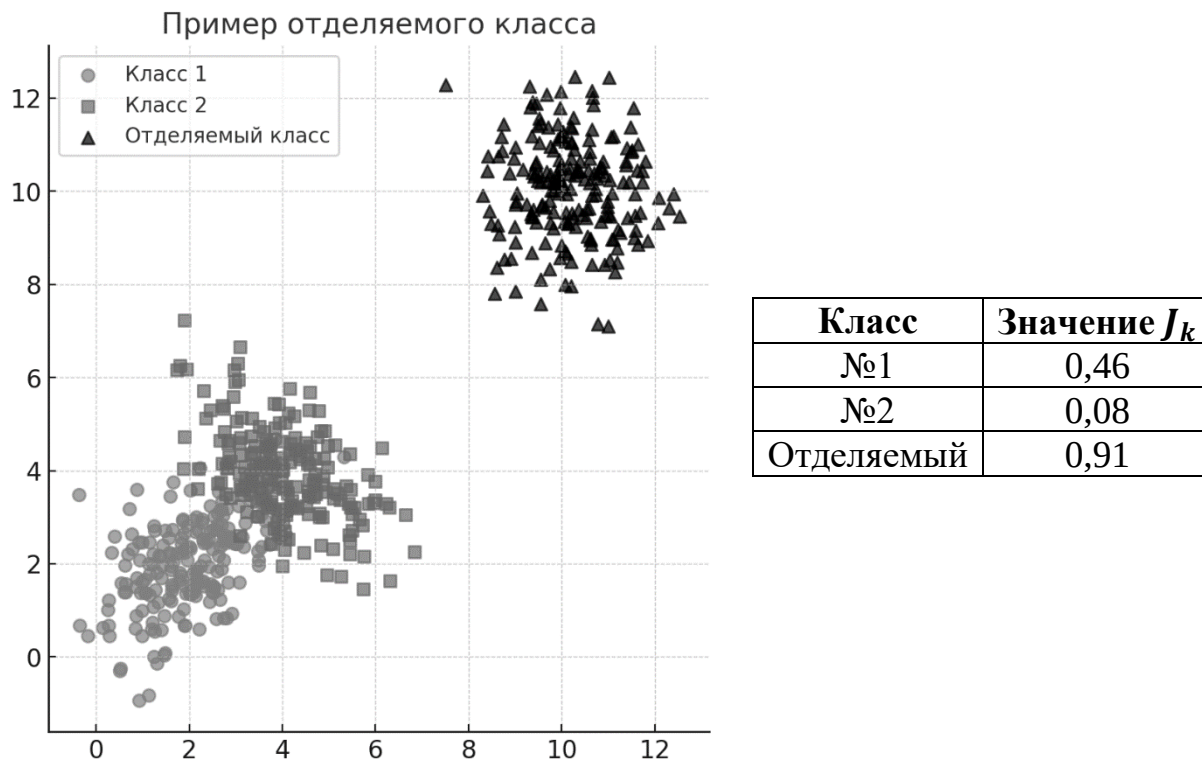


Рисунок 3.15 – Пример отдельного класса

Алгоритм MINDI является последовательным и жадным, выполняя вычисления для нахождения отдельного класса на каждой итерации формирования информативного признака. На каждой последующей итерации класс, отделенный на предыдущей итерации, исключается из рассмотрения. Таким образом, по мере нахождения отдельных классов вычислительная сложность поиска уменьшается. Выбор наиболее отдельного класса осуществляется по максимальному значению группового критерия разделимости.

Поиск информативного индекса в MINDI происходит аналогичным образом алгоритму INDI. В результате выполнения алгоритма MINDI формируется ансамбль информативных индексов:

$$I^* = \{I_1^*, I_2^*, \dots, I_K^*\},$$

где $I_k^*(x) = I_{p_k^*, q_k^*}(x)$ информативный индекс отделяющий c_k класс от остальных.

После построения ансамбля индексов каждый из них применяется к исходному изображению, в результате чего формируются матрицы

$$Z = \{Z_1, Z_2, \dots, Z_k\}, \quad Z_k(u) = I_k^*(x(u)).$$

Таким образом, каждому классу соответствует матрица значений информативного индекса. Для перехода от ансамбля матриц к итоговой карте сегментации для каждого класса c_k вычисляется оценка принадлежности пикселя u данному классу. Эта оценка задаётся выражением:

$$s_k(u) = \frac{(Z_k(u) - \mu_{\neg k})^2}{(\sigma_{\neg k})^2 + \epsilon} - \frac{(Z_k(u) - \mu_k)^2}{(\sigma_k)^2 + \epsilon}.$$

Используемая оценка отражает относительную близость значения индекса $Z_k(u)$ к распределению целевого класса и его удалённость от распределения множества остальных классов. Чем больше значение $s_k(u)$, тем выше степень согласованности пикселя u с классом c_k .

Тогда для каждого пикселя $u \in \Omega$ вычисляется набор оценок

$$s_1(u), s_2(u), \dots, s_K(u),$$

После чего определяется класс с максимальной оценкой

$$k^*(u) = \arg \max_{k \in C} s_k(u).$$

В результате чего можно сформировать карту сегментации, однако при данной постановке не задается фоновый класс. В связи с этим в алгоритме MINDI дополнительно вводится служебный класс 0, соответствующий области неопределенной классификации. Такой класс используется для пикселей, для которых решение о принадлежности одному из целевых классов нельзя считать достаточно надежным.

С этой целью для каждого пикселя вычисляется разность между наибольшей и второй по величине оценками:

$$\Delta(u) = s_{(1)}(u) - s_{(2)}(u),$$

где $s_{(1)}(u)$ и $s_{(2)}(u)$ – соответственно максимальная и вторая по величине оценки принадлежности среди всех классов. Тогда итоговая карта сегментации определяется отображением:

$$\hat{Y} : \Omega \rightarrow \{0\} \cup C.$$

По правилу

$$\hat{Y}(u) = \begin{cases} k^*(u), & \Delta(u) > \tau, \\ 0, & \Delta(u) \leq \tau, \end{cases}$$

где τ – порог уверенности классификации.

Таким образом, класс 0 представляет собой область отказа от классификации в случаях, когда различие между двумя наиболее вероятными классами оказывается недостаточным. Это позволяет снизить число ошибочных присваиваний меток и повысить точность итоговой карты сегментации.

Итоговая задача, решаемая алгоритмом MINDI, принимает следующий вид:

$$(X, Y) \rightarrow I^* \rightarrow Z \rightarrow \hat{Y}.$$

С вычислительной точки зрения алгоритм MINDI полностью организован в матрично-векторной форме. Исходные гиперспектральные данные рассматриваются как матрица наблюдений, метки классов – как вектор разметки, статистики классов – как векторы средних и матрицы рассеяния, а вычисление критерия отделимости, критериев информативности индексов и оценок принадлежности сводится к последовательности стандартных операций линейной алгебры. Это обеспечивает высокую эффективность реализации, возможность распараллеливания и масштабируемость алгоритма при увеличении числа спектральных каналов, числа классов и размера изображения.

На рисунке 3.16 приведена диаграмма, описывающая данный алгоритм в форме блок-схемы.

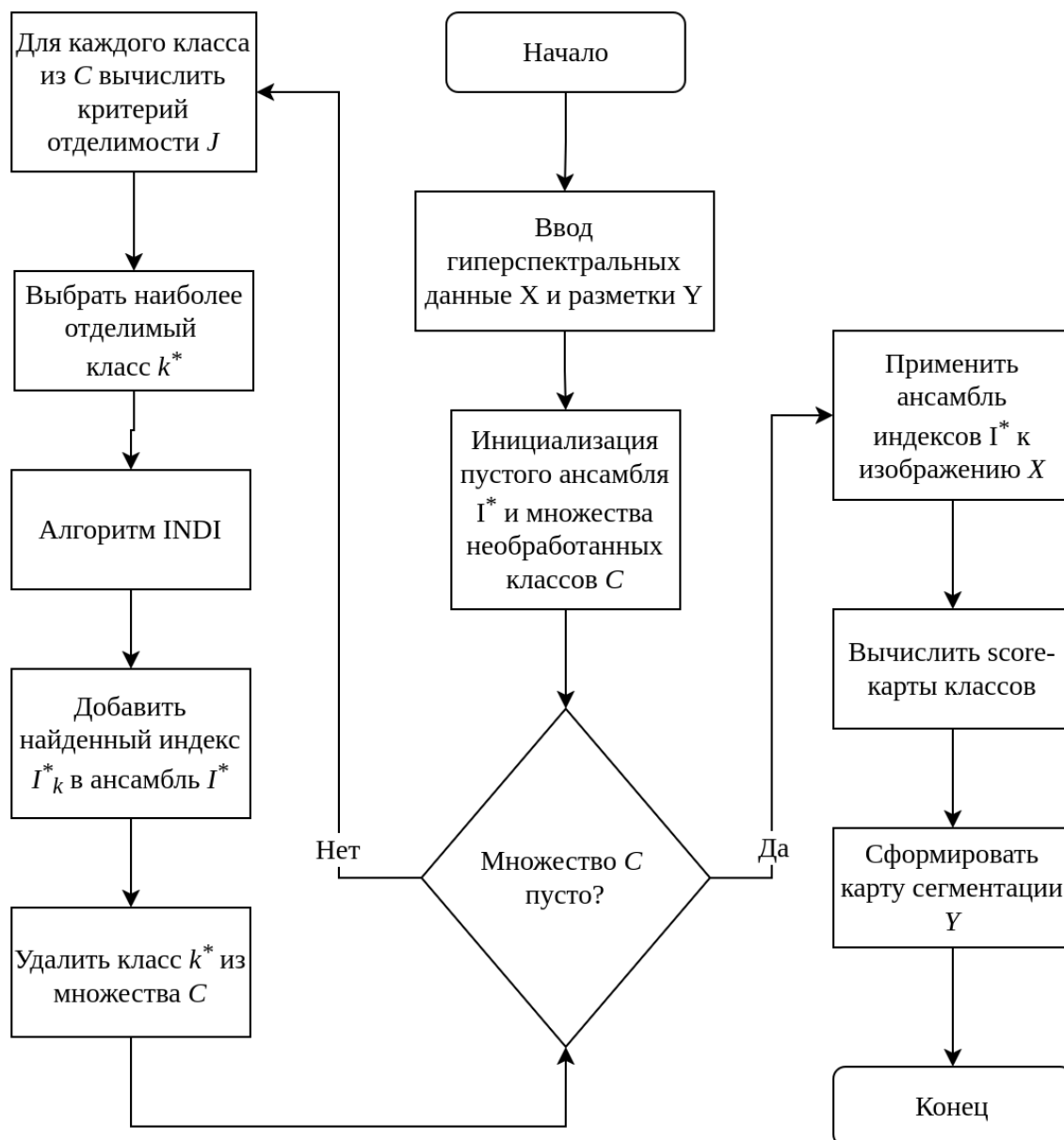


Рисунок 3.16 – Диаграмма разработанного алгоритма формирования ансамбля информативных индексов

3.3.1 Исследование применимости разработанного алгоритма

MINDI

В рамках исследования разработанного алгоритма MINDI необходимо оценить его применимость для решения задачи многоклассовой сегментации ГСИ. Для этого были поставлены следующие задачи:

- сравнить результат сегментации с исходной разметкой ГСИ;

– сравнить результаты, получаемые при независимом применении алгоритма INDI к каждому классу, с результатами алгоритма MINDI, учитывающего порядок выделения классов на основе критерия отделимости.

Для проведения эксперимента использовалось гиперспектральное изображение Salinas Scene. В таблице 3.4 представлены результаты сравнения алгоритмов INDI и MINDI при решении задачи многоклассовой сегментации.

Таблица 3.4 – Сравнительный эксперимент для оценки точности алгоритма INDI и MINDI

Класс	INDI			MINDI		
	F_2	Precision	Recall	F_2	Precision	Recall
Брокколи сорт 1	0,95	0,84	0,99	0,83	0,50	1,00
Брокколи сорт 2	0,51	0,71	0,48	0,55	0,83	0,51
Вспаханное поле	0,50	0,21	0,76	0,35	0,15	0,52
Вспаханное плугом	0,78	0,43	0,97	0,63	0,28	0,92
Вспаханное гладкое	0,74	0,48	0,86	0,53	0,61	0,52
Стернь	0,92	0,78	0,96	0,89	0,69	0,96
Сельдерей	0,82	0,48	1,00	0,89	0,62	1,00
Необработанный виноград	0,51	0,70	0,48	0,68	0,70	0,67
Почва виноградников	0,63	0,49	0,68	0,71	0,74	0,71
Кукуруза	0,57	0,29	0,75	0,30	0,11	0,56
Листовой салат №1	0,12	0,08	0,14	0,01	0,00	0,03
Листовой салат №2	0,81	0,48	0,97	0,18	0,12	0,21
Листовой салат №3	0,00	0,00	0,00	0,50	0,75	0,46
Листовой салат №4	0,00	0,00	0,00	0,84	0,66	0,91
Виноградники	0,43	0,52	0,41	0,63	0,61	0,64
Виноградники с опорами	0,46	0,24	0,61	0,35	0,29	0,37
<i>Среднее значение:</i>	0,58	0,53	0,64	0,62	0,57	0,66

В обоих случаях для формирования масок классов по найденным индексам применялась одинаковая процедура постобработки, что позволяет сопоставлять результаты именно с точки зрения влияния способа формирования и упорядочивания информативных индексов.

По результатам, представленным в таблице 3.4, видно, что применение алгоритма MINDI позволило повысить средние значения всех рассматриваемых метрик по сравнению с независимым применением INDI. Среднее значение F_2 увеличилось с 0,58 до 0,62, точности – с 0,53 до 0,57,

полноты – с 0,64 до 0,66. В относительном выражении прирост составил приблизительно 6,9% для F_2 , 7,5% для точности и 3,1% для полноты.

Полученные результаты показывают, что использование информации об отделимости классов при формировании ансамбля информативных индексов положительно влияет на качество многоклассовой сегментации в среднем по всем классам. Это подтверждает целесообразность введения этапа упорядочивания классов: алгоритм сначала выделяет классы, которые обладают большей отделимостью относительно остальных, а затем переходит к более сложным для разделения классам.

При этом следует отметить, что улучшение не является равномерным для всех классов. Для ряда классов, например "Необработанный виноград", "Почва виноградников", "Листовой салат №3", "Листовой салат №4" и "Виноградники", алгоритм MINDI обеспечивает заметное увеличение значений F_2 . Так, для класса "Листовой салат №4" значение F_2 увеличилось с 0,00 до 0,84, а для класса "Листовой салат №3" – с 0,00 до 0,50. Это указывает на то, что последовательное формирование ансамбля индексов с учётом отделимости классов позволяет обнаруживать классы, которые не были корректно выделены при независимом применении INDI.

В то же время для некоторых классов наблюдается снижение качества сегментации. Например, для классов "Брокколи сорт 1", "Вспаханное поле", "Вспаханное плугом", "Листовой салат №2" и "Кукуруза" значения F_2 при использовании MINDI оказались ниже, чем при применении INDI. Это связано с особенностями последовательной схемы формирования ансамбля индексов: ошибки, возникающие на отдельных этапах выделения классов, могут влиять на последующее построение карты сегментации. Кроме того, классы со сходными спектральными характеристиками могут конкурировать между собой, что затрудняет их устойчивое разделение с помощью одного нормализованного разностного индекса.

Таким образом, результаты эксперимента показывают, что алгоритм MINDI в среднем повышает качество многоклассовой сегментации по сравнению с независимым применением INDI, однако эффективность алгоритма зависит от спектральной разделимости конкретных классов и качества найденных информативных индексов. Это позволяет рассматривать MINDI как более устойчивый способ формирования ансамбля спектральных индексов для многоклассового анализа, но также указывает на необходимость дальнейшего совершенствования процедуры формирования итоговой карты сегментации и разрешения конфликтов между классами.

3.3.2 Результат разработки алгоритма формирования ансамбля информативных признаков

В результате проведённого исследования был разработан алгоритм MINDI, предназначенный для формирования ансамбля информативных индексов для решения задачи многоклассовой сегментации ГСИ. В отличие от независимого применения алгоритма INDI к каждому классу, алгоритм MINDI учитывает отделимость классов и формирует индексы в порядке, определяемом критерием отделимости. Такой подход позволяет использовать информацию о структуре классов в спектральном пространстве и повысить качество итоговой карты сегментации.

Разработанный алгоритм предоставляет возможность применять спектральные индексы не только для бинарного разделения объектов, но и для многоклассового анализа ГСИ. Это расширяет область применения методов на основе спектральных индексов и делает их более пригодными для задач, в которых требуется построение карты классов по данным с несколькими классами объектов.

Одним из преимуществ алгоритма MINDI является то, что для его применения не требуется большой объём обучающих данных, в отличие от многих методов машинного обучения, в том числе нейросетевых подходов. Алгоритм опирается на вычисление статистик классов и поиск

информативных спектральных индексов, что делает его применимым в условиях ограниченного числа размеченных данных. Это особенно важно для анализа ГСИ, где получение качественной экспертной разметки является трудоёмким и дорогостоящим процессом.

Экспериментальная оценка на изображении Salinas Scene показала, что использование алгоритма MINDI позволяет повысить средние значения F_2 , точности и полноты по сравнению с независимым применением INDI. Полученные результаты подтверждают практическую значимость предложенного подхода и показывают, что учёт отделимости классов при формировании ансамбля индексов может улучшать качество многоклассовой сегментации.

Таким образом, алгоритм MINDI может рассматриваться как инструмент многоклассового анализа ГСИ на основе автоматически формируемого ансамбля информативных индексов. Его применение позволяет повысить доступность индексных методов анализа, снизить зависимость от больших размеченных наборов данных и обеспечить более интерпретируемое решение по сравнению с рядом моделей машинного обучения.

По материалам данного раздела опубликованы работы [135, 136]. Также получены государственные свидетельства о регистрации программы для ЭВМ [137].

3.4 Алгоритм автоматического построения ансамбля информативных индексов (A-MINDI)

Алгоритм A-MINDI предназначен для автоматизации формирования ансамбля информативных индексов в условиях отсутствия исходной экспертной разметки. Его основная идея заключается в том, что начальная разметка генерируется автоматически, после чего используется в качестве предварительной для последующего построения ансамбля индексов и получения уточненной разметки, учитывающей спектральные признаки.

Пусть задано гиперспектральное изображение X , для которого эталонная разметка отсутствует. Тогда вводится оператор автоматического формирования предварительной разметки

$$\Phi: X \rightarrow \tilde{Y},$$

где $\tilde{Y}$ – предварительная разметка, полученная автоматически, например с использованием модели сегментации Fast Segment Anything Model (SAM) [138]. После этого данная разметка используется как вход для алгоритма формирования ансамбля информативных индексов.

Тем самым алгоритм A-MINDI решает задачу автоматизированного анализа ГСИ без необходимости ручного создания исходной разметки:

$$X \rightarrow \tilde{Y} \rightarrow I^* \rightarrow \hat{Y}.$$

Практически это означает перенос ключевых этапов подготовки данных и выбора информативных признаков с пользователя на алгоритм. В результате это позволит использовать интеллектуальную систему не только специалистами, обладающими компетенциями в области анализа ГСИ, но и пользователями, не имеющими опыта ручного конструирования индексов, выбора спектральных каналов и настройки процедур сегментации.

Разработанный алгоритм A-MINDI включает следующие этапы обработки данных:

- 1) формирование трехканального изображения методом главных компонент (PCA);
- 2) применение нейросетевой модели для автоматического формирования предварительной разметки изображения;
- 3) применение алгоритма формирования ансамбля информативных индексов (MINDI) для формирования уточненной разметки.

3.4.1 Применение метода главных компонент в рамках предложенного алгоритма

Применение метода главных компонент в рамках разработанного алгоритма связано с особенностями применения нейросетевой модели

FastSAM. Эта модель принимает на вход *только* трёхканальные изображения.

В рамках разработанного алгоритма предлагается использовать:

- трехканальные изображения, сформированные из RGB слоев ГСИ;
- изображения, составленные комбинацией трёх главных компонент, полученных в результате использования PCA.

В данной работе исследуется применение нейронной сети FastSAM к изображениям, полученным с использованием метода главных компонент (PCA). Выбор данного подхода обусловлен тем, что PCA позволяет эффективно преобразовывать исходное ГСИ произвольной размерности в трёхканальное представление. Это особенно важно в условиях отсутствия априорной информации о соответствии спектральных каналов конкретным длинам волн.

Для применения метода главных компонент исходное гиперспектральное изображение X_0 , имеющее размер (h, w, b) , где h - высота изображения, w – ширина изображения, b – количество спектральных слоев, преобразуется в матрицу X размера $(h \times w, b)$. Строки этой матрицы представляют наблюдения, а столбцы – признаки. Далее вычисляется ковариационная матрица по следующей формуле:

$$\Sigma = \frac{1}{h \times w} (X - \bar{X})^T (X - \bar{X}),$$

где $\bar{X}$ – вектор средних значений по столбцам.

После необходимо вычислить собственные значения λ_i и собственные векторы v_i для ковариационной матрицы. Собственные вектора образуют базис в пространстве признаков:

$$\Sigma v_i = \lambda_i v_i.$$

Далее собственные значения сортируются в убывающем порядке, и выбираются первые *три* собственных вектора, соответствующие наибольшим собственным значениям. Эти вектора образуют новые ортогональные главные

компоненты, на которые проецируется исходное ГСИ. В итоге получаем новую матрицу данных Z размерности $(h \times w, 3)$.

$$Z = X \times V_k,$$

где V_k – матрица, содержащая первые 3 собственных вектора.

Это позволяет сократить размерность ГСИ, сохраняя только наиболее важные компоненты. Из этих компонентов формируется трёхканальное изображение, которое используется нейронной сетью для формирования набора масок объектов.

3.4.2 Метод автоматической разметки гиперспектральных изображений

Применение нейронной сети Fast Segment Anything Model является ключевым этапом разработанного алгоритма и позволяет автоматизировать процесс формирования информативных дескрипторов. Архитектура FastSAM использует глубокие модели для извлечения признаков из изображений и состоит из трёх основных компонентов:

- 1) *image encoder* – кодировщик изображений представляет собой предобученный Vision Transformer (ViT), который извлекает высокоуровневые признаки из изображения;
- 2) *prompt encoder* – преобразует вводимые инструкции в форматы, которые понятны модели, включая точки и рамки;
- 3) *mask decoder* – преобразует признаки изображения и промптов в сегментационные маски, используя механизмы внимания и другие современные подходы.

В предлагаемой технологии текстовые инструкции не используются, так как это может ограничить процесс автоматизации. Для корректного применения модели FastSAM для ГСИ в качестве входа используются трёхканальные изображения, полученные методом главных компонент.

Основным преимуществом нейронной сети FastSAM является ее гибкость: данная модель может быть использована для решения широкого

спектра задач сегментации с высокой точностью и скоростью обработки. FastSAM способна обрабатывать неоднозначные случаи, где объекты накладываются друг на друга или имеют сложные границы. Благодаря этому FastSAM является удобным инструментом для определения областей интереса.

Однако в рамках текущей задачи у FastSAM есть существенный недостаток – данная модель не адаптирована для работы с данными высокой размерности, такими как ГСИ. Поэтому в рамках разработанного алгоритма на вход данной модели предлагается подавать трехканальные изображения, полученные обработкой исходных данных методом главных компонент.

FastSAM позволяет автоматизировать процесс предварительной разметки гиперспектральных данных, особенно когда традиционная разметка требует значительных трудозатрат или компетенций. Маски, сформированные с помощью нейронной сети, используются для дальнейшего формирования информативных индексов, различающих множество классов с помощью алгоритма MINDI.

3.4.3 Описание разработанного алгоритма A-MINDI

На рисунке 3.17 представлена структурная схема алгоритма A-MINDI. Алгоритм реализует конвейер автоматизированной обработки гиперспектральных данных, включающий три основных этапа. На первом этапе исходное ГСИ преобразуется методом главных компонент в трёхканальное изображение, пригодное для подачи на вход модели FastSAM. На втором этапе с помощью FastSAM формируется набор предварительных масок объектов, который затем фильтруется по площади и объединяется методом NMS для получения предварительной разметки. На третьем этапе исходные гиперспектральные данные совместно с предварительной разметкой используются алгоритмом MINDI для формирования ансамбля информативных индексов и построения итоговой маски классов.

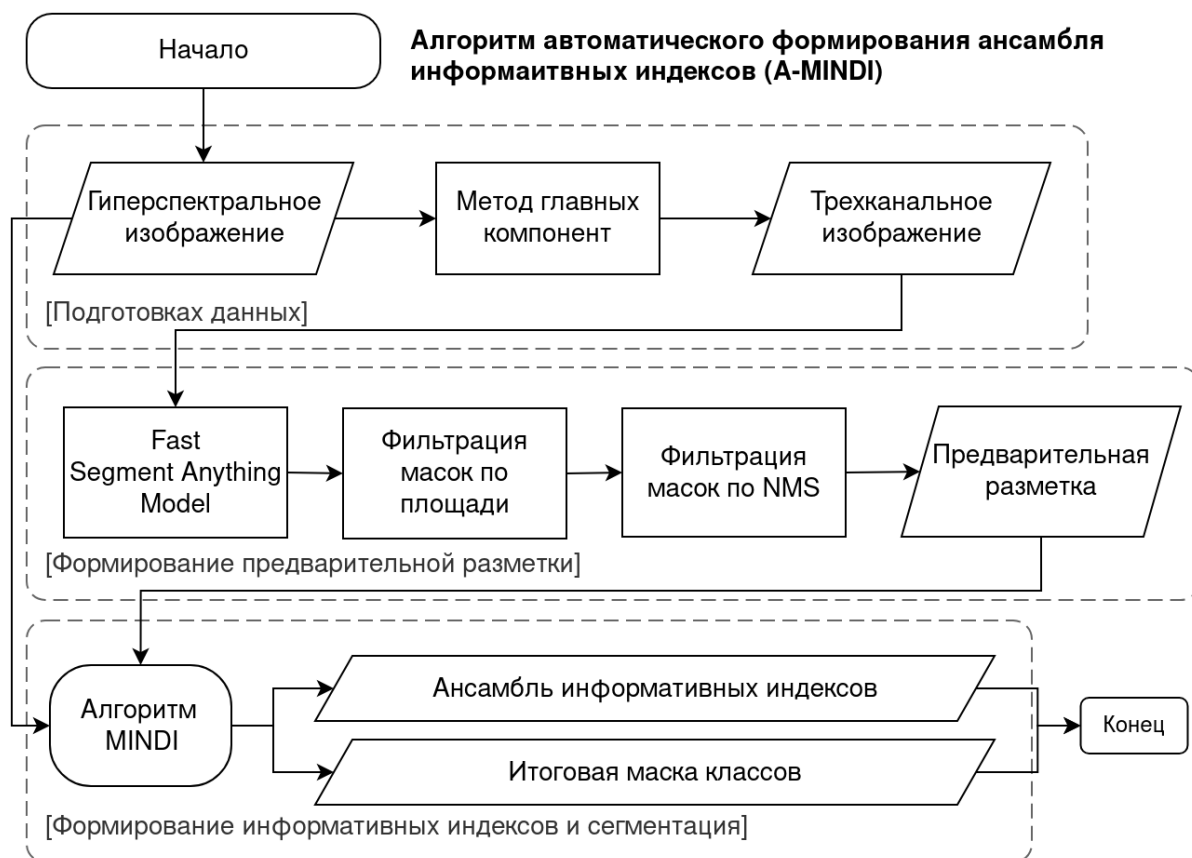


Рисунок 3.17 – Диаграмма, описывающая этапы работы предложенного алгоритма

Процесс использования данной технологии начинается с преобразования ГСИ с помощью метода главных компонент для формирования трёхканального изображения. Пример полученного таким образом изображения приведён на рисунке 3.18 в сравнении с RGB изображением.

После применения модели FastSAM и фильтрации полученных масок можно сформировать предварительную разметку, пример которой изображен на рисунке 3.19.

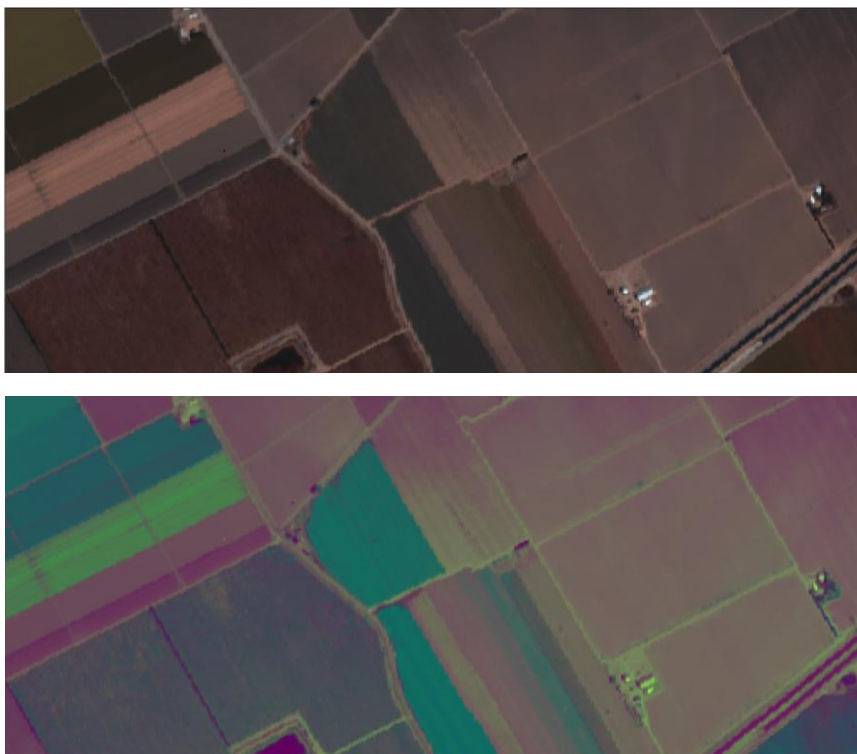


Рисунок 3.18 – Сравнение RGB компонент гиперспектрального изображения и изображения сформированного из главных компонент

Предварительная разметка



Рисунок 3.19 – Визуализация примера предварительной разметки, полученной с помощью FastSAM

Для количественной оценки качества разметки, формируемой на различных этапах алгоритма A-MINDI, используется сравнение с эталонной разметкой. В рамках эксперимента рассматриваются три карты меток: Y, Y_{sam}, Y_{amindi} . Сравнение разметок выполнялось следующим образом.

Сначала оценивается качество предварительной разметки Y_{sam} и качество уточненной разметки Y_{amindi} :

$$ARI_{pre} = ARI(Y, Y_{sam}),$$

$$ARI_{final} = ARI(Y, Y_{amindi}).$$

После чего вычисляется разница $\Delta ARI = ARI_{final} - ARI_{pre}$. Такой подход позволяет определить, насколько применение алгоритма MINDI улучшает или ухудшает предварительную разметку.

Метрика ARI (Adjusted Rand Index) используется для оценки степени совпадения двух разметок с учетом случайного совпадения кластеров. В отличие от обычной доли совпадающих пикселей, ARI не требует совпадения числовых обозначений классов и оценивает согласованность разбиений множества объектов на группы. Это особенно важно при сравнении разметок, полученных автоматически, поскольку порядок и номера классов в них могут не совпадать с эталонной разметкой. Значения метрики находятся в диапазоне $[-1, 1]$, где $ARI = 1$ соответствует полному совпадению двух разметок, $ARI \approx 0$ соответствует уровню случайного согласования, а отрицательные значения указывают на согласованность хуже случайной.

3.4.4 Экспериментальное исследование алгоритма A-MINDI

Для оценки разработанных алгоритмов формирования информативных индексов и автоматизированного получения карт сегментации был проведён вычислительный эксперимент на ГСИ из открытых наборов данных. Цель эксперимента заключалась в проверке того, насколько автоматически сформированная разметка и итоговая карта сегментации, полученная после применения алгоритма MINDI, могут быть использованы не только как самостоятельный результат сегментации, но и как основа для последующего обучения модели машинного обучения.

В качестве экспериментальных данных были выбраны **Salinas Valley**, **Indian Pines** из набора данных Hyperspectral Remote Sensing Scenes и изображения из набора WHU-HI [139]. Набор WHU-Hi включает три

отдельных изображения: *WHU-Hi-LongKou*, *WHU-Hi-HanChuan* и *WHU-Hi-HongHu*, полученные в провинции Хубэй, Китай. Данный набор предназначен для задач точной классификации сельскохозяйственных культур на изображениях высокого пространственного разрешения.

На рисунке 3.20 представлен пример одного из изображений.

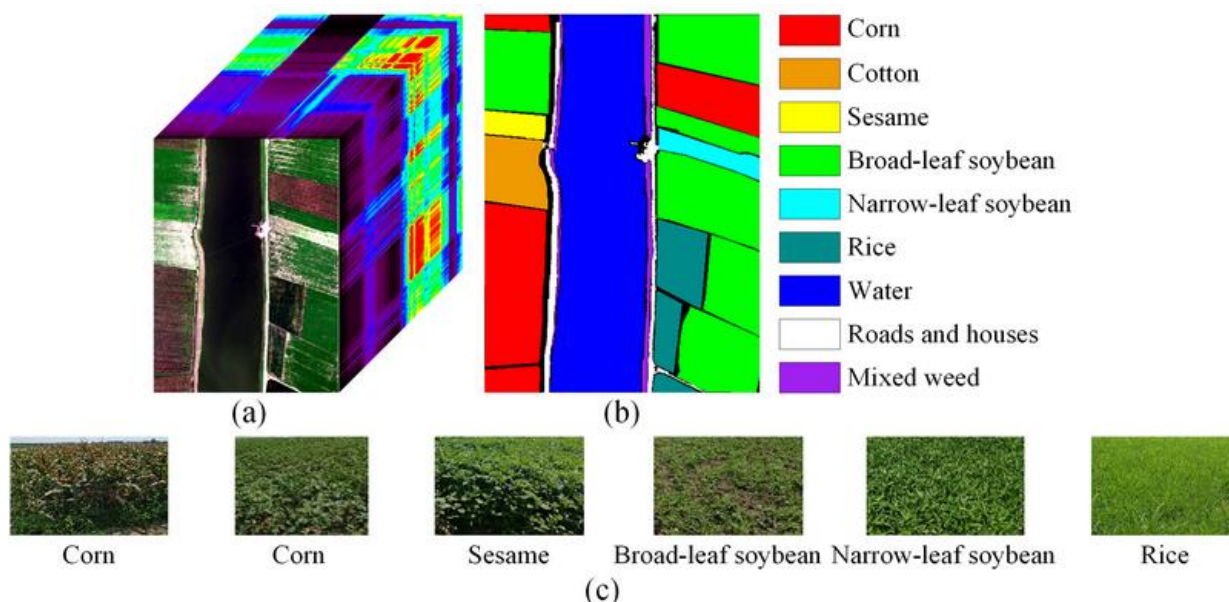


Рисунок 3.20 – Гиперспектральное изображение WHU-Hi-LongKou и его разметка

Использование набора WHU-Hi обусловлено тем, что он содержит изображения, которые характеризуются высоким пространственным разрешением, большим числом размеченных пикселей и достаточно детализированной структурой классов. Это делает его подходящим для оценки алгоритмов автоматической сегментации и формирования информативных спектральных индексов в задачах анализа сельскохозяйственных объектов.

На первом этапе алгоритм A-MINDI оценивался как самостоятельный метод автоматического анализа ГСИ с помощью информативных индексов. На этом этапе обучение моделей машинного обучения не выполнялось.

Таблица 3.5 – Сравнение предварительной разметки и маски классов, полученной с помощью A-MINDI с эталонной разметкой

Изображение	$ARI(Y, Y_{sam})$	$ARI(Y, Y_{amindi})$	ΔARI
Salinas Valley	0,89	0,32	-0,57
Indian Pines	0,55	0,19	-0,36
LongKou	0,80	0,69	-0,11
HanChuan	0,38	0,16	-0,22
HongHu	0,94	0,26	-0,68

Значения ΔARI , представленные в таблице 3.5, во всех случаях являются отрицательными. Это означает, что итоговая карта сегментации Y_{amindi} , полученная после применения ансамбля информативных индексов, хуже согласуется с эталонной разметкой по метрике ARI, чем предварительная разметка Y_{sam} , сформированная с использованием SAM.

На первый взгляд данный результат может свидетельствовать о снижении качества разметки после применения A-MINDI. Однако необходимо учитывать, что ARI оценивает согласованность разбиений множества пикселей и в первую очередь отражает совпадение структур разметки, но не характеризует спектральную однородность классов и их делимость в признаковом пространстве. Предварительная разметка, полученная с помощью SAM, может лучше сохранять пространственную структуру объектов изображения, поскольку SAM ориентирован на выделение визуально целостных областей. В то же время такие области могут быть неоднородными с точки зрения спектральных признаков.

Разметка Y_{amindi} , напротив, формируется на основе информативных спектральных индексов. Поэтому она может хуже совпадать с эталонной картой по ARI, но при этом быть более согласованной со спектральной структурой данных. Для проверки данного предположения был проведён второй этап эксперимента, в котором разметка, полученная A-MINDI,

использовалась как источник меток для обучения модели машинного обучения.

На втором этапе эксперимента оценивался сценарий, в котором результат работы A-MINDI используется как источник разметки для обучения модели машинного обучения. В этом случае алгоритм A-MINDI рассматривается как средство автоматизированной подготовки обучающих данных. В качестве модели использовался многослойный перцептрон (MLP). Модель имела два скрытых слоя размерности 100 и 50 нейронов соответственно. Обучение производилось в течение 100 эпох. Данные параметры выбраны эмпирически для сравнения качества обученных моделей на разных разметках.

Обучающая выборка формировалась по пикселям изображения. Каждому пикселю соответствовал вектор спектральных признаков, а метка класса бралась из разметок Y, Y_{sam}, Y_{amindi} . При формировании выборки использовались не все пиксели изображения. Для уменьшения пространственной зависимости между обучающими примерами применялась процедура отбора непересекающихся образцов. Пиксели выбирались последовательно, при этом после выбора очередного пикселя из дальнейшего рассмотрения исключалась его локальная окрестность в радиусе 5 пикселей. Таким образом, расстояние между выбранными обучающими примерами увеличивалось, что позволяло снизить избыточность выборки, обусловленную высокой пространственной коррелированностью соседних пикселей. После отбора пикселей, сформированная выборка разделялась на обучающую и тестовую части в соотношении 70/30. Для обеспечения воспроизводимости экспериментов зерно генератора случайных чисел было зафиксировано. В таблице 3.6 приведены результаты эксперимента.

Таблица 3.6 – Сравнение результатов обучения модели MLP на разных разметках и наборах данных

Разметка	Метрика	Indian_pines	Salinas	HanChuan	HongHu	LongKou
Y_{amindi}	F_2	0,84	0,82	0,82	0,92	0,86
	Precision	0,84	0,82	0,82	0,92	0,84
	Recall	0,85	0,83	0,83	0,93	0,86
Y_{sam}	F_2	0,17	0,67	0,83	0,76	0,88
	Precision	0,15	0,76	0,83	0,74	0,90
	Recall	0,18	0,67	0,83	0,77	0,87
Y	F_2	0,12	0,43	0,70	0,79	0,90
	Precision	0,06	0,46	0,72	0,78	0,88
	Recall	0,16	0,45	0,70	0,79	0,91

Результаты, представленные в таблице 3.6, показывают, что разметка Y_{amindi} , сформированная алгоритмом A-MINDI, в большинстве случаев является более пригодной для обучения модели MLP, чем предварительная разметка Y_{sam} и исходная разметка Y , использованная в сравнении.

Наиболее заметное преимущество Y_{amindi} наблюдается на наборах Indian Pines, Salinas и HongHu. Для Indian Pines значение F_2 при обучении на Y_{amindi} составило 0,84, тогда как для Y_{sam} и Y значения 0,17 и 0,12 соответственно.

На наборе HanChuan результаты оказались близкими. Это указывает на то, что для данного изображения предварительная разметка SAM уже достаточно хорошо согласована с признаковым пространством, поэтому применение A-MINDI не приводит к существенному дополнительному улучшению результата обучения.

Для набора LongKou наилучшее значение F_2 было получено при использовании разметки Y . Это показывает, что эффективность A-MINDI зависит от структуры конкретного изображения, качества предварительной разметки и спектральной разделимости классов.

В среднем по пяти наборам данных разметка Y_{amindi} обеспечила наибольшие значения всех рассматриваемых метрик.

Таким образом, несмотря на снижение ARI при сравнении Y_{amindi} с исходной разметкой Y , результаты обучения MLP показывают, что разметка, сформированная A-MINDI, может быть более полезной как источник обучающих меток. Данный эффект объясняется различием между задачей пространственного совпадения разметок и задачей формирования обучающей выборки. Метрика ARI оценивает согласованность двух разбиений множества пикселей, но не учитывает спектральные признаки классов и их разделимость в признаковом пространстве. В свою очередь, модель MLP обучается именно на спектральных признаках, поэтому для неё важна согласованность меток с признаковым пространством пикселей.

Следовательно, предварительная разметка Y_{sam} может лучше совпадать с эталонной разметкой как пространственное разбиение, но при этом содержать спектрально неоднородные области. Разметка Y_{amindi} , сформированная на основе информативных индексов, может хуже совпадать с эталоном по ARI, однако задавать более спектрально согласованные классы. Это положительно влияет на обучение модели машинного обучения.

3.4.5 Результат разработки алгоритма автоматического построения ансамбля информативных индексов

Алгоритм A-MINDI может использоваться как самостоятельный метод автоматического анализа ГСИ с помощью информативных индексов. Однако сравнение по метрике ARI показало, что итоговая карта сегментации Y_{amindi} в рассмотренных экспериментах уступает предварительной разметке Y_{sam} по степени согласованности с эталонной разметкой. Это означает, что в текущей конфигурации A-MINDI не всегда улучшает пространственное совпадение карты сегментации с эталоном.

В то же время результаты второго этапа эксперимента показывают, что Y_{amindi} является более пригодной для обучения модели MLP в большинстве рассмотренных случаев. Среднее значение F_2 при обучении на Y_{amindi} составило 0,852, что выше соответствующих значений для Y_{sam} и Y . Аналогичное преимущество наблюдается по средним значениям точности и полноты.

Это позволяет рассматривать A-MINDI не только как алгоритм построения карты сегментации, но и как способ автоматизированного формирования обучающей выборки для последующего обучения моделей машинного обучения. Основная ценность A-MINDI в данном сценарии заключается в том, что алгоритм формирует разметку, более согласованную со спектральным признаковым пространством, даже если такая разметка не всегда демонстрирует более высокое совпадение с эталоном по метрике ARI.

Следовательно, применение A-MINDI целесообразно рассматривать в двух различных сценариях. В первом сценарии алгоритм используется как самостоятельный инструмент автоматического построения карты сегментации. Во втором сценарии A-MINDI выступает как средство автоматизированной подготовки обучающих данных для моделей машинного обучения. Результаты эксперимента показывают, что именно второй сценарий является наиболее перспективным, поскольку позволяет снизить зависимость от ручной разметки и обеспечить формирование спектрально согласованных обучающих меток.

По материалам данного пункта опубликованы работы [140, 141].

Также получено государственное свидетельство о регистрации программы для ЭВМ [142].

3.5 Детали программной реализации алгоритмов INDI, MINDI и A-MINDI

Для применения алгоритмов формирования информативных индексов и автоматического составления их ансамбля была разработана программная

библиотека *InformativeFeatureSelection* [143]. Данная библиотека открыто распространяется через репозиторий пакетов для языка программирования Python – PyPi. На рисунке 3.21 представлена UML диаграмма компонентов, описывающая основные модули данной библиотеки:

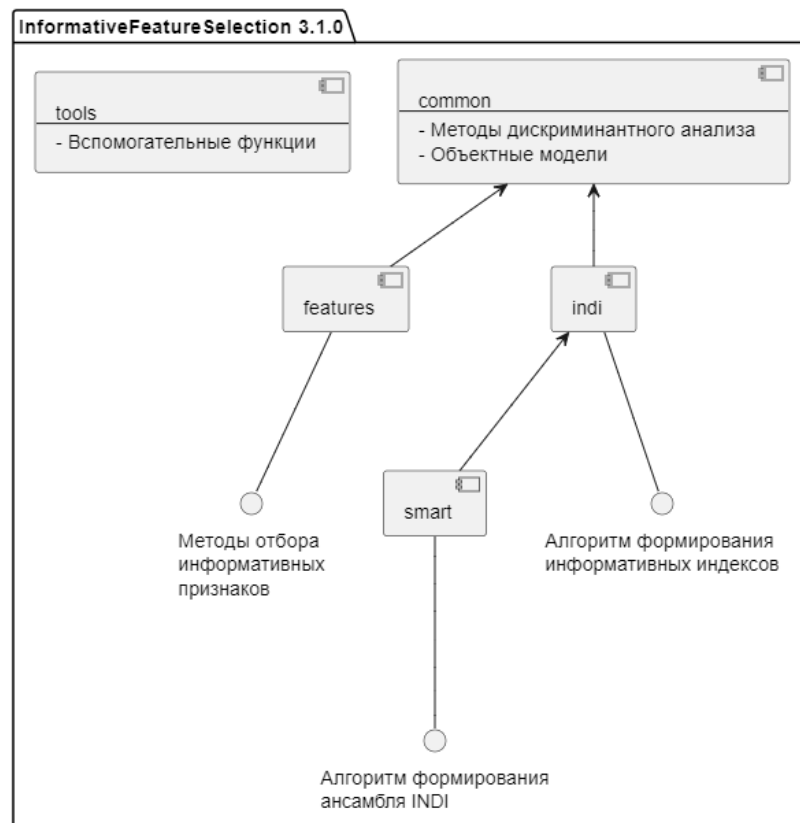


Рисунок 3.21 – Диаграмма компонентов разработанного модуля

Разработанная библиотека состоит из следующих модулей:

1) *Tools* – содержащей вспомогательные функции, например: позволяющие реализовать логирование иллюстрирующее прогресс выполнения тех или иных задач; различные функции пороговой обработки; или функции визуализации результатов применения алгоритмов.

2) *Common* – содержит реализацию базовых математических вычислений. Включает в себя математическую реализацию элементов дискриминантного анализа, используемую для вычисления группового и индивидуального критерия разделимости. Для оптимизации процесса вычислений применяется библиотека векторных вычислений *numpy*. Данная библиотека позволяет реализовывать векторные вычисления что

существенно упрощает разработку и благодаря особенностям реализации библиотеки, существенно ускоряет процесс вычислений.

3) *Features* – вспомогательный модуль, позволяющий использовать дискриминантный анализ для решения задач отбора информативных признаков в любых задачах не обязательно связанных с анализом ГСИ. В рамках данного модуля предлагаются реализации алгоритма отбора информативных признаков, основанных на используемых в работе критериях информативности и отделимости.

4) *Indi* – модуль с реализацией алгоритма формирования информативного индекса как для бинарного случая, так и для многоклассового. В рамках данного модуля реализованы базовые, абстрактные классы, определяющие интерфейс алгоритма формирования информативного индекса и часть вспомогательного функционала по предобработке исходных данных и разметки.

5) *Smart* – модуль, реализующий алгоритм автоматического построения ансамбля информативных индексов. В рамках данного модуля реализованы два класса: предоставляющий интерфейс доступа к нейронной сети, производящей автоматическую предварительную разметку данных и класс реализующий алгоритм построения ансамблей. Для реализации интерфейса доступа к нейронной сети используется библиотека *fast_segment_anything*, посредством которой производится процесс создания нейросетевой модели и загрузки ее параметров. В рамках модуля *Smart* реализован ряд методов, позволяющих автоматически генерировать разметку для переданного трехканального изображения. Данные методы также проводят предварительную фильтрацию сгенерированных масок по площади занимаемой поверхности. После чего, благодаря применению композиции, использует алгоритм MINDI для составления ансамбля информативных индексов.

Далее на рисунках 3.22 и 3.23 представлены две диаграммы, иллюстрирующие взаимосвязь классов. Реализованная библиотека, кроме предоставления алгоритмов также предоставляет набор абстрактных классов и интерфейсов для возможности реализации собственных алгоритмов, основанных на предложенной методологии формирования информативных индексов.

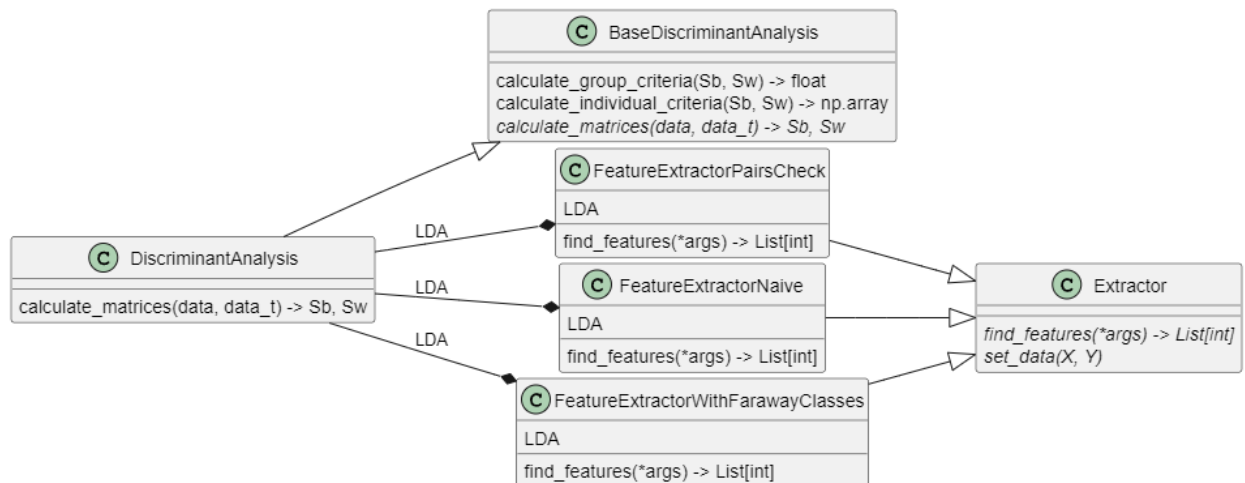


Рисунок 3.22 – Диаграмма классов модуля *Features*

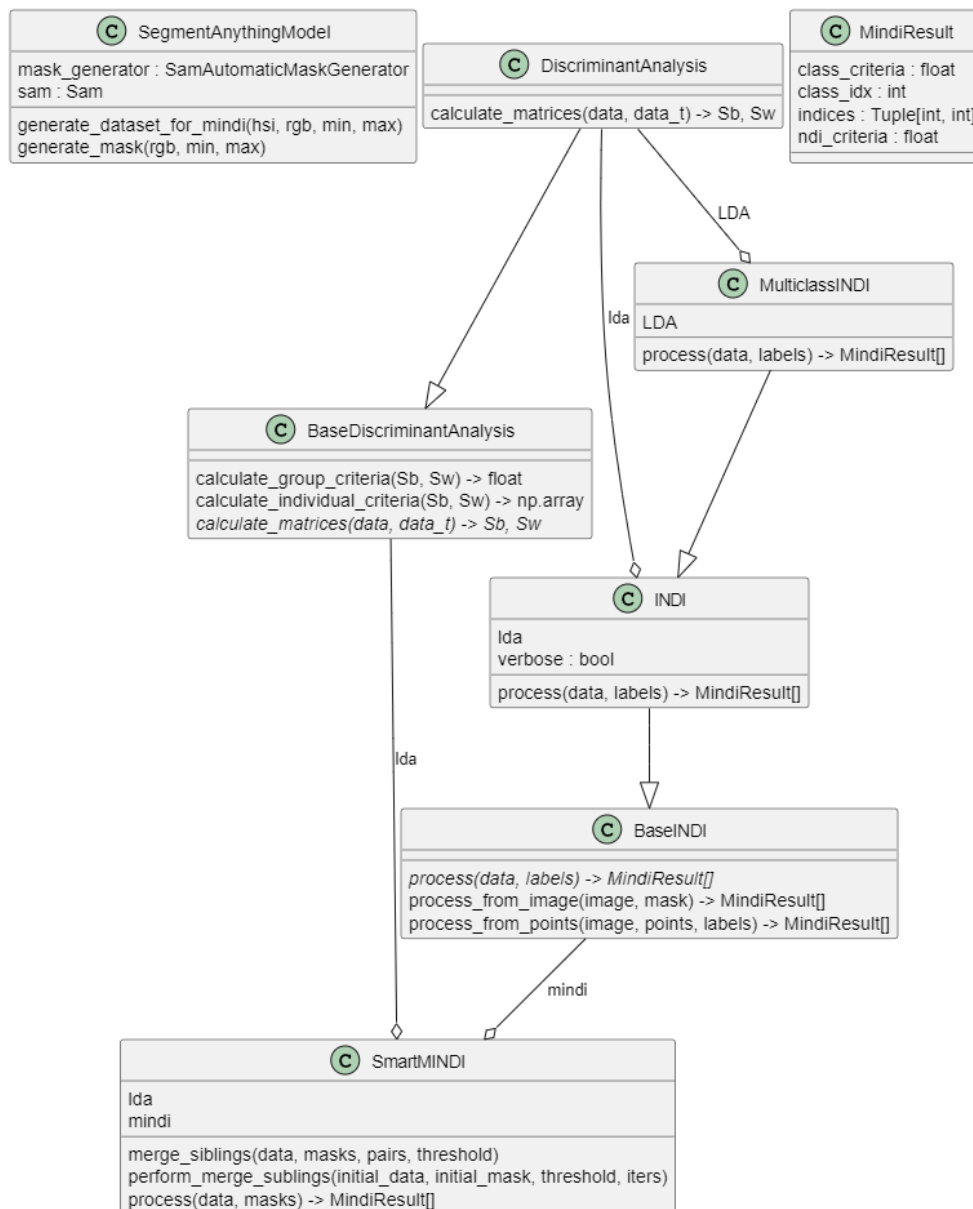


Рисунок 3.23 – Диаграмма классов модулей *Indi* и *Smart*

3.6 Метод анализа с использованием схожести с эталоном

Для проведения обработки ГСИ в рамках разработанной интеллектуальной системы также предлагается использовать более простые методы. Например, метод анализа с использованием схожести с эталоном. Данный метод является чрезвычайно эффективным с точки зрения скорости и простоты вычислений. Его результат представляется в виде тепловой карты, которая визуализирует схожесть областей на ГСИ с выбранным эталонным объектом. Сигнатура эталонного объекта задается в виде соответствующего

ему вектора значений интенсивности. Предусмотрены два способа задания эталонного объекта:

1. **Точечный объект.** В этом случае за эталон принимается вектор значений интенсивности пикселя в заданной точке.

2. **Область произвольной формы.** В этом случае за эталон принимается среднее значение векторов интенсивностей, входящих в заданную область.

Схожесть определяется путём вычисления меры расстояния и последующей нормировки полученных значений. Мера расстояния вычисляется по следующей формуле:

$$d_e(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^n (\vec{x}_i - \vec{y}_i)^2},$$

где, $\vec{x}$ является вектором значений интенсивности ГСИ в точке.

Данная мера вычисляется для каждой точки исходного ГСИ. Общее число вычислений, необходимых для составления карты схожести объектов равно общему числу пикселей у ГСИ. Вектор $\vec{y}$ является вектором значений эталонного объекта, который, задается вектором значений пикселя в заданной точке или средним значением векторов интенсивностей входящих в заданную область. Количество элементов векторов $\vec{x}$ и $\vec{y}$ равно числу спектральных слоев. Применение Евклидовой меры расстояния не обязательно. Вместо нее допускается использовать любую другую меру расстояния.

На рисунке 3.24 представлена диаграмма основных этапов метода анализа схожести с эталонным объектом.

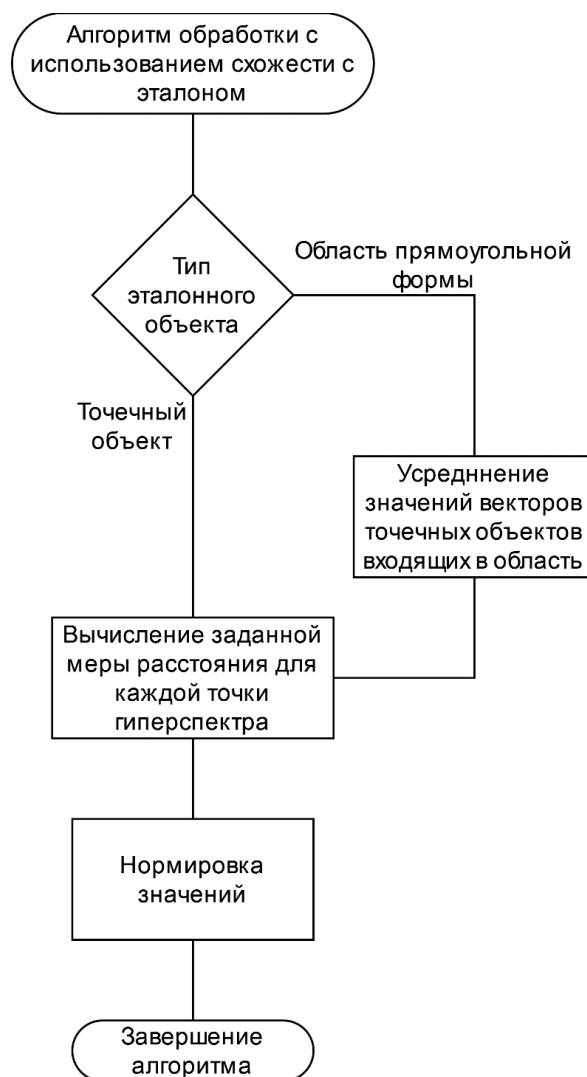


Рисунок 3.24 - Диаграмма алгоритма анализа ГСИ с использованием схожести с эталоном

На рисунках 3.25, 3.26 продемонстрирован результат применения данного алгоритма для анализа ГСИ. Так на рисунке 3.19 была выделена область, на которой изображена почва, а на рисунке 3.20 область с растительностью. Более светлые части изображений соответствуют зонам ГСИ, которые наиболее схожи с выбранной областью.

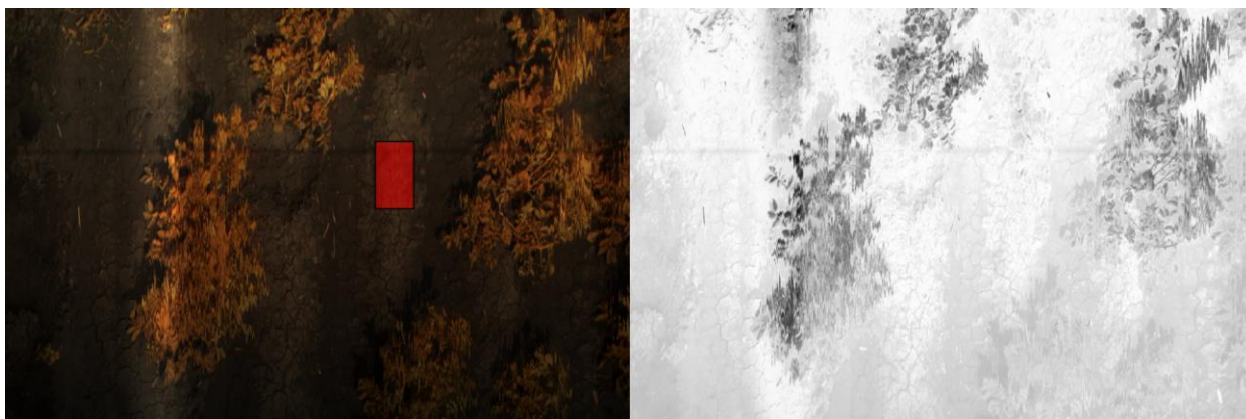


Рисунок 3.25 – Демонстрация алгоритма, выделена область с сухой почвой

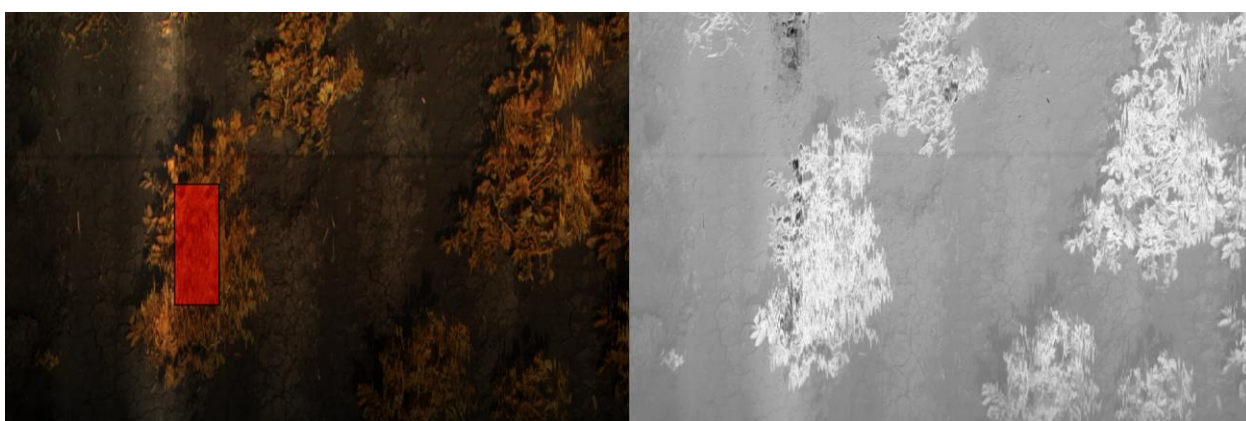


Рисунок 3.26 – Демонстрация алгоритма, выбрана область, выделена область с растительностью

Данный метод, несмотря на свою простоту предоставляет возможность эффективного анализа ГСИ с точки зрения скорости предоставления и интерпретируемости результата. Данный метод при реализации графического интерфейса позволяет производить оперативный анализ ГСИ.

3.7 Метод обработки с использованием вегетационных индексов

Вегетационные индексы являются одним из наиболее популярных способов анализа ГСИ. Они представляют собой простые вычислительные правила, позволяющие комбинировать различные слои ГСИ с целью получения тепловых карт, оценивающих те или иные характеристики объектов ГСИ. Кроме того, данные индексы, несмотря на их простоту могут учитывать различные особенности спектральных сенсоров, позволяя устранить шум или влияние внешних факторов. Рассмотрим далее ряд вегетационных индексов, наиболее часто встречаемых при анализе ГСИ.

Пусть NIR – отражение в ближней инфракрасной области спектра, RED – отражение в красной области спектра, $GREEN$ – отражение в зеленой области спектра, $BLUE$ – отражение в синей области спектра. Тогда перечень индексов для использования в рамках реализованной системы будет выглядеть следующим образом:

1) RVI (Ratio Vegetation Index)

Этот индекс основан на предположении, что здоровая растительность поглощает большую часть видимого света (красного) и отражает значительную часть ближнего инфракрасного света.

$$RVI = \frac{NIR}{RED}.$$

2) NDVI (Normalized Difference VI)

Данный индекс является индикатором здоровья растительности.

$$NDVI = \frac{NIR - RED}{NIR + RED}.$$

3) NDWI (Normalized Difference Water Index)

Данный индекс позволяет оценить увлажненность почвы и растений.

$$NDWI = \frac{NIR - GREEN}{NIR + GREEN}.$$

4) EVI (Enhanced VI)

EVI выполняет аналогичную индексам RVI и NDVI роль, но позволяет устранить влияние фонового шума и уменьшает влияние атмосферных искажений на получаемый результат.

$$EVI = 2,5 \frac{(NIR - RED)}{NIR + 6 RED - 7,5 BLUE + 1}.$$

5) IPVI (Infrared Percentage VI)

Данный индекс является аналогом индекса NDVI, но из-за различий в формулировке позволяет минимизировать влияние почвы на оценку здоровья растений.

$$IPVI = \frac{NIR}{NIR + RED}.$$

6) DVI (Difference VI)

Данный индекс является аналогом индекса NDVI не подразумевающий нормировку значений, что сказывается на чувствительности данного индекса к внешним факторам и неравномерности яркостей.

$$DVI = NIR - RED.$$

7) PVI (*Perpendicular VI*)

Данный индекс предназначен для оценки характеристик растительности путем измерения перпендикулярного расстояния от линии фона почвы до точки в пространстве, определяемом ближней инфракрасной (NIR) и красной (видимой) отражательной способностью. Концепция PVI заключается в уменьшении влияния яркости почвы при обнаружении и количественной оценке растительности.

$$PVI = \sin(\alpha) NIR - \cos(\alpha) RED,$$

где α – угол между почвенной линией и осью NIR (параметр).

8) WDV (Weighted Difference VI)

Данный индекс применяется для решения той же задачи что и NDVI, но позволяет уменьшить влияние яркости почвы за счет добавления коэффициента наклона почвенной линии.

$$WDVI = NIR - \rho RED,$$

где ρ – наклон почвенной линии.

9) SAVI (*Soil Adjusted VI*)

Данный индекс предназначен для оценки состояния растительности в случае, когда растительности на изображении мало и она рассеяна,

$$SAVI = \frac{NIR - RED}{NIR + RED + L} (1 + L),$$

где L – наклон почвенной линии.

10) TSAVI (*Transformed Soil Adjusted VI*)

Модификация индекса SAVI, добавляющая параметры используемые для коррекции яркости почвы и плотности растительности.

$$TSAVI = \frac{s (NIR - s RED - \alpha)}{\alpha NIR + RED - \alpha s + X (1 + s^2)},$$

где s – наклон почвенной линии;

α – координата пересечения почвенной линии с осью NIR;

X – коэффициент коррекции.

11) MSAVI (Modified Soil Adjusted VI)

Также является модификацией индекса SAVI, предназначенная для оптимизации оценки здоровья растительности за счет уменьшения влияния яркости почвы благодаря использованию квадратичного уравнения.

$$MSAVI = \frac{NIR - RED}{NIR + RED - L} (1 + L),$$

где L – наклон почвенной линии.

12) GEMI (Global Environmental Monitoring Index)

Данный нелинейный индекс спроектирован так, чтобы минимизировать влияние почвы и атмосферы на оценку состояния растительности. Данный индекс достаточно чувствителен к плотности расположения растений на изображении.

$$GEMI = E(1 - 0,25 E) - \frac{RED - 0,125}{1 - RED},$$
$$E = \frac{2 (NIR^2 - RED^2) + 1,5 NIR + 0,5 RED}{NIR + RED + 0,5}.$$

13) ARVI (Atmospherically Resistant VI)

Модифицированный индекс NDVI предназначенный для применения в случаях сильного влияния атмосферных эффектов и аэрозолей.

$$ARVI = \frac{NIR - RED - \alpha (RED - BLUE)}{NIR + RED - \alpha (RED - BLUE)}.$$

3.8 Анализ гиперспектральных изображений методом К-средних

Метод кластеризации К-средних является достаточно популярным способом анализа ГСИ. Его применение позволяет разделить группы пикселей на K непересекающихся кластеров. Благодаря высокой информативности ГСИ этот метод позволяет эффективно сегментировать изображение при заранее известном числе классов. Цель данного метода – сгруппировать пиксели таким образом, чтобы минимизировать расстояние между пикселями внутри каждого кластера. Это достигается минимизацией целевой функции, которая оценивает расстояние между пикселями и центрами кластеров. В большинстве

случаев в качестве меры расстояния используется Евклидова метрика. Целевая функция для минимизации может быть представлена следующим образом:

$$J = \sum_{k=1}^K \sum_{i \in C_k} \|x_i - \mu_k\|^2,$$

где C_k – множество пикселей, принадлежащих кластеру k , μ_k – центр кластера k .

После задания целевой функции можно описать метод кластеризации К-средних:

- 1) первым этапом данного метода является инициализация центроидов кластеров $\mu_1, \mu_2, \dots, \mu_K$. Чаще всего инициализация производится случайным образом;
- 2) далее каждый пиксель изображения присваивается ближайшему кластеру на основе значения Евклидовой меры расстояния:

$$k = \operatorname{argmin}_j \|x_i - \mu_j\|^2;$$

- 3) после производится обновление центра кластера. Новый центр будет является средним пикселей всех пикселей входящих в кластер:

$$\mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} x_i;$$

- 4) далее шаги (2) и (3) повторяются до тех пор, пока изменение центров кластеров между итерациями не станет достаточно малым.

Применение метода К-средних для анализа ГСИ сопряжено с рядом сложностей:

– проблема выбора числа кластеров K . Выбор числа кластеров существенно влияет на качество результата получаемого с помощью данного метода;

– влияние инициализации кластеров. Способ первоначальной инициализации центров кластеров также может существенно повлиять на итоговый результат. Чтобы добиться лучшего результата, часто применяют специальные методы инициализации или применяют алгоритм несколько раз;

– данный метод чувствителен к выбору меры расстояния. Таким образом, при использовании Евклидовой меры предполагается, что все кластеры имеют сферическую форму, что не всегда справедливо, особенно при анализе сложных материалов на ГСИ.

Несмотря на эти сложности, метод К-средних широко применяется благодаря высокой информативности данных ГСИ, что позволяет эффективно использовать простые алгоритмы.

3.9 Метод эффективного хранения гиперспектральных изображений

Гиперспектральные изображения представляют собой трехмерные массивы данных, которые, из-за высокого пространственного и спектрального разрешения, могут содержать до нескольких миллиардов значений. Это создает серьезные требования к объему памяти для их хранения и обработки, особенно в условиях реализации облачной интеллектуальной системы, где одновременно может поступать множество запросов от пользователей. Для эффективного хранения и обработки таких данных необходимо:

- 1) организовывать эффективное хранение спектральных изображений с точки зрения занимаемой ими *постоянной памяти*;
- 2) обеспечить эффективное хранение спектральных изображений в *оперативной памяти* во время выполнения вычислений.

Для решения проблемы организации физического пространства хранения ГСИ наиболее эффективным методом является формирование горячих и холодных хранилищ данных. Организация хранения данных в зависимости от частоты их использования является классическим подходом при реализации облачных сервисов, предоставляющих доступ для хранения большого числа данных. Поэтому предлагается следующий метод хранения данных:

- данные, используемые чаще чем один раз в течении семи дней, должны храниться в устройствах, обеспечивающих наиболее быструю скорость чтения и записи: твердотельные накопители (SSD SATA, SSD M.2);

- данные, хранимые на твердотельных накопителях, должны храниться без сжатия;

- данные, к которым не происходит обращения в течении семи дней должны передаваться на хранение в более емкие, но более медленные устройства: жесткие диски (HDD);

- данные, хранимые на медленных устройствах, должны быть подвергнуты максимальному сжатию с целью экономии занимаемого пространства.

При хранении ГСИ важно учитывать их формат. Так если это возможно, ГСИ должны быть сохранены в формате, подразумевающим наименьшие затраты памяти, например восьмибитные форматы чисел: `int8` или `uint8`. Если это невозможно, используется формат чисел с плавающей точкой `float`. Использование наименее объемных форматов данных также позволит существенно снизить потребление оперативной памяти при проведении вычислений.

Для эффективной организации использования ГСИ во время вычислений, с целью снижения объема потребляемой оперативной памяти, необходимо хранить ГСИ в особом формате. Поэтому, при хранении и использовании данных, предлагается использовать их “плоское” представление. Несмотря на то, что ГСИ представляют собой трехмерные массивы данных, наиболее эффективным способом их хранения в оперативной памяти является непрерывное хранение. Непрерывное хранение подразумевает собой то, что все значения многомерного массива данных хранятся в памяти непрерывно, таким образом что следующее значение массива хранится в следующей ячейке памяти. Рассмотрим данный метод хранения на примере двумерного массива (матрицы) размера 3x3:

0	1	2	
3	4	5	хранится в памяти как 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
6	7	8	

Данный способ хранения данных называется "с-contiguous" или построчный способ хранения. Он позволяет обеспечивать доступ к многомерным данным наиболее эффективным, с точки зрения скорости, способом. Однако данное представление данных теряет информацию о многомерной природе. Для обеспечения доступа к данным так, будто они являются многомерными необходимо использовать информацию о "шагах" или "strides". Данные шаги представляют собой набор чисел, значения которых определяют расстояние, на которое необходимо сдвинуться в памяти, чтобы произвести переход к следующему элементу относительно определенной оси. На примере трехмерного массива данных, данные шаги будут представляться тройкой чисел: где первое число будет отвечать за количество байт на которые нужно сдвинуться в памяти для итерации в глубину по элементам массива, второе число за итерацию по строкам, а третье по столбцам. Предположим, что мы храним трехмерный массив данных размера (2, 3, 4), два спектральных слоя, три строки (высота) и четыре столбца (ширина). Данный массив данных содержит числа в формате int8 (byte). Тогда шаги для обеспечения доступа к данному массиву данных как многомерному будут выглядеть следующим образом: (12, 4, 1), что будет означать, что:

- для доступа к следующему слою необходимо сдвинуться на 12 байт;
- для доступа к следующей строке необходимо сдвинуться на четыре байта;
- для доступа к следующему столбцу необходимо сдвинуться на один байт.

Такой способ представления многомерных данных позволяет с минимальными дополнительными затратами обеспечивать хранение данных в оперативной памяти, обеспечивая при этом наибольшую скорость доступа. Другим преимуществом данного метода является тот факт, что он позволяет

производить некоторые манипуляции с данными без необходимости непосредственного их изменения (без переадресации памяти, без копирования и изменения значений). Например, при транспонировании такого многомерного массива данных необходимо лишь изменить порядок “шагов”. Кроме этого, данный способ представления данных позволяет обращаться к данным для их частичного использования без загрузки всего массива данных в оперативную память. Например, при вычислении вегетационного индекса NDVI нет необходимости производить чтение всего ГСИ. Вместо этого достаточно взять всего лишь две спектральные полосы изображения, необходимые для вычисления вегетационного индекса.

Для реализации вышеописанного метода использования ГСИ можно использовать библиотеку *numpy* для языка программирования Python. Именно она реализует описанный ранее способ хранения и представления многомерных массивов данных. Сравним эффективность ее применения относительно использования стандартных средств языка программирования Python на примере нескольких сценариев:

- 1) чтение файла гиперспектрального изображения и вычисление индекса NDVI;
- 2) преобразование гиперспектрального изображения: транспонирование.

В таблице 3.7 приведены результаты данного сравнения.

Таблица 3.7 – Результаты сравнительного эксперимента, оценивающего преимущество использования подхода частичного чтения и библиотеки *numpy*

<i>Метод</i>	<i>Время обработки (мс.)</i>	<i>Потребление памяти (МБ)</i>
<i>Сценарий №1. Вычисление вегетационного индекса</i>		
Python vanilla	804	5,13
Numpy	0,753	4,79
Numpy memmap	0,420	1,46
<i>Сценарий №2. Транспонирование гиперспектрального изображения</i>		
Python vanilla	1170	17,86
Numpy	0,522	3,12
Numpy memmap	0,029	0,00

Отраженные в таблице 3.7 результаты показывают эффективность описанного способа представления ГСИ. В частности, отметим, что наилучший результат как по производительности, так и по затратам памяти показывает применение библиотеки *numpy* и его метода *memmap*. Обратим также внимание на отсутствие факта потребления памяти при выполнении операции транспонирования массива данных при использовании метода *memmap*. Это связано с озвученным ранее способом хранения данных. При операции транспонирования нет необходимости даже чтения исходных данных. Достаточно лишь поменять местами значения *stride*. Единственным недостатком данного подхода является необходимость хранения данных без сжатия, что на самом деле является типичным компромиссом между скоростью обработки и занимаемой памятью.

3.10 Метод мозаичной обработки данных

ГСИ, как правило, имеют большую пространственную размерность, например снятые со спутника ГСИ могут иметь разрешение порядка десятков тысяч пикселей. Передача изображений высокого разрешения пагубно сказывается на отзывчивости облачных сервисов. А обработка таких больших данных может требовать существенного количества ресурсов и времени. Решить данную проблему можно при помощи тайловой или мозаичной загрузки и обработки данных. Данный тип загрузки позволяет программному обеспечению выполнять обработку на небольших фрагментах исходного изображения, например размера 256×256 пикселей, параллельно или по очереди. Это позволяет: снизить нагрузку на сетевой канал, обрабатывать фрагменты параллельно, обрабатывать только необходимые (видимые) фрагменты. Данный тип обработки данных способен существенно снизить затраты ресурсов на выполнение анализа, а также на передачу данных. На рисунке 3.27 представлена диаграмма, визуализирующая данный процесс.

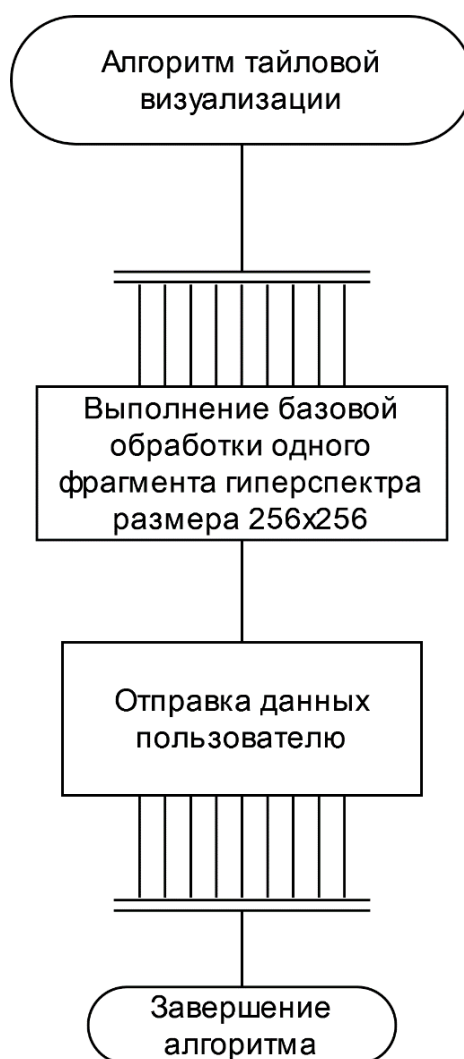


Рисунок 3.27 – Диаграмма алгоритма мозаичной обработки

Далее в таблице 3.8 отображены результаты замеров времени обработки ГСИ размера (2048, 2048, 220) используя два подхода:

- вычисление индекса NDVI на всем ГСИ;
- вычисление индекса NDVI на всех фрагментах размера (512, 512, 220).

Таблица 3.8 – Результаты применения подхода мозаичной обработки ГСИ

Метод обработки	Время обработки (мс.)	Время обработки патча (мс.)
Целиком	132,00	132,00
Мозаичная	90,20	2,46

По представленным результатам видно, что обработка изображения по фрагментам оказывается быстрее чем обработка всего изображения за один

подход. Но более важным здесь является скорость обработки фрагмента данных. По полученным результатам скорость обработки одного фрагмента данных в 53 раза быстрее, что соответственно существенным образом позволит улучшить пользовательский опыт взаимодействия с облачным сервисом. Кроме ускорения обработки данных, при использовании данного подхода будет сокращено время и ресурсы необходимые на передачу результата обработки пользователю.

3.11 Методы предварительной нормировки данных

ГСИ как правило не приведены к типу данных `uint8`, что существенно сказывается на возможности их визуализации стандартными средствами, а также возможности комбинации различных результатов обработки данных. Поэтому для приведения значений ГСИ к диапазону от нуля до 255 необходимо предусмотреть специальные методы нормировки значений. В рамках интеллектуальной системы предлагается использовать следующие подходы:

1. **Обрезка значений.** При выборе такой опции, значения пикселей изображения изменяются по следующему правилу:

$$v = \begin{cases} 0, & v < 0 \\ v, & \text{если } 0 \leq v \leq 255, \\ 255, & v > 255 \end{cases}$$

где v – значение пикселя в заданных координатах на каждом слое ГСИ.

2. **Глобальная нормировка значений.** При данном подходе применяется метод минимакс нормализации, где в качестве максимального и минимального значения используют максимум и соответственно минимум по всем спектральным слоям изображения. После чего значения умножаются на 255 для приведения их к целочисленному типу `uint8`.

3. **Нормализация по слоям.** При данном подходе также применяется метод минимакс нормализации, но для каждого спектрального слоя индивидуально. После чего значения умножаются на 255 для приведения их к целочисленному типу `uint8`.

На рисунке 3.28 данный алгоритм представлен в виде диаграммы.

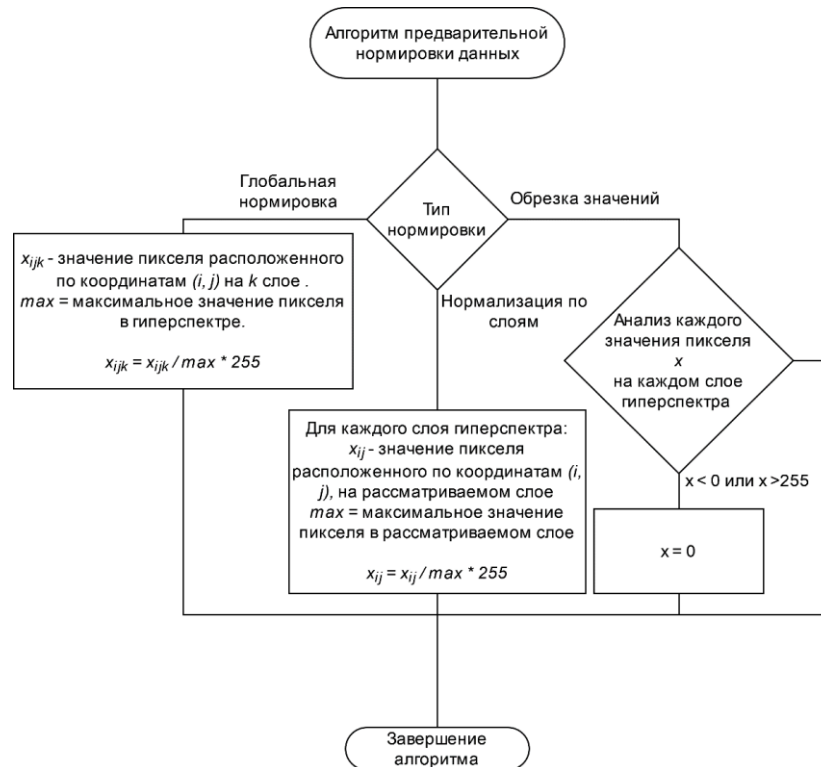


Рисунок 3.28 – Диаграмма алгоритма предварительной нормировки данных

Применение разработанных подходов нормировки ГСИ позволит эффективнее применять алгоритм цветосинтеза, речь о котором пойдет далее.

3.12 Алгоритм цветосинтеза

При наложении различных слоев, будь то исходных или результатов обработки необходимо решить проблему цветосинтеза. Наивное сложение слоев приведет к выходу значений яркости за пределы диапазона от 0 до 255, что не является допустимым. Нормировка значений после сложения слоев приводит к тому, что изображение либо становится слишком “тёмным”, либо некоторые цвета искажаются существенным образом. Использование альфа-канала не позволит явно указать на места каких-либо пересечений из-за отсутствия цветосинтеза. Для решения данных проблем предлагается подход по выбору максимального значения для каждого цветового компонента независимо по всем слоям, выбранным для визуализации. Данный метод описан посредством диаграммы, изображенной на рисунке 3.29. Данный

метод лучше всего позволяет визуализировать ГСИ и результаты их обработки, совмещая на одном полотне различные комбинации слоев.

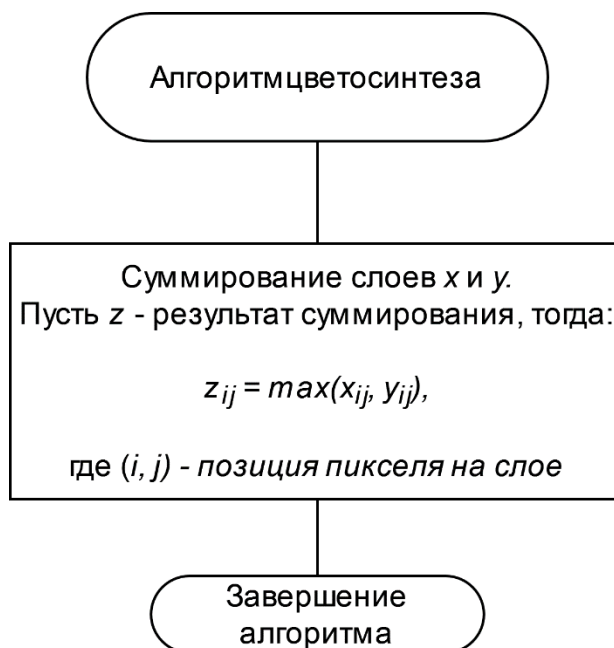


Рисунок 3.29 – Диаграмма алгоритма цветосинтеза

Применение данного подхода выбора цвета пикселя позволяет наиболее очевидно, с точки зрения пользователя, визуализировать ГСИ и результат их обработки.

3.13 Выводы и результаты по третьему разделу

В данном разделе были разработаны и исследованы алгоритмы и программные методы хранения, визуализации и обработки гиперспектральных изображений, составляющие алгоритмическую основу разработанной интеллектуальной системы. Основной акцент был сделан на автоматизации анализа ГСИ, снижении требований к квалификации пользователя и обеспечении возможности применения методов анализа без необходимости ручного выбора спектральных каналов, настройки сложных моделей и подготовки больших размеченных наборов данных.

В рамках главы была предложена методика формирования информативных спектральных индексов, на основе которой разработаны алгоритмы INDI, MINDI и A-MINDI. Алгоритм INDI предназначен для автоматического формирования информативного нормализованного

разностного индекса в бинарной постановке задачи. Экспериментальная оценка на изображении Salinas Valley показала, что алгоритм способен формировать индексы, пригодные для выделения различных классов объектов в постановке “один против остальных”. Среднее значение метрики F_2 по всем классам составило 0,65, а средняя полнота – 0,81, что подтверждает способность алгоритма обеспечивать высокий охват целевых областей. Также установлено, что постобработка бинарных масок существенно влияет на качество результата: применение медианной фильтрации и морфологических операций позволило увеличить значение F_2 с 0,465 до 0,651, то есть примерно на 40%. Дополнительное сравнение с индексом NDVI на задаче разделения классов “Сельдерей” и “Стерня” показало, что индекс, сформированный INDI, обеспечивает более высокое значение F_2 : 0,79 против 0,73.

Применимость алгоритма INDI к другим предметным задачам подтверждена экспериментами по обнаружению нефтяных загрязнений и очагов лесных пожаров. Для изображения GM1 из набора HOSD получены значения $F_2 = 0,74$, что свидетельствует о высокой полноте обнаружения нефтяных загрязнений и выраженной разделяющей способности сформированного индекса. В задаче обнаружения очагов лесных пожаров на мультиспектральном изображении из набора Active Fire Detection in Landsat-8 Imagery получены значения $F_2 = 0,94$. Это подтвердило возможность применения INDI не только к гиперспектральным, но и к мультиспектральным данным, а также показало эффективность автоматического формирования индексного признака для решения различных задач.

Алгоритм MINDI был разработан как развитие INDI для задачи многоклассовой сегментации ГСИ. Его особенностью является формирование ансамбля информативных индексов с учётом делимости классов. Эксперимент на изображении Salinas Scene показал, что применение MINDI повышает средние значения метрик по сравнению с независимым применением INDI: F_2 увеличилась с 0,58 до 0,62, точность – с 0,53 до 0,57,

полнота – с 0,64 до 0,66. Это подтверждает целесообразность использования критерия отделимости классов при формировании ансамбля индексов.

Алгоритм A-MINDI был предложен как способ автоматического анализа ГСИ с использованием информативных индексов и нейронной сети в условиях отсутствия исходной экспертной разметки. Алгоритм реализует конвейер обработки данных, включающий формирование трёхканального представления изображения методом PCA, получение предварительной разметки с использованием нейронной сети FastSAM и последующее применение MINDI для формирования ансамбля информативных индексов и итоговой карты сегментации. Эксперименты показали, что по метрике ARI предварительная разметка FastSAM в ряде случаев лучше согласуется с эталонной разметкой, чем итоговая карта A-MINDI. Вместе с тем дополнительный эксперимент с обучением модели MLP показал, что разметка, сформированная A-MINDI, в большинстве случаев является более пригодной для обучения модели машинного обучения. Среднее значение F_2 при обучении MLP на разметке Y_{amindi} составило 0,852, тогда как для $Y_{sam} = 0,662$, а для исходной разметки $Y = 0,588$. Это позволяет рассматривать A-MINDI не только как самостоятельный алгоритм построения карты сегментации, но и как средство автоматизированной подготовки обучающих данных.

Кроме алгоритмов формирования информативных индексов, в главе были рассмотрены дополнительные методы анализа ГСИ, необходимые для построения интеллектуальной расширяемой системы. Метод анализа на основе оценки схожести с эталоном показал возможность поиска областей изображения, спектрально близких к заданному образцу. Метод анализа с использованием вегетационных индексов был реализован как набор из 13 распространённых индексов, что позволяет применять интеллектуальную систему для типовых задач анализа растительности. Также был описан метод кластеризации К-средних, предназначенный для разбиения ГСИ на группы пикселей со сходными спектральными характеристиками.

Для обеспечения эффективной работы с большими ГСИ были разработаны методы хранения и обработки данных. Эксперименты показали, что использование специализированных массивов и механизма memmap позволяет существенно снизить время обработки и потребление оперативной памяти по сравнению с базовой реализацией на Python. В сценарии вычисления вегетационного индекса время обработки снизилось с 804 мс. до 0,420 мс. при использовании NumPy memmap, а потребление памяти уменьшилось с 5,13 МБ до 1,46 МБ. В сценарии транспонирования ГСИ время обработки снизилось с 1170 мс до 0,029 мс, а дополнительное потребление памяти было сведено практически к нулю. Эти результаты подтверждают целесообразность использования матрично-векторного представления данных и методов отображения файлов в память при разработке интеллектуальной системы обработки ГСИ.

Метод мозаичной обработки был разработан для обработки ГСИ фрагментами. Экспериментальные результаты показали, что мозаичная обработка позволяет уменьшить время обработки изображения с 132,0 мс до 90,2 мс, при этом среднее время обработки одного патча составило 2,46 мс. Такой подход повышает масштабируемость системы и создаёт основу для обработки изображений большого размера, в том числе при ограничениях по оперативной памяти.

Также в главе были рассмотрены методы предварительной нормализации ГСИ и алгоритм цветосинтеза. Методы нормализации обеспечивают приведение ГСИ к виду, пригодному для последующего анализа, сравнения и применения алгоритмов машинного обучения. Алгоритм цветосинтеза позволяет формировать визуальные представления ГСИ и результатов их обработки, удобные для пользователя, и тем самым повышает интерпретируемость результатов анализа.

Таким образом, в главе разработан комплекс алгоритмов и методов, обеспечивающих основные этапы работы с ГСИ: хранение, предобработку,

визуализацию, вычисление спектральных признаков, формирование информативных индексов, автоматизированный анализ, построение карт сегментации и подготовку данных для обучения моделей машинного обучения. Полученные результаты подтверждают, что предложенные решения позволяют повысить степень автоматизации анализа ГСИ, снизить зависимость от ручного выбора признаков и обеспечить более доступное использование ГСИ в прикладных задачах.

ЗАКЛЮЧЕНИЕ

В диссертационной работе решена научная задача разработки математического и программного обеспечения автоматизированной обработки гиперспектральных изображений, включающего алгоритмы формирования информативных спектральных индексов, алгоритм предварительной разметки данных и архитектуру интеллектуальной системы, обеспечивающей хранение, визуализацию, обработку гиперспектральных данных, выполнение вычислительных задач и интеграцию пользовательских алгоритмов и моделей машинного обучения.

В ходе выполнения исследования получены следующие основные результаты.

Разработана архитектура интеллектуальной системы обработки гиперспектральных изображений, отличающаяся выделением независимых компонентов управления исходными гиперспектральными данными, результатами анализа и визуализации, постановки вычислительных задач и выполнения алгоритмов. Новизна предложенного архитектурного решения заключается в предметно-ориентированном разделении функций системы, что обеспечивает горизонтальное масштабирование её компонентов, асинхронное выполнение ресурсоёмких вычислительных задач и функциональное расширение за счёт интеграции пользовательских алгоритмов и моделей машинного обучения.

На основе предложенной архитектуры реализована интеллектуальная система для хранения, визуализации и автоматизированной обработки гиперспектральных изображений. Система обеспечивает централизованное управление гиперспектральными данными, формирование моделей представления, запуск алгоритмов анализа, сохранение и повторное использование результатов обработки, а также выполнение вычислительных задач в распределённой среде. Возможность расширения системы подтверждена экспериментально: среднее значение метрики МТТА при

интеграции алгоритмов K-means и PCA по результатам 10 запусков составило 1,51 с без учёта первичной загрузки контейнерного образа. При подключении новых алгоритмов значение LoC, характеризующее необходимость изменения существующих компонентов системы, составило 0, время простоя системы также составило 0. Это подтверждает функциональную расширяемость предложенной архитектуры.

Проведено экспериментальное исследование производительности, отказоустойчивости и масштабируемости реализованной интеллектуальной системы. В распределённой среде, состоящей из двух вычислительных узлов, во всех рассмотренных сценариях нагрузочного тестирования отказов компонентов системы не зафиксировано. При независимом тестировании отдельных компонентов максимальная пропускная способность составила 1410,4 запроса в секунду для компонента организации хранения гиперспектральных изображений, 914,6 запроса в секунду для компонента взаимодействия с алгоритмами и 142,8 запроса в секунду для компонента базовой обработки. В режиме, имитирующем поведение реальных пользователей, средняя задержка составила соответственно 45,81 мс, 40,79 мс и 315,94 мс, при этом 99-й перцентиль задержки для компонента базовой обработки не превысил 700 мс. Полученные результаты подтверждают работоспособность, устойчивость и масштабируемость системы в заданной среде тестирования.

Выполнена оценка масштаба разработанной интеллектуальной системы. Без учёта файлов библиотек система включает 874 файла, содержащих 64 191 строку, из которых 54 993 строки относятся непосредственно к программному коду. Полученные значения характеризуют реализованное решение как многокомпонентную интеллектуальную систему, включающую серверную часть, клиентский интерфейс, алгоритмические модули, средства конфигурации, контейнеризации и развёртывания.

Впервые разработан алгоритм INDI для автоматического формирования информативного нормализованного разностного индекса, предназначенного для бинарной сегментации объектов на гиперспектральном изображении. Алгоритм отличается автоматическим отбором пары спектральных полос для выделения целевых объектов, заданных в рамках пользовательской задачи, что позволяет формировать интерпретируемый спектральный признак без ручного выбора спектральных каналов. В проведённом эксперименте для задачи разделения классов «Сельдерей» и «Стерня» значение метрики F_2 составило 0,79, тогда как для эталонного индекса NDVI значение F_2 составило 0,73. Применимость алгоритма к другим предметным задачам подтверждена экспериментами по обнаружению нефтяных загрязнений и очагов лесных пожаров: полученные значения F_2 составили соответственно 0,74 и 0,94, а значения AUC – 0,991 и 0,998. Результаты показывают возможность автоматизированного формирования информативных спектральных индексов для различных задач анализа и наборов данных, включая изображения с различным числом спектральных каналов.

Разработан алгоритм MINDI для формирования ансамбля информативных спектральных индексов, предназначенный для многоклассовой сегментации гиперспектральных изображений. Алгоритм отличается последовательным решением задач «один против остальных» и учётом делимости классов при определении порядка их обработки. Это позволяет использовать набор бинарных информативных индексов для поэтапного разделения нескольких классов гиперспектрального изображения. В проведённом эксперименте среднее значение метрики F_2 увеличилось с 0,58 в случае, когда не учитывалась делимость классов, до 0,62 при использовании MINDI, учитывающего делимость классов. Полученный результат показывает целесообразность согласованного формирования ансамбля информативных индексов при переходе от бинарной к многоклассовой постановке задачи.

Разработан алгоритм A-MINDI для автоматизированного формирования ансамбля информативных спектральных индексов и предварительной разметки гиперспектральных изображений. Алгоритм отличается совместным применением сегментации ГСИ с использованием нейросетевой модели и уточнения полученного результата с использованием информативных индексов, что позволяет уменьшить потребность в ручном формировании эталонной разметки при подготовке данных для последующего обучения моделей машинного обучения. Применимость A-MINDI подтверждена экспериментами с обучением модели многослойного перцептрона: при использовании разметки, сформированной A-MINDI, среднее значение метрики F_2 составило 0,852. Полученный результат показывает возможность использования автоматически сформированной разметки как основы для последующего обучения моделей машинного обучения на гиперспектральных данных.

Таким образом, в диссертационной работе разработаны новые алгоритмы формирования информативных спектральных признаков и предварительной разметки гиперспектральных изображений, а также архитектура интеллектуальной системы, обеспечивающая полный цикл работы с гиперспектральными данными: хранение, визуализацию, автоматизированный анализ, выполнение вычислительных задач, сохранение результатов и интеграцию пользовательских алгоритмов.

Список литературы

- 1 Thomas, N.C. The early history of spectroscopy / N.C. Thomas. – DOI: 10.1021/ed068p631 // Journal of Chemical Education. – 1991. – Т. 68. – № 8. – С. 631.
- 2 Imaging Spectrometry for Earth Remote Sensing / A.F.H. Goetz, G. Vane, J.E. Solomon, B.N. Rock. – DOI: 10.1126/science.228.4704.1147 // Science. – 1985. – Т. 228. – № 4704. – С. 1147-1153.
- 3 Birk, R.J. Airborne hyperspectral sensor systems / R.J. Birk, T.B. McCord. – DOI: 10.1109/62.318881 // IEEE Aerospace and Electronic Systems Magazine. – 1994. – Т. 9. – № 10. – С. 26-33.
- 4 Novel miniaturized hyperspectral sensor for UAV and space applications / H. Saari, V.-V. Aallos, A. Akujärvi [и др.]. – DOI: 10.1117/12.830284 // Sensors, Systems, and Next-Generation Satellites XIII. – SPIE, 2009. – Т. 7474. – С. 74741M.
- 5 Подлипнов, В.В. Компактный изображающий гиперспектрометр / В.В. Подлипнов, Н.А. Ивлиев, Р.В. Скиданов // V Международная конференция и молодёжная школа «Информационные технологии и нанотехнологии» (ИТНТ-2019). – 2019. – С. 438-441.
- 6 Low-Cost Hyperspectral Imaging Device for Portable Remote Sensing / E.M. Pechlivani, A. Papadimitriou, S. Pemas [et al.]. – DOI: 10.3390/instruments7040032 // Instruments. – 2023. – Vol. 7. – No. 4. – P. 32.
- 7 GeoEye-1 Satellite Sensor | Satellite Imaging Corp [Электронный ресурс]. – URL: <https://satimagingcorp.com/satellite-sensor/geoeeye-1/> (дата обращения: 01.06.2026).
- 8 EOS SAT-1 Satellite Building Process: From Design To Launch [Электронный ресурс]. – URL: <https://eos.com/find-satellite/eossat/> (дата обращения: 01.06.2026).
- 9 Guanter, L. Spectral calibration of hyperspectral imagery using atmospheric absorption features / L. Guanter, R. Richter, J. Moreno. – DOI:

- 10.1364/AO.45.002360 // Applied Optics. – 2006. – Т. 45. – № 10. – С. 2360-2370.
- 10 Pichette, J. Fast and compact internal scanning CMOS-based hyperspectral camera: the Snapscan / J. Pichette, W. Charle, A. Lambrechts. – DOI: 10.1117/12.2253614 // Photonic Instrumentation Engineering IV. – SPIE, 2017. – Т. 10110. – С. 1011014.
- 11 EuroCrops v2.0: multi-annual harmonized parcel level crop type data linked to European Union-wide survey, statistical, and Earth Observation products / M. Claverie, A. Chan, L. See [и др.]. – DOI: 10.5194/essd-18-4075-2026 // Earth System Science Data. – 2026. – Т. 18. – № 6. – С. 4075-4095.
- 12 Near Infrared Hyperspectral Image [Электронный ресурс]. – URL: <https://kaggle.com/code/nwheeler443/near-infrared-hyperspectral-image> (дата обращения: 01.06.2026).
- 13 Kemker, R. Algorithms for Semantic Segmentation of Multispectral Remote Sensing Imagery using Deep Learning / R. Kemker, C. Salvaggio, C. Kanan. – DOI: 10.1016/j.isprsjprs.2018.04.014 // ISPRS Journal of Photogrammetry and Remote Sensing. – 2018. – Т. 145. – С. 60-77.
- 14 Hyperspectral Remote Sensing Scenes - Grupo de Inteligencia Computacional (GIC) [Электронный ресурс]. – URL: https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes (дата обращения: 01.06.2026).
- 15 Guo, A.J.X. Improving Deep Hyperspectral Image Classification Performance with Spectral Unmixing / A.J.X. Guo, F. Zhu. – arXiv, 2020.
- 16 Yuan, Y. Proxy-Based Deep Learning Framework for Spectral–Spatial Hyperspectral Image Classification: Efficient and Robust / Y. Yuan, C. Wang, Z. Jiang. – DOI: 10.1109/TGRS.2021.3054008 // IEEE Transactions on Geoscience and Remote Sensing. – 2022. – Т. 60. – Proxy-Based Deep Learning Framework for Spectral–Spatial Hyperspectral Image Classification. – С. 1-15.

- 17 Efficient Graph Convolutional Self-Representation for Band Selection of Hyperspectral Image / Y. Cai, Z. Zhang, X. Liu, Z. Cai. – DOI: 10.1109/JSTARS.2020.3018229 // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. – 2020. – Т. 13. – С. 4869-4880.
- 18 Efficient Deep Learning of Nonlocal Features for Hyperspectral Image Classification / Y. Shen, S. Zhu, C. Chen [и др.]. – DOI: 10.1109/TGRS.2020.3014286 // IEEE Transactions on Geoscience and Remote Sensing. – 2021. – Т. 59. – № 7. – С. 6029-6043.
- 19 Hyperspectral Images Classification With Gabor Filtering and Convolutional Neural Network / Y. Chen, L. Zhu, P. Ghamisi [и др.]. – DOI: 10.1109/LGRS.2017.2764915 // IEEE Geoscience and Remote Sensing Letters. – 2017. – Т. 14. – № 12. – С. 2355-2359.
- 20 Prediktera – Hyperspectral Imaging Software [Электронный ресурс]. – URL: <https://prediktera.com/> (дата обращения: 01.01.2024).
- 21 ClydeHSI | Hyperspectral Imaging Cameras and Systems [Электронный ресурс]. – URL: <https://www.clydehsi.com/> (дата обращения: 01.06.2026).
- 22 The airborne visible/infrared imaging spectrometer (AVIRIS) / G. Vane, R.O. Green, T.G. Chrien [и др.]. – DOI: 10.1016/0034-4257(93)90012-m // Remote sensing of environment. – 1993. – Т. 44. – № 2-3. – С. 127-143.
- 23 Gupta, N. Hyperspectral imager, from ultraviolet to visible, with a KDP acousto-optic tunable filter / N. Gupta, V. Voloshinov. – DOI: 10.1364/ao.43.002752 // Applied Optics. – 2004. – Т. 43. – № 13. – С. 2752.
- 24 Application of hyperspectral imaging for prediction of physico-chemical and sensory characteristics of table grapes / A. Baiano, C. Terracone, G. Peri, R. Romaniello. – DOI: 10.1016/j.compag.2012.06.002 // Computers and Electronics in Agriculture. – 2012. – Т. 87. – С. 142-151.
- 25 Non-Destructive Testing of Composite Fiber Materials With Hyperspectral Imaging—Evaluative Studies in the EU H2020 FibreEUUse Project / Y. Yan, J.

- Ren, H. Zhao [и др.]. – DOI: 10.1109/tim.2022.3155745 // IEEE Transactions on Instrumentation and Measurement. – 2022. – Т. 71. – С. 1-13.
- 26 Polder, G. The hype in spectral imaging / G. Polder, A. Gowen. – DOI: 10.1255/sew.2021.a12 // Spectroscopy Europe. – 2021. – С. 12.
- 27 Lu, G. Medical hyperspectral imaging: a review / G. Lu, B. Fei. – DOI: 10.1117/1.jbo.19.1.010901 // Journal of Biomedical Optics. – 2014. – Т. 19. – № 1. – С. 010901.
- 28 Hagen, N. Snapshot advantage: a review of the light collection improvement for parallel high-dimensional measurement systems / N. Hagen. – DOI: 10.1117/1.oe.51.11.111702 // Optical Engineering. – 2012. – Т. 51. – № 11. – С. 111702.
- 29 Multivariate Hyperspectral Raman Imaging Using Compressive Detection / B.M. Davis, A.J. Hemphill, Derya Cebeci Maltaş [и др.]. – DOI: 10.1021/ac103259v // Analytical Chemistry. – 2011. – Т. 83. – № 13. – С. 5086-5092.
- 30 Benelli, A. In-field hyperspectral imaging: An overview on the ground-based applications in agriculture / A. Benelli, C. Cevoli, A. Fabbri. – DOI: 10.4081/jae.2020.1030 // Journal of Agricultural Engineering. – 2020. – Т. 51. – № 3. – С. 129-139.
- 31 Модели Формирования И Некоторые Алгоритмы Обработки Гиперспектральных Изображений / Р.Н. Ахметов, Н.Р. Стратилатов, А.А. Юдаков [и др.]. – DOI: 10.7868/S0205961414010011 // Исследование Земли Из Космоса. – 2014. – № 1. – С. 17.
- 32 A compact hyperspectral camera for measurement of perfusion parameters in medicine / Axel Kulcke, A. Holmer, P. Wahl [и др.]. – DOI: 10.1515/bmt-2017-0145 // Biomedizinische Technik. – 2018. – Т. 63. – № 5. – С. 519-527.
- 33 Pichette, J. Fast and compact internal scanning CMOS-based hyperspectral camera: the Snapscan / J. Pichette, W. Charle, A. Lambrechts. – DOI: 10.1117/12.2253614 // Proceedings of SPIE. – 2017.

- 34 Fifty years of Landsat science and impacts / M.A. Wulder, D.P. Roy, V.C. Radeloff [и др.]. – DOI: 10.1016/j.rse.2022.113195 // Remote Sensing of Environment. – 2022. – Т. 280. – С. 113195.
- 35 Sentinel-2 Data for Land Cover/Use Mapping: A Review / D. Phiri, M. Simwanda, S. Salekin [et al.]. – DOI: 10.3390/rs12142291 // Remote Sensing. – 2020. – Vol. 12. – Sentinel-2 Data for Land Cover/Use Mapping. – No. 14. – P. 2291.
- 36 UAV Multispectral Imaging Potential to Monitor and Predict Agronomic Characteristics of Different Forage Associations / J. Plaza, M. Criado, N. Sánchez [et al.]. – DOI: 10.3390/agronomy11091697 // Agronomy. – 2021. – Vol. 11. – No. 9. – P. 1697.
- 37 A low-cost multispectral imaging system for the characterisation of soil and small vegetation properties using visible and near-infrared reflectance / S. Orlando, M. Minacapilli, M. Sarno [и др.]. – DOI: 10.1016/j.compag.2022.107359 // Computers and Electronics in Agriculture. – 2022. – Т. 202. – С. 107359.
- 38 Monitoring of peanut leaves chlorophyll content based on drone-based multispectral image feature extraction / H. Qi, Z. Wu, L. Zhang [и др.]. – DOI: 10.1016/j.compag.2021.106292 // Computers and Electronics in Agriculture. – 2021. – Т. 187. – С. 106292.
- 39 Fine-scale prediction of biomass and leaf nitrogen content in sugarcane using UAV LiDAR and multispectral imaging / Y. Shendryk, J. Sofonia, R. Garrard [и др.]. – DOI: 10.1016/j.jag.2020.102177 // International Journal of Applied Earth Observation and Geoinformation. – 2020. – Т. 92. – С. 102177.
- 40 Soil Moisture Analysis by Means of Multispectral Images According to Land Use and Spatial Resolution on Andosols in the Colombian Andes / M. Casamitjana, M.C. Torres-Madroño, J. Bernal-Riobo, D. Varga. – DOI: 10.3390/app10165540 // Applied Sciences. – 2020. – Т. 10. – № 16.

- 41 Comparison of VIS/NIR spectral curves plus RGB images with hyperspectral images for the identification of *Pterocarpus* species / C.-K. Wang, P. Zhao, Z.-Y. Li, X.-H. Li. – DOI: 10.1515/hf-2021-0194 // *Holzforschung*. – 2022. – T. 76.
- 42 Hyperspectral imaging to classify and monitor quality of agricultural materials / S. Mahesh, D.S. Jayas, J. Paliwal, N.D.G. White. – DOI: 10.1016/j.jspr.2015.01.006 // *Journal of Stored Products Research*. – 2015. – T. 61. – C. 17-26.
- 43 Estimation of the spatial distribution of heavy metal in agricultural soils using airborne hyperspectral imaging and random forest / K. Tan, H. Wang, L. Chen [и др.]. – DOI: 10.1016/j.jhazmat.2019.120987 // *Journal of Hazardous Materials*. – 2020. – T. 382. – C. 120987.
- 44 Hyperspectral Images-Based Crop Classification Scheme for Agricultural Remote Sensing / I. Ali, Z. Mushtaq, S. Arif [и др.]. – DOI: 10.32604/csse.2023.034374 // *Computer Systems Science and Engineering*. – 2023. – T. 46. – C. 303-319.
- 45 Above-ground biomass estimation and yield prediction in potato by using UAV-based RGB and hyperspectral imaging / B. Li, X. Xu, L. Zhang [и др.]. – DOI: 10.1016/j.isprsjprs.2020.02.013 // *ISPRS Journal of Photogrammetry and Remote Sensing*. – 2020. – T. 162. – C. 161-172.
- 46 Application of a hyperspectral imaging system to quantify leaf-scale chlorophyll, nitrogen and chlorophyll fluorescence parameters in grapevine / Z. Yang, J. Tian, K. Feng [и др.]. – DOI: 10.1016/j.plaphy.2021.06.015 // *Plant Physiology and Biochemistry*. – 2021. – T. 166. – C. 723-737.
- 47 Lee, D.-H. Developing a p-NDVI Map for Highland Kimchi Cabbage Using Spectral Information from UAVs and a Field Spectral Radiometer / D.-H. Lee, H.-S. Shin, J.-H. Park. – DOI: 10.3390/agronomy10111798 // *Agronomy*. – 2020. – T. 10. – № 11.

- 48 Compact, UAV-mounted hyperspectral imaging system with automatic geometric distortion rectification / X. Qingsheng, B. Yang, W. Fupeng [и др.]. – DOI: 10.1364/OE.412853 // Optics Express. – 2021. – Т. 29. – С. 6092-6112.
- 49 Raspberry plant stress detection using hyperspectral imaging / D. Williams, A. Karley, A. Britten [et al.]. – DOI: 10.1002/pld3.490 // Plant Direct. – 2023. – Vol. 7. – No. 3. – P. e490.
- 50 Hyperspectral imagery to monitor crop nutrient status within and across growing seasons / N. Liu, P.A. Townsend, M.R. Naber [и др.]. – DOI: 10.1016/j.rse.2021.112303 // Remote Sensing of Environment. – 2021. – Т. 255. – С. 112303.
- 51 Leveraging high-throughput hyperspectral imaging technology to detect cadmium stress in two leafy green crops and accelerate soil remediation efforts / M. Zea, A. Souza, Y. Yang [и др.]. – DOI: 10.1016/j.envpol.2021.118405 // Environmental Pollution. – 2022. – Т. 292. – С. 118405.
- 52 Modzelewska, A. Tree species identification within an extensive forest area with diverse management regimes using airborne hyperspectral data / A. Modzelewska, F.E. Fassnacht, K. Stereńczak. – DOI: 10.1016/j.jag.2019.101960 // International Journal of Applied Earth Observation and Geoinformation. – 2020. – Т. 84. – С. 101960.
- 53 Autonomous Satellite Wildfire Detection Using Hyperspectral Imagery and Neural Networks: A Case Study on Australian Wildfire / K. Thangavel, D. Spiller, R. Sabatini [и др.]. – DOI: 10.3390/rs15030720 // Remote Sensing. – 2023. – Т. 15. – № 3.
- 54 Peyghambari, S. Hyperspectral remote sensing in lithological mapping, mineral exploration, and environmental geology: An updated review / S. Peyghambari, Y. Zhang. – DOI: 10.1117/1.JRS.15.031501 // Journal of Applied Remote Sensing. – 2021. – Т. 15. – Hyperspectral remote sensing in lithological mapping, mineral exploration, and environmental geology.

- 55 Automated Identification of Mineral Types and Grain Size Using Hyperspectral Imaging and Deep Learning for Mineral Processing / N. Okada, Y. Maekawa, N. Owada [et al.]. – DOI: 10.3390/min10090809 // Minerals. – 2020. – Vol. 10. – No. 9. – P. 809.
- 56 Hyperspectral Band Selection for Lithologic Discrimination and Geological Mapping / Y. Tan, L. Lu, L. Bruzzone [и др.]. – DOI: 10.1109/JSTARS.2020.2964000 // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. – 2020. – Т. 13. – С. 471-486.
- 57 Soydan, H. Secondary Iron Mineral Detection via Hyperspectral Unmixing Analysis with Sentinel-2 Imagery / H. Soydan, A. Koz, H.Ş. Düzgün. – DOI: 10.1016/j.jag.2021.102343 // International Journal of Applied Earth Observation and Geoinformation. – 2021. – Т. 101. – С. 102343.
- 58 Extending geometallurgy to the mine scale with hyperspectral imaging: a pilot study using drone- and ground-based scanning / I. Barton, M. Gabriel, J. Lyons-Baral [и др.]. – DOI: 10.1007/s42461-021-00404-z // Mining, Metallurgy & Exploration. – 2021. – Т. 38.
- 59 Detecting the effects of hydrocarbon pollution in the Amazon forest using hyperspectral satellite images / P. Arellano, K. Tansey, H. Balzter, D.S. Boyd. – DOI: 10.1016/j.envpol.2015.05.041 // Environmental Pollution. – 2015. – Т. 205. – С. 225-239.
- 60 Monitoring Asbestos Mine Remediation Using Airborne Hyperspectral Imaging System: A Case Study of Jefferson Lake Mine, US / Y. Jeong, J. Yu, L. Wang [и др.]. – DOI: 10.3390/rs14215572 // Remote Sensing. – 2022. – Т. 14. – № 21.
- 61 Spaceborne Mine Waste Mineralogy Monitoring in South Africa, Applications for Modern Push-Broom Missions: Hyperion/OLI and EnMAP/Sentinel-2 / C. Mielke, N.K. Boesche, C. Rogass [et al.]. – DOI: 10.3390/rs6086790 // Remote Sensing. – 2014. – Vol. 6. – Spaceborne Mine

Waste Mineralogy Monitoring in South Africa, Applications for Modern Push-Broom Missions. – No. 8. – P. 6790-6816.

62 Davies, G.E. Mapping acidic mine waste with seasonal airborne hyperspectral imagery at varying spatial scales / G.E. Davies, W.M. Calvin. – DOI: 10.1007/s12665-017-6763-x // Environmental Earth Sciences. – 2017. – Vol. 76. – No. 12. – P. 432.

63 Using imaging spectroscopy to map acidic mine waste / G.A. Swayze, K.S. Smith, R.N. Clark [et al.]. – DOI: 10.1021/es990046w // Environmental Science & Technology. – 2000. – Vol. 34. – No. 1. – P. 47-54.

64 Landslide detection, monitoring and prediction with remote-sensing techniques / N. Casagli, E. Intrieri, V. Tofani [et al.]. – DOI: 10.1038/s43017-022-00373-x // Nature Reviews Earth & Environment. – 2023. – Vol. 4. – No. 1. – P. 51-64.

65 Hyperspectral imaging as a new diagnostic tool for cervical intraepithelial neoplasia / L. Schimunek, K. Schöpp, M. Wagner [et al.]. – DOI: 10.1007/s00404-023-07171-w // Archives of Gynecology and Obstetrics. – 2023. – Vol. 308. – No. 5. – P. 1525-1530.

66 Non-Invasive Skin Cancer Diagnosis Using Hyperspectral Imaging for In-Situ Clinical Support / R. Leon, B. Martinez-Vega, H. Fabelo [и др.]. – DOI: 10.3390/jcm9061662 // Journal of Clinical Medicine. – 2020. – Т. 9. – № 6. – C. 1662.

67 Oxygen saturation measurements using novel diffused reflectance with hyperspectral imaging: Towards facile COVID-19 diagnosis / Y.H. El-Sharkawy, M.H. Aref, S. Elbasuney [et al.]. – DOI: 10.1007/s11082-022-03658-z // Optical and Quantum Electronics. – 2022. – Vol. 54. – Oxygen saturation measurements using novel diffused reflectance with hyperspectral imaging. – No. 5. – P. 322.

68 Imaging of peripheral vascular malformations — current concepts and future perspectives / V.F. Schmidt, M. Masthoff, M. Czihal [et al.]. – DOI:

10.1186/s40348-021-00132-w // Molecular and Cellular Pediatrics. – 2021. – Vol. 8. – No. 1. – P. 19.

69 A 3D attention networks for classification of white blood cells from microscopy hyperspectral images / Q. Wang, J. Wang, M. Zhou [и др.]. – DOI: 10.1016/j.optlastec.2021.106931 // Optics & Laser Technology. – 2021. – Т. 139. – C. 106931.

70 Zakian, C. Near-infrared hyperspectral imaging of teeth for dental caries detection / C. Zakian, I. Pretty, R. Ellwood. – DOI: 10.1117/1.3275480 // Journal of Biomedical Optics. – 2009. – Т. 14. – № 6. – C. 064047.

71 Hyperspectral Image Segmentation: A Preliminary Study on the Oral and Dental Spectral Image Database (ODSI-DB) / L.C. Garcia-Peraza-Herrera, C. Horgan, S. Ourselin [и др.]. – DOI: 10.1080/21681163.2022.2160377 // Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization. – 2023. – Т. 11. – Hyperspectral Image Segmentation. – № 4. – C. 1290-1298.

72 Application of hyperspectral imaging systems and artificial intelligence for quality assessment of fruit, vegetables and mushrooms: A review / J. Wieme, K. Mollazade, I. Malounas [и др.]. – DOI: 10.1016/j.biosystemseng.2022.07.013 // Biosystems Engineering. – 2022. – Т. 222. – Application of hyperspectral imaging systems and artificial intelligence for quality assessment of fruit, vegetables and mushrooms. – C. 156-176.

73 Determination of hardness for maize kernels based on hyperspectral imaging / M. Qiao, Y. Xu, G. Xia [и др.]. – DOI: 10.1016/j.foodchem.2021.130559 // Food Chemistry. – 2022. – Т. 366. – C. 130559.

74 Detection of cold injury in peaches by hyperspectral reflectance imaging and artificial neural network / L. Pan, Q. Zhang, W. Zhang [и др.]. – DOI: 10.1016/j.foodchem.2015.06.106 // Food Chemistry. – 2016. – Т. 192. – C. 134-141.

- 75 Detection of Honey Adulteration using Hyperspectral Imaging : 5th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 2016 / S. Shafiee, G. Polder, S. Minaei [и др.]. – DOI: 10.1016/j.ifacol.2016.10.057 // IFAC-PapersOnLine. – 2016. – Т. 49. – № 16. – С. 311-314.
- 76 Detection of insect-damaged wheat kernels using near-infrared hyperspectral imaging / C.B. Singh, D.S. Jayas, J. Paliwal, N.D.G. White. – DOI: 10.1016/j.jspr.2008.12.002 // Journal of Stored Products Research. – 2009. – Т. 45. – № 3. – С. 151-158.
- 77 Identification of Apple Varieties Using a Multichannel Hyperspectral Imaging System / Y. Huang, Y. Yang, Y. Sun [и др.]. – DOI: 10.3390/s20185120 // Sensors. – 2020. – Т. 20. – № 18.
- 78 Hyperspectral Imaging (HSI) for meat quality evaluation across the supply chain: Current and future trends / W. Jia, S. van Ruth, N. Scollan, A. Koidis. – DOI: 10.1016/j.crfs.2022.05.016 // Current Research in Food Science. – 2022. – Т. 5. – Hyperspectral Imaging (HSI) for meat quality evaluation across the supply chain. – С. 1017-1027.
- 79 Predicting TFe content and sorting iron ores from hyperspectral image by variational mode decomposition-based spectral feature / C. Nie, J. Jiang, J. Deng [и др.]. – DOI: 10.1016/j.jclepro.2023.139629 // Journal of Cleaner Production. – 2023. – Т. 429. – С. 139629.
- 80 Feasibility Study of Ore Classification Using Active Hyperspectral LiDAR / Y. Chen, C. Jiang, J. Hyypä [и др.]. – DOI: 10.1109/LGRS.2018.2854358 // IEEE Geoscience and Remote Sensing Letters. – 2018. – Т. 15. – № 11. – С. 1785-1789.
- 81 HYPER-Tools [Электронный ресурс]. – URL: <https://www.hypertools.org> (дата обращения: 19.03.2024).

- 82 MATLAB [Электронный ресурс]. – URL: <https://www.mathworks.com/products/matlab.html> (дата обращения: 19.03.2024).
- 83 Redmon, J. YOLOv3: An Incremental Improvement. YOLOv3 / J. Redmon, A. Farhadi. – arXiv, 2018.
- 84 Girshick, R.B. Fast R-CNN / R.B. Girshick // CoRR. – 2015. – T. abs/1504.08083.
- 85 Hyperspectral Imaging Software | ClydeHSI [Электронный ресурс]. – URL: <https://www.clydehsi.com/hyperspectral-imaging-software-spectrasens> (дата обращения: 19.03.2024).
- 86 Development of an online indices database: Motivation, concept and implementation / V. Henrich, C. Götze, A. Jung [и др.]. – 2009.
- 87 Estimating Absorbed Photosynthetic Radiation and Leaf Area Index from Spectral Reflectance in Wheat 1 / G. Asrar, M. Fuchs, E.T. Kanemasu, J.L. Hatfield. – DOI: 10.2134/agronj1984.00021962007600020029x // Agronomy Journal. – 1984. – T. 76. – № 2. – С. 300-306.
- 88 Hasan, H. A Comparison Between Support Vector Machine (SVM) and Convolutional Neural Network (CNN) Models For Hyperspectral Image Classification / H. Hasan, H.Z.M. Shafri, M. Habshi. – DOI: 10.1088/1755-1315/357/1/012035 // IOP Conference Series: Earth and Environmental Science. – 2019. – Vol. 357. – No. 1. – P. 012035.
- 89 Liu, D. Estimating wheat fractional vegetation cover using a density peak k-means algorithm based on hyperspectral image data / D. Liu, F. Yang, S. Liu. – DOI: 10.1016/S2095-3119(20)63556-0 // Journal of Integrative Agriculture. – 2021. – T. 20. – № 11. – С. 2880-2891.
- 90 Ren, Z. Improved k-means and spectral matching for hyperspectral mineral mapping / Z. Ren, L. Sun, Q. Zhai. – DOI: 10.1016/j.jag.2020.102154 // International Journal of Applied Earth Observation and Geoinformation. – 2020. – T. 91. – С. 102154.

- 91 Hyperspectral Image Classification With Convolutional Neural Network and Active Learning / X. Cao, J. Yao, Z. Xu, D. Meng. – DOI: 10.1109/TGRS.2020.2964627 // IEEE Transactions on Geoscience and Remote Sensing. – 2020. – T. 58. – № 7. – C. 4604-4616.
- 92 Shi, C. Multiscale Superpixel-Based Hyperspectral Image Classification Using Recurrent Neural Networks With Stacked Autoencoders / C. Shi, C.-M. Pun. – DOI: 10.1109/TMM.2019.2928491 // IEEE Transactions on Multimedia. – 2020. – T. 22. – № 2. – C. 487-501.
- 93 Deep hybrid: Multi-graph neural network collaboration for hyperspectral image classification / D. Yao, Z. Zhi-li, Z. Xiao-feng [и др.]. – DOI: 10.1016/j.dt.2022.02.007 // Defence Technology. – 2023. – T. 23. – Deep hybrid. – C. 164-176.
- 94 Multimodal GANs: Toward Crossmodal Hyperspectral–Multispectral Image Segmentation / D. Hong, J. Yao, D. Meng [и др.]. – DOI: 10.1109/TGRS.2020.3020823 // IEEE Transactions on Geoscience and Remote Sensing. – 2021. – T. 59. – Multimodal GANs. – № 6. – C. 5103-5113.
- 95 A Spectral-Spatial Cascaded 3D Convolutional Neural Network with a Convolutional Long Short-Term Memory Network for Hyperspectral Image Classification / W. Qi, X. Zhang, N. Wang [и др.]. – DOI: 10.3390/rs11202363 // Remote Sensing. – 2019. – T. 11. – № 20.
- 96 Hyperspectral Image Classification Using a Hybrid 3D-2D Convolutional Neural Networks / S. Ghaderizadeh, D. Abbasi-Moghadam, A. Sharifi [и др.]. – DOI: 10.1109/JSTARS.2021.3099118 // IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing. – 2021. – T. PP. – C. 1-1.
- 97 Alotaibi, B. A Hybrid Deep ResNet and Inception Model for Hyperspectral Image Classification / B. Alotaibi, M. Alotaibi. – DOI: 10.1007/s41064-020-00124-x // PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science. – 2020. – Vol. 88. – No. 6. – P. 463-476.

- 98 Haque, Md.R. Spectral-Spatial Feature Extraction Using PCA and Multi-Scale Deep Convolutional Neural Network for Hyperspectral Image Classification / Md.R. Haque, S.Z. Mishu. – DOI: 10.1109/ICCIT48885.2019.9038385 // 2019 22nd International Conference on Computer and Information Technology (ICCIT) 2019 22nd International Conference on Computer and Information Technology (ICCIT). – 2019. – C. 1-6.
- 99 Hyperspectral Band Selection Based on Deep Convolutional Neural Network and Distance Density / Y. Zhan, D. Hu, H. Xing, X. Yu. – DOI: 10.1109/LGRS.2017.2765339 // IEEE Geoscience and Remote Sensing Letters. – 2017. – T. 14. – № 12. – C. 2365-2369.
- 100 Hyperspectral Band Selection Using Attention-Based Convolutional Neural Networks / P. Ribalta Lorenzo, L. Tulczyjew, M. Marcinkiewicz, J. Nalepa. – DOI: 10.1109/ACCESS.2020.2977454 // IEEE Access. – 2020. – T. 8. – C. 42384-42403.
- 101 A New Statistical Approach for Band Clustering and Band Selection Using K-Means Clustering / M. Ahmad, Dr.I. Ulhaq, Q. Mushtaq, M. Sohaib // International Journal of Engineering and Technology. – 2011. – T. 3. – C. 606-614.
- 102 An effective feature selection method for hyperspectral image classification based on genetic algorithm and support vector machine / S. Li, H. Wu, D. Wan, J. Zhu. – DOI: 10.1016/j.knosys.2010.07.003 // Knowledge-Based Systems. – 2011. – T. 24. – № 1. – C. 40-48.
- 103 Datta, A. Clustering based band selection for hyperspectral images / A. Datta, S. Ghosh, A. Ghosh. – DOI: 10.1109/CODIS.2012.6422146 // 2012 International Conference on Communications, Devices and Intelligent Systems (CODIS) 2012 International Conference on Communications, Devices and Intelligent Systems (CODIS). – 2012. – C. 101-104.

- 104 Blinowski, G. Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation / G. Blinowski, A. Ojdowska, A. Przybyłek. – DOI: 10.1109/ACCESS.2022.3152803 // IEEE Access. – 2022. – T. 10. – Monolithic vs. Microservice Architecture. – C. 20357-20374.
- 105 Haorongbam, L. Service Oriented Architecture(SOA): A Literature Review on the Maintainability, Approaches and Design Process / L. Haorongbam, R. Nagpal, R. Sehgal. – DOI: 10.1109/Confluence52989.2022.9734153 // 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence) 2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence). – 2022. – Service Oriented Architecture(SOA). – C. 647-652.
- 106 Manchana, R. Event-Driven Architecture: Building Responsive and Scalable Systems for Modern Industries / R. Manchana. – DOI: 10.21275/SR24820051042 // International Journal of Science and Research (IJSR). – 2021. – Vol. 10. – Event-Driven Architecture. – No. 1. – P. 1706-1716.
- 107 Hazarika, A.V. Serverless Architectures: Implications for Distributed System Design and Implementation / A.V. Hazarika, M. Shah. – DOI: 10.21275/SR241216094817 // International Journal of Science and Research (IJSR). – 2024. – Vol. 13. – Serverless Architectures. – No. 12. – P. 1250-1253.
- 108 Poniszewska-Marańda, A. Kubernetes Cluster for Automating Software Production Environment / A. Poniszewska-Marańda, E. Czechowska. – DOI: 10.3390/s21051910 // Sensors. – 2021. – Vol. 21. – No. 5. – P. 1910.
- 109 Carrión, C. Kubernetes as a Standard Container Orchestrator - A Bibliometric Analysis / C. Carrión. – DOI: 10.1007/s10723-022-09629-8 // Journal of Grid Computing. – 2022. – Vol. 20. – No. 4. – P. 42.
- 110 A comparative analysis of big data processing paradigms: Mapreduce vs. apache spark / S. Ibtisum, E. Bazgir, S.M.A. Rahman, S.M.S. Hossain. – DOI: 10.30574/wjarr.2023.20.1.2174 // World Journal of Advanced Research and

Reviews. – 2023. – Vol. 20. – A comparative analysis of big data processing paradigms. – No. 1. – P. 1089-1098.

111 Ray: a distributed framework for emerging AI applications / P. Moritz, R. Nishihara, S. Wang [и др.] // Proceedings of the 13th USENIX Conference on Operating Systems Design and Implementation : OSDI'18. – USA: USENIX Association, 2018. – C. 561-577.

112 Boettiger, C. An introduction to Docker for reproducible research / C. Boettiger. – DOI: 10.1145/2723872.2723882 // ACM SIGOPS Operating Systems Review. – 2015. – T. 49. – № 1. – C. 71-79.

113 Beetz, F. GitOps: The Evolution of DevOps? / F. Beetz, S. Harrer. – DOI: 10.1109/MS.2021.3119106 // IEEE Software. – 2022. – T. 39. – GitOps. – № 4. – C. 70-75.

114 Asaithambi, S.P.R. MOBDA: Microservice-Oriented Big Data Architecture for Smart City Transport Systems / S.P.R. Asaithambi, R. Venkatraman, S. Venkatraman. – DOI: 10.3390/bdcc4030017 // Big Data and Cognitive Computing. – 2020. – T. 4. – № 3.

115 Ben Atitallah, S. FedMicro-IDA: A federated learning and microservices-based framework for IoT data analytics / S. Ben Atitallah, M. Driss, H. Ben Ghezala. – DOI: 10.1016/j.iot.2023.100845 // Internet of Things. – 2023. – T. 23. – FedMicro-IDA. – C. 100845.

116 CLIC: An Extensible and Efficient Cross-Platform Data Analytics System / Q. Chen, Z. Chen, K. Zhang, X.S. Wang. – DOI: 10.1109/TPDS.2023.3298038 // IEEE Transactions on Parallel and Distributed Systems. – 2024. – T. 35. – CLIC. – № 1. – C. 34-45.

117 The C4 model for visualising software architecture [Электронный ресурс]. – URL: <https://c4model.com/> (дата обращения: 20.03.2024).

118 Мухин, А.В. Применение нейросетевых алгоритмов для решения задачи семантической сегментации гиперспектральных данных / А.В.

- Мухин, Д.Н. Грибанов, Р.А. Парингер // Информационные технологии и нанотехнологии (ИТНТ-2022). - Т. 4 : Искусственный интеллект. – 2022.
- 119 Mukhin, A. Semantic Segmentation of Hyperspectral Imaging Using Convolutional Neural Networks / A. Mukhin, G. Danil, R. Paringer. – DOI: 10.3103/S1060992X22050071 // Optical Memory and Neural Networks. – 2022. – Vol. 31. – No. 1. – P. 38-47.
- 120 Свидетельство о государственной регистрации программы для ЭВМ № 2022665368. Модуль обучения нейросетевого дескриптора для анализа мультиспектральных данных: заявлено 29.06.2022: опубликовано 15.08.2022 / Мухин А.В., Парингер Р.А., Грибанов Д.Н., Килбас И.А.; заявитель Самарский Университет. – 1 с.
- 121 Свидетельство о государственной регистрации программы для ЭВМ № 2023610529. Модуль исследования применимости методов машинного обучения для обработки гиперспектральных данных: заявлено 15.12.2022: опубликовано 01.11.2023 / Мухин А.В., Грибанов Д.Н., Парингер Р.А., Килбас И.А.; заявитель Самарский Университет. – 1 с.
- 122 Мухин, А.В. vidis-algorithms-api: Фреймворк реализации алгоритмов анализа спектральных изображений [Электронный ресурс]. – URL: <https://github.com/Banayaki> (дата обращения: 01.06.2026).
- 123 K3s - Lightweight Kubernetes | K3s [Электронный ресурс]. – URL: <https://docs.k3s.io/> (дата обращения: 07.06.2025).
- 124 Longhorn | The Longhorn Documentation [Электронный ресурс]. – URL: <https://longhorn.io/docs/1.9.0/> (дата обращения: 07.06.2025).
- 125 Мухин, А.В. Разработка расширяемой микросервисной архитектуры программного комплекса анализа данных / А.В. Мухин. – DOI: 10.14357/20718632250307 // Информационные Технологии И Вычислительные Системы. – 2025. – № 3. – С. 73-85.
- 126 Свидетельство о государственной регистрации программы для ЭВМ № 2021663888. Облачный сервис анализа гиперкубов

hyperspectrum.copter.space: заявлено 18.06.2021: опубликовано 25.08.2021 / Парингер Р.А., Мухин А.В.; заявитель Общество с ограниченной ответственностью «Байт-Самара». – 1 с.

127 Свидетельство о государственной регистрации программы для ЭВМ № 2024615474. Открытая платформа для обработки и анализа спектральных изображений: заявлено 26.02.2024: опубликовано 03.07.2024 / Мухин А.В.; заявитель Мухин Артем Владимирович. – 1 с.

128 IDB - Index DataBase [Электронный ресурс]. – URL: <https://www.indexdatabase.de/> (дата обращения: 24.08.2026).

129 Wang, C. Construction of Remote Sensing Indices Knowledge Graph (RSIKG) Based on Semantic Hierarchical Graph / C. Wang, W. Shi, H. Lv. – DOI: 10.3390/rs16010158 // Remote Sensing. – 2024. – Т. 16. – № 1.

130 Парингер, Р.А. Формирование информативного индекса для различения заданных объектов гиперспектральных данных / Р.А. Парингер, А.В. Мухин, А.В. Куприянов. – DOI: 10.18287/2412-6179-CO-930 // Компьютерная Оптика. – 2021. – Т. 45. – № 6. – С. 873-878.

131 Hyperspectral Remote Sensing Benchmark Database for Oil Spill Detection With an Isolation Forest-Guided Unsupervised Detector / P. Duan, X. Kang, P. Ghamisi, S. Li. – DOI: 10.1109/TGRS.2023.3268944 // IEEE Transactions on Geoscience and Remote Sensing. – 2023. – Т. 61. – С. 1-11.

132 Active fire detection in Landsat-8 imagery: A large-scale dataset and a deep-learning study / G.H. de Almeida Pereira, A.M. Fusioka, B.T. Nassu, R. Minetto. – DOI: 10.1016/j.isprsjprs.2021.06.002 // ISPRS Journal of Photogrammetry and Remote Sensing. – 2021. – Т. 178. – Active fire detection in Landsat-8 imagery. – С. 171-186.

133 Свидетельство о государственной регистрации программы для ЭВМ № 2020666922. Модуль формирования информативных индексов для различения заданных объектов гиперспектральных данных: заявлено

- 10.12.2020: опубликовано 17.12.2020 / Мухин А.В., Парингер Р.А., Куприянов А.В.; заявитель Самарский Университет. – 1 с.
- 134 Свидетельство о государственной регистрации программы для ЭВМ № 2023680048. Программный модуль оценки информативности и отбора признаков, формирующих эффективное признаковое пространство: заявлено 19.09.2023: опубликовано 25.09.2023 / Мухин А.В., Парингер Р.А., Куприянов А.В.; заявитель Самарский Университет. – 1 с.
- 135 Mukhin, A. Feature selection algorithm with feature space separability estimation using discriminant analysis / A. Mukhin, R. Paringer, N. Ilyasova. – DOI: 10.1109/itnt52450.2021.9649144 // 2021 International Conference on Information Technology and Nanotechnology (ITNT). – 2021. – С. 1-4.
- 136 Мухин, А.В. Алгоритм отбора информативных признаков на основе оценки разделяющей способности пространства признаков с использованием дискриминантного анализа / А.В. Мухин, Р.А. Парингер, Н.Ю. Ильясова. – VII Международная конференция и молодёжная школа «Информационные технологии и нанотехнологии» (ИТНТ-2021), 2021. – Т. 3. – С. 30162.
- 137 Свидетельство о государственной регистрации программы для ЭВМ № 2021669454. Модуль итерационного составления набора информативных признаков для различения множества классов учитывающий отделимость классов в пространстве признаков: заявлено 17.11.2021: опубликовано 29.11.2021 / Мухин А.В., Парингер Р.А., Куприянов А.В., Килбас И.А.; заявитель Самарский Университет. – 1 с.
- 138 Fast Segment Anything / X. Zhao, W. Ding, Y. An [и др.]. – arXiv, 2023.
- 139 Li, J. WHU-OHS: A benchmark dataset for large-scale Hersepctral Image classification / J. Li, X. Huang, L. Tu. – DOI: 10.1016/j.jag.2022.103022 // International Journal of Applied Earth Observation and Geoinformation. – 2022. – Т. 113. – WHU-OHS. – С. 103022.

- 140 Алгоритм автоматического построения компактных дескрипторов сегментирующих гиперспектральные изображения / А.В. Мухин, Р.А. Парингер, Д.Н. Грибанов [и др.] // X Международная конференция и молодёжная школа «Информационные технологии и нанотехнологии» (ИТНТ-2024). – 2024. – Т. 4. – С. 040372.
- 141 Automated Lightweight Descriptor Generation for Hyperspectral Image Analysis / Artem Mukhin, R. Paringer, D. Griбанov, I. Kilbas. – DOI: 10.3103/S1060992X24700164 // Optical Memory and Neural Networks. – 2024. – Vol. 33. – No. 3. – P. 264-275.
- 142 Свидетельство о государственной регистрации программы для ЭВМ № 2023682920. Программный модуль автоматического формирования легковесных дескрипторов для анализа гиперспектральных данных: заявлено 23.10.2023: опубликовано 11.01.2023 / Килбас И.А., Грибанов Д.Н., Мухин А.В., Парингер Р.А.; заявитель Самарский Университет. – 1 с.
- 143 Мухин, А.В. InformativeFeatureSelection: Package which provides a feature selection algorithm which considers class separability and an implementation of Informative Normalized Difference Index (INDI) [Электронный ресурс]. – URL: https://gitlab.com/rustam-industries/feature_extractor (дата обращения: 14.06.2026).

**Приложение А. Акты об использовании результатов
диссертационной работы**

ООО Байт-Самара

443112, Самарская область, город Самара, поселок Управленческий,
Симферопольская ул, д. 3, ком. 7-12
ИНН 6313005255 КПП 631301001 ОГРН 1036300222530
ОКПО 48189828 ОКВЭД 62.01

АКТ о внедрении результатов исследования по теме «Интеллектуальная система для хранения, визуализации и обработки гиперспектральных данных»

Настоящий Акт свидетельствует, что результаты исследований А.В. Мухина по теме «Интеллектуальная система для хранения, визуализации и обработки гиперспектральных данных» внедрены и используются в облачном сервисе анализа гиперкубов hyperspectrum.copter.space.

Разработанная А.В. Мухиным интеллектуальная система для хранения, визуализации и обработки гиперспектральных данных используется для централизованного хранения гиперспектральных изображений и результатов их обработки, интерактивной визуализации спектральных данных, а также выполнения алгоритмов их автоматизированного анализа. Система обеспечивает единую программную среду для работы с гиперспектральными данными и позволяет сохранять, визуализировать и повторно использовать результаты обработки.

Применение разработанной системы позволило обеспечить централизованную организацию работы с гиперспектральными изображениями, сократить трудоемкость подготовки и выполнения процедур их анализа, а также упростить применение и интеграцию алгоритмов обработки данных при решении прикладных задач.

Директор
ООО «Байт-Самара»


Петров И.Е.




УТВЕРЖДАЮ

Проректор по учебной работе

/Гаврилов А.В.

2026 г.

АКТ

о внедрении (использовании) результатов
диссертационного исследования в учебный процесс

Результаты диссертационного исследования «Интеллектуальная система для хранения, визуализации и обработки гиперспектральных данных», выполненного на кафедре технической кибернетики, внедрены (использованы) в учебный(ом) процесс(е) на кафедре технической кибернетики на основании решения кафедры протокол № 8 от «31» марта 2026 г.

Указанные результаты включены в курс «Инженерия данных», направления подготовки 01.04.02 – Прикладная математика и информатика.

Председательствующий на заседании
д.ф.-м.н., профессор

/ Казанский Н.Л.

«09» 03 2026г.

Директор департамента управления
учебным процессом

к.ф.-м.н., доцент

/ Пчёлкина Ю.Ж.

«09» 03 2026г.

Соискатель ученой степени

/Мухин А.В.

«09» 03 2026г.

Начальник отдела сопровождения
научных исследований

к.т.н., доцент

/ Родионов Л.В.

«09» 03 2026г.

УТВЕРЖДАЮ

Первый проректор – проректор
по науке Самарского Университета



/А.И. Розенцвайг

2026 г.

АКТ

о внедрении (использовании) результатов
диссертационного исследования А.В. Мухина

Результаты диссертационного исследования по теме «Интеллектуальная система для хранения, визуализации и обработки гиперспектральных данных», выполненного в научно-исследовательской лаборатории автоматизированных систем научных исследований (НИЛ-602) Самарского университета, были использованы при выполнении государственного задания Минобрнауки России № FSSS-2021-0016 «Дизайн высокоточных спектральных фильтров и интегрированных на чипе спектрометров на основе структур нанопотоники для мониторинга характеристик окружающей среды».

Использование разработанной интеллектуальной системы позволило автоматизировать основные этапы работы с гиперспектральными данными в рамках государственного задания № FSSS-2021-0016, обеспечить их централизованное хранение, визуализацию и обработку, а также повысить эффективность анализа спектральной информации при проведении исследований.

Заведующий лабораторией НИЛ-602
к.ф.-м.н., доцент

Головастикиов Н.В.

УТВЕРЖДАЮ

Первый проректор – проректор
по науке Самарского Университета

/А.И. Розенцвайг

2026 г.



АКТ

о внедрении (использовании) результатов
диссертационного исследования А.В. Мухина

Результаты диссертационного исследования по теме «Интеллектуальная система для хранения, визуализации и обработки гиперспектральных данных», выполненного в научно-исследовательской лаборатории автоматизированных систем научных исследований (НИЛ-35) Самарского университета, были использованы при выполнении государственного задания Минобрнауки России № 0777-2020-0017 «Разработка теоретических основ, методического и программного обеспечения для решения задач оперативного мониторинга Земли на основе интеллектуальной обработки и тематической классификации гиперспектральных данных с использованием линейки малых космических аппаратов на базе платформы «АИСТ».

Использование разработанной интеллектуальной системы позволило автоматизировать основные этапы работы с гиперспектральными данными в рамках государственного задания №0777-2020-0017, обеспечить их централизованное хранение, визуализацию и обработку, а также повысить эффективность анализа спектральной информации при проведении исследований.

Руководитель НИЛ-35
Д.ф.-м.н., профессор


Казанский Н.Л.