

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт (национальный
исследовательский университет)»
Факультет управления и прикладной математики
Кафедра «Интеллектуальные системы»

**Построение модульной системы
видеоаналитики с использованием
нейросетевых методов**
Выпускная квалификационная работа
(бакалаврская работа)

Направление подготовки: 03.03.01 Прикладные математика и
физика

Выполнил:

студент группы Б05-9046

Линдеманн Никита Алексеевич

(подпись обучающегося)

Научный руководитель:

Доктор технических наук

Матвеев Иван Алексеевич

(подпись научного руководителя)

Москва 2023

Аннотация

Работа посвящена построению модульной системы видеоаналитики. В работе рассматриваются классические задачи компьютерного зрения: классификация, детекция, трекинг. Для решения указанных задач используются алгоритмы на основе свёрточных нейронных сетей, в том числе архитектура YoloV5. Предлагаемая система способна работать автономно (в том числе на мини-компьютерах, например, NVIDIA Jetson или Firefly Rockchip) с минимальным потреблением вычислительных и энергетических ресурсов. Разработанная система внедрена и успешно применяется на пунктах взимания платы на федеральных автомобильных дорогах М-1 и М-4 для подсчёта количества осей проезжающих автомобилей, распознавания государственных регистрационных знаков и классификации транспортных средств, на федеральной трассе М-10 для распознавания и подсчета транспортных средств, на одном из выходов московской станции метро ВДНХ для подсчета людей, а так же на других объектах.

Ключевые слова: компьютерное зрение; видеоналитика; свёрточные нейронные сети; задача классификации; задача детекции; трекинг.

Содержание

1	Введение	4
2	Анализ литературы	8
2.1	Задача классификации	8
2.2	Задача детекции	9
3	Постановка задачи	12
4	Анализ существующих решений	15
4.1	Комерческие системы видеоаналитики	15
4.2	Недостатки существующих решений	15
5	Предлагаемая архитектура	18
5.1	Модуль захватчика данных с лидара	18
5.2	Модуль анализа данных с лидара	19
5.3	Модуль детекции объектов	21
5.4	Модуль подсчета осей	23
5.5	Другие модули системы	25
5.6	Система автоматического взимания платы за проезд .	36
5.7	Система автоматического распознавания транспортных средств	41
5.8	Система автоматического распознавания лиц и подсче- та людей	45
6	Заключение	49

1 Введение

Системы видеоаналитики на сегодняшний день находят широкое применение в различных сферах, таких как безопасность, транспорт, медицина и другие. Обработка и анализ видео становятся все более актуальными задачами. В данной работе рассматривается разработка системы видеоаналитики, способной работать автономно с минимальными вычислительными и энергетическими затратами, включая возможность использования на мини-компьютерах.

Современный мир стал невозможно представить без повсеместного применения видеокамер. Они присутствуют практически во всех областях нашей жизни, начиная от общественных мест и транспортных систем до офисов и частных домов. Видеокамеры используются для обеспечения безопасности, наблюдения за процессами, контроля качества и многих других целей. Благодаря развитию технологий, видеокамеры стали более доступными и универсальными инструментами, предоставляющими ценные данные для анализа и принятия решений.

Одним из ключевых факторов, обеспечивающих успех систем видеоаналитики, является существование эффективных методов обработки изображений и видео. Развитие компьютерного зрения и машинного обучения позволяет автоматизировать процессы анализа и извлечения информации из видео. Современные алгоритмы и модели, основанные на глубоком обучении, обладают высокой точностью и скоростью работы, что позволяет оперативно обрабатывать большие объемы видеоданных. Эти методы включают распознавание объектов, классификацию, сегментацию, определение движения и другие задачи, которые помогают понять содержание видео и выявить интересующую информацию.

Для выполнения аналитики видео в реальном времени требуется высокопроизводительное оборудование. С развитием технологий появились мощные центральные процессоры, графические ускорители и специализированные вычислительные платформы, способные обрабатывать видеопотоки с высокой скоростью. Это оборудование может эффективно выполнять сложные алгоритмы обработки изображений.

ражений и видео, обеспечивая реальном времени аналитику и реагирование на события.

Историческая перспектива позволяет нам понять эволюцию систем видеоконтроля. Первые системы видеоконтроля появились в конце XX века и использовались в основном для наблюдения и регистрации происходящих событий. С течением времени технологии развивались, и системы видеоаналитики стали способными автоматически обрабатывать видеопотоки и извлекать полезную информацию. Существуют различные типы систем видеоаналитики, включая распознавание лиц, детекцию движения, анализ поведения и многое другое.

В настоящее время рынок систем видеоконтроля является значительным и продолжает расти. Системы видеоаналитики обеспечивают множество возможностей, включая повышение безопасности, оптимизацию транспортных потоков, улучшение диагностики в медицине. Они способны обрабатывать видео в реальном времени и предоставлять операторам ценную информацию для принятия решений.

В области видеоаналитики существует ряд современных решений, предназначенных для обработки и анализа видеоданных. Рассмотрим несколько из них:

1. Компьютерное зрение и машинное обучение: Эти подходы используются для распознавания объектов, определения сцен, трекинга движущихся объектов и других задач видеоаналитики. Методы глубокого обучения, в особенности сверточные нейронные сети, позволяют достичь высокой точности в распознавании и классификации объектов на видео. Однако, такие системы обычно имеют монолитную архитектуру, что ограничивает гибкость настройки под конкретные требования и задачи.
2. Интеллектуальные видеосистемы безопасности: Эти системы предназначены для обнаружения аномалий и инцидентов на основе анализа видеоданных с камер видеонаблюдения. Они используют различные методы, такие как детектирование дви-

жения, обнаружение лиц и анализ поведения, для раннего обнаружения потенциально опасных ситуаций. Однако, такие системы обычно имеют ограниченные возможности по настройке и интеграции с другими системами видеоаналитики.

3. Коммерческие платформы видеоаналитики: На рынке существует ряд коммерческих платформ, предлагающих широкий спектр функций видеоаналитики, включая распознавание объектов, трекинг движущихся объектов, анализ поведения и другие. Эти платформы обычно имеют готовые модули и инструменты для разработки приложений видеоаналитики, но они могут быть ограничены в гибкости настройки и расширении функциональности.
4. Академические исследования: В академической среде проводятся исследования, направленные на разработку новых методов и алгоритмов видеоаналитики. Эти исследования могут включать различные подходы, такие как использование графовых моделей, временных рядов или дополненной реальности для анализа видеоданных.

Из всего вышеизложенного следует, что задача построения систем видеоаналитики является крайне актуальной. С повсеместным применением видеокамер и потоком информации, поступающим от них, существует необходимость в разработке эффективных методов обработки видео, которые позволят извлечь ценную информацию и принять обоснованные решения. Кроме того, требуется высокопроизводительное оборудование, способное обрабатывать видеопотоки в реальном времени, чтобы оперативно реагировать на происходящие события. Все эти факторы подтверждают значимость и актуальность задачи построения систем видеоаналитики, которая продолжает развиваться и находить новые применения в различных областях человеческой деятельности.

Работа с системами видеоаналитики требует учета различных аспектов, таких как требования к оборудованию, возможности интеграции с существующими системами безопасности и транспортными

сетями, а также оптимизация вычислительных и энергетических затрат. Эти факторы должны быть учтены при разработке автономной системы видеоаналитики с минимальными затратами ресурсов.

В данной работе рассмотрены ключевые аспекты построения системы видеоаналитики на основе компьютерного зрения и свёрточных нейронных сетей. Также рассматриваются вопросы оптимизации вычислительных процессов и энергопотребления системы, чтобы обеспечить её работу на мини-компьютерах с ограниченными ресурсами.

В дальнейшем развитии систем видеоаналитики и применении методов компьютерного зрения с использованием искусственных нейронных сетей можно видеть большой потенциал для улучшения безопасности, оптимизации процессов и повышения эффективности в различных сферах деятельности.

2 Анализ литературы

В настоящее время существует множество работ, посвящённых задачам компьютерного зрения, таких как классификация, детекция и трекинг. Одним из самых эффективных методов для решения этих задач является свёрточная нейронная сеть. Такие сети позволяют автоматически извлекать признаки из изображений, что делает их особенно полезными для анализа видеоданных.

2.1 Задача классификации

В данном разделе мы рассмотрим статьи, посвященные применению нейросетевых методов в задаче классификации в компьютерном зрении.

1. В статье [1] авторы представляют архитектуру нейронной сети, известную как VGGNet, которая демонстрирует высокую точность классификации изображений на наборе данных ImageNet. Эта статья описывает принципы работы сверточных нейронных сетей и предлагает подход к глубокому обучению для компьютерного зрения.
2. В работе [2] авторы представляют архитектуру нейронной сети, известную как AlexNet, которая стала прорывом в области классификации изображений. С помощью глубоких сверточных нейронных сетей, авторы достигли значительного улучшения результатов на наборе данных ImageNet.
3. В статье [3] представлена архитектура ResNet, которая предлагает новый подход к обучению глубоких нейронных сетей. Авторы представляют концепцию «связей-остатков», позволяющую преодолеть проблему затухания градиента и достичь еще более глубоких архитектур с высокой точностью классификации.
4. Книга [4] является одним из ведущих исследований в области глубокого обучения и нейросетей. Она предлагает подробный

обзор различных аспектов нейросетевых методов и их применение в различных задачах компьютерного зрения, включая классификацию изображений.

5. В работе [5] авторы представляют Inception архитектуру, которая использует различные размеры фильтров и слои параллельных сверток для эффективного извлечения признаков изображений. Этот подход значительно улучшает результаты классификации изображений.

2.2 Задача детекции

В данном разделе мы рассмотрим статьи, посвященные применению нейросетевых методов в задаче детекции в компьютерном зрении.

1. В работе [6] авторы представляют архитектуру YOLO (You Only Look Once), которая позволяет выполнять обнаружение объектов в реальном времени на основе одной полной проходки через нейронную сеть. YOLO сочетает высокую скорость и точность в задаче обнаружения объектов. Для применения данного алгоритма на устройствах с ограниченными вычислительными ресурсами предлагается использовать более легковесные модели, такие как YOLOv3-tiny [7] и YOLOv4-tiny [8].
2. В статье [9] авторы представляют архитектуру Densely Connected Convolutional Networks (DenseNet), которая предлагает новый подход к глубоким сверточным нейронным сетям. DenseNet связывает каждый слой непосредственно со всеми последующими слоями, что способствует эффективному использованию информации о признаках.
3. В работе [7] авторы представляют улучшенную версию архитектуры YOLO, известную как YOLOv3. YOLOv3 обладает высокой скоростью обнаружения объектов и точностью классификации, благодаря использованию более глубокой архитектуры и различных масштабов признаков.

4. Статья [10] представляет архитектуру Faster R-CNN, которая предлагает эффективный подход к задаче детекции объектов. Авторы представляют новый модуль Region Proposal Network (RPN), интегрированный в нейросетевую архитектуру, чтобы создавать регионы предложений для объектов. Этот метод считается важным прорывом в области детекции объектов.
5. В статье [11] авторы представляют архитектуру Feature Pyramid Networks (FPN), которая улучшает производительность в задаче обнаружения объектов на разных масштабах. FPN объединяет многоуровневые признаки для более точного и эффективного обнаружения объектов.

Таким образом, существует множество нейросетевых архитектур для решения задач компьютерного зрения, и выбор подхода зависит от конкретной задачи, доступных ресурсов и других факторов.

До недавнего времени для решения задач компьютерного зрения широко использовались классические методы машинного обучения, однако с развитием нейронных сетей и появлением глубокого обучения, нейросетевые алгоритмы стали более эффективными для решения задач компьютерного зрения. Одним из главных преимуществ нейронных сетей в таких задачах является их способность автоматически извлекать признаки из входных данных без необходимости ручной настройки параметров. Классические методы компьютерного зрения основаны на признаках объектов, признаки определяются экспертом и могут быть недостаточно полными или неэффективными для данной задачи.

Кроме того, нейронные сети могут обрабатывать данные с большой вычислительной мощностью, позволяя решать задачи с высокой точностью и скоростью. Нейронные сети могут использовать многоканальные изображения и другие типы входных данных, что расширяет их применимость в задачах компьютерного зрения.

Современные фреймворки (PyTorch, Keras, TensorFlow) позволяют быстро обучать (а так же дообучать предобученные) нейронные сети на новых данных и легко интегрировать их в существующую ин-

фраструктуру. Таким образом достигается универсальность, что расширяет возможности применения нейронных сетей.

3 Постановка задачи

В последние годы использование видеоданных в различных сферах стало все более распространенным, однако обработка и анализ больших объемов видеоматериалов является трудоемким процессом, требующим значительных вычислительных ресурсов. В данной работе предлагается разработка модульной системы видеоаналитики, которая позволит повысить эффективность обработки и анализа видеоданных путем распределения вычислительных нагрузок между различными модулями.

Цель работы: исследовать, разработать и оценить модульную систему видеоаналитики, основанную на существующих методах обработки и анализа видеоданных, с целью определения основных задач видеоаналитики, разработки эффективной архитектуры системы, методов передачи данных и алгоритмов обработки, а также сравнить разработанную систему с существующими методами и системами в терминах производительности и точности анализа видеоданных. Дополнительно, предложить пути дальнейшего развития и улучшения модульной системы видеоаналитики, включая добавление новых модулей, оптимизацию методов передачи данных и улучшение алгоритмов обработки и анализа видеоданных.

В рамках исследования поставим следующие задачи:

1. Внедрить в систему видеоаналитики модуль подсчета габаритов ТС на основе данных с лидара:
 - (a) Разработать модуль захвата данных с лидара.
 - (b) Разработать модуль анализа данных с лидара.
 - (c) Добиться точности определения габаритов (высота и ширина) ТС с погрешностью не более 8%.
2. Внедрить в систему видеоаналитики модуль подсчета количества осей ТС:
 - (a) Разработать модуль детекции колес.

- (b) Добиться точности определения колес с mAP не менее 0,99.
- (c) Разработать модуль подсчета количества осей на основе данных детектора.
- (d) Добиться точности подсчета осей ТС не менее 95%.

Ожидается, что разработанная модульная система видеоаналитики будет обладать следующими характеристиками и преимуществами:

1. Модульность: Система будет состоять из набора независимых модулей, каждый из которых будет специализироваться на выполнении конкретной задачи видеоаналитики. Это позволит гибко настраивать и комбинировать модули в зависимости от конкретных требований и задач анализа видеоданных.
2. Распределение вычислительных нагрузок: Система будет распределять вычислительные задачи между различными модулями, позволяя использовать мощности процессора и других ресурсов эффективно. Это повысит скорость обработки видеоданных и снизит требования к вычислительным ресурсам.
3. Высокая точность анализа: Каждый модуль будет оптимизирован для своей конкретной задачи, что позволит достичь высокой точности в выполнении аналитических операций. Например, модуль распознавания объектов может использовать современные алгоритмы глубокого обучения, а модуль трекинга движущихся объектов — оптимизированные алгоритмы компьютерного зрения.
4. Эффективная передача данных: Разработанные методы передачи видеоданных между модулями системы будут обеспечивать минимальную потерю информации и эффективное использование ресурсов. Это позволит сохранить высокое качество видеоданных при их обработке различными модулями.

5. Практическая применимость: Разработанная модульная система видеоаналитики будет иметь широкий спектр применения, включая области безопасности, видеонаблюдения, транспорта, медицины и другие сферы. Она будет способна обрабатывать и анализировать видеоданные с различных источников, таких как видеокамеры, видеозаписи и потоковые видеоданные.

Ожидается, что разработанная модульная система видеоаналитики будет иметь преимущества по сравнению с существующими методами и системами в терминах эффективности обработки видеоданных, точности анализа и гибкости настройки под конкретные требования и задачи. Проведенное экспериментальное исследование прототипа системы позволит оценить ее производительность, точность и стабильность работы на реальных видеоданных, а также сравнить ее с существующими методами и системами.

В дальнейшем, на основе полученных результатов, предлагается исследовать возможности дальнейшего развития и улучшения модульной системы видеоаналитики. Это может включать добавление новых модулей для выполнения дополнительных задач видеоаналитики, оптимизацию методов передачи данных для еще более эффективного использования ресурсов, а также улучшение алгоритмов обработки и анализа видеоданных с целью повышения точности и скорости работы системы.

В итоге, разработка модульной системы видеоаналитики с учетом вышеописанных задач и ожидаемых результатов будет способствовать повышению эффективности обработки и анализа видеоданных, что имеет большое значение в различных областях применения, где требуется автоматизированная обработка и интерпретация видеoinформации.

4 Анализ существующих решений

4.1 Комерческие системы видеоаналитики

Рассмотрим некоторые примеры коммерческих платформ видеоаналитики, которые предлагают широкий спектр функций:

1. NVIDIA Metropolis [12]: Комплексный набор инструментов и платформ для видеоаналитики, включающий глубокое обучение, компьютерное зрение и аналитику данных для различных применений, включая безопасность, городскую инфраструктуру и розничную торговлю.
2. Hikvision iVMS-5200 Professional [13]: Платформа видеоаналитики, разработанная компанией Hikvision, предоставляющая расширенные возможности анализа видео, включая распознавание лиц, анализ поведения и детектирование аномалий.
3. Milestone XProtect [14]: Комплексная платформа видеопрограммирования, которая также включает модули видеоаналитики для обнаружения и классификации объектов, анализа поведения и других задач.
4. Avigilon Control Center [15]: Платформа видеопрограммирования с широким набором интеллектуальных функций, включая распознавание лиц, автоматический поиск и анализ видеоданных.

Это лишь некоторые примеры коммерческих платформ видеоаналитики, доступных на рынке. Важно отметить, что список таких платформ может быть более обширным и зависит от конкретных требований и целей исследования.

4.2 Недостатки существующих решений

Недостатки коммерческих платформ видеоаналитики:

1. Высокая стоимость: Многие коммерческие платформы видеоаналитики имеют значительную стоимость в связи с лицензированием, подписками и дополнительными расходами на оборудование и поддержку. Это может ограничить доступность этих решений для некоторых организаций и исследователей.
2. Ограниченная гибкость настройки: Коммерческие платформы обычно имеют predetermined набор функций и возможностей, что ограничивает гибкость в настройке и расширении системы под конкретные требования и задачи.
3. Закрытость исходного кода: Многие коммерческие платформы являются закрытыми системами с недоступным исходным кодом. Это ограничивает возможности модификации и адаптации системы под уникальные потребности исследователя или организации.

Преимущества предлагаемой архитектуры модульной системы видеоаналитики:

1. Гибкость настройки: Предлагаемая модульная система видеоаналитики позволяет гибко настраивать и комбинировать модули в соответствии с конкретными требованиями и задачами. Это дает возможность создавать индивидуальные конфигурации системы, оптимизированные под конкретные потребности пользователя.
2. Расширяемость: Модульная архитектура позволяет легко добавлять новые модули и функции в систему. Это дает возможность системе развиваться и адаптироваться к изменяющимся требованиям и новым задачам видеоаналитики.
3. Экономическая эффективность: В сравнении с коммерческими платформами, предлагаемое решение модульной системы видеоаналитики может быть более экономически эффективным. Отсутствие лицензионных и подписочных платежей, а также

возможность использования открытых исходных кодов и стандартных программных инструментов позволяют снизить затраты на разработку и поддержку системы.

4. Гибкость интеграции: Модульная система видеоаналитики может легко интегрироваться с другими существующими системами и инфраструктурой, такими как системы видеонаблюдения, управления доступом или базами данных. Это обеспечивает более гармоничную и эффективную работу системы в контексте существующей инфраструктуры.
5. Независимость от поставщика: Пользователи модульной системы видеоаналитики не зависят от определенного поставщика или вендора. Это обеспечивает свободу выбора и гибкость в использовании и поддержке системы.

Однако, необходимо отметить, что предлагаемая модульная система видеоаналитики требует дополнительной работы по разработке и интеграции модулей. Это может потребовать большего времени и ресурсов по сравнению с использованием готовых коммерческих платформ. Кроме того, необходимость опытных разработчиков и исследователей для настройки и наладки системы также является важным фактором при рассмотрении преимуществ и недостатков предлагаемого решения.

5 Предлагаемая архитектура

Главная особенность предлагаемой системы видеоаналитики - модульность. Система представляет собой микросервисную архитектуру, в которой модули могут работать независимо друг от друга. Каждый модуль работает в собственном процессе и взаимодействует с другими модулями. Такой подход позволяет ускорить разработку и сделать систему более гибкой и управляемой.

5.1 Модуль захватчика данных с лидара

В рамках работы был разработан и реализован модуль захватчика данных с лидара. Лидар является сенсорным устройством, которое непрерывно осуществляет измерения окружающей среды. Модуль разработан для эффективного получения и обработки данных с лидара с целью последующего использования в различных приложениях.

Модуль захватчика данных с лидара включает в себя два основных потока работы. В первом потоке осуществляется прием сообщений с лидара по протоколу UDP (User Datagram Protocol), который является простым и быстрым протоколом передачи данных. Полученные сообщения сохраняются в очереди для последующей обработки.

Второй поток работы модуля занимается обработкой данных с лидара. Он получает сообщения из очереди, которые были сохранены в первом потоке, и проводит их обработку. Обработка данных включает в себя отбрасывание невалидных данных (например, с расстоянием 0) и излишних данных (например, внутренне время лидара), а так же конвертацию данных в удобный формат для дальнейшей передачи.

После обработки данных модуль формирует сообщение для передачи по shared memory (разделяемая память). Shared memory позволяет нескольким процессам взаимодействовать и обмениваться данными, используя общую область памяти. Это обеспечивает эффективную передачу обработанных данных с лидара другим модулям

или приложениям, которые могут использовать эти данные для различных целей, таких как навигация роботов, картография, детектирование препятствий и другие задачи.

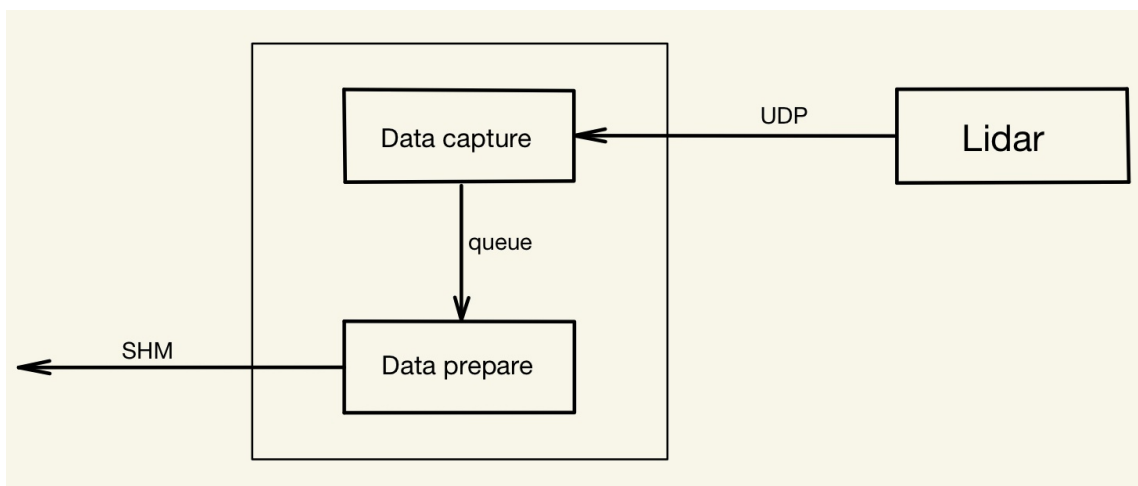


Рис. 1: Схема работы модуля захватчика данных с лидара.

Разработанный модуль захватчика данных с лидара предоставляет эффективный механизм для получения и обработки информации с лидара. Он может быть использован в широком спектре приложений, связанных с анализом окружающей среды и управлением робототехническими системами. Дальнейшие исследования и разработки могут быть направлены на улучшение производительности модуля, расширение его функциональности и адаптацию к конкретным требованиям приложений.

Модуль реализован на языке программирования Python3 и был успешно запущен на мини-компьютере Firefly-RK3399.

5.2 Модуль анализа данных с лидара

В рамках работы был разработан и реализован модуль захватчика данных с лидара. Модуль разработан с целью обработки информации, полученной от нескольких лидаров, и представления ее в декартовой системе координат. Дополнительно модуль выполняет вычисление габаритов объектов на основе полученных данных.

Первым шагом работы модуля является прием данных через shared memory от захватчика. Shared memory, как способ межпроцессорного взаимодействия, позволяет эффективно передавать информацию между различными модулями, которые запущены в разных процессах. Полученные данные содержат измерения, полученные с нескольких лидаров.

Затем модуль выполняет преобразование данных из полярной системы координат, используемой лидарами, в декартовую систему координат. Для каждой измеренной точки с координатами (r, φ) модуль вычисляет ее декартовые координаты (x, y) с использованием следующего преобразования:

$$\begin{pmatrix} x \\ y \end{pmatrix} = r \cdot \begin{pmatrix} \cos \varphi \\ \sin \varphi \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix},$$

здесь (x_0, y_0) – координаты лидара. Это преобразование позволяет представить данные с разных лидаров в общей декартовой системе координат, где каждая точка определяется своими координатами (x, y) .

Далее модуль вычисляет габариты объектов на основе полученных данных. Габариты представляют собой ограничивающий прямоугольник, охватывающий объект. Используя декартовы координаты точек, модуль определяет минимальные и максимальные значения координат по осям x и y , а затем вычисляет размеры прямоугольника. Таким образом, разработанный модуль предоставляет возможность получения данных от захватчика через shared memory, их преобразования в декартову систему координат и вычисления габаритов объектов.

В результате проведенных экспериментов была оценена точность определения габаритов. Для этого использовалось 23 транспортных средства с заранее известными высотой и шириной, при этом каждое ТС измерялось лидаром несколько раз, что дало в сумме $n = 114$ проездов. Для каждого проезда ТС с известными габаритами высоты \hat{h} и ширины \hat{w} вычислялась относительная ошибка следующим

образом:

$$e_h = \frac{|h - \hat{h}|}{\hat{h}}, \quad e_w = \frac{|w - \hat{w}|}{\hat{w}},$$

здесь h и w – вычисленные модулем высота и ширина ТС соответственно. Далее значения ошибок усреднялись:

$$e = \frac{1}{n} \cdot \sum_{i=1}^n \frac{e_h + e_w}{2} = \frac{1}{114} \cdot \sum_{i=1}^{114} \frac{e_h + e_w}{2} \approx 0,048.$$

Средняя ошибка составила 4,8%, что говорит о достаточно высокой точности работы модуля.

Таким образом, разработанные модули захватчика данных с лидара и обработки данных представляют собой важный инструмент для получения, обработки и анализа информации с лидаров. Их функциональность, точность и эффективность позволяют использовать модули в различных приложениях, где требуется высокоточное определение габаритов объектов и обработка данных в режиме реального времени.

Модуль реализован на языке программирования Python3 и был успешно запущен на мини-компьютере Firefly-RK3399.

5.3 Модуль детекции объектов

В рамках работы был разработан и реализован модуль детекции объектов, основанный на архитектуре YOLOv5 [6] с предобученными весами. Модуль разработан для автоматического обнаружения и классификации объектов на изображениях.

Первым шагом разработки модуля было использование архитектуры YOLOv5, одной из самых популярных архитектур для обнаружения объектов на изображениях. Она отличается высокой скоростью и точностью обнаружения, а также легкостью использования и настройки. В данном исследовании модуль использовал предобученные веса YOLOv5, что позволило достичь хороших результатов без необходимости обучения модели с нуля.

Для обучения и оценки модуля были использованы обучающая и тестовая выборки изображений. Обучающая выборка состояла из 2640 изображений, на которых объекты были размечены с соответствующими классами. Тестовая выборка состояла из 276 изображений и использовалась для оценки качества работы модуля.

Полученное качество модуля было измерено с помощью метрики средней точности обнаружения (mean Average Precision, mAP). Для данной метрики использовалась стандартная процедура оценки, которая учитывает как точность, так и полноту обнаружения объектов. Полученное значение mAP составило 0,995, что свидетельствует о высокой точности и надежности модуля.



Рис. 2: Батч из обучающей выборки.

Разработанный модуль детекции объектов предоставляет эффективный инструмент для обнаружения и классификации объектов на изображениях. Он может быть использован в различных приложении-

ях, включая системы видеонаблюдения, автономные транспортные средства, системы безопасности и другие области, где требуется автоматическое обнаружение объектов. Дальнейшее развитие модуля может включать расширение обучающей выборки, улучшение архитектуры модели и применение алгоритмов для отслеживания объектов во временных последовательностях изображений. Это позволит расширить функциональность модуля и улучшить его производительность в различных

Модуль реализован на языке программирования Python3. Данный программный модуль был успешно запущен на мини-компьютере Nvidia Jetson Nano 4GB.

5.4 Модуль подсчета осей

В рамках работы был разработан и реализован модуль подсчета осей, который разработан для автоматического распознавания и подсчета количества осей транспортных средств на видеопотоках.

Первым шагом модуля является получение координат боксов, охватывающих оси ТС, с помощью протокола websocket. Координаты предоставляют информацию о положении и размерах объектов на изображениях. Получение координат боксов осуществляется на предыдущих этапах системы, например, с помощью модуля детекции объектов, описанного в предыдущем разделе.

Для анализа последовательности координат боксов объектов и подсчета количества осей был использован алгоритм SORT (Simple Online and Real-time Tracking) [16] – популярный алгоритм трекинга, который позволяет отслеживать движущиеся объекты на видеопотоках или в последовательности изображений. Алгоритм использует методы ассоциации объектов между кадрами, а также модели прогнозирования для определения траекторий движения объектов.

Достигнутая точность распознавания количества осей составила 98,6%. Точность рассчитывалась как отношение количества ТС с правильно определенным количеством осей $N_{true} = 575$ к общему

количеству ТС из выборки $N_{all} = 583$:

$$accuracy = \frac{N_{true}}{N_{all}} = \frac{575}{583} \approx 0,986.$$

Это означает, что модуль демонстрирует высокую точность в подсчете осей на видеопотоках. Такое качество является важной характеристикой модуля, особенно в приложениях, где требуется точное определение количества осей для принятия решений или выполнения задач.



Рис. 3: Визуализация трека

В дальнейшем развитии модуля подсчета осей можно рассмотреть возможности оптимизации алгоритма трекинга, включая улучшение методов ассоциации объектов, использование дополнительной информации, такой как скорость или ускорение объектов, а также применение методов глубокого обучения для более точного и надежного трекинга.

Дополнительное развитие модуля подсчета осей может включать улучшение алгоритмических подходов для более точного определения осей объектов. Например, можно исследовать применение методов компьютерного зрения, таких как извлечение признаков или классификация, для более точного определения осей на изображениях. Также возможно использование глубоких нейронных сетей и обучение модели на большем объеме разнообразных данных для повышения точности и обобщающей способности модуля.

Важным аспектом дальнейшего развития модуля является его адаптация к различным сценариям и условиям применения. Модуль должен быть способен работать в реальном времени и справляться с различными условиями освещения, перспективами и шумами на изображениях. Проведение экспериментов и оценка производительности модуля в различных сценариях являются важными задачами для оптимизации и улучшения его работы.

В заключение, разработанный модуль подсчета осей представляет собой эффективный инструмент для автоматического подсчета количества осей на видеопотоках. Он показывает высокую точность в подсчете осей и имеет потенциал для дальнейшего развития и улучшения. Результаты данной работы могут быть применены в различных областях, включая автоматизацию промышленных процессов, системы безопасности, мониторинг транспорта и другие приложения, требующие точного подсчета осей транспортных средств.

Модуль реализован на языке программирования Python3 с использованием фреймворка PyTorch и был успешно запущен на мини-компьютере Nvidia Jetson Nano 4GB.

5.5 Другие модули системы

Рассмотрим другие модули системы видеоаналитики.

- **rtsp-capture.** Захватчик данных с сенсора через Ethernet – программный модуль, предназначенный для приема, обработки кадров, полученных от сенсора Sony IMX273 (далее - imx273), и их передачи другим модулям. Функции ethernet-захватчика:

- настройка `imx273` для генерации видеопотока с определенными характеристиками;
- обработка принятых кадров;
- передача обработанных кадров модулям;
- отправка модулю `dispatcher` информации о своей работоспособности.

Обработка принятых кадров включает в себя:

- установка яркости;
- установка насыщенности;
- установка контраста;
- установка зеркалирования;
- установка усиления;
- установка гамма-коррекции;
- установка баланса белого;
- изменение разрешения (при необходимости);
- изменение частоты (при необходимости);
- преобразование из одной цветовой модели в другую (при необходимости).

Модуль написан на языке программирования C++.

- **usb-capturer**. Захватчик данных с сенсора через порт USB – программный модуль, предназначенный для получения потока данных с сенсора (модуль аналогичен `eth-glosav-capturer`). Модуль написан на языке программирования C++.
- **pci-express-capture**. Захватчик данных с сенсора через шину PCI Express – программный модуль, предназначенный для получения потока данных с сенсора (модуль аналогичен `eth-capturer`).

Модуль написан на языке программирования C++.

- **h264-encoder.** Кодировщик – программный модуль, предназначенный для приема и сжатия с использованием кодеков H.264, H.265, MJPEG кадров и их рассылки другим модулям. Функции кодировщика:
 - прием от модулей кадров для сжатия;
 - преобразование входящих кадров в соответствии с заданными настройками: изменение размера, частоты (при необходимости);
 - сжатие кадров с использованием одного из кодеков H.264, H.265, MJPEG;
 - создание каналов обмена данными для рассылки сжатых кадров модулям;
 - рассылка сжатых кадров;
 - отправка информации о своей работоспособности модулю dispatcher.

Модуль написан на языке программирования C++.

- **rtsp-server.** RTSP-сервер – программный модуль, предназначенный для широковещания с использованием протокола RTSP поступивших к нему сжатых кодеками H.264 и H.265 видеопотоков и текстовых метаданных модулей аналитики (далее – метаданных). Функции rtsp-сервера:
 - отправка по именованным каналам (далее – Pipe) запросов другим модулям на получение сжатых кадров;
 - прием сжатых кадров;
 - установка соединения по WebSocket с модулями аналитики;
 - прием метаданных;
 - дублирование полученных кадров/метаданных в соответствии с количеством получателей;

- упаковка в формат протокола RTSP кадров;
- упаковка в формат протокола RTP метаданных;
- рассылка кадров/метаданных получателям.

Модуль написан на языке программирования C++.

- **target-detector**. Детектор событий – программный модуль, предназначенный для обнаружения на кадрах транспортных средств и определения направления их движения. Детектор событий функционирует по следующей схеме:
 - Прием несжатых кадров в цветовой модели *RGB* или *Gray8* от модулей.
 - Поворот кадра (при необходимости).
 - Обработка изображения мишени в кадре:
 1. определение границ мишени в кадре в соответствии с заданными координатами ее вершин;
 2. перенос пикселей мишени из наклонной области в прямоугольную матрицу;
 3. в следствие оптических искажений и/или из-за крепления видеоустройства мишень в кадре имеет наклон в одну из сторон (вправо или влево). Для корректной работы детектора транспорта пиксели из наклонного изображения мишени переносятся в отдельную специально подготовленную прямоугольную матрицу.
 4. нормализация яркости мишени в случаях ее засветки или затемнения;
 5. подсчитывается количество пикселей каждой яркости, после чего по 16% от пикселей с наибольшей и наименьшей яркостью отбрасываются. Далее вычисляется коэффициент, в соответствии с которым освободившиеся места заполняются оставшимися пикселями.
 6. определение границ клеток мишени;

7. устранение засвета части мишени путем поворота ее изображения на 180° ;
 8. определение средней яркости каждой клетки мишени как суммы яркостей ее пикселей;
 9. определение отношения контрастов соседних клеток мишени по горизонтали и вертикали;
 10. сумма яркостей клетки с большей средней яркостью делится на сумму яркостей второй клетки.
 11. контроль отношения контрастов соседних клеток мишени;
 12. отношение контрастов соседних клеток не должно быть меньше значения, заданного параметром, в противном случае мишень считается нераспознанной.
- Обработка области детектирования (далее - ROI):
1. определение ширины ROI;
Ширина ROI должна в 11 раз превышать ширину мишени. Центр ROI по горизонтали совпадает с центром мишени.
 2. определение высоты ROI;
Высота ROI определяется нижней границей дороги и высотой мишени - на 5 пикселей больше высоты мишени.
 3. определение высоты области под мишенью, в которой происходит «накопление фона»;
Высота области под мишенью определяется нижней границей мишени и нижней границей дороги, которая задается в параметром.
 4. формирование фона;
Фон формируется как сумма заданного количества кадров.
 5. вычисление разницы текущего кадра с фоном;
Разница - вычитание по абсолютному значению пикселей текущего кадра и фона. вычитание из полученной

- разницы порогового значения, заданного в параметре.
6. условное деление ROI на 3 части по горизонтали;
 7. условное деление каждой части на 3 одинаковых подчасти по горизонтали и вертикали;
 8. определение среднего значения пикселей отличных от фона для каждой подчасти;
 9. определение среднего значения пикселей отличных от фона для части как наибольшего среднего значения пикселей среди подчастей.
- Анализ текущего кадра
 1. определение наличия транспортного средства на изображении;
 2. определение направления движения транспортного средства;
При въезде: в какой части области отличие от фона наступило раньше.
При выезде: в какой части области отсутствие отличия от фона наступило раньше.
 3. исключение появления ложных событий «выезд транспортного средства» в случаях наличия просвета (разрыва) между кузовом и кабиной водителя.
 - Сохранение кадра с детектированным событием в заданный каталог в заданном формате.
 - Подготовка исходного кадра к отправке получателю:
 1. сжатие кадра до размера 64 пиксела по большей стороне;
 2. дополнение сжатого кадра прозрачным фоном по меньшей стороне до размера 64 пиксела;
 3. кодирование сжатого кадра с фоном по стандарту base64.
 - Формирование сообщения в формате JSON для отправки результатов работы детектора транспорта получателю.

– Отправка сообщения.

Модуль написан на языке программирования C++.

- **number-cars.** Модуль распознавания автономеров - программный модуль, предназначенный для распознавания номеров государственных регистрационных знаков (далее - ГРЗ) транспортных средств (далее - ТС).

Алгоритм работы модуля распознавания автономеров:

1. прием кадра;
2. изменение разрешения кадра до заданного;
3. сжатие кадра для детектирования ГРЗ;
4. детектирование ГРЗ в кадре;
5. сжатие кадра для распознавания номер
6. сохранение изображения с распознанным номером ГРЗ;
7. сохранение первого/последнего кадра с данным распознанным номером ГРЗ;
8. Отправка сообщений с результатами работы модуля.

Детекция и распознавание ГРЗ реализовано с помощью модели YOLO [6]. Модель обучена на датасете, состоящем из изображений ТС с ГРЗ 56 стран.

Модуль написан на языке программирования Python3 с использованием фреймворка PyTorch.

- **face-descriptor.** Модуль распознавания лиц – программный модуль, предназначенный для сравнения обнаруженных в кадре лиц с имеющимися в базе данных и отправки результатов сравнения получателям (клиентам).

Модуль написан на языке программирования Python3 с использованием фреймворка PyTorch.

- **dispatcher**. Диспетчер – программный модуль, предназначенный для реализации сервиса управления и мониторинга состояния видеоконвейеров и входящих в их состав программных модулей на конкретном узле (видеокамере, сервере и т.д.). Диспетчер выполняет следующие функции:

1. автоматический запуск модулей;
2. контроль за цикливанием модулей, их остановка и перезапуск;
3. сбор статистики работы модулей.

Алгоритм работы модуля:

1. Проверка наличия файла `ConveyorsList.json` с описанием всех модулей, которые необходимо контролировать. Если файл отсутствует, диспетчер работает в режиме «холостого хода»;
2. Вызов функции, которая запускает модули узла, указанные в `ConveyorsList.json`;
3. Формирование файла `conveyor.json`.
Далее диспетчер раз в секунду проверяет файл `ConveyorsList.json` на изменения (временная метка и размер файла) и, при наличии таковых, редактирует `conveyor.json`.
4. Создание области `shared memory` для проверки модулей на заикливание;
5. Проверка модулей на заикливание, их остановка и перезапуск (при необходимости);
6. Формирование файлов со статистическими данными о работе модулей;
7. Обработка запросов администратора узла.

Модуль написан на языке программирования C++.

- **postgres-database.** Модуль базы данных (далее БД) – программный модуль, предназначенный для записи, чтения и хранения данных с помощью PostgreSQL – свободной объектно-реляционной системы управления базами данных. В модуле используется клиент-серверная архитектура.

Модуль базы данных выполняет следующие функции::

1. общение с другими модулями посредством websocket;
2. создание сервера с БД;
3. создание новых таблиц;
4. запись новых данных в таблицы с их предварительной валидацией;
5. возможность чтения БД.

Модуль написан на языке программирования C++.

- **proxy.** Модуль принятия решений – программный модуль, предназначенный для сбора результатов работы модулей аналитики и формирования транзакции событий на основе этих данных. Модуль написан на языке программирования C++.

- **panorama.** Панорама – программный модуль, предназначенный для просмотра и оценки состояния охраняемого объекта (территории) посредством использования средств системы видеонаблюдения (далее СВН).

Система оператора выполняет следующие функции:

1. отображение в реальном времени видеопотоков устройств (узлов) выбранной СВН;
2. отображение архива устройств (узлов) выбранной СВН;
3. сохранение в режиме реального времени изображения с устройства на локальный ПК;
4. отображение информации о текущих событиях видеоаналитики СВН;

5. отображение информации об архивных событиях видеоаналитики СВН;
6. просмотр видеороликов событий видеоаналитики СВН;
7. создание и конфигурирование пользовательских раскладок (экранов);
8. экспорт фрагмента архива на локальный компьютер;
9. одновременный экспорт архива нескольких устройств СВН на локальный компьютер;
10. циклическое переключение раскладок СВН (функция «карусель»);
11. создание и конфигурирование плана размещения узлов (устройств) СВН на контролируемой территории (объекте).

Модуль написан на языке программирования C++ с использованием фреймворка Qt.

- **archive-writer**. Архивописатель - программный модуль, предназначенный для записи видеопотоков в файловый архив. При запуске модуля, он ожидает RTSP поток по заданному в настройках адресу, далее записывает видео в файл с заданной периодичностью (например, одна минута), и отправляет по протоколу WebSocket сообщение в базу данных о записанном файле. Писатель архива предназначен для:

- приема видеопотоков узлов;
- записи принятых видеопотоков в архив на локальный или сетевой диск, смонтированный как локальный;
- отправки метаданных записанных в архив видеопотоков на сервер метаданных.

Модуль написан на языке программирования C++.

- **archive-server**. Архивочитатель – программный модуль, предоставляющий доступ к записанным видео в архиве по протоколам HTTP и RTSP. При запуске модуля начинает работать сервер, ожидая запроса. При получении запроса модуль archive-server по протоколу WebSocket запрашивает пути из базы данных, по которым расположены необходимые видео, обрабатывает их и отдает по тому же адресу, откуда был получен запрос, и по тому же протоколу (HTTP или RTSP). Сервер видеоархива отправляет получателю в соответствии с поступившими запросами видеозаписи из архива устройств (узлов) системы видеоаналитики для просмотра. Функции сервера видеоархива:

- прием запросов от получателей на предоставление заданных видеозаписей из архива;
- расположения запрашиваемых видеозаписей (фрагментов) архива;
- прием сообщений от базы данных с адресами расположения запрашиваемых видеозаписей архива;
- передача получателю кадров видеозаписей из архива.

Модуль написан на языке программирования C++.

Вышеперечисленные модули взаимодействуют между собой в рамках микросервисной архитектуры. Каждый модуль работает в своем собственном процессе и может обмениваться данными и командами с другими модулями, используя определенные протоколы или интерфейсы.

Модульность системы позволяет разработчикам создавать новые модули или модифицировать существующие независимо друг от друга. Например, можно легко добавить новый модуль для обработки видеоданных или интегрировать сторонние модули для расширения функциональности системы.

Кроме перечисленных модулей, в системе видеоаналитики могут присутствовать и другие модули, такие как модуль обнаружения

движения, модуль распознавания лиц, модуль анализа объектов и другие. Каждый модуль выполняет конкретную функцию и способствует общей функциональности комплексу видеоаналитики.

Таким образом, модульная архитектура позволяет создавать гибкие и масштабируемые системы видеоаналитики, где каждый модуль выполняет определенную задачу и может быть независимо изменен или заменен без влияния на работу других модулей.

5.6 Система автоматического взимания платы за проезд

Рассмотрим пример использования предложенной архитектуры для решения задачи автоматического взимания платы за проезд. Для решения этой задачи необходимо:

1. Детектировать событие проезда ТС;
2. Распознавать ГРЗ транспортного средства;
3. Подсчитывать количество осей ТС;
4. Формировать транзакцию и записывать ее в базу данных;
5. Отображать события проезда в удобно читаемом интерфейсе.

Первым этапом необходимо детектировать событие проезда транспортного средства. Для этого используется модуль `target-detector`, который осуществляет детекцию события, в данном случае, проезжающих транспортных средств. Кадры для аналитики модули получают от модуля `rtsp-capture`, который захватывает `rtsp` поток и передает кадры по `shared memory` другим модулям, в том числе `target-detector`. Затем, для идентификации транспортного средства и определения его государственного регистрационного знака (ГРЗ), применяется модуль `number-cars`, который выполняет детекцию и распознавание ГРЗ.

Далее, для определения количества осей транспортного средства, используются модули `object-detector` и `axis-detector`. Первый модуль

осуществляет детекцию колес и передает их координаты второму, который осуществляет подсчет количества осей.

Так же в данной реализации системы используются датчики высоты, установленные на уровне 2 и 2,6 метров над уровнем земли, которые позволяют точнее определять тип проезжаемого ТС.

Результаты аналитики и данные с датчиков высоты передаются модулю проху, который принимает решение о том, был ли проезд, и формирует транзакцию за принятие решений о формировании транзакции проезда ТС. Транзакция записывается в базу данных.

Модуль postgres-database используется для хранения и управления данными системы. Он представляет базу данных PostgreSQL, где сохраняются информация о проездах транспортных средств, распознанных ГРЗ, количестве осей и других связанных данных.

Модули archive-writer и archive-reader предоставляют функционал записи видео с камер и их просмотра. Работу всех модулей (а именно: запуск, завершение работы, контроль зависания) контролирует модуль dispatcher. Так же этот модуль собирает статистику работы всех модулей, взаимодействие с которыми происходит с помощью POSIX.

Для отображения и представления результатов аналитики система использует модуль rapogama, который предоставляет удобный интерфейс для визуализации данных. Через этот интерфейс оператор может просматривать статистику по проездам транспортных средств, результаты распознавания ГРЗ, информацию о габаритах и количестве осей, а также другие сводные данные, представленные в понятной и информативной форме. На рисунках 9 и 6 показаны примеры интерфейса.

На рисунке 9 представлен интерфейс системы, где оператор может просматривать видеозаписи с камер и осуществлять навигацию по событиям проезда. Оператор имеет возможность просматривать записи проездов транспортных средств, а также получать доступ к дополнительной информации, такой как распознанные ГРЗ и количество осей.

На рисунке 6 показан интерфейс системы, позволяющий оператору просматривать события проезда и получать детальную инфор-

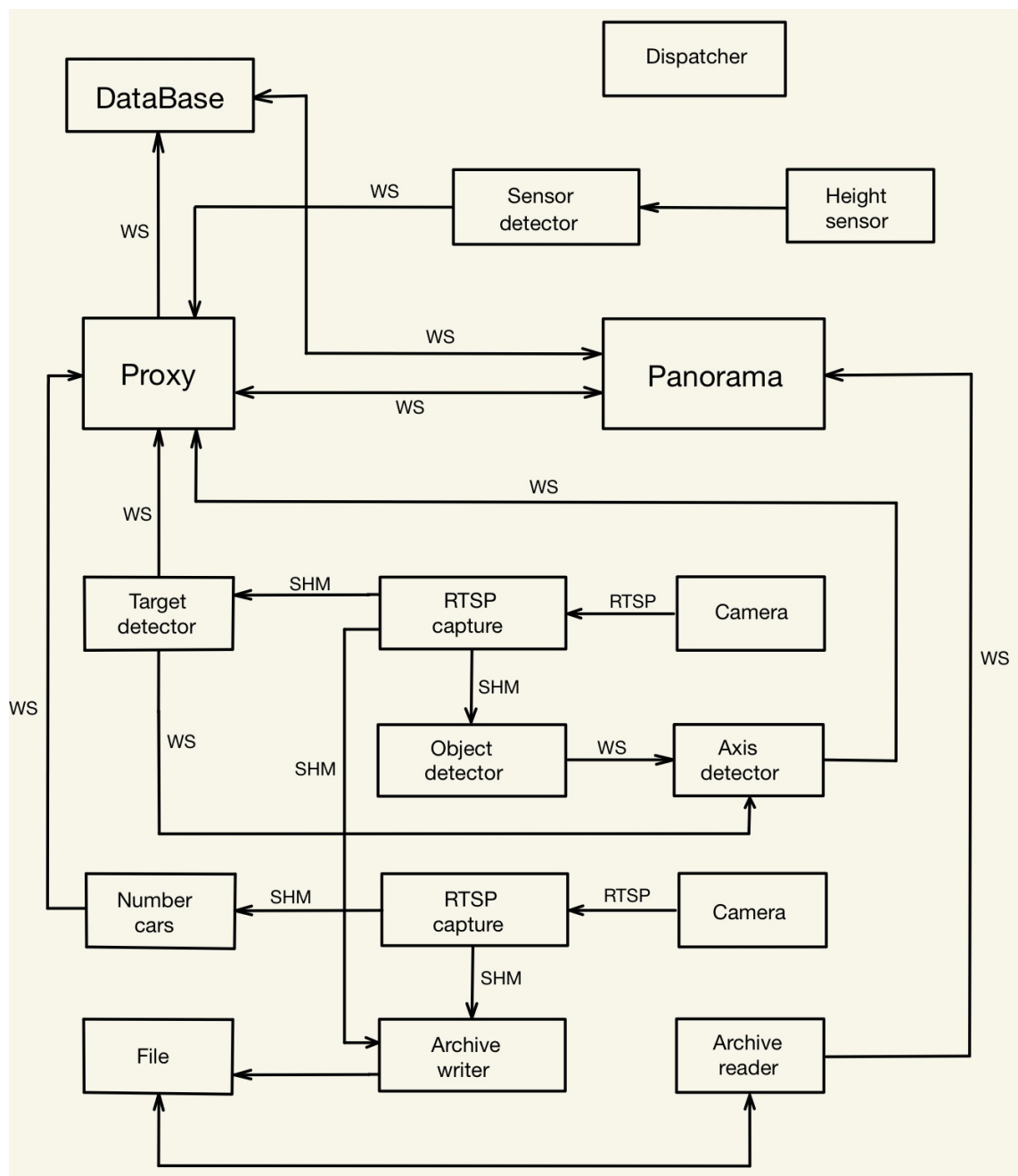


Рис. 4: Схема взаимодействия модулей системы автоматического взимания платы за проезд.

мацию о каждом проезде. Оператор может видеть данные о распознанных ГРЗ, количестве осей, времени проезда и другую полезную

информацию. Интерфейс также предоставляет возможность фильтрации и сортировки данных для удобного анализа и поиска нужной информации.

Система автоматического взимания платы за проезд, построенная на основе предложенной архитектуры, обладает высокой производительностью, точностью и надежностью в распознавании транспортных средств, определении их ГРЗ, подсчете количества осей и формировании соответствующих транзакций. Она облегчает процесс взимания платы за проезд, автоматизирует сбор данных и предоставляет оператору удобный интерфейс для анализа и управления системой.

Используемые модули:

1. `rtsp-capture` – модуль захвата данных с видеокамеры;
2. `number-cars` – модуль детекции и распознавания ГРЗ;
3. `object-detector` – модуль детекции объектов;
4. `axis-detector` – модуль подсчета количества осей ТС;
5. `target-detector` – модуль детекции событий;
6. `dispatcher` – модуль запуска и контроля работы модулей;
7. `proxy` – модуль принятия решений;
8. `postgres-database` – модуль базы данных;
9. `archive-writer` – модуль записи видеопотока с камер;
10. `archive-reader` – модуль чтения видеоархива;
11. `panorama` – модуль интерфейса аналитики.



Рис. 5: Интерфейс системы автоматического взимания платы за проезд, просмотр видео с камер.

Время въезда	Время выезда	Датчик 1 (2.0 м)	Датчик 2 (5.6 м)	Количество осей	Страна (Спереди)	ГРНЗ (Передний)	Страна (Сзади)	ГРНЗ (Задний)	Направление
2023-05-29 10:14:03	2023-05-29 10:14:06	Transport	Transport	2	Россия	P459350 (score: 90)	Россия	P459350 (score: 100)	Выезд в прямом направлении
2023-05-29 10:12:22	2023-05-29 10:12:23	Free	Free	2	Россия	H464XA77 (score: 100)	Россия	H464XA77 (score: 100)	Выезд в прямом направлении
2023-05-29 10:11:09	2023-05-29 10:11:10	Free	Free	2	Россия	Y361CC790 (score: 100)	Россия	Y361CC790 (score: 100)	Выезд в прямом направлении
2023-05-29 10:10:59	2023-05-29 10:11:00	Free	Free	2	Россия	Y399TY799 (score: 100)	Россия	Y399TY799 (score: 100)	Выезд в прямом направлении
2023-05-29 10:10:27	2023-05-29 10:10:29	Free	Free	2	Россия	P848BH68 (score: 100)	Россия	P848BH68 (score: 96)	Выезд в прямом направлении
2023-05-29 10:09:49	2023-05-29 10:09:51	Free	Free	2	Россия	O108CX799 (score: 100)	Россия	O108CX799 (score: 100)	Выезд в прямом направлении
2023-05-29 10:06:52	2023-05-29 10:06:53	Free	Free	2	Россия	B206YX90 (score: 100)	Россия	B206YX90 (score: 100)	Выезд в прямом направлении
2023-05-29 10:06:25	2023-05-29 10:06:26	Free	Free	2	Россия	B333HO198 (score: 100)	Россия	B333HO198 (score: 100)	Выезд в прямом направлении
2023-05-29 10:06:15	2023-05-29 10:06:16	Free	Free	2	Россия	P450AH977 (score: 100)	Россия	P450AH977 (score: 100)	Выезд в прямом направлении
2023-05-29 10:03:41	2023-05-29 10:03:43	Transport	Free	2	Россия	T501XE799 (score: 97)	Россия	T501XE799 (score: 100)	Выезд в прямом направлении
2023-05-29 10:01:12	2023-05-29 10:01:14	Free	Free	2	Россия	T110KE797 (score: 88)	Россия	T110KE797 (score: 100)	Выезд в прямом направлении
2023-05-29 10:00:28	2023-05-29 10:00:29	Free	Free	2	Россия	BH45399 (score: 100)	Россия	BH45399 (score: 100)	Выезд в прямом направлении
2023-05-29 09:59:17	2023-05-29 09:59:18	Free	Free	2	Россия	K339EV777 (score: 100)	Россия	K339EV777 (score: 100)	Выезд в прямом направлении
2023-05-29 09:58:36	2023-05-29 09:58:38	Free	Free	2	Россия	A430KK50 (score: 100)	Россия	A430KK50 (score: 100)	Выезд в прямом направлении
2023-05-29 09:57:16	2023-05-29 09:57:19	Transport	Transport	3	Россия	T201KE750 (score: 99)	Россия	T201KE750 (score: 100)	Выезд в прямом направлении
2023-05-29 09:55:18	2023-05-29 09:55:20	Free	Free	2	Беларусь	6848H11 (score: 100)	Беларусь	6848H11 (score: 100)	Выезд в прямом направлении

Рис. 6: Интерфейс системы автоматического взимания платы за проезд, просмотр событий.

Как показали эксперименты, данная система автоматического взимания платы за проезд, разработанная на основе предложенной архитектуры, является эффективным и надежным решением. Она может быть применена на пунктах взимания платы на дорогах, автостоянках и других местах, где требуется точное распознавание транспортных средств и учет платы за их проезд.

5.7 Система автоматического распознавания транспортных средств

Рассмотрим пример использования предложенной архитектуры для решения задачи автоматического распознавания транспортных средств. Для решения этой задачи необходимо:

1. Детектировать событие проезда ТС;
2. Распознавать ГРЗ транспортного средства;
3. Определять габариты (высоту и ширину) ТС;
4. Подсчитывать количество осей ТС;
5. Формировать транзкцию и записывать ее в базу данных;
6. Отображать события проезда в удобно читаемом интерфейсе.

Первым этапом в работе системы является детектирование события проезда транспортного средства. Ввиду особенностей расположения видеокамеры на этом объекте, есть возможность детектировать и распознавать ТС с помощью модуля `object-detector`. Этот же модуль на кадре сразу детектирует колеса транспортных средств, что в дальнейшем позволяет подсчитать их количество. Кадры с камеры модулям аналитики передает модуль `rtsp-capture`.

После детектирования события проезда система переходит к распознаванию государственного регистрационного знака (ГРЗ) транспортного средства. Для этой задачи используется модуль `number-cars`, который основывается на алгоритмах компьютерного зрения и

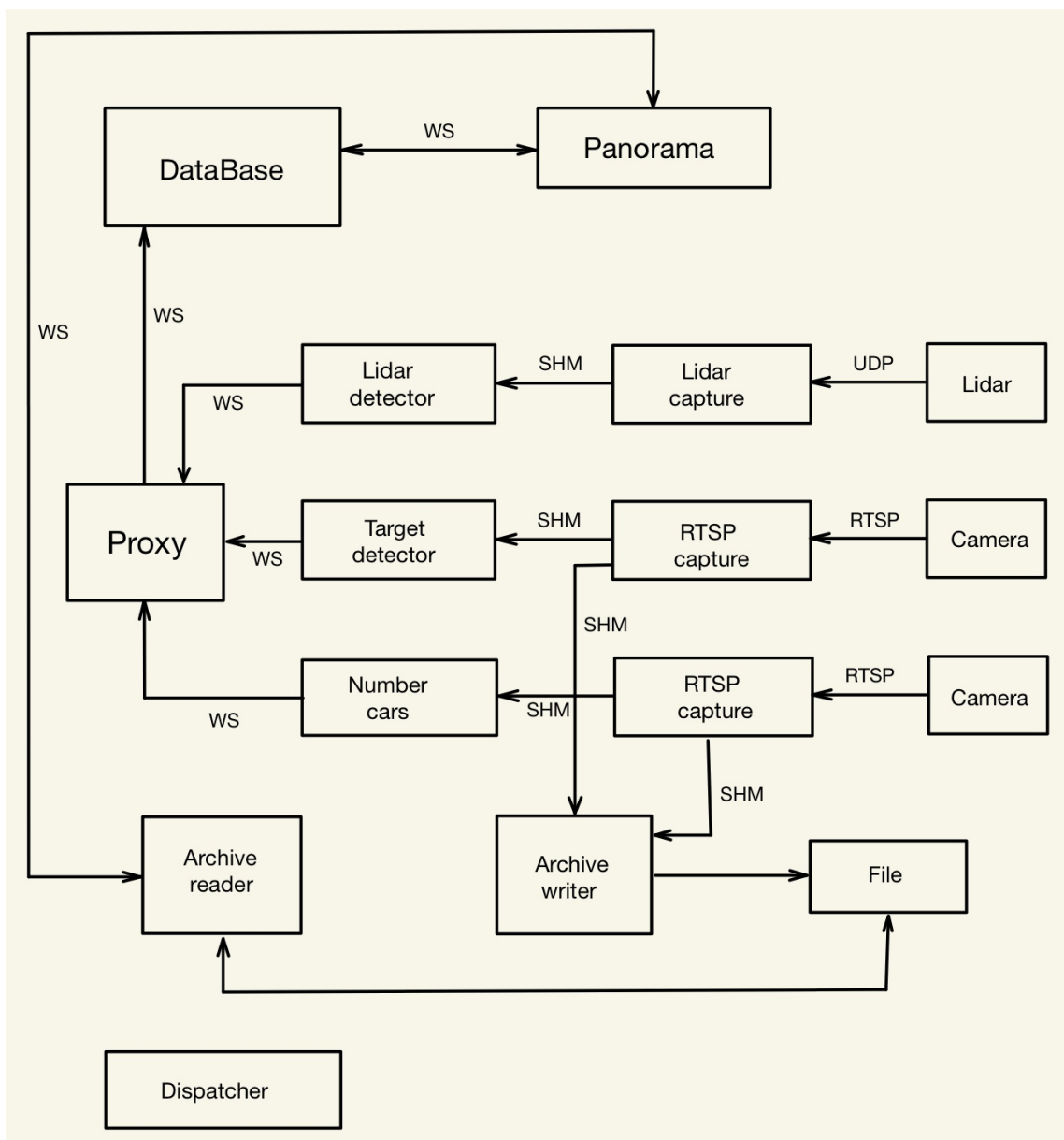


Рис. 7: Схема взаимодействия модулей автоматического распознавания ТС.

машинного обучения. Этот модуль способен распознавать и извлекать информацию с ГРЗ, обеспечивая высокую точность и скорость распознавания.

Для дополнительного анализа транспортных средств система так-

же определяет их габариты, включая высоту и ширину. Для этого используется модуль `lidar-capture` для захвата данных с лидара и модуль `lidar-detector` для анализа этих данных. Лидар предоставляет точные измерения удаленности объектов и позволяет системе определить габариты транспортного средства с высокой точностью.

После завершения анализа транспортного средства система формирует транзакцию, содержащую информацию о произошедшем событии проезда. Эта информация включает в себя данные о распознанном ГРЗ, габаритах, количестве осей и других соответствующих параметрах. Все данные аналитики передаются модулю `prohu`, который формирует транзакции и записывает ее в базу данных.

Для хранения и управления данными системы используется модуль `postgres-database`, который представляет собой базу данных PostgreSQL. Этот модуль обеспечивает запись и хранение информации о распознанных транспортных средствах, ГРЗ, габаритах, количестве осей и других связанных данных.

Для отображения и представления результатов аналитики система использует модуль `rapograma`, который предоставляет удобный интерфейс для визуализации данных. Через этот интерфейс оператор может просматривать статистику по проездам транспортных средств, результаты распознавания ГРЗ, информацию о габаритах и количестве осей, а также другие сводные данные, представленные в понятной и информативной форме.

Так же, как и в предыдущем случае, работают модули `dispatcher`, `archive-writer` и `archive-reader`.

Система автоматического распознавания транспортных средств, основанная на предложенной архитектуре, обладает высокой точностью и надежностью в решении соответствующих задач. Модули системы работают в синхронизации, обеспечивая автоматическую обработку видеоданных и предоставляя оператору полную и точную информацию о проездах транспортных средств. Результаты работы системы могут быть использованы для контроля транспортного потока, определения нарушений правил дорожного движения, учета проезжающих транспортных средств и других приложений, где требуется автоматизированное распознавание и анализ данных о транс-

портных средствах.

Система автоматического распознавания транспортных средств, основанная на предложенной архитектуре, успешно применяется на пунктах взимания платы на федеральных автомобильных дорогах М-1 и М-4. Она обеспечивает подсчет количества осей проезжающих автомобилей, распознавание государственных регистрационных знаков и классификацию транспортных средств. Также система успешно применяется на федеральной трассе М-10 для распознавания транспортных средств, определения габаритов и подсчета их количества.

На рисунке 8 представлен пример интерфейса системы. Он позволяет оператору наглядно просматривать и анализировать данные о проездах, распознанных ГРЗ, габаритах и количестве осей транспортных средств.

Используемые модули:

1. `rtsp-capture` – модуль захвата данных с видеокамеры;
2. `object-detector` – модуль детекции объектов;
3. `lidar-capture` – модуль захвата данных с лидара;
4. `lidar-detector` – модуль анализа данных с лидара;
5. `number-cars` – модуль детекции и распознавания ГРЗ;
6. `dispatcher` – модуль запуска и контроля работы модулей;
7. `proxy` – модуль принятия решений;
8. `postgres-database` – модуль базы данных;
9. `archive-writer` – модуль записи видеопотока с камер;
10. `archive-reader` – модуль чтения видеоархива;
11. `panorama` – модуль интерфейса аналитики.

Время проезда	Направление	ГРНЗ	Страна	Тип ГРНЗ	Количество осей	Кадр (ГРНЗ)	Класс ТС (габариты)	ETC/N28	Кадр (Спереди)	Кадр (Сзади)	Полоса движения	Скорость (км/ч)	Длина ТС (м.)	Ширина ТС (м.)	Высота ТС (м.)	12-тонный грузовой	
2023-05-12 09:33:43.655	Проезд в прямом направлении	O474TK198 (85)	Россия	тип 1	6		artic_tanker (41)					1	85.94	13.55	2.52	3.99	Да
2023-05-12 09:33:41.656	Проезд в прямом направлении	P254CK69 (100)	Россия	тип 1	2		car (92)					1	86.37	3.84	1.47	1.4	Нет
2023-05-12 09:33:36.923	Проезд в прямом направлении	H883TY27 (75)	Россия	тип 1	2		car (94)					1	91.54	4.45	1.7	1.67	Нет
2023-05-12 09:33:35.122	Проезд в прямом направлении	E747AP799 (72)	Россия	тип 1	5		artic_container (63)					1	88.68	17.45	2.66	4	Да
2023-05-12 09:33:27.189	Проезд в прямом направлении	Y608KO193 (36)	Россия	тип 1A	2		car (91)					1	83.66	2.91	1.38	1.51	Нет
2023-05-12 09:33:24.466	Проезд в обратном направлении	X185AE774 (100)	Россия	тип 1	5		artic_container (58)					3	80.72	18.15	2.67	4.02	Да
2023-05-12 09:33:22.599	Проезд в обратном направлении	E307CK750 (100)	Россия	тип 1	5		artic_container (70)					3	82.61	18.12	2.61	4.02	Да
2023-05-12 09:33:18.266	Проезд в обратном направлении	M079BV797 (100)	Россия	тип 1	2		van_comm (46)					3	86.65	7.61	2.29	3.12	Нет
2023-05-12 09:33:09.933	Проезд в обратном направлении	H917YH32 (100)	Россия	тип 1	2		car (93)					3	76.24	3.71	1.53	1.33	Нет
2023-05-12 09:33:08.199	Проезд в обратном направлении	K374KO198 (100)	Россия	тип 1	5		artic_container (78)					3	75.01	17.56	2.6	3.98	Да
2023-05-12 09:33:06.332	Проезд в обратном направлении	T664MA790 (100)	Россия	тип 1	2		truck_container (93)					3	77.88	9.42	2.63	3.55	Возможно
2023-05-12 09:32:36.866	Проезд в обратном направлении	C275PK69 (100)	Россия	тип 1	2		car (91)					3	81.63	3.84	1.45	1.43	Нет
2023-05-12 09:32:33.333	Проезд в обратном направлении	K490OC69 (100)	Россия	тип 1	2		car (93)					3	76.26	3.9	1.56	1.34	Нет
2023-05-12 09:32:31.466	Проезд в обратном направлении	P981XK174 (100)	Россия	тип 1	5		artic_container (68)					3	79.59	18.78	2.58	3.99	Да
2023-05-12 09:32:29.599	Проезд в обратном направлении	X594OO123 (87)	Россия	тип 1	5		artic_container (63)					3	78.56	18.89	2.68	4.08	Да

Рис. 8: Интерфейс системы автоматического распознавания ТС.

Как показали проведенные эксперименты, система автоматического распознавания транспортных средств, разработанная на основе предложенной архитектуры, представляет собой эффективное решение для автоматизации и оптимизации процесса распознавания и анализа данных о транспортных средствах. Модули системы взаимодействуют между собой, обеспечивая высокую точность и надежность работы. Результаты, полученные с помощью этой системы, могут быть использованы в различных областях, таких как управление дорожным движением, контроль безопасности, учет транспорта и другие.

5.8 Система автоматического распознавания лиц и подсчета людей

Рассмотрим пример использования предложенной архитектуры для решения задачи распознавания лиц и подсчета людей. Для решения этой задачи необходимо:

1. Детектировать лица в кадре;

2. Распознавать лица;
3. Подсчитывать количество людей;
4. Формировать транзкцию и записывать ее в базу данных;
5. Отображать события в удобно читаемом интерфейсе.

Первым этапом в решении задачи распознавания лиц и подсчета людей является детектирование лиц в кадре, которые формирует и передает по shared memory модуль захватчика rtsp-capture. Для детекции лиц система использует модуль object-detector, который способен обнаруживать объекты, в данном случае лица, на кадрах.

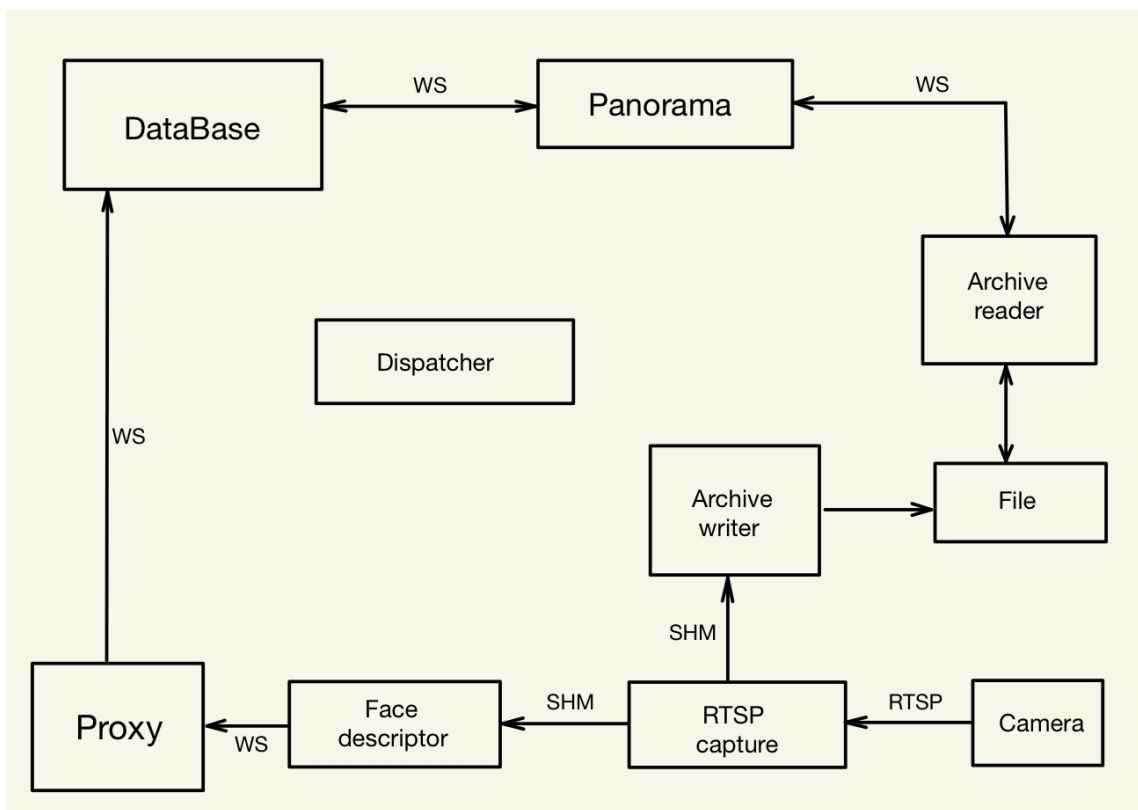


Рис. 9: Схема взаимодействия модулей системы автоматического распознавания лиц и подсчета людей.

После успешного детектирования лиц система переходит к следующему этапу – распознаванию лиц. Для этой задачи используется

модуль face-descriptor, который применяет алгоритмы компьютерного зрения и глубокого обучения для извлечения характеристик и уникальных признаков лиц. Эти признаки затем используются для сравнения и идентификации лиц в базе данных или среди предварительно известных лиц.

Для хранения и управления данными системы используется модуль postgres-database, который представляет собой базу данных PostgreSQL. Этот модуль обеспечивает сохранение информации о распознанных лицах, событиях проезда и других связанных данных. Он также позволяет осуществлять поиск, фильтрацию и агрегацию данных для более детального анализа.

Наконец, для отображения и представления результатов аналитики система использует модуль рапогама, который предоставляет удобный интерфейс для визуализации данных. Через этот интерфейс оператор может просматривать статистику, результаты распознавания лиц, а также другие сводные данные, представленные в понятной и информативной форме.

Система распознавания лиц и подсчета людей, основанная на предложенной архитектуре, обладает высокой точностью и эффективностью в решении соответствующих задач. Модули системы взаимодействуют между собой, обеспечивая надежную и автоматизированную обработку видеоданных. Результаты работы системы могут быть использованы для различных целей, таких как обеспечение безопасности, контроль доступа, аналитика посещаемости и другие приложения, где важны распознавание лиц и подсчет посетителей.

Используемые модули:

1. rtsp-capture – модуль захвата данных с видеокамеры;
2. object-detector – модуль детекции объектов;
3. face-descriptor – модуль распознавания лиц;
4. dispatcher – модуль запуска и контроля работы модулей;
5. proxy – модуль принятия решений;

6. postgres-database – модуль базы данных;
7. panorama – модуль интерфейса аналитики.



































Дата	Время	Камера	Аналитика			Описание
2023-05-12	12:20:30	Glosav-IR100-1	Главная камера			Dasha
2023-05-12	12:20:30	Glosav-IR100-1	Главная камера			Dasha
2023-05-12	12:20:30	Glosav-IR100-1	Главная камера			Dasha
2023-05-12	12:20:30	Glosav-IR100-1	Главная камера			Dasha
2023-05-12	12:19:41	Glosav-IR100-1	Главная камера			Dasha
2023-05-12	12:19:02	Glosav-IR100-1	Главная камера			Misha
2023-05-12	12:19:02	Glosav-IR100-1	Главная камера			Misha
2023-05-12	12:19:01	Glosav-IR100-1	Главная камера			Misha
2023-05-12	12:19:01	Glosav-IR100-1	Главная камера			Misha
2023-05-12	12:19:01	Glosav-IR100-1	Главная камера			Misha
2023-05-12	12:19:01	Glosav-IR100-1	Главная камера			Misha
2023-05-12	12:18:47	Glosav-IR100-1	Главная камера			Misha
2023-05-12	12:18:47	Glosav-IR100-1	Главная камера			Misha
2023-05-12	12:18:47	Glosav-IR100-1	Главная камера			Misha
2023-05-12	12:18:46	Glosav-IR100-1	Главная камера			Misha
2023-05-12	12:18:46	Glosav-IR100-1	Главная камера			Misha
2023-05-12	12:18:46	Glosav-IR100-1	Главная камера			Misha

Рис. 10: Интерфейс системы распознавания лиц.

6 Заключение

В работе представлена система видеоаналитики, которая успешно решает задачи классификации, детекции и трекинга на основе свёрточных нейронных сетей. Было уделено внимание возможности системы работать автономно с минимальными вычислительными и энергетическими затратами, что делает её применимой для использования на мини-компьютерах, таких как NVIDIA Jetson или Firefly Rockchip.

Эксперименты показали эффективность предложенной системы в задачах подсчёта количества осей проезжающих автомобилей, распознавания государственных регистрационных знаков, классификации транспортных средств, определение габаритов транспортных средств, что позволило внедрить и успешно применять разработанную систему видеоаналитики на различных объектах.

В данной работе была поставлена задача разработки модульной системы видеоаналитики для повышения эффективности обработки и анализа видеоданных. Была проведена обширная аналитическая работа, в ходе которой были изучены существующие методы обработки и анализа видеоданных, включая компьютерное зрение, машинное обучение и глубокое обучение, а также основные задачи видеоаналитики, такие как распознавание объектов, трекинг движущихся объектов, определение сцен и обнаружение аномалий.

На основе проведенного анализа была разработана модульная архитектура системы видеоаналитики, состоящая из независимых модулей, способных выполнять различные задачи видеоаналитики. Такой подход позволяет гибко настраивать систему под конкретные требования и задачи, а также распределить вычислительные нагрузки между модулями, повышая эффективность обработки видеоданных.

Были предложены методы передачи данных между модулями системы с минимальной потерей информации и максимальной эффективностью использования ресурсов. Это позволяет сохранить высокое качество видеоданных и обеспечить эффективное использование вычислительных ресурсов.

На основе разработанной архитектуры и методов передачи дан-

ных был реализован прототип системы видеоаналитики. Проведенное экспериментальное исследование прототипа позволило оценить его производительность, точность и стабильность работы на реальных видеоданных. Результаты экспериментов подтверждают эффективность разработанной модульной системы видеоаналитики.

Сравнение разработанной модульной системы видеоаналитики с существующими методами и системами показало ее преимущества в терминах эффективности обработки видеоданных, точности анализа и гибкости настройки. Модульность системы позволяет добавлять новые модули, оптимизировать методы передачи данных и улучшать алгоритмы обработки и анализа видеоданных для дальнейшего развития и улучшения системы.

В целом, разработка модульной системы видеоаналитики представляет значимый вклад в область обработки и анализа видеоданных. Ее применение может быть полезным в различных сферах, таких как безопасность, видеонаблюдение, транспорт, медицина и другие.

Результаты данной работы подтверждают, что модульная система видеоаналитики обладает рядом преимуществ, включая гибкую конфигурацию, эффективное распределение вычислительных нагрузок, высокую точность анализа и эффективную передачу данных между модулями. Это позволяет системе эффективно обрабатывать и анализировать видеоданные с различных источников, а также приспособляться к различным задачам видеоаналитики.

Однако, разработанная модульная система видеоаналитики также имеет потенциал для дальнейшего развития и усовершенствования. Возможности добавления новых модулей, оптимизации методов передачи данных и улучшения алгоритмов обработки и анализа видеоданных могут привести к дальнейшему улучшению производительности и точности системы.

В целом, разработка модульной системы видеоаналитики представляет перспективную область исследований, с применением которой можно достичь более эффективной обработки и анализа видеоданных. Результаты данной работы могут быть использованы в дальнейших исследованиях и разработках в области видеоаналитики.

ки, а также применены в практических приложениях, где требуется автоматизированная обработка и интерпретация видеоинформации.

Таким образом, разработанная система видеоаналитики на основе свёрточных нейронных сетей представляет собой эффективный инструмент для решения задач компьютерного зрения. В будущем можно рассмотреть возможность дальнейшего развития системы с целью её улучшения и расширения функционала.

Список литературы

- [1] Simonyan, Karen и Andrew Zisserman: *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556, 2014.
- [2] Krizhevsky, Alex, Ilya Sutskever, и Geoffrey E Hinton: *Imagenet classification with deep convolutional neural networks*. Communications of the ACM, 60(6):84–90, 2017.
- [3] He, Kaiming, Xiangyu Zhang, Shaoqing Ren, и Jian Sun: *Deep residual learning for image recognition*. В *Proceedings of the IEEE conference on computer vision and pattern recognition*, страницы 770–778, 2016.
- [4] Goodfellow, Ian, Yoshua Bengio, и Aaron Courville: *Deep learning*. MIT press, 2016.
- [5] Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, и Zbigniew Wojna: *Rethinking the inception architecture for computer vision*. В *Proceedings of the IEEE conference on computer vision and pattern recognition*, страницы 2818–2826, 2016.
- [6] Redmon, Joseph, Santosh Divvala, Ross Girshick, и Ali Farhadi: *You only look once: Unified, real-time object detection*. В *Proceedings of the IEEE conference on computer vision and pattern recognition*, страницы 779–788, 2016.
- [7] Redmon, Joseph и Ali Farhadi: *Yolov3: An incremental improvement*. arXiv preprint arXiv:1804.02767, 2018.
- [8] Jiang, Zicong, Liquan Zhao, Shuaiyang Li, и Yanfei Jia: *Real-time object detection method based on improved YOLOv4-tiny*. arXiv preprint arXiv:2011.04244, 2020.
- [9] Huang, Gao, Zhuang Liu, Laurens Van Der Maaten, и Kilian Q Weinberger: *Densely connected convolutional networks*. В

Proceedings of the IEEE conference on computer vision and pattern recognition, страницы 4700–4708, 2017.

- [10] Ren, Shaoqing, Kaiming He, Ross Girshick, и Jian Sun: *Faster r-cnn: Towards real-time object detection with region proposal networks*. *Advances in neural information processing systems*, 28, 2015.
- [11] Lin, Tsung Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, и Serge Belongie: *Feature pyramid networks for object detection*. В *Proceedings of the IEEE conference on computer vision and pattern recognition*, страницы 2117–2125, 2017.
- [12] *NVIDIA Metropolis*. <https://clck.ru/34p9Ji>. Дата обращения 29 июня 2023 г..
- [13] *Hikvision Software*. <https://www.hikvision.com/en/products/software>. Дата обращения 29 июня 2023 г..
- [14] *Milestonesys video-technology*. <https://goo.su/od1ab>. Дата обращения 29 июня 2023 г..
- [15] *Avigilon*. <https://www.avigilon.com/products>. Дата обращения 29 июня 2023 г..
- [16] Wojke, Nicolai, Alex Bewley, и Dietrich Paulus: *Simple online and realtime tracking with a deep association metric*. В *2017 IEEE international conference on image processing (ICIP)*, страницы 3645–3649. IEEE, 2017.