

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»

Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

РАБОТА ПРОВЕРЕНА

Консультант

_____ П.О. Шабуров
«___» _____ 2023 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой ЭВМ

_____ Д.В. Топольский
«___» _____ 2023 г.

РАЗРАБОТКА СИСТЕМЫ ВИЗУАЛИЗАЦИИ ПАРКОВОЧНЫХ МЕСТ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ
ЮУРГУ-090301.2023.201 ПЗ ВКР

Руководитель работы,

к.пед.н., доцент каф. ЭВМ

_____ Ю.Г. Плаксина
«___» _____ 2023 г.

Автор работы,

студент группы КЭ-406

_____ Д.Е. Бабыкина
«___» _____ 2023 г.

Нормоконтролёр,

ст. преп. каф. ЭВМ

_____ И.Г. Сяськов
«___» _____ 2023 г.

Челябинск-2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Южно-Уральский государственный университет
(национальный исследовательский университет)»
Высшая школа электроники и компьютерных наук
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

_____ Д.В. Топольский

«___» _____ 2023 г.

ЗАДАНИЕ

на выпускную квалификационную работу бакалавра

студенту группы КЭ-406

Бабыкиной Дарье Евгеньевне,

обучающемуся по направлению

09.03.01 «Информатика и вычислительная техника»

1. Тема работы: «Разработка системы визуализации парковочных мест»
утверждена приказом по университету № 753-13/12 от «25» апреля 2023 г.

2. Срок сдачи студентом законченной работы: «1» июня 2023 г.

3. Исходные данные к работе:

- сетевая камера DH-IPC-HFW2431SP-S-0360B с питанием PoE 12V ± 30%;
- инжектор Ubiquiti Networks POE-48-24W-G.

4. Перечень подлежащих разработке вопросов:

- 1) аналитический обзор научно-технической и методической литературы по тематике работы, поиск аналогичных решений;
- 2) определение требований, предъявляемых к проекту;
- 3) проектирование технической и программной архитектур системы;
- 4) разработка системы визуализации парковочных мест;
- 5) составление программы тестовых испытаний, с последующим их проведением;

б) оценка пути дальнейшего развития и масштабирования системы.

5. Дата выдачи задания: « 2 » декабря 2022 г.

Руководитель работы _____ / *Ю.Г. Плаксина* /

Студент _____ / *Д.Е. Бабыкина* /

КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Введение и обзор литературы	01.03.2023	
Определение требований, предъявляемых к проекту. Проектирование технической и программной архитектур системы	01.04.2023	
Разработка системы визуализации парковочных мест	01.05.2023	
Проведение тестовых испытаний; Оценка пути дальнейшего развития и масштабирования системы	15.05.2023	
Компоновка текста работы и сдача на нормоконтроль	24.05.2023	
Подготовка презентации и доклада	30.05.2023	

Руководитель работы _____ / Ю.Г. Плаксина /

Студент _____ / Д.Е. Бабыкина /

АННОТАЦИЯ

Д.Е. Бабыкина. Разработка системы мониторинга доступности парковочных мест. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШ ЭКН; 2023, 72 с., 43 ил., библиогр. список – 32 наим.

Ключевые слова: автоматизированная парковка, web-приложение, анализ видеоизображения, отслеживание парковочных мест.

В рамках выпускной квалификационной работы производится проектирование и реализация системы визуализации парковочных мест, для предоставления помощи водителям при ориентировании на парковочном пространстве.

Объектом исследования является организация комфортного перемещения автомобилей на крупных парковках.

Целью выпускной квалификационной работы является разработка системы визуализации парковочных мест.

В работе поднимается проблема поиска свободных парковочных мест на крупных парковках. Предлагаемый вариант решения позволит водителям ускорить поиск свободного места. Разработанная система визуализации парковочного пространства, являющаяся частью программного комплекса «EasyParking», и включает в себя два модуля. Модуль распознавания наличия автомобилей на парковочных местах. Модуль интерфейсного взаимодействия пользователей с системой. Работы по тестированию проводилась на реальном объекте – автомобильной придомовой парковке.

Область применения: крупные автомобильные парковки.

Данная работа соответствует приоритетным направлениям научно-технического развития страны.

ОГЛАВЛЕНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	8
ВВЕДЕНИЕ.....	9
1 АНАЛИТИЧЕСКИЙ ОБЗОР СОВРЕМЕННОЙ НАУЧНО-ТЕХНИЧЕСКОЙ И МЕТОДИЧЕСКОЙ ЛИТЕРАТУРЫ.....	11
1.1 Обзор методов для решения задачи распознавания изображения.....	12
1.2 Обзор аналогичных решений для достижения заданной цели.....	17
2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ	40
2.1 Функциональные требования	40
2.2 Нефункциональные требования	44
3 ПРОЕКТИРОВАНИЕ	46
3.1 Алгоритмы работы системы	46
3.2 Архитектура разрабатываемого решения.....	50
4 РЕАЛИЗАЦИЯ	53
4.1 Подбор комплектующих	53
4.2 Выбор стека программного обеспечения	56
4.3 Результаты разработки программного кода	60
5 СОСТАВЛЕНИЕ ПРОГРАММЫ ТЕСТОВЫХ ИСПЫТАНИЙ.....	61
5.1 Контрольное задание 1	61
5.2 Контрольное задание 2	62
5.3 Контрольное задание 3	63
5.4 Контрольное задание 4	65

6 ОЦЕНКА ПУТИ ДАЛЬНЕЙШЕГО РАЗВИТИЯ И МАСШТАБИРОВАНИЯ СИСТЕМЫ.....	67
ЗАКЛЮЧЕНИЕ	68
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	69

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

Администратор – оператор системы визуализации парковочных мест

ИВ – интернет вещей

ИИ – искусственный интеллект

КЗ – контрольное задание

ОС – операционная система

ПАК – программно-аппаратный комплекс

ПО – программное обеспечение

Пользователь – участник дорожного движения, непосредственно взаимодействующий с элементами интерфейса системы умной парковки

СУБД – это набор программ, позволяющий организовывать, контролировать и администрировать базы данных

IoT (от англ. Internet of Things) – интернет вещей

RTSP (от англ. Real Time Streaming Protocol) – потоковый протокол реального времени

TCP (от англ. Transmission Control Protocol) – протокол управления передачей

UML (от англ. Unified Modeling Language) – унифицированный язык моделирования

ВВЕДЕНИЕ

Для жителей крупных мегаполисов проблема с парковкой автотранспорта является насущной. Очень часто водитель попадает на крупную парковку, что ставит его в затруднительное положение, а именно поиск свободного места. Подобная ситуация может возникнуть, так как водитель не владеет полной информацией о свободных парковочных местах, сидя за рулем автомобиля, что побуждает автовладельца блуждать на автомобиле по всей парковке.

Современный водитель обладает достаточным уровнем цифровизации дорожного движения, так как, в своем большинстве, использует потребительские электронные устройства, такие как современные мультимедийные автомобильные системы с поддержкой Google Android Auto и Apple CarPlay, а также смартфон, что позволяет ему активно взаимодействовать как с цифровым окружением, так и с другими водителями. Поэтому таким автовладельцам можно достаточно просто представить информацию о дорожной ситуации в цифровом виде. Однако существуют категории автовладельцев, которые не являются активными пользователями современных гаджетов. Как только в образ каждого водителя окончательно войдет цифровая составляющая, то в качестве естественного помощника дорожного движения сможет выступать цифровой интерфейс – цифровой двойник дорожного пространства, таким образом общество сможет сделать ещё один шаг к внедрению полностью автоматизированному дорожному движению.

Однако на данном этапе эволюционного развития дорожного движения подспорьем для водителя могут выступать более простые сервисы, такие как цифровые табло, web-интерфейсы приложений навигаторов, голосовые ассистенты.

Гипотеза исследования: применение современных технологий web-программирования и компьютерного зрения существенно повысит удобство использования городских парковочных пространств, позволит сократить временные затраты участников дорожного движения.

Целью исследования является разработка системы визуализации парковочных мест.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести аналитический обзор научно-технической и методической литературы по тематике работы, а также найти аналогичные решения поставленной цели;

- определить перечень требований, предъявляемых к разрабатываемому проекту;

- спроектировать два типа архитектур системы: техническую и программную;

- разработать систему визуализации парковочных мест для отслеживания состояния занятости парковочных мест;

- составить программу тестовых испытаний, с последующим их проведением;

- оценить пути дальнейшего развития и масштабирования системы.

Объектом исследования является информационная система для автоматизации работы парковочных зон. Предметом исследования является web-сервис для анализа видеотрансляции, передаваемого с записывающего устройства, и отслеживания состояния парковочных мест.

Научная новизна и теоретическая значимость исследования: работа продолжает развивать направление применения информационных технологий для оптимизации качества эксплуатации парковочных площадей.

1 АНАЛИТИЧЕСКИЙ ОБЗОР СОВРЕМЕННОЙ НАУЧНО-ТЕХНИЧЕСКОЙ И МЕТОДИЧЕСКОЙ ЛИТЕРАТУРЫ

«Компьютерным зрением называется преобразование данных, полученных от фото или видеокамеры, результатом которого является некоторое решение или новое представление» [1]. Данные преобразовываются ради достижения определенной цели. Так, целью компьютерного зрения можно назвать формирование полезных выводов применительно к объектам и сценам реального мира на основе применения анализа изображения, полученного с помощью периферийных устройств сбора данных.

Контекстная информация возможна для включения в исходные данные, например «камера установлена на учреждения» или «известно, что дальномер показывает расстояние до объекта равное 1 метр». Выходные данные могут принять вид «на этой сцене присутствует человек» или «на этой парковке 42 возможных места». Под новым представлением может пониматься исключение влияния естественного движения камеры из последовательности изображений или преобразование цветного изображения в градацию серого [2].

Для системы машинного зрения компьютер является субъектом сбора последовательности чисел от камеры или с накопительного диска. Для системы машинного зрения входными данными является матрица чисел. Реальность такова, что довольно сильный шум примешан к любому числу матрицы, поэтому само по себе число дает мало информации. Одна из задач машинного зрения это преобразование зашумленной числовой матрицы в образ реальной сцены. Получаемые искажения или шум обусловлены естественной изменчивостью внешней среды (например, перемещения, погода, отражения, освещение), несовершенством оптических и механических устройств, определенной дискретностью времени регистрации сенсоров (размытость из-за движения), присутствующими электрическими шумами в датчиках или других электронных приборах, а также артефактами захваченного изображения, получаемыми после

сжатия. При проектировании реальной системы часто можно воспользоваться знаниями о контексте для преодоления ограничения зрительных сенсоров.

Методы машинного обучения позволяют явно моделировать контекстную информацию. Зная о следующих скрытых переменных, как размер, положение в пространстве и прочее, в последствии можно произвести корреляцию со значениями в заданном обучающем наборе. Существует и другой вариант – можно прибегнуть к попытке измерить скрытые переменные с помощью дополнительных устройств, например, используя лазерный дальномер для измерения глубины объекта, что позволит точно узнать его размер. Не стоит забывать о другой проблеме – шумы или помехи. Зачастую, чтобы устранить шумы применяют статистические методы. Явно отражающим примером послужит отсутствие возможности распознать границу в изображении, просто сравнив выбранную точку с ее ближайшими соседями, однако если принять к сведению статистическое распределение точек в локальной области, то это позволит значительно упростить распознавание границ изображения. Таким образом истинная граница определяется как последовательность точек, для которых ориентация локальных областей должна быть согласована с соседями. Другим способом компенсации шумов является вычисление статистики по времени. Отметим, что существуют методы подавления помех, основанные на построении явных моделей, обучаемых по имеющимся данным.

1.1 Обзор методов для решения задачи распознавания изображения

Распространенным методом обработки видео в компьютерном зрении является вычитание фона. Областью интереса (ROI – region of interest) в видео может быть несколько объектов: все транспортные средства и люди. Основная цель вычитания фона – это получение последовательных кадров с камеры, обнаружение всех объектов переднего плана как разницы между текущим кадром и изображением статичного фона. Можно выделить множество методов вычитания фона [3], у которых возможно выделить различные сильные и слабые стороны с точки зрения вычислительных требований и производительности. С

изменениями освещения, повторяющимися движениями и изменениями погодных условий справится только надежный алгоритм вычитания фона [4]. Использование одного изображения в качестве фона в градациях серого/цветного, не содержащее движущихся объектов, является наиболее простым способом по созданию фона [5]. Для применения этого метода кадр должен быть сделан в отсутствие движения, или же сделан с помощью временного медианного фильтра.

Группа методов, называемых обычными, включает в себя несколько основных подходов, таких как разница кадров, статическая разница кадров, взвешенное скользящее среднее, временное среднее, взвешенная скользящая дисперсия, адаптивное фоновое обучение, адаптивная медиана Макфарлейна и Шофилда [6], временная медиана Куккиара и др. [5] и Кальдерара и др. [7]. Эти методы могут быть рекурсивными или не рекурсивными подходами.

Разность кадров является одним из самых простых методов вычитания фона. Метод прост – в нем предыдущий кадр считается фоном. Из этого следует, что целевой объект определяется путем вычитания текущих рамок фоновой модели из предыдущей [8, 9]. Для определения разности кадров в момент времени $t + 1$, он вычитывается, а фоном считается кадр в момент времени t . Разностное изображение покажет только интенсивность для местоположений пикселей, которые изменились за смену кадра. Если все пиксели переднего плана движутся, а все пиксели фона статичны, то этот подход сработает, в иных случаях он не может быть применён [5, 10].

Одним из наиболее часто используемых алгоритмов способов вычитания фона является медианная фильтрация. В его основе лежит оценка модели фона при помощи вычисления медианного значения каждого входного пикселя. Объект перестает считаться фоном, если он прошел более половины кадра. Одним из главных преимуществ этого способа является простота конструкции метода, скорость процесса и простота использования. Отметим, что модели и фон не фиксированы, поэтому они меняются со временем. Недостатками обоих

подходов являются зависимость точности от скорости движения цели, а также частые сбои при отслеживании целей на анимированных фонах.

В методе фильтрации MIN-MAX используются три разных значения, для того чтобы определить какой именно пиксель является частью фоновой модели. Мишень снижает интенсивность точек на фоне, в течение определенного промежутка времени [11]. Харитаоглу и др. [12] предложили другой метод, чтобы достичь цели локальной адаптации к шуму. Этот метод включает такие следующие параметры: каждый фоновый пиксель имеет максимальную M_s , минимальную m_s и максимальную разность последовательных кадров D_s , наблюдаемую в обучающей последовательности. Отметим, что большинство схем для определения отношения вклада прошлых наблюдений, которое следует ниже, используют факторы забывания или экспоненциальное взвешивание. Их можно использовать для вычитания и оценки фона [11].

Для вычисления среднего изображения последовательности кадров используется одиночная гауссовская фильтрация. Затем вычитание каждого нового входного кадра и проверка значений разности по заранее определенному порогу – один из самых простых методов вычитания фона [13]. Алгоритм, представленный Реном [11], используется для нормального распределения со стандартным отклонением и определенным средним значением для каждого оцениваемого фонового пикселя с использованием цветового пространства, называемого цветовым пространством YUV.

Макфарлейн и Шофилд [6] представили простой рекурсивный фильтр для оценки медианы каждого пикселя во времени – приближенная медианная фильтрация. Этот метод был принят некоторыми подходами и предназначается для вычитания фона при отслеживании состояния городского трафика из-за его значительной скорости.

Методом, который является одним из самых известных рекурсивных методов, является фильтрация Калмана. Если мы предположим, что значения интенсивности пикселей в изображении подчиняются нормальному распределению, например, когда простые адаптивные фильтры отвечают за

обновление среднего значения и дисперсии модели фона, чтобы компенсировать изменения освещения и включить объекты с длительными остановками в модели фона. Оценка фона с использованием фильтрации Калмана была объяснена в [9]. Основное различие между ними заключается в используемом пространстве состояний для отслеживания процесса. Самыми простыми являются те, которые основаны только на яркости [5, 9, 10].

Все упомянутые модели способны приспосабливаться к постепенным изменениям освещения. Однако, если происходят значительные изменения интенсивности, все они сталкиваются с серьезными проблемами. Модель Маркова – метод, который может моделировать изменения интенсивности пикселей. Он пытается смоделировать эти вариации как дискретные состояния, основанные на режимах окружающей среды, например, облачное/солнечное небо или включение/выключение света. НММ с тремя состояниями был показан для моделирования интенсивности пикселя в приложениях мониторинга трафика [14].

Статистические методы.

Моделирование фона одним изображением, как и в обычных методах, требует строго фиксированного фона, свободного от шумов. Поскольку это требование не может быть выполнено в постоянном режиме в реальной жизни, многие модели с каждым фоновым пикселем в функции плотности вероятности (PDF – probability density function) обучались на серии обучающих кадров. Статистические методы, использующие одну гауссиану, имеют две подпоследовательности: гауссово среднее было предложено Реном [11]. Он не справляется с мультимодальными фонами. Многие исследователи работали над статистическими методами с использованием нескольких гауссианов, которые называются гауссовской смешанной моделью (GMM – Gaussian Mixture Model). Некоторые из этих исследований были проведены Штауффером и Гримсоном [4], Живковичем [15]. Для учета фонов из анимированных текстур (таких как деревья, раскачиваемые ветром) некоторые авторы предложили использовать

мультимодальные функции плотности вероятности, такие как метод Штауффера и Гримсона [16].

Непараметрические методы.

Эльгаммаль и его коллеги предложили алгоритм оценки плотности ядра (KDE – Kernel Density Estimation). Неструктурированный подход также можно использовать для моделирования мультимодальной PDF. С этой точки зрения Эльгаммаль предложил оценку окна Парцена для каждого фонового пикселя. Проблема этого метода заключается в размере требования к памяти, времени для вычисления значений ядра. Также могут быть предусмотрены более сложные методы, такие как Mittal и Paragios [17], которые основаны на «ядрах с переменной пропускной способностью». Хофманн предложил адаптивный сегментатор на основе пикселей (PBAS – PixelBased Adaptive Segmenter).

1.2 Обзор аналогичных решений для достижения заданной цели

Главным отличием представленных систем от привычных парковок торговых центров будет в том, что последние могут выполнять только функции высчитывания свободные места

1.2.1 Система интернета вещей на базе платформы «СТРИЖ»

Один из резидентов объединения космических технологий и телекоммуникаций Фонда «Сколково» – Компания «СТРИЖ Телематика», запустила беспроводную сеть для интернета вещей. Разработчики в «Сколково» будут не только параллельно внедрять IoT-сервисы, но и получают доступ к сети, что позволит создавать свои собственные «умные» устройства и приложения на базе отечественной технологии для интернета вещей. Им представится возможность тестировать и отлаживать проекты в режиме реального времени.

Компания занимается одной из пилотных разрабатываемых отечественной беспроводной энергоэффективной технологии, предназначенной для передачи данных в IoT-сетях и межмашинного «общения» (M2M). Продукт «СТРИЖ» представляет собой IoT-платформу для постройки «умных» LPWA-сетей и разработки компактных устройств, которые способны передавать сигнал на расстояние свыше 50 км и при этом автономно работать в течение 10 лет.

Архитектурная схема IoT-системы «СТРИЖ» представлена на рисунке 1.

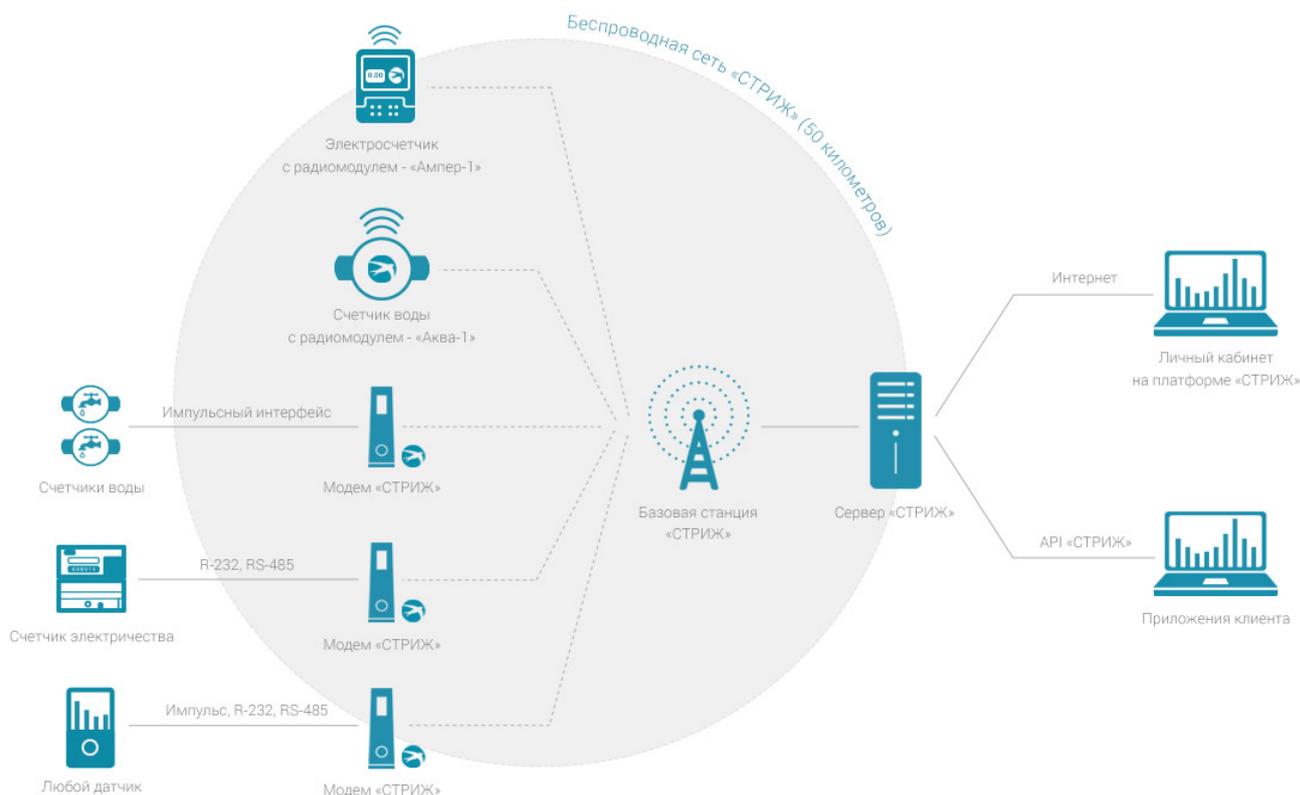


Рисунок 1 – Техническая схема системы интернета вещей «СТРИЖ» [18]

Представленный проект организации управления парковочным пространством по максимуму использует достижения инновационной LPWAN-технологии отечественного разработчика IoT-систем «СТРИЖ Телематика».

Технология LPWAN (Low-Power Wide-Area Network) – подход в радиосвязи, применяемый для различных устройств и крупных беспроводных сетей телеметрии. Особенность данной технологии заключается в том, что она отвечает критериям низкого энергопотребления (low-power) и широким территориальным охватом (wide-area).

Перечислим ключевые особенности технологии LPWAN:

- 1) большая дальность – позволит телеметрии сниматься с датчиков, распределенных на большом расстоянии, а именно более 10 км в городской черте;
- 2) энергоэффективность – сенсоры, вмонтированные в дорожное покрытие, не требуют замены батареи в течение 10 лет.

Любые существующие беспроводные технологии передачи данных обладают такими характеристиками как дальность, скорость и энергоэффективность. Причем одновременно можно соответствовать лишь 2-м

из 3-х (рисунок 2). Схематично представим график зависимости энергоэффективности от дальности передачи сигнала от точки до точки, где элементами множества на плоскости будут технологии беспроводной передачи данных (рисунок 3).

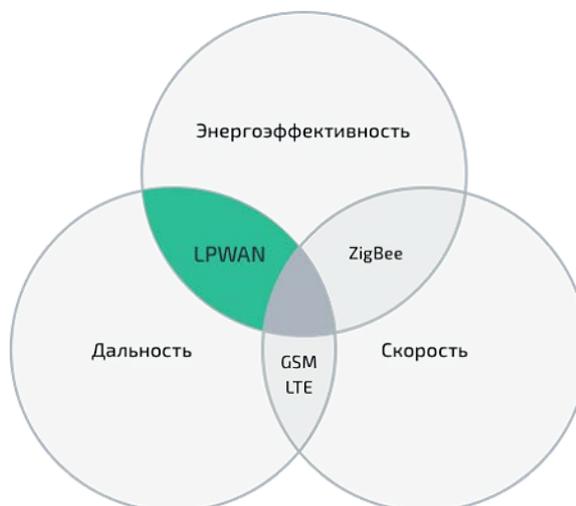


Рисунок 2 – Диаграмма совокупностей характеристик беспроводных технологий



Рисунок 3 – Диаграмма с ключевыми преимуществами LPWAN

Отметим еще одно преимущество технологии, а именно высокую проникающую способность сигнала: с уверенным приемом-передачей данных от датчиков в плотной застройке успешно справились всего 2 базовых станции.

Парковочный датчик присутствия автомобиля – беспроводной датчик определения занятости парковочного места. Устройство предназначено для фиксации изменений электромагнитного поля и передачи сигнала в систему управления парковочными местами.

Датчик присутствия автомобиля используется для организации интеллектуальной системы парковки. Устройство размещается в асфальте, вровень с поверхностью дорожного полотна (рисунок 4).



Рисунок 4 – Парковочный датчик присутствия автомобиля
(актуален на 2015 год)

Встроенный электромагнитный сенсор реагирует на изменения электромагнитного поля. Радиомодуль отправляет сигнал в систему управления и контроля занятости парковочными местами для дальнейшей обработки и отображения на парковочном дисплее количества свободных мест.

Специальные алгоритмы фильтрации шумов исключают реагирование датчика на посторонние электромагнитные колебания, вызываемые, например, поездами метрополитена.

Доступны две модели датчиков: на основе радиомодулей СТРИЖ РМ4 и LoRa LL-RLP-20/LL-RXR-27.

1.2.2 Коммерческий проект «SPOTParking»

Продукт является разработкой исследователя Томского политехнического университета. Данная система называется Smart Parking Occupation Tracking или сокращенно «SPOT». Аппаратная часть включает в себя видеокамеру и небольшой вычислительный узел, основанный на базе одноплатного компьютера. На тех парковках, которые оборудованы собственным видеонаблюдением, необходимость применения дополнительных камер отпадает. Программная часть использует алгоритмы на базе методов машинного обучения для анализа видеопотока на предмет числа занятых и свободных мест [20].

На апрель 2020 года устройство было готово и протестировано на примере парковки на ул. Карпова в Томске. Кроме того, демонстрировалась версия системы, интегрированная с картами 2GIS. За изменениями, происходящими на тестовой парковке, можно наблюдать на сайте [20]. Свободные места на стоянке отображаются зелеными маркерами, а занятые — красными.

В перспективе функциональность платформы может быть расширена: система, к примеру, сможет обнаруживать автомобили, которые припаркованы не по правилам, занимают сразу два парковочных места и пр [21]. Демонстрация работы системы SPOTParking для операционной системы под управлением iOS представлена на рисунке 5.

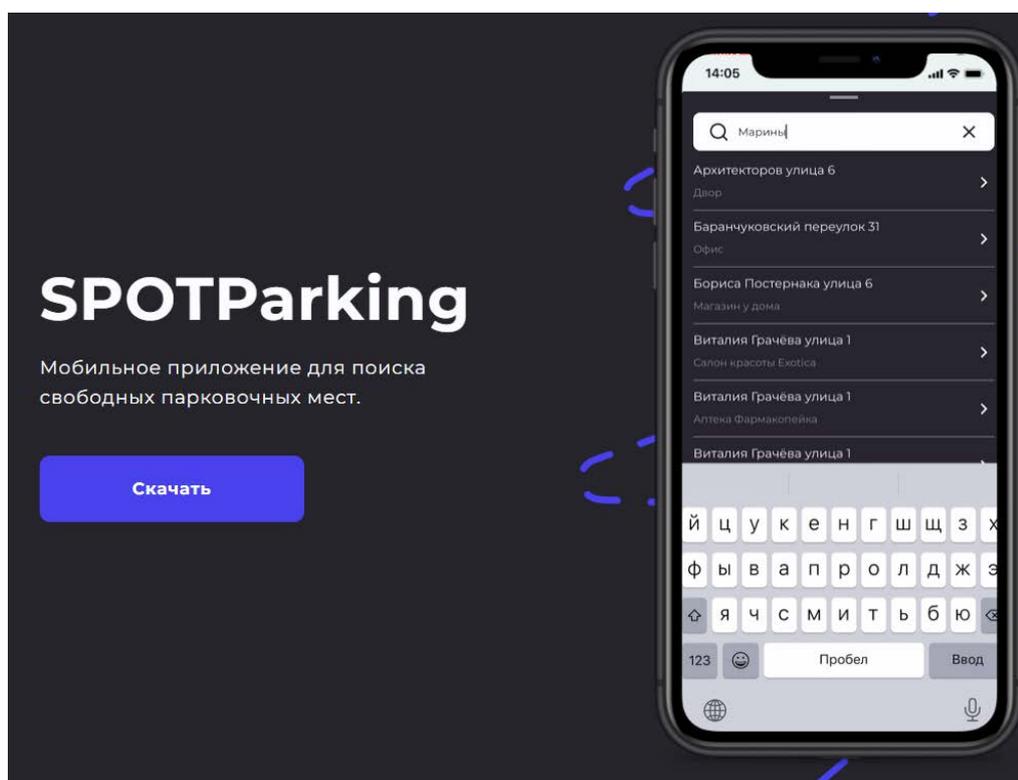


Рисунок 5 – Скриншот анимированной демонстрации рабочего продукта [21]

1.2.3 Умная парковка «ParkiT»

Команда старших инженеров-электриков и IT-специалистов Университета Райса США разработала проект под названием «ParkiT» [22]. Цель проекта – избавиться от излишних хлопот водителей и оператора парковки.

Разработчики описывают свою систему как более дешёвую, чем предыдущая внедрённая система умной парковки, основанная на использовании датчиков и технологий интернета вещей. Новая система предоставит управляющему и обслуживающему персоналу парковок в режиме реального времени собирать сведения о том, где именно находятся свободные парковочные места. Затем эта информация транслируется водителям с помощью электронных указателей или удобного для водителя приложения.

Разработчики отказались от датчиков в пользу камеры и технологий компьютерного зрения. Они используют камеры на больших парковках, рассчитанные на 50-100 и более мест.

Алгоритм работы технологии описывается следующим образом: камера делает снимок парковочной зоны каждую минуту, затем камера отправляет изображение на сервер, где оно обрабатывается с помощью разработанных

алгоритмов компьютерного зрения и машинного обучения. После получения результата обработки, он отображается в пользовательском интерфейсе оператора парковки. Как только оператор стоянки получает доступ к информации о состоянии парковочных мест, он передает ее водителям с помощью цифровых вывесок, указывающих на направления свободных парковочных мест, или через мобильное приложение.

Авторы описывают подход использования технологий интернета вещей на парковочных местах как дорогостоящий. Стоимость одного парковочного места составляет около 400-800 долларов, также данному подходу присуща проблема монтажа необходимых датчиков для каждого парковочного места. Использование компьютерного зрения и машинного обучения позволило снизить стоимость внедрения системы «ParkiT» до 50 долларов за парковочное место. Демонстрация работоспособного проекта представлена на рисунке 6.

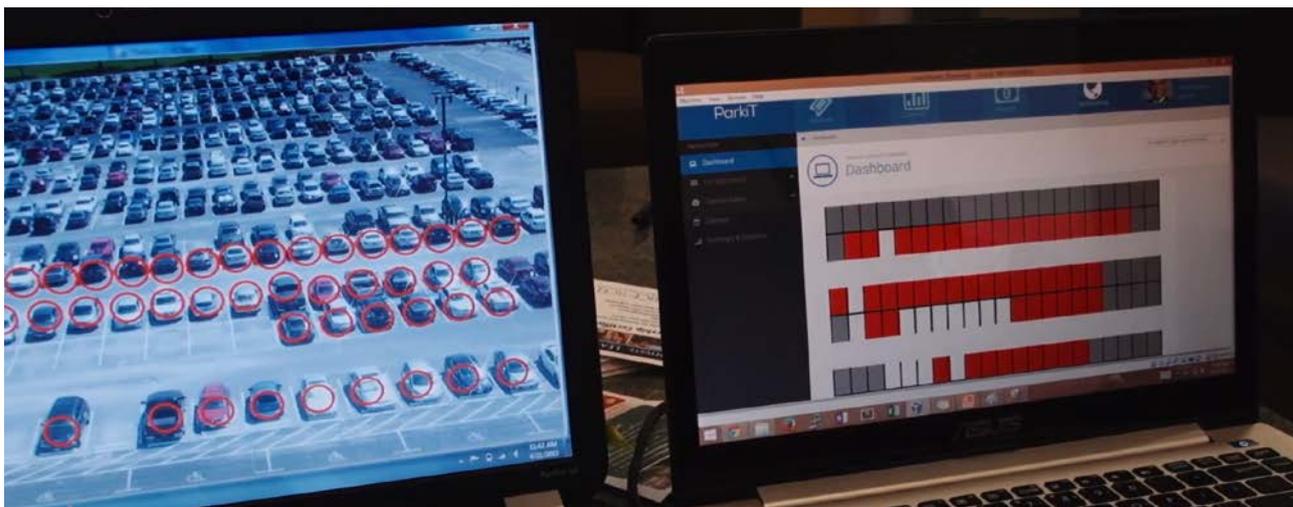


Рисунок 6 – Демонстрация проекта ParkiT [22]

1.2.4 Streetline

Технология Streetline [23], одноименной Американской компании, предоставляет мобильным приложениям, менеджерам парковок, операторам, продавцам и OEM-производителям информацию о наличии свободных парковочных мест и правилах парковки в режиме реального времени. Доступ к этой информации осуществляется двумя способами: API-сервисы и через мобильное приложение.

API - сервисы.

Наличие информации о наличии свободных парковочных мест и правилах парковки в режиме реального времени обеспечивает бесшовную интеграцию точных данных с реальностью. Доступная информация включает информацию о наличии парковочных мест в режиме реального времени по кварталам, а также тарифы на парковку в режиме реального времени, часы работы, правила и ограничения.

Приложение Parker™.

Parker – это бесплатное мобильное справочное приложение для потребителей, которое предоставляет доступ к точной информации в режиме реального времени о наличии свободных парковочных мест. Parker доступен на iOS и Android для обеспечения. Сервисы доступные в приложении:

Интерфейс мобильного приложения представлен на рисунке 7.

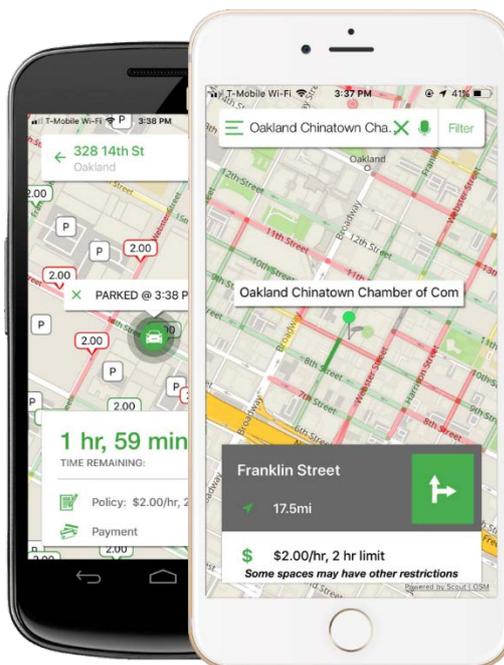


Рисунок 7 – Интерфейс мобильного приложения Parker [23]

1.2.5 Система компании ParkApp

Мобильное приложение резидента "Сколково", компании ParkApp, позволяет оплачивать банковскими картами московскую городскую парковку, а также парковки аэропорта Домодедово, ТЦ "Авиапарк" и ТЦ "Европейский".

Главная цель проекта – создание единого международного сервиса, который поможет свести к минимуму время и усилия на поиск и оплату парковки [24].

Сервисом пользуются более 700 тыс. человек. Помимо основного функционала, в приложении ParkApp имеется опция, которая прогнозирует появление свободных мест на паркинге.

Сегодня проект развивается в нескольких направлениях. Одно из них – Blockchain. Все данные о платежах и транзакциях предполагается фиксировать в децентрализованном Blockchain. Такая технология поможет избежать несправедливо выписанных штрафов и дальнейших разбирательств между водителем и муниципалитетом из-за человеческого фактора или системных сбоев.

Другое направление – совершенствование системы оплаты, продления и автоматического прекращения оплаты уличной парковки. Идет работа над системой автоматического въезда, выезда и оплаты коммерческих паркингов – без необходимости получения билетов или карт.

Еще одно важное направление – унификация пользовательского опыта и экспансия. Создание единого приложения и платформы для мирового парковочного пространства и всех платежей, которые может произвести водитель, находясь за рулем.

1.2.6 Умная парковка от Интерсвязь

«Умные парковки» от компании «Интерсвязь» – это сервис в бесплатном мобильном приложении, который позволяет любому пользователю узнать о загруженности конкретных паркингов в круглосуточном режиме. Сейчас доступно две функции [26].

Первая – узнать количество и расположение свободных мест, где отдельно обозначаются места для инвалидов. Вторая – это система безопасности «Мониторинг автомобиля», которая позволяет поставить свой автомобиль на круглосуточное наблюдение.

Если «Мониторинг автомобиля» видит, что автомобиль сдвинулся с места, то сервис сразу сообщит об этом автовладельцу. Также сейчас IT-специалисты компании «Интерсвязь» работают над внедрением новой функции «мониторинг активностей», которая будет сообщать о «необычных» движениях возле автомобиля. Например, система будет реагировать на подозрительные действия людей, находящихся поблизости. Это позволит не допустить неприятных ситуаций, связанных с авто, и отпугнет злоумышленников, так как те будут знать, что автомобиль охраняется искусственным интеллектом.

Сервис работает круглосуточно, при любых погодных условиях и освещении. Снег или дождь не оказывают особого влияния на работу «Умных парковок». Система «Умные парковки», работает благодаря сверхточным нейронным сетям. Кроме «Мониторинга» пользователям сервиса доступны и другие полезные функции. «Умные парковки» отображают на стоянках места для инвалидов, а также помогают найти свободную парковку благодаря голосовому помощнику «Алиса».

Функционал модуля ПАРКОВКА:

- подсчет и отображение свободных парковочных мест;
- разметка зон, нежелательных для парковки (газоны, тротуары, проезды), а также инвалидных мест;
- уведомление пользователей о появлении свободных мест, когда парковка заполнена.

Внешний вид модуля актуален на 22.02.23, представлен на рисунке 8.



Рисунок 8 – Внешний вид модуля ПАРКОВКА [26]

Функционал модуля КУПОЛ:

- уведомления о подозрительных действиях вблизи автомобиля;
- уведомления в случае, если автомобиль начнет движение.

Внешний вид модуля актуален на 22.02.23, представлен на рисунке 9.



Рисунок 9 – Внешний вид модуля КУПОЛ [26]

Технические требования. Камеры видеонаблюдения:

1) Full-HD IP-камеры, расположенные сверху парковки (например, на 9-этажном здании);

2) сервер на 150 камер:

– CPU: 40 логических ядер;

– GPU: 4 видеокарты, каждая по 8ГБ видеопамяти, уровня gtx1070 и выше;

– RAM: 128ГБ;

– Сеть: 1 гигабит.

Затраты на внедрение системы, полученные с официального сайта компании, представлены на рисунках 10 и 11 [26].

SaaS Облачное решение	1 500 ₽ за 1 камеру в месяц	3 000 ₽ за 1 камеру в месяц
Предоставление доступа к облачному сервису	✓	✓
API для интеграции со сторонними мобильными приложениями и веб-порталами	✓	✓
Модуль ПАРКОВКА	✓	✓
Модуль КУПОЛ	✗	✓
	Узнать подробнее	Узнать подробнее

Рисунок 10 – Затраты на внедрение на основе облачных технологий

Stand-Alone Автономное решение	24 000 ₽ единовременный платеж за 1 камеру	48 000 ₽ единовременный платеж за 1 камеру
Развертывание системы на серверах заказчика в автономном контуре	✓	✓
Техническая поддержка и обновления системы с дообучением нейросетевых моделей	✓	✓
Модуль ПАРКОВКА	✓	✓
Модуль КУПОЛ	✗	✓
	Узнать подробнее	Узнать подробнее

Рисунок 11 – Затраты на внедрение на основе частного размещения

1.2.7 Проект «Как я нахожу парковочное место за 5 секунд»

Автор спроектировал и реализовал достаточно простое решение обозначенной проблемы [27].

Автором были обозначены некоторые возможные источники видеотрансляции:

- получить видеопоток от управляющей компании дома, однако данный способ не всегда возможен, и имеет ограничение – отсутствие камер в домах старой застройки;

- разместить на крыше дома или в квартире, из которой есть полный обзор на парковку.

Для подбора подходящего оборудования автор рекомендует придерживаться следующих технических ограничений:

- чем больше разрешение, тем лучше будет работать детекция объектов, поэтому предлагается использовать камеры от 2Мп;

- должен быть большой угол обзора – не менее 90 градусов – это нужно для того, чтобы решить проблему меньшим количеством камер;

- камера должна быть с фокусным расстоянием 2,5 – 2,8 мм;

– наличие режима "день/ночь" позволит проводить анализ в темное время суток.

Проблему передачи видеопотока от камеры на web-сервер автор предлагает решать несколькими вариантами:

– разместить сервер у себя дома (в одной сети с камерой). Возможно, как временное решение;

– через ПО, которое будет находится в той же сети и передавать видео на сервер. Минус этого в том, что это ПО должно инсталлировано и запущено на каком-то сервере, т.е. для него нужна постоянно работающая машина;

– настройка DDNS на камере, то есть прокси-сервер, через доступные для камеры сервисы;

– покупка статического IP адреса у провайдера интернета. У этого решения есть как плюсы, так и минусы. В качестве плюсов у крупных провайдеров, например, достаточно сделать два клика в личном кабинете и настроить порты прослушивания;

– настроить VPN на роутере до сервера;

– через интеграцию с облачным сервисом камеры.

В качестве архитектурного решения автор использует клиент-серверную парадигму. В качестве сервера будет использоваться сервис по анализу состояния занятости парковочных мест на основе Mask R-CNN. А также автор предлагает использовать в качестве альтернативной основы работы сервиса проприетарный сервис компании Macroscop. Демонстрация работы представлена на рисунке 12.

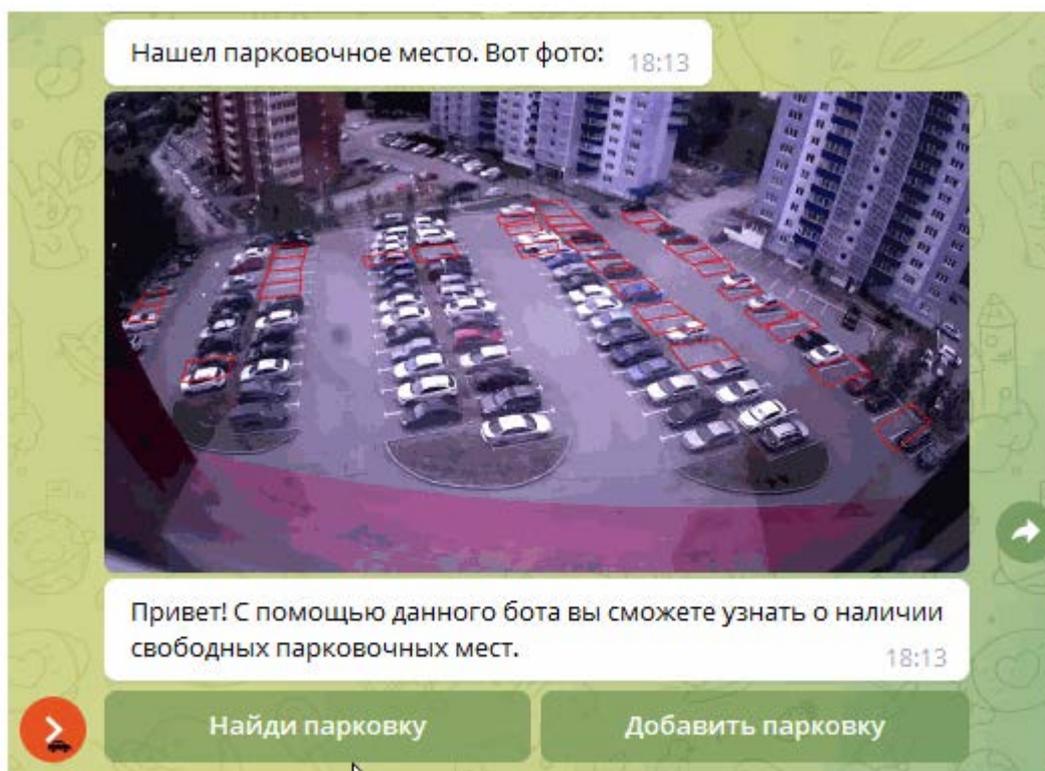


Рисунок 12 – Демонстрация работы системы [27]

В итоге автор получил работоспособный ПАК для анализа состояния занятости парковочных мест.

1.2.8 Проект «Трекинг парковочных мест на базе Fast RCNN»

Автор задаётся вопросом вариативности использования модели. Первым представленным вариантом является YOLOv7 с работой на процессоре, при этом она выдавала довольно хорошую производительность. Вторым вариантом был рассмотрен фреймворк OpenVINO, в котором есть предобученные модели от самого Intel, однако модели от Intel были не готовы распознавать машины при виде сверху (рисунок 13) [28].

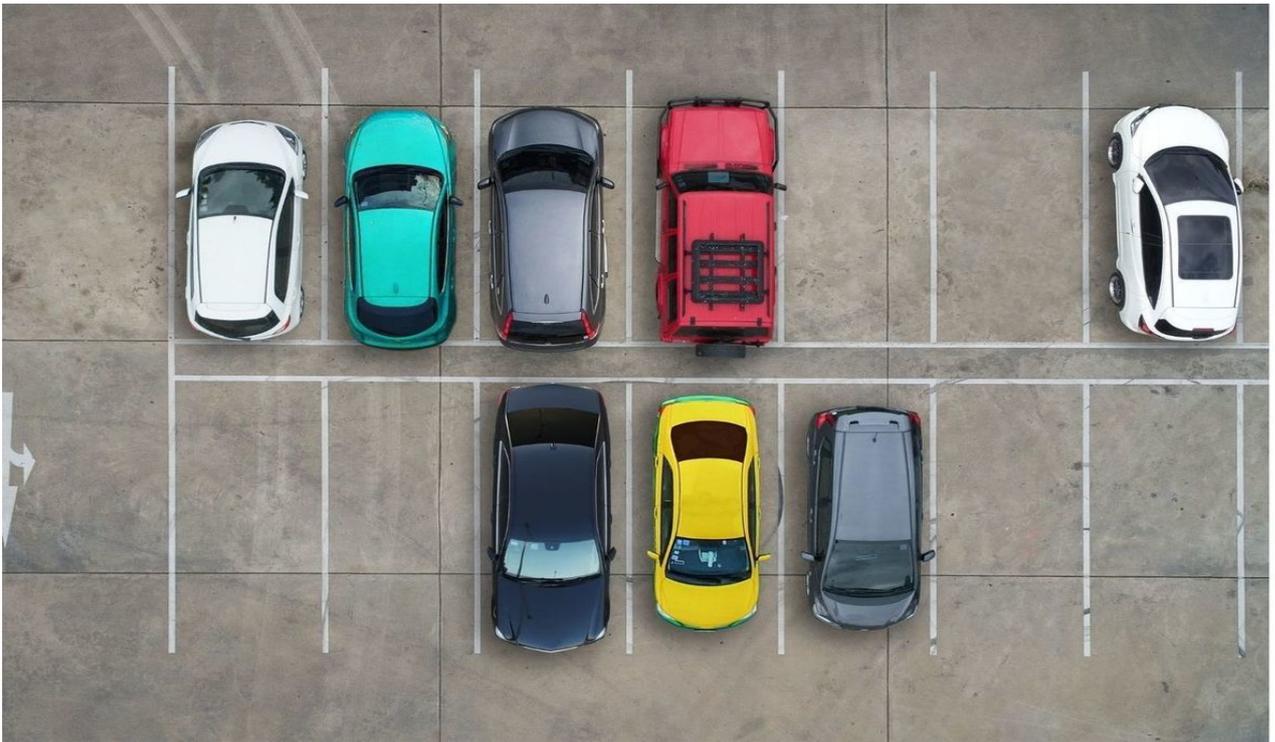


Рисунок 13 – Результат работы нейронной сети от Intel

В качестве альтернативного автор выбрал решение найти простое и удобное в использовании. В процессе анализа инструментов была выбрана библиотека torchvision. В качестве исходных плюсов в ней есть предобученные веса. В итоге за основу я взял модель Fast RCNN с явно выраженным минусом в области скорости обработки. Результат работы FRCNN изображён на рисунке 14.



Рисунок 14 – Результат работы FRCNN

Парковочные места. Автор разметил парковочные места на модели. Очевидно, что у этого способа много недостатков в виде сдвига камеры, трудозатрат и всякого подобного. Но обучить модель определять парковочные места мне кажется будет довольно сложно, так как места либо никак не размечены на парковке,

В итоге автор получил ПО, которое считает свободные парковочные места, причем очень точно если камера находится ровно перпендикулярно над парковкой. Однако эта реализация и имеет ряд недостатков, что говорит о проблеме ведения продукта в эксплуатацию.

1.2.9 Проект «Трекинг свободных парковочных мест на базе Mask R-CNN»

Автор разработал рабочий прототип с глубоким обучением [29]. Для простоты разработки автор предлагает спроектировать систему на основе следующих этапов алгоритма (рисунок 15).



Рисунок 15 – Схематическое изображение алгоритма работы системы [29]

В первой фазе алгоритма видеопоток поступает с веб-камеры, направленной в окно. Скриншот получаемого видеопотока представлен на рисунке 16.



Рисунок 16 – Скриншот видеопотока [29]

Алгоритм работы системы предусматривает передачу каждого кадра видео на каждый этап, по одному за раз.

Первым шагом автор предлагает начать распознавание всех возможных парковочных мест на кадре. Затем на каждом последующем кадре видеопотока производится поиск всех машин, что позволит отслеживать движение каждой машины от кадра к кадру. Третьим шагом происходит определение: какие места заняты машинами, а какие – нет. Для этого необходимо совместить результаты первых двух шагов. В конечном шаге алгоритма, программа должна прислать оповещение, когда освободится парковочное место. Это будет определяться за счёт изменений в расположении машин между кадрами видео.

В связи с ограничением ресурсов, в видеопотоке обрабатывается лишь несколько кадров в секунду.

В системе распознавания используется «новый» подход с глубоким обучением Mask R-CNN, который совмещает в себе точность CNN и набор технических хитростей, сильно повышающих скорость распознавания.

Архитектура Mask R-CNN разработана таким образом, что она распознаёт объекты на всём изображении, эффективно тратя ресурсы, и при этом не

использует подход скользящего окна. Mask R-CNN отдаёт на выходе много информации о каждом распознанном объекте: местоположение каждого объекта и его контур (маску) (рисунок 17). Для разработки своего проекта автор решил использовать open-source модель от Matterport для ускорения прототипирования.



Рисунок 17 – Пример работы open-source модели от Matterport

Модель распознает не только машины, но и такие объекты, как светофоры и люди. Для каждого распознанного объекта модель Mask R-CNN возвращает 4 параметра:

- тип обнаруженного объекта (целое число). Предобученная модель COCO умеет распознавать 80 разных часто встречающихся объектов вроде машин и грузовиков;
- степень уверенности в результатах распознавания. Чем выше число, тем сильнее модель уверена в правильности распознавания объекта;
- ограничивающая рамка для объекта в форме XY-координат пикселей на изображении;
- «Маска», которая показывает, какие пиксели внутри ограничивающей рамки являются частью объекта. С помощью данных маски можно найти контур объекта.

Далее было принято решение уйти от полного списка параметров для каждого объекта (рисунок 18).

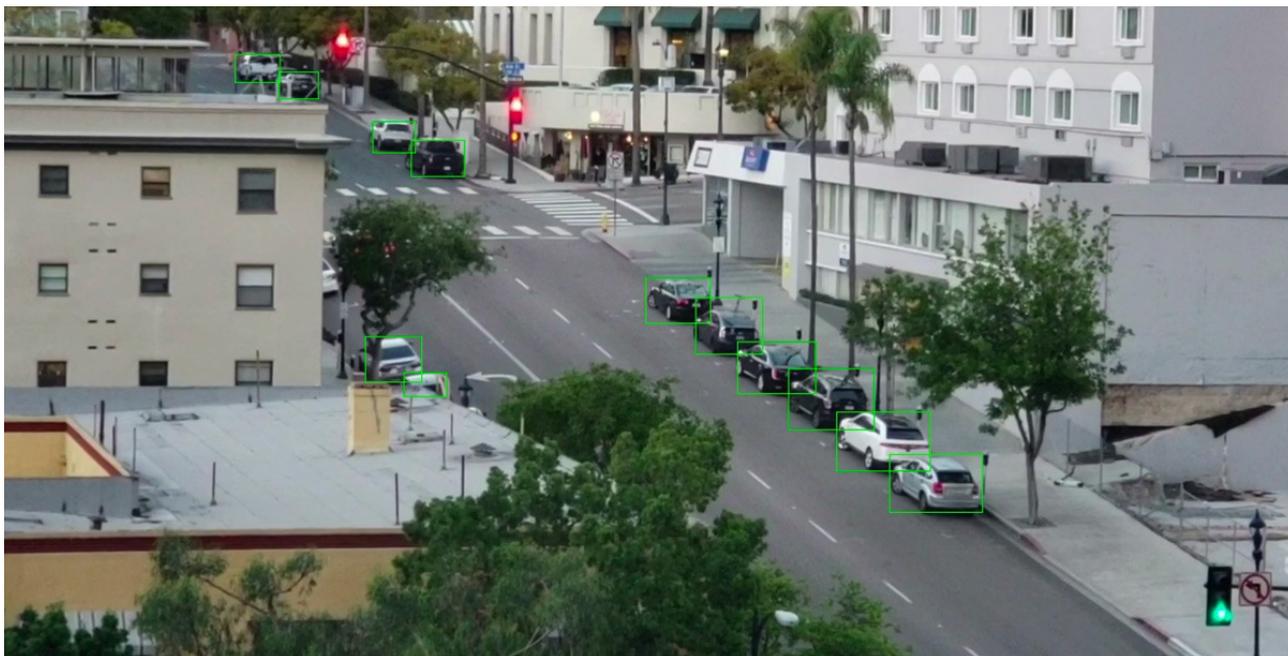


Рисунок 18 – Выделение рамок на автомобилях

Для распознавания пустых парковочных мест используются пиксельные координаты каждой машины. На основе нескольких последовательных кадров определяется, какие из машин двигались.

В случае, если степень пересечения двух объектов (рамок) велика, то можно предполагать, что место занято.

Чтобы искать только «наиболее пустые» рамки автор предлагает воспользоваться методом под названием Intersection Over Union (отношение площади пересечения к сумме площадей) или IoU. IoU можно найти, посчитав количество пикселей, где пересекаются два объекта, и разделить на количество пикселей, занимаемых этими объектами (рисунок 19).

$$\text{IoU} = \frac{\text{Площадь пересечения}}{\text{Сумма площадей}}$$

Рисунок 19 – Формула метода Intersection Over Union

Данный метод позволяет понять, как сильно ограничивающая рамка машины пересекается с рамкой парковочного места. Если значение IoU низкое, например 0.15, значит, машина занимает малую часть парковочного места. А если оно высокое, вроде 0.6, то это значит, что машина занимает большую часть места и там нельзя припарковаться.

Отслеживание пересечения рамок предлагается производить с помощью двумерного массива. В этом двумерном массиве каждая строка отражает одну рамку парковочного места. А каждый столбец говорит о том, насколько сильно каждое из мест пересекается с одной из обнаруженных машин.

Для отправки сообщений автор использует Twilio. Twilio — это популярный API, который позволяет отправлять SMS из практически любого языка программирования с помощью всего нескольких строк кода.

1.3 Вывод по разделу 1

Произведем подбор критериев особенностей реализации технологий «Умных парковок», по которым будем сравнивать группы решений, представленных ранее, составим для этого таблицу 1. В перечень параметров для включения: «Требования к оборудованию», «Требования к дорожному покрытию», «Стоимость внедрения», «Скорость обработки данных», «Скорость внедрения».

Произведем подбор критериев, по которым будем сравнивать несколько аналогичных решений, представленных ранее, по отношению к проектируемой системе EasyParking, составим для этого таблицу 2. В перечень параметров для включения: «Стоимость», «Распознавание подозрительных действий», «Уведомление о появлении свободных мест», «Интерфейс взаимодействия», «Доступ к исходному коду», «Работа системы в Челябинске».

На основании выводов полученных при сравнительном анализе таблиц 1 и 2 можно заключить, что наиболее целесообразным будет реализация проекта на основе обработки изображения с видеокамер и алгоритмов машинного зрения.

Таблица 1 – Особенности реализации технологий умных парковок

	Требования к оборудованию	Требования к дорожному покрытию	Стоимость внедрения	Скорость обработки данных	Скорость внедрения
Мониторинг парковочных мест с использованием видеокамеры и компьютерного зрения	Установка камеры и установка сервера для сбора и обработки видео	Нет никакой привязки к дорожному полотну	Средняя стоимость установки варьироваться от 20000 до 50000 рублей за каждое парковочное место в зависимости от сложности и масштабы проекта	обработка изображения и определение наличия автомобилей на парковке с задержкой, что может снизить скорость реакции на заполнение парковки	установка видеоборудования с покрытием всей парковки, настройка системы, обучение персонала ~ до 4х недель
Мониторинг парковочных мест с использованием интернета вещей	Датчики необходимо интегрировать в элементы инфраструктуры, требуется установить сервер для сбора и обработки данных с датчиков	Датчики, смонтированные на покрытие, могут препятствовать чистке снега в зимний период уборочными машинами	Средняя стоимость установки варьироваться от 50000 до 150000 рублей за каждое парковочное место в зависимости от сложности и масштабы проекта	данные от датчиков передаются в режиме реального времени, что позволяет оперативно реагировать на изменения парковочной ситуации	установка датчиков для каждого парковочного места отдельно, настройка системы, обучение персонала ~ от 4х недель

Таблица 2 – Сравнительная таблица некоторых представленных ранее аналогов

Наименование системы	Стоимость		Распознавание подозрительных действий	Навигация до парковочного места	Уведомление о появлении свободных мест	Интерфейс взаимодействия	Доступ к исходному коду	Работа системы в Челябинске
	Использование пользователем	Внедрение на парковку						
Сервис от Интерсвязь "Умные парковки"	Бесплатно	за одну камеру от 1,5 до 3 тыс. рублей / мес или единовременно от 24 до 48 тыс. рублей	Да	Нет	Да	web-сервисы (сайт, приложение для iOS и Android)	Нет	Да
Система SPOTParking	беспл. для 1 парковки, 149р/мес ко всем открытым парковкам	есть форма для заявки	Да	Да	Да	приложение для iOS и Android	Нет	Нет
Зарубежная система ParkiT	Бесплатно	-	Нет	-	-	информационный стенд	Нет	Нет
Система Streetline	Бесплатно	-	Нет	Да	Да	web-сайт, приложение для iOS и Android	Нет	Нет
Система мониторинга EasyParking	Бесплатно	за одну камеру от 10 тыс. рублей единовременно	Нет	Нет	Да	чат-бот	Да	Да

2 ОПРЕДЕЛЕНИЕ ТРЕБОВАНИЙ

Исходные данные к работе: имеющиеся технические ресурсы и требования к разрабатываемому научно-техническому продукту.

Для реализации разрабатываемого проекта необходим следующий набор подсистем:

1) модуль сбора и обработки внешних данных, который отвечает за прием видеопотока с IP-камеры, и его обработку с помощью компьютерного зрения. В свою очередь, в качестве выходных данных модуль определяет состояние занятости парковочных мест, общее число парковочных мест, карту парковочных мест;

2) web-сервер, который будет отвечать за бизнес-логику управления и сохранения требуемых данных. Сервер выступит промежуточным звеном в цепи между программой обработки данных и клиентом;

3) база данных для хранения информации, требуемой для администрирования, а также для потенциального включения в базу данных сведений о пользователях системы и сведений о состоянии каждого парковочного места. База данных может содержать информацию о номере места, его состоянии (свободно или занято), времени занятия и освобождения места, а также другую необходимую информацию;

4) клиентский интерфейс предназначен для удобного взаимодействия пользователя с системой умной парковки. В первичном цикле разработки системы в качестве клиента требуется разработать чат-бот для мессенджера Telegram, который позволяет получить доступ к требуемой информации по сети.

2.1 Функциональные требования

В данном разделе опишем функциональные требования, предъявляемые к разрабатываемой системе. Для четкого разграничения предъявляемых

требований от разных участников взаимодействия к системе обозначим основные группы такие как пользователи и администраторы.

Для роли взаимодействия типа Пользователь определим следующие функциональные требования:

1) пользователь должен иметь возможность просматривать на экране информационного табло количество свободных парковочных мест и общее количество парковочных мест;

2) пользователь должен иметь возможность просматривать на экране информационного табло направление движения по территории парковки в сторону ближайшего свободного парковочного места;

3) пользователь должен иметь возможность запустить диалог с чат-ботом, входящем в совокупность модулей системы, в Телеграмм;

4) пользователь должен иметь возможность запросить состояние парковочных мест у телеграмм-бота, входящего в совокупность модулей системы;

5) пользователь должен иметь возможность запросить направление движения в сторону ближайшего свободного парковочного места у телеграмм-бота, входящего в совокупность модулей системы;

6) пользователь должен иметь возможность получить ответ, содержащий актуальное состояние парковочных мест от телеграмм-бота, входящего в совокупность модулей системы;

7) пользователь должен иметь возможность получить ответ, содержащий направление движения в сторону ближайшего свободного парковочного места, от телеграмм-бота, входящего в совокупность модулей системы.

Для краткой иллюстрации функциональных требований к системе со стороны пользователя представим диаграмму взаимодействия в нотации UML на рисунке 20.

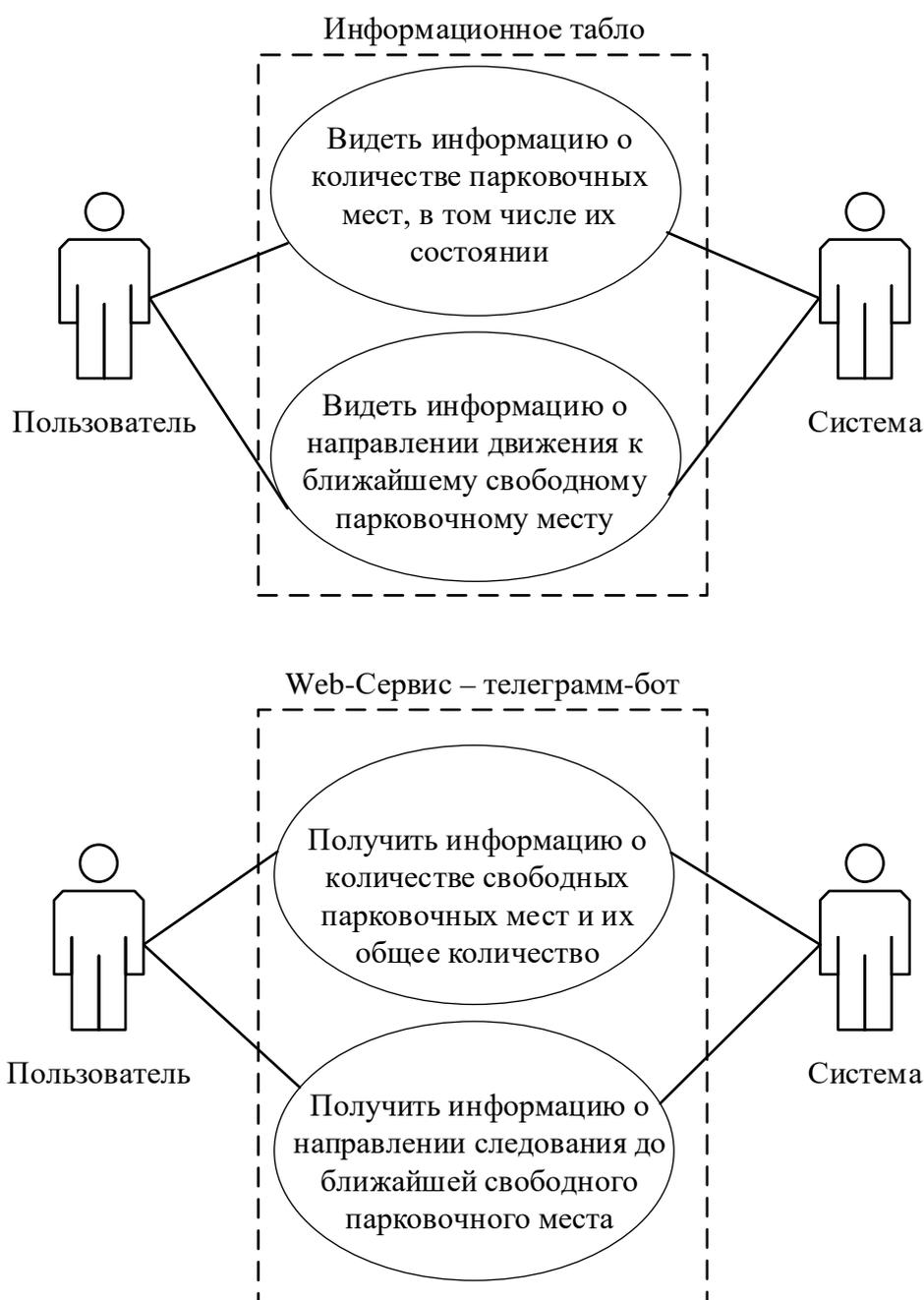


Рисунок 20 – Диаграмма взаимодействия пользователя в рамках системы

Для роли взаимодействия типа Администратор определим следующие функциональные требования:

- 1) администратор должен иметь возможность зарегистрировать учётную запись в системе;
- 2) администратор должен иметь возможность войти в систему по заранее созданным учётным данным;
- 3) администратор должен иметь возможность загружать в рабочий интерфейс сетку областей обработки изображения, получаемых с видеопотока;

4) администратор должен иметь возможность настраивать сетку областей обработки изображения;

5) администратор должен иметь возможность сохранять настроенную сетку областей обработки изображения;

6) администратор должен иметь возможность включать и отключать систему мониторинга парковочных мест;

7) администратор должен иметь возможность конфигурировать и переконфигурировать параметры IP-камеры;

8) администратор должен иметь возможность редактировать чат меню чат-бота;

9) администратор должен иметь возможность настраивать источник трансляции видеопотока системы.

Для краткой иллюстрации функциональных требований к системе со стороны администратора представим диаграмму использования в нотации UML на рисунке 21.

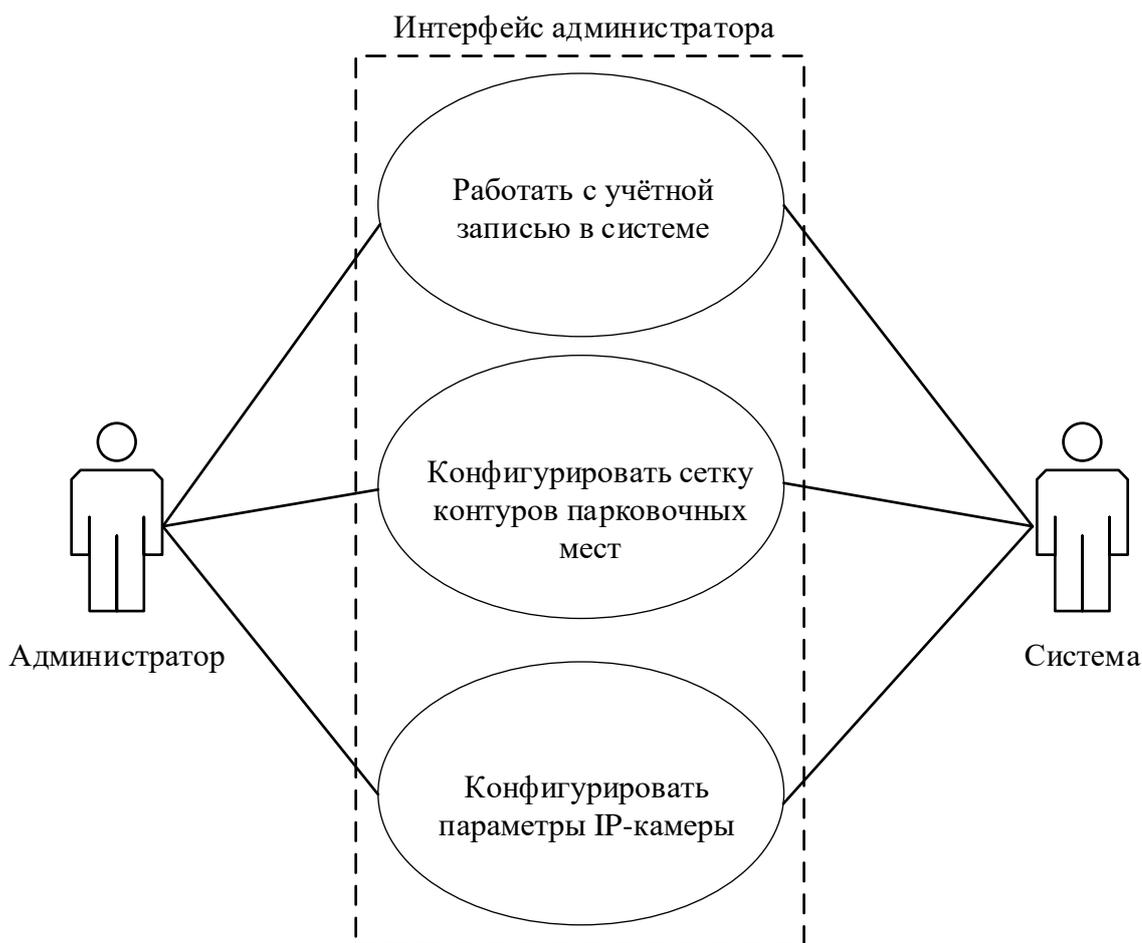


Рисунок 21 – Диаграмма взаимодействия администратора в рамках системы

2.2 Нефункциональные требования

Разрабатываемая система должна соответствовать следующим нефункциональным требованиям:

- 1) передача прямой трансляции потокового видео с устройства видеofиксации должна вестись по протоколу RTSP;
- 2) время, затрачиваемое на анализ изображения и регистрацию в системе после въезда автомобиля в парковочную область, не должно превышать 1 с.;
- 3) время, затрачиваемое на анализ изображения и регистрацию в системе после выезда автомобиля из парковочной области, не должно превышать 1 с.;
- 4) база данных реализуется на основе СУБД SQLite, версии не старше 3.0.0;
- 5) время ответа на запрос пользователя в чат-бот не более 5 с.;
- 6) питание IP-камеры должно осуществляться по технологии PoE для минимизации прокладываемых кабельных систем.

2.2.1 Вывод по разделу 2

В этом разделе были определен состав модулей в система визуализации парковочных мест. Также были заданы две группы требований, такие как функциональные и нефункциональные, что позволит произвести дальнейшую работу над системой, то есть разработать архитектуру системы.

3 ПРОЕКТИРОВАНИЕ

В этой главе опишем алгоритмы решения поставленных задач, спроектируем техническую и программную архитектуру системы.

3.1 Алгоритмы работы системы

Для описания пошаговой работы системы составим соответствующие алгоритмы. Общий алгоритм работы системы представлен на рисунке 22.



Рисунок 22 – Алгоритм запуска и работы системы

Алгоритмы работы системы распознавания занятости парковочных мест приведём на рисунках 23, 24 и 25.

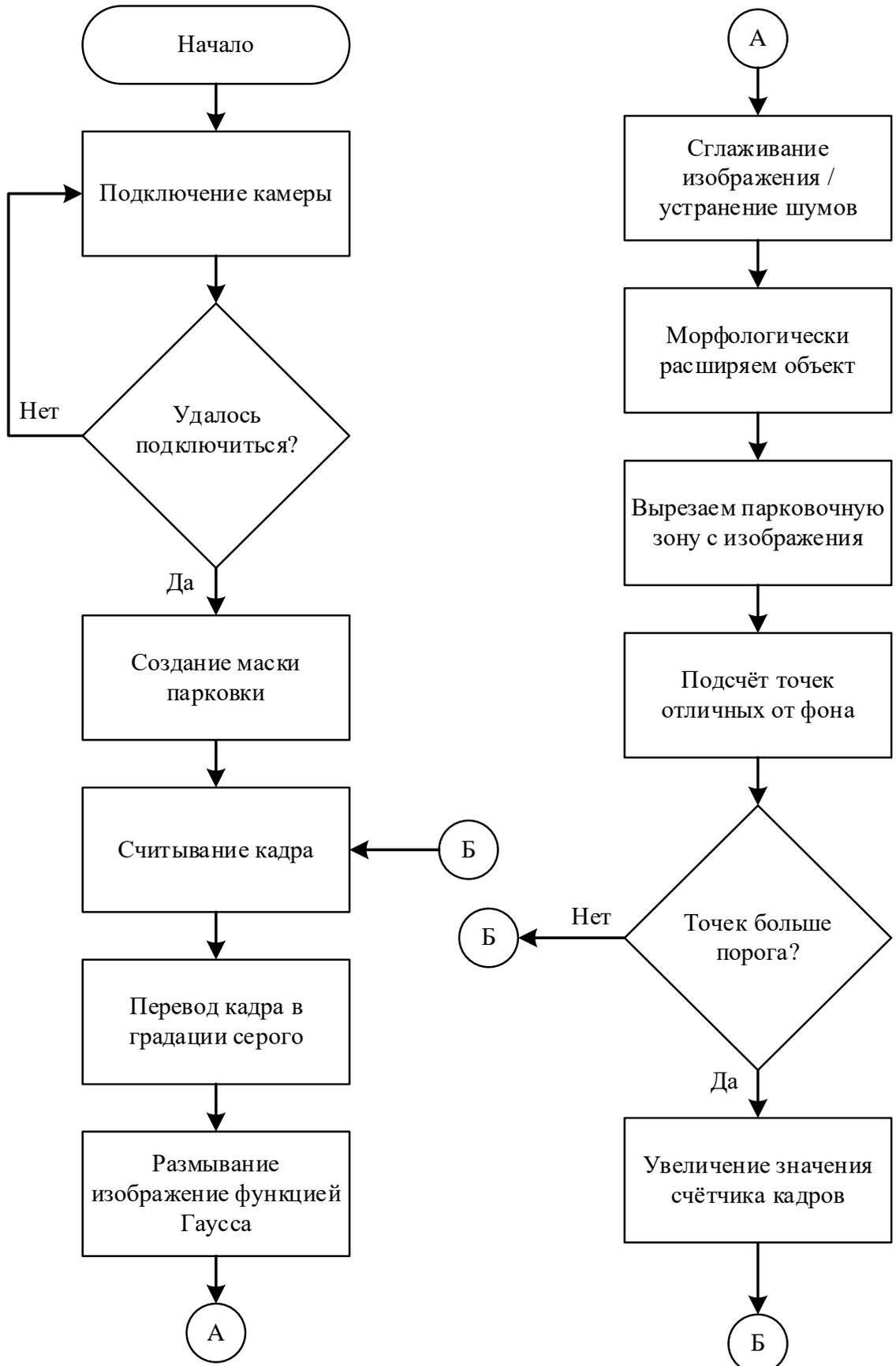


Рисунок 23 – Алгоритм работы основного цикла обработки изображения

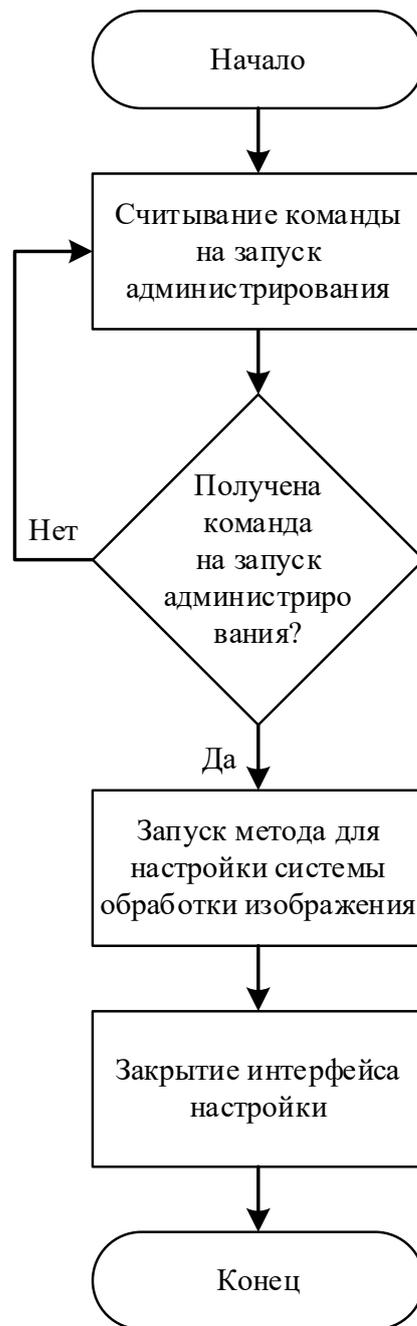


Рисунок 24 – Алгоритм работы подмодуля настройки контуров парковочной зоны

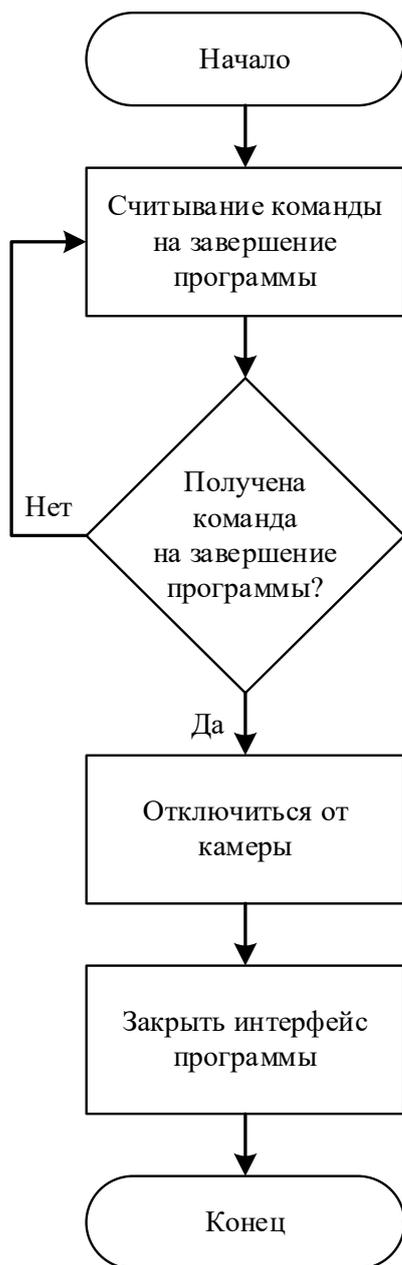


Рисунок 25 – Алгоритм завершения работы программы

Составим диаграмму последовательности для каждого из пользователей системы. На рисунке 26 отобразим диаграмму последовательности для Пользователя.

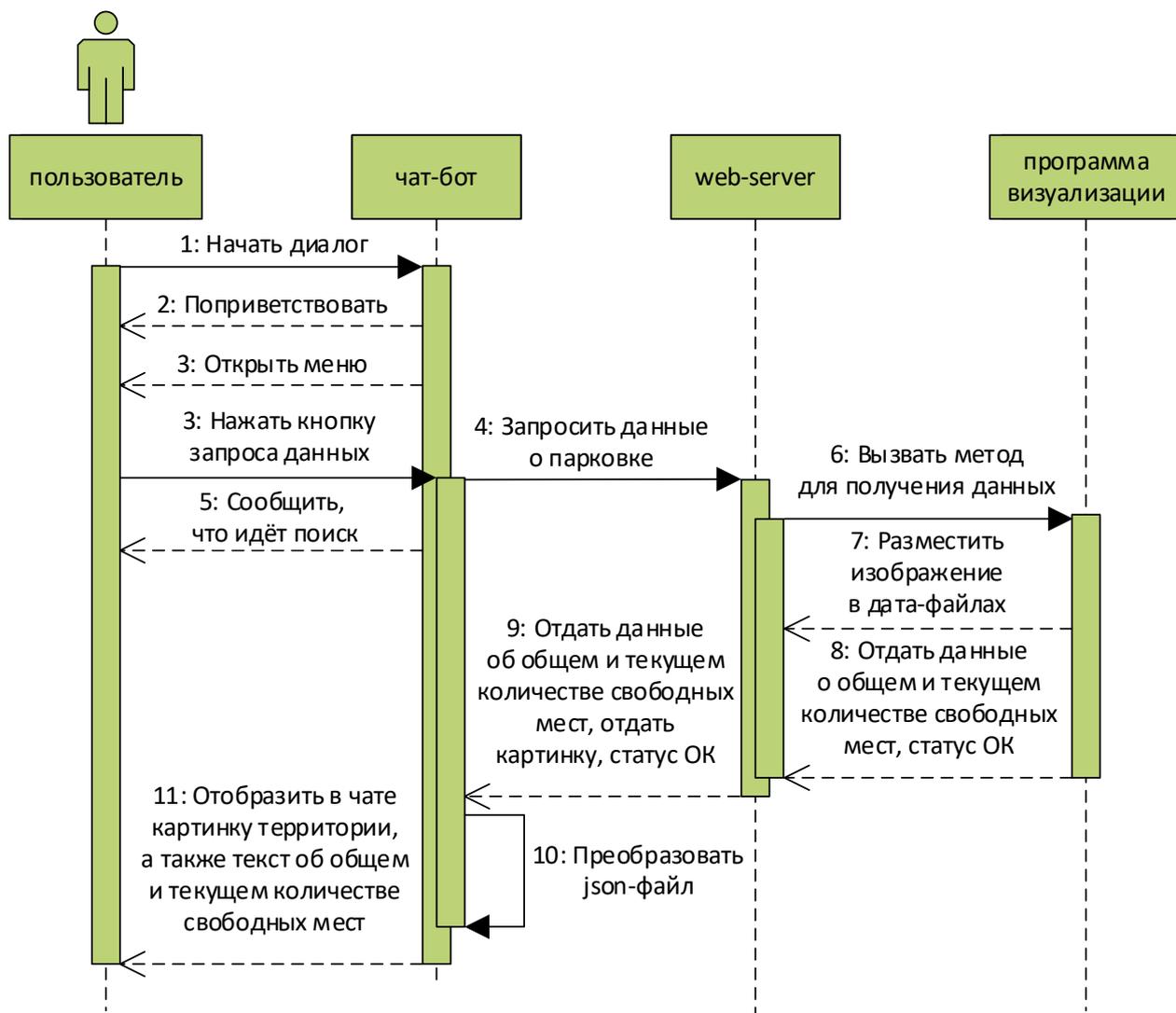


Рисунок 26 – Диаграмма последовательности сценария Пользователя

3.2 Архитектура разрабатываемого решения

Рассмотрим техническую архитектуру разрабатываемого ПАК. Проектируемая система состоит из 3х компонентов (рисунок 27):

- периферийное устройство – Ip-камера;
- физический сервер с запущенными приложениями, который отвечает за считывание видеопотока с Ip-камеры, за обработку видеопотока для получения количества свободных мест, за передачу изображения через интерфейсы сетевого взаимодействия;

- редактор матрицы контуров парковочных мест;
- дисплей для отображения информации на парковке и телеграмм-бот, отвечающий за сетевое взаимодействие Пользователя с программой.

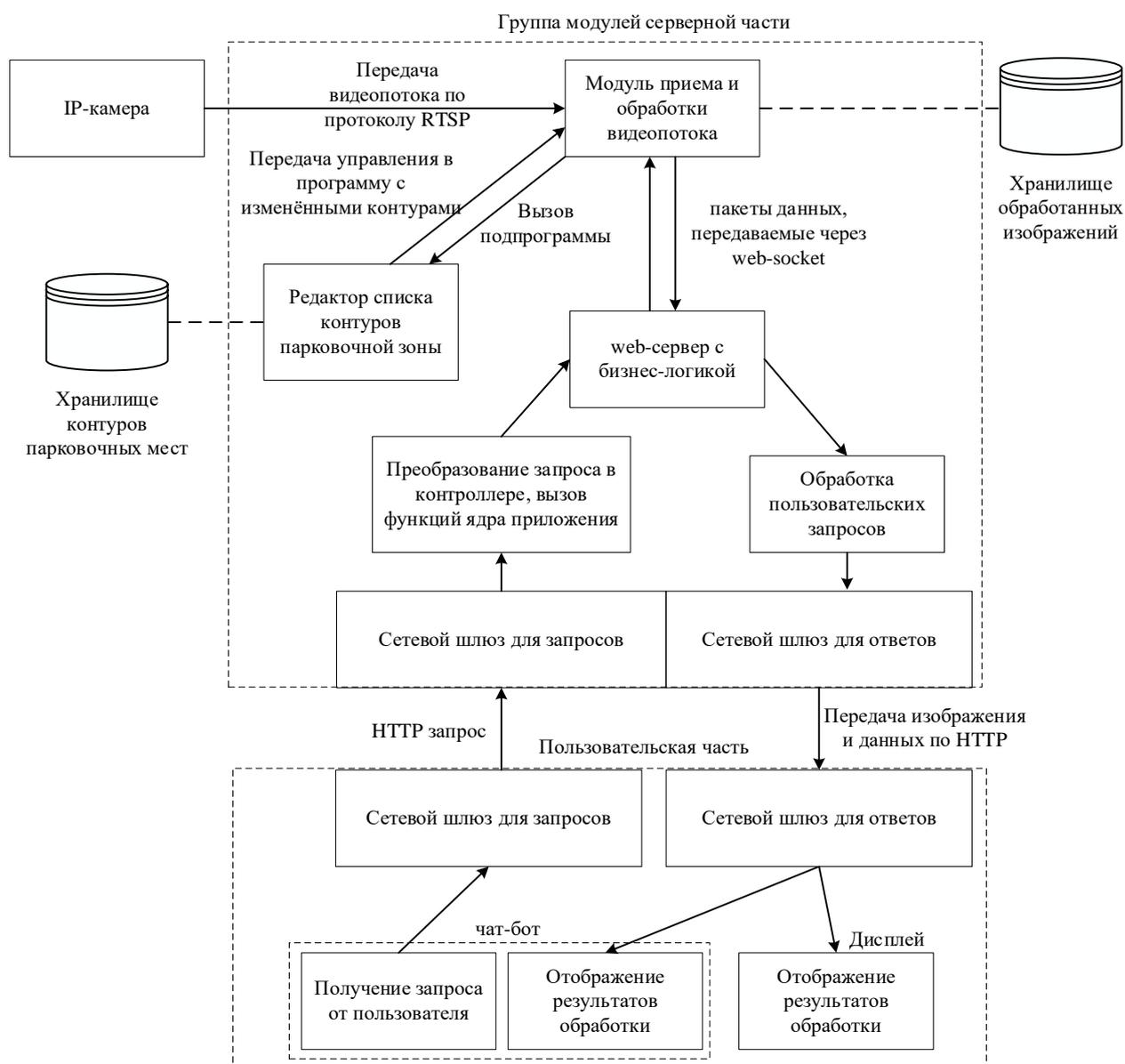


Рисунок 27 – Техническая архитектура разрабатываемого ПАК

3.2.1 Архитектура редактора матрицы контуров парковочных мест

Редактор сетки контуров парковочных мест – подпрограмма, предназначенная для создания и изменения сетки контуров парковочных мест на изображении с камеры видеонаблюдения парковки. Редактор используется при построении сетки контуров парковочных мест для идентификации отдельных парковочных мест. Редактор контуров парковочных мест предназначен для

выделения контуров парковочных мест, которые в дальнейшем анализируются приложением для обработки состояния парковочных мест.

Редактор контуров осуществляет взаимодействие с хранилищем парковочных мест. На рисунке 28 представлена архитектура редактора.

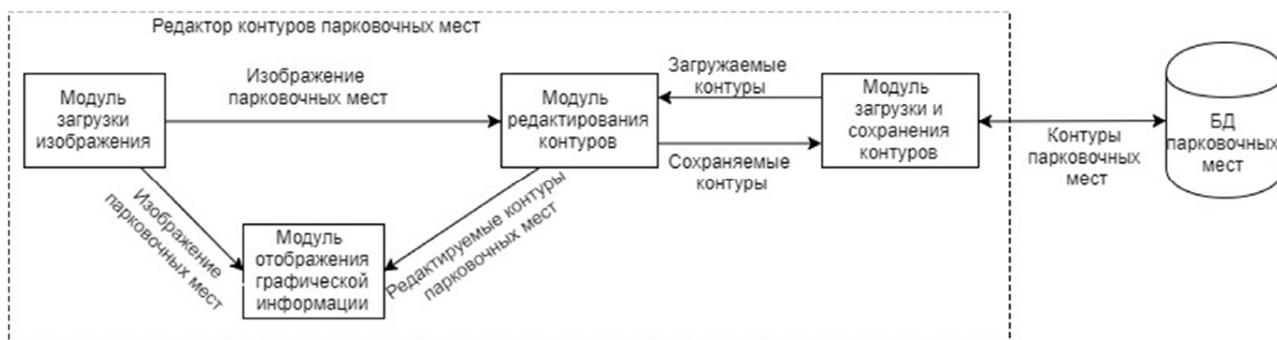


Рисунок 28 – Архитектура редактора контуров парковочных мест

На вход редактора подается последний кадр, полученный с камеры видеонаблюдения парковки. Администратор может изменять размер изображения для более удобной работы с ним.

После начальной подготовки необходимо задать контуры, обозначающие места, на которых в дальнейшем будут припаркованы машины. Минимальный контур может состоять из трёх точек, в то время как максимального ограничения нет. Существует 3 опции при работе с контурами парковок:

1) нанесение контуров, для этого необходимо навести курсор на необходимую точку изображения, и нажать левую кнопку мыши. Выбрав достаточное количество точек необходимо нажать на клавишу «пробел», чтобы редактор добавил контур в список и сохранил его;

2) удаление контуров, для это необходимо поместить курсор вовнутрь граней требуемого контура и произвести удаление правой кнопкой мыши. После нанести точки заново, см. п. 1;

3) удаление выбранных точек, для этого необходимо нажать на клавишу “с” («си»), после администратор получит уведомление, что все не сохранённые точки были удалены.

Отображаемые контуры извлекаются из модуля редактирования контуров.

4 РЕАЛИЗАЦИЯ

Для осуществления реализации системы потребуется выполнить ряд задач:

- 1) подобрать комплектующие под заданные ограничения системы;
- 2) выбрать стек программного обеспечения;
- 3) произвести разработку программного кода в соответствии с современными стандартами;
- 4) произвести сборку и пилотный запуск системы.

4.1 Подбор комплектующих

4.1.1 Сетевая камера с фиксированным фокусным расстоянием DH-IPC-HFW2431S

Устройство представляет собой сетевую камеру с фиксированным фокусным расстоянием из младшей серии сетевых камер Dahua Lite с модельным номером DH-IPC-HFW2431S (рисунок 29). Далее для простоты обращения с названием сетевой камеры Dahua DH-IPC-HFW2431S определим название как «Dahua».



Рисунок 29 – Изображение сетевой камеры «Dahua»

Благодаря усовершенствованной технологии кодирования H.265 сетевая камера Dahua обладает эффективной способностью кодирования видео, что экономит полосу пропускания и место для хранения. Эта камера использует новейшую технологию starlight и отображает более качественное цветное изображение в условиях низкой освещенности. Камера поддерживает хранение записываемых материалов на SD-картах, пылезащитную функцию,

водонепроницаемую функцию и антивандальную функцию, соответствующую стандартам IP67.

Приведём часть технических характеристик камеры «Dahua» в таблице 3.

Таблица 3 – Технические характеристики камеры

Техническая характеристика	Заявленное производителем значение
Макс. Разрешение	2688 (H) × 1520 (V)
Кодеки сжатия видео	H.265; H.264; H.264B; MJPEG (поддерживается только подпоток)
Частота кадров видео	Основной поток: 2688 × 1520 (1 кадр в секунду-20 кадров в секунду) 2560 × 1440 (1 кадр в секунду-25/30 кадров в секунду) Вспомогательный поток: 704 × 576 (1 кадр в секунду-25 кадров в секунду) 704 × 480 (1 кадр в секунду-30 кадров в секунду)
Поддерживаемые Ethernet протоколы	IPv4; IPv6; HTTP; HTTPS; TCP; UDP; ARP; RTP ; RTSP; RTCP; RTMP; SMTP; FTP; SFTP; DHCP; DNS; DDNS; QoS; UPnP; NTP; Multicast; ICMP; IGMP; NFS; PPPoE; 802.1x; Bonjour
Напряжение питания	12V DC/PoE (802.3af)
Условия эксплуатации	-40°C to +60°C / относительная влажность ≤ 95%
Стандарт пыле-/влагзащиты	IP67

Исходя из условий эксплуатации (на улице в средней полосе в России при любых погодных условиях), из требований к кодированию (сжатию) изображения и поддерживаемым протоколам: камера полностью удовлетворяет потребностям и ограничениям разрабатываемой системы.

4.1.2 Инжектор Ubiquiti Networks POE-48-24W-G

PoE-инжектор – это устройство, которое используется для передачи питания по сетевому кабелю Ethernet (PoE) к устройствам, которые не имеют собственного источника питания. Таким образом PoE-инжектор добавляет питание к Ethernet-кабелю, чтобы устройства могли получать питание и данные от одного кабеля.

Ubiquiti Networks POE-48-24W-G обеспечивает передачу питания до 48 В по Ethernet-кабелю и имеет мощность 24 Вт. Он также имеет защиту от перегрузки, короткого замыкания и перенапряжения, что делает его безопасным для использования в различных условиях. Инжектор изображен на рисунке 30.



Рисунок 30 – Внешний вид устройства Ubiquiti Networks POE-48-24W-G

4.1.3 Патч-корд

Патч-корды используются при построении ЛВС (локальных вычислительных систем) и СКС (структурированной кабельной системы). Применяются для подключения коммутаторов, рабочих станций и другого сетевого оборудования.

Характеристики патч-корда [32]:

- тип кабеля – UTP;
- категория (TIA/EIA) - 5e;
- разъемы - RJ-45 - RJ-45;
- экранирование – нет;
- количество пар – 4;
- проводник – медь;
- материал оболочки - ПВХ (PVC) LSZH (самозатухающий пластик);
- длина - 2 м.

4.1.4 Серверное оборудование

Для тестовых запусков системы не потребуется больших вычислительных мощностей, так как будет использоваться лишь одна видеочасть, а запросы к серверу будут приходить лишь от одного пользователя, таким образом не возникает потребности аренды высокопроизводительного вычислительного оборудования (частного виртуального сервера).

Работу всего программного обеспечения может удовлетворить низкопроизводительное оборудование, которое может быть представлено как ПК, размещенный в проветриваемом помещении.

Таким образом для работы системы потребуется следующий перечень аппаратных устройств, полностью удовлетворяющих запросу в аппаратном обеспечении на данном этапе развития проекта:

- 1) центральный процессор – Intel® Core™ i5-8265U с 4-мя физическими ядрами;
- 2) видео ускоритель NVIDIA® GeForce® MX150;
- 3) 8 ГБ ОЗУ с тактовой частотой 2333 МГц;
- 4) постоянный накопитель на 256 ГБ.

4.2 Выбор стека программного обеспечения

4.2.1 Среда разработки

PyCharm – это интегрированная среда разработки (IDE) для языка программирования Python, которая была разработана компанией JetBrains. Она предоставляет разработчикам множество функций и возможностей, которые помогают ускорить процесс разработки и повысить эффективность.

Основные функции PyCharm:

- редактор кода: PyCharm предоставляет мощный редактор кода, который поддерживает автодополнение, отображение ошибок и подсветку синтаксиса;
- отладчик: PyCharm имеет встроенный отладчик, который помогает разработчикам находить и исправлять ошибки в коде;

- интеграция с Git: PyCharm интегрируется с системой контроля версий Git, что позволяет разработчикам управлять своим кодом и коммитами;
- виртуальные окружения: PyCharm позволяет создавать виртуальные окружения для проектов, что помогает изолировать зависимости и упрощает управление проектами;
- рефакторинг: PyCharm предоставляет множество функций рефакторинга, которые помогают улучшить качество кода и упростить его структуру;
- тестирование: PyCharm поддерживает тестирование кода, что позволяет разработчикам быстро проверять работу своих приложений;
- поддержка различных фреймворков: PyCharm поддерживает множество фреймворков для Python, включая Django, Flask, Pyramid и многие другие;
- интеграция с другими инструментами: PyCharm интегрируется с другими инструментами разработки, такими как Docker, Vagrant, AWS и многие другие.

PyCharm доступен в двух версиях: Community и Professional. Community – это бесплатная версия, которая предоставляет базовые функции разработки. Professional – это платная версия, которая предоставляет более продвинутые функции и возможности. Версия среды разработки Professional используется в крупных компаниях для системной коммерческой разработки, поэтому целесообразно использовать бесплатную Community версию.

4.2.2 Язык программирования

Python – это высокоуровневый язык программирования, который был создан в конце 1980-х годов Гвидо ван Россумом. Он был разработан с учетом простоты и читаемости кода, что делает его одним из наиболее популярных языков программирования в мире.

Python имеет множество преимуществ, включая:

- простота использования: Python имеет простой и понятный синтаксис, который делает его легким для изучения и использования.

– мультипарадигмальность: Python поддерживает несколько парадигм программирования, включая объектно-ориентированное, функциональное и процедурное программирование.

– мощные библиотеки: Python имеет огромную библиотеку, которая позволяет разработчикам быстро и легко создавать различные приложения.

– кроссплатформенность: Python может работать на разных операционных системах, включая Windows, Linux и Mac OS.

– большое сообщество: Python имеет большое сообщество разработчиков, которые создают новые библиотеки и модули, а также помогают другим разработчикам.

4.2.3 Выбор библиотек

Библиотека в программировании – это набор готовых функций, процедур и классов, созданных для решения определенных задач. Она представляет собой часть программного обеспечения и позволяет разработчикам использовать готовые решения для решения различных задач, таких как работа с файлами, работа с сетью, математические операции и т.д.

Использование библиотек позволяет ускорить процесс разработки, так как не нужно писать код с нуля, что позволяет использовать готовые решения.

Для нормализации изображения и выделения контуров, а также доступа к видеопотоку камеры используется библиотека OpenCV.

OpenCV (англ. Open Source Computer Vision Library, библиотека компьютерного зрения с открытым исходным кодом) – библиотека алгоритмов компьютерного зрения, обработки изображений и численных алгоритмов общего назначения с открытым кодом. Реализована на C/C++, также разработана для Python, Java, Ruby, Matlab, Lua и других языков. Может свободно использоваться в академических и коммерческих целях – распространяется в условиях лицензии BSD.

Для сериализации и десериализации объектов Python была выбрана библиотека pickle.

Сериализация – это процесс преобразования объекта Python в поток байтов, который можно сохранить на диск или передать по сети. Десериализация – это обратный процесс, при котором поток байтов преобразуется обратно в объект Python.

Библиотека `pickle` поддерживает сериализацию и десериализацию практически всех типов объектов Python, включая встроенные типы данных, списки, кортежи, словари, пользовательские классы и т.д. Она также поддерживает сериализацию объектов, содержащих ссылки на другие объекты, что позволяет сохранять и восстанавливать целые графы объектов.

`pickle` имеет два основных метода: `dump()` и `load()`. Метод `dump()` используется для сериализации объекта Python в поток байтов, который можно сохранить на диск или передать по сети. Метод `load()` используется для десериализации потока байтов и восстановления объекта Python.

`pickle` также поддерживает компрессию данных, что позволяет уменьшить размер сериализованных объектов и сократить время передачи по сети. Однако, использование `pickle` не рекомендуется для передачи данных между различными версиями Python или между различными платформами, так как формат сериализованных данных может отличаться.

4.2.4 Система контроля версий

Git – это система контроля версий, которая позволяет управлять изменениями в коде и сотрудничать над проектами с другими разработчиками. Система контроля версий обязательна в современной разработке, приведем 3 преимущества и 3 недостатка по отношению к Git.

Преимущества Git:

- Git позволяет легко отслеживать изменения в коде и возвращаться к предыдущим версиям, если что-то пошло не так;
- Git позволяет работать над проектом в команде, делиться изменениями и объединять их;
- Git обеспечивает безопасность данных и защиту от потери информации.

Недостатки Git:

- Git может быть сложен в использовании для новичков, которые не знакомы с основными концепциями систем контроля версий;
- Git может занимать много места на диске, особенно если проект содержит большое количество файлов или история изменений длительная;
- Git может быть медленным при работе с большими проектами.

4.3 Результаты разработки программного кода

Каталоги программы отслеживания состояния парковочных мест, web-сервера и программы чат-бота располагаются в одной директории (рисунок 31).

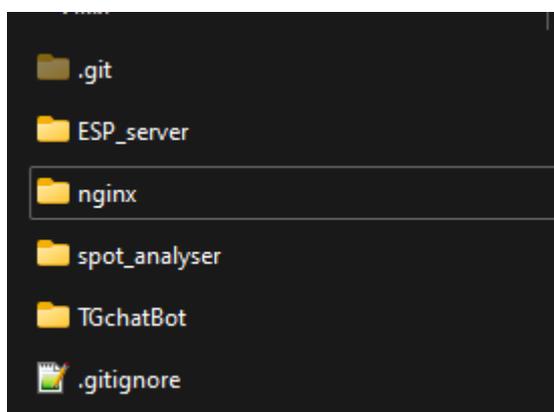


Рисунок 31 – Состав файлов и каталогов внутри корневой директории

Более детальное рассмотрение структуры каталогов и подробное описание работы проекта можно изучить на удаленном репозитории проекта в веб-инструменте жизненного цикла DevOps с открытым исходным кодом GitLab [31].

4.4 Вывод по разделу 4

Для сборки системы требуется два патч-корда, предназначенных для соединения устройств в локальную вычислительную сеть. Затем макет размещается в таком месте, которое позволяет произвести тестовые испытания.

5 СОСТАВЛЕНИЕ ПРОГРАММЫ ТЕСТОВЫХ ИСПЫТАНИЙ

Программа тестирования информационного взаимодействия с системой «EasyParking».

Настоящая программа тестовых испытаний разработана в целях определения порядка и условий проведения тестирования информационного взаимодействия с системой «EasyParking».

Объектом испытаний является информационная система визуализации парковочных мест.

Целью проведения испытаний является проверка соответствия информационного взаимодействия системы визуализации парковочных мест требованиям установленным в разделе 3 настоящей ПЗ.

Критерий успешности: при выполнении задания полученные результаты совпадают с ожидаемыми.

Правила работы с подпрограммой настройки контуров находятся в 3.2.1 настоящей ПЗ.

5.1 Контрольное задание 1

При отсутствии соединения модуля обработки изображения с средством считывания внешней информации – IP-камерой, ожидается получение уведомления о невозможности соединиться с заданным устройством и прекращении работы программы. Результат КЗ представлен на рисунке 32.

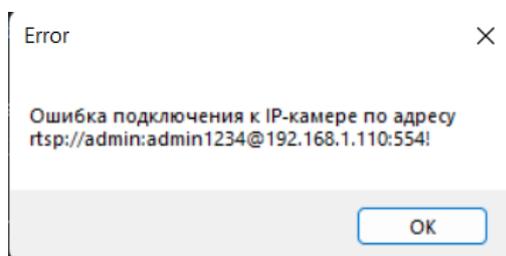


Рисунок 32 – Результат обработки ошибки подключения к камере
Результат проведения КЗ – успешно.

5.2 Контрольное задание 2

При первом запуске программы модуль считывания файла контуров не сможет найти искомый файл с сериализованным списком контуров, поэтому Администратор увидит предупреждение (рисунок 33).

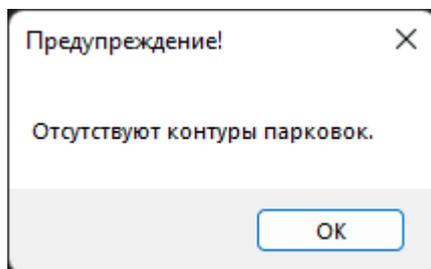


Рисунок 33 – Предупреждение об отсутствии контуров парковок

После этого последует ожидание команды о настройке контуров (рисунок 34).

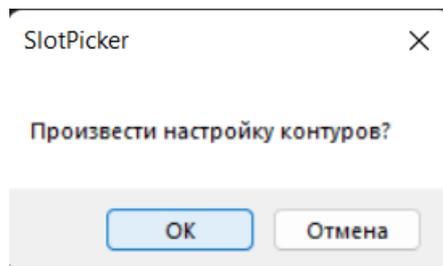


Рисунок 34 – Ожидание выбора команды для возможной настройки контуров

Если нажать на кнопку «да», то откроется подпрограмма настройки контуров парковочных мест (рисунок 35).

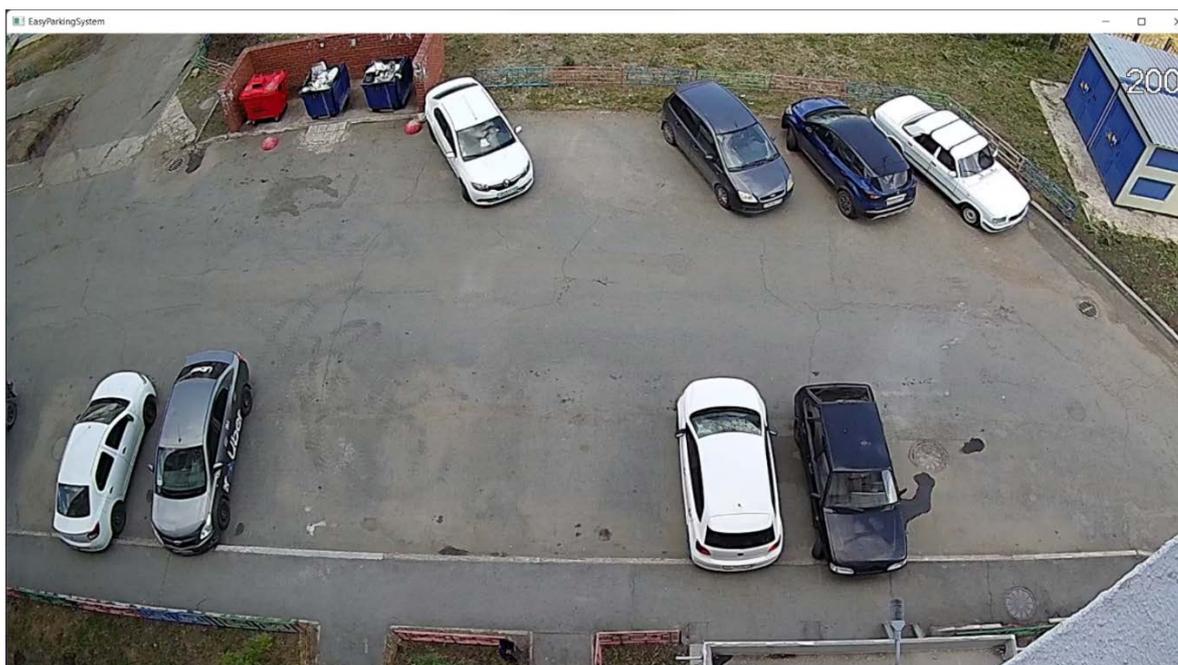


Рисунок 35 – Конфигурация контуров парковочных мест

Если нажать кнопку «нет», то запустится основная программа модуля обработки изображения без заданных контуров (рисунок 36).

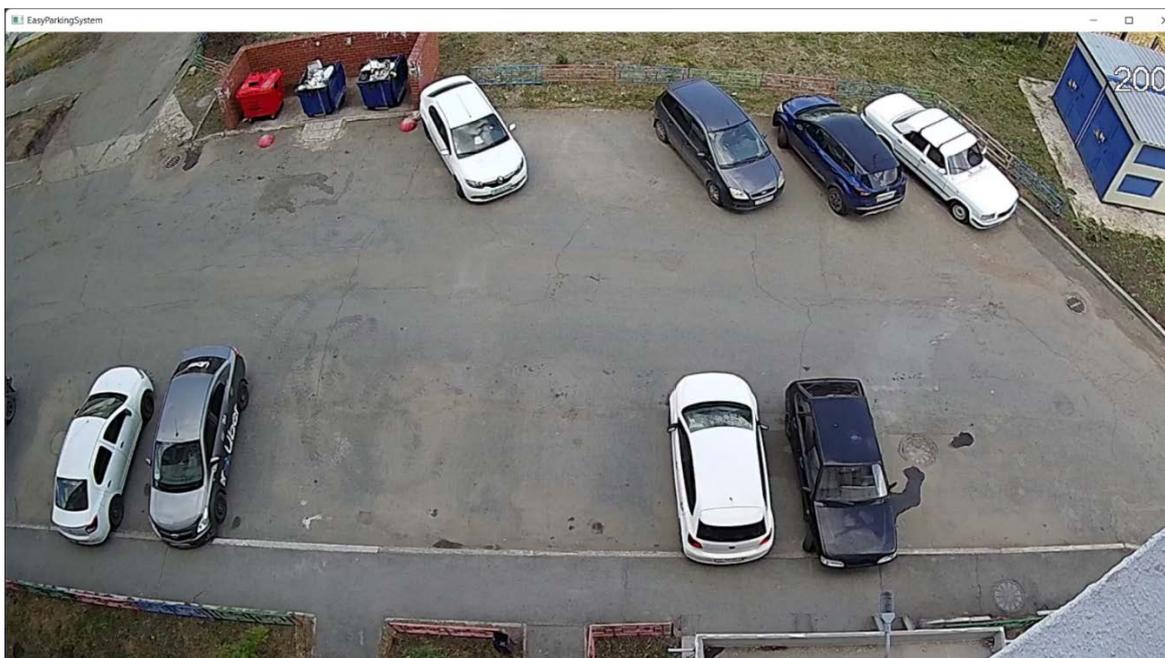


Рисунок 36 – Работа программы без контуров
Результат проведения КЗ – успешно.

5.3 Контрольное задание 3

После загрузки модуля конфигурации сетки контуров парковочных мест, и выполнении операций по настройке контуров, будет достигнута подобная цель (рисунок 37).

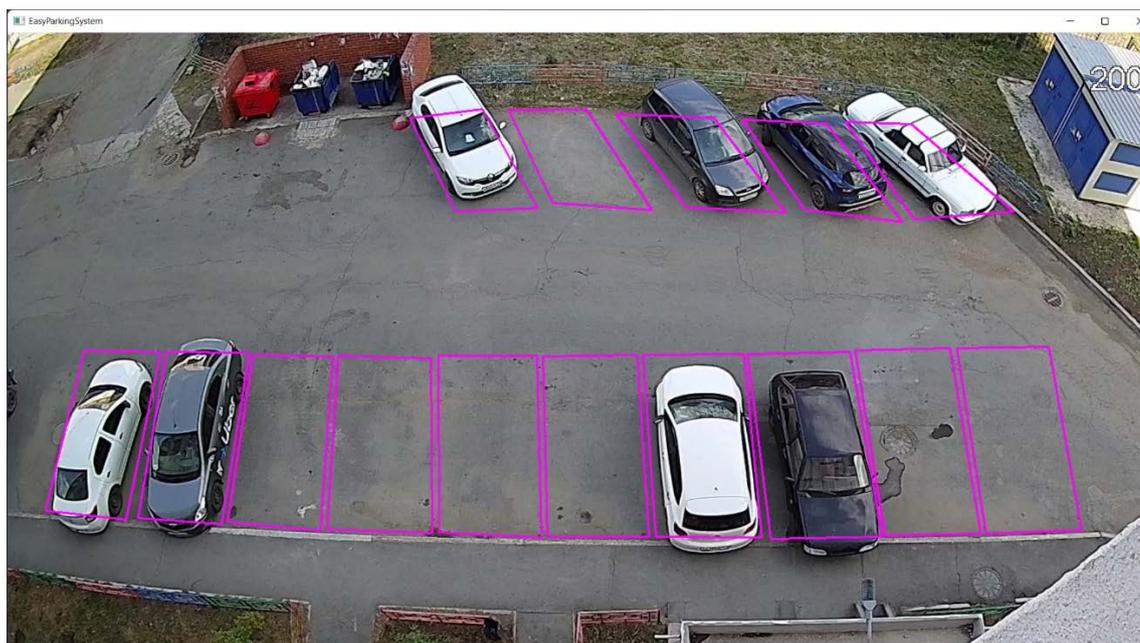


Рисунок 37 – Настройка сетки границ парковочных мест

При попытке задать фигуру контура на основе двух точек Администратор получит предупреждение, т.к. минимальной возможной фигурой является полигон – треугольник (рисунок 38).

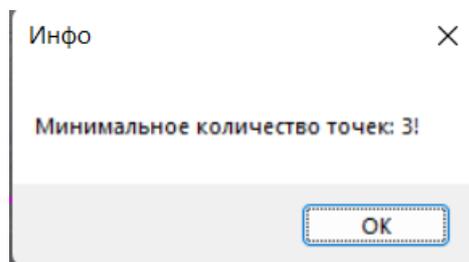


Рисунок 38 – Информация об отказе добавить фигуру из двух точек

Ожидается, что при ошибке проектирования контуров, например, как представлено на рисунке 39, Администратор удалит не нужный контур (рисунок 40).

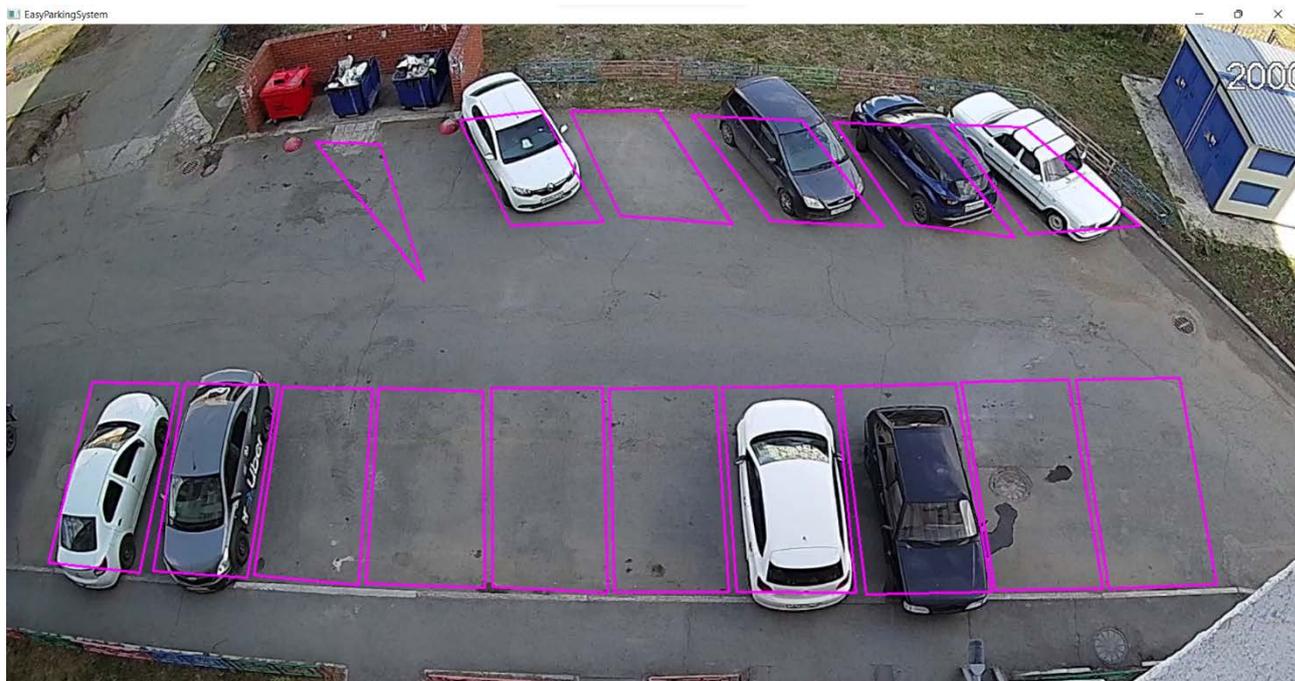


Рисунок 39 – Ошибка проектирования контуров

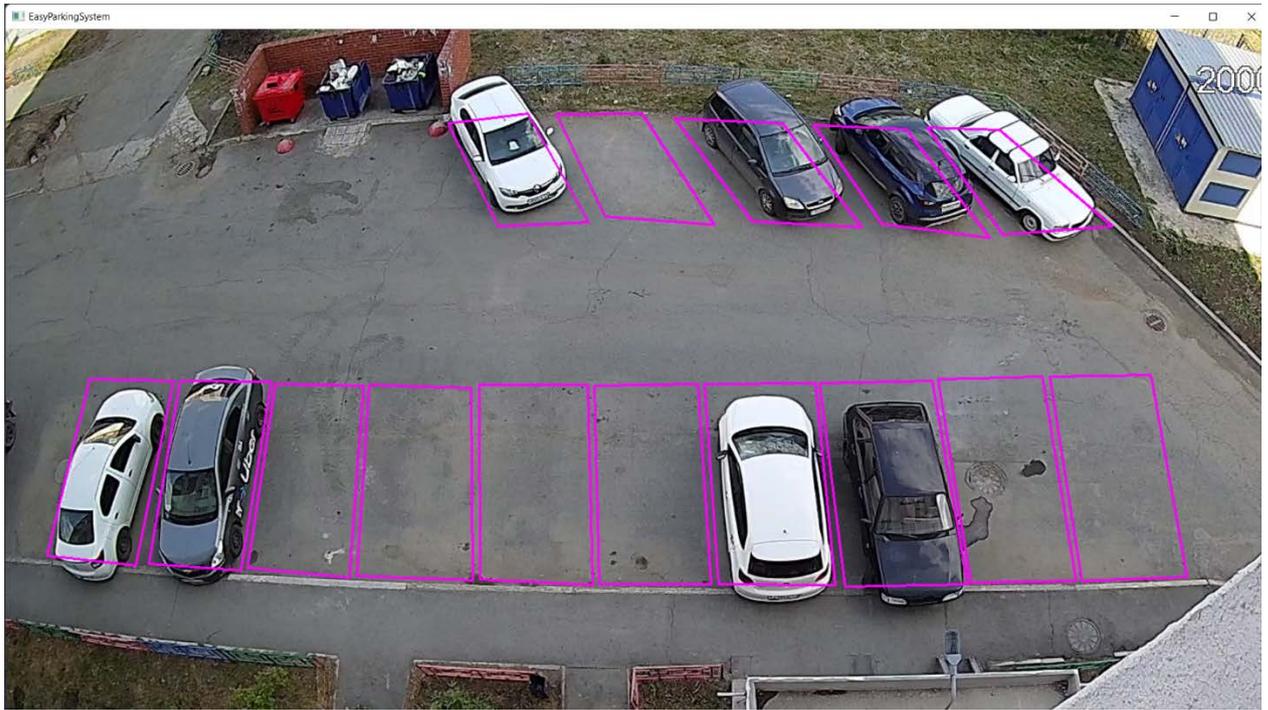


Рисунок 40 – Возвращение к исходной сетке контуров
Результат проведения КЗ – успешно.

5.4 Контрольное задание 4

При переходе в основную программу распознавания. После произведём запуск основного модуля программы, результаты работы представим на рисунке 41.

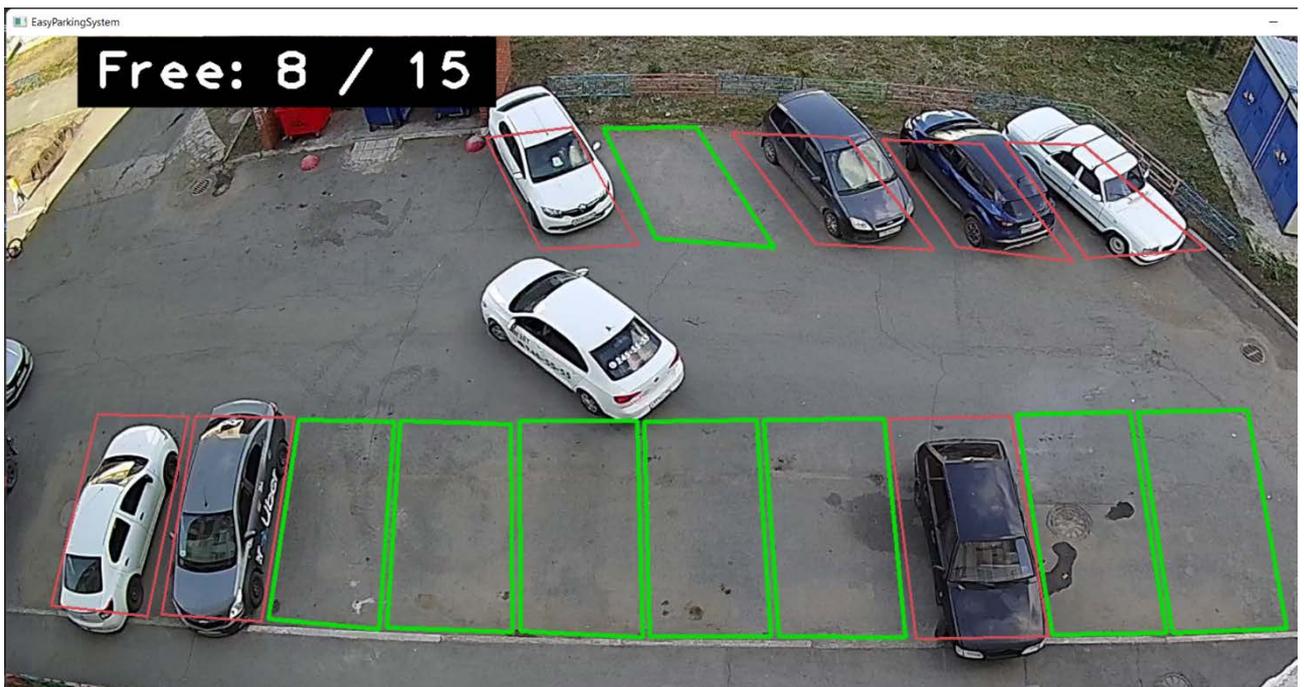


Рисунок 41 – Штатное выполнение программы

Запросим у чат-бота статус на парковке (рисунок 42).

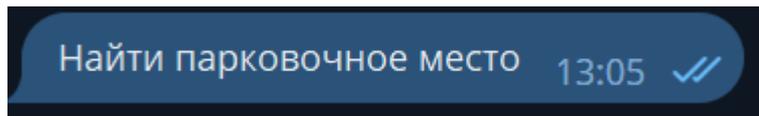


Рисунок 42 – Запрос пользователя статуса парковки

Через 2 секунды в ответ бот присылает скриншот из программы обработки изображения (рисунок 43).

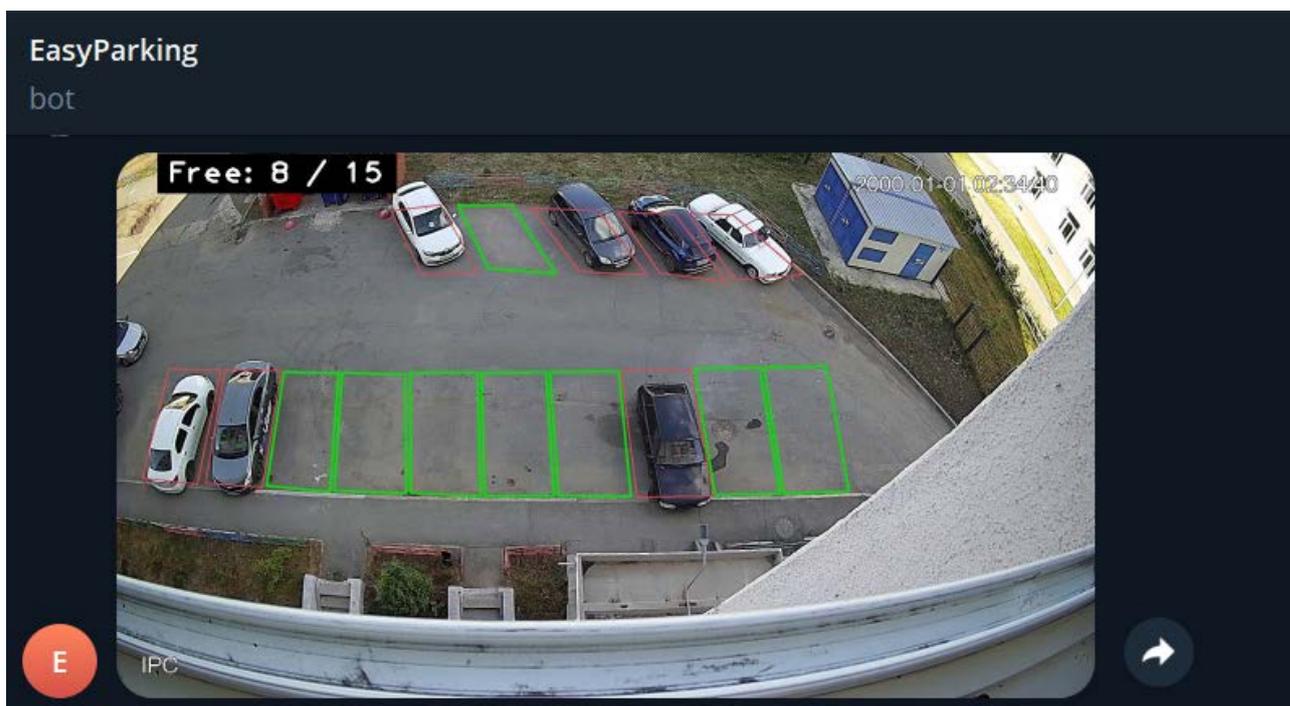


Рисунок 43 – Результат ответа бота на запрос
Результат проведения КЗ – успешно.

6 ОЦЕНКА ПУТИ ДАЛЬНЕЙШЕГО РАЗВИТИЯ И МАСШТАБИРОВАНИЯ СИСТЕМЫ

В дальнейшем планируется расширение функционала системы:

- проектирование web-приложения для пользователей и администрирования;
- подключение цифрового дисплея для увеличения источников вывода результатов работы системы визуализации парковочных мест;
- разработка мобильного приложения для большего охвата пользователей;
- расширение функционала на распознавание номерных знаков.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы было произведено проектирование и реализация системы визуализации парковочных мест, для предоставления помощи водителям при ориентировании на парковочном пространстве.

В работе была затронута проблема поиска свободных парковочных мест на крупных парковках. Предлагаемый вариант решения позволит водителям ускорить поиск свободного места.

По ходу работы были решены следующие задачи:

- был проведен аналитический обзор научно-технической и методической литературы по тематике работы, а также были найдены аналогичные решения поставленной цели;

- определён перечень требований, предъявляемых к разрабатываемому проекту;

- спроектирована архитектура системы;

- разработана система визуализации парковочных мест для отслеживания состояния занятости парковочных мест;

- составлена программа тестовых испытаний, после программа была проведена;

- оценены пути дальнейшего развития и масштабирования системы.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Шапиро, Л. Компьютерное зрение / Л. Шапиро, Дж. Стокман ; пер. с англ. – 4-е изд., электрон. – М. : Лаборатория знаний, 2020. – 763 с. – ISBN978-5-00101-696-0.
- 2 Кэлер, А. Изучаем OpenCV 3 / А. Кэлер, Г. Брэдки ; пер. с англ. А. А. Слинкина. – М. : ДМК Пресс, 2017. – 826 с. – ISBN: 978-5-97060-471-7.
- 3 Патрик, С. Показатели оценки производительности для отслеживания видео / С. Патрик, Ю. В. Вун, Р. Комли // Технический обзор IETЕ. 2011. – 493 с.
- 4 Тамерсой, Б. Вычитание фона / Б. Тамерсой // Конспект лекций. – 2009.
- 5 Куккиара, Р. Вероятностная классификация поз для анализа человеческого поведения, транзакций в системах / Р. Куккиара, К. Грана, А. Прати, Р. Веццани. – Текст : электронный // IEEE Transactions on Systems, Man, and Cybernetics: Systems. – 2005. – С. 42–54.
- 6 Макфарлейн, Н.Дж.Б. Сегментация и отслеживание поросят на изображениях / Н.Дж.Б. Макфарлейн, К.П. Шофилд. – Текст : электронный // Машинное зрение и приложения. – 1995. – С. 187–193.5.
- 7 Кальдерара, С. Надежный фон / С. Кальдерара, А. Прати, Р. Куккиара. – Текст : электронный // Доп. для сложных сцен. – 2006.
- 8 Практикум по визуальному наблюдению. – Текст : электронный // Приложения, Третий IEEE Int. – 2000. – С. 37–45.
- 9 Халеви, Г. Движение возмущений: обнаружение и отслеживание нежесткого движения нескольких тел / Г. Халеви, Д. Вайншолл // Machine Vision and Applications. – 1999. – С. 122–137.
- 10 Пиккарди, М. Методы фонового вычитания: обзор / М. Пиккарди // Международная конференция IEEE по системам / Человек и кибернетика. – 2004. – №4. – С. 3099-3104.
- 11 Понг, П.К.Т. Улучшенная адаптивная модель смещения фона для отслеживания в реальном времени с обнаружением теней / П.К.Т. Понг,

Р. Боуден // Конференция «Семинар по передовым системам видеонаблюдения». – 2001. – 11 с.

12 Харитаоглу, И. Наблюдение за людьми и их действиями в режиме реального времени, анализ закономерностей и машинный интеллект / И. Харитаоглу, Д. Харвуд, Л. Дэвис. – Текст : электронный // Scientific Research: Open Access Library Journal : библиотечный журнал открытого доступа. – 2017. – URL: [https://www.scirp.org/\(S\(351jmbntvnsjt1aadkozje\)\)/journal/paperinformation.aspx?paperid=81262](https://www.scirp.org/(S(351jmbntvnsjt1aadkozje))/journal/paperinformation.aspx?paperid=81262) (дата обращения: 02.05.2023).

13 Мохамад, А.А.Х. Техника вычитания фона адаптивного медианного фильтра с использованием нечеткой логики / А.А.Х. Мохамад, М. Осман. – Текст : электронный // Международная конференция по вычислительной технике, электротехнике и электронной технике (ICSEEE). – 2013. – С. 115-120. – URL: <https://ieeexplore.ieee.org/document/6633917> (дата обращения 02.05.2023).

14 Чинг, С. Надежные методы вычитания фона в видеозаписи городского трафика / С. Чинг, С. Ченг, К. Камат. – Текст : электронный // ResearchGate. – октябрь 2003. – URL: https://www.researchgate.net/publication/228924900_Robust_Techniques_for_Background_Subtraction_in_Urban_Traffic_Video (дата обращения 02.05.2023).

15 Живкович, З. Улучшенная адаптивная гауссовская смешанная модель для вычитания фона / З. Живкович. – Текст : электронный // Proceedings of the 17th International Conference on Pattern Recognition. – 2004. – С. 115-120. – URL: <https://ieeexplore.ieee.org/document/1333992/references#references> (дата обращения 02.05.2023).

16 Штауффер, К. Адаптивные смешанные модели фона для отслеживания в реальном времени / К. Штауффер, В.Э.Л. Гримсон. – Текст : электронный // Международная конференция по компьютерному зрению и распознаванию образов / ResearchGate. – 2007. – URL: https://www.researchgate.net/publication/215722011_Adaptive_background_mixture_models_for_real-time_tracking (дата обращения 02.05.2023).

17 Миттал, А. Вычитание фона на основе движения с использованием адаптивной оценки плотности ядра / А. Миттал, Н. Парагиос // Материалы международной конференции по компьютерному зрению и распознаванию образов / ResearchGate. – 2004. – URL: https://www.researchgate.net/publication/4082269_Motion-based_background_subtraction_using_adaptive_kernel_density_estimation (дата обращения: 22.02.2023).

18 СТРИЖ. – Текст. Изображение (неподвижное ; двухмерное) : электронные // СТРИЖ : [сайт]. – URL: <https://strij.tech/iot-platforma-strij> (дата обращения: 22.02.2023).

19 Стриж: Парковочный датчик присутствия автомобиля. – Текст. Изображение (неподвижное ; двухмерное) : электронные // Tadviser : [сайт]. – URL: <https://clck.ru/34UdYB> (дата обращения: 22.02.2023).

20 ТПУ: Smart Parking Occupation Tracking (SPOT). – Текст. Изображение (неподвижное ; двухмерное) : электронные // Tadviser : [сайт]. – URL: <https://clck.ru/33hBN9> (дата обращения: 22.02.2023).

21 SPOTParking. – Текст. Изображение (неподвижное ; двухмерное) : электронные // SPOT : [сайт]. – URL: <https://checkthespot.ru/#parkings> (дата обращения: 22.02.2023).

22 Taking the hassle out of parking. – Текст. Изображение (неподвижное ; двухмерное) : электронные // Rice University News and Media Relations : [сайт]. – URL: <https://news2.rice.edu/2015/04/28/taking-the-hassle-out-of-parking-2/> (дата обращения: 22.02.2023).

23 Streetline's Solutions. – Текст. Изображение (неподвижное ; двухмерное) : электронные // Streetline : [сайт]. – URL: <https://www.streetline.com/our-solutions> (дата обращения: 22.02.2023).

24 Парковочные войны: тест мобильных приложений для поиска стоянки. – Текст. Изображение (неподвижное ; двухмерное) : электронные // РИА НОВОСТИ: [сайт]. – 21 августа 2017. – URL: <https://ria.ru/20170719/1498774011.html> (дата обращения: 22.02.2023).

25 Сколковская компания поможет водителю найти и оплатить парковку. – Текст. Изображение (неподвижное ; двухмерное) : электронные // ТАСС : [сайт]. – 16 ноября 2017. – URL: <https://tass.ru/skolkovo/4735893> (дата обращения: 22.02.2023).

26 Умная парковка. – Текст. Изображение (неподвижное ; двухмерное) : электронные // Интерсвязь : [сайт]. – URL: <https://vision.is74.ru/parking#rec324792434> (дата обращения: 22.02.2023).

27 Как я нахожу парковочное место за 5 секунд. – Текст. Изображение (неподвижное ; двухмерное) : электронные // Habr : [сайт]. – 29 августа 2022. – URL: <https://habr.com/ru/post/685108/> (дата обращения: 22.02.2023).

28 Как я писал трекинг парковочных мест. – Текст. Изображение (неподвижное ; двухмерное) : электронные // Habr : [сайт]. – 3 декабря 2022. – URL: <https://habr.com/ru/post/703276/> (дата обращения: 22.02.2023).

29 Ищем свободное парковочное место с Python. – Текст. Изображение (неподвижное ; двухмерное) : электронные // Habr : [сайт]. – 10 мая 2019. – URL: <https://habr.com/ru/post/451164/> (дата обращения: 22.02.2023).

30 IPC-HFW2431SP-S-S2. – Текст. Изображение (неподвижное ; двухмерное) : электронные // dahua TECHNOLOGY : [сайт]. – URL: <https://www.dahuasecurity.com/ru/products/All-Products/Network-Cameras/Lite-Series/4-MP/IPC-HFW2431SP-S-S2> (дата обращения: 22.02.2023).

31 EasyParkingSpot. – Текст : электронный // GitLab : [сайт]. – URL: <https://gitlab.com/mr.Hazelnut/easyparkingspot> (дата обращения: 05.05.2023).

32 Витая пара UTP 4 пары AWG 24 категория 5е внутренняя Cu Standart. – Текст. Изображение (неподвижное ; двухмерное) : электронные // ЭТМ : [сайт]. – URL: <https://www.etm.ru/cat/nn/9137530> (дата обращения: 05.05.2023).