

ОТЧЕТ

о проектной работе

по теме: Телеграмм бот для распознавания знаков парковки и остановки

по дисциплине: Проектный практикум 1A

Команда: Персеиды

Тимлид: Козулина Надежда РИ-140931

Аналитик: Андреева Светлана РИ-140941

Тестировщик: Стрельникова София РИ-140931

Бэкенд-разработчик: Борков Владимир РИ-140911

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_heading=h.br7h4ewoer5z)

[1. Целевая аудитория 5](#_heading=h.uh359f5ca0uc)

[2. Определение проблемы 8](#_heading=h.6bi3qt9g81xt)

[3. Подходы к решению проблемы 10](#_heading=h.hoalrdvzedj)

[4. Анализ аналогов 13](#_heading=h.4blivgpdj75e)

[5. Сценарии использования 15](#_heading=h.fpuecmxfarbz)

[6. Требования к продукту и к MVP 16](#_heading=h.c4bp3aou1t97)

[7. Стек для разработки 19](#_heading=h.mxcchcfg2d7t)

[8. Прототипирование 22](#_heading=h.ppjfh5xfylkd)

[9. Проектирование и разработка системы 25](#_heading=h.w6fwt6ddstpl)

[ЗАКЛЮЧЕНИЕ 29](#_heading=h.yijsci5436af)

[СПИСОК ЛИТЕРАТУРЫ 34](#_heading=h.jsfane4hoaw8)

# **ВВЕДЕНИЕ**

Современные мегаполисы сталкиваются с растущей проблемой неправильной парковки, приводящей к штрафам, эвакуации автомобилей и снижению эффективности городского движения. Несмотря на развитие цифровых сервисов, водители продолжают испытывать трудности в интерпретации знаков, особенно в нестандартных ситуациях: при наличии дополнительных табличек, временных ограничений или региональных исключений. Проблема особенно остро стоит перед начинающими водителями, туристами, пользователями каршеринга и теми, кто редко садится за руль.

Проведенный опрос показал, что 50% водителей получали штрафы за неправильную парковку, а 89% участников сталкивались с трудностями в понимании знаков, особенно временных. При этом 88% выразили готовность использовать Telegram-бота, если он поможет быстро понять, можно ли парковаться в конкретном месте. Это свидетельствует о реальной потребности в доступном инструменте, который позволил бы оперативно интерпретировать знаки в городских условиях.

Существующие решения — карты, навигаторы и справочники ПДД — не решают проблему полностью. Они либо не учитывают временные и комбинированные знаки, либо требуют от пользователя значительного времени на поиск и анализ. В условиях плотного трафика и ограниченного времени на принятие решений это становится критически неудобно. Особенно уязвимы водители, арендующие авто, не знакомые с регионом, и новички без уверенных навыков интерпретации дорожной обстановки.

В рамках проекта разрабатывается Telegram-бот, использующий нейросеть для распознавания дорожных знаков, связанных с парковкой, по фотографии. Бот предоставляет мгновенный ответ — можно ли парковаться или останавливаться в данном месте с учётом всех табличек и комбинаций. Это узкоспециализированное решение сфокусировано на одной важной задаче, игнорируемой большинством крупных сервисов, и способно существенно снизить количество правонарушений, повысить уверенность водителей и предотвратить неоправданные штрафы.

Цель проекта:

Создание интуитивно понятного Telegram-бота, способного с помощью компьютерного зрения распознавать знаки, связанные с парковкой и остановкой, и предоставлять пользователю точную и оперативную информацию о действующих ограничениях.

Задачи проекта:

* Исследовать типы дорожных знаков, регулирующих парковку и остановку, включая дополнительные таблички и временные ограничения.
* Разработать и обучить модель нейросети для распознавания данных знаков по изображению.
* Интегрировать модель с Telegram-ботом, способным мгновенно реагировать на пользовательский запрос.
* Сформировать базу правил для всех распознаваемых ситуаций с возможностью последующего обновления.
* Провести тестирование на реальных пользователях и сценариях, оценив точность, удобство и скорость системы.

# **Целевая аудитория**

Определение целевой аудитории проекта выполнено с применением методики 5W (What? Who? Why? When? Where?) Марка Шеррингтона. Такой подход позволяет детально сегментировать потребителей и выявить основные мотивации к использованию продукта.

Что мы предлагаем?

Проект представляет собой Telegram-бота, который с помощью нейросети распознаёт дорожные знаки парковки/остановки по фото и предоставляет пользователю понятную информацию о действующих ограничениях. Это удобный инструмент для быстрого получения информации без необходимости самостоятельно изучать ПДД.

Кто целевая аудитория?

На основе анализа результатов опроса (n >150) и выявленных поведенческих паттернов, ключевые сегменты целевой аудитории следующие:

* Пользователи каршеринга — 26%
 Часто паркуются в незнакомых местах, не всегда уверены в региональных правилах.
* Начинающие водители (до 1 года стажа) — 23%
 Только начинают ориентироваться в реальной дорожной обстановке, сталкиваются с неуверенностью в знаках и сочетаниях табличек.
* Водители, редко садящиеся за руль — 25%
 Могут подзабыть значение знаков и нюансы временных или комбинированных ограничений.
* Туристы и иностранцы — 18%
 Плохо ориентируются в местных ПДД и не знакомы с региональными особенностями обозначения знаков.

Примечание: Опытные водители с постоянным стажем вождения (8% от выборки) не входят в целевую аудиторию, поскольку чаще уверены в собственных знаниях и не ощущают потребности в подобном помощнике.

Социально-демографический портрет:

* Пол: мужчины и женщины;
* Возраст: от 18 лет;
* География: крупные города и мегаполисы (Москва, Санкт-Петербург, Екатеринбург и др.);

Почему потребители будут пользоваться ботом?

* Чтобы быстро и без ошибок понять, можно ли парковаться в конкретном месте;
* Избежать штрафов и эвакуации;
* Получить моментальную помощь без установки отдельных приложений — прямо в мессенджере.

Когда нужен продукт?

* В момент поиска парковки в незнакомом районе;
* При сомнении в значении сложного или временного знака;
* Во время временных ограничений (например, в выходные или при дорожных работах);
* При аренде автомобиля, когда водитель не знаком с районом и знаками

Где принимается решение о пользовании?

* В Telegram, на мобильном устройстве;
* На улице, в момент парковки или перед ней;
* В автомобиле, во время краткой остановки;

# **Определение проблемы**

В процессе подготовки проекта была проведена комплексная работа по выявлению реальных проблем, с которыми сталкиваются водители при парковке в городских условиях. Для этого применялись следующие методы: анкетирование целевой аудитории, анализ пользовательских отзывов, изучение данных о правонарушениях, а также конкурентный анализ существующих решений.

На основе проведенного опроса среди водителей (n > 150) были получены следующие результаты, подтверждающие наличие значительной проблемы:

* 50% опрошенных сообщают, что получали штрафы за неправильную парковку;
* 70% иногда испытывают трудности при интерпретации знаков парковки/остановки, 19% — часто, и только 11% не сталкивались с такой проблемой;

Дополнительно, анализ пользовательских отзывов на картах и навигационных сервисах (Яндекс.Карты, Google Maps и др.) показал, что пользователи часто жалуются на следующие аспекты:

* Непонимание временных ограничений, табличек и дополнительных знаков;
* Недостаточная информативность существующих сервисов;
* Отсутствие простого и быстрого способа получить однозначный ответ: «Можно ли здесь парковаться?».

Проблема проявляется на нескольких уровнях:

1. Технический уровень:
 Водители сталкиваются с ситуациями, когда:

* Знаки содержат сложные или противоречивые комбинации (например, таблички "по четным дням", "работает по будням", "кроме служебного транспорта");
* Используются временные знаки, которые не отражены в навигационных системах;

2. Финансовый уровень:
 Ошибки в трактовке знаков приводят к:

* Штрафам (в среднем от 1 000 до 3 000 рублей за неправильную парковку);
* Эвакуации автомобиля (до 7 000 рублей);

3. Поведенческий и эмоциональный уровень:

* Пользователи испытывают неуверенность и стресс из-за страха получить штраф;
* Сомневаются в собственных знаниях ПДД, особенно в нестандартных ситуациях;

Выявленные барьеры:

* Сложность и неэффективность самостоятельной интерпретации дорожных знаков;
* Недостаток уверенности у начинающих водителей;
* Отсутствие решения, которое интегрировано в привычную среду общения — Telegram.

Общая проблема — недостаточная уверенность в интерпретации правил парковки — актуальна для всех перечисленных групп. Эти категории водителей часто оказываются в новых или непривычных ситуациях и нуждаются в инструменте, который поможет им быстро принять верное решение.

# **Подходы к решению проблемы**

Эффективное решение проблемы начинается с глубокого анализа текущей ситуации, понимания потребностей пользователей и выработки максимально широкого набора альтернативных решений.

В рамках проекта применялись следующие методы для генерации и отбора решений:

* Мозговые штурмы внутри команды;
* Поиск нестандартных, технологически возможных решений с высокой эффективностью;
* Анализ существующих решений на рынке: боты, справочники, мобильные приложения по ПДД.

На этапе проектирования были рассмотрены различные варианты устранения проблемы распознавания знаков и проверки правил парковки:

1. Создание мобильного приложения с возможностью загрузки фото дорожного знака и получения расшифровки:
	* Минусы: долгий путь установки/регистрации, зависимость от платформ, сложность поддержки.
	* Плюсы: больше функционала, возможна интеграция с картами и ГИБДД.
2. Создание Telegram-бота без нейросети, в котором пользователь вручную выбирает знак из списка:
	* Минусы: сложность использования в реальных условиях, занимает много времени, нет автоматизации.
	* Плюсы: технически просто реализуем, не требует обучения модели.
3. Telegram-бот с нейросетью, распознающей знак с фото и выдающей информацию о правилах парковки:
	* Минусы: требуется обучение модели, настройка взаимодействия бота с моделью.
	* Плюсы: высокая скорость, простота использования, масштабируемость, хорошая UX-архитектура.
4. Интеграция с внешними сервисами (например, API карт):
	* Минусы: низкая точность по табличкам и дополнительным условиям, отсутствие анализа изображений.
	* Плюсы: нет необходимости в обучении собственной модели.

После анализа всех вариантов по ключевым критериям (удобство пользователя, точность, скорость обработки, масштабируемость), предпочтение было отдано Telegram-боту с интегрированной нейросетью, как наиболее сбалансированному решению

| **Критерий** | **Мобильное приложение** | **Бот без нейросети** | **Бот с нейросетью** | **Внешний сервис** |
| --- | --- | --- | --- | --- |
| Удобство пользователя | Среднее | Низкое | Высокое | Среднее |
| Скорость обработки | Средняя | Средняя | Высокая | Низкая |
| Масштабируемость | Средняя | Средняя | Высокая | Высокая |
| Точность результата | Средняя | Низкая | Высокая | Средняя |

# **Анализ аналогов**

Для определения конкурентов продукта был проведён анализ прямых и косвенных аналогов. Целью анализа стало выявление сильных и слабых сторон существующих решений, а также определение уникальности и конкурентных преимуществ нашего продукта.

Цели анализа:

* Понять, какие альтернативные решения уже существуют;
* Определить, какие из них действительно удобны и полезны для пользователей;
* Выявить незакрытые потребности и боли целевой аудитории;
* Найти возможности для улучшения и отличия от конкурентов.

| Тип ресурса | Примеры | Особенности работы |
| --- | --- | --- |
| Интернет-ресурсы | Drom.ru, Auto.ru и др. | Пользователь вручную ищет знак или загружает фото; требуется участие пользователя для анализа. |
| Книжные источники | Учебники ПДД, справочники | Самостоятельный поиск по изображениям; отнимает время, нет автоматизации. |
| Мобильные приложения | "Дорожные AR знаки", RoadAR и др. | Автоматическое распознавание по фото; требует установки, нередко перегружены интерфейсом. |
| Наш Telegram-бот | --- | Фото загружается в чат, бот сам анализирует знаки и выводит нужную информацию в диалоге. |

Таблица 1 - Особенности работы

Прямые конкуренты:

* Мобильные приложения с AR-распознаванием знаков (например, RoadAR) – позволяют определять дорожные знаки через камеру, но работают в формате отдельных приложений.
* Сайты и интернет-сервисы, где можно загрузить фото или вручную искать знак по визуальному сходству – удобны при стабильном интернете, но менее эффективны из-за необходимости самостоятельного анализа.
* Книжные источники – актуальны при отсутствии техники, но крайне неудобны в условиях города и требуют времени и знаний.

В таблице 2 представлен пример анализа конкурентов по трем критериям.

|  | Drom.ru | Книжные источник | Дорожные AR знаки | Наш телеграмм-бот |
| --- | --- | --- | --- | --- |
| Возможность распознать знак по фото | **-** | **-** | **+** | **+** |
| Возможность больше узнать о знаке | **+** | **+** | **+** | **+** |
| Часто и быстро обновляемый контент | **+** | **-** | **+** | **+** |
| Возможность получить бесплатный доступ к информации | есть | нет, нужно купить книгу | есть | есть |
| Быстрое получение информации о знаке  | - | – | - | + |

Таблица 2 – Анализ конкурентов

# **Сценарии использования**

В рамках реализации Telegram-бота для распознавания дорожных знаков были разработаны и проанализированы типовые сценарии его использования. Сценарии охватывают основные ситуации, в которых целевая аудитория может взаимодействовать с системой.

1. Получение информации о незнакомом дорожном знаке
 Пользователь фотографирует дорожный знак, значение которого ему неизвестно, и отправляет изображение в бот. Бот определяет тип знака и предоставляет расшифровку и описание. Это помогает пользователю быстро понять, как нужно действовать в соответствии с правилами дорожного движения.
2. Распознавание нескольких знаков на одном изображении
 В случаях, когда на участке дороги или парковке установлено несколько дорожных знаков, пользователь делает общее фото и отправляет его в бот. Система определяет все распознанные знаки на изображении и выдаёт информацию о каждом. Это упрощает навигацию и соблюдение правил на сложных участках дороги.
3. Проверка правильности установки знаков на парковках и во дворах
 Пользователь может зафиксировать знак, который вызывает сомнение (например, подозрительно установленный или самодельный). После отправки фото в бот, пользователь получает официальную информацию о данном знаке.

# **Требования к продукту и к MVP**

Для построения полной картины требований использовалась иерархическая модель: от пользовательских задач — к функциональным и нефункциональным характеристикам, необходимым для реализации минимально жизнеспособного продукта (MVP).

1. Требования клиентов (пользовательские требования)

* Водители, желающие быстро получить информацию о незнакомом дорожном знаке;
* Участники дорожного движения, заинтересованные в проверке корректности размещения знаков;

Основные пользовательские задачи:

* Узнать значение неизвестного дорожного знака;
* Получить быструю справку по нескольким знакам;
* Использовать бота в ситуациях неуверенности в расшифровки знаков.

2. Функциональные требования

* Возможность отправки изображения дорожного знака через Telegram-бот;
* Распознавание одного или нескольких знаков на изображении;
* Вывод расшифровки и объяснения каждого знака;
* Обработка ошибок (например, если знак не распознан).

3. Нефункциональные требования

* Время ответа системы не должно превышать 5 секунд при стабильном соединении;
* Система должна быть доступна 24/7;
* Интерфейс Telegram-бота должен быть интуитивно понятным для пользователей любого уровня;
* Надежность: бот должен устойчиво обрабатывать запросы и исключения;
* Обеспечение безопасности пользовательских данных (не хранить фотографии без необходимости).

4. Производные требования

* Использование модели YOLOv7-tiny как компромисса между качеством распознавания и быстродействием;
* Фото должно сохраняться временно на сервере для обработки, с последующим удалением;
* Telegram-бот должен уметь запускать модель через subprocess с передачей параметров;
* Сервер должен иметь минимально необходимые ресурсы для обработки изображений в реальном времени;
* Возможность масштабирования решения при росте числа пользователей (потенциал для размещения модели в облаке или контейнеризации).

MVP (Минимально жизнеспособный продукт)

Для MVP определены следующие обязательные компоненты:

* Telegram-бот с возможностью загрузки фото;
* Алгоритм обработки фото и вызова модели YOLOv7-tiny;
* Интерфейс выдачи пользователю результата — текстовое описание найденных знаков;
* Обработка типичных ошибок: слишком большое изображение, отсутствие знаков и пр.;
* Минимальная документация для пользователей (в т.ч. инструкция по использованию).

В MVP не включаются:

* Личный кабинет пользователя;
* Механизмы сохранения истории запросов;
* Поддержка многоязычности.
* Интеграцию с картами и сторонними сервисами для поиска парковки
* Голосовой ввод запросов

# **Стек для разработки**

В рамках реализации проекта по созданию Telegram-бота для распознавания дорожных знаков на парковке был выбран стек, который позволил быстро запустить MVP, сосредоточившись на ключевой функциональности.

Язык программирования - Python.
 Выбран за простоту, лаконичность, широкую распространенность, а также наличие большого количества библиотек и инструментов для быстрой разработки, особенно в области машинного обучения и создания ботов.

Telegram-бот

* Библиотека telebot (pyTelegramBotAPI)
 Позволяет удобно создавать и настраивать Telegram-ботов. Имеет простой API, отличную документацию и активно поддерживается сообществом.
* Модули os, subprocess
 Используются для обработки файлов, запуска внешних процессов и интеграции нейросети в обработку запросов пользователя.

Нейросеть - YOLOv7-tiny
 Легковесная модель нейросети для детектирования объектов. Обеспечивает баланс между производительностью и точностью, что критично при работе в реальном времени. Позволяет оперативно получать результаты даже при ограниченных ресурсах.

Данные для обучения - открытый датасет дорожных знаков.
 Использована база изображений. Данные были разделены на обучающую, валидационную и тестовую выборки, что позволило построить корректную и воспроизводимую модель.

Организация архитектуры - разделение логики
 Telegram-бот и модель работают как отдельные компоненты, связанные через систему вызова внешних процессов. Это позволило:

* легко масштабировать проект в будущем;
* отдельно дорабатывать части системы без полного переписывания;
* упростить отладку и повторное использование.

Выбранный стек ориентирован на быструю реализацию минимально жизнеспособного продукта (MVP) с максимальной переиспользуемостью существующих решений. Такой подход позволил:

* минимизировать технические риски;
* сосредоточиться на распознавании дорожных знаков;

Использованные технологии просты, доступны, понятны команде и обладают высокой перспективой поддержки. Стек подобран не из соображений «идеальности», а исходя из практичности, гибкости и готовности к быстрому запуску проекта в условиях реального времени.

# **Прототипирование**

Создание прототипа является важным этапом в разработке программного продукта, позволяющим быстро визуализировать основные идеи проекта, протестировать пользовательский интерфейс и внести корректировки до начала полной разработки.

Процесс создания прототипа в рамках проекта прошёл следующие стадии:

1. Определение начальных требований
 Были зафиксированы базовые сценарии взаимодействия пользователя с системой: загрузка фотографии дорожного знака, получение информации и отзывчивость интерфейса.
2. Разработка первого варианта прототипа
 Был создан интерфейс Telegram-бота, реализующий базовый визуальный сценарий: отправка фотографии, автоматический запуск нейросети и получение результата. Интерфейс Telegram сам по себе минималистичен и ограничен рамками мессенджера, что упростило реализацию.
3. Получение обратной связи
 Проведено тестирование прототипа с участием ральных пользователей. Были собраны замечания по логике взаимодействия, скорости отклика, а также предложены идеи по улучшению формулировок ответов от бота.
4. Переработка прототипа
 Внесены изменения в структуру ответов бота, добавлены поясняющие сообщения, а также реализован вывод нескольких возможных вариантов знаков, если нейросеть выдает несколько гипотез.

Характеристики эффективного прототипа

* Прототип Telegram-бота был собран за минимально возможное время, благодаря использованию готовых библиотек и фреймворков (telebot, YOLOv7-tiny и др.), что позволило быстро начать тестирование.
* Основное внимание уделено взаимодействию пользователя с системой — распознаванию фото и удобству восприятия ответа. Интерфейсная часть Telegram была реализована с фокусом на UX.
* Прототип создавался как временное решение, чтобы зафиксировать ключевые сценарии, протестировать идеи и получить раннюю обратную связь. Использованные средства позволили минимизировать затраты.
* Прототип легко модифицировался на всех этапах, благодаря модульной архитектуре (отдельные части для UI, обработки изображений, вывода результатов).

Эволюция требований к прототипу

На ранней стадии ключевыми были скорость создания и простота модификации, что обеспечивало возможность быстрой доработки. По мере стабилизации прототипа на первый план начали выходить функциональность, надежность работы и визуальная цельность ответов в Telegram-чатах.

# **Проектирование и разработка системы**

На этапе проектирования была сформирована общая структура бота, включающая нейросеть для распознавания дорожных знаков и Telegram-бот, осуществляющий взаимодействие с пользователем.

Архитектура системы

Программный комплекс включает в себя следующие ключевые модули:

1. Модуль Telegram-бота

Язык: Python

Библиотека: telebot

Задачи:

* + - Приём изображений от пользователя;
		- Обработка сообщений и команд;
		- Передача изображения на вход нейросети;
		- Получение результата и вывод пользователю;
		- Обработка ошибок и недопустимых входов.
1. Модуль обработки изображений

Язык: Python

Библиотеки: os, cv2, subprocess

Задачи:

* + - Сохранение изображения на диск;
		- Подготовка файла к подаче в модель;
		- Запуск предобученной нейросети через внешнюю оболочку;
		- Передача результатов обратно в основной поток.
1. Модуль нейросети

Фреймворк: PyTorch
Модель: YOLOv7-tiny

Задачи:

* + - Распознавание дорожных знаков на изображении;
		- Формирование набора гипотез;
		- Вывод класса знака и степени уверенности;
		- Визуализация распознанного знака (по возможности).

Алгоритм работы приложения

* Пользователь отправляет фотографию знака в Telegram-бот;
* Бот сохраняет изображение локально;
* Изображение передаётся на вход модулю нейросети;
* YOLOv7-tiny выполняет распознавание знаков;
* Результаты распознавания (название знака и вероятность) отправляются обратно в бот;
* Бот формирует текстовый ответ и отправляет его пользователю.

Разработка велась поэтапно:

1. Первичная реализация Telegram-бота, способного принимать и сохранять изображения;
2. Интеграция с нейросетью, уже прошедшей предварительное обучение на датасете дорожных знаков;
3. Отладка взаимодействия между модулями — обеспечение корректной передачи файлов, стабильный запуск нейросети из Python-скрипта;
4. Формирование финальной логики ответа — добавление обработки ошибок, исключительных ситуаций, форматирование результата;
5. Тестирование и сбор обратной связи — выявление точек отказа, корректировка интерфейса и скорости работы.

# **ЗАКЛЮЧЕНИЕ**

В ходе выполнения проекта был разработан бот для распознавания дорожных знаков на парковках с помощью нейросетей, интегрированная в Telegram-бот. Целью проекта являлось создание удобного и быстрого в использовании решения, способного распознавать дорожные знаки по фотографии, предоставляемой пользователем, и предоставлять результат в интерактивной форме через мессенджер.

В рамках проекта были выполнены следующие ключевые задачи:

1. Проведён анализ предметной области, в том числе изучены особенности дорожных знаков на парковках и сложности, возникающие при их распознавании;
2. Определены пользовательские, функциональные и нефункциональные требования к системе, выделены требования к MVP;
3. Разработаны сценарии использования, описывающие поведение системы в ситуациях взаимодействия пользователя с Telegram-ботом;
4. Подобран стек технологий, соответствующий требованиям к скорости разработки, масштабируемости и простоте поддержки;
5. Произведено проектирование системы, реализован набор модулей: Telegram-бот, модуль запуска нейросети, модуль хранения изображений и предобработки, нейросетевая модель;
6. Проведено первичное прототипирование и тестирование пользовательского интерфейса;
7. Разработан полноценный рабочий прототип системы, способный функционировать в условиях реального использования.

Таким образом, были достигнуты все поставленные цели, а бот спроектирован таким образом, чтобы в будущем его масштабировать.

Положительные и отрицательные стороны решения

Положительные стороны:

* Использование готовых и проверенных компонентов (YOLOv7-tiny, telebot), что обеспечило быструю разработку и стабильную работу;
* Высокая скорость обработки изображений;
* Интуитивный и привычный для пользователей интерфейс взаимодействия через Telegram;
* Модульная архитектура, позволяющая легко заменять отдельные компоненты или дорабатывать функциональность;
* Поддержка основных пользовательских сценариев с минимальными временными задержками.

Отрицательные стороны:

* Отсутствие собственных алгоритмов обучения и дообучения нейросети в текущей версии (используется предобученная модель);
* Зависимость от качества входного изображения и условий съёмки (освещение, ракурс, искажения);
* Отсутствие графического интерфейса помимо Telegram, что может ограничивать применение в других сценариях (например, в веб-приложении).

Перспективы развития

Разработанный продукт может быть полезен в различных сферах, включая парковочные комплексы, логистику. Однако текущая реализация является лишь первым шагом в развитии проекта. В перспективе планируется:

1. Расширение набора распознаваемых знаков и типов дорожной разметки;
2. Создание веб-интерфейса с расширенными возможностями визуализации и загрузки изображений;
3. Дообучение модели на локальных выборках, включающих редкие, повреждённые знаки;
4. Интеграция с городскими информационными системами или административными базами данных;
5. Разработка мобильного приложения как альтернативы Telegram-боту, с возможностью работы в офлайн-режиме.

Проект можно считать успешным с точки зрения выполнения поставленных задач. Он продемонстрировал, как с использованием современных подходов к быстрой разработке можно реализовать работоспособное решение, отвечающее реальной прикладной потребности.

Основной акцент в дальнейшей разработке будет сделан на устойчивость к разнообразию условий съёмки, удобство использования и расширение функциональности. Это позволит в будущем превратить прототип в полноценный значимый продукт.

# **СПИСОК ЛИТЕРАТУРЫ**

1. Правила дорожного движения Российской Федерации / Pddmaster.ru. – URL: <https://pddmaster.ru/pdd> / (дата обращения: 21.05.2025)
2. ГИБДД РФ. Официальный сайт / Gibdd.ru. – URL:<https://gibdd.ru/> (дата обращения: 21.05.2025).
3. OpenStreetMap – база данных дорожной инфраструктуры / OpenStreetMap.org. – URL:<https://www.openstreetmap.org/> (дата обращения: 21.05.2025).
4. Telegram Bot API – документация / Telegram.org. – URL: <https://core.telegram.org/bots/api> (дата обращения: 21.05.2025).
5. Хабр. Разработка Telegram-ботов на Python / Habr.com. – URL: <https://habr.com/ru/post/358404/> (дата обращения: 21.05.2025).
6. MSU Traffic Sign Dataset. Датасет дорожных знаков / Graphics.cs.msu.ru. – URL:<https://graphics.cs.msu.ru/projects/traffic-sign-recognition.html> (дата обращения: 21.05.2025).
7. VC.RU. Как составить портрет клиента (целевой аудитории) / VC.RU. – 2020. – URL:<https://vc.ru/marketing/156147-kak-sostavit-portret-klienta-celevoy-auditorii-instrukciya-s-primerami> (дата обращения: 21.05.2025).
8. Киселёв А. А. SMART-цели в управлении проектами // Управление проектами и программами. – 2021. – №3. – С. 36–39. – URL: <https://cyberleninka.ru/article/n/smart-tseli-v-upravlenii-proektami> (дата обращения: 21.05.2025).
9. Маркетинг на практике. Целевая аудитория и сегментация / Skillbox. – URL: <https://skillbox.ru/media/marketing/> (дата обращения: 21.05.2025).
10. Финогенов А. Е. Методы изучения целевой аудитории: практический подход // Маркетинг и маркетинговые исследования. – 2022. – № 2. – С. 20–26.