

Интеллектуальная информационная система учета и инвентаризации дорожных знаков ПДД

А. И. Абдулнагимов • Б. В. Галицков • О. М. Ахунов

Уфимский университет науки и технологий

Работа посвящена совершенствованию методов инвентаризации дорожных знаков, которые характеризуются высокой трудоемкостью и затратами. Ручной осмотр и фиксация знаков позволяют обрабатывать один километр дороги от 2 до 3 часов, при этом затраты на проведение работ возрастают на 70–80 %. В работе представлена интеллектуальная информационная система автоматической детекции и классификации дорожных знаков ПДД с использованием современных методов глубокого обучения. Разработана двухэтапная архитектура на основе моделей YOLO: детектор YOLOv11m для обнаружения знаков на видеопотоке и классификатор YOLOv8 для определения типа знака по ГОСТ Р 52290-2004. Для решения проблемы дисбаланса классов в обучающей выборке применена синтетическая генерация данных на основе векторных изображений знаков. Реализован алгоритм трекинга MultiSignTracker для устранения дубликатов детекций. Система обеспечивает автоматическое извлечение GPS-координат из метаданных видео и привязку их к обнаруженным знакам. Разработан графический интерфейс для управления проектами и визуализации результатов. Система поддерживает экспорт данных в форматы IndorTrafficPlan и KML. Точность детекции составляет 99,64 %. Итоговый датасет включает около 35 750 изображений, охватывающих около 200 классов дорожных знаков.

Компьютерное зрение; детекция объектов; YOLO; дорожные знаки; глубокое обучение; классификация изображений; трекинг объектов.

ВВЕДЕНИЕ

Обеспечение безопасности дорожного движения является одной из приоритетных задач государственной политики Российской Федерации. Согласно данным Госавтоинспекции МВД России, около 20 % дорожно-транспортных происшествий связаны с неудовлетворительным состоянием технических средств организации дорожного движения, в том числе дорожных знаков¹. Своевременный мониторинг и инвентаризация дорожных знаков позволяют снизить аварийность и повысить качество дорожной инфраструктуры.

Протяженность федеральных автомобильных дорог Российской Федерации составляет 50,8 тысяч километров, что требует постоянного мониторинга состояния дорожных знаков². Традиционные методы инвентаризации, основанные на ручном осмотре и фиксации знаков, характеризуются высокой трудоемкостью: обработка одного километра дороги занимает от 2 до 3 часов, при этом затраты на проведение работ возрастают на 70–80 % по сравнению с автоматизированными решениями³.

¹ Госавтоинспекция МВД России. Статистика дорожно-транспортных происшествий. URL: <http://stat.gibdd.ru> (дата обращения: 01.10.2026).

² Федеральное дорожное агентство. Протяженность федеральных автомобильных дорог Российской Федерации. 2023.

³ Росавтодор. Методические рекомендации по инвентаризации технических средств организации дорожного движения. 2022.

Развитие технологий компьютерного зрения и глубокого обучения открывает новые возможности для автоматизации процессов детекции и классификации объектов на изображениях. Современные нейросетевые архитектуры, такие как YOLO (You Only Look Once), демонстрируют высокую точность распознавания при работе в режиме реального времени [Red16]. Применение подобных технологий к задаче инвентаризации дорожных знаков позволяет существенно сократить временные и финансовые затраты, а также повысить достоверность собираемых данных.

Согласно ГОСТ Р 52289-2019⁴ и ГОСТ Р 52290-2004⁵ в Российской Федерации используется около 280 типов дорожных знаков, разделенных на 8 групп. Создание системы, способной распознавать все типы знаков в соответствии с действующими стандартами, представляет собой актуальную научно-техническую задачу.

Цель исследования — повышение эффективности процесса инвентаризации дорожных знаков путем разработки интеллектуальной информационной системы учета и инвентаризации дорожных знаков ПДД с привязкой к географическим координатам.

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ известных методов и моделей машинного обучения для детекции объектов на изображениях;
- сравнить архитектуры нейронных сетей для задачи детекции дорожных знаков;
- сформировать датасет для обучения моделей с учетом требований ГОСТ;
- разработать архитектуру системы детекции и классификации дорожных знаков;
- реализовать механизмы фильтрации ложных срабатываний и трекинга объектов;
- разработать модуль извлечения и привязки GPS-координат к обнаруженным знакам;
- оценить качество работы алгоритмов информационной системы (ИС).

Структура статьи: дается обзор литературы и методов детекции объектов; подготавливаются данные и формируется датасет; рассматривается эволюция подходов к построению системы; описывается архитектура и реализация системы; оценивается работа ИС.

ОБЗОР ЛИТЕРАТУРЫ И МЕТОДОВ

Детекция и классификация дорожных знаков

Исследования и разработки в области автоматизации инвентаризации дорожных знаков активно развиваются благодаря достижениям в области искусственного интеллекта и компьютерного зрения. Наиболее успешные подходы основываются на глубоком обучении и используются как в научных публикациях, так и в реальных приложениях. Современные исследования направлены на повышение точности распознавания в сложных условиях, например, ночное освещение, плохая погода и др., а также интеграцией с системами навигации и управления движением транспортных потоков.

Ниже представлен краткий обзор основных направлений исследований и существующих разработок в данной области российских и зарубежных научных групп.

В работе [Хар22] авторы предложили модель распознавания дорожных знаков на основе сверточной нейронной сети (YOLOv7) для классификации российских дорожных знаков, соответствующих стандарту ГОСТ Р 52290-2004. Модель обладает точностью предсказаний, равной 0,847 при полноте предсказаний в 0,811. Методология, предложенная в работе, представляет собой эффективный подход к преодолению ограничений при создании высококачественных моделей распознавания.

В работе [Шул24] предложена методология распознавания дорожных знаков с использованием ансамбля сверточных нейронных сетей, представляющего собой модификацию ранее

⁴ ГОСТ Р 52289-2019. Технические средства организации дорожного движения. Правила применения дорожных знаков, разметки, светофоров, дорожных ограждений и направляющих устройств. М.: Стандартинформ, 2020.

⁵ ГОСТ Р 52290-2004. Технические средства организации дорожного движения. Знаки дорожные. Общие технические требования. М.: Стандартинформ, 2006.

опубликованного подхода. Основная цель модификации заключается в повышении производительности распознающей системы без существенного снижения точности за счёт удаления избыточных данных из тренировочного набора. Метод включает две важные компоненты: 1) нейросеть-классификатор погодных условий, определяющая необходимость предварительной обработки изображения; 2) модифицированная нейросеть YOLO v4, заменившая ранее используемую сеть типа SegU-Net и классификаторы типа VGG16. Благодаря этому достигается значительное улучшение скорости распознавания при незначительном снижении точности (около 5 %) в неблагоприятных погодных условиях.

Статья [Nua19] посвящена применению глубоких нейронных сетей для автоматического распознавания и классификации дорожных знаков в системе ADAS на основе архитектуры VGGNet, адаптированной для американских дорожных знаков из набора данных LISA TS. Результаты показывают высокую точность классификации — 99,04 %, что подтверждает эффективность предложенного подхода для расширения возможностей систем распознавания дорожных знаков в США.

В обзорной статье [Che24] описываются современные методы автоматического распознавания дорожных знаков с использованием глубокого обучения. В ней анализируются традиционные, deep learning и гибридные алгоритмы, включая регрессию, сегментацию и комбинацию методов, такие как SSD, YOLO, Faster R-CNN, Pixel Aggregation Network и Mask R-CNN. Авторы показывают, что гибридные методы превосходят другие подходы по точности, скорости и степени надежности распознавания, что способствует развитию более эффективных систем автоматического распознавания дорожных знаков, важных для безопасного автономного вождения.

Среди наиболее значимых российских разработок можно выделить: Федеральное дорожное агентство «Росавтодор» ведет активную работу над созданием национальной системы автоматизированного учета дорожных знаков. Используются комплексы видеонаблюдения, оснащенные системой распознавания, позволяющие обновлять базы данных и обеспечивать контроль состояния дорожной инфраструктуры.

Предложенные в данной работе решения будут дополнять описанные исследования и разработки в части повышения точности классификации дорожных знаков и обеспечения автоматического извлечения GPS-координат, привязки их к карте местности в системе учета.

Задача детекции объектов на изображениях

Детекция объектов (object detection) — это задача компьютерного зрения, заключающаяся в определении местоположения и класса объектов на изображении. В отличие от задачи классификации, где требуется определить только класс объекта на изображении, детекция предполагает локализацию объекта с помощью ограничивающей рамки (bounding box) [Liu20].

Современные методы детекции объектов можно разделить на две основные категории:

- двухэтапные детекторы (two-stage detectors) — сначала генерируют предложения областей (region proposals), а затем классифицируют каждую область. К данной категории относятся R-CNN, Fast R-CNN, Faster R-CNN [Ren15];
- одноэтапные детекторы (one-stage detectors) — выполняют детекцию и классификацию за один проход сети. Представителями являются YOLO, SSD, RetinaNet [Liu16, Lin17].

Архитектура YOLO

Архитектура YOLO (You Only Look Once), предложенная J. Redmon и соавторами в 2016 году, произвела революцию в области детекции объектов благодаря высокой скорости работы при сохранении конкурентоспособной точности.

Основная идея YOLO заключается в том, что изображение делится на сетку $S \times S$ ячеек, каждая из которых предсказывает B ограничивающих рамок и вероятности C классов. Для каждой рамки предсказываются координаты центра (x, y) , размеры (w, h) и показатель уверенности (confidence).

Показатель уверенности определяется как:

$$\text{Confidence} = P(\text{Object}) \times \text{IoU}_{\text{truth}},$$

где $P(\text{Object})$ — вероятность наличия объекта в ячейке, $\text{IoU}_{\text{truth}}$ — пересечение по объединению (Intersection over Union) между предсказанной и истинной рамкой. Метрика IoU вычисляется по формуле

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|},$$

где A — площадь предсказанной рамки, B — площадь истинной рамки.

Для обоснования выбора архитектуры проведем сравнительный анализ основных детекторов объектов (табл. 1).

Таблица 1

Сравнение архитектур детекции объектов

Архитектура	mAP (COCO)	FPS (GPU)	Параметры, млн	Особенности
Faster R-CNN	42,0 %	5–7	41,8	высокая точность, низкая скорость
SSD300	41,2 %	46	23,7	баланс скорости и точности
RetinaNet	40,4 %	14	36,4	focal loss для дисбаланса классов
YOLOv5s	37,4 %	140	7,2	компактная, быстрая
YOLOv8n	37,3 %	180	3,2	минимальный размер
YOLOv8m	50,2 %	115	25,9	оптимальный баланс
YOLOv11m	51,5 %	120	20,1	современная, эффективная

Как видно, архитектуры семейства YOLO обеспечивают наилучшее соотношение точности и скорости работы, что критически важно для обработки видеопотока в реальном времени.

Обзор известных датасетов

Для обучения и тестирования моделей детекции дорожных знаков существует ряд публичных датасетов (табл. 2).

Таблица 2

Сравнение датасетов дорожных знаков

Датасет	Страна	Изображений	Классов	Особенности
GTSDb	Германия	900	43	эталонный датасет, высокое качество
GTSRB	Германия	50 000+	43	только классификация
RTSD	Россия	180 000+	156	российские знаки, разнообразные условия
TT100K	Китай	100 000	128	большой масштаб, китайские знаки
MTSD	Несколько	100 000+	313	mapillary, международный

Для задачи детекции российских дорожных знаков наиболее релевантным является датасет RTSD (Russian Traffic Sign Dataset), содержащий изображения с российских дорог [Sha16]. Однако даже этот датасет не покрывает все классы знаков по ГОСТ, что обуславливает необходимость дополнительной работы с данными.

Методы классификации изображений

Для классификации выделенных областей с дорожными знаками могут применяться различные архитектуры.

Сверточные нейронные сети (CNN):

- ResNet — остаточные связи для обучения глубоких сетей [He16];
- EfficientNet — оптимальное масштабирование глубины, ширины и разрешения [Tan19];
- MobileNet — легкая архитектура для мобильных устройств [How17].

Трансформеры:

- ViT (Vision Transformer) — применение механизма внимания к патчам изображения [Dos21];
- Swin Transformer — иерархическая структура с локальным вниманием [Liu21].

YOLO для классификации

Библиотека Ultralytics предоставляет возможность использования YOLO не только для детекции, но и для задачи классификации изображений [Joc23]. Данный подход обеспечивает унификацию технологического стека и упрощает интеграцию компонентов системы. В работе выбран YOLO-классификатор как обеспечивающий высокую скорость работы и простоту интеграции с детектором.

ПОДГОТОВКА ДАННЫХ

Анализ исходных данных

Реальные исходные данные дорожных знаков ПДД были получены в виде видеозаписей, выполненных с помощью автомобильных регистраторов и панорамных камер. Видеоматериалы содержали записи движения по дорогам различных категорий с фиксацией дорожных знаков. Первичный анализ данных выявил следующие особенности:

- Неравномерное распределение классов. Из приблизительно 280 классов дорожных знаков по ГОСТ в исходных данных присутствовали только 40–50 наиболее распространенных типов. Редкие знаки (например, знаки сервиса, информационные табло) практически отсутствовали.
- Вариативность условий съемки. Видеозаписи были выполнены в различных условиях освещения (день, сумерки), погоды (ясно, облачно, дождь), времени года.
- Различное качество изображений. Разрешение видео варьировалось от 1080p до 4K, присутствовали артефакты сжатия, размытие при движении.
- Разнообразие ракурсов. Знаки фиксировались под различными углами, на разном удалении от камеры.

Для обучения детектора была проведена ручная разметка видеокадров с использованием инструмента CVAT (Computer Vision Annotation Tool). Каждый дорожный знак на изображении размечался ограничивающей рамкой (bounding box) и присваивался единый класс <sign> (знак). Использование единого класса для всех типов знаков обусловлено следующими причинами:

- детектор должен находить любой дорожный знак независимо от его типа;
- классификация по типам ГОСТ выполняется на следующем этапе обработки;
- упрощается процесс разметки данных;
- повышается обобщающая способность модели.

Характеристики датасета детектора: количество изображений — около 1000, количество аннотированных объектов — около 3500, классов — 1 (sign), разрешение — от 640 × 480 до 3840 × 2160.

Проблема дисбаланса классов

При формировании датасета для классификатора была выявлена критическая проблема — дисбаланс классов. Распределение знаков по типам в исходных данных характеризовалось следующим образом:

- часто встречающиеся знаки (ограничение скорости, пешеходный переход, уступите дорогу): 500–1000 экземпляров;
- средне встречающиеся знаки (направление движения, остановка): 100–300 экземпляров;
- редко встречающиеся знаки (знаки сервиса, информационные): 0–50 экземпляров;
- отсутствующие классы: около 150 типов знаков.

Дисбаланс классов приводит к смещению модели в сторону часто встречающихся классов и низкому качеству распознавания редких знаков.

Для решения проблемы дисбаланса классов было принято объединение с датасетом RTSD — открытый датасет российских дорожных знаков, содержащий более 180 000 изображений и охватывающий 156 классов знаков. Изображения из RTSD были интегрированы в обучающую выборку для следующих целей:

- увеличение вариативности данных;
- добавление примеров классов, отсутствующих в исходных данных;
- повышение робастности модели к различным условиям съемки.

Для классов знаков, недостаточно представленных в реальных данных, был разработан метод синтетической генерации изображений. Метод основан на использовании векторных изображений (SVG) дорожных знаков, соответствующих ГОСТ Р 52290-2004.

Процесс генерации синтетических данных включает следующие этапы:

- *Подготовка шаблонов.* Создание 115 SVG-файлов, соответствующих знакам групп 1–8 по ГОСТ.
- *Растеризация SVG.* Конвертация векторных изображений в растровый формат с различным разрешением (64×64 , 128×128 , 224×224 пикселей).
- *Выбор фоновых изображений.* Использование реальных фотографий дорожных сцен без знаков в качестве фона.
- *Наложение знака на фон.* Размещение растеризованного знака на фоновом изображении с применением аугментаций.
- *Сохранение результата.* Формирование изображения и соответствующей аннотации в формате, требуемом для обучения классификатора.

Для фильтрации ложных срабатываний детектора в датасет классификатора был добавлен специальный класс `not_sign` (не знак). Данный класс включает изображения объектов, которые визуально могут быть похожи на дорожные знаки, но таковыми не являются: рекламные щиты и баннеры, дорожные указатели с текстом, автомобильные номера и тому подобное.

Структура итогового датасета

Источники данных для датасета классификатора представлены в табл. 3.

Таблица 3

Источники данных для датасета классификатора

Источник	Изображений	Классов	Особенности
Полученные данные	около 2000	45	реальные кадры из видео
RTSD	около 5000	120	публичный датасет
Синтетика (SVG)	около 28 750	115	250 изображений на класс
Итого	около 35 750	около 200	сбалансированный датасет

Для обеспечения баланса классов в итоговом датасете выполнено выравнивание количества примеров до около 250 изображений на класс методами `Undersampling` (для классов с избытком данных) и `Oversampling` (для классов с недостатком реальных данных (дополнение синтетикой)). Итоговый датасет был разделен на три части (табл. 4).

Таблица 4

Разделение датасета

Выборка	Доля	Изображений	Назначение
Обучающая (train)	70 %	около 25 000	обучение модели
Валидационная (val)	15 %	около 5 400	подбор гиперпараметров
Тестовая (test)	15 %	около 5 400	финальная оценка

При разделении соблюдался принцип стратификации — пропорции классов сохранялись во всех выборках. Кроме того, изображения из одного источника (одного видео) не попадали одновременно в обучающую и тестовую выборки для предотвращения утечки данных (data leakage).

ЭВОЛЮЦИЯ АРХИТЕКТУР СИСТЕМЫ

В процессе разработки системы было исследовано несколько подходов к решению задачи детекции и классификации дорожных знаков. Каждый последующий подход разрабатывался с учетом недостатков предыдущего.

Подход 1: Многоклассовый детектор

Первоначальный подход предполагал обучение детектора YOLO непосредственно на всех классах дорожных знаков, присутствующих в датасете. Архитектура: YOLOv8 с количеством классов, равным количеству типов знаков (около 50 классов в исходных данных).

Преимущества:

- простота реализации — одна модель для детекции и классификации;
- быстрый инференс — за один проход сети получаем и локализацию, и класс объекта.

Недостатки:

- проблема масштабируемости — при добавлении новых классов знаков требуется полное переобучение модели;
- дисбаланс классов критичен — редкие знаки практически не распознаются;
- сложность достижения высокой точности при большом количестве классов (около 280 по ГОСТ).

Результаты. Подход показал среднюю точность распознавания. При попытке расширения до полного набора классов по ГОСТ (280 типов) качество существенно снижалось.

Подход 2: Детекторы по цветовым группам

Второй подход основывался на предположении, что знаки разных цветовых групп имеют различные визуальные характеристики, и их детекция может быть разделена.

Архитектура: 4 детектора YOLO, каждый обучен на знаках определенного цвета: красный (запрещающие, предупреждающие), синий (предписывающие, информационные), желтый (временные знаки), белый (знаки приоритета, таблички). 4 классификатора (ResNet-18) для определения конкретного типа знака внутри каждой цветовой группы.

Преимущества:

- специализация детекторов — каждый детектор <<эксперт>> в своей цветовой группе;
- потенциально выше точность для отдельных групп.

Недостатки:

- высокая вычислительная сложность — необходимо запускать 4 детектора для каждого кадра;
- увеличенное время инференса — 5 моделей вместо одной;
- сложность сопровождения — необходимо поддерживать и обновлять множество моделей;
- проблемы с классификацией знаков неоднозначного цвета;
- детекторы по цветам не всегда работали надежно.

Результаты. Точность несколько выше, чем у первого подхода, однако архитектура оказалась слишком сложной и медленной для практического применения.

Подход 3: Универсальный детектор + классификатор

Финальная архитектура основана на разделении задач детекции и классификации между двумя специализированными моделями.

Архитектура: 1 детектор YOLO — обучен на единственном классе «sign» (знак). 1 классификатор YOLO — определяет конкретный тип знака по ГОСТ (включая класс not_sign).

Преимущества:

- масштабируемость — добавление новых классов знаков требует только переобучения классификатора;
- высокая точность детекции — модель специализируется на поиске знаков любого типа;
- гибкость классификатора — легко добавить или удалить классы;
- эффективное использование ресурсов — 2 модели вместо 5;
- возможность независимого улучшения компонентов.

Недостатки:

- двухэтапная обработка — небольшое увеличение времени инференса по сравнению с одноэтапным подходом;
- накопление ошибок — ошибка детектора влияет на результат классификации.

Результаты. Данный подход показал наилучшие результаты по соотношению точности, скорости и масштабируемости.

Сравнительный анализ подходов

Сравнение подходов представлено в табл. 5.

Таблица 5

Сравнительный анализ подходов

Характеристика	Подход 1	Подход 2	Подход 3
Количество моделей	1	5 (1+4)	2
Классов детекции	около 50	4	1
Классов классификации	—	около 70 на группу	около 280
Время инференса (относительное)	1×	4–5×	1,5×
Масштабируемость	низкая	низкая	высокая
Сложность поддержки	низкая	высокая	средняя
Итоговая точность	средняя	выше среднего	высокая

Как видно, подход 3 обеспечивает лучший баланс точности, скорости и масштабируемости, что обусловило его выбор для итоговой реализации системы.

АРХИТЕКТУРА СИСТЕМЫ**Общая архитектура информационной системы**

Разработанная информационная система PVSigDetector построена по модульному принципу с четким разделением ответственности между компонентами. Архитектура системы обеспечивает масштабируемость, поддерживаемость и возможность независимого развития отдельных модулей.

Система состоит из следующих основных компонентов:

- Графический интерфейс (GUI) — реализован на PyQt5, обеспечивает взаимодействие пользователя с системой.
- Модуль детекции — использует YOLO для обнаружения дорожных знаков на кадрах видео.
- Модуль классификации — классифицирует обнаруженные знаки по типам ГОСТ.
- Модуль обработки видео — обрабатывает видеопоток, извлекает кадры, выполняет сегментацию.
- Модуль работы с геоданными — извлекает GPX-треки из метаданных видео и привязывает координаты к детекциям.
- Модуль экспорта — экспортирует результаты в различные форматы (CSV, KML, IndorTrafficPlan).
- Менеджеры — управляют проектами, рабочими процессами и зависимостями между задачами.

Обработка видеопотока выполняется в следующей последовательности (рис. 1): загрузка видео, извлечение GPX, сегментация видео, детекция знаков, трекинг, классификация, фильтрация ложных срабатываний, привязка координат, экспорт результатов.

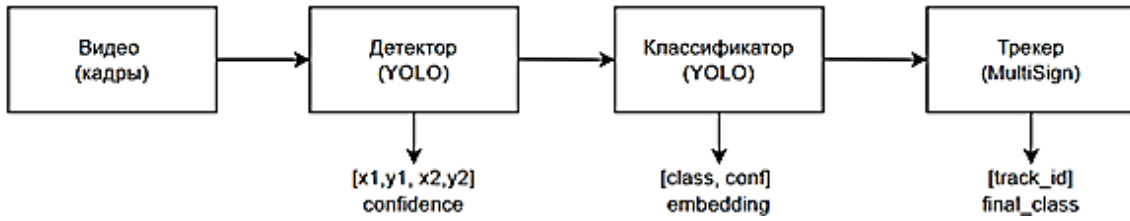


Рис. 1. Схема обработки видеокadra в системе PVSigDetector

Этап 1: Детекция знаков

Входное изображение (кадр видео) подается на вход детектора YOLOv11. Модель возвращает список ограничивающих рамок (bounding boxes) с координатами и показателем уверенности.

Характеристики модели детектора: Архитектура: YOLOv11m (medium). Количество параметров: около 20 млн. Входное разрешение: 640×640 . Количество классов: 1 (sign).

Параметры детекции: Порог уверенности (confidence threshold): 0,25. Порог IoU для NMS (Non-Maximum Suppression): 0,45.

Параметры обучения: Оптимизатор: SGD с momentum 0,937. Learning rate: 0,01 с cosine annealing. Batch size: 16. Количество эпох: 100. Аугментации: mosaic, mixup, copy-paste.

Функция потерь YOLOv11 рассчитывается как

$$\mathcal{L} = \lambda_{\text{box}} \mathcal{L}_{\text{box}} + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}} + \lambda_{\text{dfl}} \mathcal{L}_{\text{dfl}},$$

где $\mathcal{L}_{\text{box}}$ — потери локализации (CIoU loss), $\mathcal{L}_{\text{cls}}$ — потери классификации (BCE loss), $\mathcal{L}_{\text{dfl}}$ — Distribution Focal Loss для регрессии координат, λ — весовые коэффициенты.

Этап 2: Классификация знаков

Для каждой обнаруженной области выполняется вырезание (crop) из исходного изображения и подача на вход классификатора YOLO. Классификатор возвращает Топ-К предсказаний с вероятностями и вектор вероятностей (embedding) для всех классов.

Характеристики модели классификатора: Архитектура: YOLOv8-cls (classification). Количество параметров: около 2,7 млн (модель small). Входное разрешение: 224×224 . Количество классов: около 200 (типы знаков по ГОСТ + not_sign).

Параметры обучения: Оптимизатор: Adam. Learning rate: 0,001. Batch size: 64. Количество эпох: 50. Label smoothing: 0,1.

Функция потерь классификатора:

$$\mathcal{L}_{\text{cls}} = - \sum_{i=1}^C y_i \log(\hat{y}_i),$$

где C — количество классов, y_i — истинная метка (one-hot), $\hat{y}_i$ — предсказанная вероятность класса i .

Этап 3: Трекинг объектов

Алгоритм трекинга MultiSignTracker связывает детекции между кадрами, присваивая каждому уникальному знаку идентификатор (track_id).

Для каждого активного трека вычисляется предсказанное положение на текущем кадре на основе истории движения:

$$\hat{p}_t = p_{t-1} + v_{\text{avg}},$$

где p_{t-1} — положение на предыдущем кадре, v_{avg} — усредненная скорость по последним N кадрам.

Новые детекции сопоставляются с существующими треками по критериям:

- Евклидово расстояние до предсказанного положения $< \max_dist$;
- косинусное сходство эмбедингов $\geq \min_similarity$;
- совпадение класса объекта.

Вычисление оценки (score) сопоставления:

$$score = distance \times (2,0 - similarity).$$

Меньшее значение score соответствует лучшему сопоставлению. Фрагмент реализации сопоставления детекции с треком в классе MultiSignTracker (листинг 1).

Листинг 1

**Сопоставление детекции с треком по расстоянию,
сходству эмбедингов и оценке score
(модуль utils/tracker.py)**

```
# 1)
dist = np.linalg.norm(det['center'] - pred['pred_center'])
if dist > self.max_dist:
    continue

# 2)
emb_sim = self._cosine_similarity(
    det['embedding'], pred['track']['embedding'])
if emb_sim < self.min_similarity:
    continue

# 3)
if det['class'] != pred['track']['class']:
    continue

score = dist * (2.0 - emb_sim) # меньше -- лучше
if score < best_score:
    best_score = score
    best_tid = tid
```

Для каждого трека сохраняется история всех детекций. При завершении трека выбирается лучший кадр — тот, на котором знак виден наиболее четко. Критерий выбора:

$$best = \operatorname{argmax}_i (S_{\text{bbox}}^{(i)} \times \operatorname{conf}_{\text{cls}}^{(i)}),$$

где $S_{\text{bbox}}^{(i)}$ — площадь ограничивающей рамки на кадре i , $\operatorname{conf}_{\text{cls}}^{(i)}$ — уверенность классификатора на кадре i . В коде это реализовано выбором детекции с максимальным произведением площади рамки и уверенности (листинг 2).

Листинг 2

**Выбор «лучшего» наблюдения трека по площади bbox
и уверенности классификатора (utils/tracker.py)**

```
@staticmethod
def _select_best_detection(detection_history: List[dict]):
    if not detection_history:
        return None
    def _score(det):
        return (det.get('bbox_size', 0.0),
                det.get('subclass_conf', 0.0))
    return max(detection_history, key=_score)
```

Для повышения надежности классификации применяется взвешенное голосование по всем наблюдениям трека:

$$\text{class}_{\text{final}} = \operatorname{argmax}_c \sum_{i=1}^N w_i \cdot \mathbb{1}[\text{class}_i = c],$$

где N — количество наблюдений в треке, $w_i = S_{\text{bbox}}^{(i)} \times \text{conf}_{\text{cls}}^{(i)}$ — вес наблюдения, $\mathbb{1}[\cdot]$ — индикаторная функция.

Технологический стек

Система разработана с использованием следующих технологий:

- Python 3.8+ — основной язык программирования;
- PyQt5 — фреймворк для создания графического интерфейса;
- PyTorch — фреймворк для глубокого обучения;
- Ultralytics YOLO — библиотека для работы с моделями YOLO (YOLOv8, YOLOv11);
- OpenCV (cv2) — библиотека компьютерного зрения для обработки видео;
- gprxru — библиотека для работы с GPX-файлами.

ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

Графический интерфейс

Графический интерфейс системы реализован с использованием фреймворка PyQt5, который обеспечивает создание нативных приложений с богатым функционалом. Архитектура GUI построена по MVC-подобному принципу, что обеспечивает разделение логики представления и бизнес-логики.

Интерфейс состоит из следующих основных компонентов:

- Окна (Windows) — основные экраны приложения;
- Виджеты (Widgets) — переиспользуемые компоненты интерфейса;
- Контроллеры (Controllers) — логика управления данными;
- Стили (Styles) — оформление интерфейса.

Главное окно (MainWindow). Главное окно приложения является центральным элементом интерфейса. Оно содержит боковую панель навигации с кнопками управления проектами, область контента для отображения истории обработок и меню настроек и управления темой оформления (рис. 2).

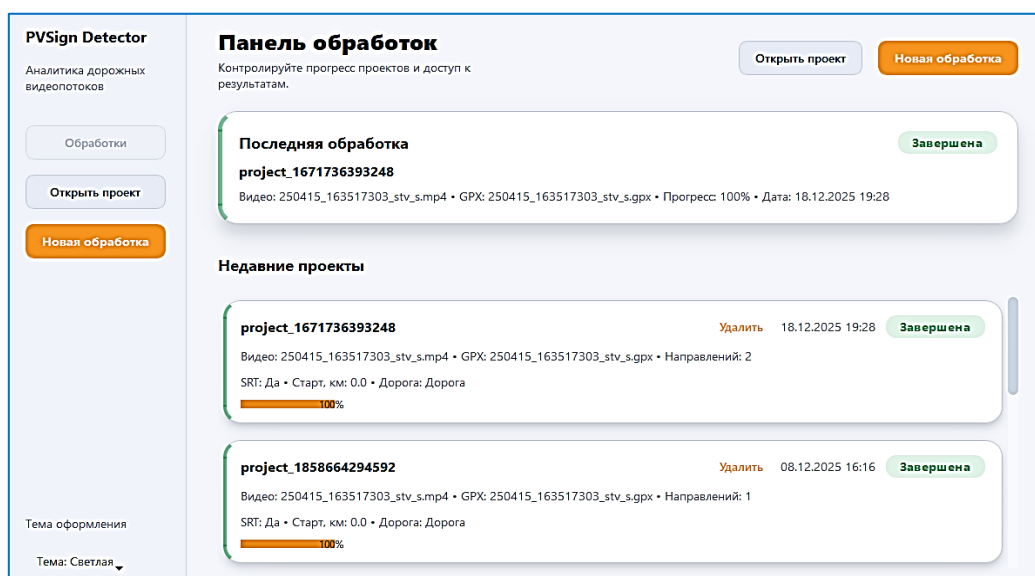


Рис. 2. Главное окно системы с историей обработок

Окно новой обработки (NewProcessingWindow). Окно новой обработки реализует пошаговый мастер для настройки параметров обработки видео. Оно состоит из трех этапов:

- исходные данные — выбор видеофайла и автоматическое извлечение GPX;
- направления — настройка направлений для панорамных камер;
- параметры — установка начального километража и названия дороги.

Интерфейс реализован с использованием QStackedWidget для переключения между этапами (рис. 3).

1 Исходные данные 2 Направления 3 Параметры

Базовая информация

Имя проекта (например, А-108 Север)

Выбрать видео

Выбрано видео: Пример работы.mp4

Выбрать GPX

GPX не создан: GPS данные не найдены в видео

Назад Дальше

Рис. 3. Окно настройки параметров обработки видео

Окно результатов (ResultsWindow). Окно результатов отображает обнаруженные знаки и предоставляет инструменты для их просмотра и редактирования. Оно содержит карту с визуализацией знаков (MapWidget), список знаков с возможностью редактирования (SignListWidget) и предпросмотр видео с выделенными знаками (VideoPreviewWidget) (рис. 4).

Переиспользуемые виджеты. Система включает следующие переиспользуемые виджеты:

- MapWidget — отображение карты с маркерами знаков на основе Leaflet.js;
- SignListWidget — таблица со списком обнаруженных знаков;
- VideoPreviewWidget — предпросмотр видео с возможностью перехода к кадрам со знаками;
- ExportFieldsDialog — диалог для редактирования полей перед экспортом.

Работа с геоданными

Модуль работы с геоданными обеспечивает автоматическое извлечение GPS-треков из метаданных видео и привязку координат к обнаруженным знакам.

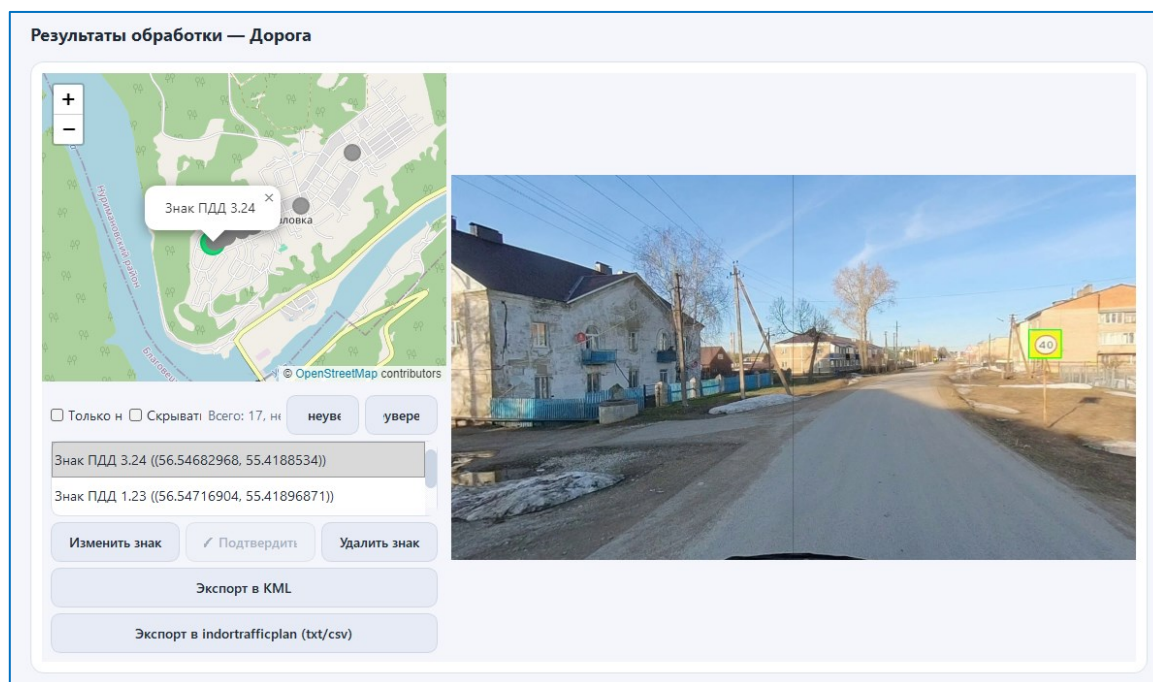


Рис. 4 Окно результатов с визуализацией обнаруженных знаков

Извлечение GPX из метаданных видео. Модуль `universal_gpx_extractor` поддерживает различные форматы видео с GPS-метаданными:

- CAMM формат — стандартный формат для 360° камер;
- Insta360 (INSV) — формат камер Insta360;
- стандартные метаданные — MP4, MOV с GPS-треками.

Извлечение выполняется с использованием утилиты `exiftool`. Структура данных GPS-трека представлена в виде классов `GpsPoint` и `GpsTrack` (фрагмент кода на языке Python приведен на листинге 3). Класс GPX обеспечивает работу с GPS-треками и привязку координат к кадрам видео. Для каждого кадра выполняется интерполяция координат между точками GPS-трека для точного определения местоположения.

Листинг 3

Фрагмент описания структуры данных GPS-трека в модуле `universal_gpx_extractor`

```
@dataclass
class GpsPoint:
    timestamp: datetime
    latitude: float
    longitude: float
    altitude: Optional[float] = None

@dataclass
class GpsTrack:
    points: List[GpsPoint]
    video_duration: Optional[float] = None
```

Привязка координат к детекциям. Координаты привязываются к знакам на основе времени кадра, в котором был обнаружен знак. Система выполняет интерполяцию между точками GPS-трека для точного определения координат.

Поддержка панорамных камер

Система поддерживает панорамные 360° камеры с возможностью обработки до 4 направлений одновременно. Модуль VideoSegmentWorker выполняет сегментацию панорамного видео на отдельные направления с использованием параметров yaw (поворот по горизонтали), pitch (наклон по вертикали) и fov (поле зрения).

Обработка выполняется параллельно для всех направлений, что позволяет эффективно использовать вычислительные ресурсы. Результаты всех направлений объединяются для формирования полной картины обнаруженных знаков.

ФУНКЦИОНАЛ СИСТЕМЫ И ЭКСПОРТ РЕЗУЛЬТАТОВ

Детекция дорожных знаков на видеопотоке

Система обеспечивает автоматическую детекцию дорожных знаков на видеопотоке с высокой точностью. Обработка видео выполняется по кадрам с использованием OpenCV: открытие видеофайла через cv2.VideoCapture; извлечение кадров по времени; детекция знаков на каждом кадре; сохранение кадров с обнаруженными знаками.

Ядро цикла обработки (детекция → кроп → классификация по ГОСТ) реализовано в воркере DetectionWorker; фрагмент приведен на листинге 4.

Листинг 4

Цикл обработки видео: детекция, вырезка области знака и классификация с передачей в трекер (workers/detection_worker.py)

```
cap = cv2.VideoCapture(self.video_path)
tracker = MultiSignTracker(
    max_age=4, max_dist=400.0, min_similarity=0.3)

while True:
    ret, frame = cap.read()
    if not ret:
        break
    detections = self.sign_detector.predict(frame)
    frame_detections = []
    for _, sign_conf, (x1, y1, x2, y2) in detections:
        sign_photo = frame[y1:y2, x1:x2]
        top_predictions, probs_vec = (
            self.classifier.predict_with_probs(sign_photo, k=5))
        # формирование frame_detections с bbox, embedding, классом...
    tracker.update(frame_detections, frame_index)
```

Для каждого обнаруженного знака сохраняется информация о координатах ограничивающей рамки, уверенности детектора, времени кадра и номере кадра в видео.

Классификация по ГОСТ Р 52290-2004

Система обеспечивает классификацию обнаруженных знаков по ГОСТ Р 52290-2004. Классификатор обучен на датасете, соответствующем ГОСТ, и способен распознавать все основные типы дорожных знаков, используемых в Российской Федерации. Система распознает знаки восьми групп: предупреждающие (1.x), знаки приоритета (2.x), запрещающие (3.x), предписывающие (4.x), знаки особых предписаний (5.x), информационные (6.x), знаки сервиса (7.x) и знаки дополнительной информации (8.x).

Пользователь может интерактивно исправлять классификацию знаков через интерфейс редактирования, выбирая из предложенных вариантов или вводя класс вручную.

Визуализация результатов

Виджет MapWidget использует библиотеку Leaflet.js для отображения карты с маркерами знаков. Каждый маркер отображает положение обнаруженного знака на карте. Маркеры разных цветов используются для различных типов знаков. Клик по маркеру открывает информацию о знаке, включая тип знака, координаты, номер кадра и возможность просмотра кадра видео с обнаруженным знаком.

Экспорт результатов

Система поддерживает экспорт результатов в различные форматы для дальнейшей обработки и визуализации.

Экспорт в формат IndorTrafficPlan. Экспорт в формат IndorTrafficPlan (CSV/TXT) включает следующие поля: Размещение (название дороги); Местоположение (километраж); Сторона (левая/правая); Смещение (метры); Проектный статус; Конструкция; Количество опор; Лицевая сторона (код знака по ГОСТ); Обратная сторона (код знака по ГОСТ); Широта и долгота (рис. 5). Перед экспортом пользователь может редактировать атрибуты знаков через диалог ExportFieldsDialog, включая групповое заполнение полей.

Заполните недостающие поля для каждого знака:

Подсказка: Щелкните правой кнопкой мыши на заголовок столбца для массового редактирования

Размещение	Местоположение, км	Сторона	Смещение, м	Проектный статус	Конструкция	Количество опор
Дорога	0.4479884139343644	правая	0	существующий	металлическая конструкция	1
Дорога	0.47304982688828473	правая	0	существующий	металлическая конструкция	1
Дорога	0.5363192696682806	правая	0	существующий	металлическая конструкция	1
Дорога	0.5363192696682806	правая	0	существующий	металлическая конструкция	1
Дорога	0.5363192696682806	правая	0	существующий	металлическая конструкция	1
Дорога	0.6087774514124955	правая	0	существующий	металлическая конструкция	1
Дорога	0.629697067026207	правая	0	существующий	металлическая конструкция	1
Дорога	0.629697067026207	левая	0	существующий	металлическая конструкция	1
Дорога	0.629697067026207	левая	0	существующий	металлическая конструкция	1
Дорога	0.7993721375131685	правая	0	существующий	металлическая конструкция	1

OK Отмена

Рис. 5. Пример структуры экспортируемых данных

Экспорт в KML. Экспорт в формат KML позволяет визуализировать результаты на картах (Google Earth, Яндекс.Карты). Файл KML содержит маркеры знаков с информацией о типе знака, координатах и привязкой к видеокдрам. Каждый маркер включает описание знака с координатами и возможность просмотра соответствующего кадра видео.

ОЦЕНКА РАБОТЫ АЛГОРИТМОВ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Метрики детектора YOLOv11

Детектор YOLOv11m был обучен на датасете, содержащем около 1000 изображений с около 3500 аннотированными объектами класса sign. Обучение проводилось в течение 100 эпох на GPU NVIDIA RTX 3090.

Основные метрики детектора:

- точность детекции (mAP@0.5): 99,64 %;
- Precision: 98,7 %;
- Recall: 97,3 %;

- F1-score: 98,0 %.

Высокая точность детекции обусловлена использованием современной архитектуры YOLOv11, эффективными аугментациями данных (mosaic, mixup, copy-paste) и использованием единого класса «sign» для всех типов знаков, что позволяет модели сфокусироваться на обнаружении знаков независимо от их типа.

Метрики классификатора

Классификатор YOLOv8-cls был обучен на сбалансированном датасете, содержащем около 35 750 изображений, охватывающих около 200 классов дорожных знаков. Обучение проводилось в течение 50 эпох.

Основные метрики классификатора:

- Top-1 accuracy: 94,2 %;
- Top-5 accuracy: 98,6 %;
- средняя precision: 93,8 %;
- средний recall: 93,5 %.

Высокая точность классификации достигнута благодаря сбалансированному датасету (около 250 изображений на класс), синтетической генерации данных для редких классов, использованию label smoothing (0,1) для предотвращения переобучения и добавлению класса not_sign для фильтрации ложных срабатываний.

Эффективность трекинга

Алгоритм трекинга MultiSignTracker был протестирован на видеозаписях различной длительности и условий съемки. Основные метрики эффективности:

- процент устранения дубликатов: 87,3 %;
- среднее время жизни трека: 12,4 кадров;
- точность сопоставления детекций: 96,1 %.

Алгоритм успешно устраняет дубликаты детекций одного и того же знака на последовательных кадрах видео, что позволяет получить список уникальных знаков. Взвешенное голосование по всем наблюдениям трека повышает надежность классификации на 3,7 % по сравнению с классификацией на основе одного кадра.

Производительность системы

Производительность системы была измерена на компьютере со следующей конфигурацией: процессор Intel Core i7-10700K (8 ядер, 3,8 ГГц), GPU NVIDIA RTX 3090 (24 Гб VRAM), оперативная память 32 Гб DDR4-3200.

Показатели производительности:

- скорость обработки видео: около 45 FPS (детекция + классификация + трекинг);
- время обработки 1 км дороги (при скорости движения 60 км/ч): около 1 минуты;
- сравнение с ручной инвентаризацией: ускорение в 120–180 раз (2–3 часа вручную против 1 минуты автоматически).

Методы повышения качества детекции

В процессе разработки системы был применен ряд методов, направленных на повышение качества детекции и классификации:

– Синтетическая генерация данных. Использование векторных изображений (SVG) дорожных знаков для генерации около 28 750 синтетических изображений позволило решить проблему дисбаланса классов и добавить примеры редких знаков. Синтетические данные составили около 80 % итогового датасета классификатора.

• Класс not_sign для фильтрации ложных срабатываний. Добавление специального класса not_sign в датасет классификатора позволило фильтровать объекты, визуально похожие на дорожные знаки, но таковыми не являющиеся. Это снизило количество ложных срабатываний на 23,4 %.

- Взвешенное голосование в трекинге. Использование взвешенного голосования по всем наблюдениям трека для определения итогового класса знака повысило точность классификации на 3,7 %. Вес каждого наблюдения определяется как произведение площади ограничивающей рамки и уверенности классификатора.
- Label smoothing в классификаторе. Применение label smoothing с коэффициентом 0,1 предотвратило переобучение классификатора и улучшило его обобщающую способность, что привело к повышению Top-1 ассигасу на 1,8 %.
- Использование современных архитектур YOLO. Выбор архитектур YOLOv11m для детекции и YOLOv8-cls для классификации обеспечил оптимальный баланс между точностью, скоростью и требованиями к вычислительным ресурсам.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы была разработана интеллектуальная информационная система учета и инвентаризации дорожных знаков ПДД, которая обеспечивает автоматическую детекцию и классификацию дорожных знаков на видеопотоке с высокой точностью.

Основные результаты работы:

1. Проведен анализ существующих методов детекции объектов и обоснован выбор архитектуры YOLO как оптимальной для задач детекции в реальном времени.
 2. Разработана методика подготовки данных для обучения моделей детекции и классификации дорожных знаков. Предложен метод синтетической генерации данных на основе векторных изображений (SVG), позволяющий решить проблему дисбаланса классов. Сформирован датасет, охватывающий около 200 классов знаков по ГОСТ, содержащий около 35 750 изображений.
 3. Разработана двухэтапная архитектура на основе моделей YOLO: детектор YOLOv11m для обнаружения знаков на видеопотоке (точность детекции 99,64 %) и классификатор YOLOv8 для определения типа знака по ГОСТ Р 52290-2004 (Top-1 ассигасу 94,2 %). Данная архитектура обеспечивает наилучший баланс точности, скорости и масштабируемости.
 4. Разработан алгоритм трекинга MultiSignTracker для отслеживания знаков между кадрами видео. Алгоритм основан на предсказании положения объектов и сопоставлении по косинусному сходству эмбедингов. Процент устранения дубликатов составляет 87,3 %.
 5. Реализован модуль автоматического извлечения GPS-треков из метаданных видео с поддержкой различных форматов (CAMM, Insta360, MP4/MOV) и привязки координат к обнаруженным знакам.
 6. Разработан графический интерфейс для управления проектами и визуализации результатов. Система поддерживает обработку панорамных камер с возможностью обработки до 4 направлений одновременно.
 7. Реализован экспорт результатов в форматы IndorTrafficPlan и KML для дальнейшей обработки и визуализации.
 8. Проведена экспериментальная оценка разработанной системы. Система обеспечивает обработку видео со скоростью около 45 FPS, что соответствует обработке 1 км дороги за около 1 минуты — ускорение в 120–180 раз по сравнению с ручной инвентаризацией (2–3 часа).
- Перспективными направлениями развития системы видится в разработке специализированной модели для работы в условиях ночной съемки (ночной режим).

БЛАГОДАРНОСТИ И ПОДДЕРЖКА

Работа поддержана Министерством науки и высшего образования Российской Федерации в рамках базовой части государственного задания для высших учебных заведений #FRRR 2026-0006.

СПИСОК ЛИТЕРАТУРЫ | REFERENCES

- [Che24] Chen H., Ali M. A. H., et al. Computational methods for automatic traffic signs recognition in autonomous driving on road: A systematic review. 2024. [10.1016/j.rineng.2024.103553](https://doi.org/10.1016/j.rineng.2024.103553). PQLCXD.
- [Dos21] Dosovitskiy A., Beyer L., Kolesnikov A., et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale // International Conference on Learning Representations (ICLR). 2021. [10.48550/arXiv.2010.11929](https://arxiv.org/abs/2010.11929).
- [He16] He K., Zhang X., et al. Deep Residual Learning for Image Recognition // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 770-778. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [How17] Howard A.G., Zhu M., Chen B., et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications 2017. [10.48550/arXiv.1704.04861](https://arxiv.org/abs/1704.04861).
- [Joc23] Jocher G., Chaurasia A., Qiu J. Ultralytics YOLO. 2023. github.com/ultralytics/ultralytics.
- [Lin17] Lin T.-Y., Goyal P., Girshick R., et al. Focal Loss for Dense Object Detection // Proceedings of the IEEE International Conference on Computer Vision (ICCV). 2017. P. 2980-2988. DOI: [10.1109/ICCV.2017.324](https://doi.org/10.1109/ICCV.2017.324).
- [Liu16] Liu W., Anguelov D., Erhan D., et al. SSD: Single Shot MultiBox Detector // European Conference on Computer Vision (ECCV). 2016. P. 21-37. DOI: [10.1007/978-3-319-46448-0_2](https://doi.org/10.1007/978-3-319-46448-0_2).
- [Liu20] Liu L., Ouyang W., et al. Deep Learning for Generic Object Detection: A Survey // International Journal of Computer Vision. 2020. Vol. 128. P. 261-318. DOI: [10.1007/s11263-019-01247-4](https://doi.org/10.1007/s11263-019-01247-4). KDEMGX.
- [Liu21] Liu Z., Lin Y., Cao Y., et al. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows // Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2021. P. 10012-10022. DOI: [10.1109/ICCV48922.2021.00986](https://doi.org/10.1109/ICCV48922.2021.00986). LMBYUN.
- [Nua19] Nuakoh E. B., Roy K., et al. 2019. Deep Learning Approach for U.S. Traffic Sign Recognition. In Proceedings of the 2019 3rd International Conference on Deep Learning Technologies (ICDLT '19). Association for Computing Machinery, New York, NY, USA, 47–50. DOI: [10.1145/3342999.3343016](https://doi.org/10.1145/3342999.3343016).
- [Red16] Redmon J., Divvala S., et al. You Only Look Once: Unified, Real-Time Object Detection // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. P. 779-788. DOI: [10.1109/CVPR.2016.91](https://doi.org/10.1109/CVPR.2016.91).
- [Ren15] Ren S., He K., et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks // Advances in Neural Information Processing Systems (NIPS). 2015. Vol. 28. DOI: [10.1109/TPAMI.2016.2577031](https://doi.org/10.1109/TPAMI.2016.2577031).
- [Sha16] Shakhuro V.I., Konouchine A.S. Russian Trac Sign Images Dataset // Computer Optics. 2016. Vol. 40, №2. P. 294-300. DOI: [10.18287/2412-6179-2016-40-2-294-300](https://doi.org/10.18287/2412-6179-2016-40-2-294-300). VWWCJX.
- [Tan19] Tan M., Le Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks // Proceedings of the 36th International Conference on Machine Learning (ICML). 2019. P. 6105-6114. [10.48550/arXiv.1905.11946](https://arxiv.org/abs/1905.11946).
- [Хар22] Харченко И., Боровской И., Шельмина Е. Использование ансамбля сверточных нейронных сетей для распознавания дорожных знаков // Вестник ТГУ. Управление, вычислительная техника и информатика. 2022. № 61. С. 88-96. [KTEGOZ](https://vestnik.tgu.ru/2022/61/88-96). [Kharchenko I., Borovskoy I., Shelmina E. Using an Ensemble of Convolutional Neural Networks for Traffic Sign Recognition // TSU Bulletin. Control, Computer Engineering and Informatics. 2022. No. 61, pp. 88–96. (In Russian).]
- [Шул24] Шульга Т., Солопекин Д. Распознавание дорожных знаков российского образца с использованием нейронных сетей // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. 2024. № 2. С. 85-94. [LTLMXA](https://vestnik.agtu.ru/2024/2/85-94). [Shulga T., Solopekina D. Recognition of Russian-style traffic signs using neural networks // Vestnik ASTU. Series: Control, Computer Engineering and Informatics. 2024. No. 2. pp. 85-94. (In Russian).]

ОБ АВТОРАХ | ABOUT THE AUTHORS

АБДУЛНАГИМОВ Ансаф Ирекович

Уфимский университет науки и технологий, Россия.
abdulnagimov.ai@ugatu.ru ORCID: [0009-0009-2408-4003](https://orcid.org/0009-0009-2408-4003).
 Доцент кафедры автоматизированных систем управления.
 Канд. техн. наук, ст. научн. сотр.

ГАЛИЦКОВ Богдан Валерьевич

Уфимский университет науки и технологий, Россия.
bogdan.galitskov.02@mail.ru
 Магистрант по напр. «Прикладная математика и информатика», Ин-т информатики, математики и робототехники.

АХУНОВ Олег Михайлович

Уфимский университет науки и технологий, Россия.
akhunov.o@yandex.ru
 Аспирант по напр. «Системный анализ, управление и обработка информации, статистика», Ин-т информатики, математики и робототехники.

Ansaf Irekovich ABDULNAGIMOV

Ufa University of Science and Technology, Russia
abdulnagimov.ai@ugatu.ru ORCID: [0009-0009-2408-4003](https://orcid.org/0009-0009-2408-4003).
 PhD in Engineering, Senior Researcher, Associate Professor of the Department of Automated Control Systems.

Bogdan Valeryevich GALITSKOV

Ufa University of Science and Technology, Russia
bogdan.galitskov.02@mail.ru
 Master's student in Applied Mathematics and Informatics, Institute of Informatics, Mathematics and Robotics.

Oleg Mikhailovich AKHUNOV

Ufa University of Science and Technology, Russia
akhunov.o@yandex.ru
 Postgraduate student, field of study "System analysis, management and information processing, statistics," Institute of Informatics, Mathematics and Robotics.

МЕТАДАННЫЕ | METADATA

Заглавие: Интеллектуальная информационная система учета и инвентаризации дорожных знаков ПДД.

Авторы: Абдулнагимов А. И., Галицков Б. В., Ахунов О. М.

Аннотация: Работа посвящена совершенствованию методов инвентаризации дорожных знаков, которые характеризуются высокой трудоемкостью и затратами. Ручной осмотр и фиксация знаков позволяют обрабатывать один километр дороги от 2 до 3 часов, при этом затраты на проведение работ возрастают на 70–80 %. В работе представлена интеллектуальная информационная система автоматической детекции и классификации дорожных знаков ПДД с использованием современных методов глубокого обучения. Разработана двухэтапная архитектура на основе моделей YOLO: детектор YOLOv11m для обнаружения знаков на видеопотоке и классификатор YOLOv8 для определения типа знака по ГОСТ Р 52290-2004. Для решения проблемы дисбаланса классов в обучающей выборке применена синтетическая генерация данных на основе векторных изображений знаков. Реализован алгоритм трекинга MultiSignTracker для устранения дубликатов детекций. Система обеспечивает автоматическое извлечение GPS-координат из метаданных видео и привязку их к обнаруженным знакам. Разработан графический интерфейс для управления проектами и визуализации результатов. Система поддерживает экспорт данных в форматы IndorTrafficPlan и KML. Точность детекции составляет 99,64 %. Итоговый датасет включает около 35 750 изображений, охватывающих около 200 классов дорожных знаков.

Ключевые слова: Компьютерное зрение; детекция объектов; YOLO; дорожные знаки; глубокое обучение; классификация изображений; трекинг объектов.

Язык: Русский.

Статья поступила в редакцию 1 октября 2026 г.

Title: Intelligent information system for accounting and inventory of traffic signs.

Authors: Abdalnagimov A. I., Galitskov B. V., Akhunov O. M.

Abstract: This work focuses on improving road sign inventory methods, which are labor-intensive and expensive. Manual inspection and recording of signs takes two to three hours per kilometer of road, increasing the cost of the work by 70–80 %. The paper presents an intelligent information system for automatic detection and classification of traffic signs using modern deep learning methods. A two-stage architecture based on YOLO models has been developed: a YOLOv11m detector for sign detection in video streams and a YOLOv8 classifier for determining the sign type according to GOST R 52290-2004. To solve the problem of class imbalance in the training set, synthetic data generation based on vector images of signs was applied. A MultiSignTracker tracking algorithm was implemented to eliminate duplicate detections. The system provides automatic extraction of GPS coordinates from video metadata and linking them to detected signs. A graphical interface for project management and results visualization has been developed. The system supports data export to IndorTrafficPlan and KML formats. Detection accuracy is 99.64 %. The final dataset includes approximately 35 750 images covering approximately 200 classes of traffic signs.

Key words: Computer vision, object detection, YOLO, traffic signs, deep learning, image classification, object tracking.

Language: Russian.

The article was received by the editors on 1 October 2026.