

# Система оценки характеристик транспортного потока с использованием компьютерного зрения

Р. А. Бурдин

*Санкт-Петербургский государственный  
электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)*

Rodiogch@gmail.com

Н. И. Куракина

*Санкт-Петербургский государственный  
электротехнический университет  
«ЛЭТИ» им. В.И. Ульянова (Ленина)*

NKurakina@gmail.com

**Аннотация.** Разработана система автоматизированного подсчёта и классификации транспортных средств с использованием нейросетевой архитектуры YOLOv11n и алгоритма трекинга DeepSORT. Основные этапы включают сбор и разметку датасета из 6000 изображений, обучение модели для распознавания четырёх классов ТС (легковые автомобили, микроавтобусы, грузовики, автобусы), создание программы для анализа видеопотока и тестирование её эффективности. Полученные результаты подтверждают перспективность использования компьютерного зрения для анализа транспортных потоков.

**Ключевые слова:** транспортный поток; интенсивность движения; компьютерное зрение; нейросеть; загрязнение воздуха.

## I. ВВЕДЕНИЕ

Оценка характеристик транспортного потока и его воздействия на загрязнение атмосферного воздуха является одной из ключевых задач в области экологии. Современные мегаполисы сталкиваются с непрерывным ростом числа транспортных средств, что приводит к значительному увеличению выбросов загрязняющих веществ и ухудшению качества атмосферного воздуха. В этой связи точный анализ интенсивности дорожного движения становится необходимым для оценки экологического воздействия автотранспорта.

Традиционные методы подсчета транспортных потоков, такие как ручной учет или использование стационарных датчиков, часто оказываются трудоемкими, дорогими и ограниченными в охвате исследуемых участков дорожной сети. Для решения этой проблемы предложено применение современных технологий компьютерного зрения и машинного обучения, которые позволяют автоматизировать процесс сбора данных и повысить точность анализа.

В рамках настоящего исследования разработан метод автоматизированной оценки интенсивности дорожного движения на основе алгоритма YOLO (You Only Look Once), адаптированного для задачи детекции и классификации транспортных средств. Использование данной технологии обеспечивает высокую производительность обработки данных и возможность масштабирования системы для анализа различных участков дорожной сети. Полученные результаты могут быть использованы для моделирования транспортных потоков и оценки уровня загрязнения атмосферного воздуха.

Целью работы является создание автоматизированного инструмента для оценки интенсивности дорожного движения, который позволит повысить точность расчетов выбросов загрязняющих веществ. Предложенный подход демонстрирует значительный потенциал для применения в современных системах мониторинга загрязнения атмосферного воздуха и оптимизации транспортных потоков.

## II. МЕТОДЫ ОЦЕНКИ ТРАНСПОРТНЫХ ПОТОКОВ

Существует несколько методов оценки транспортного потока, которые можно разделить на традиционные и современные.

Традиционные методы:

- Ручной подсчёт – анализ количества транспортных средств наблюдателями.
- Индуктивные датчики – установка сенсоров на дорожном покрытии для фиксации проезжающих автомобилей.

Современные методы:

- Компьютерное зрение – использование видеокamer и алгоритмов обработки изображений для автоматизированного анализа транспортного потока.
- Анализ данных с мобильных устройств – получение информации о передвижении автомобилей на основе GPS и мобильных сетей.
- Датчики IoT – подключённые к интернету устройства, устанавливаемые на светофорах и вдоль дорог, которые передают данные о транспортном потоке.
- Моделирование транспортных потоков – использование математических моделей и симуляций для прогнозирования нагрузки на дорожную сеть и оценки воздействия различных транспортных стратегий.

Компьютерное зрение в сочетании с нейронными сетями открывает принципиально новые возможности для анализа транспортного потока. Современные технологии позволяют не просто собирать данные с видеокamer, но и глубоко интерпретировать их с помощью алгоритмов искусственного интеллекта. Например, нейросетевые архитектуры вроде YOLO (You Only Look Once) или Faster R-CNN обеспечивают сверхточное детектирование транспортных средств на видео и их классификацию по типам — от легковых автомобилей до грузовиков, автобусов и мотоциклов.

Это особенно критично для экологического анализа: автоматическое определение класса ТС позволяет учитывать специфические характеристики выбросов каждого типа транспорта. Благодаря нейросетям обработка видео становится не только быстрой, но и масштабируемой — системы способны анализировать потоковые данные с десятков камер в реальном времени, обеспечивая детализированную картину транспортной обстановки. Такая интеграция компьютерного зрения и ИИ делает метод экономически эффективным, точным и адаптивным к сложным сценариям, например, при высокой интенсивности движения.

#### *А. Нейросеть YOLO*

YOLO — это одна из наиболее популярных нейросетей, предназначенных для обнаружения объектов на изображениях. Её ключевое преимущество заключается в высокой скорости работы при сохранении приемлемого уровня точности. В отличие от таких методов, как R-CNN или Faster R-CNN, YOLO выполняет детекцию объектов за один прогон сети, что существенно снижает время обработки изображений. Данный подход делает YOLO особенно полезной в реальных приложениях, где важна скорость обработки, например, в системах видеонаблюдения, беспилотных автомобилях и автоматизированных промышленных решениях.

В основе YOLO лежит сверточная нейросеть, которая разбивает входное изображение на сетку фиксированного размера. Процесс детекции объектов включает несколько ключевых этапов. На стадии подготовки данных изображение делится на сетку ячеек, каждая из которых будет использоваться для предсказания. Далее на этапе предсказания нейросеть формирует ограничивающие рамки (bounding boxes) и вычисляет их координаты, размеры, вероятность наличия объекта и его класс. Затем на этапе фильтрации рамки с низкой вероятностью отбрасываются, а метод подавления немаксимальных значений (Non-Maximum Suppression, NMS) устраняет дублирующие предсказания, оставляя только наиболее значимые. В результате формируется итоговое изображение, на котором отмечены обнаруженные объекты с соответствующими им классами.

#### *В. Алгоритм трекинга DeepSORT*

DeepSORT (Deep Simple Online and Realtime Tracking) является усовершенствованной версией алгоритма SORT (Simple Online and Realtime Tracking), которая объединяет объекты детекции и глубокое обучение для повышения точности и надежности трекинга объектов в реальном времени. Основная идея заключается в использовании сверточных нейронных сетей для генерации описательных признаков объектов, что позволяет более точно различать и группировать их в условиях динамического изменения сцены.

Ключевым компонентом DeepSORT является использование предобученной сверточной нейронной сети для извлечения признаков объектов. Эти признаки представляют собой векторные представления объектов, которые затем используются для сравнения и группировки объектов в процессе трекинга.

Для того чтобы обеспечить эффективный трекинг, DeepSORT использует несколько стратегий. Во-первых,

алгоритм применяет косинусное расстояние для оценки сходства между признаковыми векторами объектов. Это позволяет точнее определять, принадлежат ли два объекта к одной и той же цели. Кроме того, алгоритм включает фильтр Калмана для предсказания будущего положения объекта и корректировки треков на основе новых данных. Это особенно полезно в условиях, когда объект временно исчезает из поля зрения камеры или когда несколько объектов пересекаются.

Для улучшения точности трекинга DeepSORT также использует методы фильтрации и обновления треков. Например, алгоритм может временно игнорировать новые объекты, если они слишком близко расположены к уже существующим трекам, что помогает избежать ошибок при создании новых треков для одного и того же объекта. Кроме того, DeepSORT может использовать исторические данные о движении объектов для улучшения предсказаний их будущих позиций, что делает его особенно полезным в сложных сценах с множеством объектов.

Для повышения производительности и снижения влияния шумов, DeepSORT включает механизмы обработки ложных срабатываний и устранения треков, которые не соответствуют реальным объектам. Например, если трек не обновляется в течение некоторого времени, он может быть автоматически удален. Это особенно важно в условиях, где объекты могут временно выходить из поля зрения камеры или где есть высокий уровень шума.

Для обучения и использования DeepSORT требуется предобученная сверточная нейронная сеть, которая может быть взята из открытых источников или обучена на специальных наборах данных. В данной работе будет использоваться обученная на YOLO нейронная сеть.

### III. ОБУЧЕНИЕ НЕЙРОННОЙ СЕТИ

Разработка модели на основе YOLO для автоматизированной детекции и классификации транспортных средств включает в себя следующие этапы:

- Подготовка датасета изображений для каждой категории транспорта.
- Выполнение разметки объектов.
- Разделение на выборки.
- Создание конфигурационного файла.
- Обучение модели.
- Тестирование модели.

#### *А. Подготовка датасета*

Подготовка датасета для работы с архитектурой YOLO представляет собой основной этап в процессе обучения модели для задач компьютерного зрения. Этот процесс требует тщательной организации данных, их предварительной обработки и форматирования в соответствии с требованиями архитектуры. В первую очередь необходимо собрать исходные изображения, которые будут использоваться для обучения и тестирования модели.

Датасет должен быть репрезентативным, охватывать разнообразные сценарии (условия освещения, ракурсы, размеры объектов, фоновые помехи) и соответствовать

задаче классификации транспортных средств по пяти категориям описанных выше. Размеры фотографий должны быть не меньше 640x640 пикселей.

Для получения исходных данных существует множество открытых датасетов, содержащие как просто фотографии, так и уже размеченные файлы. Так для создания собственного датасета был использованы фотографии из датасета COCO, а также фотографии, отобранные вручную в результате поиска в интернете, например с записей онлайн камер. Итого получился датасет на 6000 фотографий.

### В. Разметка объектов

После сбора изображений требуется разметить их, то есть указать на каждом изображении местоположение и классы всех объектов, которые модель должна научиться распознавать. В обучаемой модели будет 4 класса транспортных средств: «car», «van», «truck» и «bus».

Разметка выполняется вручную или с использованием специализированных инструментов. В данной работе разметка выполнялась инструментом LabelImg. Для YOLO разметка представлена в виде текстовых файлов, где для каждого объекта указываются координаты ограничивающего прямоугольника (bounding box) в нормализованном формате. Нормализация подразумевает приведение координат к относительным значениям в диапазоне от 0 до 1, что позволяет модели работать с изображениями различных размеров без потери точности. Каждый текстовый файл содержит строки, соответствующие всем объектам на изображении, где каждая строка состоит из идентификатора класса и четырёх чисел: центральных координат прямоугольника (x, y) и его ширины и высоты (w, h). Например, для фото ниже разметочный файл будет содержать «1 0.507812 0.534038 0.562500 0.626761».



Рис. 1. Пример разметки объекта

### С. Разделение выборки

Следующим шагом является разделение датасета на обучающую, валидационную и тестовую выборки. Это делается для того, чтобы модель могла обучаться на одной части данных, проверять свою производительность на другой части и, наконец, демонстрировать окончательные результаты на независимой тестовой выборке. Обычно рекомендуется использовать соотношение 80/10/10 или 70/15/15 для

обучающей, валидационной и тестовой частей соответственно. При этом важно учитывать баланс классов, чтобы каждая категория была достаточно представлена во всех выборках.

### Д. Создание конфигурационного файла

После подготовки данных необходимо создать конфигурационный файл, который определяет структуру датасета и параметры обучения. Этот файл представляют собой текстовый документ формата yaml. Конфигурация датасета «data.yaml» содержит в себе пути к данным, список классов и метаданные. Он является обязательным для обучения модели YOLO. Важно отметить, что качество датасета напрямую влияет на эффективность модели. Ошибки в разметке, несбалансированное представление классов или недостаточное количество данных могут привести к снижению точности детектирования. Поэтому перед началом обучения следует провести тщательную проверку данных, убедиться в их корректности и при необходимости внести исправления.

### Е. Обучение модели

Для обучения модели необходимо указать определённые параметры. Почти все из них подбираются автоматически, поэтому пользователю для большинства случаев хватит указать только самые основные.

- Data – Путь к конфигурационному файлу .yaml, который содержит информацию о датасете.
- Epochs – Количество эпох обучения (сколько раз модель пройдет через весь датасет). Для малых датасетов оптимальным количеством будет 50-100 эпох, для больших 30-50, дабы избежать переобучения.
- Imgsz – Размер входных изображений (ширина и высота в пикселях). YOLO изменяет размеры всех изображений в датасете до заданного. Стандартное значение 640, для ускоренного обучения можно использовать 320. Главное условие: размер должен быть кратен 32.
- Batch – Размер батча (сколько изображений обрабатывается за один шаг обучения).

Также можно указать множества дополнительных параметров для оптимизации обучения, аугментации данных, дополнительных метрик и прочих деталей обучений, но в рамках данной работы достаточно и автоматической настройки.

### Ф. Тестирование модели

Запуск процесса обучения нейросети происходит через разработанный python скрипт. Сначала происходит загрузка предобученной модели YOLOv11n. Далее происходит обучение нейросети на датасете пользователя. После обучения модель тестируется на валидационных данных. В результате, в директории скрипта создаётся папка «runs», которая содержит файлы модели, графики метрик, тестовые данные и визуализации. По полученным графикам метрик можно оценить точность и успешность обучения модели.

Результаты анализа показали, что модель хорошо различает типы транспорта и приблизительно определяет их положение, но не достигает максимальной точности в локализации. Так как для нашей задачи главное подсчёт

количества ТС, а не идеальные границы, то такой результат является допустимым.

#### IV. ПРОГРАММА ПОДСЧЕТА КОЛИЧЕСТВА ТРАНСПОРТНЫХ СРЕДСТВ

В результате обучения нейронной сети формируются веса – это параметры модели, которые определяют силу связи между нейронами в разных слоях. Они представляют собой числовые значения, настраиваемые во время обучения, чтобы минимизировать функцию потерь и улучшить качество предсказаний. Для задачи детекции транспорта необходимо загрузить модель в YOLO, используя полученные при обучении веса.

Чтобы продемонстрировать результаты работы модели, был написан скрипт, который обрабатывает видео в реальном времени и выводит пользователю обработанное изображение. Модель обрабатывает каждое изображение и отмечает рамкой обнаруженные объекты. Также указывается класс объекта и вероятность его принадлежности к классу.

Но для задачи подсчёта транспорта только YOLO недостаточно. Так как модель обрабатывает кадры независимо друг от друга, то есть уже подсчитанный транспорт на прошлом кадре, на новом будет снова идентифицирован как уникальный и лишней раз подсчитан. Для решения этой проблемы была внедрена система трекинга на основе библиотеки DeepSORT. DeepSORT сохраняет уникальные ID объектов, связывая их между кадрами через алгоритмы предсказания движения и сравнения визуальных признаков. Это позволяет точно подсчитывать количество уникальных транспортных средств, игнорируя повторные детекции.

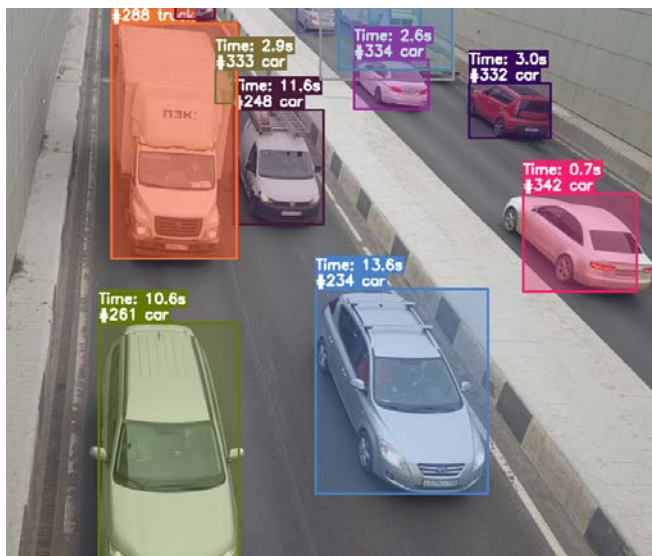


Рис. 2. Пример обработки видео с использованием DeepSORT

При использовании DeepSORT, теперь помимо информации о виде транспорта, указывается персональный id объекта, запоминаемый между кадрами (для наглядности на рис. 2 указывается и время нахождения объекта в видео).

Как итог, для выполнения данной задачи, была разработана программа на языке Python. Программа реализована в виде приложения на встроенной в Python

библиотеке tkinter. Пользователь выбирает путь к обученной модели и к видео. Также, по желанию, есть возможность сохранить видео с отмеченной разметкой. Прогресс анализа отображается в нижней части окна. Программа обрабатывает каждый кадр с помощью выбранной модели и получает bbox каждого объекта и принадлежность к классу. Эти данные проходят обработку через алгоритм DeepSORT, который присваивает объектам id и обновляет их треки. После завершения обработки в папке приложения создаётся файл формата .csv, удобный для переноса в ГИС. В файле хранится таблица, состоящая из названия классов и соответствующее классу количество объектов на видео.

#### V. ЗАКЛЮЧЕНИЕ

В результате проведённого исследования была разработана и протестирована модель машинного обучения на основе нейросетевой архитектуры YOLOv11n и алгоритма трекинга DeepSORT для автоматизированного анализа транспортных потоков. Модель продемонстрировала высокую эффективность в задачах детекции и классификации транспортных средств по четырём категориям (легковые автомобили, микроавтобусы, грузовики, автобусы). Было разработано программное решение, способное обрабатывать видеопоток в реальном времени, отслеживать уникальные объекты с помощью трекинга и формировать отчёт в формате CSV для последующего анализа. Высокая точность распознавания автобусов и легковых автомобилей подтверждает перспективность использования комбинированного подхода на основе компьютерного зрения и глубокого обучения. Вместе с тем, ошибки классификации между грузовиками и микроавтобусами, а также зависимость точности от ракурса видеозаписи и качества изображений свидетельствуют о недостаточной сбалансированности датасета и необходимости его пополнения. Анализ показал, что погрешность подсчёта транспортных средств варьируется в зависимости от условий съёмки, что связано с преобладанием в обучающей выборке изображений с низким разрешением и боковым ракурсом. Для повышения надёжности модели рекомендуется расширить датасет за счёт добавления кадров с верхнего ракурса, увеличить объём данных для менее представленных классов и применить методы аугментации. Кроме того, оптимизация гиперпараметров и переход к более глубоким версиям YOLO позволит улучшить локализацию объектов и снизить ошибки при перекрытии транспортных средств.

#### СПИСОК ЛИТЕРАТУРЫ

- [1] Силионов И.Н. Применение нейронных сетей в области компьютерного зрения для создания систем умного наблюдения и безопасности // Вестник науки. 2025. Т.3 №3(84) С.594-601
- [2] Куракина Н.И., Мышко Р.А., Мухутдинов А.Р., Бурдин Р.А. Оценка дорожного трафика с использованием нейронных сетей // Приборы. 2024. № 11 (293). С. 30-36.
- [3] Мышко Р.А., Куракина Н.И., Бурдин Р.А. Система оценки загрязнения атмосферного воздуха в условиях городской среды // Материалы XXVIII Междунар. конфер. по мягким вычислениям и измерениям (SCM'2025), СПбГЭТУ, 28-30 мая 2025 г.
- [4] Ultralytics. YOLOv11 Documentation [Электронный ресурс]. 2024–2025. — URL: <https://docs.ultralytics.com/models/yolo11/> (дата обращения: 11.03.2026)