

Общее руководство пользователя системы  
«РоадАР Трекер»

## **Содержание**

Введение	2
Назначение и условия применения система “РоадАР Трекер”	4
1 Назначение программы	4
2 Базовый функционал система “РоадАР Трекер” включает:	4
3 Программное обеспечение рабочего места	4
4 Виды пользователей Системы	4
5 Описание Системы	5

## **Введение**

Система “РоадАР Трекер” – система, разработанная компанией ООО “РоадАР”, предназначенная для распознавания транспортных средств (далее - ТС) и их трекинга в системах фотовидеофиксации и видеонаблюдения.

Текущая версия системы “РоадАР Трекер” позволяет решать следующие задачи:

- идентификация изображений для анализа;
- поиск на изображении ТС;
- определение ТС с помощью нейронной сети и их трекинг;
- возврат сообщения с трекингом ТС.

Система “РоадАР Трекер” поддерживает механизмы взаимодействия со сторонними информационными системами. Текущая версия системы предполагает выбор и реализацию конкретного механизма взаимодействия с внешними системами по согласованию с заказчиками.

Функционал системы:

- обработка изображений;
- определение ТС и их трекинг;
- информационный обмен с внешними системами.

# Назначение и условия применения система “РоадАР Трекер”

## 1 Назначение программы

- Обработка изображений для определения Трекер ТС.

## 2 Базовый функционал система “РоадАР Трекер” включает:

- обработка изображений;
- определение ТС и их трекинг;
- информационный обмен с внешними системами.

## 3 Программное обеспечение рабочего места

Библиотека может быть встроена в программное обеспечение, исполняемое на рабочем месте пользователя. В этом случае клиентская часть системы “РоадАР Трекер” может использоваться на любом рабочем месте, имеющем подключение к сети Internet (или сети передачи данных предприятия) и установленный браузер (в среде операционной системы Windows 7 и выше). Разрешающая способность видеосистемы и монитора – не ниже 1280x1024. Рекомендуется широкоформатный монитор.

Вид ПО	Программный продукт
ОС (приведены варианты)	Windows 7 и выше Apple Mac OS X 10.6 и выше Linux Android 4.0 и выше iOS 7.0 и выше

Таблица 1. Системные требования

Данные требования могут меняться в зависимости от особенностей программного обеспечения, которое использует библиотеку системы “РоадАР Трекер”.

## 4 Виды пользователей Системы

Пользователем системы является пользователь программного обеспечения, в которое встраивается библиотека системы “РоадАР Трекер”. Ниже перечислены базовые минимальные пользователи, необходимые для функционирования программного обеспечения на базе библиотеки, при этом могут присутствовать другие виды пользователей. Предусмотрены следующие виды пользователей:

- *Пользователь «Администратор».* Может обладать правами на изменение основных настроек системы “РоадАР Трекер”.

- *Пользователь.* Учётные записи этих пользователей создаются, удаляются и редактируются Администратором. Такой пользователь может только просматривать основные данные.

Описание доступных возможностей API выполняется для пользователя «Администратор», как обладающего максимально возможными правами по доступу к программе.

## 5 Описание Системы

API Системы написано на C++ и предоставляет несколько основных классов для распознавания и трекинга ТС.

### ## Класс `InferenceEngine`

Создание нейросетевого движка для детектора ТС.

Инициализация распознавателя с минимальным набором аргументов:

...

```
RoadarRecognitionFactory::buildVehicleInferenceEngine(threadsNum,      "../networks",
"../networks", RoadarNumbers::NNPreferredBackend::OPENCV, 0);
```

...

### ### Конструктор

Конструктор принимает 6 аргументов, 2 из них опциональные:

- `numThreads` Максимальное число потоков доступных для работы. Если значение равно 0, то алгоритм будет работать без распараллеливания.

- `networksPath` Путь до папки networks, где лежат файлы с нейронными сетями..

- `cachePath` Путь для кеша. Если строка пустая, то кэш не будет сохраняться.(Используется при обработке нейросетей на GPU)

- `preferredBackend` Предпочитаемый тип бэкенда для нейронок.

- `device` Номер устройства.(Используется при обработке нейросетей на GPU)

- `saveCacheToRAM` Флаг для сохранения кеша в RAM.

После инициализации, InferenceEngine передаётся в buildVehicleDetector для получения системы трекинга ТС - IVehicleDetector:

...

```
auto vehicleRecognizer = RoadarRecognitionFactory::buildVehicleDetector(inferenceEngine,
false);
```

...

### ### Метод `processFrame`

Синхронный обработчик фреймов.

...

```
virtual std::vector<VehicleTrack> processFrame(const Frame &inputFrame,  
                                              const NumbersTracksHistory &numbersTracksHistory) = 0;
```

...

**\*\*Параметры:\*\***

- `inputFrame` - Входной кадр

- `numbersTracksHistory` - Текущие треки номеров, нужны для сопоставления с машинами

Возвращает вектор треков ТС.

**\*\*Использование:\*\***

Метод следует использовать для процессинга кадров детекции и трекинга ТС.

**### Метод `getCurrentTracks`**

Возврат текущих распознающихся треков ТС

...

```
virtual std::vector<VehicleTrack> getCurrentTracks(uint64_t &lastProcessedFrameLabel) =  
0;
```

...

**\*\*Параметры:\*\***

- `lastProcessedFrameLabel` - Метка последнего обработанного кадра

Возвращает текущие распознающиеся треки ТС.

**\*\*Использование:\*\***

Метод следует использовать, чтобы получить информацию о текущих распознающихся треках ТС.