VisionLabs
MACHINES CAN SEE

VISIONLABS LUNA CARS STREAM

Инструкция по установке

ООО «ВижнЛабс»
123458, г. Москва, ул. Твардовского д. 8, стр. 1
+7 (499) 399 3361
info@visionlabs.ru
www.visionlabs.ru

Оглавление

Глоссарий

Термин	Определение
TrackEngine	Библиотека, используемая для отслеживания положения ТС или ГРЗ в последовательности кадров и выбора лучшего кадра.
ГРЗ	Государственный регистрационный знак транспортного средства.
Дескриптор	Набор данных в закрытом, двоичном формате, подготавливаемый системой распознавания лиц на основе анализируемой характеристики.
Лучший кадр	Кадр видеопотока, на котором ТС или ГРЗ зафиксировано в оптимальном ракурсе для дальнейшего использования в системе.
ПО	Программное обеспечение
ТС	Транспортное средство.

Введение

Настоящий документ представляет собой руководство администратора сервиса CARS.Stream.

Руководство определяет порядок установки, настройки, логирования и администрирования сервиса.

Перед установкой и эксплуатацией сервиса рекомендуется внимательно ознакомиться с настоящим руководством.

Общие сведения

VisionLabs LUNA CARS – система, предназначенная для определения атрибутов транспортных средств и распознавания автомобильных номеров. Система состоит из трёх сервисов: CARS.Analytics, CARS.API и CARS.Stream.

VisionLabs LUNA CARS.Stream – сервис, предназначенный для детекции и трекинга транспортных средств и ГРЗ в видеопотоке или детекции на изображениях. Основные функции системы представлены ниже:

- Обработка видеопотока;
- Детекция и трекинг ТС и ГРЗ;
- Выбор лучшего кадра;
- Отображение результатов детекции и распознавания.

Системные требования

Для установки и работы ПО CARS.Stream необходимо учитывать ряд требований и условий, указанных ниже. Список системных требований представлены в Таблице 1.

Таблица 1. Системные требования

Необходимый ресурс	**Рекомендовано**
Процессор (CPU)	Intel, не менее 4 физических ядер с тактовой частотой не менее 2,0 ГГц и поддержкой инструкций AVX 2.
Оперативная память (RAM)	Не менее 8 Гб DDR4.
Объем свободного дискового пространства (HDD/SSD)	Не менее 20 Гб.
Операционная система (OS)	РЕД ОС 7.3
Сторонние зависимости	Python 3.6; PostgreSQL 9.6; postgreSQL-libs; GCC 8.1.
Поддерживаемые версии браузеров	Microsoft Edge (версия 44.0 и выше); Mozilla Firefox (версия 60.3.0 и выше); Google Chrome (версия 50.0 и выше).

На требования к системе влияют несколько факторов:

- Количество обрабатываемых видеопотоков;
- Частота и разрешение кадров видеопотоков;
- Параметры настройки CARS.Stream. Настройки по умолчанию являются наиболее универсальными. В зависимости от условий эксплуатации приложения с помощью значений настроек можно повлиять на качество или производительность.

CARS.Stream также может работать в режиме ускорения вычислений за счет:

- Использования AVX-инструкций (определяется автоматически при установки).

1. Установка

1.1. Подготовка к установке

Дистрибутив CARS.Stream представляет собой архив «cars-stream_v.*.zip».

Архив содержит компоненты, необходимые для установки и эксплуатации сервиса. Архив не включает некоторые зависимости, которые входят в стандартную поставку РЕД ОС и могут быть загружены из открытых источников.

> Дальнейшие действия необходимо выполнять под учетной записью суперпользователя (с root-правами).

1.2. Настройка конфигурационного файла «TrackEngine.conf»

Данный раздел описывает параметры конфигурационного файла «TrackEngine.conf», которые используются для настройки CARS.Stream. Конфигурационный файл находится в дистрибутиве CARS.Stream в директории /bin/data/trackengine.conf.

1.2.1. Detector-step

Параметр «detector-step» позволяет указать с какой частой происходит детекция ТС и ГРЗ в заданной области до выполнения детекции (редецекция). Редетекция требует меньше ресурсов, но объект может быть потерян при задании большого количества кадров.

```
<!-- detector-step: The count of frames between frames with full detection,
[0 .. 30] ('7' by default). -->
<param name="detector-step" type="Value::Int1" x="7" />
```

1.2.2. Detector-scaling

Параметр «detector-scaling» позволяет масштабировать кадр перед обработкой. Масштабирование кадра необходимо для того, чтобы понизить размер кадра, а следовательно, понизить время обработки.

```
<!-- detector-scaling: Scale frame before detection for performance reasons,
[0, 1] ('0' by default). -->
<param name="detector-scaling" type="Value::Int1" x="0" />
```

1.2.3. Scale-result-size

Параметр «Scale-result-size» задаёт максимальный размер кадра после масштабирования по наибольшей из сторон кадра. Если исходный кадр имел размер 1920x1080 и значение «scale-result-size» равно 640, то разрешение кадра будет изменено до 640x360.

Время детекции понижается при уменьшении разрешения кадра.

```
<!-- scale-result-size: If scaling is enable, frame will be scaled to this
```

```
    size in pixels (by the max dimension - width or height).
    Upper scaling is not possible. ('640 by default') -->
    <param name="scale-result-size" type="Value::Int1" x="640" />
```

> Если кадр был обрезан с помощью параметра «roi», то масштабирование будет применено к уменьшенному кадру. В этом случае значение «scale-result-size» следует задавать в зависимости от наибольшей стороны обрезанного кадра.

1.2.4. Frg-subtractor

Параметр «frg-subtractor» включает режим учёта перемещений в кадре. Последующая детекция ТС и ГРЗ будет выполняться только в областях с движением.

Включение параметра «frg-subtractor» увеличивает производительность работы CARS.Stream.

```
    <!-- frg-subtractor: Use foreground subtractor for filter of frames, [0, 1]
    ('1' by default). -->
    <param name="frg-subtractor" type="Value::Int1" x="1" />
```

1.2.5. Frg-regions-alignment

Параметр «frg-regions-alignment» позволяет задать выравнивание кадра для областей с движением.

```
    <!-- frg-regions-alignment: frg regions alignment. Useful for detector
    better batching utilization. -->
    <!-- 0 or negative values mean using non aligned regions, (0 by default).
    -->
    <param name="frg-regions-alignment" type="Value::Int1" x="0" />
```

1.2.6. Frg-regions-square-alignment

При включённом параметре «frg-regions-square-alignment» ширина и высота области с движением всегда будут равны.

```
    <!-- align frg regions to rect with equal sides (max side choosen). See frgregions-
    alignment, [0, 1] ('1' by default). -->
    <param name="frg-regions-square-alignment" type="Value::Int1" x="1" />
```

1.2.7. Batched-processing

Параметр «batched-processing» включает пакетную обработку кадров. При работе с несколькими камерами, с каждой камеры собирается по кадру. После чего происходит обработка данного пакета кадров. Если параметр отключён, то кадры обрабатываются один за другим.

```
    <!-- batched-processing: Process streams frames in batch or separately, [0,
    1] ('1' by default). -->
    <param name="batched-processing" type="Value::Int1" x="1" />
```

> При использовании режима пакетной обработки увеличивается задержка перед обработкой, но при этом сама обработка выполняется быстрее.
> Рекомендуется включать параметр и при использовании GPU, и при использовании CPU.

1.2.8. Min-frames-batch-size

Параметр «min-frames-batch-size» задаёт минимальное количество кадров, которое следует накопить со всех камер перед обработкой.

```
<!-- min-frames-batch-size: stream frames min batch size value to process, (
'0' by default). -->
<!-- higher values lead to higher processing latency but increase throughput
and device utilization. -->
<!-- zero/negative values disable this feature, so any stream frames will be
processed if they are available -->
<!-- note: this parameter should be regulated with 'max-frames-batch-gathertimeout'
(see below) -->
<param name="min-frames-batch-size" type="Value::Int1" x="0" />
```

> Рекомендуется выставлять значение параметра «min-frames-batch-size» равным количеству подключённых потоков при использовании GPU.
> Рекомендуется выставлять значение параметра «min-frames-batch-size» равным «2» при использовании CPU.

1.2.9. Max-frames-batch-gather-timeout

Параметр «max-frames-batch-gather-timeout» задаёт время между обработкой пакетов кадров. Если обработка одного кадра укладывается в указанное время и остается запас, то CARS.Stream ожидает дополнительные кадры.

Если «max-frames-batch-gather-timeout» равен 20 мс, то за это время обрабатывается предыдущий пакет и собирается новый пакет. Через 20 мс начинается следующая обработка, даже если не удалось собрать число кадров равное «min-frames-batch-size». Следующая обработка кадров не может начаться до окончания обработки предыдущего пакета кадров.

Если значение параметра равно «0», то задержки на наполнение пакета кадрами отсутствует, а обработка происходит непрерывно и «min-frames-batch-size» игнорируется.

```
<!-- max-frames-batch-gather-timeout: max available timeout to gather next
stream frames batch (see 'min-frames-batch-size') from last processing
begin time point (measured in ms), ('-1' by default). -->
<!-- negative values disable this feature (no timeout, so only stream frames
batches with size no less than 'min-frames-batch-size' value will be
processed) -->
<!-- note: this parameter is complementary to 'min-frames-batch-size' and
controls min average fps of stream frames batches processing -->
<param name="max-frames-batch-gather-timeout" type="Value::Int1" x="-1" />
```

> Рекомендуется выставлять значение параметра «max-frames-batch-gather-timeout» равным «0» и при использовании GPU, и при использовании CPU.

1.3. Установка через Ansible

Необходимо установить пакет ansible, выполнив команды:

```
# обновление менеджера пакетов
yum update
# установка дополнительных репозиториев
yum install epel-release
# установка ansible
yum install ansible
```

1.3.1. Настройка SSH

Необходимо сгенерировать SSH-ключ и добавить его на целевой сервер. Для начала необходимо проверить и настроить SSH сервис

```
# проверка работоспособности ssh сервиса
systemctl status sshd
# если сервис не активен, необходимо его запустить
systemctl start sshd
# если сервис не установлен, необходимо его установить
yum install -y openssh-server
```

После этого необходимо сгенерировать ключ, выполнив команду:

```
ssh-keygen
```

При необходимости можно задать ключевую фразу или оставить ее пустой. Для копирования ключа на целевой сервер необходимо ввести команду:

```
# копирование открытого SSH-ключа на целевой сервер
ssh-copy-id username@hostname
```

где «username» - имя авторизованного пользователя, а «hostname» - IP-адрес целевого сервера.

> Данный способ не является единственно возможным. Вы можете использовать любой другой удобный способ для обеспечения ssh-доступа на целевой сервер.

1.3.2. Настройка конфигурационного файла «hosts»

В комплекте поставки в директории /ansible находится файл «hosts», в разделе CARS.Stream необходимо задать внешний IP-адрес сервера куда будет устанавливаться приложения:

```
#CARS.Stream
#Multiple hosts allowed
[stream]
<IP_адрес>
```

1.3.3. Настройка путей

Перед началом установки необходимо произвести настройку системы в файле «all.yml», расположенного в директории /ansible/group_vars. Описание основных параметров находится в Таблице 2.

Таблица 2. Основные параметры системы.

#	Параметр	Описание
1	luna_cs_vers	Указывает имя архива. Например, carstream_linux_v.1.0.6.
2	luna_cs_zip_location	В этом параметре задается путь до архива CARS.Stream. Например, /distr/stream/carstream_linux_v.zip.

1.3.4. Запуск установки через Ansible

> В процессе установки будет отключен встроенный firewall и selinux (в дальнейшем потребуется перезагрузка).
> В некоторых случаях может появиться ошибка о недоступности репозиториев, в этом случае необходимо перезапустить установку.

Для запуска установки необходимо находиться в директории /ansible и выполнить команду на запуск процедуры установки:

```
ansible-playbook -i hosts install_stream.yml
```

1.3.5. Проверка работоспособности

По окончании установки необходимо проверить состояние сервиса CARS.Stream командой:

```
systemctl status luna-cars-stream
```

При корректной работе система не выдает сообщения об ошибке.

1.4. Установка при помощи Docker

1.4.1. Установка docker и docker-compose

Используйте официальную инструкцию для установки docker и docker-compose под РЕД ОС.

1.4.2. Настройка файла окружения «.env»

Параметры файла окружения «.env» представлены в Таблице 3. Файл окружения находится в архиве CARS.Stream.

Таблица 3. Параметры файла окружения.

#	Параметр	Описание
1	HASP_license_server	Задает путь до сервера, к которому установщик обращается за доступной сетевой лицензией на продукт. Если сетевой лицензии нет, то она должна задаваться локально.
2	HASP_wait_time	Задает время ожидания ответа при отправке запроса о доступной лицензии на сервер. Задается в минутах.
3	Emirates	Выбор страны распознавания ГРЗ. Доступные значения: • true – система будет распознавать ГРЗ только ОАЭ; • false – система будет распознавать ГРЗ РФ, СНГ, ЕС.
4	ENG	Задает язык системы. Доступные значения: • true – английский язык системы; • false – русский язык системы.

1.4.3. Запуск установки

Скопируйте архив CARS.Stream в директорию /distr/stream.

В корневой папке (где расположен docker-compose.yml) выполните команду

```
docker -compose up –d
```

После изменения любых настроек необходимо запускайте систему с ключом пересборки:

```
docker -compose up -d –build
```

1.5. Управление системой

Для управления сервисом используйте следующие команды:

```
# Запуск сервиса
sudo systemctl start luna-cars-stream
# Перезапуск сервиса
sudo systemctl restart luna-cars-stream
# Узнать состояние сервиса
sudo systemctl status luna-cars-stream
# Остановить сервис
sudo systemctl stop luna-cars-stream
```

Процесс добавления камер описан в файле «CarStreamServerApi.html», который находится в архиве CARS.Stream в директории /doc.

2. Настройка зоны детекции

Зона детекции представляет собой область кадра в виде прямоугольника определенного размера, внутри которого происходит извлечение атрибутов ТС. Зона детекции используется для фиксации распознанных ТС (событий), проезжающих через данную область кадра.

Настройка зоны детекции происходит в файле «input_roi_config.ini», который находится в директории /stream/bin/data. Описание параметров файла представлено в Таблице 4.

Таблица 4. Параметры файла настройки.

#	Параметр	Описание
1	camera_name	Имя камеры, для которой будет производиться настройка зоны детекции.
2	min_height	Значение верхней границы зоны детекции.
3	max_height	Значение нижней границы зоны детекции.
4	min_width	Значение левой границы зоны детекции.
5	max_width	Значение правой границы зоны детекции.

> Все значения границ задаются в пикселях. Система отсчет координат находится в левом верхнем углу кадра.

После внесения изменений необходимо перезапустить сервис CARS.Stream:

```
sudo systemctl restart luna-cars-stream
```

> Если планируется использовать CARS.Stream совместно с CARS.Analytics, то зоны детекции и распознавания рекомендуется настраивать в веб-интерфейсе.

3. Визуальный режим CARS.Stream

CARS.Stream позволяет просматривать запущенные видеопотоки.

Список запущенных видеопотоков находится по адресу http://<IP_адрес>:34569/api/1/streams.

> Необходимо подставить вместо <IP_адрес> адрес сервера, где установлен CARS.Stream.

На странице веб-браузера представлена информация о запущенных видеопотоках:


```
    [{"alive":1,"health_check":{"max_error_count":10,"period":3600,"retry_delay":5},"id":"
fd69115d-3274-468b-8fcc-8dcd1f12c1e4",
    "input":{"roi":[356,148,1896,1045],
    "rotation":0,
    "transport":"tcp","url":"rtsp://admin:Seven182@11.12.82.113:574/cam/realmonitor?channe
l=1&subtype=0"},
    "name":"StreetView","output":{"token":"","url":""},
    "preview_url":"/api/1/streams/preview/fd69115d-3274-468b-8fcc-8dcd1f12c1e4",
    "video_info":{"bit_rate":0,"frame_rate":25,"gop_size":12,"height":1296,"start_time":"2
021-08-09T16:53:17 MSK","width":2304}}]
```


Для просмотра видеопотока необходимо скопировать значение параметра «preview_url» и добавить его к адресу CARS.Stream: http://<IP_адрес>:34569/api/1/streams/preview/fd69115d-3274-468b-8fcc-8dcd1f12c1e4.

При этом открывается окно в браузере (Рисунок 1), где в режиме реального времени в видеопотоке система производит детекцию и слежение за ТС в зоне детекции, а справа отображаются распознанные системой данные о ТС (номер трека, модель, марка, категория) и ГРЗ (изображение и распознанные системой символы). Границы кадра определяются настройками зоны детекции (п.2).

Просмотр обрабатываемого видео используется для отладки работы камеры.

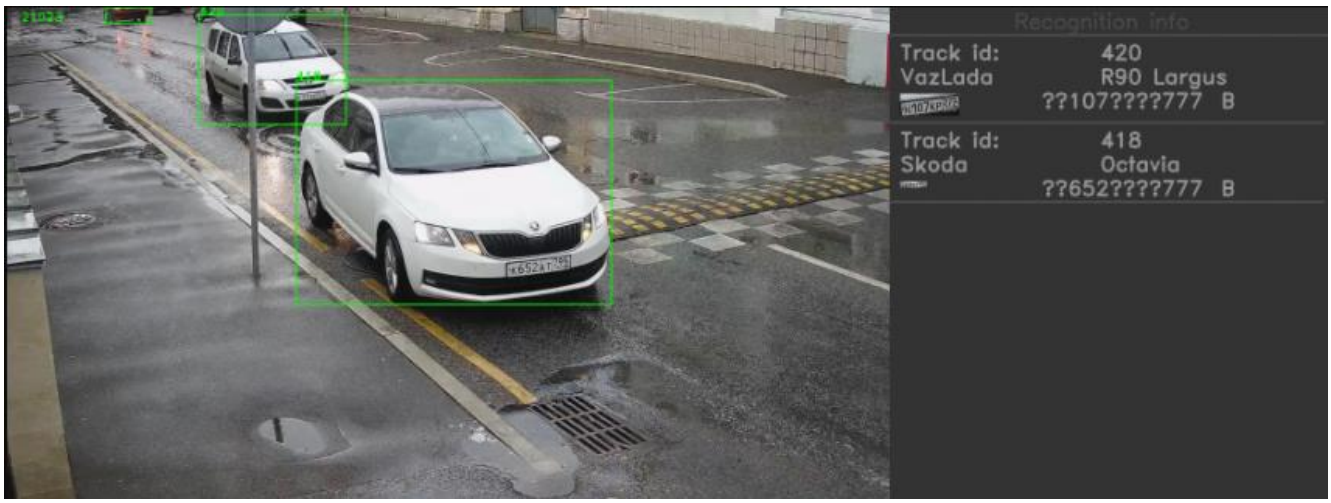

Рисунок 1. Окно просмотра видеопотока камеры.

4. Логирование

Файлы логов сохраняются в /var/lib/luna/cars/stream/bin/log.

Настройка уровня логирования осуществляется в файле /bin/data/csConfig.conf.

4.1. Severity

Severity – параметр, определяющий информацию, которую пользователь хочет получать в логах. Доступны следующие фильтры информации:

- 0 – выводить всю информацию;
- 1 – выводить только предупреждения системы;
- 2 – выводить только ошибки.

```
"severity":
{
   "value": 1,
   "description" : "Logging severity levels …"
}
```

4.2. Mode

Mode – параметр, задающий режим логирования приложения: файл или консоль. Существует три режима:

- l2c – выводить информацию только в консоль;
- l2f – выводить информацию только в файл;
- l2b – выводить информацию и в файл, и в консоль.

```
"mode":
{
   "value": "l2b",
   "description": " Mode of logging … "
}
```

Приложения

Приложение 1.

Таблица 5. Список используемых портов по умолчанию

Порт	Сервис	К порту обращается
34569	CARS.Stream	CARS.Analytics backend, Пользователь (stream preview)
81	Nginx перед CARS.API	CARS.Stream, CARS.Analytics backend
8100+	Начальный порт CARS.API	Nginx
8000	CARS.Analytics backend	CARS.Analytics frontend, CARS.Stream, Пользователь (admin)
8080	CARS.Analytics frontend	Пользователь (UI)
1947	HASP	CARS.Stream
5432	Postgre SQL	CARS.Analytics backend
6379	Redis	CARS.Analytics backend

Примечание 2. История изменений.

Дата	Версия	Описание
05.08.2021	1.1	Полная ревизия и обновление документа
23.12.2020	1.0	Первичная версия документов