

Инструкция по инсталляции ПАУК Трафик

1. БАЗОВОЕ СИСТЕМНОЕ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Системные программные средства, для которых обеспечивается эффективная работа

Программы:

- ❑ процессорный модуль Jetson Orin Nano;
- ❑ процессор – шестиядерный 64-разрядный процессор NVIDIA Carmel с архитектурой ARM®v8.2 6 МБ L2 и 4 МБ L3
- ❑ графический процессор – архитектура NVIDIA Volta™ с 384 ядрами NVIDIA CUDA® и 48 тензорными ядрами
- ❑ оперативная память – 8 ГБ LPDDR4x 128-бит, 59,7 Гбит/с
- ❑ операционная система – Ubuntu 20.04.

Для имплементации ПО необходимо следующее программное обеспечение:

- ❑ GStreamer 1.16.3, пакет, используемое в ПО для трансляции видеопотока;
- ❑ Eigen 3.3.7, библиотека с эффективной реализацией работы с матрицами;
- ❑ OpenCV 4.5.5, библиотека с эффективной реализацией работы с изображениями и видеопотоками.
- ❑ Docker Engine 20.10.21, программное обеспечение для автоматизации развёртывания и управления приложениями в средах с поддержкой контейнеризации, контейнеризатор приложений.

2. ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Программа и все необходимые ей компоненты должны устанавливаться только на процессорный модуль Jetson Orin Nano.

Минимальные технические характеристики серверного оборудования:

- ❑ Шестиядерный 64-разрядный процессор с архитектурой ARM, 1.2 GHz;
- ❑ память 8GB LPDDR4x;
- ❑ графический процессор с 384 ядрами CUDA и 48 тензорными ядрами;
- ❑ свободное дисковое пространство 20GB (дополнительно пространство для размещения прикладных систем и баз данных).

3. УСТАНОВКА

Перед установкой убедитесь, что в системе установлены следующие пакеты:

- **libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev libgstreamer-plugins-bad1.0-dev gstreamer1.0-plugins-base gstreamer1.0-plugins-good gstreamer1.0-plugins-bad gstreamer1.0-plugins-ugly gstreamer1.0-libav gstreamer1.0-tools gstreamer1.0-x gstreamer1.0-alsa gstreamer1.0-gl gstreamer1.0-gtk3 gstreamer1.0-qt5 gstreamer1.0-pulseaudio**
- **libeigen3-dev**
- **python3-opencv**
- **docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin**

Все пакеты устанавливаются из стандартных репозиториях с помощью команды:

```
sudo apt install
```

Процедура установки Программы:

1. Скачать и разархивировать дистрибутив.
2. Перейти в директорию дистрибутива.
3. Загрузить докер-образ приложения: **docker load --input cartrack_2.tar**
4. Загрузить докер-образ nginx-сервера: **docker load --input nginx_car.tar**
5. Редактировать файл **docker-compose.yml**, в полях `cartrack_2-working_dir` и `cartrack_2-volumes` записать абсолютный путь до директории `cartrack_2` из дистрибутива.
6. Исполнить команду для настройки и запуска мультидокер приложения **docker-compose up -d**

Настройка конфигурационных файлов:

1. Необходимо ввести в браузер адрес подключения к webGUI по IP адресу (если настройка осуществляется не удаленно, то необходимо ввести 127.0.0.1:5000)
2. Ввести данные аутентификации, по умолчанию:
Username: admin
Password: 1z2v3e4r
3. Перейти в меню Add camera с помощью навигационной панели в верхнем правом углу и нажать на появившуюся кнопку Add camera.
4. Заполнить поля для загрузки разметки камеры, а именно имя камеры, адрес камеры, файл разметки (пример в файле `Файл_разметки_полос_движения_TC.json`) и файл СМС (опционально, пример в файле `Файл_разметки_смс.json`). После заполнения необходимых полей нажать на кнопку Add camera для добавления камеры, с которой будет работать ПО.
Примеры файлов доступны по ссылке: https://disk.yandex.ru/d/bkdPCBSzK_SmxA
5. Далее необходимо загрузить файлы полиномов скорости (пример для одной полосы в файле `Файл_полинома_скорости.json`) для каждой полосы движения (опционально).
Загрузка данных файлов требуется для вычисления скорости ТС.
Примеры файлов доступны по ссылке: https://disk.yandex.ru/d/bkdPCBSzK_SmxA
6. Для задачи новых параметров работы ПО необходимо перейти в меню Config с помощью навигационной панели в верхнем правом углу и изменить желаемые параметры:
dets_threshod (параметр для кол-ва детекций у трека для подсчета ТС)
density_time (параметр указывающий кол-во времени в секундах для которого будет считаться плотность движения)
draw_lines (флаг для отрисовки статистики по полосам вместо дорог)
draw_minizone (флаг для контроля и отрисовки минизон)
zone_verbose (флаг для отображения просчитанных коэффициентов)
overload_threshod (параметр характеризующий занятость зоны)
work_camera (обрабатываемые камеры)
fps (fps на камерах по умолчанию)
visualize (флаг для визуализации обработки)
vis_size (разрешение визуализации)
record (флаг для записи видео)
save_path (путь до места записи)
no_bd (флаг отключения базы данных)
no_vel (флаг отключения расчета скорости)
avg_vel (время для расчета средней скорости по полосе в минутах)
cpp_vis (флаг для использования c++ визуализации)
nc (флаг включения сохранения bbox'ов машин)
stream (тип транслируемого видеопотока (h264, mjpeg, hls))

7. При необходимости внести изменения в уже загруженные файлы разметки, нужно перейти в меню Admin с помощью навигационной панели в верхнем правом углу.