



# РУКОВОДСТВО ПО УСТАНОВКЕ

## **Программное обеспечение «Система поиска CVS КОЗ 1 Автономный поиск»**

### Аннотация:

В документе содержится информация о процессе установки программного обеспечения «Система поиска CVS КОЗ 1 Автономный поиск», а также пошаговые инструкции для успешного выполнения этой задачи

Листов 10  
Москва 2024

## СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ .....	3
ВВЕДЕНИЕ .....	4
1. Требования к программному обеспечению.....	5
2. Требования к аппаратному обеспечению .....	6
2.1. Технические характеристики Jetson Orin Nano 8GB .....	6
3. Установка и настройка .....	8
3.1. Описание окружения, зависимостей и внешних библиотек, необходимых ПО.....	8
3.2. Порядок установки зависимостей и внешних библиотек.....	8
3.3. Описание порядка подготовки Обучающего набора данных для создания файлов .....	9
3.4. Описание привлеченных сторонних библиотек.....	9
3.5. Описание создания файлов для запуска и использования системы .....	9

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

Сокращение	Детальная расшифровка
«Система»	Система поиска CVS КОЗ 1 Автономный поиск
ПО	Программное обеспечение
CPU	Central Processing Unit
GPU	Graphics Processing Unit
ОС	Операционная система

## ВВЕДЕНИЕ

В данном документе представлены подробные инструкции по установке, настройке и использованию «Системы поиска CVS КОЗ 1 Автономный поиск». Руководство охватывает все этапы процесса установки, начиная с подготовки системы и заканчивая проверкой работоспособности установленного ПО.

Для работы с системой необходимы технические специалисты с высшим образованием и квалификацией не ниже инженера.

## 1. Требования к программному обеспечению

Требования к программному обеспечению необходимые для запуска «Системы»:

- ОС Jetson Linux;
- Установленный пакет CMAKE для сборки ПО;

## 2. Требования к аппаратному обеспечению

Система оптимизирована для работы под Jetson Orin Nano 8GB. Также возможны работа на всех Jetson серии Orin с Jetpack6.

### 2.1. Технические характеристики Jetson Orin Nano 8GB

- Процессор: Шестиядерный процессор Arm Cortex-A78AE v8.2, 64-разрядный, с тактовой частотой 1,5 ГГц, 1,5 МБ L2 + 4 МБ L3
- Графический процессор: 1024-ядерной архитектурой NVIDIA Ampere и 32 тензорными ядрами
  - Производительность ИИ: 40 TOPS
  - Память: 8GB 128-bit LPDDR5, 68GB/s
  - Хранилище: Слот MicroSD
  - M.2 Key M для NVMe SSD
  - Вывод видео: DisplayPort 1.2 (+MST)
  - Интерфейс камеры: 2x MIPI CSI-2
  - Декодирование видео: K60 | 2x 4K30 | 5x 1080p60 | 11x 1080p30 (H.265)
  - Кодирование видео: 1080p30 поддерживается 1-2 ядрами ЦП (т. е. программным кодированием)
  - Сеть: Gigabit Ethernet
  - USB: 4x USB 3.2 Gen2
  - 1x устройство USB-C (также используется для отладки)
  - Расширение: Разъем M.2 Key E (PCIe x1, USB 2.0, UART, I2S и I2C)
  - 40-контактный разъем расширения с сигналами GPIO, I2C, I2S, SPI, UART
  - M.2 Key Разъем M (PCIe Gen3 x4)
  - M.2 Разъем Key M (PCIe Gen3 x2)

- Источник питания: 9–20 В (предусмотрено питание 19 В) через разъем постоянного тока
- Дополнительные разъемы PoE и резервного питания
- Режимы мощности: 7-15 Вт
- Охлаждение: Радиатор + вентилятор
- Размеры: 100 x 79 x 21 мм

### 3. Установка и настройка

#### 3.1. Описание окружения, зависимостей и внешних библиотек, необходимых ПО

Для запуска решения требуется устройство Jetson Orin Nano 8GB с установленным на него Jetpack 6. Запуск осуществляется в Docker-контейнере с образом `ultralytics/ultralytics:latest-jetson-jetpack6`, который содержит все необходимые зависимости:

- `Opencv 4.8.0-1-g6371ee1`
- `TensorRT 8.6.2.3-1+cuda12.2`
- `CUDA Toolkit 12.2.12-1`

#### 3.2. Порядок установки зависимостей и внешних библиотек

Установка дополнительных библиотек не требуется, поскольку все необходимые компоненты присутствуют в Docker-образе `ultralytics/ultralytics:latest-jetson-jetpack6`.

Для загрузки и запуска Docker-образа выполните следующие команды: программного обеспечения NAFT состоит из нескольких этапов:

- `#` Переменная с именем Docker-образа  
`t=ultralytics/ultralytics:latest-jetson-jetpack6`
- `#` Запуск Docker-контейнера с примонтированной текущей директорией

```
docker run -it --rm --ipc=host -v  
$(pwd) :/mnt --runtime=nvidia $t
```

Так же рекомендуется включить режим максимальной производительности, для ускорения образа нейронной сети следующими командами:

- Включение всех ядер центрального и графического процессоров.  
`sudo nvpmodel -m 0`

– Включение Jetson Clocks — это гарантирует, что все ядра ЦП и ГП будут работать на максимальной частоте.

```
sudo jetson_clocks
```

### **3.3. Описание порядка подготовки Обучающего набора данных для создания файлов**

Предобработка данных включала в себя следующие этапы:

- удаление дубликатов;
- фильтрация изображений, на которых не было людей;
- удаление слишком похожих изображений на основе их embedding-ов, полученных с помощью предобученной модели CLIP;
- ручная доразметка изображений и разделение на тренировочную и тестовую выборки.

### **3.4. Описание привлеченных сторонних библиотек**

В датасет для обучения модели для системы были включены:

- Датасет, собранный во время участия в конкурсе «Экстренный поиск» (испытания Сателлит №2 и финальные испытания)
- LADD (GNU General Public License v3.0) – <https://www.kaggle.com/datasets/mersico/lacmus-drone-dataset-ladd-v40>
- Heridal (Creative Commons Attribution 3.0 Unported License) – <http://ipsar.fesb.unist.hr/HERIDAL%20database.html>
- SARD (Creative Commons Attribution) – <https://ieee-dataport.org/documents/search-and-rescue-image-dataset-person-detection-sard>

### **3.5. Описание создания файлов для запуска и использования системы**

Для сборки и подготовки файлов выполните следующие шаги в командной строке, находясь в директории с исходными файлами решения:

– Запуск Docker-контейнера с примонтированной текущей директорией:

```
t=ultralytics/ultralytics:latest-jetson-jetpack6
docker run -it --rm --ipc=host -v
$(pwd) :/mnt --runtime=nvidia $t
```

– Установка CMake (необходим для сборки cvs\_cpp\_detector\_module):

```
apt update
apt install -y cmake
```

– Сборка модуля cvs\_cpp\_detector\_module:

```
cd /mnt/cvs_detector
mkdir build
cd build
cmake -DCMAKE_BUILD_TYPE=Release ..
make -j4 cvs_cpp_detector_module
```

– Копирование скомпилированного модуля в папку с решением:

```
cp cvs_cpp_detector_module.cpython-310-aarch64-linux-gnu.so
../../solution
```

– Переход в директорию с решением:

```
cd ../../solution
```

Далее запуск системы производится через модуль solution (solution.py) с функцией predict реализованной в соответствии с требованиями платформы.