

Инструкция по эксплуатации
программного обеспечения «Система поиска GigaFlex C1 Экстренный поиск»

Москва, 2023

Термины и определения

БВС (беспилотные воздушные суда): Автономные или дистанционно управляемые воздушные суда, не предназначенные для нахождения человека на борту во время полета.

Консоль: Интерфейс командной строки, позволяющий пользователю взаимодействовать с программным обеспечением без использования графического интерфейса.

Технологии искусственного интеллекта: Технологические решения и алгоритмы, имитирующие человеческий интеллект, используемые для автоматизации различных процессов, включая распознавание образов.

Linux (Ubuntu): Операционная система, основанная на ядре Linux и использующая рабочее окружение GNOME по умолчанию. Ubuntu 18.04 и выше обозначают конкретные версии этой операционной системы.

Conda: Система управления пакетами и средой, которая позволяет устанавливать, запускать и обновлять библиотеки и зависимости в различных языках программирования, таких как Python и R, в изолированных средах, называемых «кондой». Conda облегчает управление пакетами и зависимостями в рамках проектов, что способствует воспроизводимости и портативности кода.

Miniconda: Минимальная установка conda, которая включает только conda, Python, зависимости, которые используются conda, и несколько других полезных пакетов.

Python: Высокоуровневый язык программирования, ориентированный на повышение производительности разработчика и читаемости кода. Python поддерживает несколько парадигм программирования, включая структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование.

PyTurboJpeg: Библиотека для работы с изображениями в формате JPEG. PyTurboJpeg предоставляет Python-интерфейс для библиотеки TurboJPEG, что позволяет выполнять быстрое сжатие и декомпрессию изображений, а также обеспечивает поддержку работы с изображениями высокого разрешения.

Intersection over Union (IoU): Метрика, используемая для оценки точности алгоритмов объектного обнаружения, измеряющая степень перекрытия между предсказанным и реальным ограничивающим прямоугольником. Значения варьируются от 0 (отсутствие перекрытия) до 1 (полное совпадение).

Пороговое значение (Threshold): Значение, используемое для определения, должен ли объект быть классифицирован как интересующий нас объект или отброшен. Используется для управления уровнем уверенности алгоритма в правильности классификации объекта.

Ограничивающий прямоугольник (Bounding Box): Прямоугольник, который рисуется вокруг обнаруженного объекта на изображении, показывая расположение и размеры объекта.

1. Общие сведения

Данный документ является руководством по установке и эксплуатации программного обеспечения «Система поиска GigaFlex C1 Экстренный поиск» (далее – ПО). Все исключительные права на ПО принадлежат Фонду НТИ (далее – Компания).

ПО не имеет пользовательского графического интерфейса, управление осуществляется с использованием консоли.

ПО разработано с целью автоматизации процесса поиска людей в условиях природной местности.

ПО способно автоматически идентифицировать человеческие фигуры на изображениях, полученных после полета беспилотных воздушных судов (БВС), используя технологии искусственного интеллекта.

Эффективность распознавания оптимизирована для работы на высотах от 45 до 70 метров. Максимальная эффективность достигается при высоте полета близкой к 45 метрам.

2. Требования к программно-аппаратному обеспечению

2.1. Требования к программному обеспечению

Программное обеспечение «Система поиска GigaFlex C1 Экстренный поиск» разработано с учетом спецификации работы в операционных системах семейства Linux, начиная с версии Ubuntu 18.04 и выше. Для ПО необходимы следующие компоненты:

Зависимости:

- Python версии 3.10 и выше;
- PyTorch версии 2.1.0 и выше;
- PyTurboJpeg версии 1.7.2;
- Miniconda версии 23.5 и выше;
- требуется поддержка CUDA версии 11.8.

Для проверки наличия ПО необходимо выполнить следующую инструкцию:

1. Использовать комбинацию клавиш «Ctrl+Alt+T» для открытия командной строки
2. Для проверки версии операционной системы Linux, выполните команду:
«lsb_release -a»
3. Для проверки версии miniconda, выполните команду:
«conda —version»
4. Для активации среды, в которой будет запускаться ПО:
«conda activate название_среды»
5. Для проверки версии Python, выполните команду:
«python3 —version»
6. Для проверки версии PyTorch, выполните команду:
«python -c "import torch; print(torch.__version__)"
7. Для проверки наличия и версии PyTurboJpeg, выполните команду:
«python -c "import turbojpeg; print(turbojpeg.__version__)"»
8. Для проверки версии CUDA, выполните команду:
«nvcc -version»

2.2. Требования к аппаратному обеспечению

Для корректной работы программного обеспечения «Система поиска GigaFlex C1 Экстренный поиск» рекомендуется использование компьютера со следующими минимальными характеристиками:

Процессор: Intel® Core™ i9-13900HX с частотой 2.2-5.4GHz 24/32 или эквивалентный от AMD

Видеокарта: Nvidia® GeForce RTX™ 4090

Оперативная память: 32 GB

Свободное место на жестком диске: не менее 5 GB

3. Установка программы

Необходимо произвести загрузку ПО «Система поиска GigaFlex C1 Экстренный поиск» (*Программное обеспечение предоставляется безвозмездно для целевого использования некоммерческих организаций. Информация о предоставлении права использования по лицензионному договору может быть предоставлена по отдельному запросу через электронную почту ntifundsoft@nti.fund либо по номеру телефона +7 (495) 120-10-45*).

Для установки ПО необходимо выполнить следующую инструкцию.

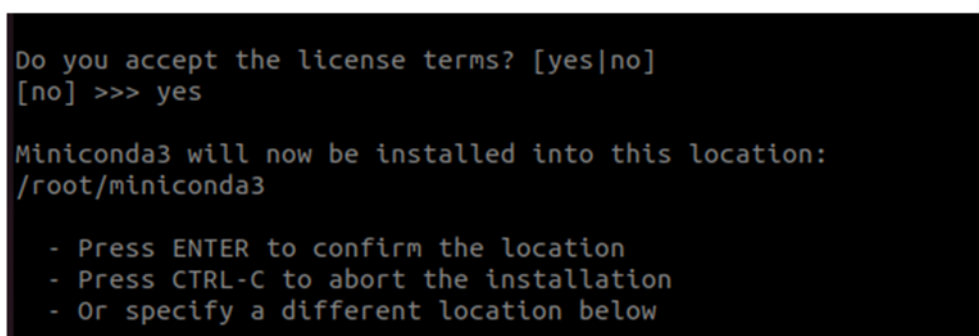
1. Установить miniconda, выполнив следующую инструкцию:

– использовать комбинацию клавиш «Ctrl+Alt+T» для открытия терминала Linux;

– в открытом терминале Linux выполнить команду: «`wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh`»;

– после окончания загрузки выполнить команду: «`sh Miniconda3-latest-Linux-x86_64.sh`»;

– с помощью нажатия клавиши «Enter» прочитать пользовательское соглашение до надписи на рисунке:



```
Do you accept the license terms? [yes|no]
[no] >>> yes

Miniconda3 will now be installed into this location:
/root/miniconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below
```

– после появления надписи ввести «yes», тем самым согласившись с условиями лицензионного соглашения;

– подтвердить выбор пути установки по умолчанию с помощью нажатия клавиши «Enter», после появления следующей надписи:

```
Miniconda3 will now be installed into this location:
/home/vasiliy/miniconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below
```

– инициализировать окружение через команду «conda init», с помощью ввода команды «yes» после появления следующей надписи:

```
Do you wish the installer to initialize Miniconda3
by running conda init? [yes|no]
[no] >>>
```

– закрыть терминал.

После завершения процесса установки miniconda можно проверить ее версию, выполнив команду «conda —version» в терминале Linux.

2. Установка CUDA:

Перед установкой CUDA требуется создать новое окружение. Для этого необходимо выполнить следующую инструкцию:

– использовать комбинацию клавиш «Ctrl+Alt+T» для открытия терминала Linux;

– в открытом терминале Linux выполнить команду: «conda create --name main python==3.10», где «main» — это название окружения, а «3.10» — это версия Python, которая будет использоваться в данном окружении;

– в процессе создания нового окружения в терминале Linux будет выведен текст, приведенный на рисунке, с просьбой подтвердить установку, в данном случае необходимо ввести команду «у» в окне терминала Linux;


```

root@3885ed7c3cbc:~/miniconda3# source /root/.bashrc
(base) root@3885ed7c3cbc:~/miniconda3# conda --version
conda 23.5.2
(base) root@3885ed7c3cbc:~/miniconda3# conda create --name main python==3.10
Collecting package metadata (current_repodata.json): done
Solving environment: unsuccessful attempt using repodata from current_repodata.json, retrying with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
current version: 23.5.2
latest version: 23.9.0

Please update conda by running

  $ conda update -n base -c defaults conda

Or to minimize the number of packages updated during conda update use

  conda install conda=23.9.0

## Package Plan ##

environment location: /root/miniconda3/envs/main

added / updated specs:
- python==3.10

The following packages will be downloaded:

package | build |
-----|-----|
ca-certificates-2023.08.22 | h06a4308_0 | 123 KB
libffi-3.3 | he6710b0_2 | 50 KB
openssl-1.1.1w | h7f8727e_0 | 3.7 MB
pip-23.3 | py310h06a4308_0 | 2.7 MB
python-3.10.0 | h12debd9_5 | 23.5 MB
setuptools-68.0.0 | py310h06a4308_0 | 936 KB
wheel-0.41.2 | py310h06a4308_0 | 109 KB
-----|-----|
Total: | 31.1 MB

The following NEW packages will be INSTALLED:

_libgcc_mutex pkgs/main/linux-64::_libgcc_mutex-0.1-main
_openmp_mutex pkgs/main/linux-64::_openmp_mutex-5.1-1_gnu
bzip2 pkgs/main/linux-64:bzip2-1.0.8-h7b6447c_0
ca-certificates pkgs/main/linux-64:ca-certificates-2023.08.22-h06a4308_0
ld_impl_linux-64 pkgs/main/linux-64:ld_impl_linux-64-2.38-h1181459_1
libffi pkgs/main/linux-64:libffi-3.3-he6710b0_2
libgcc-ng pkgs/main/linux-64:libgcc-ng-11.2.0-h1234567_1
libgomp pkgs/main/linux-64:libgomp-11.2.0-h1234567_1
libstdcxx-ng pkgs/main/linux-64:libstdcxx-ng-11.2.0-h1234567_1
libuuid pkgs/main/linux-64:libuuid-1.41.5-h5eee18b_0
ncurses pkgs/main/linux-64:ncurses-6.4-h6a678d5_0
openssl pkgs/main/linux-64:openssl-1.1.1w-h7f8727e_0
pip pkgs/main/linux-64:pip-23.3-py310h06a4308_0
python pkgs/main/linux-64:python-3.10.0-h12debd9_5
readline pkgs/main/linux-64:readline-8.2-h5eee18b_0
setuptools pkgs/main/linux-64:setuptools-68.0.0-py310h06a4308_0
sqlite pkgs/main/linux-64:sqlite-3.41.2-h5eee18b_0
tk pkgs/main/linux-64:tk-8.6.12-h1ccaba5_0
tzdata pkgs/main/noarch:tzdata-2023c-h04d1e81_0
wheel pkgs/main/linux-64:wheel-0.41.2-py310h06a4308_0
xz pkgs/main/linux-64:xz-5.4.2-h5eee18b_0
zlib pkgs/main/linux-64:zlib-1.2.13-h5eee18b_0

Proceed ([y]/n)?

```

- дождаться завершения загрузки и извлечения пакетов;
- активировать новое окружение, выполнив в терминале команду: «conda activate main», где «main» — это название окружения, выбранное пользователем в процессе создания окружения;

– установить PyTorch, который установит в том числе CUDA требуемой версии, выполнив в терминале Linux команду: «conda install pytorch torchvision pytorch-cuda=11.8 -c pytorch -c nvidia»;

– подтвердить установку, введя команду «у», после появления запроса в терминале; пример запроса приведен на рисунке:

```
nettle                pkgs/main/linux-64::nettle-3.7.3-hbbd107a_1
networkx              pkgs/main/linux-64::networkx-3.1-py310h06a4308_0
numpy                 pkgs/main/linux-64::numpy-1.24.3-py310hd5efca6_0
numpy-base           pkgs/main/linux-64::numpy-base-1.24.3-py310h8e6c178_0
openh264              pkgs/main/linux-64::openh264-2.1.1-h4ff587b_0
openjpeg              pkgs/main/linux-64::openjpeg-2.4.0-h3ad879b_0
pillow                pkgs/main/linux-64::pillow-10.0.1-py310ha6cbd5a_0
pyscparser            pkgs/main/noarch::pyscparser-2.21-pyhd3eb1b0_0
pyopenssl             pkgs/main/linux-64::pyopenssl-23.2.0-py310h06a4308_0
pysocks               pkgs/main/linux-64::pysocks-1.7.1-py310h06a4308_0
pytorch               pytorch/linux-64::pytorch-2.1.0-py3.10_cuda11.8_cudnn8.7.0_0
pytorch-cuda          pytorch/linux-64::pytorch-cuda-11.8-h7e8668a_5
pytorch-mutex         pytorch/noarch::pytorch-mutex-1.0-cuda
pyyaml                pkgs/main/linux-64::pyyaml-6.0-py310h5eee18b_1
requests              pkgs/main/linux-64::requests-2.31.0-py310h06a4308_0
six                   pkgs/main/noarch::six-1.16.0-pyhd3eb1b0_1
sympy                 pkgs/main/linux-64::sympy-1.11.1-py310h06a4308_0
torchtriton           pytorch/linux-64::torchtriton-2.1.0-py310
torchvision           pytorch/linux-64::torchvision-0.16.0-py310_cu118
typing_extensions     pkgs/main/linux-64::typing_extensions-4.7.1-py310h06a4308_0
urllib3               pkgs/main/linux-64::urllib3-1.26.16-py310h06a4308_0
yaml                  pkgs/main/linux-64::yaml-0.2.5-h7b6447c_0
zstd                  pkgs/main/linux-64::zstd-1.5.5-hc292b87_0

Proceed ([y]/n)? 
```

– дождаться завершения загрузки и извлечения пакетов;

– закрыть терминал.

3. Установка PyTurboJpeg:

Перед установкой библиотеки PyTurboJpeg, необходимо установить требуемые библиотекой зависимости. Для этого необходимо выполнить следующую инструкцию:

– использовать комбинацию клавиш «Ctrl+Alt+T» для открытия терминала Linux;

– в открытом терминале Linux выполнить команду: «sudo apt-get update && apt-get install libturbojpeg –у»;

– дождаться завершения загрузки и извлечения пакетов.

После установки зависимостей, необходимо установить библиотеку PyTurboJpeg. Для этого необходимо в открытом терминале Linux выполнить команду: «`pip install -U git+https://github.com/lilohuang/PyTurboJPEG.git`», и дождаться завершения загрузки.

4. Подготовка ПО:

После загрузки ПО «Система поиска GigaFlex C1 Экстренный поиск» необходимо разархивируем файл, выполнив в терминале Linux команду: «`unzip Gigaflex_sattelit1_inference.zip`».

После завершения разархивирования ПО готово к использованию. Для тестового запуска вводим команду «`python main.py /home/user/findpeople/images output.csv`», где «`/home/user/findpeople/images`» — путь до изображений, которые подлежат обработке, а «`output.csv`» — название выходного файла, в котором будет содержаться детализированная информация о найденных людях на соответствующем кадре. Файл структурирован в формате CSV (Comma-Separated Values) и его структура описана в п.6 «Завершение программы»

4. Запуск и выполнение программы

Перед выполнением программы в файле «part_inference.py» можно поменять следующие параметры:

– IOU_THRES: параметр относится к метрике Intersection over Union (IoU), которая обычно используется для оценки точности алгоритмов объектного обнаружения. IoU измеряет степень перекрытия между предсказанным ограничивающим прямоугольником и реальным ограничивающим прямоугольником, и его значения варьируются в диапазоне от 0 до 1, где 0 означает отсутствие перекрытия, а 1 – полное совпадение. Значение по умолчанию – 0.2;

– GLOBAL_THRESHOLD: общий порог, который определяет, будет ли обнаруженный объект считаться телом или нет, на основе уверенности алгоритма в своем предсказании. Параметр используется для фильтрации результатов, чтобы уменьшить количество ложных срабатываний. Объекты, уверенность в обнаружении которых ниже этого порога, будут отфильтрованы. Увеличение этого значения приведет к тому, что только объекты, в которых алгоритм уверен на заданный порог или более, будут сохранены в результатах. Уменьшение значения увеличит количество детектированных объектов, но также может увеличить количество ложных срабатываний. Значение по умолчанию – 0.9.

Для запуска вводим команду: «python main.py /home/user/findpeople/images output.csv», где «/home/user/findpeople/images» – путь до изображений, которые подлежат обработке, а «output.csv» – название выходного файла, в котором будет содержаться детализированная информация о найденных людях на соответствующем кадре.

Во время исполнения, программа будет предоставлять следующую информацию в консоль:

– Количество изображений и подпроцессов: перед началом обработки изображений программа выведет в консоль два числа, разделенных пробелом. Первое число указывает на общее количество изображений, которые будут

обработаны. Второе число обозначает количество подпроцессов, которые будут задействованы в процессе обработки. Это позволяет пользователям иметь представление о масштабе обработки и распределении ресурсов для выполнения задачи. Пример вывода: «100 4», где 100 – количество изображений, а 4 – количество сабпроцессов.

– Время обработки каждого изображения: после обработки каждого индивидуального изображения, программа будет выводить в консоль время, потраченное на его обработку, измеряемое в секундах. Это позволяет пользователям отслеживать производительность программы и эффективность обработки изображений в реальном времени. Пример вывода одного из запусков показан ниже:

```
(main) shkeep@DmitryPC:~/tmp/Gigaflex_satellite_inference$ python main.py ~/datasets/FindPeople/test_data/test_data/images/ output.csv
10 2
5.2229156494140625
5.274887323379517
6.8299782276153564
7.000067472457886
2.801920175552368
2.7475028038024902
2.8732147216796875
2.782987356185913
2.8386459350585938
2.8647422790527344
```

5. Завершение программы

Если программа успешно обработает все входные изображения без ошибок или сбоев, дополнительных сообщений или ошибок выводиться не будет. Количество выведенных в консоль временных замеров будет соответствовать количеству обработанных изображений. Это служит индикатором того, что все изображения были успешно обработаны, и программа завершила свою работу корректно.

В результате работы программы будет сгенерирован файл, название которого было указано вторым аргументом к запуску программы. В данном файле содержится детализированная информация о найденных людях на соответствующем кадре. Файл структурирован в формате CSV (Comma-Separated Values) и включает в себя следующие столбцы:

- image_id: Идентификатор изображения.
- xc: X-координата центра обнаруженного объекта, нормализованная относительно ширины изображения.
- yc: Y-координата центра обнаруженного объекта, нормализованная относительно высоты изображения.
- w: Ширина обнаруженного объекта, нормализованная относительно ширины изображения.
- h: Высота обнаруженного объекта, нормализованная относительно высоты изображения.
- label: Метка класса обнаруженного объекта. В данной системе используется только единственная метка. Метка «0» присваивается в случае обнаружения человека на изображении.
- score: Вероятность правильности обнаружения объекта.

Пример строки из выходного файла:

```
«DJI_EXAMPLE,0.6313326286547112,0.5802247883456301,0.0252808541962594,0.0201106828433076,0,0.9501953»
```

Для связи с разработчиками писать на почту ntifundsoft@nti.fund