

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
(НИУ ВШЭ)

Программное обеспечение
«ФРЕЙМВОРК АВТОМАТИЗАЦИИ СИНТЕЗА МОДЕЛЕЙ
КОМПЬЮТЕРНОГО ЗРЕНИЯ AutoOD»

Инструкция по установке

на 8 листах

Москва 2024

ОГЛАВЛЕНИЕ

1. Общие сведения.....	3
1.1 Назначение и функции программы.....	3
1.2 Сведения о технических и программных средствах	3
2. Настройка программы.....	4
2.1 Подготовка настройки программы	4
2.2 Состав и структура дистрибутива.....	4
3. Порядок установки ПО	6
3.1 Установка зависимостей	6
3.2 Получение дистрибутива	6
3.3 Установка ПО.....	7
4. Проверка программы.....	8
4.1 Описание способа проверки корректности установки	8

1. Общие сведения

1.1 Назначение и функции программы

Программное обеспечение «Фреймворк автоматизации синтеза моделей компьютерного зрения AutoOD» предназначено для синтеза моделей компьютерного зрения с использованием нейросетевых моделей глубинного обучения разных типов архитектур в целях применения пользователем, не обладающим глубокими навыками машинного обучения и анализа данных и/или не имеющего опыта применения алгоритмов искусственного интеллекта.

1.2 Сведения о технических и программных средствах

Для установки, запуска и стабильного функционирования программы требуется сервер со следующими минимальными системными требованиями (в случае разворачивания ПО на одном сервере):

- Количество физических ядер процессора (CPU) - 4;
- Оперативная память (RAM) - 32 Гб;
- Дисковое пространство (HDD) – 256 Гб;
- GPU Nvidia с поддержкой CUDA, объём видеопамяти - не менее 8 гигабайт;
- Сервер должен быть подключен к ИБП.

Окружение для штатной работы веб-сервера Python должно включать:

- Операционная система Ubuntu версии 22.XX и выше;
- Интерпретатор языка Python версии не старше 3.9
- Модули языка Python:
 - openmim
 - mmengine \geq 0.7.0
 - ninja
 - psutil
 - pytorch
 - torchvision
 - mmdetection
 - norfair \geq v2.2.0

2. Настройка программы

2.1 Подготовка настройки программы

Необходимо установить дополнительное программное обеспечение на сервере в соответствии с указаниями в Таблице 1:

Таблица 1 – Дополнительное программное обеспечение

Наименование ПО	Описание
Python 3.9	Интерпретатор языка программирования Python
openmim	Менеджер пакетов для фреймворка машинного зрения OpenMMLab

2.2 Состав и структура дистрибутива

- Директория «auto_od»;
- Директория «dist»;
- Директория «docs»;
- Директория «tests».
- Dockerfile
- README.md
- requirements.txt
- settings.yaml
- setup.py

Директория «auto_od» содержит исходные коды ПО.

Директория «dist» содержит служебные файлы, необходимые для установки ПО как модуля языка Python.

Директория «docs» содержит документацию, сгенерированную инструментом Sphinx.

Директория «tests» содержит исходные коды автоматических тестов компонентов ПО.

Файл Dockerfile содержит инструкции для сборки образа системы контейнеризации Docker с окружением для использования ПО.

Файл README.md содержит базовые инструкции по эксплуатации ПО.

Файл requirements.txt содержит список зависимостей - модулей языка Python, необходимых для эксплуатации ПО.

Файлы settings.yaml содержит пример базовой конфигурации фреймворка MMDetection.

Файл setup.py является служебным файлом модуля языка Python, необходимым для установки модуля в локальное окружение.

3. Порядок установки ПО

3.1 Установка зависимостей

1. Установить интерпретатор Python версии не старше 3.9.
2. Установить менеджер пакетов фреймворка OpenMMLab `openmim`.

```
pip install -U openmim
```

Рисунок 1 – Команда установки менеджера пакетов `mim`

3. Установить пакет `mmengine`.

```
mim install "mmengine>=0.7.0"
```

Рисунок 2 – Команда установки пакета `mmengine`

4. Установить пакет `mmdcv`.

```
mim install "mmdcv>=2.0.0"
```

Рисунок 3 – Команда установки пакета `mmdcv`

5. Установить пакет `mmdet`.

```
mim install mmdet
```

Рисунок 4 – Команда установки пакета `mmdet`

6. Проверка корректность установки пакета MMDetection (`mmdet`).
 - a. Загрузка файла весов и конфигурации моделей из хранилища предобученных моделей фреймворка MMDetection.

```
mim download mmdet --config rtmDET_tiny_8xb32-300e_coco -dest
```

- b. Вставьте следующий код в скрипт на языке Python и выполните.

```
from mmdet.apis import init_detector, inference_detector
config_file = 'rtmDET_tiny_8xb32-300e_coco.py'
checkpoint_file = 'rtmDET_tiny_8xb32-300e_coco_20220902_112414-78e30dcc.pth'
model = init_detector(config_file, checkpoint_file,
device='cpu') # или device='cuda:0'
inference_detector(model, 'demo/demo.jpg')
```

3.2 Получение дистрибутива

Получение дистрибутива ПО возможно в двух формах.

- 1) ZIP-архив.
- 2) Скачивание из git-репозитория.

В случае получения ZIP-архива необходимо распаковать его содержимое в любую локальную директорию. В случае получения ПО через git-репозиторий дополнительных действий не требуется.

3.3 Установка ПО

После скачивания или распаковки необходимо выполнить следующую команду для установки ПО в директории, куда был скачан или распакован репозиторий ПО.

```
python setup.py install
```

4. Проверка программы

4.1 Описание способа проверки корректности установки

После установки ПО можно проверить корректность установки выполнением следующего кода в интерпретаторе языка Python.

```
from auto_od.check.env import EnvironmentCollector
env_collector = EnvironmentCollector()
env_collector.display_env_info()
```