

# MAN TruckScenes

---

## Overview

- [Supplementary Material Content](#)
- [Download dataset](#)
- [Setup environment](#)
- [Install devkit](#)
- [Explore TruckScenes](#)
- [Train model](#)
- [Evaluate model](#)

## Supplementary Material Content

The supplementary material includes the following content.

1. code

Source code of the man-devkit and mmdetection3d framework. The devkit is used to interact with the dataset and provides a simple tutorial to get familiar with TruckScenes. The mmdetection3d framework is used to train and evaluate the baseline model.

2. data

The data folder includes instructions on how to download the MAN TruckScenes dataset. It also includes a short description of the hosting, licensing, and maintenance plan. Detailed information on these points can be found in the dataset documentation.

3. documentation

The documentation folder holds the required dataset documentation in form of [data cards](#).

4. specification

Specification includes annotation instructions and dataset specifications as well as all sensor data sheets of our sensor setup.

## Download dataset

The full dataset will soon be available on our [Website](#) and the [AWS Open Data Registry](#).

In the meantime we provide a mini version of the dataset via LRZ Sync+Share.

1. Download the mini dataset by using the provided link and credentials.

Dataset link: <https://syncandshare.lrz.de/getlink/fiAUuVVuTSEab8fP9JTgyQ/manscenes.zip>

Password: eT7959i4eSfG

The mini dataset will be available on LRZ Sync+Share during the review phase. Afterwards, we refer to our [Website](#) or the [AWS Open Data Registry](#) to download MAN TruckScenes. Therefore, just search for TruckScenes in the registry and download the provided archives.

2. Extract all files to `/data/manscenes`.

Eventually you should have the following folder structure:

```
/data/manscenes
  samples - Sensor data for keyframes.
  sweeps  - Sensor data for intermediate frames.
  v1.0-mini - JSON tables that include all the meta data and
  annotations.
```

## Setup environment

Working with the dataset requires a Python environment. To install Python, please check [here](#). We recommend either Python 3.6 or Python 3.8.

If you like, you can setup a virtual environment like so:

1. Create a virtual environment

```
python -m venv venv
```

2. Activate the virtual environment  
POSIX (bash)

```
source venv/bin/activate
```

Windows (cmd)

```
venv\Scripts\activate.bat
```

Windows (PowerShell)

```
venv\Scripts\Activate.ps1
```

## Install devkit

The MAN devkit is used to interact with the TruckScenes dataset. The installation of the devkit is done via pip.

1. Extract the devkit to `<path to man-devkit>`
2. Change directory to the devkit directory

```
cd <path to man-devkit>
```

### 3. Install the devkit via pip

```
python -m pip install -e .
```

Note: Microsoft Visual c++ 14.0 or greater is required under Windows. To install, please check [here](#).

If your Python installation does not include pip, please check [here](#).

## Explore TruckScenes

To get yourself familiar with TruckScenes, we recommend the tutorial in the devkit.

### 1. Install jupyter notebook

```
python -m pip install notebook
```

### 2. Start the tutorial and change the `dataroot` path to `/data/manscenes` (or your individual `<path to data folder>`).

```
jupyter notebook ./python-sdk/tutorials/manscenes_tutorial.ipynb
```

The tutorial should now launch in your browser.

## Train model

If you want to train the baseline model, please follow the following steps. Be aware that this procedure is based on docker, requires a GPU and is only tested under Ubuntu. To install docker on your machine, please check [here](#).

### 1. Extract MMDetection3D to `<path to mmdetection3d>`

### 2. Change directory to the mmdetection3d directory

```
cd <path to mmdetection3d>
```

### 3. Build docker image

```
docker build --no-cache -t mmdetection3d:man docker/
```

### 4. Run docker container (insert your devkit, mmdetection3d, and data path)

```
docker run \  
  --name mmdet \  
  --gpus all \  
  --shm-size=8g \  
  -it \  
  --mount 'type=bind,src=<path to man-devkit>,dst=/man-devkit' \  
  --mount 'type=bind,src=<path to mmdetection3d>,dst=/mmdetection3d' \  
  \  
  -v <path to data folder>:/mmdetection3d/data mmdetection3d:man
```

## 5. Install devkit in container

```
python -m pip install -e /man-devkit/
```

## 6. Create data for the mini dataset

```
python tools/create_data.py manscenes \  
  --root-path ./data/manscenes \  
  --out-dir ./data/manscenes \  
  --extra-tag manscenes \  
  --max-sweeps 18 \  
  --version v1.0-mini
```

## 7. Train model on the mini dataset

```
python tools/train.py \  
  /mmdetection3d/configs/centerpoint/centerpoint_voxel01_second_secfn_head-dcn-circlenms_8xb4-cyclic-20e_man-3d.py \  
  --cfg-options val_dataloader.dataset.metainfo.version=v1.0-mini
```

## 8. Test model on the mini dataset

```
python tools/test.py \  
  /mmdetection3d/configs/centerpoint/centerpoint_voxel01_second_secfn_head-dcn-circlenms_8xb4-cyclic-20e_man-3d.py \  
  work_dirs/centerpoint_voxel01_second_secfn_head-dcn-circlenms_8xb4-cyclic-20e_man-3d/epoch_1.pth \  
  --cfg-options test_evaluator.jsonfile_prefix=work_dirs/cp-man-test/results_eval \  
  --cfg-options test_dataloader.dataset.metainfo.version=v1.0-mini
```

## Evaluate model

The evaluation of the detection performance on the mini dataset is done via

```
python -m manscenes.eval.detection.evaluate \
    /mmdetection3d/work_dirs/cp-man-
test/results_eval/pred_instances_3d/results_mansc.json \
    --output_dir /mmdetection3d/work_dirs/cp-man-test/ \
    --dataroot /mmdetection3d/data/manscenes/ \
    --eval_set mini_val \
    --version v1.0-mini \
    --evaluate_tags 1
```

## Author Statement

"The authors bear responsibility for the text and images published in the paper, subject to the CC-BY-SA 4.0 License, under which both the paper and dataset are made available."