



OAX7700 CarSentry User Guide

v0.2.0, Jul, 2025

Table of Contents

- 1. Overview 2
- 2. Prerequisites 3
 - 2.1. Platform 3
 - 2.2. Quartz Environment Setup 3
- 3. Simulation Execution 4
 - 3.1. Input Data Preparation 4
 - 3.2. Simulator Execution 4
 - 3.3. Output Data 4
- 4. How to configure the modules 6
 - 4.1. ISP 6
 - 4.2. OVTPD (Human Detection) 6
 - 4.3. MD 6
 - 4.4. Skip some steps with existing results 7

Copyright © 2025 OmniVision Technologies, Inc. All rights reserved.

This document is provided "as is" with no warranties whatsoever, including any warranty of merchantability, non-infringement, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample.

OmniVision Technologies, Inc. and all its affiliates disclaim all liability, including liability for infringement of any proprietary rights, relating to the use of information in this document. No license, expressed or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

The information contained in this document is considered proprietary to OmniVision Technologies, Inc. and all its affiliates. This information may be distributed to individuals or organizations authorized by OmniVision Technologies, Inc. to receive said information. Individuals and/or organizations are not allowed to re-distribute said information.

Trademark Information

OmniVision and the OmniVision logo are registered trademarks of OmniVision Technologies, Inc. All other trademarks used herein are the property of their respective owners.

AI Camera

OAX7700 CarSentry User Guide

v0.2.0

Jul, 2025

Chapter 1. Overview

This document is for OAX7700 CarSentry Simulator User Guide. It provides information on how to use the OAX7700 carsentry simulator and the various modules in the simulator.

Chapter 2. Prerequisites

Prior to utilizing the OAX7700 carsentry simulator, ensure your platform meets system requirements, all prerequisite packages are installed and the python virtual environment are activated.

2.1. Platform

- Windows 10 or above with Python 3.9
- Linux ubuntu 22.04 with Python 3.9

2.2. Quartz Environment Setup

- Install the Quartz environment:

```
# Use the OV CDK2 package  
package: cdk2_24.08
```

Refer to the complete installation guide at: cdk2_24.08/docs/html/quartz/INSTALL.html

- Post-installation checks:
 - Verify console output for errors
 - Confirm successful installation
- Please make sure to activate the environment after installation, and run the following steps on it.

Chapter 3. Simulation Execution

3.1. Input Data Preparation

1. Copy the test_sequence.zip file to the `OA7700_sim/dataset/` directory.
2. Extract the zip file to this `dataset` directory, and the dataset directory is as follows:

```
OA7700_sim/dataset/test_sequence/
├── bcap20250606_150516_LeftCAM_Person_0.5m
│   ├── bcap1.raw
│   ├── bcap2.raw
│   ├── bcap3.raw
│   ├── ...
│   └── bcap50.raw
├── bcap20250626_151229_RightCAM_Item651_Bicycle_1.0m
└── ...
```

3.2. Simulator Execution

- Execute the simulator:

```
cd OA7700_Sim/app/oa7700_carsentry/
python ../qa/run_qa.py -c sim_cfg.json -p filelist=testseq.csv
```

3.3. Output Data

The simulation generates outputs in the following directory hierarchy:

```
output/
├── _workdir/                # Temporary working directory
├── YUV_original_isp/        # Raw ISP outputs
│   ├── *.yuv                # 960x768 YUV (P422)
│   └── *.yuv.json           # ISP metadata
├── YUV_256x208/             # Scaled video (256x208)
│   ├── *_256x208_P422.yuv
│   └── *.yuv.json
├── YUV_480x384/             # Scaled video (480x384)
│   ├── *_480x384_P422.yuv
│   └── *.yuv.json
└── ...
```

```

├── YUV2RGB_256x208/           # Color conversion
│   ├── *.rgb
│   └── *.rgb.json
├── OVTPD/                     # Object detection
│   ├── *.mp4                 # Annotated videos
│   └── *.json                # Detection metadata
├── MD/                         # Motion detection
│   ├── *.yuv                 # Processed video
│   ├── *.json
│   ├── md_data/              # Analytics data
│   ├── bitmap_1m.bin         # 1-meter bitmap
│   └── bitmap_0.5m.bin       # 0.5-meter bitmap
├── CS_PP/                     # Post-processing
│   └── *.json                # Final metadata
└── CS_QA/                     # Quality assurance
    ├── *.mp4                 # QA videos
    └── *.json                # QA metrics

```

== Final QA result for list ==

```

MD: Total 450.0, TN 313.0 TP 101.0 FN 5.0 FP 31.0, TP_roi 66.3014
MD: Precision 0.7652, Recall 0.9528, Accuracy 0.9200, F1-Score 0.8487
MD ROI: Precision 0.5023, Recall 0.6255, Accuracy 0.8429, F1-Score 0.5572
OD: Total 450.0, TN 344.0 TP 77.0 FN 29.0 FP 1.0, TP_roi 59.0320
OD: Precision 0.9872, Recall 0.7264, Accuracy 0.9335, F1-Score 0.8370
OD ROI: Precision 0.7568, Recall 0.5569, Accuracy 0.8936, F1-Score 0.6417
CS: Total 450.0, TN 343.0 TP 106.0 FN 0.0 FP 1.0
CS: Precision 0.9907, Recall 1.0000, Accuracy 0.9978, F1-Score 0.9953

```

Chapter 4. How to configure the modules

4.1. ISP

- Configuration files location: `OA7700_Sim/app/oa7700_carsentry/config/isp/`
 - `isp_para.txt` - Daytime mode configuration
 - `isp_para_ll.txt` - Low-light mode configuration
- Configuration Method:
 - Command Line Configuration:

```
python ../qa/run_qa.py -c sim_cfg.json -p  
filelist=testseq.csv,isp.param.lp=config/isp/isp_para_ll.txt
```

4.2. OVTPD (Human Detection)

- Available Model:
 - `208x256.c1q2`
- Configuration Method:
 - Command Line Configuration:

```
#Modify the confidence threshold  
python ../qa/run_qa.py -c sim_cfg.json -p  
filelist=testseq.csv,ovtpd.param.conf_thresh=0.8
```

4.3. MD

- Configuration Files Location: `OA7700_Sim/app/oa7700_carsentry/config/md/`
 - `default_8100.cfg` - Daytime mode
 - `default_carsentry_lowlight.cfg` - Low-light mode
 - `md_parameters_configuration.md` - Tuning guide
- Configuration Method:
 - Command Line Configuration:

```
python ../qa/run_qa.py -c sim_cfg.json -p  
filelist=testseq.csv,md.param.d=config/md/default_carsentry_lowlight.cfg
```


4.4. Skip some steps with existing results

- Skipping unchanged modules significantly improves execution speed
- In `sim_cfg.json`, for the 'force' key in each module
 - `force=0`: to skip the module (default or unchanged module)
 - `force=1`: to force run the module (required when:
 - Changing module cfg, parameters
 - Using different input files
 - Need fresh output)
- Configuration Method:
 - Command Line Configuration:

```
#'force' to run module. Take ISP and OVTPD module as an example
python ../qa/run_qa.py -c sim_cfg.json -p
filelist=testseq.csv,isp.force=1,ovtpd.force=1
```

```
#'force' to skip module. Take ISP, yuvscale, yuv2rgb, yscale, and MD modules as
an example
python ../qa/run_qa.py -c sim_cfg.json -p
filelist=testseq.csv,isp.force=0,yuvscale.force=0,yuv2rgb.force=0,yscale.force=0
,md.force=0
```