

The Tsinghua-Daimler Cyclist Benchmark (TDCB)

X. Li, F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, K. Li and D. M. Gavrila

1) Contact

Please feel free to contact us with any questions, suggestions or comments:

Fabian Flohr (Daimler AG, Dep. of Environment Perception)
Email: [tdcb at fabian-flohr.de]

2) Citation

When using the dataset please add the following citation:

X. Li, F. Flohr, Y. Yang, H. Xiong, M. Braun, S. Pan, K. Li and D. M. Gavrila,
"A new benchmark for vision-based cyclist detection",
In proceedings of IEEE Intelligent Vehicles Symposium (IV), pages 1028-1033, June 2016

3) License agreement

This dataset is made freely available for non-commercial purposes such as academic research, teaching, scientific publications, or personal experimentation. Permission is granted to use, copy, and distribute the data given that you agree:

1. That the dataset comes "AS IS", without express or implied warranty. Although every effort has been made to ensure accuracy, Daimler (or the website host) does not accept any responsibility for errors or omissions.
2. That you include a reference to the above publication in any published work that makes use of the dataset.
3. That if you have altered the content of the dataset or created derivative work, prominent notices are made so that any recipients know that they are not receiving the original data.
4. That you may not use or distribute the dataset or any derivative work for commercial purposes as, for example, licensing or selling the data, or using the data with a purpose to procure a commercial gain.
5. That this original license notice is retained with all copies or derivatives of the dataset.
6. That all rights not expressly granted to you are reserved by Daimler.

4) General - The Tsinghua-Daimler Cyclist Benchmark

The Tsinghua-Daimler Cyclist Benchmark provides a benchmark dataset for cyclist detection. Bounding Box based labels are provided for the classes: ("pedestrian", "cyclist", "motorcyclist", "tricyclist", "wheelchairuser", "mopedrider")

The dataset consist of 4 subsets:

Train

Usually used for training, contains 9741 images with annotations only for "cyclist". Only cyclists which are fully visible (occlusion<10%) and higher than 60 pixels have been labeled here.

Valid

1019 images to be used for validation of hyper parameters. Annotations for ("pedestrian", "cyclist", "motorcyclist", "tricyclist", "wheelchairuser", "mopedrider"). Only objects higher than 20 pixels have been labeled here.

Test

2914 images normally used for testing with annotations for ("pedestrian", "cyclist", "motorcyclist", "tricyclist", "wheelchairuser", "mopedrider"). Only objects higher than 20 pixels have been labeled here.

NonVRU

1000 images in which **no** object of interest ("pedestrian", "cyclist", "motorcyclist", "tricyclist", "wheelchairuser", "mopedrider") is present.

We provide demo videos with annotation examples. Maybe these videos help to decide if this dataset is appropriate for you:

Demo data type	Link
leftImg8bit/ test	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_demo_leftImg8bit_test.avi
leftImg8bit/ train	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_demo_leftImg8bit_train.avi
disparity/ test	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_demo_disparity_test.avi
disparity/ train	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_demo_disparity_train.avi

Feel free to use this dataset for academic research. See the licence for further information. Please add the above mentioned citation to your publication.

If you detect any error in the dataset (missing images, wrong labels etc.) please contact us so that we can improve this dataset. Thank you.

5) Setup

Manually

Download all of the following files and install them into the **same** directory:

Dataset part	Link	MD5 hash sum
labelData/ train	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_labelData_train.tar.gz	7e17bca380056f d75e6e586ef0ed 598a
labelData/ test	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_labelData_test.tar.gz	cf63c6d0485fb7a 0c8e7bd470f88c d66
labelData/ valid	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_labelData_valid.tar.gz	2adb1ffc2f2b289 108f1dfbae27fa2 c4
labelData/ NonVRU	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_labelData_NonVRU.tar.gz	d589c5e2729972 1b69d63cd3face 88d0
camera/ train	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_camera_train.tar.gz	1925b6b9897b28 65656cd00ba830 c4c4
camera/ test	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_camera_test.tar.gz	1434097ea7daeb 22e64c15a54cd1 02c6
camera/ valid	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_camera_valid.tar.gz	d0ae2e6d31b678 80c0529ce97720 e396
camera/ NonVRU	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_camera_NonVRU.tar.gz	ce67797cc6bb48 78c621143b4747 69b9
leftImg8bit/ train	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_leftImg8bit_train.tar.gz	03a0c18fd281ea 770244d094936e edfe

leftImg8bit/ test	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_leftImg8bit_test.tar.gz	82fc1f7983ab452 bd0b1a5c874d04 e89
leftImg8bit/ valid	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_leftImg8bit_valid.tar.gz	f08c9fc63173b47 fa0716e619e3b5 a55
leftImg8bit/ NonVRU	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_leftImg8bit_NonVRU.tar.gz	64d76ee7c65f5a e7f8bb0b790e19 84ec
disparity/ train	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_disparity_train.tar.gz	2f4b2cd11d5066 b6751a6b112d2a 7405
disparity/ test	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_disparity_test.tar.gz	d72375f453c83b ce522edf78cfab9 aa6
disparity/ valid	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_disparity_valid.tar.gz	6366b94297a0ef d2e1e1ab177a14 f1fd
disparity/ NonVRU	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_disparity_NonVRU.tar.gz	b063cfd2670ca5 22d172b24e993c 1e61
scripts (git)	http://www.gavrila.net/data/Daimler/iv16-li-flohr-gavrila-etal/tdcb_scripts.tar.gz	No hashsum because of git repo

md5sum [file]

By script (beta!)

We provide a setup script for the data in

`scripts/setup.py`

This script downloads and unzips the data in a specified folder.

Execute

```
python setup.py -o /home/user/exampleDatasetRoot
```

to download tar.gz into /home/user/exampleDatasetRoot and install it in the same folder.

Execute

```
python setup.py -o /home/user/exampleDatasetRoot -t /home/user/tars
```

to download tar.gz into /home/user/tars and install it in the folder

/home/user/exampleDatasetRoot.

6) Dataset Structure

The folder structure of the Tsinghua-Daimler Cyclist Benchmark is organized as

```
{root}/{type}/{split}/tsinghuaDaimlerDataset/tsinghuaDaimlerDataset_{seq:0>6}_{  
frame:0>9}_{type}{ext}
```

The meaning of the individual elements is:

- **root**: the root folder of the Tsinghua-Daimler Cyclist Benchmark. Our scripts are using relative paths pointing to this folder.
- **type**: the type/modality of data, e.g. 'labelData' for ground truth, or 'leftImg8bit' for left 8 bit images.
- **split**: the split, i.e. train/valid/NonVRU/test/. Thus, do not be surprised to find empty folders.
- **seq**: the sequence number using 6 digits.
- **frame**: the frame number using 6 digits. Note that some cities have only very few sequences, but many frames within those, while some cities have many sequences, but only the 19th frame is annotated.

- **ext**: the extension of the file and optionally a suffix, e.g. _labelData.json for ground truth files

Possible values of 'type'

- **LabelData**: Annotated bounding box data in JSON format, if available.
- **LeftImg8bit**: the left images in 8 bit LDR format. These are rectified annotated images.
- **disparity**: the left disparity images calculated with a Semi Global Matching Algorithm; Represented as 16bit unsigned image. Disparity has been scaled with a factor of 256. To get the original disparity use $\text{orig_disparity} = (\text{scaled_disparity} - 1.) / 256..$
- **camera**: extrinsic and intrinsic calibration
- **detections**: detection data produced by YOUR algorithm (it is a good idea to save your detections beside the data)

More types might be added over time and also not all types are initially available. Please let us know, if you need any other meta-data to run your approach.

Possible values of 'split'

- **train**: usually used for training, contains 9741 images with annotations only for "cyclist". Only cyclists which are fully visible (occlusion<10%) and higher than 60 pixels have been labeled here.
- **valid**: 1019 images to be used for validation of hyper parameters. Annotations for "pedestrian", "cyclist", "motorcyclist", "tricyclist", "wheelchairuser", "mopedrider". Only objects higher than 20 pixels have been labeled here.
- **test**: used for testing, contains 2914 images with annotations for pedestrian, riders with all kinds of two-wheelers ("cyclist", "motorcyclist", "tricyclist", "wheelchairuser", "mopedrider"). Only objects higher than 20 pixels have been labeled here.
- **NonVRU**: 1000 images in which no object of interest ("pedestrian", "cyclist", "motorcyclist", "tricyclist", "wheelchairuser", "mopedrider") is present.

Directory structure

When unzipped into the same folder, this results in the following directory structure:

TsinghuaPublish (root)

```

-----|-camera
-----|-test
-----|-valid
-----|-train
-----|-NonVRU
-----|-scripts (see section 4)
-----|-icons
-----|-disparity
-----|-test
-----|-valid
-----|-train
-----|-NonVRU
-----|-labelData
-----|-test
-----|-valid
-----|-train
-----|-NonVRU
-----|-leftImg8bit
-----|-test
-----|-valid
-----|-train
-----|-NonVRU

```

If extracting from downloaded single ZIP, please keep the directory structure. Script paths are relative!

7) JSON-Layout

General

Within this dataset each frame is represented by one json-file with a predefined format. The name of the json-file follows the scheme: tsinghuaDaimlerDataset_{seq:0>6}_{frame:0>9}_{type}.json

Possible values of type:

- **labelData**: Annotated bounding box data
- **detections**: Detections represented by bounding boxes produced by YOUR algorithm

Attention: During the evaluation process the correspondence is performed based on the filename of the detection and ground truth files.

Ground truth annotation and detection fields:

Line number	Variable name	Meaning
mincol, minrow, maxcol, maxrow	[unsigned integer]	Defining the position and size of the bounding box by its upper left and bottom right point
identity	[string]	Specifies the object class. Possible values are ["unlabeled", "pedestrian", "cyclist", "motorcyclist", "tricyclist", "wheelchairuser", "mopedrider"]
tags	[list of strings]	List (possibly empty) of specified tags. Used during the evaluation (occlusion level only). Possible tags are ["out_of_image", "unsure_box", "occluded>10", "occluded>40", "occluded>80"]

Ground truth only fields

uniqueid	[integer]	Unique number for each ground truth annotation, specifies one annotated object at one specific time
trackid	[string]	Unique string for each real object, consistent over the whole dataset (Layout: "{identity}_{0>6}")

Detection only fields

score	[float]	Detection score (confidence) of the given detection, higher is better
-------	---------	---

See the sample json-files in scripts/
detection -> Detections_sample.json

ground truth -> GroundTruthAnnotations_sample.json

Note – Empty json files

Be aware that in the test/valid sets the ground truth json file can also contain no objects. In this case there is no object of interest available in this frame.

How to convert your detection results into the expected json-format

The correspondence between the detection results and ground truth data during the evaluation is performed using the filename of the json-files (one file per frame).

Name scheme:

```
tsinghuaDaimlerDataset_{seq:0>6}_{frame:0>9}_{type}.json
```

Therefore, pay attention to use the right name during the detection parsing.

Default detection ext:

```
*_{type}.json = *_detection.json
```

Assuming your detection data is accessible in python, use the predefined Detection class from `scripts/detection.py`, with:

```
from detection import Detection
```

(remember to include `detection.py` to your python-import path)

The detection class can be used to save the set of all detections of one frame in the right json-format. Therefore, create a new Detection object with the `fileName` of your currently processed frame. Use the

```
addDetAnnotation()
```

function for each detection.

At the end, use the

```
toJsonFile(save_loc)
```

function to save the current frame along with all detections as a json-file in the right format.

8) Scripts

Please get the latest version of the scripts by checking out the git-repository via:

```
git clone https://github.com/FabianDai/tsinghuaDaimlerScripts [tsinghuaRoot]
```

There are several scripts included with the dataset in a folder named 'scripts':

[scripts/tsinghuaDaimlerViewer.py](#)

view the images and overlay the annotations and, if activated, your produced detection results.

Possible needed configurations:

Line number	Variable name	Meaning
78	<code>dataRootPath</code>	Specifies the absolute path of the sandbox's root directory

[scripts/annotation.py](#)

internal representation of ground truth annotations used by `scripts/tsinghuaDaimlerViewer.py` also implements an json-exporter

[scripts/detection.py](#)

a good start for loading the detections from your own format and convert it to the json format used by our scripts

See the main function for some sample calls and how to use the Detection class

[scripts/evaluation.py](#)

Python script version of the original used Matlab evaluation used in the paper to generate precision-recall curves.

Line number	Variable name	Meaning
41	self.difficulty	Change the used difficulty level (easy, moderate, or hard)
58ff	file_extensions	Change to use other file extensions for detection and groundtruth data
481	pathToGtFiles	Path to the folder including all the ground truth files (remember: one json-file per frame in the abovementioned format)
488ff	detectionEvalList	A list of tuples containing the name of the detection method as well as the path to the folder containing all the detection results (again one json-file per frame. same naming scheme)

See chapter 9) for further information on evaluation and upcoming features.

Run the evaluation with:

```
scripts/ -> python evaluation.py
```

[scripts/camera.py](#)

See for how to load camera data and use functions to convert from one coordinate system to the other (image-->camera, image-->world)

[scripts/depth.py](#)

See for how to load scaled disparity images and calculate distance from disparity

[scripts/setup.py \(beta\)](#)

Setup scripts for the dataset.

9) Evaluation

Run the evaluation with: `scripts/ -> python evaluation.py`. Precision/Recall curves are plotted. You can compare multiple of your algorithms to the ground truth and plot them together.

Once you want to test your method on the test set, please run your approach on the provided test images and submit your results on our evaluation server (available beginning of next year).

Future work includes also a non-public test set on Tsinghua-Daimler Benchmark Dataset to make a fair comparison possible (available beginning of next year).