

BrnoCompSpeed Dataset

Dataset available at: <https://github.com/JakubSochor/BrnoCompSpeed>

(This dataset supports report **Vehicle and Pedestrian Trajectory and Gap Estimation for Traffic Conflict Prediction**)

This U.S. Department of Transportation-funded dataset is preserved in the GitHub Repository (<https://github.com/>), and is available at <https://github.com/JakubSochor/BrnoCompSpeed>

The related final report **Vehicle and Pedestrian Trajectory and Gap Estimation for Traffic Conflict Prediction**, is available from the National Transportation Library's Digital Repository at <https://rosap.ntl.bts.gov/view/dot/60837>

Metadata from the GitHub Repository record:

Below is the attached README, that can be found in the GitHub Repository:

BrnoCompSpeed Dataset Evaluation Code

Dataset published with paper **SOCHOR Jakub et al. Comprehensive Data Set for Automatic Single Camera Visual Speed Measurement, IEEE T-ITS**

Download

To download the dataset, reach out to Jakub Špaňhel - ispanhel@fit.vutbr.cz.

How to use the code

1. Install **PYTHON2**, packages you might NOT have: numpy, scipy, matplotlib, tabulate
2. Download the [dataset](#) and place the results and dataset folders from the downloaded archive on the same level as the code folder (root of the repository).
 - TIP: to save disk space use following command to get and unpack the dataset (WARNING: it has ~200GB)
 - ```
curl https://medusa.fit.vutbr.cz/traffic/data/2016-ITS-BrnoCompSpeed-full.tar | tar xv
```
2. (Optional) Modify paths in file code/dataset\_info.py

3. Check file code/config.py. The most important variables are RUN\_FOR\_SYSTEMS and RUN\_FOR\_VIDEOS.
4. Run in code directory: `python eval.py` and wait. The results will be computed, shown and cached in results directory. The script `eval.py` has several arguments, so you can use `--help` for explanation of the arguments. I STRONGLY recommend to use `ipython`, `spyder` or similar terminals.

## Additional information

- For information about vehicle types from our country (Czech Republic), you can use following datasets: [COD20K](#), [BoxCars21k](#).
- The dataset itself contains videos, mask, screenshots, and pkl file with the ground truth data.
- The dataset contains extra session0 which was annotated manually and is not included in the paper.
- The session0 is meant to be as training for all the splits (A,B,C - see the paper).

## How to generate your own result JSON files

- Just place them into appropriate subdirectory in the results directory.
- The JSON files should have same structure as the already existing JSON files.
- The structure of the files should be following:

```
{
 "camera_calibration":
 {"vp1": [x,y],
 "vp2": [x,y],
 "pp": [x,y],
 "scale": lambda}, (see the paper and supplementary pdf for
information how this is used)
 "cars":[
 {
 "id": number,
 "frames", "posX", "posY": each key defining list of
positions of the car in the video
 posX and posY should contain x,y coordinates of a point of
the vehicle which is on the road plane
 and frames should contain frame numbers of the reported
points. The length of the vectors must be equal.
 Examples can be found in the results directory.
 }
]
}
```

# Contact

{isochor,herout,ijuranek}@fit.vutbr.cz

---

## Recommended citation:

Jakub Špaňhel; Chinmaya Meher, " BrnoCompSpeed Dataset",  
<https://github.com/JakubSochor/BrnoCompSpeed>

## Dataset description:

This dataset contains 1 .zip file collection described below.

### BrnoCompSpeed-master.zip:

- BrnoCompSpeed-master Folder
  - README.md
  - results Folder
    - .gitignore
  - dataset Folder
    - checksums.sha
    - .gitignore
  - code Folder
    - utils.py
    - loading.py
    - eval.py
    - dataset\_info.py
    - config.py

### File Type Descriptions:

- File extension .md is among others related to texts and source codes in Markdown markup language. Markdown is a lightweight markup language, to write using an easy-to-read, easy-to-write plain text format, then convert it to structurally valid XHTML (or HTML) (for more information on .md files and software, please visit <https://www.file-extensions.org/md-file-extension>).
- File extension .sha is associated with the Unix operating system and used for Unix SHAR self-executing archives. A .sha file is basically a shell script, and executing it will recreate the files (for more information on .sha files and software, please visit <https://www.file-extensions.org/sha-file-extension>).
- File extension gitignore is associated with Git, a version control system developed by Linus Torvalds for various platforms that can run on local machine also as server app. These .gitignore files are text configuration files used by Git used to determine which files and directories to ignore, before user make a commit (for more information on

.gitignore files and software, please visit <https://www.file-extensions.org/gitignore-file-extension>).

- The .py file extension is commonly used for files containing source code written in Python programming language. Python is a dynamic object-oriented programming language that can be used for many kinds of software development (for more information on .py files and software, please visit <https://www.file-extensions.org/py-file-extension>).

**National Transportation Library (NTL) Curation Note:**

As this dataset is preserved in a repository outside U.S. DOT control, as allowed by the U.S. DOT's Public Access Plan (<https://ntl.bts.gov/public-access>) Section 7.4.2 Data, the NTL staff has performed *NO* additional curation actions on this dataset. NTL staff last accessed this dataset at <https://github.com/JakubSochor/BrnoCompSpeed> on 2022-03-06. If, in the future, you have trouble accessing this dataset at the host repository, please email [NTLDataCurator@dot.gov](mailto:NTLDataCurator@dot.gov) describing your problem. NTL staff will do its best to assist you at that time.