

A hybrid spatial-temporal deep learning architecture for lane detection

Efficient sequential neural network based on spatial-temporal attention and linear LSTM for robust lane detection using multi-frame images

This is the source code of the two papers:

[1] Dong, Y., Patil, S., Van Arem, B., & Farah, H. (2023). A hybrid spatial-temporal deep learning architecture for lane detection. Computer-Aided Civil and Infrastructure Engineering, 38(1), 67-86. <https://doi.org/10.1111/mice.12829>

[2] Patil, S.#, Dong, Y.#,*, Farah, H., & Hellendoorn, J. (2024). Efficient Sequential Neural Network Based on Spatial-Temporal Attention and Linear LSTM for Robust Lane Detection Using Multi-Frame Images (Under Review)

keywords: {Lane detection; Feature extraction; Neural networks; Image segmentation; Deep neural network; Deep learning; Computer vision}

In this document, we provide the dataset and the model description.

tvtLANE Dataset

Description: (adapted from <https://github.com/qinnzou/Robust-Lane-Detection>)

This dataset contains 19383 image sequences for lane detection, and 39460 frames of them are labeled. These images were divided into two parts, a training dataset contains 9548 labeled images and is augmented by four times, and a test dataset has 1268 labeled images. The size of images in this dataset is 128*256 pixels.

+ Training set:

- Data augmentation:

The training set is augmented. By flipping and rotating the images at three degrees, the data volume is quadrupled. These augmented data are separated from the original training set, which is named by “origin”. “f” and “3d” after “-” are represented for flipping and rotation. Namely, the “origin- 3df” folder is the rotated and flipped training set.

- Data construction:

The original training set contains continuous driving scene images, and they are divided into image sequences by twenty frames per second. All images are contained in “clips_all”, and there are 19096 sequences for training. Each 13th and 20th frame in a sequence are labeled, and the 38192 image and their labels are in “clips_13(_truth)” and “clips_20(_truth)”.

The original training dataset has two parts. Sequences in “0313”, “0531” and “0601” subfolders are constructed on the TuSimple lane detection dataset, containing scenes in American highways. The four “weadd” folders are added images in rural road in China.

+ Test set:

- Testset #1:

The normal test set, named Testset #1, is used for testing the overall performance of algorithms. Sequences in “0530”, “0531” and “0601” subfolders are constructed on the TuSimple lane dataset. 270 sequences are contained, and each 13th and 20th image is labeled.

- Testset #2:

Testset #2 is used for testing the robustness of algorithms. 12 kinds of hard scenes for human eyes are contained. All frames are labeled.

Using:

Index files are provided. For detecting lanes in continuous scenes, the input size is 5 in the implementation. Thus, the former images are additional information to predict lanes in the last frame, and the last (5th) frame is the one with labeled ground truth.

We use different sampling strides to get 5 continuous images, as shown in the paper. Each row in the index files represents a sequence and its label for training. Refer to (<https://github.com/qinnzou/Robust-Lane-Detection/blob/master/LaneDetectionCode/save/result/lane3.png>)

Download:

You can download this ****dataset**** from

BaiduYun: <https://pan.baidu.com/s/1IE2CjuFa9OQwLIbi-OomTQ> With passcodes: tf9x

Or

Google Drive:

https://drive.google.com/drive/folders/1MI5gMDspzuV44lfwzpK6PX0vKuOHUbb_?usp=sharing

You can also download the ****pre-trained model**** from the following link,

BaiduYun: <https://pan.baidu.com/s/1ioOFRj6wZzORl6i5c73vCw> With passcodes: y3sl

Or

Google Drive:

https://drive.google.com/drive/folders/1wF5m4GUEAgLWx3yb0GB09WOjNq2i3OZX?usp=drive_link

Set up

Requirements

PyTorch 1.10.2

Python 3.9

CUDA 8.0

cuDNN 11.1

Preparation

Data Preparation

The tvtLANE dataset contains 19383 continuous driving scenes image sequences, and 39460 frames of them are labeled. The size of the images is 128*256.

The training set contains 19096 image sequences. Each 13th and 20th frame in a sequence are labeled, and the image and their labels are in “clips_13(_truth)” and “clips_20(_truth)”. All images are contained in “clips_all”.

Sequences in “0313”, “0531” and “0601” subfolders are constructed on TuSimple lane detection dataset, containing scenes in American highways. The four “weadd” folders are added images in rural road in China.

The test set has two parts: Testset #1 (270 sequences, each 13th and 20th image is labeled) for testing the overall performance of algorithms. Testset #2 (12 kinds of hard scenes, all frames are labeled) for testing the robustness of algorithms.

To input the data, the authors provide three index files (train_index, val_index, and test_index). Each row in the index represents a sequence and its label, including the former 5 input images and the last ground truth (corresponding to the last frame of 5 inputs).

The tvtLANE dataset can be downloaded and put into “./LaneDetectionCode/data/”. If you want to use your own data, please refer to the format of the dataset and corresponding index files.

Training

Before training, change the paths including “train_path”(for train_index.txt), “val_path”(for val_index.txt), “pretrained_path” in config.py to adapt to your environment.

Choose the models (UNet_ConvLSTM | SCNN_UNet_ConvLSTM | SCNN_SegNet_ConvLSTM) and adjust the arguments such as class weights, batch size, and learning rate in config.py.

Then simply run: train.py

Test

To evaluate the performance of a trained model, please put the trained model listed above or your own models into “./LaneDetectionCode/model/” and change “pretrained_path” in test.py at first, then change “test_path” for test_index.txt, and “save_path” for the saved results.

Choose the right model that would be evaluated, and then simply run: test.py

The quantitative evaluations of Accuracy, Precision, Recall and F1 measure would be printed, and the result pictures will be saved in `"/LaneDetectionCode/save/test/"`.

Citation:

Please cite our paper if you use this code or data in your own work:

```
@article{dong2023hybrid,  
  title={A hybrid spatial--temporal deep learning architecture for lane detection},  
  author={Dong, Yongqi and Patil, Sandeep and Van Arem, Bart and Farah, Haneen},  
  journal={Computer-Aided Civil and Infrastructure Engineering},  
  volume={38},  
  number={1},  
  pages={67--86},  
  year={2023},  
  publisher={Wiley Online Library}  
}
```

Copy Right:

This dataset was collected for academic research only.

Contact:

For any problem with this code implementation, please contact Yongqi Dong (qiyuandream@gmail.com).