DET: A High-resolution DVS Dataset for Lane Extraction

Wensheng Cheng[1*], Hao Luo[1*], Wen Yang[1], Lei Yu[1], Shoushun Chen[2], and Wei Li[3]

[1]Wuhan University, [2]CelePixel Technology Co., Ltd, [3]Shanghai Baolong Automotive Corporation

{cwsinwhu,luohaowhu,yangwen,ly.wd}@whu.edu.cn, shoushun.chen@celepixel.com

liwei@chinabaolong.net

Abstract

Lane extraction is a basic yet necessary task for autonomous driving. Although past years have witnessed major advances in lane extraction with deep learning models, they all aim at ordinary RGB images generated by frame-based cameras, which limits their performance in nature. To tackle this problem, we introduce Dynamic Vision Sensor (DVS), a type of event-based sensor to lane extraction task and build a high-resolution DVS dataset for lane extraction (DET). We collect the raw event data and generate 5,424 *event-based sensor images with a resolution of* 1280×800*, the highest one among all DVS datasets available now. These images include complex traffic scenes and various lane types. All images of DET are annotated with multi-class segmentation format. The fully annotated DET images contains* 17,103 *lane instances, each of which is labeled pixel by pixel manually. We evaluate state-of-the-art lane extraction models on DET to build a benchmark for lane extraction task with event-based sensor images. Experimental results demonstrate that DET is quite challenging for even state-of-the-art lane extraction methods. DET is made publicly available, including the raw event data, accumulated images and labels*[1].

1. Introduction

Autonomous driving has received much attention in both academia and industry. The goal is to understand the environment of the car comprehensively through the use of various sensors and control modules. It consists of many challenging tasks, including lane extraction, traffic marks recognition, pedestrians detection [35, 19, 31], etc. Among them, lane extraction is a fundamental yet important one, as it helps car to adjust its position according to lanes precisely. It becomes the basis for following applications, including lane departure and trajectory planning functions. Hence performing accurate lane extraction is a key factor of autonomous driving.

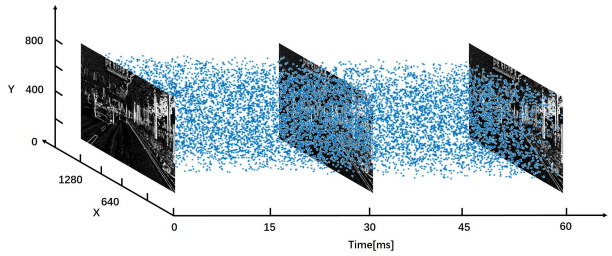


Figure 1. Visualization of the event output in space-time. Blue dots represent individual asynchronous events.

Researchers have proposed lots of methods for this task. These methods are either based on handcrafted features and heuristic algorithms [3, 6, 11, 13, 29, 34] or end-to-end Convolutional Neural Network (CNN) models [7, 14, 12, 9, 20, 17]. Although they have achieved promising results, there still exist problems in practice.

In real situation, cars would meet various complex and extreme scenes. For instance, these methods can not work well when the light is extremely dark or changes rapidly. In these situations, frame-based cameras can not capture scenes clearly and these methods fail due to the terrible input [2]. In nature, these difficulties come from RGB images generated by these standard cameras. Therefore, we turn to event-based camera. Event-based camera is a novel vision sensor developed in recent years. Fig.1 shows a visualization of the event output. It has two key characteristics: low latency and high dynamic range. Latency is based on the sensor sampling rate and data process time. Since event-based camera transmits data with events, which denotes illumination change, it has a latency of microseconds (μs), compared with 50-200 millisecond (ms) of standard cameras [24]. With such low latency, event-based camera can capture the environment and generate image much faster than standard cameras. This property ensures that it won't

*Equal contribution.

[1]DET website is https://spritea.github.io/DET/

be affected by motion blur, which is a troublesome problem for frame-based cameras. Besides, with much shorter response time brought by low latency, it also makes the autonomous cars much more agile than others.

As for dynamic range, event-based sensor owns a typical dynamic range of 130 dB v.s. 60 dB of standard ones, which is 7 orders of magnitude larger [24]. This character makes it able to deal with scenes featured by large illumination changes, which is a crucial point in autonomous driving. Suppose a car is going through the tunnel, the moments it enters and leaves the tunnel would result in such illumination change and corresponding images would become highly dark or light. This makes it almost impossible to recognize lanes from these images. But for event-based camera, lanes are still clear due to the high dynamic range. This is illustrated in Fig.2.

Furthermore, event-based sensor generates semi-dense images, due to the event stream data. Hence images produced by the sensor only contain pixels whose brightness changes, which are usually moving objects. These objects are exactly what we care in autonomous driving, including cars, pedestrians, traffic marks and lanes. Background things, or redundant information like sky, road surface, etc., are removed in nature, which benefits following processes.

A potential problem for adopting event-based sensor to lane extraction task is the image resolution. Ordinary event-based camera generates images with resolution as low as 240×180, which is definitely insufficient for this task requiring rich details.

For reasons above, we construct a high-resolution DVS dataset for lane extraction (DET). There are 5,424 event-based sensor images of 1280×800 pixels with corresponding labels. Note that the raw event data is also provided for those algorithms using event data directly [28, 16]. These images are split into training set of 2,716 images, validation set of 873 images and test set of 1,835 images. We provide two kinds of event-based sensor images, raw images generated by the sensor directly and images after filtering. Because the sensor is sensitive to illumination changes and there would be lots of noise pixels in raw image. We further offer two kinds of labels, per-pixel label without distinguishing different lanes and per-pixel label with distinguishable lanes. The reason is state-of-the-art models for lane extraction are either based on semantic segmentation or instance segmentation, which requires different labels. We then test state-of-the-art models on DET and report the results. As far as we know, this is the first dataset for lane extraction with event-based sensor images. It's also the first DVS dataset with such high resolution.

In summary, our contributions are:

- We provide a DVS dataset for lane extraction, including the raw event data and accumulated images with labels. To our knowledge, DET is the first event camera dataset for this task and the first event camera dataset with such high resolution images of 1280×800 pixels.
- We benchmark state-of-the-art lane extraction algorithms on DET comprehensively, which becomes the baseline for future research.

The purpose of this work is to exploit event-based sensor usage in lane extraction, a major aspect of autonomous driving, provide a benchmark for this task, and draw autonomous driving community's attention to this promising camera.

2. Related Work

2.1. Event Camera Dataset

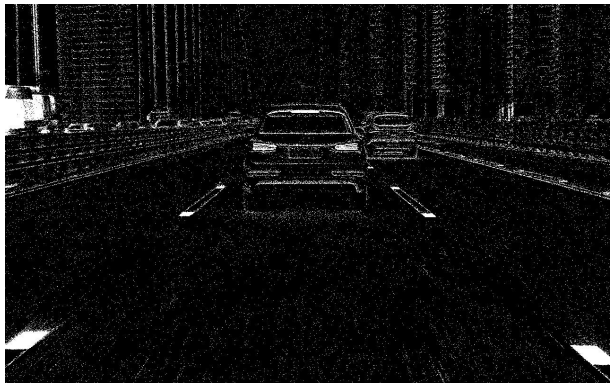
Synthesized Dataset. [24] have proposed a Dynamic and Active-pixel Vision sensor (DAVIS) dataset and simulator. DAVIS combines a global-shutter camera with an event-based sensor. The collection of datasets are captured with event-based sensor in a variety of synthetic and real environments. It contains not only global-shutter intensity images and asynchronous events, but also movement and pose parameters. It consists of various scenes, such as wall poster, outdoors, office and urban. The dataset is used for pose estimation, visual odometry and SLAM. It has a resolution of 240×180.

Classification Dataset. CIFAR10-DVS [18] is an event-stream dataset for object classification. They convert 10,000 frame-based images of CIFAR-10 dataset into 10,000 event streams using a event-based sensor with 128×128 resolution, which becomes an event-stream dataset of intermediate difficulty in 10 different classes. They adopt a repeated closed-loop smooth (RCLS) movement of frame-based images to convert frame-based images to event streams. With this transformation, they generate rich local intensity changes in continuous time which are quantized by each pixel of the event-based camera.

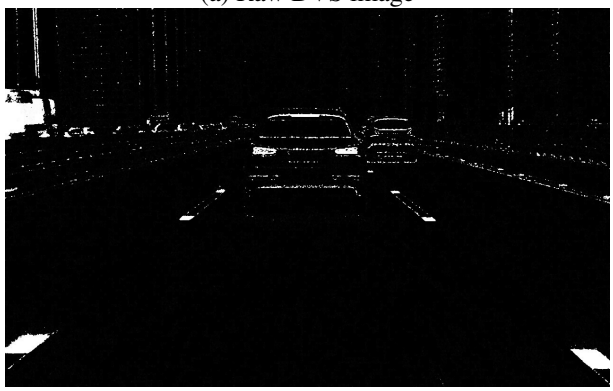
Recognition Dataset. [10] releases a series of DVS benchmark datasets. They convert established visual video benchmarks for object tracking, action recognition and object recognition into spiking neuromorphic datasets, recorded with DAViS240C camera of 240×180 resolution. They transform four widely used dynamic datasets: the VOT challenge 2015 Dataset [33], Tracking Dataset [32], the UCF-50 Action Recognition Dataset [27] and the Caltech-256 Object Category Dataset [8].

Driving Dataset. DDD17 [2] is an open dataset of annotated DAVIS driving recordings. It has 12 hours of a 346×260 pixel DAVIS sensor recording highway and city driving in various weather conditions, along with vehicle speed and GPS position. It also includes driver steering, throttle and brake captured from the car's on-board diagnostics interface. This dataset owns data coming from a variety

Figure 2. The process of coming out of the tunnel ($T1 < T2 < T3 < T4 < T5$). The first row shows gray images captured by frame-based camera. The second row shows corresponding event-based sensor images captured the same moment. Traditional cameras are largely affected by the sudden light change due to the low dynamic range, while event-based camera doesn't suffer from that with much higher dynamic range.



(a) Raw DVS image



(b) Filtered DVS image

Figure 3. Comparison of raw DVS image and corresponding filtered DVS image. Although (b) is clearer, it contains less details than (a).

of sensors and devices, which has a great significance for autonomous driving.

DVS datasets listed above are proposed for general computer vision or robotic control tasks. None of them is aimed at lane extraction task. Besides, event-based images in these datasets only have a low spatial resolution, like 128×128 or 240×180. The low resolution puts a hard bound on algorithms' performance on these datasets.

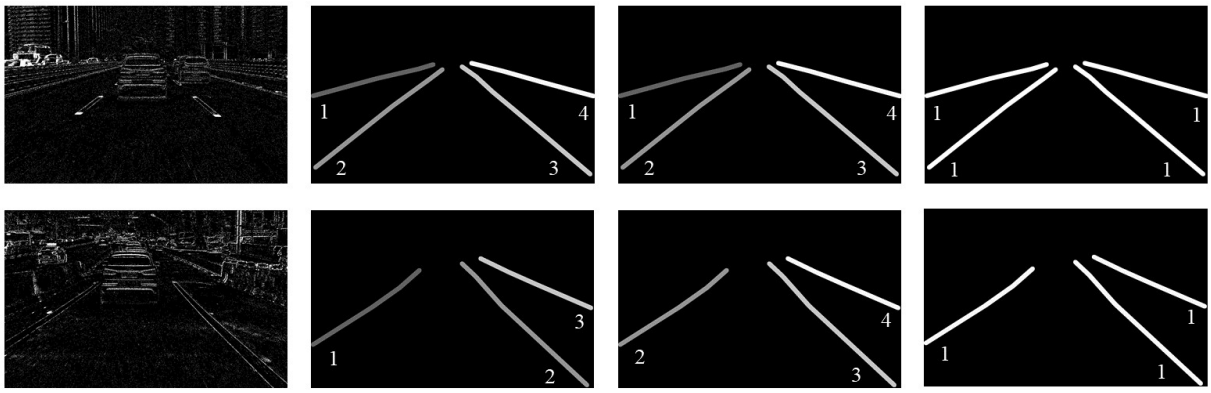
2.2. Lane Dataset

Caltech Lanes Dataset. This dataset[1] is composed of clips on different types of urban streets, with/without shadows and on straight and curved streets. It labels all visible lanes in four clips, totaling 1,224 labeled frames containing 4,172 marked lanes. This is an early dataset published in 2008.

tuSimple Dataset. This dataset[30] has 6,408 labeled images, which are split into 3,626 training images and 2,782 test images. These images are captured under good and medium weather condition. They contain highway roads with a different number of lanes, like 2 lanes, 4 lanes or more. For each image, 19 previous frames are also provided, but without annotation.

CULane Dataset. This dataset[26] contains 133,235 frames extracted from 55 hours of video. They divide the dataset into 88,880 training images, 9,675 validation images, and 34,680 test images. These images are undistorted and have a resolution of 1640×590. The test set is further split into normal and other challenging categories, including crowded and shadow scenes.

These lane datasets are all based on RGB images generated by frame-based cameras. Illumination changes and motion blur would affect model's performance based on theses images seriously, which should definitely be avoided in real traffic situation.



(a) (b) (c) (d)

Figure 4. Comparison of different label formats. (a) shows input images. (b) shows the label format that sets a fixed order and annotates lanes from left to right. (c) shows our label format based on the relative distance between lane and event camera. (d) shows the binary label format. For the left lane most close to event camera, it looks similar in different images and should be annotated with same label. (b) gives it a label of 2 in the image above, but gives it a label of 1 in the image below. Our format (c) annotates it exactly in both images.

2.3. Event Camera in Autonomous Driving

Since event camera is still fairly new compared with standard frame-based cameras, there are only a few projects employing event camera in autonomous driving. Two typical applications would be introduced in this section.

Steering Prediction. [23] tries to make use of event camera to predict a vehicle's steering angle. They adapt fancy convolutional architectures to the output of event sensors and extensively evaluate the performance of their approach on public dataset. They further show that it is possible to leverage transfer learning from pretrained convolutional networks on classification tasks, though the network is trained on frames collected by traditional cameras.

Car Detection. [5] attempts to detect cars with event camera and pseudo-labels coming from gray images generated by traditional cameras. They transferred discriminative knowledge from a state-of-the-art frame-based CNN to the event-based modality through intermediate pseudo-labels, which are used as targets for supervised learning. The model can even complement frame-based CNN detectors, which suggests that it has learnt generalized visual representations.

Although these works explore event camera's usage in autonomous driving, researchers haven't paid attention to the fundamental task, lane extraction. This is an area with great potential for event cameras, which shows obvious advantage over traditional frame-based cameras on this task.

3. Construction of DET

3.1. Data Collection

To collect data, we mount the event-camera CeleX V with high resolution 1280×800 on a car in different locations and record event streams by driving in Wuhan City at different time. As a metropolis in China, Wuhan City provides complex and various traffic scenes which are challenging for lane extraction.

We record over 5 hours of event stream with a sampling rate of MHz, which equals a sampling interval of μs. We compress the raw event stream along the time dimension with $\triangle t = 30\ ms$. $\triangle t$ denotes the event stream time span that one single image corresponds to. This is illustrated in Fig.1. Then we get over 150,000 images from raw event stream. We carefully choose 5,424 images containing various scenes to label.

For these images, we found that there are some noise pixels due to the event camera imaging characteristic. We simply use median filter to remove these pixels and get clean images. We provide *both* raw images and filtered images publicly, as shown in Fig.3. We recommend researchers to adopt raw images as they are more close to real world and retain more details. Filtered images would lose some details anyway.

3.2. Data Annotation

Task Definition. Lane extraction task has been defined in two ways. One is to extract lanes without discriminating between lanes, and the other one is to differentiate lanes from each other. We argue the latter is more practical, because it's essential for autonomous driving system to use the location of each lane to decide car's position. Therefore, we define lane extraction here as extracting lanes from traffic scenes while discriminating between lanes.

For existing CNN-based lane extraction algorithms consistent with this definition, we divide them into two types: semantic segmentation method and instance segmentation method. Semantic segmentation method regards lane extraction problem as a multi-class segmentation task. It clas-

sifies each pixel into $(n+1)$ categories, where n denotes lane types and 1 denotes background. Lanes with same label are supposed to be similar in some sense. Instance segmentation method is same with that, except it doesn't guarantee lanes with same label are similar. It only separates lanes into individual ones, without considering the similarity of lanes from different images. As a result, lanes with same label generated by instance segmentation method in different images may differ a lot.

Annotation Details. Both semantic segmentation method and instance segmentation method in this task require multi-class label. In our dataset, there are 4 lanes at most in one image. Hence it's a five-class classification task. We give each pixel one of five labels, i.e., $\{0, 1, 2, 3, 4\}$. 0 is for background and others for lanes. Here comes the question, that how we decide the label for each lane.

Generally, there are two kinds of rules to decide the specific label. One is to set a fixed order and label each lane with this order. The other one is to give lanes with similar characteristics same label. We argue that the latter is better. The former label format is only related to the number of lanes in the image, without considering lane's appearance at all. Under this format, we label each lane with a fixed order, like 1 to 4 for lanes from left to right, and lanes with same label from different images may differ a lot. A typical example is shown in Fig. 4 (b). The main cause is that the relative instance between lanes with same label from different images and event camera varies largely during the process of driving. This would impede the training process of multi-class semantic segmentation model obviously.

For reasons above, we choose the latter format to label images. The key point is to define the similarity. We find the shape and size of lanes in image mainly depend on the relative distance between lanes and event camera. Lanes whose distances from event camera are alike seem similarly. Therefore, for the two lanes most close to event camera, a.k.a. ego lanes [15], we give the left lane label of 2, and the right lane label of 3, no matter the number of lanes in the image. Then other lanes' labels are confirmed by their distance to these two lanes, as Fig. 4 (c) illustrates. In this way, we give lanes with similar appearances same label, which is more reasonable for multi-class semantic segmentation method. The lane width is fixed as 20 pixels.

Considering methods defining this task as extracting lanes without discriminating between them, or extracting lanes first then differentiating them in post-process stage, we also provide binary labels for researchers interested in this. This is shown in Fig. 4 (d).

3.3. Data Split

To ensure that the training data and test data distributions match approximately, we randomly extract 1/2 of original images as training set, 1/6 as validation set and 1/3 as test set. We would provide all original images, including raw DVS images and filtered images with corresponding labels, containing multi-class labels and binary labels publicly.

Table 1. Distribution of images containing various number of lanes. One represents the image containing only one lane, the same with others. Quantity is the number of images containing certain number of lanes. Percentage is the corresponding proportion.

Statistics	One	Two	Three	Four	Total
Quantity	161	1,114	1,918	2,231	5,424
Percentage %	2.97	20.54	35.36	41.13	100

4. Properties of DET

4.1. High-resolution Image

Existing event camera datasets have a typical resolution of 346×260 pixels, which is fairly low compared with RGB images generated by frame-based cameras. For complex scenes in autonomous driving, event camera images of this resolution containing little information can not complete this task well. Therefore, we adopt the CeleX-V DVS released in 2018 for our dataset. It is featured by the highest resolution of 1280×800 pixels among all event cameras available now and a latency as low as $5ns$. All images in DET have the same resolution of 1280×800.

4.2. Various Lane Types

For dataset of lane extraction, the lane diversity plays an important role. The more diverse lane type is, the more close to real world dataset becomes. For this purpose, we collect images containing various lane types, including single dashed line, single solid line, parallel dashed lines, parallel solid line and dashed line, etc. Note that parallel lines are labeled as one whole line for consistency. Samples of these lane types are shown in Fig. 5.

4.3. Various Lane Number

About images containing various numbers of lanes, lanes would look different due to the relative distance between some lane and event camera, as explained in Sec. 3.2. Therefore, we gather images including different numbers of lanes by driving on roads with various numbers of carriageways. Samples containing different numbers of lanes are shown in Fig. 5. Tab. 1 summarizes the lane number distribution.

4.4. Various Traffic Scenes

In addition to the diversity of lane itself, we consider the diversity of scenes as important too. Because a robust lane extraction is supposed to recognize lanes under various traffic scenes, which is exactly what autonomous driving requires. Due to this, we record images containing various

Figure 5. Samples of DVS images and labels in DET. First rows show various line types, including single dotted line, single solid line, parallel dotted line, parallel solid line and dotted line. Middle rows show various lane number, from 1 to 4. Last rows show various traffic scenes, urban, tunnel, bridge and overpass.

traffic scenes by driving on tunnels, bridges, overpasses, urban areas, etc. These traffic scenes are presented in Fig. 5.

4.5. Various Camera Views

To simulate the real situation, we further mount event-based sensor with different locations on our car. In this way, even lanes with same label in different images would differ to some extent. In other words, we increase intraclass variance of DET. Although this would make it more difficult to train lane extraction models, models trained with this dataset would handle complex scenes better than those with single camera view. We argue this is necessary for complicated real traffic scenes, which might be even more complex than DET.

5. Evaluation

5.1. Evaluation Settings

Dataset Setting. We conduct evaluations with two kinds of lane extraction methods, semantic segmentation based method and instance segmentation based method. To fully take use of labeled data, we use training set and validation set together to train these models, and test set to check their performance.

Training Details. All models are trained on one Titan Xp GPU with 12 GB memory. The batch size is set as 4. Stochastic gradient descent (SGD) with momentum is adopted to train this network. Momentum is set as 0.9. We apply poly learning rate policy to adjust learning rate, which reduces the learning rate per iteration. This could be expressed as:

$$LR = initial\ LR \times (1 - \frac{current\ iter}{max\ iter})^{power} \quad (1)$$

where LR is current learning rate, $initial\ \ LR$ is initial learning rate, $current\ \ iter$ is current iteration step, and $max\ iter$ is the max iteration step. The $initial\ LR$ is set as 0.01 and $power$ is set as 0.9. For all models, $max\ iter$ is set to 50,000 in our experiment to make sure these networks converge.

Metrics. As there exist some differences between the metric of semantic segmentation and instance segmentation, we choose two metrics for both of them to compare numerically. Specifically, we adopt widely used *F1 score* (F1) and *intersection over union* (IoU) as metric.

F1 is defined as

$$\mathrm{F1} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \quad (2)$$

and

$$\mathrm{Precision} = \frac{TP}{TP + FP},\ \mathrm{Recall} = \frac{TP}{TP + FN} \quad (3)$$

where $TP,\ FP,\ TN,\ FN$ are the number of true positive, false positive, true negative, false negative separately. They all count the number of *pixels*. IoU is defined as:

$$\mathrm{IoU}(P_m, P_{gt}) = \frac{\mathbb{N}(P_m \cap P_{gt})}{\mathbb{N}(P_m \cup P_{gt})} \quad (4)$$

where P_m is the prediction pixels set and P_{gt} is the ground truth pixels set. $\cap$ and $\cup$ mean the intersection and union set respectively. $\mathbb{N}$ denotes the number of pixels in the intersection or union set.

For semantic segmentation methods, F1 and IoU are calculated across all five classes. But for instance segmentation method, they don't correspond with manual annotations strictly, because lane label predictions of this method are generated by clustering. They are in nature binary classification task. Hence we consider all lanes the same and regard this as a binary classification task to evaluate.

Besides, we use precision-recall (PR) curve to assess the relation between precision and recall for lane extraction task. To be specific, we adopt different thresholds from 0 to 1 to predicted score map. Then we get several precision values with corresponding recall values and plot the PR curve. Note that for semantic segmentation method, we ignore the difference between lanes for comparison with instance segmentation method.

Table 2. Evaluation results of lane extraction methods on DET. Mean F1 (%) and Mean IoU (%) are the average F1 score and IoU of all classes, respectively. Values in bold are the best and values underlined are the second best.

Methods	Mean F1	Mean IoU
FCN	60.39	47.36
DeepLabv3	59.76	47.30
RefineNet	63.52	50.29
LaneNet	<u>69.79</u>	<u>53.59</u>
SCNN	**70.04**	**56.29**

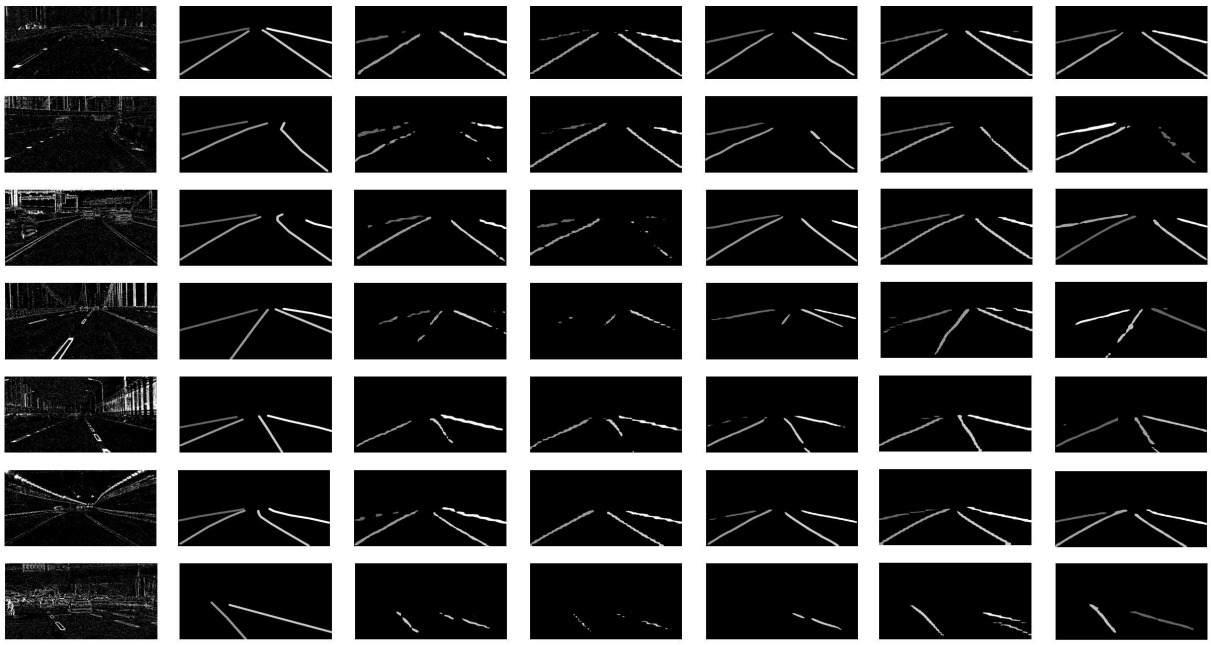
5.2. Lane Extraction Baselines

We benchmark typical lane extraction methods, including semantic segmentation based method, like FCN [22], DeepLabv3 [4], RefineNet [21], SCNN [26] and instance segmentation based method LaneNet [25]. FCN, RefineNet and DeepLabv3 are typical semantic segmentation methods for general computer vision tasks. FCN is the first work regarding semantic segmentation as pixel-level classification task. It builds fully convolutional neural network first and utilizes skip architecture to combine shallow layer semantic information with deep one. DeepLabv3 combines atrous spatial pyramid pooling with global pooling to introduce image-level global context. RefineNet explicitly exploits information along the down-sampling process to enable high-resolution prediction using long-range residual connections.

SCNN and LaneNet are specialized models for lane extraction task. SCNN generalizes traditional deep layer-by-layer convolutions to slice-by-slice convolutions within feature map, which enables message passings between pixels across rows and columns in a layer. This makes it particular suitable for long continuous shape structure recognition, like lane extraction. SCNN achieves state-of-the-art performance on tuSimple [30] dataset. LaneNet casts the lane extraction problem as an instance segmentation problem, and applies a learned perspective transformation based on the image instead of a fixed "bird's-eye view" transformation. It generates each lane instance by clustering. Hence it can handle scenes where lane category varies, although it can not assign similar lanes same label. Tab.2 is the result of lane extraction baselines. Fig.6 presents visual comparison of these methods. Fig.7 shows PR curves of these methods.

5.3. Experimental Analysis

Tab.2 shows that LaneNet and SCNN outperforms other semantic segmentation methods significantly. We argue that FCN, DeepLabv3 and RefineNet are general semantic segmentation methods, and they do not design specific modules for lane extraction task particularly. They do not apply any prior information or structural feature neither, which is of significant importance for lane extraction task. SCNN and



(a) (b) (c) (d) (e) (f) (g)

Figure 6. Visual comparison of lane extraction methods. (a) shows input images. (b) shows the corresponding label. (c-g) show results of FCN, DeepLabv3, RefineNet, SCNN and LaneNet. The label of (g) is random because it is generated by clustering. Note that they are all multi-class labels and lanes with same label are different from those with other labels in the gray value.

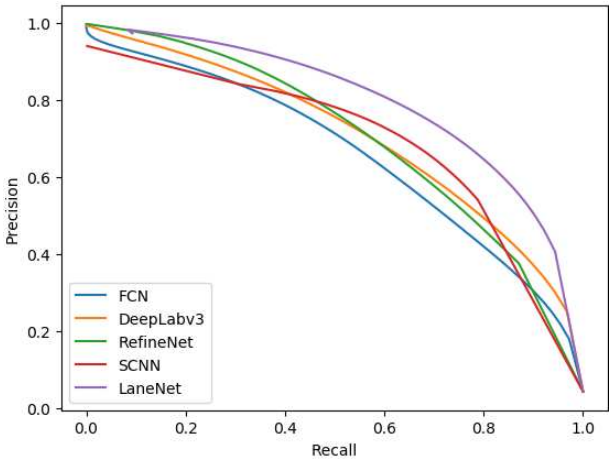


Figure 7. Precision-recall (PR) curves of lane extraction baselines.

LaneNet either adopts slice-by-slice convolution for continuous structure, or learns a perspective transformation to fit the lane into image. These special modules benefits their performance greatly for this problem.

Fig.7 shows that the area under LaneNet PR curve is larger than SCNN, which seems in conflict with Tab.2. We assume that methods except LaneNet are in nature multi-class classification task, but we draw the PR curve by dealing with their multi-class classification results as binary classification results for comparison with LaneNet, as Sec. 5.1 illustrates. This makes them seem worse than LaneNet, which is in fact not exactly. Evaluation results show this dataset is challenging, even for state-of-the-art lane extraction models.

6. Conclusion

In this paper, a high-resolution DVS dataset for lane extraction task, DET, is constructed. It consists of the raw event data, accumulated images and corresponding labels. 5,424 event-based images with resolution of 1280×800 are extracted from 5 hours of event streams with sampling rate of MHz. To provide a comprehensive labeled pairs, two types of annotations for lanes in the images are given, multi-class format and binary format. We also benchmark the state-of-the-art models for lane extraction and analyze the experimental results. Evaluation results of various lane extraction methods show this dataset is challenging.

To the best of our knowledge, DET is the first dataset of event-based sensor for lane extraction task, which is a fundamental yet important problem in autonomous driving. Furthermore, taking use of event-based sensor in autonomous driving is a rising area with great potential and we believe that DET would inspire community's enthusiasm for adopting event-based sensor to autonomous driving and exploring its usage in more applications.

References

[1] Mohamed Aly. Real time detection of lane markers in urban streets. In *IEEE Intelligent Vehicles Symposium*, pages 7–12, 2008. 3

[2] Jonathan Binas, Daniel Neil, Shih-Chii Liu, and Tobi Delbruck. DDD17: End-to-end davis driving dataset. *arXiv preprint arXiv:1711.01458*, 2017. 1, 2

[3] Amol Borkar, Monson Hayes, and Mark T Smith. A novel lane detection system with efficient ground truth generation. *IEEE Transactions on Intelligent Transportation Systems*, 13(1):365–374, 2012. 1

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 7

[5] Nicholas FY Chen. Pseudo-labels for supervised learning on dynamic vision sensor data, applied to object detection under ego-motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 644–653, 2018. 4

[6] Hendrik Deusch, Jürgen Wiest, Stephan Reuter, Magdalena Szczot, Marcus Konrad, and Klaus Dietmayer. A random finite set approach to multiple lane detection. In *IEEE Conference on Intelligent Transportation Systems*, pages 270–275, 2012. 1

[7] Raghuraman Gopalan, Tsai Hong, Michael Shneier, and Rama Chellappa. A learning approach towards detection and tracking of lane markings. *IEEE Transactions on Intelligent Transportation Systems*, 13(3):1088–1098, 2012. 1

[8] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. *Technical report*, 2007. 2

[9] Bei He, Rui Ai, Yang Yan, and Xianpeng Lang. Accurate and robust lane detection based on dual-view convolutional neutral network. In *IEEE Intelligent Vehicles Symposium*, pages 1041–1046, 2016. 1

[10] Yuhuang Hu, Hongjie Liu, Michael Pfeiffer, and Tobi Delbruck. Dvs benchmark datasets for object tracking, action recognition, and object recognition. *Frontiers in neuroscience*, 10:405, 2016. 2

[11] Junhwa Hur, Seung-Nam Kang, and Seung-Woo Seo. Multi-lane detection in urban driving environments using conditional random fields. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 1297–1302, 2013. 1

[12] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, et al. An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*, 2015. 1

[13] Heechul Jung, Junggon Min, and Junmo Kim. An efficient lane detection algorithm for lane departure detection. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 976–981, 2013. 1

[14] Jihun Kim and Minho Lee. Robust lane detection based on convolutional neural network and random sample consensus. In *International Conference on Neural Information Processing*, pages 454–461, 2014. 1

[15] Jiman Kim and Chanjong Park. End-to-end ego lane estimation based on sequential transfer learning for self-driving cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 30–38, 2017. 5

[16] Xavier Lagorce, Garrick Orchard, Francesco Galluppi, Bertram E Shi, and Ryad B Benosman. Hots: a hierarchy of event-based time-surfaces for pattern recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(7):1346–1359, 2017. 2

[17] Seokju Lee, Junsik Kim, Jae Shin Yoon, Seunghak Shin, Oleksandr Bailo, Namil Kim, Tae-Hee Lee, Hyun Seok Hong, Seung-Hoon Han, and In So Kweon. Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1947–1955, 2017. 1

[18] Hongmin Li, Hanchao Liu, Xiangyang Ji, Guoqi Li, and Luping Shi. Cifar10-dvs: an event-stream dataset for object classification. *Frontiers in neuroscience*, 11:309, 2017. 2

[19] Jianan Li, Xiaodan Liang, ShengMei Shen, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Scale-aware fast r-cnn for pedestrian detection. *IEEE transactions on Multimedia*, 20(4):985–996, 2018. 1

[20] Jun Li, Xue Mei, Danil Prokhorov, and Dacheng Tao. Deep neural network for structural prediction and lane detection in traffic scene. *IEEE transactions on neural networks and learning systems*, 28(3):690–703, 2017. 1

[21] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017. 7

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 7

[23] Ana I Maqueda, Antonio Loquercio, Guillermo Gallego, Narciso García, and Davide Scaramuzza. Event-based vision meets deep learning on steering prediction for self-driving cars. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5419–5427, 2018. 4

[24] Elias Mueggler, Henri Rebecq, Guillermo Gallego, Tobi Delbruck, and Davide Scaramuzza. The event-camera dataset and simulator: Event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research*, 36(2):142–149, 2017. 1, 2

[25] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In *IEEE Intelligent Vehicles Symposium*, pages 286–291, 2018. 7

[26] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 3, 7

[27] Kishore K Reddy and Mubarak Shah. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981, 2013. 2

[28] Amos Sironi, Manuele Brambilla, Nicolas Bourdis, Xavier Lagorce, and Ryad Benosman. Hats: Histograms of averaged time surfaces for robust event-based object classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1731–1740, 2018. 2

[29] Huachun Tan, Yang Zhou, Yong Zhu, Danya Yao, and Keqiang Li. A novel curve lane detection based on improved river flow and ransa. In *IEEE Conference on Intelligent Transportation Systems*, pages 133–138, 2014. 1

[30] The tusimple lane challange. http://benchmark.tusimple.ai/. 3, 7

[31] Yonglong Tian, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Pedestrian detection aided by deep learning semantic tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5079–5087, 2015. 1

[32] Tracking dataset. http://cmp.felk.cvut.cz/~vojirtom/dataset/tv77/. 2

[33] Vot2015 benchmark. http://www.votchallenge.net/vot2015/. 2

[34] Pei-Chen Wu, Chin-Yu Chang, and Chang Hong Lin. Lane-mark extraction for automobiles under complex conditions. *Pattern Recognition*, 47(8):2756–2767, 2014. 1

[35] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. Traffic-sign detection and classification in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2110–2118, 2016. 1