

# 20220621

---

[CARLANE: A Lane Detection Benchmark for Unsupervised Domain Adaptation from Simulation to multi...](#)

[Motivation](#)

[Dataset](#)

[Overview](#)

[Data collection](#)

[Annotation](#)

[Experiments](#)

[Ultra Fast Structure-aware Deep Lane Detection](#)

[混合车道表示法](#)

[有序分类](#)

[局限](#)

[学习目标](#)

[Loss](#)

[Experiments](#)

本次分享：王源彬、陈源

**CARLANE: A Lane Detection Benchmark for Unsupervised Domain Adaptation from Simulation to multiple Real-World Domains** [[Paper](#)] [[Dataset](#)]

---

# CARLANE: A Lane Detection Benchmark for Unsupervised Domain Adaptation from Simulation to multiple Real-World Domains

---

Julian Gebele<sup>\*,1</sup> Bonifaz Stuhr<sup>\*,1,2</sup> Johann Haselberger<sup>\*,1,3</sup>

<sup>1</sup>University of Applied Science Kempten

<sup>2</sup>Autonomous University of Barcelona

<sup>3</sup>Technische Universität Berlin

carlane.benchmark@gmail.com

## Motivation

UDA一般都做检测/分割，很少有做车道线的，原因是缺少benchmark

## Dataset

### Overview

- MoLane: model vehicle
- TuLane
- MuLane: multi-target



Figure 1: Images sampled from our CARLANE Benchmark.

Table 1: Dataset overview. Unlabeled images denoted by \*, partially labeled images denoted by \*\*

| Dataset | domain                        | total images | train   | validation | test  | lanes    |
|---------|-------------------------------|--------------|---------|------------|-------|----------|
| MoLane  | CARLA simulation              | 84,000       | 80,000  | 4,000      | -     | $\leq 2$ |
|         | model vehicle                 | 46,843       | 43,843* | 2,000      | 1,000 | $\leq 2$ |
| TuLane  | CARLA simulation              | 26,400       | 24,000  | 2,400      | -     | $\leq 4$ |
|         | TuSimple [20]                 | 6,408        | 3,268   | 358        | 2,782 | $\leq 4$ |
| MuLane  | CARLA simulation              | 52,800       | 48,000  | 4,800      | -     | $\leq 4$ |
|         | model vehicle + TuSimple [20] | 12,536       | 6,536** | 4,000      | 2,000 | $\leq 4$ |

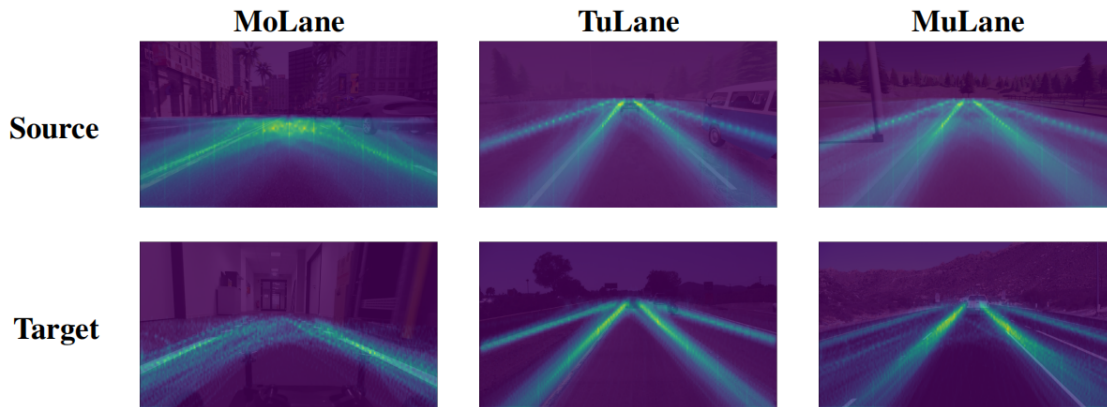


Figure 3: Lane annotation distributions of the three subsets of CARLANE. Since the real-world training data of MoLane and MuLane is unlabeled, we utilize their validation data for visualization.

### Data collection

- source: CARLA Simulator
- target: model vehicle + TuSimple, 预先设定轨道, 采黑色和灰色

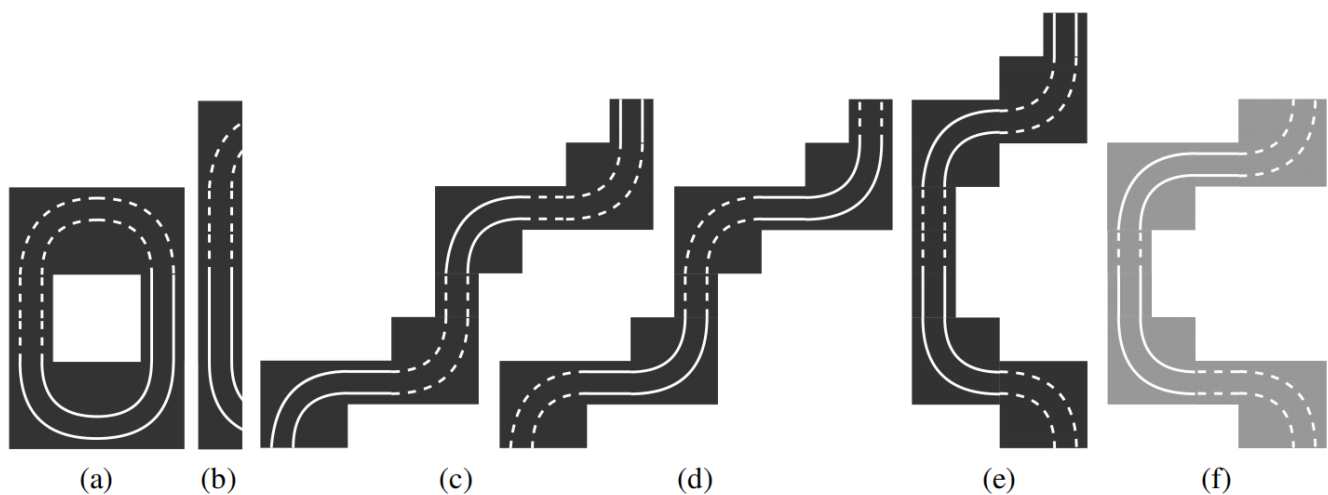


Figure 2: Overview of our track types for MoLane. (a) - (d) show the black version of the training and validation tracks. These tracks are also constructed using a light gray surface material. (e) and (f) depict our test tracks.

## Annotation

和TuSimple一样的格式，在固定的y值上给x

## Experiments

Table 3: Performance on the test set. Lane accuracy (LA), false positives (FP), and false negatives (FN) are reported in %.

| ResNet-18    | MoLane            |                  | LA                | TuLane<br>FP      | FN                | MuLane            |                   |                   |
|--------------|-------------------|------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
|              | LA                | FP & FN          |                   |                   |                   | LA                | FP                | FN                |
| UFLD-SO      | 89.39             | 25.25            | 87.43             | 34.21             | 23.48             | 88.02             | 50.24             | 26.08             |
| DANN [12]    | 87.65±0.48        | 29.97±1.21       | 88.74±0.32        | 32.71±0.52        | 21.64±0.65        | 86.01±0.67        | 55.33±1.22        | 36.30±1.90        |
| ADDA [13]    | 92.85±0.17        | 10.61±0.77       | 90.72±0.15        | 29.73±0.36        | 17.67±0.42        | 89.83±0.33        | 46.79±0.43        | 20.57±0.63        |
| SGADA [21]   | 93.82±0.10        | <b>7.13±0.22</b> | <b>91.70±0.13</b> | <b>28.42±0.34</b> | <b>16.10±0.43</b> | 90.71±0.10        | <b>45.13±0.32</b> | <b>17.26±0.36</b> |
| SGPCS (ours) | <b>93.94±0.04</b> | 7.16±0.16        | 91.55±0.13        | 28.52±0.21        | 16.16±0.26        | <b>91.57±0.22</b> | 45.49±0.63        | 17.39±0.88        |
| UFLD-TO      | 97.35             | 0.50             | 94.97             | 18.05             | 3.84              | 96.57             | 34.06             | 2.49              |
| ResNet-34    | LA                | FP & FN          | LA                | FP                | FN                | LA                | FP                | FN                |
| UFLD-SO      | 90.35             | 22.25            | 89.42             | 32.35             | 21.19             | 89.17             | 48.86             | 23.67             |
| DANN [12]    | 90.91±0.42        | 19.73±1.51       | 91.06±0.14        | 30.17±0.20        | 18.54±0.25        | 88.76±0.22        | 48.93±0.47        | 24.16±0.89        |
| ADDA [13]    | 92.39±0.26        | 12.17±0.84       | 91.39±0.16        | 28.76±0.30        | 16.63±0.36        | 90.22±0.39        | 45.84±0.54        | 19.49±0.90        |
| SGADA [21]   | 93.31±0.10        | 9.41±0.16        | 92.04±0.09        | 28.18±0.20        | 15.99±0.24        | <b>91.63±0.03</b> | <b>44.18±0.12</b> | <b>16.23±0.16</b> |
| SGPCS (ours) | <b>93.53±0.25</b> | <b>8.24±0.91</b> | <b>93.29±0.18</b> | <b>25.68±0.48</b> | <b>12.73±0.59</b> | 91.55±0.17        | 44.75±0.28        | 16.41±0.44        |
| UFLD-TO      | 97.21             | 0.30             | 94.43             | 20.74             | 7.20              | 96.54             | 33.76             | 2.03              |

LA – 对点的评估

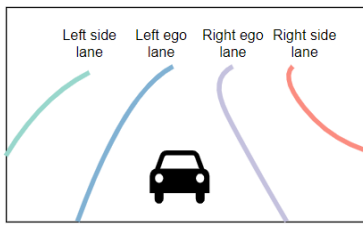
FP/FN – 对线的评估

## Ultra Fast Structure-aware Deep Lane Detection

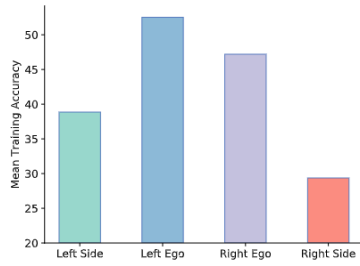
- 关注点: fast and global

### 混合车道表示法

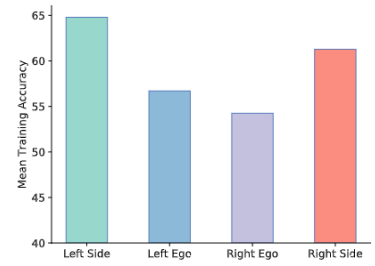
- 之前基于行的表示方式，不能适应所有类型的车道，会放大定位问题
-



(a) Lane definition



(b) Lane-wise Acc. with row anchor



(c) Lane-wise Acc. with column anchor

- 做了一个实验，首先道路分为ego lane和side lane，当基于行的表示时，比较弯曲的side lane的表示精度较低，当基于列的表示时，较为平直的ego lane精度较差。
- 这个现象称为magnified localization error problem
- 本文对于ego lane采取基于行的表示，对于side lane采用基于列的表示（CULane 和 TuSimple数据集只标了这四个车道）

## 有序分类

- 通过回归的方式预测车道线，不利于利用长程信息和全局信息，
- 本文的解决方案是采取基于分类的方法，使用不同的类别来表示不同的坐标
- 此处分类采取有序分类，即相邻的两个类别之间存在有序关系（例如，第8类车道坐标在空间上总是在第7类右侧）
- 这样也使得类别这个离散值现在变成了连续值，即7.5类这样的表述现在是有意义的
- 为了实现有序分类，使用两个损失函数一起来建模类之间的顺序关系，包括基本分类损失和数学期望损失，即因为现在类是连续的，所以可以采用数学期望而不是概率的最大值来表示类别

## 局限

- 车道数固定，并且混合行列表示也是固定的。（是预设值的超参数）。理论上对于弯曲车道预测有问题，即有个前提，车道线只能和anchor有一个交点，论文种对于多个交点采取底部交点优先的原则

## 学习目标

- $T^r, T^c$  分别表示基于行的车道表示和基于列的车道表示（即坐标矩阵）
- 假设  $T^r, T^c$  是标准化过的矩阵，除了用于表示没有点的-1之外其他都在0~1
- 做一个映射，将标准化之后的坐标值乘以类别总数，向下取整，得到每个坐标和类别标签的映射关系

- 这样就能将坐标学习问题转化为分类问题

$$\begin{cases} T_{cls\_i,j}^r = [T_{i,j}^r N_{dim}^r] \\ T_{cls\_m,n}^c = [T_{m,n}^c N_{dim}^c] \end{cases}$$

- s.t.  $i \in \{1, \dots, N_{row}\}, j \in \{1, \dots, N_{lane}^r\},$   
 $m \in \{1, \dots, N_{col}\}, n \in \{1, \dots, N_{lane}^c\},$

- 对于坐标不存在的情况 (-1) , 增加了一个二分类:

$$T_{ext\_i,j}^r = \begin{cases} 1, & \text{if } T_{i,j}^r \neq -1 \\ 0, & \text{otherwise} \end{cases},$$

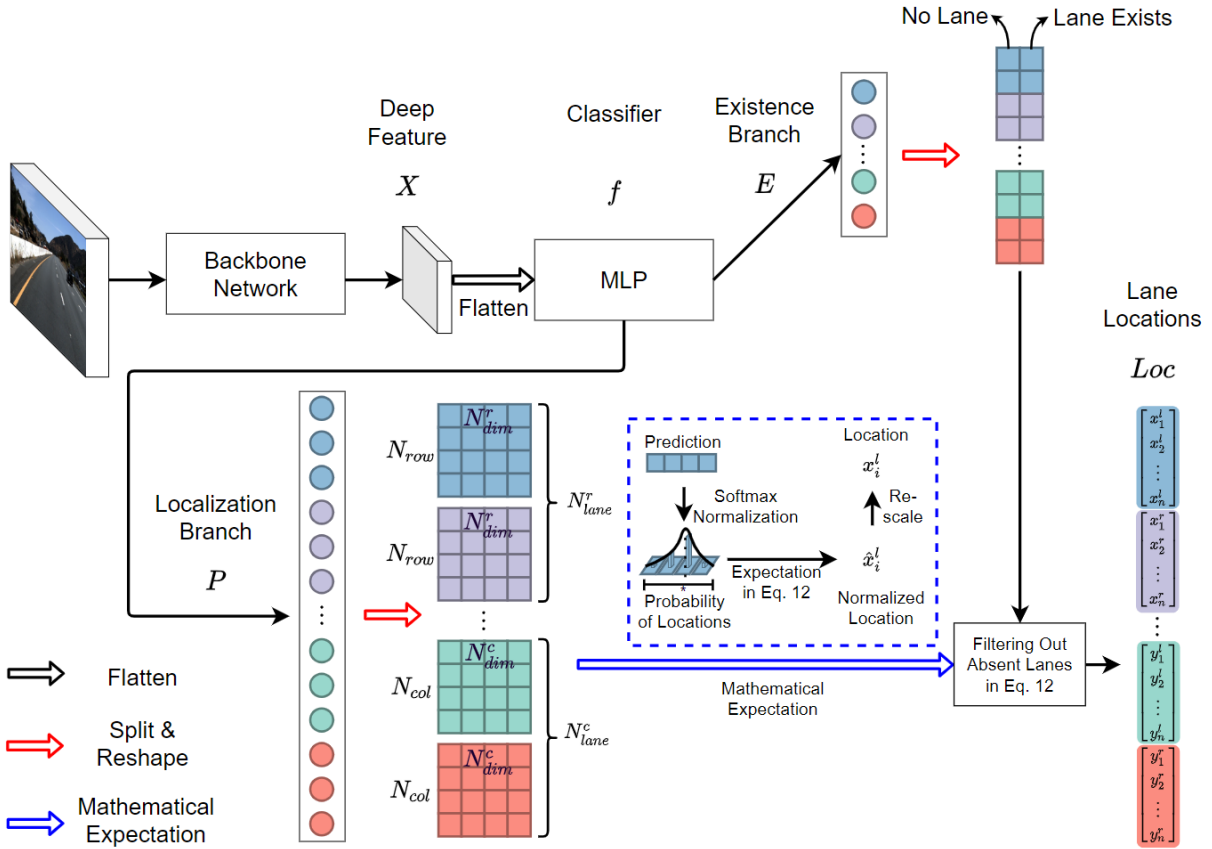
- 整个网络分为两个branch, 分别学习  $T_{cls}$  和  $T_{ext}$  , 即定位分支和存在性分支, 这两个分支的输出都是三维张量, 表示每个车道每个坐标的每个分类的概率:  $N_{lane}^r \times N_{row} \times N_{dim}^r$
- 同时这里提到, 提取出的图像特征直接展平输入到分类器, 而不是使用更常用的全局平均池化, 这样做的理由是, 空间信息对于有序分类极为重要, 而GAP会消除空间信息

## Loss

- 对于定位分支, 使用基本的分类损失和期望损失 (利于利用有序分类先验)
- 

$$L_{cls} = \sum_{i=1}^{N_{lane}^r} \sum_{j=1}^{N_{row}} L_{CE} (P_{i,j}^r, \text{onehot} (T_{cs\_i,j}^r)) + \sum_{m=1}^{N_{lane}^c} \sum_{n=1}^{N_{col}} L_{CE} (P_{m,n}^c, \text{onehot} (T_{cls\_m,n}^c))$$

- 基本分类损失就是交叉熵
- 期望损失主要利用了有序分类的先验, 因为已经得到了每个类别对应的概率  $P$  , 并且我们事先知道第k类就代表一个坐标k, 所以可以直接计算其期望作为坐标预测值, 然后将这个预测值与GT做L1 loss
- 对于存在分支, 就是预测出的二分类概率矩阵和GT做交叉熵
- 整个结构就是个非常简单的分类架构, 主干网络提取特征, 直接展平送入两个MLP分类器
-



## Experiments

TABLE 7  
Comparison of F1-measure and runtime on CULane testing set with IoU threshold=0.5.

| Method                   | Normal | Crowded | Dazzle | Shadow | No line | Arrow | Curve | Crossroad | Night | FPS | Total |
|--------------------------|--------|---------|--------|--------|---------|-------|-------|-----------|-------|-----|-------|
| SCNN [2]                 | 90.6   | 69.7    | 58.5   | 66.9   | 43.4    | 84.1  | 64.4  | 1990      | 66.1  | 8   | 71.6  |
| SAD [3]                  | 90.1   | 68.8    | 60.2   | 65.9   | 41.6    | 84.0  | 65.7  | 1998      | 66.0  | 75  | 70.8  |
| ERFNet-IntRA-KD [36]     | -      | -       | -      | -      | -       | -     | -     | -         | -     | 100 | 72.4  |
| SIM-CycleGAN [61]        | 91.8   | 71.8    | 66.4   | 76.2   | 46.1    | 87.8  | 67.1  | 2346      | 69.4  | -   | 73.9  |
| CurveLanes-NAS-S [6]     | 88.3   | 68.6    | 63.2   | 68.0   | 47.9    | 82.5  | 66.0  | 2817      | 66.2  | -   | 71.4  |
| CurveLanes-NAS-M [6]     | 90.2   | 70.5    | 65.9   | 69.3   | 48.8    | 85.7  | 67.5  | 2359      | 68.2  | -   | 73.5  |
| CurveLanes-NAS-L [6]     | 90.7   | 72.3    | 67.7   | 70.1   | 49.4    | 85.8  | 68.4  | 1746      | 68.9  | -   | 74.8  |
| RESA (ResNet-34) [35]    | 91.9   | 72.4    | 66.5   | 72.0   | 46.3    | 88.1  | 68.6  | 1896      | 69.8  | 45  | 74.5  |
| RESA (ResNet-50) [35]    | 92.1   | 73.1    | 69.2   | 72.8   | 47.7    | 88.3  | 70.3  | 1503      | 69.9  | 36  | 75.3  |
| LaneATT (ResNet-18) [46] | 91.1   | 72.9    | 65.7   | 70.9   | 48.3    | 85.4  | 63.3  | 1170      | 68.9  | 250 | 75.1  |
| LaneATT (ResNet-34) [46] | 92.1   | 75.0    | 66.4   | 78.1   | 49.3    | 88.3  | 67.7  | 1330      | 70.7  | 171 | 76.6  |
| SGNet [47]               | 91.4   | 74.0    | 66.8   | 72.1   | 50.1    | 87.1  | 67.0  | 1164      | 70.6  | 117 | 76.1  |
| FOLOLane [38]            | 92.7   | 77.8    | 75.2   | 79.3   | 52.1    | 89.0  | 69.4  | 1569      | 74.5  | 40  | 78.8  |
| UFLD(ResNet-18) [20]     | 87.7   | 66.0    | 58.4   | 62.8   | 40.2    | 81.0  | 57.9  | 1743      | 62.1  | 323 | 68.4  |
| UFLD(ResNet-34) [20]     | 90.7   | 70.2    | 59.5   | 69.3   | 44.4    | 85.7  | 69.5  | 2037      | 66.7  | 175 | 72.3  |
| UFLDv2(ResNet-18)        | 91.7   | 73.0    | 64.6   | 74.7   | 47.2    | 87.6  | 68.7  | 1998      | 70.2  | 330 | 74.7  |
| UFLDv2(ResNet-18)*       | 91.8   | 73.3    | 65.3   | 75.1   | 47.6    | 87.9  | 68.5  | 2075      | 70.7  | 310 | 75.0  |
| UFLDv2(ResNet-34)        | 92.5   | 74.9    | 65.7   | 75.3   | 49.0    | 88.5  | 70.2  | 1864      | 70.6  | 165 | 75.9  |
| UFLDv2(ResNet-34)*       | 92.5   | 74.8    | 65.5   | 75.5   | 49.2    | 88.8  | 70.1  | 1910      | 70.8  | 156 | 76.0  |