

面向嵌入式平台的车道线检测方法

杜中强¹, 唐林波¹, 韩煜祺^{2*}

- (1. 北京理工大学信息与电子学院 北京理工大学嵌入式实时信息处理技术北京市重点实验室, 北京 100081;
2. 清华大学 计算机科学与技术系, 北京 100084)

摘要: 车道线检测在自动驾驶和高级辅助驾驶中起着举足轻重的作用,然而,传统的车道线检测技术鲁棒性较差,而大多数基于深度学习的方法复杂度又较高,难以在嵌入式平台实时应用。提出一种面向嵌入式平台的轻量级车道线检测网络,将车道线检测转化为语义分割问题,该网络借鉴 U-Net 与 Segnet 网络结构,使用了小尺度卷积等轻量化组件设计计算高效的语义分割网络。在检测车道线的基础上,计算车辆距离两侧车道线的距离,以及车道线的曲率,同时当车辆偏离车道线或检测出现异常时进行预警,最后将整个系统移植到海思平台。实验结果表明:该系统具有较高的检测精度以及检测速度,准确率达到 97.5%,速度达到 50 FPS,满足实时性要求,因此该系统能够用于面向嵌入式平台的实时车道线的检测、测距、曲率计算以及预警。

关键词: 车道线检测; 嵌入式平台; 深度学习; 语义分割; 自动驾驶

中图分类号: TP751 **文献标志码:** A **DOI:** 10.3788/IRLA20210753

Lane line detection method for embedded platform

Du Zhongqiang¹, Tang Linbo¹, Han Yuqi^{2*}

- (1. Beijing Key Laboratory of Embedded Real-time Information Processing Technology, School of Information and Electronic, Beijing Institute of Technology, Beijing 100081, China;
2. Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)

Abstract: Lane line detection plays a pivotal role in autonomous driving and advanced assisted driving. However, traditional lane line detection technology was less robust, and most methods based on deep learning were more complex and difficult to embed platform real-time application. A lightweight lane line detection network for embedded platforms was proposed, which converts lane line detection into a semantic segmentation problem. The network draws on U-Net and Segnet network structures, and uses small-scale convolution and other lightweight components to design and calculate efficiently semantic segmentation network. Based on the detection of the lane line, calculate the distance between the vehicle and the lane line on both sides, as well as the curvature of the lane line, and give an early warning when the vehicle deviates from the lane line or the detection was abnormal. Finally, the entire system was transplanted to the HiSilicon platform. Experimental results show that the system has high detection accuracy and detection speed, the accuracy rate reaches 97.5%, the speed reaches 50 FPS, and meets real-time requirements. Therefore, the system can be used for real-time lane line

收稿日期:2022-01-20; 修订日期:2022-02-15

基金项目:中国博士后科学基金(2021TQ0177)

作者简介:杜中强,男,硕士生,主要从事数字图像处理、目标检测与识别等计算机视觉方面的研究。

导师简介:唐林波,男,讲师,硕士生导师,博士,主要从事嵌入式实时图像处理、智能目标识别与跟踪方面的研究。

通讯作者:韩煜祺,男,博士后,博士生,主要从事遥感图像处理及计算机视觉等方面的研究。

detection, ranging, and distance measurement for embedded platforms. Curvature calculation and early warning.

Key words: lane line detection; embedded platform; deep learning; semantic segmentation; autonomous driving

0 引言

随着人工智能的快速发展,无人驾驶成为了国内外交通行业发展的新趋势。伴随着计算机软硬件的不断更新迭代,深度学习在诸多方面取得了成功应用,特别是在 U-net^[1]、Segnet^[2] 等语义分割模型中的应用,使得利用深度学习方法进行车道线检测也得以实现。目前关于车道线检测方法较多,对于传统的车道线检测方法,何旭光等^[3] 提出利用 Hough 变换的方法来检测车道线,该方法考虑了道路的特点即车道线边界平行的性质;李后隆等^[4] 提出用改进的 Hough 变换方法进行车道线检测,该方法利用极角约束 Hough 变换检测车道线的初始位置;胡晟圆等^[5] 提出了结合 Hough 变换与运动估计的车道线提取方法,该方法巧妙地将分块 Hough 变换和图像块的运动估计相结合,极大地降低了算法的复杂度;周宏宇等^[6] 提出基于 Haar 特征耦合级联分类器的车道线检测方法,通过提取感兴趣的区域 (ROI),借助 LSD 线段检测器,结合生长策略拟合车道线。然而,这些传统的车道线检测方法都是利用手工设计的特征,鲁棒性较差,难以适用于复杂多变的真实场景。对于基于深度学习的车道线检测方法,邓天民等^[7] 等提出基于改进 Segnet 的车道线检测方法;L. Haixia 等^[8] 提出了一种使用级联卷积神经网络结构进行端到端车道检测的方法,首先对车道边界进行实例分割,然后通过线性分类器对车道分类进行分类;Q. Zou 等^[9] 结合卷积神经网络 (CNN) 和循环神经网络 (RNN) 提出了一种混合深度架构;W. Van Gansbeke 等^[10] 提出了一种以端到端的方式训练车道检测器的方法,直接回归车道参数。虽然这些基于深度学习的车道线检测方法检测准确率较高,但也较为复杂,难以移植到嵌入式平台做到实时检测。如何准确、快速地检测车道线已成为无人驾驶汽车领域需要解决的重要问题。

针对车道线检测准确率和鲁棒性的要求,文中将车道线检测转化为语义分割问题,提出一种面向嵌入式平台的车道线检测网络。首先,该网络借鉴 U-Net 与 Segnet 的编码-解码结构,使用了小尺度卷积等轻量化组件设计计算高效的语义分割网络。其次,在准

确检测车道线的基础上,使用滑动窗口曲线拟合策略拟合车道线的曲线方程,计算车辆中央距两侧车道线的距离,以及车道线的曲率,同时当车辆偏离车道线或检测出现异常时进行预警。最后,将上述完整车道线检测系统移植到海思平台。

1 系统设计与实现

1.1 系统概述

该系统通过安装在车辆前端的摄像头实时检测车道线,并计算车辆与两侧车道线的距离,以及车道线的曲率。同时当车辆偏离车道线或检测出现异常时则进行报警。系统的框图如图 1 所示。

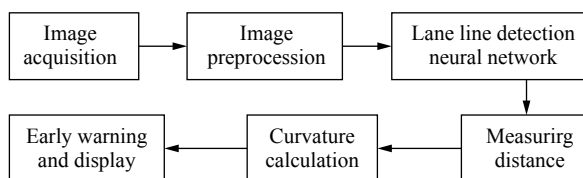


图 1 系统框图

Fig.1 Block diagram of system

由此可以看出,该系统由图中几个模块组成,其中图像采集部分是采用 ADAS 系统对视频进行采集,采集的帧频为 25 HZ,图像大小为 1280×720 pixel 或 1920×1080 pixel;图像预处理部分对图像进行图像增强等处理,使图像特征更加明显,易于后面的处理,并且改变图像的大小,以满足车道线检测部分神经网络的输入图片大小的要求;测距部分是在神经网络检测出车道线之后,通过检测出的车道线可以计算出车辆的中间位置到两侧车道线的距离;曲率计算部分是在神经网络检测出车道线之后,通过检测出的车道线可以计算出车道线的曲率;预警部分是根据设定的阈值,当车辆偏离车道线或检测出现异常时进行提醒;图像处理显示部分是把检测出的车道线、测量的车道线的距离以及计算出的车道线的曲率显示在原始的图像上面;由于车道线检测以及测距系统最终要移植到海思平台,因此需要满足实时检测的要求。

1.2 图像预处理

在图像预处理中,分别对输入图像的 3 个通道,

使用 3×3 的滑动窗口进行中值滤波, 去除干扰噪声, 然后对滤波后的图像进行尺度变换, 以满足神经网络对输入图片大小的要求, 在进行尺度变换时, 采用双线性插值的方法。

1.3 车道线检测神经网络

针对利用深度学习进行车道线检测方法采用的骨干网络计算复杂度高, 难以在资源受限的嵌入式平台实时应用, 以及检测效果有限的问题。在考虑了 U-net 和 Segnet 网络结构的优点后, 通过分析认为, 可

以借鉴 U-net 与 Segnet 来设计一种新的网络来满足在嵌入式平台实现实时车道线检测, 并保证检测的准确性。借鉴 U-net 和 Segnet 网络设计车道线检测网络时, 分别考虑了编解码器-解码器的层数、最基本的卷积单元、卷积核大小, 上采样方式及初始化方式等。

改进后的车道线检测神经网络结构如图 2 所示, 将车道检测建模为语义分割任务, 能够实现目标的像素级语义分割, 具有参数量少、运算速度快的特点, 满足车道线检测的实时性和准确性要求。

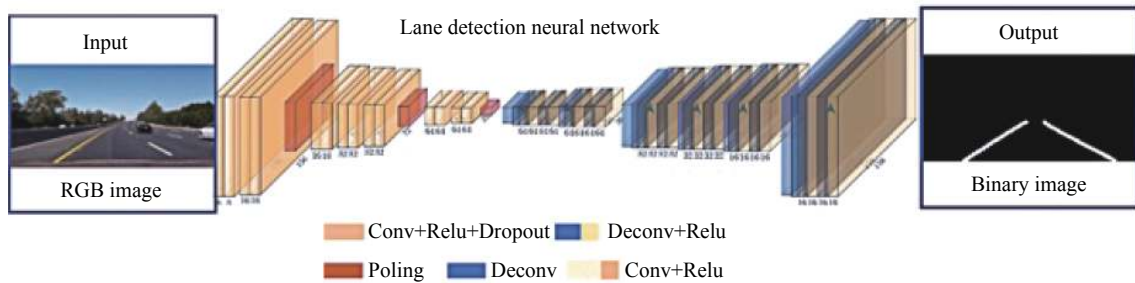


图 2 车道线检测神经网络结构

Fig.2 Lane line detection neural network structure

1.3.1 编码器网络实现

分析 Segnet 和 U-net 网络, 再根据它们的结构优点, 所设计的编码器网络结构如图 3(a) 所示, 为进一步提高计算速度, 减少了网络层数, 把下采样次数缩减为 3 次, 只使用 3 个最大池化实现下采样, 使网络可在相对较小的训练集中进行训练。同时为了防止

过拟合, 以及考虑到嵌入式平台移植实时性的问题, 使用了 Dropout 层来代替 batch normalized 层, 采用的卷积核仍为 3×3 , 池化层窗口的大小仍为 2×2 。

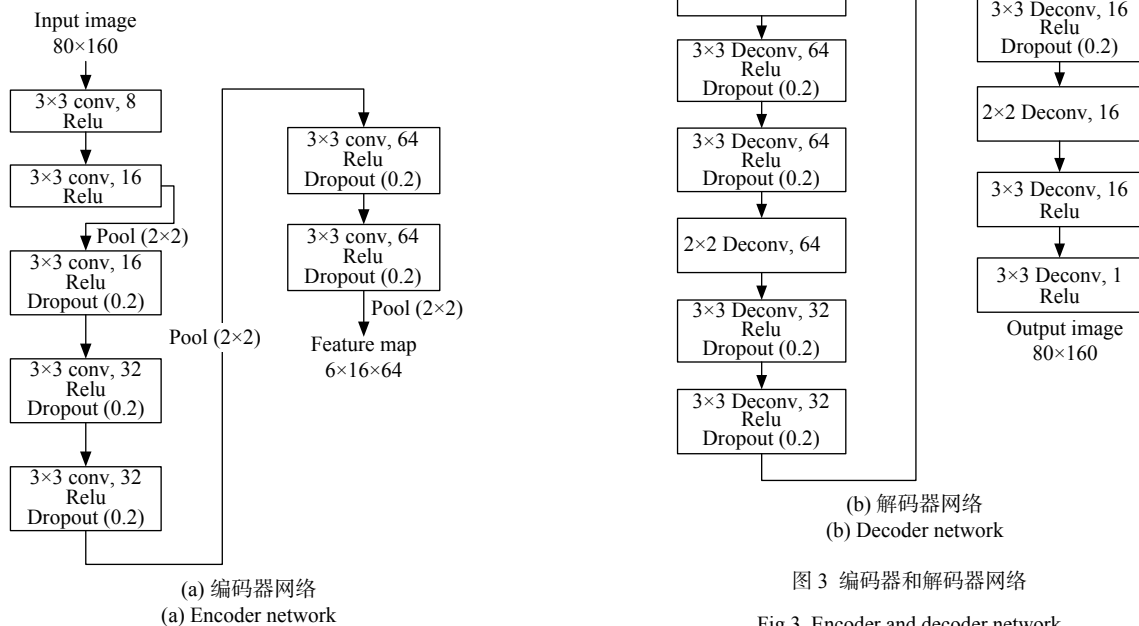


图 3 编码器和解码器网络

Fig.3 Encoder and decoder network

1.3.2 解码器网络实现

解码器网络如图 3(b) 所示, 使用 3×3 卷积核的转置卷积和 Relu 激活函数, 同时使用 2×2 卷积核的转置卷积实现上采样, 在上采样时使用双线性插值的初始化方法, 以提高神经网络的训练效果以及训练速度, 最后一层通过使用 3×3 的转置卷积层和一个 Relu 层, 得到像素级的分类结果。

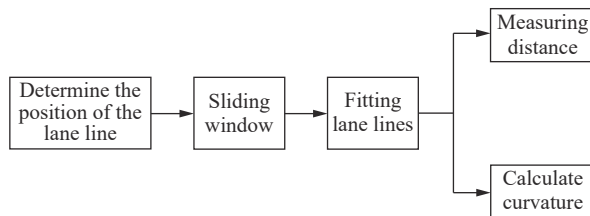
1.4 曲率计算与测距

曲率计算与测距模块系统框图如图 4(a) 所示, 在检测出车道线后, 就可以测量车辆距离两侧车道线的距离。首先对车道线检测神经网络输出的灰度图像做中值滤波, 滤除干扰噪声, 然后通过设置合适的阈值, 生成二值图像, 如图 4(b) 所示。

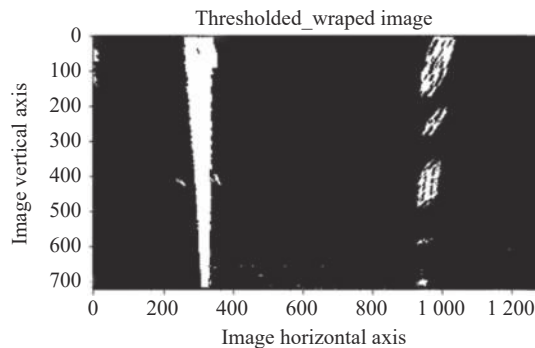
在检测出车道线后, 需确定车道线在图像中的位置, 首先要确定车道线在图像水平方向的位置, 由于车道线所在的像素都集中在一定范围内, 如图 4(c) 所示, 从水平方向上将图像以中心为轴一分为二, 左右两部分图像在水平方向上的像素分布峰值就是车道线集中的位置, 然后, 对两部分图像在水平方向做直方图统计, 从峰值的位置开始通过滑动窗口搜寻整个车道线, 如图 4(d) 所示。

在拟合车道线时, 采用滑动窗口的方法, 首先, 设置滑动窗口的高和宽, 其次, 从直方图峰值的位置开始, 以当前位置为中心, 做一个网格化搜寻, 其中初始位置为 x , 宽度可以人为设定, 高度根据图片大小除以搜寻窗口数目计算得到, 这里假设窗口数目为 4。然后, 对每个搜寻窗口分别做水平和垂直方向的直方图统计, 统计出每一个搜索框区域内非零像素的个数, 通过设置阈值 T , 过滤掉非零像素数目小于 T 的框。最后, 通过计算每一个搜索框中非零像素坐标的均值, 把均值作为当前搜索框的中心, 考虑到计算复杂度, 以及大多数路况以直道和弯道为主, 因此采用二阶多项式拟合这些搜索框的中心点, 就可以得到当前车道线的二阶多项式曲线参数, 在利用二阶多项式逼近后, 会得到一个曲线方程: $y = ax^2 + bx + c$ 。

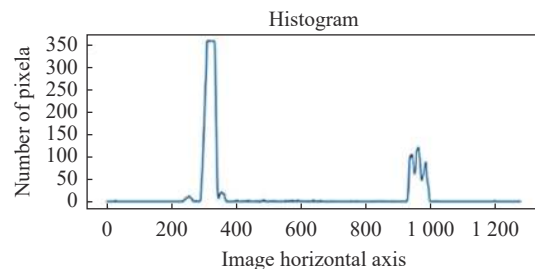
在检测出左右两条车道线后, 需进行测距和曲率计算, 由于 ADAS 采集系统安装在汽车的正中央, 因此把图像的正中央作为车的中央即 x_0 , 通过选取 y 值, 就可以得到车道线水平位置, x_1, x_2 , 从而计算出车道线的距离, $d_1 = |x_1 - x_0|$, $d_2 = |x_2 - x_0|$ 。同时可以利用公式 (2), 计算出右侧车道线的曲率。



(a) 曲率计算与测距模块框图
(a) Block diagram of curvature calculation and ranging module



(b) 车道线二值图像
(b) Lane line binary image



(c) 车道线二值图像直方图
(c) Lane line binary image histogram



(d) 车道线曲线拟合过程
(d) Track curve fitting process

图 4 车道线拟合

Fig.4 Lane line fitting

假设车道线的二次拟合曲线为:

$$y = ax^2 + bx + c \tag{1}$$

则在 x m 远处的车道线的曲率半径为:

$$R = \frac{(1 + (2ax + b)^2)^{1.5}}{2a} \tag{2}$$

由于计算时是以像素为单位, 而计算出的距离应

当是实际距离,因此需找出实际距离与像素之间的关系。通过测量分析,分别计算出水平方向和垂直方向对应的比例关系分别为:水平方向: $m/\text{pixel} = 30/720$;垂直方向: $m/\text{pixel} = 3.7/700$ 。

考虑到车道线被遮挡、人行横道、十字路口等特殊情况对车道线检测结果的影响,从而影响测距和曲率计算,结合我国道路建设规划中规定道路宽度为 3.75 m,基于该数据对测量距离进行判断,如果测量出的距离不合理,则舍弃,保留合理数据。

1.5 预警

考虑到驾驶场景的复杂性,以及确保驾驶的安全可靠,需对测量结果进行判断,并预警。车辆位于车道中央行驶时,根据测距部分测量出距离 d_1 、 d_2 ,若 $d_1 < d_2$,则判断为左偏;若 $d_1 > d_2$,则判断为右偏;其他情况为正常驾驶,同时应设置阈值 T ,当 $d < T$ 时,进行预警,表示车辆即将偏离车道线。同时为避免测量结果出现异常,设置阈值 T_2 ,当 $d > T_2$ 时,表示测量结果受到十字路口、人行横道、前方车辆等因素干扰,此时测量结果不可信,应及时剔除,以达到准确预警的效果。

1.6 海思平台移植

海思芯片中有针对神经网络进行加速处理的硬件单元 NNIE (Neural Network Inference Engine),由于目前 NNIE 配套软件及工具链仅支持 Caffe 框架,使用其他框架的网络模型需要转化到 Caffe 框架下。因此笔者的神经网络采用 Caffe 框架进行设计,将训练好的模型通过 RuyiStudio 进行测试、仿真,且生成 WK 文件,最终移植到海思平台 Hi3559 CV100,见图 5。



图 5 海思 Hi3559 CV100

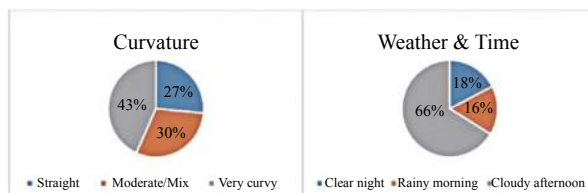
Fig.5 Hisilicon Hi3559 CV100

2 实验验证与结果分析

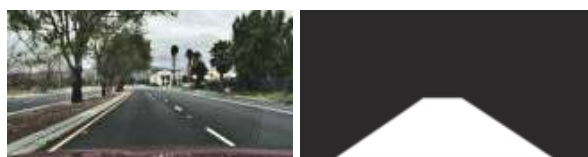
2.1 数据集

目前关于车道线检测的数据集较多,包括 CULane、Caltech Lanes、Mvirgo、TuSimple 数据集等。

Mvirgo 数据集是由 Michael Virgo 进行拍摄并标注,是专门为车道线检测算法的研发测试建立的数据集,总共包含 12764 张图片,数据集内包括直道、弯道、晴天、雨天、多云等多种场景的车道线数据,各种场景的数据分布如图 6(a) 所示,原始的图片 and 标签分别如图 6(b)~(c) 所示,对数据集进行处理后的图片和标签分别如图 6(d)~(e) 所示。

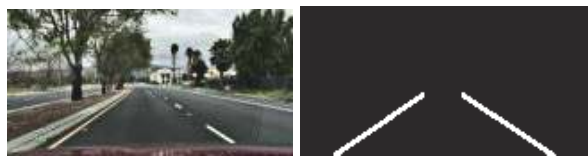


(a) 数据集分布
(a) Data sets distribution



(b) Mvirgo 原始数据集
(b) Mvirgo original data sets

(c) Mvirgo 原始标签
(c) Mvirgo original label



(d) Mvirgo 数据集
(d) Mvirgo data sets

(e) 新标签
(e) New label

图 6 数据集

Fig.6 DataSets

2.2 不同数据集训练对比

在确定了神经网络结构后,对 TuSimple 数据集, CULane 数据集和 Mvirgo 数据集进行训练,并测试,测试结果如表 1 和图 7 所示,可以看到 Mvirgo 准确率最高,效果也最好。

由于 3 个数据各有特点,因此测试的准确率也各不相同,其中 CULane 数据集是由行车记录仪拍摄的照片,包含部分车身,车道线存在遮挡,因此测试效果最差,准确率也最低。TuSimple 数据集训练效果也不

表 1 不同数据集的准确率

Tab.1 Accuracy of different data sets

Data sets	TuSimple	CULane	Mvirgo	TuSimple + Mvirgo
Accuracy	97.5	93.21	98.23	97.8

好, 主要原因是 Tusimple 数据集拍摄的图片较暗, 车道线不明显, 对神经网络的训练会产生一定的影响, 导致检测准确率较低, 对虚线的测量效果较差, 虚警率高, 测试结果如图 7(b) 所示。而 Mvirgo 数据集检

测效果较好, 主要原因是该数据集包含较多的场景, 图片质量较高, 缺点是数据集数据量较少, 并且只标注了当前道路的两条车道, 对于一些特殊情况, 无法处理, 测试结果如图 7(c) 所示。另外, 结合 Tusimple 数据集和 Mvirgo 数据集进行了训练, 训练结果与只用 Mvirgo 数据集训练结果相似, 测试结果如图 7(d) 所示, 但对虚线的检测结果较差, 因此选择了 Mvirgo 数据集。

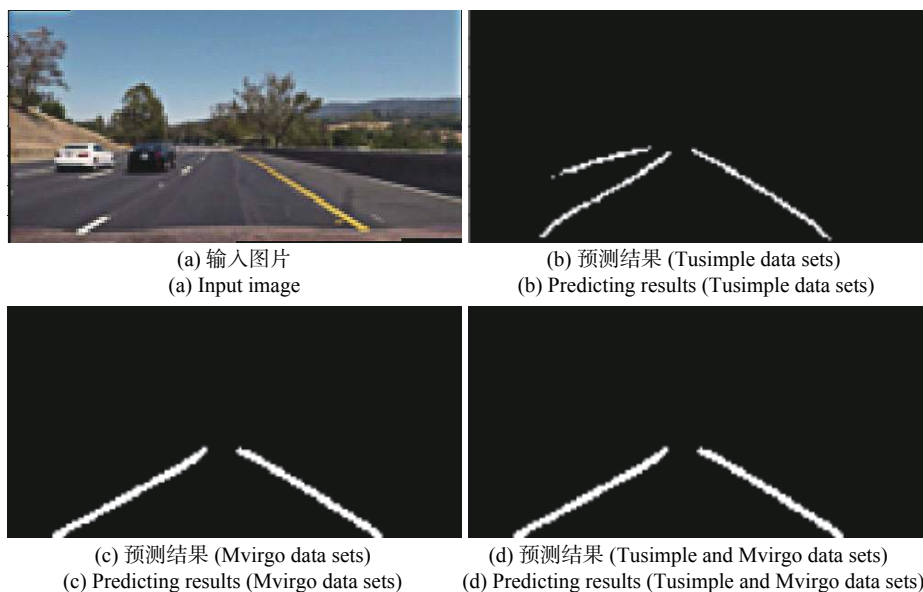


图 7 不同数据集的训练结果

Fig.7 Training results of different data sets

2.3 不同车道线检测网络结构的对比

笔者对比了现有的车道线检测网络, 其中时间可能由于设备差异, 存在区别, 结果如表 2 所示, 其中 SegNetConvLSTM 准确率最高, 但是运行速度较慢,

无法移植到海思平台做到实时检测, 笔者提出的网络准确率与 SegNetConvLSTM 相当, 但是速度方面具有明显的优势, 可以移植到海思平台做到实时检测, 在海思平台上的检测速度达到 50 FPS, 满足系统的要求。

表 2 不同车道线检测网络结构的对比

Tab.2 Comparison of different lane line detection network structures

Method	Accuracy	Time/ms	Data sets
Segnet ^[9]	96.78	5.2	tuSimple
Unet ^[9]	96.46	4.6	tuSimple
SegNetConvLSTM ^[9]	98.52	25	tuSimple
UNetCpvnLSTM ^[9]	98.46	20.3	tuSimple
Ours	97.5	1.2	tuSimple
Enet-SAD ^[11]	96.64%	13.4	tuSimple
LaneNet ^[12]	96.38%	16.6	tuSimple
ERFNet ^[10]	95.80%	14.1	tuSimple

2.4 结果分析

考虑到驾驶场景的复杂性和多样性,选择多种场景的数据进行验证,测试结果如图 8 所示,结果表明该车道线检测方法在大多数场景中检测效果较好,但

在人行横道、十字路口、车道线被完全遮挡的情况下表现不佳,针对这种情况,通过测距和设置阈值的方法,可以准确判断,依然能达到准确预警的效果。



图 8 车道线检测结果

Fig.8 Lane line detection result

3 结论

文中借鉴 Unet 和 Segnet 网络结构,提出了一种轻量级的车道线检测网络,该网络具有较强的鲁棒性,并且能够移植到海思平台,实现实时检测,准确率达到 97.5%,速度达到 50 FPS,满足实时性要求。在车道线检测的基础上实现了测量车辆到两侧车道线

的距离,以及车道线的曲率,同时当车辆偏离车道线或者检测出现异常时进行预警。

与其他网络相比,文中所提出的车道线检测网络具有较高的准确率与速度,此外,在具有挑战性的驾驶场景的数据集上对该网络进行测试,以检查其鲁棒性。结果表明,该网络能够在大多数场景中稳定准确

检测车道线,并能很好地避免虚警。

针对人行横道、十字路口、车道线被全遮挡情况下表现不佳的问题,后续将通过结合车辆行人检测和交通标志检测,优化车道线检测方法,进一步提升检测性能。

参考文献:

- [1] Ronneberger O, Fischer P and Brox T. U-net: convolutional networks for biomedical image segmentation[C]// International Conference on Medical Image Computing and Computer-assisted Intervention, 2015: 234-241.
- [2] Badrinarayanan V, Kendall A, Cipolla R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39(12): 2481-2495.
- [3] He Xuguang, Jiang Lei, Luo Yiping, et al. Lane line detection algorithm design based on Hough transform [J]. *Agricultural Equipment and Vehicle Engineering*, 2019, 57(11): 90-91, 107. (in Chinese)
- [4] Li Houlong, Ma Liming, Zhong Linwei. Lane line detection based on improved hough transform [J]. *Practical Automobile Technology*, 2021, 46(7): 16-18. (in Chinese)
- [5] Hu Shengyuan, You Bo, Cui Lidong. Lane detection and departure warning simulation based on improved Hough transform [J]. *New Industrialization*, 2021, 11(5): 122-125. (in Chinese)
- [6] Zhou Hongyu, Song Xu, Liu Guoying. Lane detection algorithm based on Haar feature coupling cascade classifier [J]. *Computer Engineering and Design*, 2020, 41(6): 1719-1724. (in Chinese)
- [7] Deng Tianmin, Wang Lin, Yang Qizhi, et al. Lane line detection method based on improved SegNet algorithm [J]. *Science Technology and Engineering*, 2020, 20(36): 14988-14993. (in Chinese)
- [8] Haixia L, Xizhou L. Flexible lane detection using CNNs[C]//2021 International Conference on Computer Technology and Media Convergence Design (CTMCD), 2021: 235-238.
- [9] Zou Q, Jiang H, Dai Q, et al. Robust lane detection from continuous driving scenes using deep neural networks [J]. *IEEE Transactions on Vehicular Technology*, 2020, 69(1): 41-54.
- [10] Gansbeke W V, Brabandere B D, Neven D M, et al. End-to-end lane detection through differentiable least-squares fitting[C]//2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 2019: 905-913.
- [11] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, et al. ENet: A deep neural network architecture for real-time semantic segmentation [J]. *Computer Science*, 2016, arXiv: 1606.02147.
- [12] Neven D, Brabandere D B, Georgoulis S, et al. Towards end-to-end lane detection: An instance segmentation approach[C]// 2018 IEEE Intelligent Vehicles Symposium (IV), 2018: 286-291.