

# 基于视觉的嵌入式路面标识检测算法研究

许大展, 吴晓雨<sup>1</sup>

(中国传媒大学信息与通信工程学院, 北京 100024)

**摘要:** 路面标识为无人驾驶提供重要的道路视觉信息, 路面标识的正确识别是行车安全的前提。由于实际道路场景的复杂多变性, 传统路面标识检测方法在嵌入式平台下算法鲁棒性和实时性方面仍面临着一些挑战。本文提出了嵌入式平台 Jetson TX2 下路面标识检测网络模型及优化算法: 首先给出了基于层合并的区域全卷积网络 R-FCN(Region-based Fully Convolutional Networks)的简化模型, 实现了路面标识的高精度检测; 接着, 为了满足实际应用中实时推理需求, 将简化的 R-FCN 网络模型部署在嵌入式平台 NVIDIA Jetson TX2 上, 构建了基于 TensorRT 的模型推理优化加速方法, 在嵌入式平台上实现了快速准确路面标识算法。提出的算法在自建路面标识库和相应的公开数据库进行了测试, 实验结果验证了算法的有效性。

**关键词:** 路面标识检测; 模型加速; 嵌入式平台 TX2; TensorRT

中图分类号: TP391.4

文献标识码: A

## Embedded Road Marking Detection Algorithm Based on Vision

XU Dazhan, WU Xiaoyu

(School of Information and Communication Engineering, Communication University of China, Beijing 100024, China)

**Abstract:** The road marking provides important road visual information for autonomous driving, so the correct identification is a prerequisite for driving safety. Due to the complex variability of real road scenes, the traditional road marking detection methods still face some challenges in terms of algorithm robustness and real-time performance of the embedded platform. So this paper proposes the road marking detection network model and optimization algorithm deployed on the embedded platform TX2: firstly, the high-precision detection of road marking is realized based on layer integration to simplify R-FCN(Region-based Fully Convolutional Networks); then, in order to meet the real-time reasoning requirements in practical applications, the simplified R-FCN network model is deployed on the embedded platform NVIDIA Jetson TX2, and the model inference optimization acceleration method based on TensorRT is proposed, which accelerates the inference phase of the network model. A fast and accurate road marking detection algorithm is implemented on the embedded platform. The proposed algorithm is verified in the self-built road marking database and the related public dataset, and the experimental results proved the effectiveness of this algorithm.

**Key word:** road marking detection; model acceleration; embedded platform TX2; TensorRT

基金项目: 国家自然科学基金项目资助 (61801441)

作者简介: 许大展 (1994-), 男, 硕士研究生. E-mail: 2749328959@qq.com

吴晓雨 (1979-), 女, 博士, 教授, 主要研究领域: 计算机视觉、智能视频分析. Email: wuxiaoyu@cuc.edu.cn

# 1 引言

随着人工智能的快速发展,传统汽车行业与信息技术的结合,无人驾驶逐渐出现在人们的视野中。路面标识作为无人驾驶和辅助驾驶的重要视觉信息,对它实时、正确的识别是行车安全的必要条件。

路面标识检测隶属于目标检测技术范畴,传统的目标检测方法<sup>[1][2]</sup>主要是基于目标轮廓、目标特征的检测方法,需要人为设计特征,存在一定的局限性。基于深度学习的目标检测方法<sup>[3][4][5]</sup>,避免了人为设计特征因素的影响,模型的泛化能力有较好的提升,但计算代价较高,收集的数据全部上传到云端处理将会带来较大时延,无法满足低延时的应用需求。嵌入式终端计算设备更接近数据来源处,可对数据进行及时处理,减少云计算带来较大时延的问题,但是嵌入式终端计算能力有限,无法实时处理复杂模型。因此,研究既准又快路面标识检测算法,既能处理路面标识受损、阴影、遮挡和天气等因素,又能适应车载终端硬件性能,是一项非常具有挑战性且应用性的工作。

针对以上问题,本文提出了一种基于深度学习的路面标识检测模型,能够在复杂场景下实现对路面标识的正确定位和分类,同时结合满足实际应用的需求,将模型部署在嵌入式 AI 计算设备 Jetson TX2 上,并基于 NVIDIA 推出的 TensorRT 对网络模型设计了优化加速和前向推理算法,以提高满足模型在实际应用中的计算速度。

## 2 相关工作

路面标识检测是实现无人驾驶技术的重要环节之一,受到了许多专家学者的关注和研究。Wei Liu 等人<sup>[1]</sup>使用传统的检测方法实现了多类别的路面标识检测,利用 Haar-lik 特征 Adaboost 分类器将不存在标识的区域剔除,并用 BW-HOG 特征的极限学习机 ELM(Extreme Learning Machine)分类器进行具体分类。该方法可实现较好的检测率,但不能满足实时性的要求。Tairui Chen 等人<sup>[4]</sup>提出了一种二值化归一梯度的路面标识检测方法,并使用 PCANet 分类器对检测出的目标进行分类。该方法首次使用了深度学习的检测方法对路面标识进行识别,与传统方法相比有较好的效果。Eduardo Romera 等人<sup>[5]</sup>将目标分割与检测的方法用于路面标识识别,该方法可实现端到端的训练,能学到图像的关键信息,不仅可检测出路面标识,还可检测出道路、车道线、车辆、行人,能实现像素级别的分割,但与此同时消耗了大量的计算时间,不能满足实时性的要求。TT-CNN 方法<sup>[6]</sup>在 OverFeat<sup>[7]</sup>的基础上进行改进的对路上的交通标识的识别网络,但该算法较 Faster RCNN<sup>[8]</sup>等方法计算耗时更长。基于分割的 SegNet<sup>[9]</sup>、Mask-RCNN<sup>[10]</sup>和 RPM-Net<sup>[11]</sup>网络虽可以得到较精确的分割结果,但是计算速度上无法满足应用场景区的实时性要求。综上,上述方法借助云端进行路面标识的检测,将网络模型直接部署到嵌入式终端平台后因嵌入式平台硬件资源有限,这些方法不能实现网络的实时性推理,无法满足实际应用需求。

网络模型被部署到嵌入式终端平台前需要对网络模型进行压缩和加速,以提高模型推理速度。现有的深度学习模型压缩方法主要分为四类:参数修剪和共享<sup>[12]</sup>、低秩因子分解<sup>[13]</sup>、紧凑卷积滤波器<sup>[14]</sup>、知识蒸馏<sup>[15]</sup>。基于参数修剪和共享的方法针对模型参数的冗余性,试图去除冗余和不重要的项,可以从预训练模型或从头开始训练,较为灵活有效。基于低秩因子分解的技术使用矩阵/张量分解来估计深度学习模型的信息参数。基于紧凑卷积滤波器的方法设计了特殊的结构卷积滤波器来降低存储和计算复杂度。知识蒸馏方法通过学习一个蒸馏模型,训练一个更紧凑的神经网络来重现一个更大的网络的输出。以上方法大多在模型训练阶段进行,实现模型的压缩和加速。本文重点研究嵌入式设备 Jetson TX2 的路面标识检测算法,Jetson TX2 嵌入式设备是 NVIDIA 公司推出的一款超好性能、低功耗的超级计算机<sup>[16]</sup>,适用于智能型尖端装置,如机器人、无人机、可携式医疗设置等,可为终端提供人工智能技术支持<sup>[17][18]</sup>,该平台采用 TensorRT 推理引擎,以上提到的模型加速方法并不直接适用于该引擎。

综上,目前已公开的无论是传统的检测方法还是基于深度学习的检测方法,不能同时满足复杂

道路环境下路面标识的检测精度以及嵌入式平台下实时性要求。因此，针对当前路面标识识别方法存在的缺陷，本文重点研究了嵌入式平台 Jetson TX2 下快速准确的路面标识检测算法，首先提出了基于层合并的 R-FCN 简化模型，而后将模型部署到嵌入式平台 TX2，提出了基于 TensorRT 网络模型的前向推理优化加速方法，提出的算法在自建路面标识库和公开数据库均取得了较好的测试结果。

### 3 路面标识检测网络模型及其简化

本文基于深度学习的目标检测方法，针对路面标识的特点，从算法的准确性要求角度选用 R-FCN<sup>[19]</sup>网络作为实现对路面标识识别的基础网络，并采用了批量归一化 BN(Batch Normalization)层与卷积层参数融合的改进策略，减少模型内存占用，提升推理速度。

#### 3.1 网络模型

R-FCN 沿用了 Faster-RCNN 中的 RPN 网络结构，与之不同的是 R-FCN 使用了无全连层的 ResNet<sup>[20]</sup>网络。更重要的是 R-FCN 引入了位置敏感得分图的概念，解决了分类需要满足平移不变性和目标识别需要满足平移可变性之间的矛盾。R-FCN 在特征提取部分共享，极大减小了计算量，较 Faster-RCNN 网络在速度上有明显的提升，在检测准确率方面在检测准确率方面明显优于 one-stage 的 SSD<sup>[21]</sup>和 FSSD<sup>[22]</sup>等方法。

在满足准确性要求的前提下，为了降低由网络层数的加深所带来的巨大时间消耗，本文将 R-FCN 的基础网络 ResNet50 替换为 ResNet18，降低网络的深度以达到网络加速的目的。

#### 3.2 网络结构简化

R-FCN 模型中的 BN 层在训练阶段对数据进行归一化，加速网络收敛，但在前向推理过程中占用大量内存，降低了推理速度。且 BN 层和卷积层都是线性变换，因此为了加快推理速度，将上述基于 R-FCN 的路面标识检测网络模型中的 BN 层与其相连的卷积层合并。

BN 层可由公式(1)表示：

$$y_i \leftarrow \gamma \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \equiv BN_{\gamma, \beta}(x_i) \quad (1)$$

式中  $i$  为  $[0, n]$  区间的正数， $\mu$  和  $\sigma^2$  分别为数据的均值与样本的方差， $\gamma$  和  $\beta$  训练过程中可学习的两个超参数， $\varepsilon$  为趋于 0 的常数。

卷积层的输出可由公式 (2) 表示，式中  $W$  为网络学习到的权重系数， $B$  为偏差。

$$Y = W \times X + B \quad (2)$$

将 BN 层与卷积层的参数融合，将卷积层的输出作为 BN 层的输入，可得到新的卷积层的表示，如公式 (3) 表示：

$$\begin{aligned} Y &= \gamma \times \left( \frac{W \times X + B - \mu}{\sqrt{\sigma^2 + \varepsilon}} \right) + \beta \\ &= \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} \times W \times X + \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} \times (B - \mu) + \beta \\ &= W \times \alpha \times X + (B - \mu) \times \alpha + \beta \\ &= W_{merged} \times X + B_{merged} \end{aligned} \quad (3)$$

其中  $\alpha$  为可学习的超参数,  $W_{merged}$  和  $B_{merged}$  分别为融合后的卷积层的权重系数和偏差, 如公式 (4) (5) (6) 所示。

$$\alpha = \frac{\gamma}{\sqrt{\sigma^2 + \varepsilon}} \quad (4)$$

$$W_{merged} = W \times \alpha \quad (5)$$

$$B_{merged} = (B - \mu) \times \alpha + \beta \quad (6)$$

在推理阶段的模型解析中, 合并后的层的计算量减小, 内存占用减小, 有利于推理速度的提升。简化的网络模型在准确率和效率上优于原始的 R-FCN 模型, 实验结果及讨论见实验部分 5.2.1。

## 4. 基于 TensorRT 优化加速的目标检测算法

为实现目标检测模型的实时推理, 将合并层后的 R-FCN 网络部署在嵌入式平台 Jetson TX2 上, 结合 NVIDIA 公司推出的 TensorRT 引擎, 本文提出了基于 TensorRT 的路标标识检测网络模型的优化加速方法。

### 4.1 TensorRT 流程

TensorRT 是 NVIDIA 公司推出的高性能优化推理引擎, 可部署在嵌入式平台, 并且不需要任何深度学习框架的支持, 即可实现对已有网络模型的推理加速。TensorRT 的工作过程主要分为 3 个部分, 即网络模型的解析, 引擎的优化、推理执行阶段, 如图 1 所示。

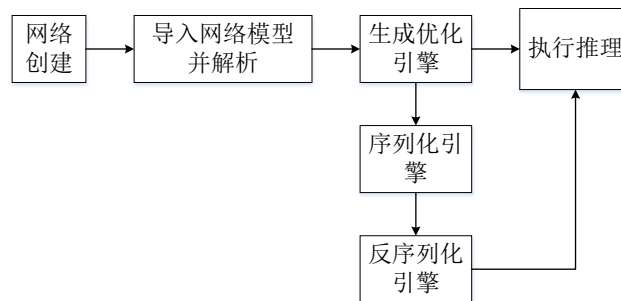


图 1 TensorRT 加速引擎构建流程

Fig. 1 The pipeline of TensorRT acceleration engineer

首先 TensorRT 使用 Parser 模型解析器对训练有素的深度学习模型进行解析, 解析层与层之间的关系, 消除未使用的输出层, 以减小计算, 实现加速。TensorRT 作为优化工具, 支持深度学习的一些通用层的直接解析, 如卷积层、池化层等, 但对于一些特定的层是不能解析的, 如本文采用的网络 R-FCN 中的位置敏感候选区域池化 (Position Sensitive ROI Pooling) 层和 Proposal 层。

其次, 引擎构建器会根据网络部署的平台以及参数配置选择最优的 Kernel, 这些 Kernel 针对所选目标平台、推理任务的复杂程度以及一系列参数进行了手动调整和优化。在优化阶段, TensorRT 将自动从 Kernel 库中选择最适合目标平台以及一系列参数的优化算法, 从而构建优化加速引擎。

最后, 在推理时创建上下文环境, 主要是预先分配资源, 并执行上一步获得的优化引擎从而进行推理。不过在此之前可将生成的优化引擎序列化存储在磁盘上备用, 当下次需要执行推理时, 再反序列化该引擎执行推理, 无需每次执行推理时重复创建该引擎。

## 4.2 基于 TensorRT 前向推理的优化加速方法

在前向推理阶段，本文对所部署的简化 R-FCN 采用 TensorRT 提出了进一步优化和重构策略：

一是进行网络层垂直整合。将网络层进行整合消除未使用的输出层以避免不必要的计算。这里将目前主流神经网络的 Bias、ReLU 和卷积层进行垂直结构整合，以提高推理的效率，如图 2 所示。

二是采用量化与半精度的压缩。推理与训练的过程不同，在推理的过程中，只需要进行前向传播，因此可以使用较低的精度，如 Float16 或者 Int8。研究表明，在 CNN 的推理过程中使用 Float16 或者 Int8 对精度影响较小。考虑到平台的配置以及本实验任务的精度要求，本文使用半精度 FP16 代替 32 位 float 数据精度，以获得高效的推理性能。

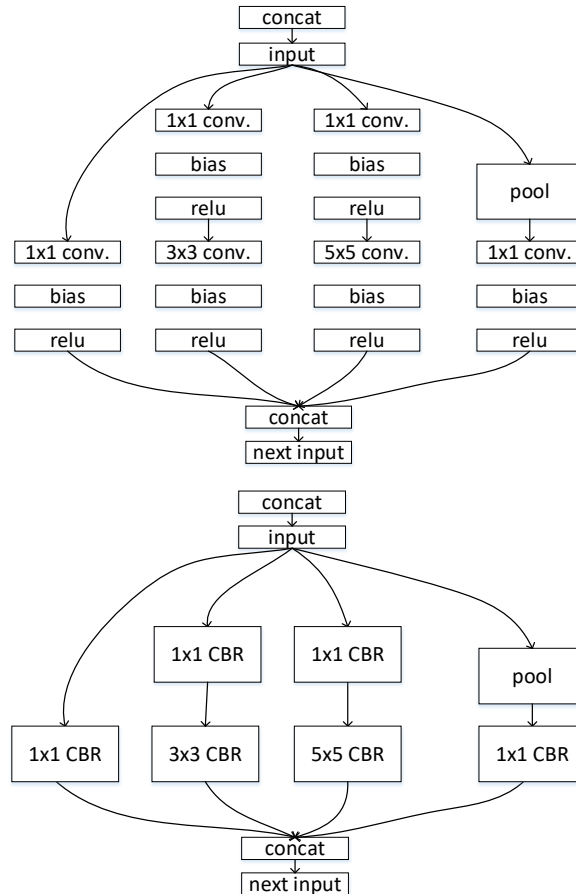


图 2 网络层垂直整合

Fig. 2 Network layer vertical integration

## 5 实验

### 5.1 数据集

Road Marking Dataset 数据集<sup>[23]</sup>是国外的路面标识数据集，但我国的路面标识和国外的存在一定的差异。文献[6]提出的 Tsinghua-Tencent100K 国内路面标识数据集和百度 apollo 公开数据集<sup>[24]</sup>路面标识的数量并不多、比较杂乱。因此，为了满足我国实际道路场景的条件，本文自建了路面标识数据集。自建路面标识数据库是从上述库中选出 1728 张图片（2048×2048）作为原始数据。路面标识主要分为左转箭头、右转箭头、直行加左转箭头、直行加右转箭头、直行箭头以及人行横道 6 类。

由于可得到路面标识数据量较少且存在路面标识类别不均衡的问题，直行加左转箭头和直行加右转箭头的数量要远远低于其他种类的路面标识，因此需要对自建数据集进行数据扩充和均衡等预处理，以提高深度学习模型的检测性能。本文采用了旋转、亮度与对比度变化、色度变化、多尺度

缩放等多种组合的数据扩充方法和过采样及欠采样相结合的数据均衡方法。均衡前后的数据构成如表 1 所示。从均衡后数据取出 1200 张的图像作为测试集，并将余下的数据以 7:3 的比例划分训练集以及验证集，并确保同一张原始数据扩充之后的数据与原始数据在同一个集合中。

表 1 自建路面标识数据集的原始数据、扩充 数据、均衡数据

Tab. 1 Raw data, extended data, and balanced data of self-built road marking dataset

标识	原始数量	扩充后的数量	均衡后的数量
左转箭头	226	9345	5115
右转箭头	549	9345	5115
直行+左转箭头	45	4936	4981
直行+右转箭头	65	4936	4981
直行箭头	345	9210	5270
人行横道	498	8964	5964

为了验证模型的泛化性，除了上述提到的取出均衡后的 1200 张图像作为测试集外，本文在测试数据中又加入了 1300 张不含以上 6 种路面标识的图像作为负样本。

## 5.2 实验结果及分析

### 5.2.1 简化后的 R-FCN 网络模型实验

为验证 3.2 部分提出的简化后网络模型的有效性，本文首先在 PC 机上进行了算法性能，使用的系统是 Ubuntu16.04，显卡是 NVIDIA 1080Ti。在自建的训练库和测试库上，比较 TT-CNN<sup>[6]</sup>、SSD<sup>[21]</sup>、FSSD<sup>[22]</sup>、原始的基于 ResNet-50 的 R-FCN<sup>[19]</sup>和本文简化后 R-FCN 网络模型的性能，称简化后的模型为 R-FCN\*，实验结果见表 2。表 2 分别给出了左转、右转等 6 种常见路面标识的 AP (Average Precision) 值及各方法最终 mAP (mean Average Precision) 值和计算速度，其中 AP 值表示各个类别的平均识别精度，mAP 值表示所有类别的平均识别精度。从表 2 可以看出，基于 two-stage 的 R-FCN 和本文简化后网络在准确率达到了 90% 以上，较 SSD 和 FSSD 算法 mAP 值有明显提高。同时本文简化后网络较原始的 R-FCN 方法在 mAP 性能基本相当的情况下，计算速度由 105ms/frame 减少至 56ms/frame。

表 2 自建路面标识数据集的不同算法性能比较

Tab. 2 Performance comparison on self-built road marking dataset

标识 \ 算法	TT-CNN <sup>[6]</sup>	SSD <sup>[21]</sup>	FSSD <sup>[22]</sup>	R-FCN <sup>[19]</sup>	图像预处理 +R-FCN <sup>[19]</sup>	图像预处理 + R-FCN*
左转箭头	0.627	0.619	0.668	0.911	0.905	0.903
右转箭头	0.649	0.623	0.657	0.905	0.905	0.900
直行+右转箭头	0.648	0.620	0.669	0.898	0.897	0.896
直行+左转箭头	0.660	0.631	0.659	0.899	0.899	0.896
直行箭头	0.820	0.859	0.887	0.906	0.903	0.901
人行横道	0.906	0.897	0.916	0.924	0.913	0.904
mAP	0.718	0.708	0.743	0.907	0.903	0.900
速度(ms/frame)	1200	33	41	105	90	56

此外，针对路面标识多出现在图片下方的特点，本文对输入网络的图像首先进行了预处理，对图像顶部 1/3 进行了裁剪。原始基于 ResNet50 的模型对预处理后的图像处理速度由 105ms/frame 缩短到 90ms/frame，简化后的网络模型的处理速度缩短至 56ms/frame，而平均 mAP 基本保持不变，说明简化后的算法模型在 PC 机环境下满足实时性的要求。

### 5.2.2 基于 TensorRT 的优化加速算法实验

本文将上述简化后的 R-FCN 模型部署到嵌入式平台 TX2 上，表 3 给出了在自建库上基于 TensorRT 优化加速前后模型性能的比较。若未采用优化加速算法，简化后的 R-FCN 模型在 TX2 平台上计算速度为 241ms/frame，这也说明了嵌入式平台计算能力远不及 PC 服务器的计算能力。本文提出的基于网络层垂直整合和半精度量化的优化网络模型，在 TX2 平台上计算速度由 241ms/frame 提高到 123ms/frame，加速效果显著。

表 3 自建数据集优化加速前后性能对比

Tab. 3 Performance comparison of acceleration on self-built road marking dataset

标识	加速优化前	加速优化后
速度(ms/frame)	241	123
左转箭头	0.900	0.890
右转箭头	0.903	0.894
直行+左转箭头	0.896	0.885
直行+右转箭头	0.896	0.881
直行箭头	0.901	0.896
人行横道	0.904	0.894
mAP	0.900	0.890

### 5.2.3 Road Marking 数据集的实验

为进一步验证本文提出算法的鲁棒性，我们亦在国外公开数据库 Road Marking Dataset 上，进行了多种路面标识的检测测试。虽然原始的 Road Marking 数据集每张图像存在一个或多个目标，但每张图像的标注信息只包含一个目标。因此，为了满足实际场景中同时测试多种路标识的应用需求，本文对该数据集 1443 张图像（800×600）重新标注，该路面标识主要分为“35”，“40”，“left turn”，“right turn”，“forward”，“stop”，“ped”，“bike”，“xing”，“rail” 10 类。由于数据量较少的原因，同样需要扩充数据，并针对数量偏少的类别进行均衡。表 4 给出了基于简化后网络模型在 TX2 平台上优化加速前后模型性能的最终比较。

表 4 Road Marking 数据集优化前后性能对比

Tab.4 Performance comparison on Road Marking dataset

标识	加速前	优化后
速度(ms/frame)	130	66
35	0.909	0.903
40	0.909	0.901
bike	0.909	0.900
forward	0.909	0.901
left turn	0.813	0.810
ped	0.909	0.901
rail	0.909	0.901
right turn	0.908	0.903
stop	1.000	0.997
xing	0.817	0.813
mAP	0.899	0.893

如表 4 所示，经过优化后，模型在推理时间性能上有近一半的提升，在检测准确率上仅有较小的损失，其中 Road Marking 数据集中的类别除“left turn”，“xing”外检测准确率均达到 90%左右。

“left turn”这类路标在一些测试图像中对应到磨损的小目标区域，故存在一定比例的漏检，“xing”路标在数据库中数量较少，一定程度上影响了其特征学习导致检测准确率相对其他类别较低。

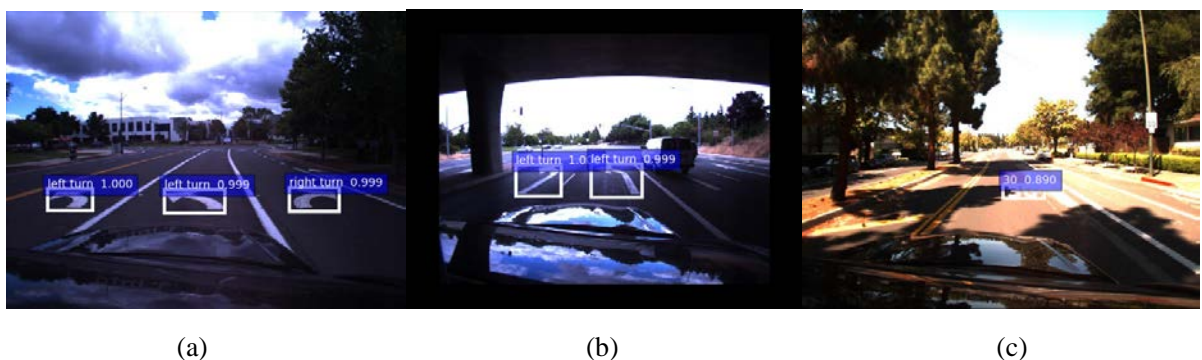
### 5.3 实验结果可视化与分析

图 3 和图 4 分别给出了自建库和 Road Marking 测试集上部分可视化结果。如图 3(a)和图 3(b)所示，在不同光照条件下实现了对路面标识的准确检测，图 3(c)和图 3(d)实现了在交通路口以及在路面标识有阴影遮挡情况下的准确检测。从图 4 中也可看出在不同场景、不同天气及光照下该算法均能实现较为准确的检测，图 4(e)和图 4(f)实现了在不同场景下的多类目标的准确检测。实验结果表明本文提出的路面标识算法在复杂环境下具有较好的鲁棒性，且在检测正确前提下，优化加速后的模型在嵌入式平台上推理速度有较大提升。



图 3 自建库测试集部分可视化结果

Fig. 3 Partial visualization results on test set of self-built dataset



(a)

(b)

(c)



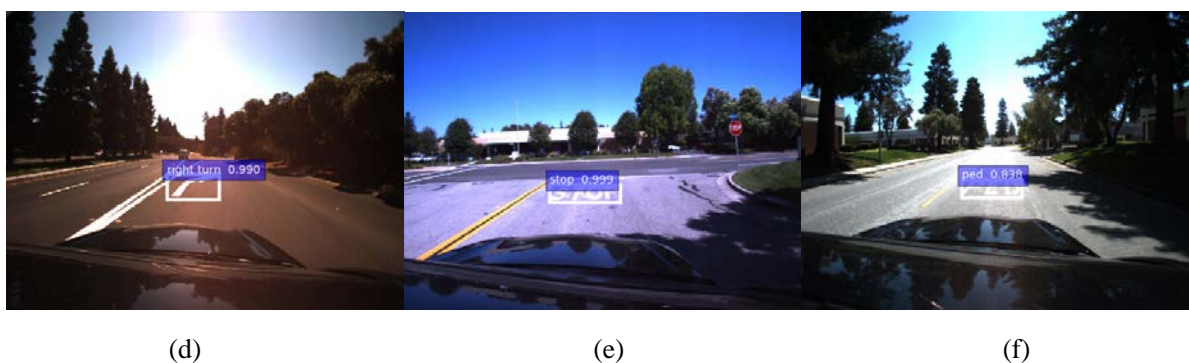


图 4 Road Marking 测试集部分可视化结果

Fig. 4 Partial visualization results on test set of Road Marking dataset

## 6 结论

为了满足无人驾驶中路面标识实时检测的要求，本文将路面标识检测模型部署在嵌入式平台 TX2 上，对网络模型进行优化加速。使用 TensorRT 优化加速工具，在推理过程中对网络层进行融合，并采用量化的方法实现了网络模型的前向加速，在满足准确率的前提下，实现了推理运行时上的较大提升，具有一定的研究意义和应用价值。

### 参考文献：

- [1] Liu W, Lv J, Yu B, et al. Multi-type road marking recognition using adaboost detection and extreme learning machine classification[C]// 2015 IEEE Intelligent Vehicles Symposium, 2015: 41-46
- [2] Chira IM, ChibulcuteanA, Danescu R G. Real-time detection of road markings for driving assistance applications[C]//International Conference on Computer Engineering and Systems. IEEE, 2010:158-163.
- [3] Tian J, Yuan J, Liu H. Road Marking Detection Based on Mask R-CNN Instance Segmentation Model[C]// 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL). 2020.
- [4] Chen T, Chen Z, Shi Q, et al. Road marking detection and classification using machine learning algorithms[C]// 2015 IEEE Intelligent Vehicles Symposium, 2015:617-621.
- [5] Eduardo Romera, Luis M. Bergasa, Roberto Arroyo. A Real-Time Multi-scale Vehicle Detection and Tracking Approach for Smartphones[C]//IEEE Intelligent Transportation Systems Conference (ITSC), 2015.
- [6] Zhu Z, Liang D, etc. Traffic-Sign Detection and Classification in the Wild[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016:2110-2118.
- [7] Sermanet Z, Eigen D, etc. OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks[C]// International Conference on Learning Representations, 2014.
- [8] Ren S, He K, Girshick R, and Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6):1137-1149.
- [9] Badrinarayanan Vijay, Kendall Alex and Cipolla Roberto. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation[J]. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2017, 39(12):2481-2495.

- 
- [10] He K, Gkioxari G, Dollár P, and Girshick R. Mask R-CNN[C]// IEEE International Conference on Computer Vision, 2017: 2980–2988.
- [11] Youngeun K, Seokeon C, Hankyeol L, *et al.* RPM-Net: Robust Pixel-Level Matching Networks for Self-Supervised Video Object Segmentation[C]//IEEE Winter Conference on Applications of Computer Vision, 2020: 2046-2054.
- [12] Ullrich K, Meeds E, and Welling M. Soft weight-sharing for neural network compression[C]// 5th International Conference on Learning Representations, 2017.
- [13] Tai C, Xiao T, Wang X, and E W. Convolutional neural networks with low-rank regularization[C]// 4th International Conference on Learning Representations, 2016.
- [14] Wu B, Iandola F N, Jin P H, and Keutzer K. Squeezenet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving[C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2016:446-454.
- [15] Geoffrey Hinton, Oriol Vinyals, Jeffrey Dean. Distilling the Knowledge in a Neural Network [C]// Neural Information Processing Systems 2014 Deep Learning Workshop, 2014.
- [16] NVIDIA. JETSON TX2 [EB/OL].  
<https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-tx2>
- [17] Robert J. Wang, Xiang Li, Charles X. Ling. Pelee: A Real-Time Object Detection System on Mobile Devices[C]// 32nd Conference on Neural Information Processing Systems, 2018: 1967-1976.
- [18] Nuha H. A., Hadeel N. A. Enhancement Performance of Multiple Objects Detection and Tracking for Real-time and Online Applications [J]. International Journal of Intelligent Engineering and Systems, 2020(13):533-545.
- [19] Dai J, Li Y, He K, and Sun J. R-FCN: Object detection via region-based fully convolutional networks[C]// Neural Information Processing Systems, 2016:379-387.
- [20] He K, Zhang X, Ren S, and Sun J. Deep residual learning for image recognition[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016:770-778.
- [21] Liu W, Anguelov D, Erhan D, Szegedy C, and Reed S. SSD: Single-shot multibox detector[C]// European Conference on Computer Vision, 2016:21-37.
- [22] Li Z, Zhou F. FSSD: Feature Fusion Single Shot Multibox Detector[C]//The Computing Research Repository, 2017.
- [23] Wu T, Ranganathan A. A practical system for road marking detection and recognition[C]// 2012 IEEE Intelligent Vehicles Symposium, 2012: 25–30.
- [24] Huang X, Cheng X, Geng Q, Cao B, Zhou D, Wang P, Lin Y, Yang R. The ApolloScape Dataset for Autonomous Driving[C]// 2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018:954-960.