

基于改进 SOLO 的列车主动避障视觉算法研究

姚巍巍 田野 李晨

(数据与智能技术中心株洲中车时代电气股份有限公司 株洲 412001)

摘要: 随着我国经济的高速全面发展,人民生活水平的日益提高,我国对于交通运输方面的需求越来越大。为了满足列车安全运行的需求,本文提出铁路场景下基于单阶段实例分割的列车主动避障视觉算法,针对铁路场景中侵线情况下检测物体多重叠的特点对算法模型进行了优化,改进主干网络和多尺度融合方法提高了模型的精度,利用 TensorRT 半精度加速和 CUDA 重构对模型进行了加速,并对本文方法和其他方法进行性能评价与对比试验。最终,本文方法在嵌入式平台 Xavier 上实现了 71.2MAP 和 108 ms 的速度,实现了车载部署下列车前方环境的高效高精度检测。

关键词: 深度学习;实例分割;标签分配;模型加速

中图分类号: TP2 **文献标识码:** A **国家标准学科分类代码:** 510.1050

Research on vision algorithm of the train's advanced driver assistance system base on improved SOLO

Yao Weiwei Tian Ye Li Chen

(Data and Intelligence R&D Center, Zhuzhou CRRC Times Electric Co., Ltd., Zhuzhou 412001, China)

Abstract: With the rapid and comprehensive development of China's economy and the improvement of people's living standards, China's demand for transportation is increasing. In order to meet the higher requirements of locomotive operation safety, a visual algorithm of the advanced driver assistance system of trains based on single-stage instance segmentation in railway scene is proposed. The algorithm model is optimized for detecting the characteristics of multiple overlaps of objects in railway scenes. The accuracy of the model is improved by improving the Backbone network and multi-scale fusion method. The model is accelerated by TensorRT semi-precision acceleration and CUDA code refactoring. The performance evaluation and comparative test of this method and other methods are carried out. Finally, this method achieves 71.2MAP and 108 ms on the embedded platform Xavier. High-precision detection of the surrounding environment of the train under vehicle deployment is realized.

Keywords: deep learning; instance segmentation; label assignment; model acceleration

0 引言

随着我国经济的高速全面发展,人民生活水平的日益提高,我国对于交通运输方面的需求越来越大。轨道交通由于其运费低、速度快和运量大的特性,在我国交通运输中的地位愈为重要。自 2011 年“四横四纵”高速铁路干线陆续建成通车和 2016 年“八纵八横”的高速铁路网络开始建设以来,我国铁路运输的速度和运输总量实现了质的突破,促进了国家经济的发展和相关产业的升级。随着铁路运输生产力的高速发展,铁路行车的安全问题也越来越受到重视。近年来,轨道交通领域经常出现行人、障碍物侵线的事实,对列车的安全行驶造成严重威胁。列车辅助驾驶系统

的研制对保障列车安全行驶、实现列车无人驾驶有着重要的作用,搭建一种车载的列车辅助驾驶系统,实现非接触式主动障碍物检测,增强列车对前方障碍物及异常情况的检测能力便成为一项极有意义的工作。

利用视觉图像和雷达实现对列车运行环境的自动感知是实现列车辅助驾驶系统的重要手段。随着计算机科学在视觉处理领域的快速发展,利用机器学习方法处理图像和雷达点云信息已被广泛的应用于铁轨检测,为利用视觉感知列车前方运行环境提供了基础。基于传统机器学习方法,已有很多方法实现了铁轨区域的分割和检测。文献[1]通过边缘检测算子实现了铁轨识别。文献[2]基于曲率映射图实现近距离轨道识别,基于局部梯度信息实现了远距

离轨道识别。传统机器学习方法无法应对铁路场景复杂的光照环境,也无法满足障碍物检测的需求。因此,基于深度学习的方法被引入到铁路场景中。文献[3]基于深度学习的语义分割算法实现了较高精度的当前轨道识别。文献[4]将障碍物检测转化为物体检测,并将 Faster RCNN 引入铁路障碍物检测中。文献[5]提出了特征融合细化神经网络(FR-Net)改善了铁路调车环境中中小目标的检测。文献[6]采用实例分割算法 Mask RCNN^[7]完成对铁轨区域的识别与分割,通过联合标定视觉传感器与激光雷达传感器,完成多传感器的数据融合,实现对列车行驶轨道区域进行障碍物检测。当前,大多数基于深度学习的算法存在着模型复杂耗时较长的问题,同时模型的精度仍有较大的提升空间。

为了满足列车主动避障需求,实现铁轨区域分割和障碍物检测的同步进行,本文采用深度学习中效果优异的单阶段实例分割算法 SOLO^[11],针对铁路场景进行了优化,通过改进标签匹配的方法避免了铁轨区域和障碍物体的相互干扰,改进主干网络和多尺度融合方法提高了模型的精度,利用 TensorRT 半精度加速和计算统一设备架构(compute unified device architecture, CUDA)重构对模型推理进行了加速。最终将其成功的部署在嵌入式边缘计算设备 Xavier 上,实现了车载部署下列车前方环境的高精度检测。

1 模型介绍及优化

1.1 SOLO 模型

列车周围环境检测最为关键的任务是判断列车前方是否存在人员或障碍物侵线,传统的目标检测任务无法满足对铁道轮廓的检测,无法准确得到当前轨道的边界,从而证实是否有人或者障碍物侵线,因此我们采用实例分割算法对当前轨道附近出现的目标进行检测识别。图 1 展示了目标检测、实例分割之间的区别,目标检测仅能得到物体的检测框,而实例分割可以得到每个物体的具体轮廓。

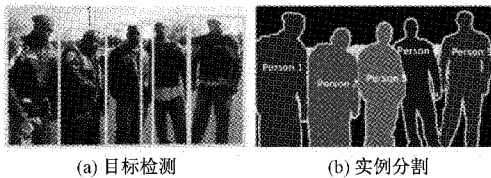


图 1 目标检测、实例分割的区别

当前,主流的实例分割算法主要可以分为两类:二阶段和单阶段。随着深度学习在计算机视觉领域的应用,实例分割的精度得到了极大的提高。当前,二阶段的实例分割算法先通过目标检测的方法找出图片中所有的目标并画框标注,再对所有检测框内的像素进行语义分割得到目标的边界轮廓,每个分割结果都作为一个不同的实例输出,如 Mask RCNN、Cascade RCNN^[8]等。二阶段的实例分割算法精度高,速度慢,但如果目标框预测的不准,则分割结果

也会不准。因此对于一些边缘精度要求高的任务而言,并不是一个较好的方案。单阶段实例分割算法免去了先找目标框这一步骤,直接得到目标的分割结果,如 Yolact^[9]、CenterMask^[10]、SOLO 等。其中,2020 年被提出的 SOLO 算法精度最高,推理速度快,达到了最先进的(state of the art, SOTA)的水平。因此,本文依据 SOLO 算法,根据列车感知周围环境的需求进行了诸多改进。

SOLO 模型如图 2 所示,主要由主干网络(Backbone)、多尺度金字塔融合结构(feature pyramid networks, FPN)、掩码分支(Mask feat branch)、卷积核分支(Kernel branch)和分类分支(Category branch)组成。输入的图片分别经过 Mask feat branch、Category branch 和 Kernel branch 后得到 $H/4 \times W/4 \times E$ 的 Mask feature(掩码特征图),不同 level 的 $S \times S \times C$ 的分类特征图(Cate features)和 $S \times S \times E$ 的卷积核特征图(Kernel features),其中 H, W 为原图的长宽, C 为要识别的类别数量, S 为不同 level 的特征图长宽, E 为 Kernel features 和 Mask feature 的维度。

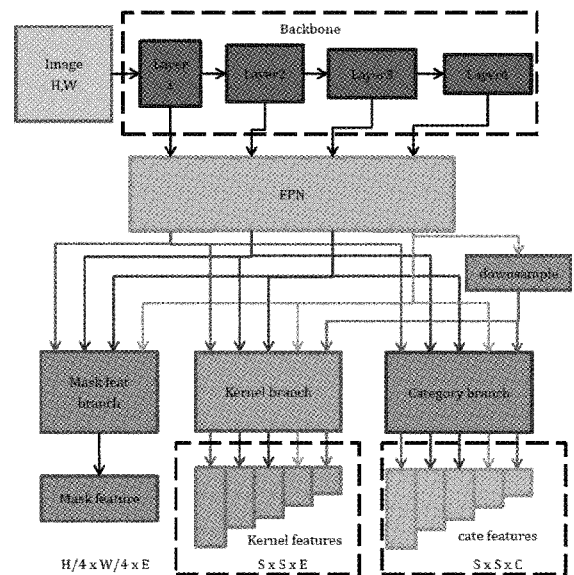


图 2 SOLO 模型框架示意图

SOLO 不再生成和利用目标框,而是将分割问题转化为位置分类问题。在具体推理时,如果图片中物体的质心落在 Cate features 中的网格 (i, j) 时,那网格 (i, j, c) 就负责检测到物体的类别, Kernel features 中对应位置的特征向量作为语义分割的卷积核,与整个 Mask feature 进行 1×1 的卷积,从而得到物体的实例,如图 3 所示。多个 level 的 Cate features 和 Kernel features 全方面覆盖了不同大小的物体,而尺寸较大的 Mask feature 相对保存了更多的分割边缘的细节。

1.2 标签匹配优化

标签分配(Label assignment)指在训练过程中,为图片中出现物体的相应位置打上标签作为正样本、负样本或者忽略,从而实现模型的监督训练。标签分配在很大程度上

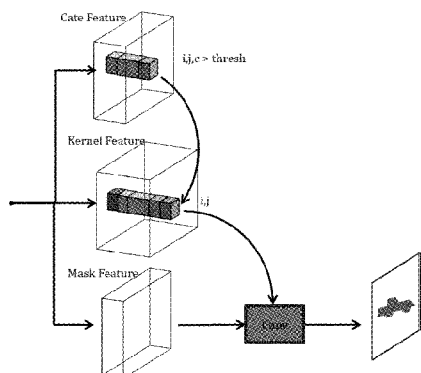


图 3 SOLO 模型推理示意图

影响着模型精度的上限。SOLO 作为单阶段实例分割算法,在训练中将图片分为 $S \times S$ 块网格,将物体的质心所在网格作为正样本进行标记,其余位置作为负样本。这种分配标签从概念上讲是最为直观的,为以往的各种单阶段网络提供了出色的检测性能。但是,这种传统的标记方法有着一个严重的问题。当图片中两个物体大面积重合,或者一个物体被另一个物体包围时,这两个物体的质心则会落在同一个网格中。尽管 SOLO 采用了五个 level 的特征图,使不同尺寸的物体实例尽量落在不同的特征图中以避免上述情况的发生。但对于大小尺寸相近的物体,仍有很大可能落在同一张特征图的同一网格中,如图 4 所示。

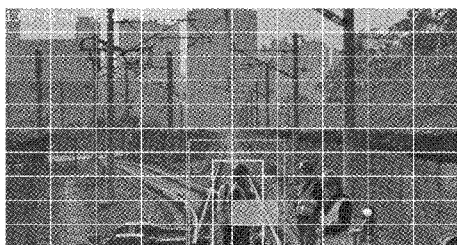


图 4 传统标签匹配方法使铁轨和人的质心落在同一个网格中

当出现这种情况,这个网格就负责同时检测两种不同类别的物体实例。在训练过程中,同一个网格同时对两种不同外观的物体进行损失梯度下降,造成两个物体的轮廓混淆,严重干扰识别效果,如图 5 所示。

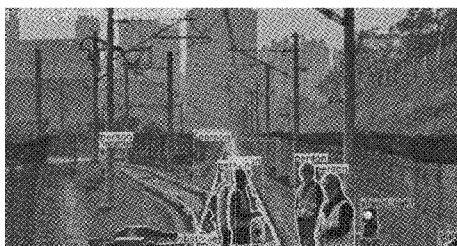


图 5 不正确的标签分配造成的物体轮廓混淆

本文参考 Sun 等^[12]在单阶段目标检测算法 OneNet 中使用的最小损失匹配 (minimum cost assignment) 方法,提

出用于单阶段实例分割算法的最小损失匹配方法。最小损失标签匹配方法不再将物体的质心所在网格作为正样本,而是将正样本分配视为一个自适应匹配的问题,通过计算真实目标框内每个网格与真实目标的定位以及分类损失,将损失最小值所在的网格视为正样本。由于 SOLO 并不对目标框进行回归,因此本文针对 SOLO 等单阶段实例分割算法直接得到实例分割的特点,通过将样本匹配损失改为分类的损失和分割结果的损失之和,综合每个网格与真实目标的分类和语义分割损失。具体如下式所示:

$$\text{cost}_{i,j} = \text{sigmoid}(\text{focalloss}(P_{i,j}, P_{gt})) + \text{sigmoid}(\text{diceLoss}(m_{i,j}, m_{gt})) \quad (1)$$

$$A = \text{argmin}\{\text{cost}_{i,j}\} \quad (2)$$

其中, $P_{i,j}$ 为 Cate features 中 i, j 处网格的分类预测值, P_{gt} 为样本真实值, focalloss 为分类的损失函数, $m_{i,j}$ 为 Kernel features 中 i, j 处网格的实例分割预测值, m_{gt} 为样本真实轮廓, diceLoss 为语义分割损失函数。 i, j 为真实目标框映射到 $S \times S$ 块网格中的范围, sigmoid 函数将分类损失和语义分割损失缩减到 $(0 \sim 1)$ 范围内,达到综合分类损失和语义分割损失的目的, A 为真实目标框内中损失最小的网格,即应该分配正样本的网格。最小损失标签匹配方法同时考虑了综合分类损失和语义分割损失,其匹配的网格中的特征就是最能代表物体的特征,因此,采用本方法对网络进行训练也将会提高精度。在训练中,本文将真实目标按面积从小到大排列,依次计算得到其正样本所在网格并进行记录,从下次所要计算的网格范围中排除已经标记过的网格,以避免重叠物体的正样本标记在同一网格,如图 6 所示。

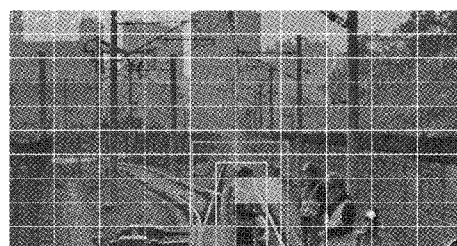


图 6 最小损失标签匹配,使负责检测物体的网格落在损失最小的位置

1.3 多尺度融合优化

由于需要识别的实例的大小不一,SOLO 模型通过 FPN 结构进行多尺度融合,以提高实例分割的检测效果。FPN 结构^[20]如图 1~7 所示,通过上采样操作依次将高维度小尺寸的特征图与低维度大尺寸特征图进行融合,使输出的特征图不仅包含了较强的类别信息特征,又得到了高分辨率位置信息^[18]。

然而,FPN 只能使高层信息流向低层,却无法将低层信息带向高层,是一种单向的融合方式。白文献提出 PANNet^[14]以来,双向融合的有效性被得到了证明。2020

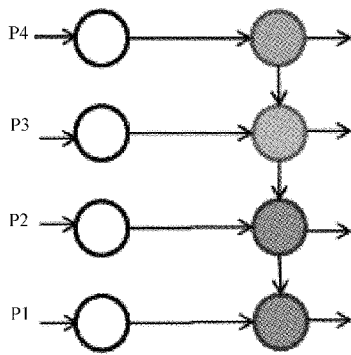


图 7 FPN 结构示意图

年,文献[15]提出加权双向特征金字塔网络 (weighted bidirectional feature pyramid network, BIFPN), BIFPN 在 PANNet 的基础上增加了同层之间的计算结构,若输入和输出结点是同一尺度的,则添加一条额外的边,在不增加计算量的同时融合更多的特征,并引入了权重结构,用来更好地平衡不同尺度的特征信息。因此,本文引入 BIFPN 替代 FPN 以提高模型的精度。BIFPN 中的权重结构是一种基于训练集的先验性统计,即权重的大小来自训练集的训练,在测试中权重完全是固定的,而且一个权重对应于整张特征图,无法针对特征图的局部进行权重的细分。因此,我们针对 BIFPN 的权重结构进行了改进,将先验性的权重训练改为了两张特征图的层级融合,通过 1×1 的卷积操作进行特征图局部和局部之间的自适应权重分配,去除了较为费时的 softmax 而使用了 relu 激活函数,在采用卷积计算的同时也不会大幅的增加模型的推理耗时,如图 8 所示。

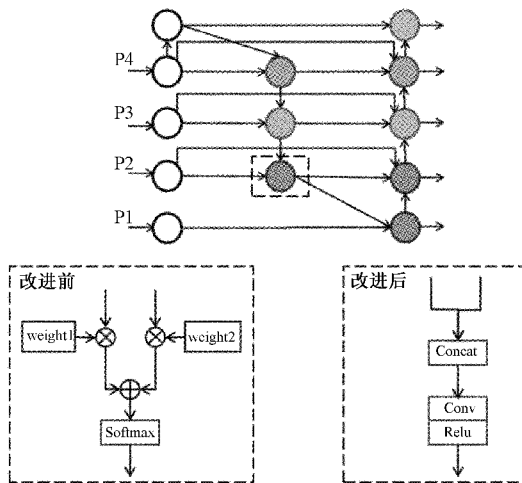


图 8 BIFPN 结构示意图

1.4 主干网络优化

为进一步提高 SOLO 模型的精度,同时减少模型的计算量,本文采用跨阶段融合网络 (cross stage partial network, CSPNet) 替换原有的 ResNet 作为 SOLO 模型的 Backbone。CSPNet 由文献[16]于 2019 年提出,相较于 ResNet^[17]更适用于目标检测、实例分割领域,有着计算量

较低、精度高的特点。CSPNet 的主要改进点包括以下部分:相较于 ResNet 中的层层连接的残差结构,CSPNet 将基础层的特征图分割成两部分,通过提出的跨阶段分层结构进行合并,在相同的网络层数下实现计算量的缩减和更丰富的梯度组合,保持模型准确率,如图 9 所示。

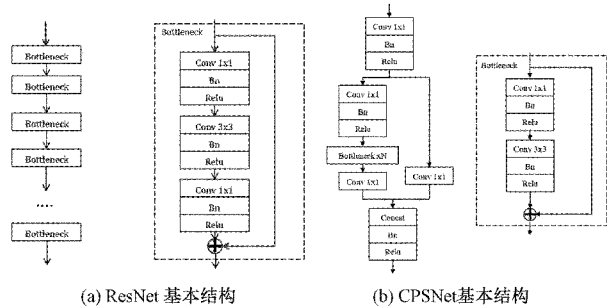


图 9 ResNet 和 CSPNet 的结构区别示意图

其次,本文参照文献^[18]采用 Mish 取代了 ReLU 作为激活函数。ReLU 和 Mish 函数如下:

$$\text{relu}(x) = \max\{0, x\} \tag{3}$$

$$\text{mish}(x) = x * \tanh(\log(1 + \exp(x))) \tag{4}$$

相较于 ReLU, Mish 激活函数在 0 点附近具有平滑和非单调的特性,允许高维的信息深入神经网络,能加速模型的收敛,从而得到更好的准确性和泛化能力^[19]。

2 模型部署加速

尽管相比其他实例分割算法,SOLO 模型的推理速度是较快的,然而也无法满足在 Xavier 这样计算能力受限的设备上运行的需求。因此,本文通过使用 CUDA 编程和 TensorRT 框架对模型进行重构,利用半精度的计算方式对模型主要推理部分进行提速,利用 CUDA/C++ 编程对模型的前后处理部分进行加速。

2.1 TensorRT 加速

TensorRT 是英伟达 (NVIDIA) 公司为满足深度学习的高速推理需求所推出的神经网络框架,致力于在 GPU 设备上实现神经网络的高速推理。TensorRT 主要通过以下方式对神经网络进行加速:1) 解析模型结构,消除模型中不使用或不进行输出的图层;2) 整合模型中多个图层为一层;3) 聚合具有足够相似的参数和张量的操作;4) 利用整数型或者 16 位半精度浮点型降低模型运行时计算所需的字节位数。采用整数型或者半精度都可以大幅度提高模型的运行速度,但当生成整数型模型时,需要为模型的每层准备校准表来确定模型每层输出的动态范围,从而确定适当缩放因子。相关实验表明,半精度模型相较原模型精度基本无变化,且加速效果明显。而整数型模型往往仅比半精度模型缩减很少的推理耗时,但却很容易带来较大的精度损失。因此,本文采用半精度实现对模型的加速。本文的模型采用 Pytorch 搭建,无法直接转换为 TensorRT 可读的参数格式,因此,本文先将 Pytorch 模型转换为 TensorRT 支持

的 onnx 格式模型,再转换为 TensorRT 模型,如图 10 所示。

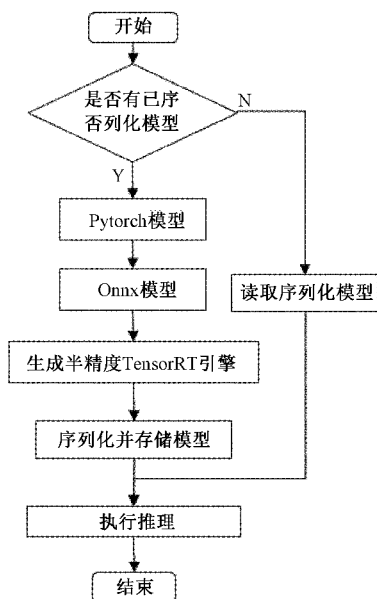


图 10 TensorRT 模型加速步骤框图

2.2 前后处理加速

图像在输入神经网络前,需要经过图片读取、调整尺寸和归一化处理。这些操作可被统称为前处理。而在经过 SOLO 模型处理后,将得到三组输出特征图:Mask feature、Cate features 和 Kernel features。如 1.1 节所述,还要经过从 Cate features 取出置信的网格,Mask feature 与 Kernel features 进行卷积得到物体的实例分割、nms 去除重复物体和在原图中画出分割轮廓等操作。这些操作可以被统称为模型的后处理。由于嵌入式设备 Xavier 的 CPU 频率较低,前后处理的耗时也被严重放大。因此,本文采取 CUDA 和 C++ 混合编程对原 python 代码进行重构,将大部分操作放在处理速度较快的 Xavier GPU 上,通过 GPU 并行计算能力实现提速的目的。

3 实验及结果分析

为了验证本文所进行的优化的效果,本文搭建 SOLO 网络模型进行了监督训练,并针对各项优化的效果进行评估。

3.1 数据收集与参数设置

本文采用的实例分割算法是一种有监督的深度学习方,因此需要进行数据的采集并制作训练集和测试集。采集的数据样本涵盖了试车线、正线和站场等不同场景的数据。为了获取最重要的侵线数据,在正线隧道内、试车线、站场进行了人员和障碍物侵线模拟。

在对采集的数据进行采样、去冲、清洗后,标注行人、火车、汽车、信号机、标志牌、工程车、站台、隧道入口、自行车/电动车、土挡、当前轨道、障碍物、隧道安全门和隧道内广告

牌 14 个类别的实例对象,共标注 22 137 张图片,数据按 10:1 的比例随机分为训练集和测试集,训练集 20 064 张图片,测试集 2 073 张图片。

本文在 3 块 V100 上针对第 1 节提出的各种优化策略进行了训练,各项超参数均设为一致:模型卷积层参数使用 kaiming 初始化方法,BN 层参数使用 constant 初始化,batch size 设置为 8,epoch 为 200,采用 SGD 梯度下降方法,初始学习率为 0.01,并采用 cosine 学习率下降策略。本文采用面积交并比(intersection over union, IOU)阈值在 0.5:0.95 的掩码平均精度均值(mean average precision, MAP)作为模型精度水平的衡量标准,测试集图片统一缩放为(1 152,640)。

3.2 实验结果分析

表 1 统计了训练后不同优化策略在测试集上的精度效果。

表 1 训练测试结果

改进标签 分配方法	改进多尺度 融合方法	改进主 干网络	MAP@.5:.95
			0.698
✓			0.701
✓	✓		0.706
✓	✓	✓	0.713

对 SOLO 算法进行标签分配方法改进、多尺度融合方法和主干网络改进均在 MAP 上有一定程度的提升。证实了本文所采用优化方法的可行性。质心匹配与最小损失样本匹配方法的对比如图 11 所示,图 11(a)中图片中央穿绿色反光安全服的人员被同时识别为“railway”和“person”,而图 11(b)中则没有出现轮廓混淆的问题,图 11(c)中的中间人员边界混淆不清,图 11(d)中三个人的轮廓完整清晰。由此,证实了本文所提出用于单阶段实例分割的最小损失匹配方法可以有效防止重叠物体的轮廓混淆。

随后,本文在 Xavier 中进行了模型加速的实验,多次测量各个部分的耗时情况并取均值,如表 2 和 3 所示。在模型转换为 TensorRT FP16 半精度后,对精度没有显著的影响,而推理耗时约为 pytorch 模型的 1/3。经过 CUDA/C++ 处理的前处理耗时约为原版 python 的 1/2,后处理耗时同样约为原版 python 的 1/2。模型总体耗时为 108 ms,基本实现了实时处理。

最后,本文将改进后的 SOLO 模型与同样经过 TensorRT 半精度加速后的 Mask RCNN 模型进行了对比,如表 4 所示。本文所改进的 SOLO 模型在速度和 MAP 上

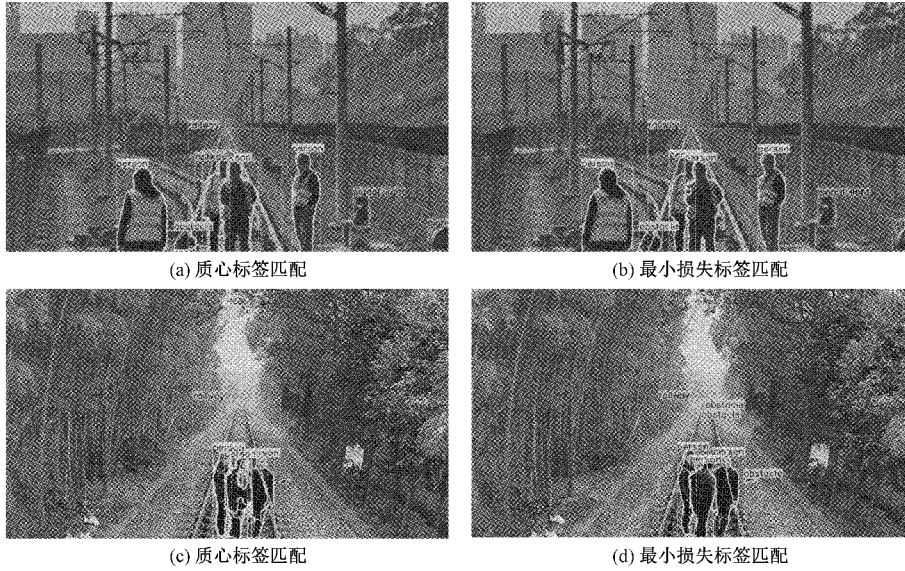


图 11 不同标签分配方法效果

表 2 TensorRT 模型加速效果

精度	MAP@.5:.95	推理耗时/ms
Fp32	0.713	275
Fp16	0.712	87

表 3 前后处理加速效果 ms

前后处理方式	前处理耗时	后处理耗时
Python	23	27
CUDA/C++	9	12

表 4 改进后 SOLO 与 Mask RCNN 模型对比

方法	MAP	MAP			耗时
		small	medium	large	
改进后 SOLO	0.712	0.365	0.670	0.804	108 ms
Mask RCNN	0.68	0.269	0.562	0.760	127 ms

均超越了 Mask RCNN 模型,在小目标和中目标物体的提升最为显著。图 12 展现了 SOLO 和 Mask RCNN 的精度-召回曲线。在轮廓细节上,SOLO 表现出了更好的边缘分割能力,如图 13 所示。

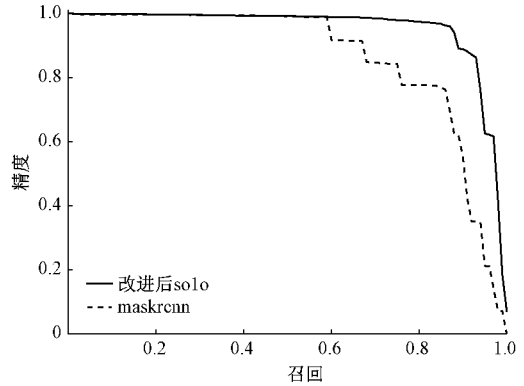


图 12 Mask RCNN 与改进后 SOLO 的精度-召回曲线

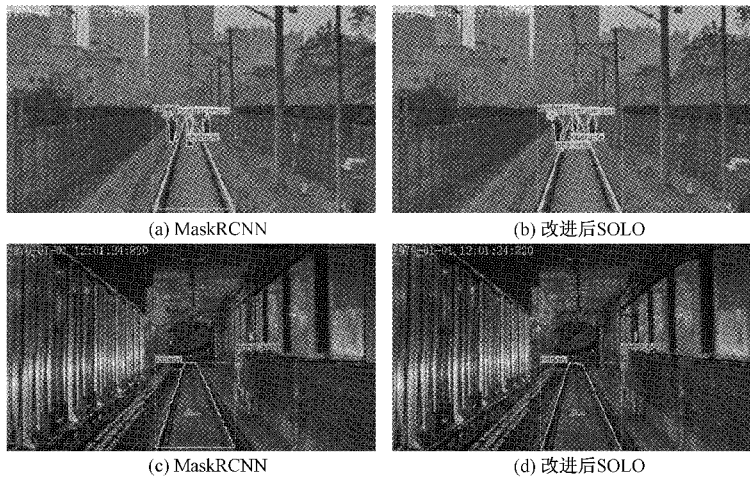


图 13 Mask RCNN 与改进后 SOLO 的细节对比

4 结 论

在这项工作中,本文对单阶段实例分割算法 SOLO 进行了优化和提速,改进标签匹配的方法避免了重叠物体的实例分割结果的相互干扰,改进主干网络和多尺度融合方法提高了模型的精度,利用 TensorRT 和 CUDA 重构加速了 SOLO 的运行速度。最终,本文提出的模型在边缘计算设备 Xavier 实现了 0.712 MAP 的精度和 108 ms 的处理速度,超越了 Mask RCNN 算法。

目前,本文所研究的算法基于单帧图像,对需要考量前后时空的目标判别仍存在问题。例如当列车经过道岔后,算法无法推理出列车所行驶的轨道。当前,列车 ADAS 的视觉与雷达信息的融合较为简单,是雷达和视觉各自算法取得结果后的结果融合。因此,后续将进一步展开综合前后帧的视觉算法和雷达与视觉的前融合算法研究。

参考文献

- [1] 柴世红. 基于边缘检测的轨枕识别[J]. 铁路计算机应用, 2009, 18(4):1-3.
- [2] 王云泽. 列车前方轨道识别算法的设计和实现[D]. 杭州:浙江大学, 2017.
- [3] WANG Y, WANG L D, HU H Y, et al. RailNet: A segmentation network for railroad detection[J]. IEEE Access, 2019, PP(99): 143772-143779.
- [4] YU M, YANG P, WEI S. Railway obstacle detection algorithm using neural network[J]. AIP Conference Proceedings, 2018, 1967:0400171-0400176.
- [5] YE T, WANG B, SONG P, et al. Railway traffic object detection using differential feature fusion convolution neural network[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 22: 1375-1387.
- [6] 管晓勇. 基于深度学习的列车障碍物检测与识别技术研究[D]. 北京:北京交通大学, 2020.
- [7] KAIMING H, GEORGIA G, PIOTR D, et al. Mask R-CNN[C]. 2017 IEEE International Conference on Computer Vision(ICCV), 2017.
- [8] ZHAO W CAI, NUNO V. Cascade R-CNN: Delving into high quality object detection [C]. 2018 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2018.
- [9] DANIEL B, CHONG Z, FANYI X, et al. YOLACT: Real-time instance segmentation [C]. 2019 IEEE International Conference on Computer Vision(ICCV), 2019.
- [10] YOUNGWAN L, JONGYOUL P. CenterMask: Real-time anchor-free instance segmentation[C]. 2020 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2020.
- [11] WANG X, ZHANG R, KONG T, et al. SOLOv2: Dynamic and fast instance segmentation[C]. NeurIPS, 2020.
- [12] SUN P, JIANG Y, XIE E, et al. OneNet: Towards end-to-end one-stage object detection [J]. ArXiv Preprint, 2020, ArXiv:2012.05780.
- [13] TSUNG Y L, PIOTR D, ROSS G, et al. Feature pyramid networks for object detection[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2017.
- [14] LIU SH, QI L, QIN H F, et al. Path aggregation network for instance segmentation [J]. ArXiv Preprint, 2018, ArXiv: 1803.01534.
- [15] TAN M, PANG R, LE Q V. EfficientDet: Scalable and efficient object detection [C]. 2020 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2020.
- [16] WANG C Y, LIAO H, WU Y H, et al. CSPNet: A new backbone that can enhance learning capability of CNN[J]. ArXiv Preprint, 2019, ArXiv:1911.11929.
- [17] HE K M. Deep residual learning for image recognition [C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2016.
- [18] WANG C Y, BOCHKOVSKIY A LIAO H. Scaled-YOLOv4: Scaling cross stage partial network [J]. ArXiv Preprint, 2020, ArXiv:2011.08036.
- [19] DIGANTA M. Mish: A self regularized non-monotonic neural activation function [J]. ArXiv Preprint, 2019, ArXiv:1908.08681.
- [20] 崔宵洋, 林建辉, 陈春俊, 等. FPN 在高速列车接触网定位器检测应用[J]. 电子测量技术, 2019, 6(15): 144-149.

作者简介

姚巍巍, 硕士, 助理工程师, 主要研究方向为轨道交通图像识别分析及相关场景落地。

E-mail: yaoww1@csrzc.com