计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2020,29(4):41−47 [doi: 10.15888/j.cnki.csa.007348]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于模型剪枝和半精度加速改进 YOLOv3-tiny 算法的实时司机违章行为检测①

姚巍巍, 张　洁

(西南交通大学 机械工程学院, 成都 610031)
通讯作者: 张　洁, E-mail: lsy55zj@home.swjtu.edu.cn

摘　要: 为解决在嵌入式设备上实时、高精度检测司机安全驾驶监督的问题, 本文基于目标检测中经典的深度学习神经网络 YOLOv3-tiny, 运用通道剪枝技术成功在目标检测任务中实现了模型压缩, 在精度不变的情况下减少了改进后神经网络的计算总量和参数总数. 并基于 NVIDIA 的推理框架 TensorRT 进行了模型层级融合和半精度加速, 部署加速后的模型. 实验结果表明, 加速模型的推理速度约为原模型的 2 倍, 参数体积缩小一半, 精度无损失, 实现了高精度下实时检测的目的.

关键词: 深度学习; 目标检测; 司机行为识别; 模型剪枝; 半精度加速

引用格式: 姚巍巍,张洁.基于模型剪枝和半精度加速改进 YOLOv3-tiny 算法的实时司机违章行为检测.计算机系统应用,2020,29(4):41–47. http://www.c-s-a.org.cn/1003-3254/7348.html

Real-Time Drivers' Violation Behaviors Detection Based on Improved YOLOv3-tiny Algorithm Based on Model Pruning and Half-Precision Acceleration

YAO Wei-Wei, ZHANG Jie

(School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China)

Abstract: In order to optimize the method of real-time and high-precision detection of drivers' safe driving supervision, based on the classic deep learning neural network-YOLOv3-tiny-in object detection, this study successfully uses the channel pruning technology to achieve model compression in the object detection task, and reduces the calculated total amount and parameters of the improved neural network under the condition of constant accuracy. Based on NVIDIA's inference platform TensorRT, model level fusion and half-precision acceleration are performed, and the accelerated model is deployed. The experimental results show that the speed of inference of the acceleration model is about 2 times that of the original model, the parameter volume is reduced by half, and the accuracy is not lost, which realizes the purpose of real-time detection under high precision.

Key words: deep learning; object detection; drivers' behaviors recognition; model pruning; half-precision acceleration

随着经济高速全面发展, 我国对交通运输的需求越来越大, 轨道交通由于其便捷和运量大的特性, 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以来, 我国铁路运输的速度和运输总量实现了质的突破. 铁路运输生产力的高速发展, 对铁路行车的安全保障提出了更高的要求. 确保机车的平稳运行已成为铁路运输部门的重中之重, 提高铁路机务部门对机车运行安全的监控水平也成为

① 基金项目: 国家自然科学基金 (51775449, 51205323)
Foundation item: National Natural Science Foundation of China (51775449, 51205323)
收稿时间: 2019-09-06; 修改时间: 2019-10-08; 采用时间: 2019-10-21; csa 在线出版时间: 2020-04-05

当务之急.

铁路安全是一个复杂的系统工程, 由铁路运输生产人员、铁路设备和铁路环境这 3 部分所组成. 随着科技的发展, 车辆设备和铁路环境都有了较大的发展进步, 铁路运输生产人员逐渐成为提高铁路行车安全的重要因素. 因此对系统中的人加强监管就显得极为重要.

司机行为识别是目标检测的一个重要的应用场景. 在行车途中, 司机的驾驶行为是否符合安全规范直接关系到全车人的人身安全,所以对司机进行视频监控是一项重要的安防措施. 现有的行为分析以人工挑选分析 LKJ 中保存的视频数据为主, 抽样不均匀、分析质量不高等问题突出, 且只用于司机的绩效评定. 实现实时自动分析以及预判的安全驾驶监督系统已经成为铁路行业的迫切需求.

安全驾驶监督系统离不开目标检测技术的发展. 自 2012 年深度学习问世以来, 其在图像识别中表现出了极佳的效果, 并逐渐取代了传统特征机器学习的地位. 在目标检测领域, 一批高精度的算法不断刷新了识别精度的上线, 并逐渐为工业界所使用. 然而, 当前国内外大多数智能司机行为识别集中于汽车行业, 由于汽车内部狭小, 驾驶员人脸和手部等有明确的特征, 可以达到实时行为识别的目标[1–4]. 而对于机车来说, 现有的系统一般搭载于昂贵的远程大型服务器上, 只能在列车运行结束后收集运行保存的监控视频检测, 无法实现实时和随车检测, 只能进行司机非安全行为发生后的追查和定责.

因此, 采用一种低成本, 可随车一起运行的嵌入式设备, 在其上部署可以实时运行的监控系统, 以便实时传回违章信息便成为一项极有意义的工作. 为解决实时、高精度检测司机安全驾驶监督的问题, 本文选择玩手机这种较为难以肉眼识别的行为为例, 对目标检测中检测速度较快的 YOLOv3-tiny[5,6]算法进行了加速, 最终将其成功的部署在了计算能力较低的嵌入式设备上部署, 实现了较高精度下实时运行的目的.

1 实验数据

1.1 实验数据采集

为能够确保数据的多样性, 本文采集多个铁路局机务段货车及客车的驾驶室原始视频数据, 标记其中出现手机的图片共 5491 张. 标注, 从中随机选择 4919 张图片 (含 5890 个手机对象) 作为训练集, 剩余的 572 张图片 (含 892 个手机对象) 作为验证集, 如表 1 所示.

表 1 司机驾驶室图片数据集

参数	数量
图片总数	5491 张
无遮挡手机总数	2512 个
有遮挡数总数	4270 个

1.2 数据增强与标注

为了对抗过拟合、提高样本多样性和图片质量, 在数据集较少的情况下对数据进行增强, 采用双边滤波、随机水平镜像翻转、随机亮度改变和 0°~10°随机角度旋转对数据集进行扩增, 采用 LabelImg 进行人工画框标注, 只标注被遮挡面积比小于 0.5 的手机图片以确保精确性, 如图 1 所示.

图 1 增强和标注数据集

2 加速目标检测网络

2.1 YOLO 算法原理

为实现在线实时高精度检测, 本文选取精度较高同时推理速度较快的 YOLOv3 作为基础网络. YOLOv3 是一种 one-stage(单阶段) 的目标检测深度学习算法, 其集精确的和推理的快速性于一身, 自其提出至今, 在工业界已有了广泛的应用[7–10].YOLOv3 在 Coco 和 Pascal-voc 等开源数据集上均有不俗的表现, 其速度和精准度在相同输入大小的情况下均优于 SSD[11]、faster-RCNN[12]等主流算法, 尤其是 YOLOv3-tiny 网络

以其非常快的检测速度、较少的参数总量在嵌入式领域和边缘计算受到亲睐. YOLOv3 主要包含以下 3 种网络, 如表 2 所示.

表 2 YOLO 网络对比

模型	YOLOv3	YOLOv3-spp	YOLOv3-tiny
精度 (mAP)	0.553	0.601	0.331
检测速度 (FPS)	2	无法运行	17
计算复杂度 (Bn)	65.86	141.45	5.56
模型体积 (MB)	248.0	252.2	34.7
运行时模型占用显存/内存 (MB)	443.65	951.91	75.07

表 2 中精度一栏是在 Coco 测试数据集上得到的精度, 检测速度一栏是在嵌入式设备 jetson Nano 中测试得到. 可见虽然 YOLOv3 和 YOLOv3-spp 网络的精度较高, 但由于其复杂的网络在计算量受限的嵌入式设备很难达成实时运行的目标. 决定网络运行消耗的时间由模型在每一层计算所耗时间和前后网络层参数传递所耗时间决定, YOLOv3 和 YOLOv3-spp 较深的网络层数使得运行需要在完成上一层网络推理之后才能传入到下一层网络进行推理, 对于嵌入式这种并行处理能力差的设备来说, 如非直接缩减网络的层数, 则仅对每层的参数进行半精度优化和剪枝也无法缩减网络在前后传递中浪费的时间. 而 YOLOv3-tiny 的网络层数较少, 主要运行时间消耗在网络每一层计算中, 这就为优化提够了条件. 尽管精度仅为 YOLOv3 的一半, 但由于文中针对的是单一司机违章行为目标的目标检测而非 coco 数据集中 80 个类别的目标检测任务, YOLOv3-tiny 针对此类任务仍有较好的性能和向下优化的空间, 如下文中的实验所证. 此外, YOLOv3-tiny 在占用硬盘空间和运行内存均处于优势. 因此, 我们选用 YOLOv3-tiny 算法作为实现实时违章检测的主干网络.

YOLOv3-tiny 网络结构如图 2 所示. 与双阶段目标检测算法, 如 faster-RCNN 等网络不同的是, 在输入图片经过主体网络推理后, YOLOv3 网络直接通过 3 个不同尺度 (13×13, 26×26, 52×52) 的输出通道直接得到包含目标框坐标、目标置信度和目标框内物体分类在内的特征图, 最后经过非极大值抑制算法 (nms) 去掉重复目标后得到最终输出结果. 而 YOLOv3-tiny 为了节省计算量, 只在两个尺度上输出特征图, 分别为[B, 3×(4+1+C), 13, 13]和[B, 3×(4+1+C), 26, 26], 其中 B 为每次载入图片的数量, 3×(4+1+C) 为输出特征图的通道数, C 为每一类的概率, 3×(4+1+C) 代表特征图上每个 1×1 的点对三个 anchor(锚定框) 进行目标框的回归, 以此增强对不同大小物体的识别, 如图 3 所示.

2.2 模型剪枝

深度学习通过众多的参数计算推理得到预测结果, 其中有相当多的参数都是冗余且对预测结果无影响的. 当模型训练时, 原始网络需要一个足够大的参数空间以充分的寻找最优解. 但当模型训练完之后, 我们只需要保留最优的参数也一样可以达到和原参数空间一样的效果. 我们可以把剪枝视为在原有模型构成的参数空间里中搜寻了一条最有价值的计算路径[13], 这样模型的精度不会降低, 而使模型运行的更有效率, 这就是进行模型剪枝的意义. 本文采用模型剪枝的方法对训练后模型进行保持原精度情况下计算量和参数总量的缩减. 模型剪枝可以根据细粒度和粗粒度分为权重剪枝和通道剪枝两类, 通道剪枝方法由于其独有的简单、可行性、总计算量较小和不需要特殊的硬件库的优势, 在近几年得到了相当迅速的发展, 并已实际应用于一些工程中[14–16]. 通道剪枝本质上从通道这一层面上对卷积层中的卷积核的重要性进行区分, 去除对网络输出结果影响小的卷积核, 以此实现计算总量和模型体积的缩小.

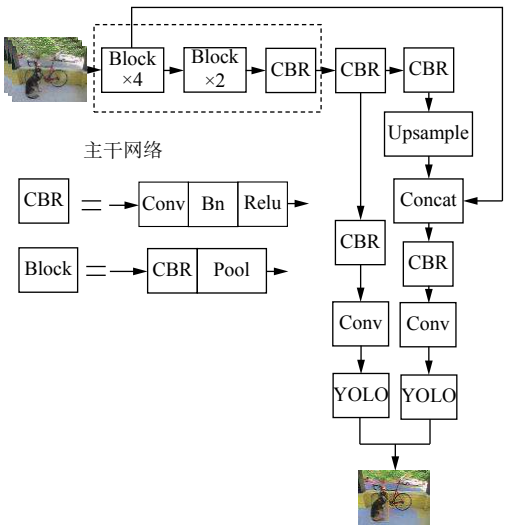

图 2 YOLOv3-tiny 网络

我们采用了对 Batch Normalization 层 (批标准化层, 下称为 BN 层) 进行 L1 惩罚的方式进行通道剪枝. 在神经网络的计算中, BN 层实际上进行了两步运算, 如式 (1) 所示: (1) 对输入特征图所有参数规整到均值为 0, 方差为 1 的正态分布范围内. (2) 让每个规整后所

有参数在训练过程中学习到对应的两个调节因子 γ 和 β, 对标准化后的值进行微调, 使之更适于梯度下降.

$$\begin{cases}\tau = \dfrac{a_i - u}{\sigma_i} \\ a_i^{norm} = \gamma_i \cdot \tau + \beta_i\end{cases} \tag{1}$$

其中, a_i 代表输入的每个通道的特征图, 为 $B^l \times H^l \times W^l$; u, σ_i 分别是均值和方差; γ_i 和 β_i 是每个通道特征图所对应的两个调节因子. 实际上, γ 可以视为 BN 层特征图每一通道的权重, 如果当前输入的通道 C_i 对应的权重 γ_i 出现 $\gamma_i=0$ 或 $\gamma_i\approx 0$ 的情况, 那么就会有 $\gamma_i \times \tau = 0$, 特征图对应的输出通道即全为常数 0, 不会再对接下来的运算产生影响. 因此, 我们可以利用 BN 层中的缩放因子 γ 衡量特征图每个通道的重要性, 当 $\gamma_i=0$ 或者 $\gamma_i\approx 0$ 时, 即可剪去 γ_i 对应特征图的通道, 最终剪去 $\gamma=0$ 对应的上下卷积层的卷积核, 以此完成减少计算量和缩小模型体积的任务, 如图 4 所示.

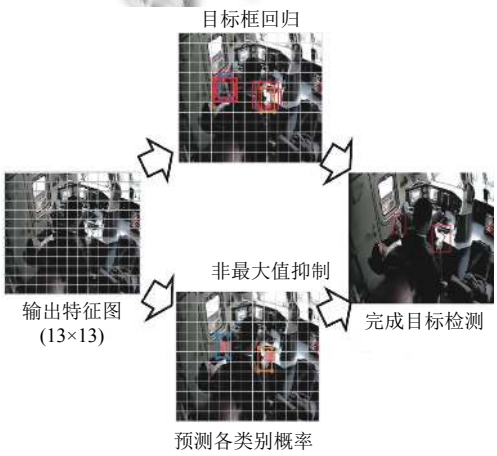

图 3 YOLO 检测过程图

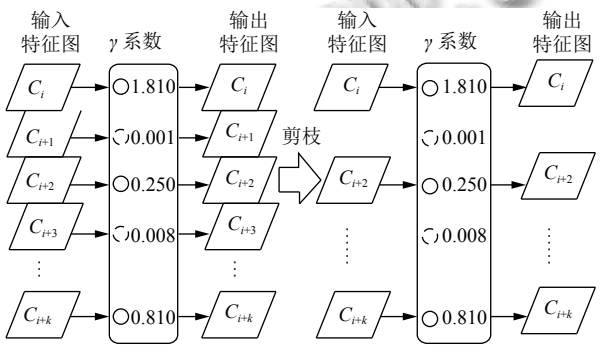

图 4 BN 层剪枝示意图

然而, 一般训练之后神经网络中的 γ 值通常呈正态分布, 并不会有很多参数等于或者靠近 0. 因此, 我们需要在训练网络的同时减小 γ, 这被称之为稀疏化训练. 通常使用 L1 次梯度法稀疏每个 BN 层的 γ 值, 如式 (2) 所示.

$$\gamma_{l+1} = (|\gamma_l - u\nabla l| - \eta) \cdot sgn(\gamma_l - u\nabla l) \tag{2}$$

其中, u 代表损失函数的学习率; $u\nabla l$代表训练中原始损失函数的梯度, 它是从损失的反向传播中得出的; η 是超参数, 决定了 L1 次梯度法的每次梯度下降大小, 根据经验, η 应在 1e–4~1e–5 之间取得[15]; $sgn(\gamma_l - u\nabla l)$ 决定了 L1 次梯度法的损失方向.

2.3 TesnorRT 模型简化及半精度加速

TensoRT 是 NVIDIA 为深度学习的高速推理需求所推出的一个高性能深度学习推理平台. 它包括深度学习推理优化器和运行器, 可为深度学习推理应用提供 INT8 和 FP16(半精度) 优化, 例如视频、语音识别、推荐系统和自然语言处理等. 在深度学习的部署中, 降低精度推断可显着减少应用程序延迟, 这是许多实时服务, 自动和嵌入式应用程序的要求.

在搭建 TensorRT 加速引擎时, 需要将原模型转化为 TensorRT 可以读取的形式, 如 caffe、onnx 的模型结构. TensorRT 自带的接口 NvcaffeParser 提供了 Caffe 结构模型中卷积层、激活层和池化层等常见层的接口, 而 YOLOv3-tiny 中的 Upsample 层 (上采样层) 及 YOLO 层 (检测层) 需要自行编写后自定义插入. Upsample 层实现了数据图 H, W 方向上的调整, 放大输入的特征图以适应联结、矩阵加法等操作, 常见的 Upsample 方法有线性插值、最近邻值等. YOLO 层实现了从特征图中挑选置信度 (confidence) 大于预设值的目标输出的目的, 并将挑选出的特征与锚定框大小计算回归出目标框真实位置和长宽值. 本文将最近邻值的 Upsample 层和 YOLO 层添加到了 TensorRT 加速引擎中.

TensorRT 通过实现模型简化和半精度对模型进行加速, 其中模型简化通过整合卷积层、激活层和 BN 层为 CBR 层实现; 半精度加速通过在平台支持的情况下, 将数据精度需求从 32 位 float (浮点数) 降低为 16 位 float, 可极大的提升计算效率.

2.4 总体流程

我们提出的算法总体流程如下: (1) 首先进行稀疏训练, 在训练模型的同时使 BN 层 γ 因子尽可能地收缩; (2) 当稀疏训练完成时, 执行修剪以移除冗余参数; (3) 微调修剪后的模型以获得最终模型; (4) 搭建

TensorRT 推理加速引擎, 编写自定义层; (5) 将模型读入 TensorRT 加速引擎中, 部署在嵌入式设备上. 训练流程图如图 5 所示.

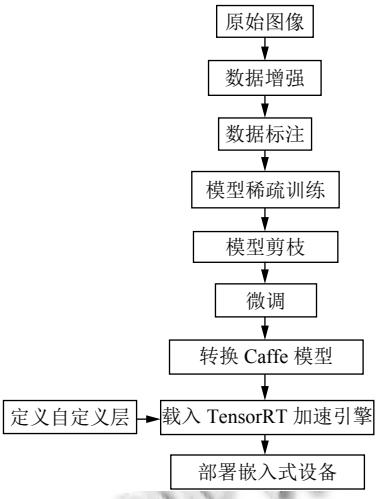

图 5 训练部署流程

3 实验结果与分析

本文使用了 Pytorch 框架完成 YOLO 算法的搭建, 主要训练硬件设备为: ①处理器: Intel 9400f; ②GPU 显卡: RTX2070 8G. 操作系统为 Ubuntu16.04, python 环境为 python3.6.8. 部署嵌入式平台为较为廉价、适于工业部署的 NVIDIA Jetson Nano, 如图 6 所示. Jetson Nano 是 NVIDIA 公司 2019 年新推出的嵌入式高性能开发板, 其采用 NVIDIA Maxwell™架构, 配备 128 个 CUDA 核心, 内存为 4 GB, 搭载的运行环境为 JetPack 4.2.

图 6 Jetson Nano

为提高显存利用率, 增加 batch size(批大小), 采用了 NVIDIA 的 apex 技术进行混合精度训练. 选取准确率 P (precision)、召回率 R (recall)、平均精确率均值 mAP (mean Average Precision)、$F1$ 值、检测速度和参数总量作为评价准则. 其中, precision 指的是被正确检测出的物体占总被检测出的物体的比例, recall 指被正确检测出的物体占验证集中所有物体的比例. precision 和 recall 一般情况下是成反比关系, 即 recall 越大, precision 越小, 反之同理. 而 mAP 为目标检测任务中最重要的指标, 决定了检测的效果. mAP 一般通过改变检测时置信度 (confidence) 阈值得到 Precision-recall 曲线, 计算 Precision-recall 曲线的面积得到 mAP 值. 为了在计算机中更快速精确的求 mAP 的大小, 通常使用插值近似的方法, 如式 (3) 所示.

$$mAP=\frac{1}{n}\sum_{i=1}^{n}\int_{0}^{1}P(r)dr\approx\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{M}P_{inperp(k)}\Delta r(k) \quad (3)$$

本文同时以正常训练模式和稀疏化训练模式训练了两个 YOLOv3-tiny 模型, 设置稀疏训练超参数 η=1e–4 进行训练. 表 3 展示了 YOLOv3-tiny 网络稀疏化训练后的各项指标, 可见稀疏化训练后的各项指标均处于相当优秀的水平, 0.965 的 mAP 保证了高精度检测的目的, 甚至要高于正常训练, 这是因为稀疏化训练也可以视为进行了 L1 正则化, 降低了模型对训练数据的过拟合. 图 7 显现了稀疏化训练后 γ 系数的稀疏化水平. YOLOv3-tiny 网络共有 11 层 BN 层, 这 11 层 BN 层共有 4512 个 γ 系数, 40% 的 γ 参数的绝对值下降到 0 值附近, 满足进行剪枝的条件. 随后进行剪枝并进行微调.

表 3 YOLOv3-tiny 训练结果

训练方式	P	R	mAP	$F1$
稀疏化训练	0.927	0.976	0.965	0.951
正常训练	0.892	0.987	0.962	0.937

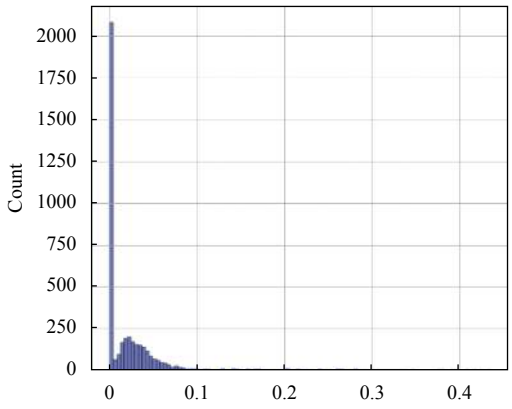

图 7 稀疏化训练后全 γ 系数分布图

表 4 展示了剪枝后各个卷积层变化情况, 剪枝后的模型各卷积层通道大小由之前的 2 的 n 次方数变为不再有规律可循, 这便是非结构剪枝在现有模型中搜寻了一条最为有效的计算路径的结果.

表 4 剪枝后的模型各卷积层大小

网络层	剪枝前	剪枝后
Conv1	16	13
Conv2	32	29
Conv3	64	50
Conv4	128	97
Conv5	256	180
Conv6	512	319
Conv7	1024	440
Conv8	256	141
Conv9	512	227
Conv10	128	68
Conv11	256	155

表 5 展示了剪枝后并微调后改进模型 (下文称为 Prune-YOLOv3-tiny) 在 Jetson Nano 上的表现. 可见在微调后, Prune-YOLOv3-tiny 模型的精度略有上升, 而在检测速度相较原模型提高了 76.5%, 计算总量降低为原模型的 43.2%, 在参数总量上降低为原模型的 31.4%. 尽管只有 40% 的 γ 参数被剪去, 模型参数总量依然下降很大. 由表 4 可见, Conv1、Conv2、Conv3 等参数较少的卷积层被剪去的较少, 而如 Conv6、Conv7、Conv8 等参数较多的卷积层剪去的参数较多, 参数最多的 Conv7 有高于一半的参数被剪去, 这是剪枝后模型参数总量要更少的原因. 而对于计算复杂度, 参数最多的 Conv7 层并非是计算量最大的层, 考虑计算量需要综合每层输入输出的特征图大小. 对 YOLOv3-tiny 模型来说, Conv6 是模型计算量最大的层并没有如 Conv7 一样被剪去一半以上, 这是计算总量的缩小程度低于参数总量的原因.

表 5 原网络与改进网络对比

模型	YOLOv3-tiny	Prune-YOLOv3- tiny
P	0.927	0.928
R	0.976	0.976
mAP	0.965	0.966
$F1$	0.951	0.951
检测速度 (FPS)	17	30
计算总量 (Bn)	5.46	2.36
参数总量 (MB)	34.7	10.9

随后, 将原 YOLOv3-tiny 模型加载到搭建好的 TensorRT 加速引擎中以生成半精度加速的模型. 表 6 展现了 YOLOv3-tiny 模型在 TensorRT 加速引擎中的效果. 相较于原模型, TensorRT 加速的半精度模型的精度没有变化, 检测速度提高 52.9%, 而参数总量没有明显下降.

表 6 YOLOv3-tiny 在不同推理平台上对比

运行框架	Pytorch	TensorRT
精度	单精度	半精度
P	0.927	0.927
R	0.976	0.976
mAP	0.965	0.965
$F1$	0.951	0.951
检测速度 (FPS)	17	26
参数总量 (MB)	34.7	33.5

最后, 我们将 Prune-YOLOv3-tiny 模型加载到搭建好的 TensorRT 加速引擎中完成最终的加速模型 (下文称为最终模型), 表 7 展现了最终模型的效果评估. 经过模型剪枝和半精度加速的最终模型相较于原模型精度没有下降, 检测速度提高了 117.6%, 而在参数总量上降低为原模型的 39.5%. 图 8 展示了最终模型的识别效果.

表 7 原模型与最终加速模型对比

运行框架	Pytorch	TensorRT
模型	YOLOv3-tiny	Prune-YOLOv3-tiny
精度	单精度	半精度
P	0.927	0.928
R	0.976	0.976
mAP	0.965	0.966
$F1$	0.951	0.951
检测速度 (FPS)	17	37
参数总量 (MB)	34.7	13.7

图 8 加速模型检测效果

4 结语

在这项工作中, 我们成功的对目标检测中经典的

深度学习神经网络 YOLOv3-tiny 进行了通道剪枝和半精度加速, 在精度不变的情况下减少了改进后神经网络的计算总量和参数总数. 我们在计算能力较低的嵌入式设备上成功部署了剪枝后的模型. 所提出的模型在嵌入式设备可以达到 37 帧每秒的检测速度,与现有模型相比, 我们的模型除了需要在稀疏化训练中增加一个 η 超参数外, 在速度和体积上均占有优势, 而精确率、召回率和平均精确率均值不变, 实现了高精度下实时检测的目的.

参考文献

1 宫法明. 交通驾驶员脸疲劳驾驶行为优化图像识别. 计算机仿真, 2015, 32(11): 199–202. [doi: 10.3969/j.issn.1006-9348.2015.11.045]

2 杨林川. 基于深度神经网络的司机行为识别技术研究与实现[硕士学位论文]. 成都: 电子科技大学, 2018.

3 李俊俊, 杨华民, 张澍裕, 等. 基于神经网络融合的司机违规行为识别. 计算机应用与软件, 2018, 35(12): 228–233, 325.

4 黄占鳌, 史晋芳. 多特征融合的驾驶员疲劳检测研究. 机械科学与技术, 2018, 37(11): 1750–1754.

5 Redmon J, Divvala SK, Girshick R B, *et al*. You only look once: unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA. 2016. 779–788.

6 Kim DH. Evaluation of COCO validation 2017 dataset with YOLOv3. Evaluation, 2019, 6(7): 10356–10360.

7 黎洲, 黄妙华. 基于 YOLO_v2 模型的车辆实时检测. 中国机械工程, 2018, 29(15): 1869–1874. [doi: 10.3969/j.issn.1004-132X.2018.15.015]

8 薛月菊, 黄宁, 涂淑琴, 等. 未成熟芒果的改进 YOLOv2 识别方法. 农业工程学报, 2018, 34(7): 137–179. [doi: 10.11975/j.issn.1002-6819.2018.07.018]

9 张富凯, 杨峰, 李策. 基于改进 YOLOv3 的快速车辆检测方法. 计算机工程与应用, 2019, 55(2): 12–20. [doi: 10.3778/j.issn.1002-8331.1810-0333]

10 刘军, 后士浩, 张凯, 等. 基于增强 Tiny YOLOV3 算法的车辆实时检测与跟踪. 农业工程学报, 2019, 35(8): 118–125. [doi: 10.11975/j.issn.1002-6819.2019.08.014]

11 Liu W, Anguelov D, Erhan D, *et al*. SSD: Single shot multiBox detector. European Conference on Computer Vision. Berlin, Germany. 2016. 21–37.

12 Girshick RB. Fast R-CNN. 2015 IEEE International Conference on Computer Vision. Santiago, Chile. 2015. 1440–1448.

13 Liu Z, Sun MJ, Zhou TH, *et al*. Rethinking the value of network pruning. International Conference on Learning Representations. New Orleans, LA, USA. 2019. 1–11.

14 He YH, Lin JJ, Liu ZR, *et al*. AMC: AutoML for model compression and acceleration on mobile devices. European Conference on Computer Vision. Berlin, Germany. 2018. 815–832.

15 Liu Z, Li JG, Shen ZQ, *et al*. Learning efficient convolutional networks through network slimming. 2017 IEEE International Conference on Computer Vision (ICCV). Venice, Italy. 2017. 2755–2763.

16 Ye JB, Lu X, Lin Z, *et al*. Rethinking the smaller-norm-lessinformative assumption in channel pruning of convolution layers. International Conference on Learning Representations. Vancouver, BC, Canada. 2018. 1–13.