Computer Science and Application 计算机科学与应用, 2021, 11(5), 1291-1300
Published Online May 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.115131

Hans汉斯

基于改进YOLOv4-Tiny算法的移动端实时司机违章行为检测

秦丹峰，尹相辉，龚学余

南华大学电气工程学院，湖南 衡阳
Email: 1083610012@qq.com, yinxh@usc.edu.cn, gongxueyu@126.com

收稿日期：2021年4月17日；录用日期：2021年5月11日；发布日期：2021年5月18日

摘　要

司机违章行为在日常生活中屡见不鲜，为解决在移动端上高精度、实时监测司机的驾驶行为的问题，基于一种轻量级的目标检测算法YOLOv4-tiny，通过引入跨通道部分的连接机制，减少了模型的参数，在YOLOv3-tiny的基础上改进残差的思想，并在损失函数上将定位损失替换为CIOU损失实现更精确的边框，最后通过知识蒸馏的手段，利用教师模型YOLOv4指导tiny训练进一步提升其性能。通过在公开数据集和自建数据集上的多项实验对比结果显示，YOLOv4-tiny达到了更高的精度(分别提升7%和10%)，实现了对算力较低的嵌入式设备或移动端的实时检测。

关键词

司机行为检测，YOLOv4-Tiny，知识蒸馏，实时检测

Real-Time Drivers' Violation Detection on Mobile Terminal Based on Improved YOLOv4-Tiny

Danfeng Qin, Xianghui Yin, Xueyu Gong

School of Electrical Engineering, University of South China, Hengyang Hunan
Email: 1083610012@qq.com, yinxh@usc.edu.cn, gongxueyu@126.com

Received: Apr. 17th, 2021; accepted: May 11th, 2021; published: May 18th, 2021

Abstract

Driver violations are common in daily life. In order to solve the problem of monitoring drivers'

文章引用: 秦丹峰, 尹相辉, 龚学余. 基于改进YOLOv4-Tiny算法的移动端实时司机违章行为检测[J]. 计算机科学与应用, 2021, 11(5): 1291-1300. DOI: 10.12677/csa.2021.115131

driving behavior on mobile with high accuracy and in real time, based on a lightweight target detection algorithm YOLOv4-tiny, the parameters of the model are reduced by introducing a connection mechanism across channel sections, improving the idea of residuals based on YOLOv3-tiny, and replacing the localization loss with CIOU loss in the loss function to achieve more accurate edges, and finally, by means of knowledge distillation, the teacher model YOLOv4 is used to guide tiny training to further improve its performance. The results of several experiments comparing public and self-built datasets show that YOLOv4-tiny achieves higher accuracy (7% and 10% improvement, respectively) and realizes real-time detection on embedded devices or mobile with low computing power.

Keywords

Driver Behavior Detection, YOLOv4-Tiny, Knowledge Distillation, Real-Time Detection

1. 引言

随着计算机和人工智能的飞速发展，深度神经网络已经成为计算机视觉领域近年来最为热门的研究方向，并且在机器视觉、语言识别和文本处理等诸多应用领域中取得了重大突破。

目标检测是计算机视觉领域的一个经典的任务，是进行场景内容分析和理解等高级视觉任务的基本前提。现有的目标检测算法的框架主要分为两大类：一阶段算法和二阶段算法。二阶段算法，如 Faster R-CNN [1]利用选择性搜索算法从图像提取候选框，对图像进行一次特征提取，通过空间金字塔池化映射为固定长度的特征向量，再通过全连接层进行分类来预测边界框的坐标，从而对候选框进行修正。基于候选区域的算法在主流数据测试集中达到了非常高的检测精度。但是二阶段算法处理速度较慢，没有先进的硬件设备使得该类算法很难在实时的场景中落地。而一阶段算法的性能达到实时性要求且精度满足大部分需求。典型的一阶段算法如 YOLO 系列算法，通过将候选区域与分类合二为一，利用整张图的特征来预测多个类别的概率和和每个边界框的置信度。YOLO 系列经过 v1~v4 [2] [3] [4] [5]多个版本的迭代，已经在精度方面逐渐逼近二阶段算法，由于极高的实时性可以轻松在应用在各个类型的场景中。

司机违章行为检测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。一般的车辆行驶中，在特定路段设置的抓拍系统只能对固定范围进行监控，仅能在一定程度上约束司机的违章行为。而日常的大部分时间里，在司机行驶过程中，通常会出现打电话、抽烟及喝水等违章行为，这些属于违章行为。在未被监控的时段，司机的行为仅靠个人的自觉遵守，很难对其进行监管。特别在公交系统，出租车、客运车、货车、冷链车等大型交通工具中，这些违章行为存在极大的风险系数，危害着公共安全。

安全驾驶监督系统离不开目标检测技术的发展。当前国内外大多数智能司机行为检测集中于汽车行业，由于驾驶舱内部狭小，司机人脸和手部等有明确的特征，可以达到实时行为检测的目标。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，本文针对打电话、喝水、抽烟三种违章行为进行算法设计，采用轻量级的目标检测算法 YOLOv4-tiny，成功地在计算能力较低的嵌入式设备上部署，并实现了较高精度下实时运行的目的。

2. 目标检测算法

2.1. CSPDarknet-53

YOLO 系列目标检测算法作为一阶段算法实时性的代表，常用于大量的实际场景中。经过多个版本的迭代，速度不断提高，而在精度方面正不断的赶超二阶段的复杂网络结构算法。DarkNet 网络作为一种小众的 Backbone，在图像分类的性能堪比 ResNet [6]系列的卷积神经网络，应用在 YOLOv2 及之后的版本。YOLOv3 的作者将 DarkNet 网络从最初 DarkNet-19 更新到 DarkNet-53，增加了卷积层的深度和宽度，同时引入了残差单元，在通用特征表示方面得到了极大的提升。

最近的研究从网络层的梯度流动方向出发，在减少网络计算量的同时，实现一个更丰富的梯度结合。CSPNet [7]通过切分梯度流动，使得梯度流动是通过不同的网络路径进行传播。CSPDarkNet-53 在原始结果 DarkNet-53 基础上改进了每个阶段的梯度，在每次下采样后将输出特征图的通道一分为二，其中一个路径经过该阶段的所有残差单元，在该阶段的末端与另一个路径进行合并。如图 1 所示。

该主干网络增强了 CNN 的学习能力，使得在轻量化的同时保持准确性，降低了计算瓶颈和内存成本。

DarkNet-53 的卷积层采用卷积 + BN 层 + Leaky RELU 的组合，而在 CSPDarkNet-53 中将 Leaky RELU 替换成 Mish，关于 Mish 激活函数的优越性质有如下三点：无上界且有界、非单调函数、无穷阶连续性和光滑性。Mish 激活函数的应用，在 YOLOv4 中得到了明显的性能提升。

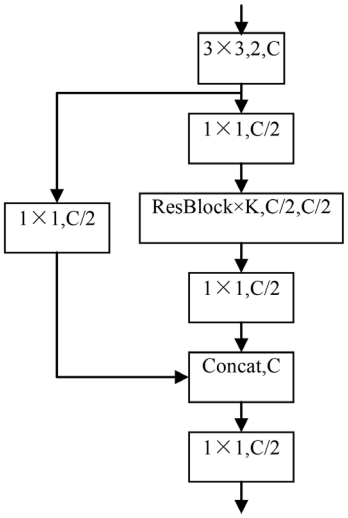

Figure 1. CSP structure of CSPDarkNet-53
图 1. CSPDarkNet-53 的 CSP 结构

2.2. 检测层与损失函数

在目标检测领域，为了更好的提取融合特征，通常在 Backbone 和输出层之间插入一些层，这个部分称为 Neck。YOLOv4 使用了 SPP [8]模块，相比单纯的使用 k*k 最大池化的方式，更有效的增加主干特征的接收范围，显著的分离了最重要的上下文特征。

另外，YOLOv4 在 FPN 层的后面还添加了一个自底向上的特征金字塔。其中包含两个 PAN 结构。

这样结合操作，FPN 层自顶向下传达强语义特征，而特征金字塔则自底向上传达强定位特征，两者相辅相成，从不同的主干层对不同的检测层进行参数聚合，在输出层进行边框回归时，传统的目标检测模型(比如 YOLOv3)等，直接根据预测框和真实框的中心点坐标以及宽高信息设定 MSE (均方误差)损失函数。MSE 损失函数将检测框中心点坐标和宽高等信息作为独立的变量对待的，而实际上两者之间是有关联的。从直观上来考量，框的中心点和宽高的确存在着一定的关系。使用 IOU 损失相比 MSE 损失更能表征其联系，CIOU [9]损失在 IOU 损失的基础上考虑了边框的重合度、中心距离和宽高比的尺度信息，最终 YOLOv4 在设计损失函数时，在定位损失部分使用了 CIOU 损失，即 $L(bbox)$。总损失函数为：

$$Loss = L(bbox) + L(conf) + L(cls) \tag{1}$$

$$L(bbox) = \lambda_{coord} \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{obj} \left(1 - \mathrm{IOU}\left(b_i, \hat{b}_i\right) + \frac{\rho^2\left(b_i, \hat{b}_i\right)}{C^2} + \frac{\upsilon^2}{\left(1 - \mathrm{IOU}\left(b_i, \hat{b}_i\right)\right) + \upsilon} \right) \tag{2}$$

$$\begin{aligned} L(conf) = &-\sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{obj} \left[\hat{c}_i \log(c_i) + (1-\hat{c}_i) \log(1-c_i) \right] \\ &-\lambda_{coord} \sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{noobj} \left[\hat{c}_i \log(c_i) + (1-\hat{c}_i) \log(1-c_i) \right] \end{aligned} \tag{3}$$

$$L(cls) = -\sum_{i=0}^{K\times K} \sum_{j=0}^{M} I_{ij}^{obj} \left[\hat{p}_i(c) \log\left(p_i(c)\right) + \left(1-\hat{p}_i(c)\right) \log\left(1-p_i(c)\right) \right] \tag{4}$$

损失函数由定位损失 $L(bbox)$、置信度损失 $L(conf)$和分类损失 $L(cls)$构成。其中，定位损失采用了 CIOU 损失，置信度损失和分类损失都使用了交叉熵损失。IOU 为真实框与预测框的交并比，ρ 表示欧式距离，C 表示真实框与预测框的最小外界矩形的对角线距离，υ 用来度量长宽比的相似性，λ 为超参数，K 为特征图的大小，M 为锚框的数量，第 ij 个位置为目标时等于 1，c 为类别，$p(c)$代表类别的置信度。

2.3. 改进的 YOLOv4-Tiny

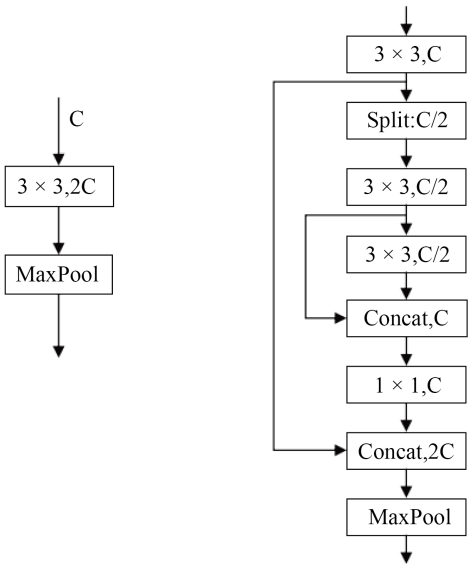

Figure 2. CSP structure of YOLOv4-tiny
图 2. YOLOv4-tiny 的 CSP 结构

为了方便 YOLOv4 在移动平台或边缘设备进行部署，使用一个更精简的模型 YOLOv4-tiny，该检测网络在 YOLOv3-tiny 的基础上，引入了残差单元，采用 YOLOv4 类似的 CSP 结构，在原有 7 层卷积层

增加至 18 层，改进的卷积网络能提取更多的语义信息，特征提取能力提升显著。

YOLOv4-tiny 的骨干网络使用残差单元进行特征表示，并进行微小改进，将 3 × 3 卷积前后的特征进行堆叠，保留了原始的信息。类似 CSP 的结构，如图 2 所示，跨阶段的特征融合，在残差单元之前，摒弃一半通道的特征，在参数上更精简，梯度传导更加高效，去除了大量的计算冗余。整个骨干网络保留了 YOLOv4 原有架构 4 个阶段的形式，32 倍下采样，使用两个分支进行预测，覆盖了所有多个尺度的特征。

YOLOv3-tiny 仅使用 3 × 3 + 最大池化的组合方式进行特征提取与尺度变换，而 YOLOv4-tiny 考虑了更高效的特征表示，并使用梯度截断使得大量的冗余梯度被剔除，两者的特征表示与梯度传导如下：

$$X_k = MaxPool\left(W_k * X_{k-1}\right) \tag{5}$$

$$\begin{cases} X_{k-4} = W_{k-4} * X_{k-5} \\ X_{k-3} = W_{k-3} * X_{k-4}^{\left[0:\frac{C}{2}\right]} \\ X_{k-2} = W_{k-2} * X_{k-3} \\ X_{k-1} = W_{k-1} * \left[X_{k-3}, W_{k-2} * X_{k-3}\right] \\ X_k = MaxPool\left(\left[X_{k-4}, X_{k-1}\right]\right) \end{cases} \tag{6}$$

其中，*为卷积操作，$X^{[0:C]}$表示只选取 X 中 0~C 的通道，$[X_0,X_1]$代表 X_0 与 X_1 的堆叠操作(即 Concat)，$MaxPool(X)$为池化，公式(5)为 YOLOv3-tiny 基本结构的特征表示，公式(6)为 YOLOv4-tiny 的 CSP 结构的特征表示，如果使用反向传播算法更新权重，则上述两者的权重更新方程可写为：

$$W_k' = f\left(W_k, g_{k-1}\right) \tag{7}$$

$$\begin{cases} W_{k-4}' = f\left(W_{k-4}, g_{k-5}\right) \\ W_{k-3}' = f\left(W_{k-3}, g_{k-4}^{\left[0:\frac{C}{2}\right]}\right) \\ W_{k-2}' = f\left(W_{k-2}, g_{k-3}\right) \\ W_{k-1}' = f\left(W_{k-1}, g_{k-3}, g_{k-2}\right) \\ W_k' = f\left(W_{k-4}, g_{k-1}\right) \end{cases} \tag{8}$$

其中，$f(X)$为权重更新函数，g_i 代表反向传播到第 i 层的梯度，从公式(7)中可以发现当串联多个基本结构时，大量的梯度信息被重新用于更新不同密集层的权重。这将导致不同的卷积层重复学习复制的梯度信息。而公式(8)使用跨阶段部分残差连接，并与最原始层进行堆叠后经过另一个过渡层 W_{k-1}，可以看到来自残差变换层的梯度是独立积分的。另一个路径中没有经过残差变换层的特征图 X_{k-1} 也是独立积分。因此用于更新权重的梯度信息中，两者之间都不包含属于彼此重复的梯度信息。

跨阶段的特征融合，在残差单元之前，摒弃一半通道的特征，在参数上更精简，梯度传导更加高效，去除了大量的计算冗余。整个骨干网络保留了 YOLOv4 原有架构 4 阶段的形式，32 倍下采样，使用两个检测分支进行预测，可覆盖多个尺度的特征。

该算法训练过程的数据增强采用了马赛克增强，在基于 CutMix [10]的两张图片拼接的基础上，采用 4 张随机缩放、随机裁剪或随机排布的图片进行拼接。这极大地丰富了检测数据集，特别是通过随机缩放增加了更多的小目标，使得网络的鲁棒性更好。

对于损失函数，需要用预测概率去拟合真实概率，而拟合 one-hot 的真实概率函数会带来两个问题。深度学习的样本中通常会存在少量错误标签，这些错误标签会影响到预测的效果。采用标签平滑策略在

训练时假设标签可能存在错误，避免“过分”相信训练样本的标签。

3. 司机违章行为检测

3.1. 行为分析

对于司机的违章行为，分为三个类别：打电话(包括玩手机)、喝水、吸烟。通过给共用车辆装备车载终端等方式，面向车内司机部署红外摄像头持续监控司机的异常行为，被系统成功识别的行为进行语音提示并将异常行为上报至云端。

YOLO 系算法由于其可观的检测性能以及较快的检测速度被广泛应用于不同的物体检测实际场景中。YOLO 的网络架构中采用简易的残差单元和 4 个阶段结构使得在模型转换以及边缘端部署具备良好的保障，因此司机的违章行为检测采用物体检测的方法，通过对红外摄像头的视频帧逐帧处理，对图像中的手机，水瓶，香烟等小目标进行定位与分类，并融合连续帧之间的多个检测结果，确定最终司机的驾驶状态。

3.2. 行为数据集制作

通用数据集中关于手机，水瓶，香烟等物体的图片集，由于背景的复杂以及图片类型，形状的差异，使得该数据集并不适用于驾驶舱的红外特定场景中。

通过在驾驶舱中放置多个固定的红外相机，持续采集不同司机的规定动作，分别录制被采集人不用人脸角度下左右手手持手机、水瓶、香烟等动作。具体设计的动作考虑到了行车过程中可能会发现的情形，因此为保证最终检测算法的性能，设计了多种动作。关于持手机的动作，分为放置耳边打电话、放置耳边听语音、放置嘴边发语音、玩手机等；关于持香烟的动作，分为嘴叼烟、不同夹烟手势吸烟等；关于持水瓶的动作，直接录制拿起水瓶到喝的过程，如图 3 所示。

Figure 3. Different action design of calling, smoking, drinking
图 3. 不同的打电话、抽烟、喝水动作设计

利用 Opencv 录制分辨率为 1280 × 720，帧率为 25，时长为 30 分钟的被采集人视频，使用 VOTT 进行标注，最终的违章动作数据集分为 3 个类别，分别为打电话、喝水、抽烟。总标注图片数量为 5000 张。

3.3. 移动平台部署

TensorFlow Lite 是一种用于边缘端推断的开源轻量级深度学习框架。可以把训练好的 TF 模型通过转化、部署和优化三个步骤，达到提升运算速度，减少内存、显存占用的效果。

在 TensorFlow 的深度学习框架下，利用大型图像分类数据集 ImageNet2012 [11]预训练 CSPDarkNet-53，通过迁移学习的方式在违章动作数据集上微调检测网络 YOLOv4，将其作为教师模型将知识蒸馏给学生模型 YOLOv4-tiny，将训练好的 TF 框架模型转换至 TFLite，将 Python 端的整套模型推理流程移植到 Android 端，采用 JAVA 语言调用 TFLite 模型推理框架，实现 YOLOv4-tiny 在移动端的部署。

3.4. 应用层逻辑

在驾驶舱中安装车载终端以及红外摄像头，当系统稳定运行时，司机违章动作识别系统读取摄像头视频帧，对图像像素进行检测，从而定位图像中的物体以及预测该物体的类别。

漏检与误检是目标检测难以解决的两大难题，为了保障算法在车载终端上的稳定性，直接采用检测结果判断司机状态显然不合适。在实际的服务中，通过建立一个缓存队列，保存当前检测结果和历史检测结果。根据检测类别在缓存队列的比例输出当前的司机状态，对于违章动作给予告警以及信息上报。

应用层的主要流程：挂载一个 APP 的后台服务进行算法部署，通过安卓 API 访问系统摄像头，读取每一视频帧，对图像进行预处理，输入到 TFLite 模型推理框架，将预测的检测结果输送至缓存队列，判定当前驾驶员的状态，从而实现对司机违章行为的持续监测。

4. 实验

4.1. 评价标准

深度学习模型需要在“验证/测试”数据集上来评估性能，性能衡量使用各种统计量如准确度，精确率，召回率等。选择的统计量通常针对特定应用场景和用例。对于每个应用场景，选择一个能够客观比较模型的度量指标非常重要。

mAP (Mean Average Precision)是目标检测问题中的最常用评估指标。多个类别物体检测中，每一个类别都可以根据召回率和准确率绘制一条曲线，AP 就是该曲线下的面积，mAP 为多个类别 AP 的平均值。

4.2. 公开数据集的性能对比与分析

为了更直观的评价算法之间的差异，在 PASCAL VOC 和 COCO 两大公开数据集上分别对比 YOLOv3 与 YOLOv4 的性能。

在 VOC 数据集上，训练集采用 VOC2007-train 和 VOC2012-train，在 VOC2007-test 上测试 mAP，性能对比如图 4 所示。

在 COCO 数据集中，YOLOv3 和 YOLOv4 对比了 320 × 320、416 × 416、512 × 512 三种不同尺度，在 YOLOv3-tiny 和 YOLOv4-tiny 则使用 416 × 416 单尺度测试，如图 5 所示。

从实验数据中可以看出，VOC 数据集上，YOLOv4 相比于 YOLOv3，其性能提升了约 5%，精简版 YOLOv4-tiny 在 YOLOv3-tiny 的基础上提升了将近 10%，与 YOLOv3 仅相差约 5%。而在 COCO 数据集上，YOLOv4 在 3 个尺度上的性能全面优于 YOLOv3，YOLOv4-tiny 的性能超越了 YOLOv3-tiny 近 7%。

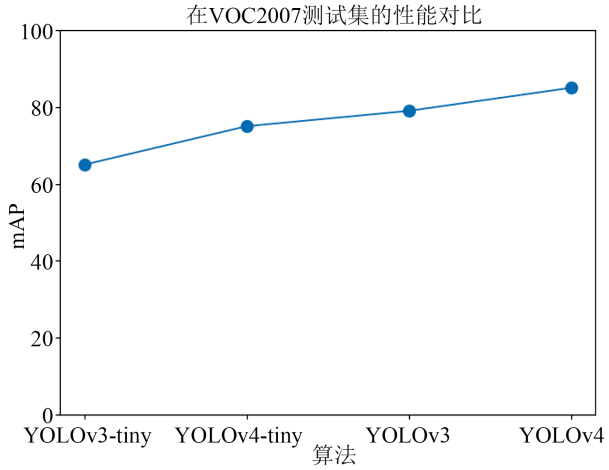

Figure 4. Performance comparison on VOC dataset
图 4. VOC 数据集上的性能对比

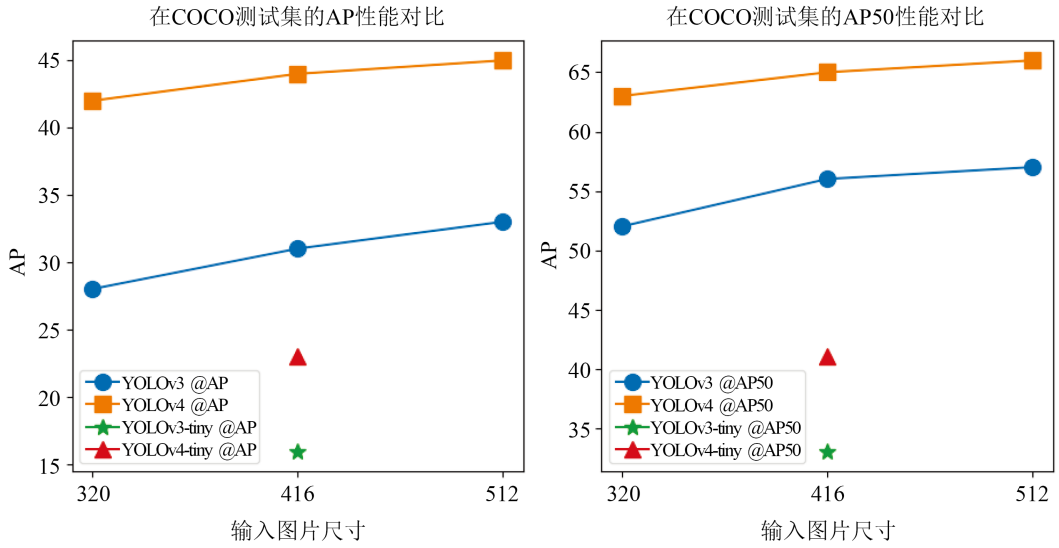

Figure 5. Performance comparison on COCO dataset
图 5. COCO 数据集上的性能对比

4.3. 自建数据集的性能对比与分析

在自建的司机违章动作数据集上，将数据集分为两个部分，80%的数据用于模型训练，20%的数据用于性能测试。4 个模型使用了相同的训练策略，批次大小为 16，使用 k-均值重新聚类的锚框，设置相同的数据增强，统一的优化器 Adam，训练共 100 个 epoch。性能对比如下表 1 所示。

Table 1. Performance comparison of driver violation datasets
表 1. 司机违章行为数据集性能对比

模型	YOLOv3-tiny	YOLOv3	YOLOv4-tiny	YOLOv4
mAP	48.29%	53.91%	63.72%	70.12%

结果分析：YOLOv4 达到了 70%，而 YOLOv4-tiny 在该数据集的性能比 YOLOv3 更好，高出近 10%，

这得益于知识蒸馏的策略方法使得 tiny 版本更好的学习了 YOLOv4 的泛化性能，实验表明 YOLOv4-tiny 相比于 YOLOv3 和 YOLOv3-tiny 在性能上有了巨大的提升。

问题：由于采集多种人脸角度、不同遮挡情况的目标图像，同一个类别之间的差异较大。对于手机这一类别，其颜色通常为黑色，在红外场景中，存在大量相似的黑色物体，其手机不同角度形状各异；对于水瓶，其种类繁多，其体积各异，颜色丰富；对于香烟，目标极小。众多的难样本导致所有的算法在测试集上精度并不高。

4.4. 多平台的速度对比与分析

实时性作为目标检测算法评估的重要指标，通常决定着是否能应用至实际场景中。特别在驾驶场景，其实时性的重要程度不言而喻。因此，在多个平台上测试 YOLOv4-tiny 的处理速度，使用 FPS 作为参考指标，如图 6 所示：

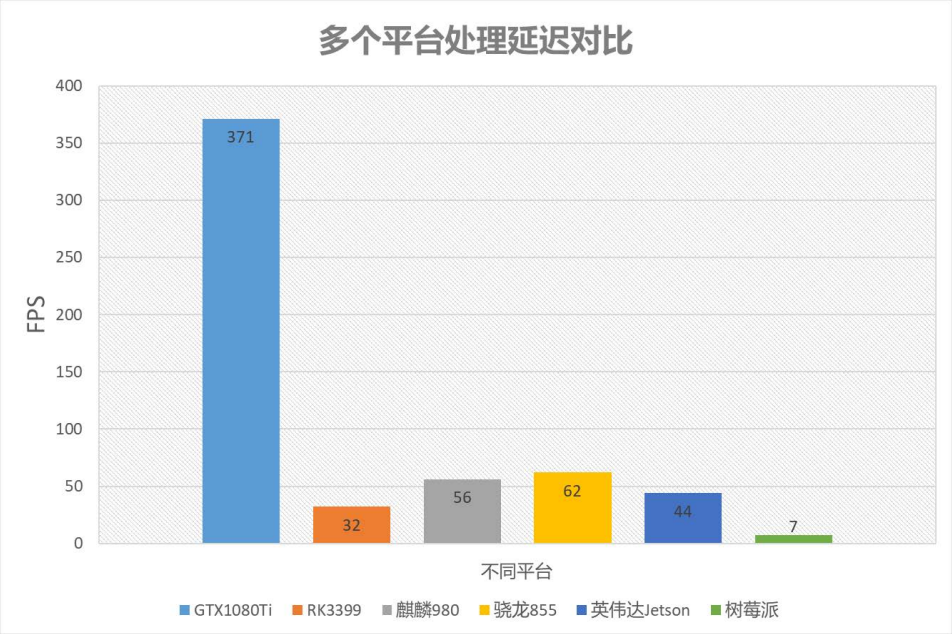

Figure 6. Multi-platform processing delay comparison
图 6. 多平台处理延迟对比

通过数据可知，YOLOv4-tiny 在服务端和边缘端都具备极快的推理速度，在移动手机上的两款常用芯片麒麟和骁龙系列测试结果显示具备良好的实时性，在 RK3399 上的处理速度也中规中矩，在硬件资源极差的树霉派上达到了 7FPS，这说明 YOLOv4-tiny 在车载终端具备优异的处理速度，使得检测算法在更多的实时场景中落地提供了方向。

5. 结语

本文基于目标检测的方法对司机行车过程中的违章动作进行识别与研究。在 YOLOv3 的基础上，改进其骨干网络，引入新的激活函数 Mish，将 CIOU 损失替换原有定位损失函数，为方便算法在边缘端的部署，采用更小的网络结构，并使用知识蒸馏的方法继承大模型的精度。实验表明，新的模型在精度和速度两个方面全面领先上一个版本，且在多个平台中都具备极佳的检测速度。该模型在司机这个特定场景中表现优异，使得算法为在各类追求实时性的场景中的落地提供了坚实的基础，具有巨大的工程意义。

参考文献

[1] Ren, S., He, K., Girshick, R., *et al.* (2017) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**, 1137-1149.

[2] Redmon, J., Divvala, S., Girshick, R., *et al.* (2016) You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 779-788. https://doi.org/10.1109/CVPR.2016.91

[3] Redmon, J. and Farhadi, A. (2017) YOLO9000: Better, Faster, Stronger. */Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, 21-26 July 2017, 7263-7271. https://doi.org/10.1109/CVPR.2017.690

[4] Redmon, J. and Farhadi, A. (2018) YOLOv3: An Incremental Improvement. arXiv:1804.02767 [cs.CV]

[5] Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M. (2020) YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv:2004.10934 [cs.CV]

[6] He, K., Zhang, X., Ren, S., *et al.* (2016) Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 770-778. https://doi.org/10.1109/CVPR.2016.90

[7] Wang, C.Y., Liao, H.Y.M., Wu, Y.H., *et al.* (2020) CSPNet: A New Backbone That Can Enhance Learning Capability of CNN. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Seattle, 14-19 June 2020, 390-391. https://doi.org/10.1109/CVPRW50498.2020.00203

[8] He, K., Zhang, X., Ren, S., *et al.* (2015) Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**, 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824

[9] Zheng, Z., Wang, P., Liu, W., *et al.* (2020) Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *Proceedings of the AAAI Conference on Artificial Intelligence*, **34**, AAAI-20 Technical Tracks 7. https://doi.org/10.1609/aaai.v34i07.6999

[10] Yun, S., Han, D., Oh, S.J., *et al.* (2019) CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. Proceedings of the IEEE International Conference on Computer Vision, Seoul, 27 October-2 November 2019, 6023-6032. https://doi.org/10.1109/ICCV.2019.00612

[11] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Li, F.-F. (2009) ImageNet: A Large-Scale Hierarchical Image Database. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Miami, 20-25 June 2009, 248-255. https://doi.org/10.1109/CVPR.2009.5206848