



Research on Dangerous Driving Behavior Detection Based on YOLOV5 Algorithm

He Jun^{*}, Zhong Kejia, Wu Shengke, Liu Pengzhen

College of Information and Engineering, Nanchang University, Nanchang, China

Email address:

1554066674@qq.com (He Jun), 1974816961@qq.com (Zhong Kejia)

^{*}Corresponding author

To cite this article:

He Jun, Zhong Kejia, Wu Shengke, Liu Pengzhen. Research on Dangerous Driving Behavior Detection Based on YOLOV5 Algorithm. *Science Discovery*. Vol. 10, No. 4, 2022, pp. 242-247. doi: 10.11648/j.sd.20221004.16

Received: July 19, 2022; **Accepted:** August 16, 2022; **Published:** August 17, 2022

Abstract: Aiming at the problems of high cost, difficult implementation and low efficiency of preventing traffic accidents, this paper proposes a deep learning object detection algorithm based on yolov5, which realizes the detection of dangerous driving behavior. Firstly, the dangerous driving data set is labeled and segmented, and re-clustered by K-means method. Secondly, the Non-maximum suppression in yolov5 algorithm is optimized into full category Non-maximum suppression. Finally, the data set is put into the model for iterative training, and the optimal weight is taken. In the test set results, the average accuracy of this algorithm is 97.8%, the average detection time of each image is 8.1ms, and the model size is 13.7mb. Under the same data and experimental environment, this paper compares with the most common dangerous driving behavior detection algorithms such as SSD, yolov3 and yolov4. The results show that this algorithm is better than several common dangerous driving detection algorithms in accuracy, detection speed and model size. In the actual driving process, the algorithm applied in this paper can better meet the requirements of real-time detection and judgment accuracy. Because the model is smaller, the deployment cost in car is lower, and meets the actual requirements.

Keywords: Dangerous Driving, YOLOV5, Behavior Detection

基于YOLOV5的危险驾驶行为检测方法

何俊^{*}, 钟可佳, 吴慎珂, 刘鹏政

南昌大学信息工程学院, 南昌, 中国

邮箱

1554066674@qq.com (何俊), 1974816961@qq.com (钟可佳)

摘要: 针对预防交通事故的发生成本高、执行难、效率低等问题, 本文提出一种基于YOLOV5的深度学习目标检测算法, 实现了对危险驾驶行为的检测。首先, 本文对危险驾驶数据集进行标注和分割, 通过K-means方法进行重新聚类, 其次, 将YOLOV5算法中的非极大值抑制优化为全类别非极大值抑制, 最后, 把数据集放入模型中进行迭代训练, 取最优权重。在测试集结果中, 本文算法的平均准确率为97.8%, 平均每张图片检测时间为8.1ms, 模型大小为13.7MB。在相同的数据和实验环境下, 本文和SSD、YOLOV3和YOLOV4等最几种常见危险驾驶行为检测算法进行了对比, 结果证明, 本文算法在准确度、检测速度和模型大小上均优于几种常见的危险驾驶检测算法, 能更好的满足驾驶过程中对准确性和实时性要求, 并且部署成本更低。

关键词: 危险驾驶, YOLOV5, 行为检测

1. 引言

随着我国经济的飞速发展,我国汽车保有量飞速增加,汽车成为我国国民出行不可缺少的一部分。出行次数的不断增加也带来了各种的交通问题,2018年全国交通事故总数就达到了24.49万例,带来了13.84亿元的经济损失[1],目前这个数字还在不断增加。据研究表明,驾驶人不当驾驶是造成交通事故的主要原因[2]。因此减少不当驾驶行为(开车玩手机、打电话、抽烟等危险驾驶行为)是减少交通事故发生最有效的手段之一。国内外学者针对驾驶人不当驾驶行为提出了一系列的基于计算机视觉的危险驾驶行为检测方法。

基于图像处理的驾驶人驾驶状态检测主要是指基于传统图像处理方法的驾驶人驾驶状态检测方法。Singh A等人[3]通过驾驶人员的面部几何特征定位眼睛的位置,并且利用二值化处理的方法获取眉毛和眼睛的二值化图,通过眉毛和眼睛位置关系判断是否睁眼,从而判断驾驶人是否疲劳。赵慧敏等人[4]利用Adaboost人脸检测算法进行人脸检测,根据人脸检测算法,检测人脸特征区域,最后通过PERCLOS原理检测疲劳状态。基于图像处理的驾驶人驾驶状态检测相比前两种检测方法实用性更高,只需要利用车内摄像头就可以捕捉驾驶人行为状态,但是在光照不足的情况下很难捕捉到关键特征点,在夜晚等光照弱的场景识别率不足。而且基于图像处理的驾驶人驾驶状态检测和基于深度学习的图像处理检测方法相比,更难提取特征,检测准确率也低于深度学习的检测方法。

基于深度学习的危险检测方法分为两类,一种是用深度卷积神经网络提取驾驶员图像空间特征,对驾驶员行为进行分类。王坤等人[5]提出了基于SSD改进算法的分心驾驶检测行为,对驾驶人员的关键行为动作进行检测,从而得到驾驶人员的驾驶状态。第二类是基于深度学习的行为跟踪检测算法,Xiaoxi Ma等人[6]利用Kinect构建了疲劳驾驶的数据集,融合了深度图像帧的空间特征和时间矢量表示的时间特征,实现了驾驶人员的疲劳检测模型。两年后,Xiaoxi Ma等人[7]又提出基于红外的视频数据的疲劳检测算法,构建了完整的驾驶员疲劳行为检测模型。基于深度学习的驾驶人危险驾驶行为检测方法解决了传统图像处理检测方法的弱项,不需要花时间选择有效面部特征,精准度也上升了很多。但是目前也存在一些问题,首先是一般的深度学习检测驾驶人危险驾驶行为的算法都是深度神经网络,大参数的神经网络提取图像特征在保证精准度的同时,也带来了检测速度很难满足实时性的问题。其次,一般深度学习的模型的大小都达到数百兆,没有考虑实验平台是在汽车上处理器性能较弱的问题。

基于上述问题,本文提出了一种基于YOLOV5算法模型为基础的深度学习目标检测算法来对驾驶人危险驾驶行为进行检测。YOLOV5模型对比其他的深度学习目标检测算法有更简单的网络结构,网络层次也更低,是一种网络深度更浅的深度学习算法。

本文也在使用YOLOV5算法模型的同时也应用了K-means先验框和全局非极大值抑制等方法提升了模型的精度。最后在相同的实验环境下,对比之前普遍使用的

SSD[8]目标检测算法和YOLOV3[9, 10]、YOLOV4[11]深度学习目标检测算法,本文使用的驾驶人危险驾驶行为检测算法在平均精度均值(mAP)和检测速度上都得到了极大的提升,其中平均精度均值对比YOLOV3深度学习算法提升了5.1%,对比SSD深度学习算法提升了14%,检测速度上对比YOLOV3的32FPS和SSD的43FPS,本文算法达到了122FPS,并且模型的大小缩小到了13.7MB,能更加好的应用在低性能的处理器的上,且能够保证实时性,可以在第一时间对驾驶人驾驶状态做出正确的判断。

2. 基于YOLOV5的检测模型

YOLOV5[12]总共有四种模型,分别是YOLOV5s, YOLOV5m, YOLOV5l和YOLOV5x四个模型。四个模型的主要框架都差不多,主要是网络深度各有区别,应为驾驶人危险检测算法注重实时性和轻量性,所以本文选择网络深度最浅检测速度最快的YOLOV5s模型。

YOLOV5算法是经典的一阶段检测算法,分为输入端、Backbone、Neck和Prediction四个部分。YOLOV5s具体的网络结构如图1所示。

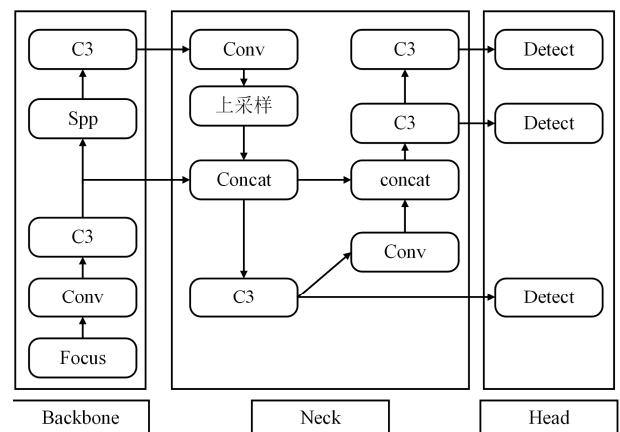


图1 YOLOV5模型框架。

2.1. 输入端

YOLOV5的输入端用了Mosaic数据增强算法,Mosaic算法是参考2019年提出的CutMix[13]算法,Mosaic使用了四张照片进行了随机裁切、缩放和旋转。这样增加了数据集的数量和多样性,而且通过缩放拼接,增加了小目标和多目标的数据样本,极大增加了算法的鲁棒性。而且由于同时读取四张图片,这样等于变相增加了训练的batch数目,减少了训练时间,在GPU不是很强的情况下也能有一个比较好的训练结果。

2.2. Backbone端

YOLOV5的Backbone结构是YOLOV5算法的骨干部分,是由Focus、Conv、C3、Spp等多个模块组成的。图片经过输入模块处理后进入Focus模块,Focus模块对图像进行切片操作,图片的通道数会增加。如:原始图片为608*608*3的三通道图片经过Focus模块会变成

304*304*12 的图片。经过切片会对输出结果进行一次32个卷积核的卷积操作变成304*304*32的图片。经过Focus操作后的图片参数扩大了4倍,减少了图片信息的丢失。Conv是YOLOV5中最基本的卷积模块,它由二维卷积层、正则层、激活层三个部分组成,对输入图像进行卷积综合处理。C3模块是一种CSP结构,他是由多个Bottleneck模块组成,是一种残差模块,他能增加网络深度,避免梯度丢失等问题。在Bottleneck模块中,数据经过两次综合卷积操作后,和原始输入数据进行叠加,最后输出结果。Spp是空间金字塔池化层[14],他可以对三种不同尺寸的图像进行池化操作最后组合在一起。

2.3. Neck端

Neck端,位于Backbone端和输出端中间,在目标检测中是为了提高模型的特征提取能力,相当于YOLOV5检测模型的颈部。Neck模块采用FPN (Feature Pyramid Networks) +PAN (Path Aggregation Networks) 的结构。这种自上而下的信息传递结构将高层的特征信息通过上采样的方式进行着传递。这样的传递方式可以使高级特征和低级特征实现互补,FPN结构将上层的目标类别向下层传播,PAN将下方的大目标的坐标特征和小目标类别特征向上传播。两者结合在传播过程中取长补短,提升模型的特征提取能力。

2.4. Head端

Head端为YOLOV5的预测端,图片经过Backbone端和Neck端的处理后分为大、中、小三种图片类型进行结果预测,克服了传统CNN网络小目标对象经过多次图像处理,最后丢失的普遍难题,进一步提高了小目标的检测的准确度。

Head端使用Ciou_Loss做Bounding_box损失函数[15],具体公式如下。

$$CIOU_Loss = 1 - CIOU \quad (1)$$

$$CIOU = IOU - \frac{Distance^2}{Distance_C^2} - \frac{v^2}{(1 - IOU) + v} \quad (2)$$

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w^p}{h^p} \right)^2 \quad (3)$$

其中 $Distance$ 为预测框的中心点和真实框的中心的欧式距离, $Distance_C$ 为 C 的对角线距离。 v 是长宽比影响因子, w 为宽, h 为高, gt 为真实框, p 为预测框, IOU 是交并比, IOU_Loss 指预测框和真实物体框的交集的面积与并集面积的比值。 IOU 是使用的最广泛的检测框loss。但是当两个物体不相交时, IOU 无法返回梯度,不能反应真实框和预测框的距离。 $CIOU_Loss$ 考虑了中心点距离的信息,就算不相交也可以表示真实框和预测框的距离,不会造成梯度消失,而且 $CIOU_Loss$ 还考虑了边界宽高比的尺度信息,可以进一步提高预测框的精度和速度。

当目标检测的过程中,经常会出现多个预测目标同时出现的情况,而且有可能两个检测目标会重叠,导致检测

目标信息的丢失,NMS非极大值抑制可以改善这种情况,提高多目标检测的准确度。本文由于不会出现多检测目标重叠的情况,所以使用了全局非极大值抑制,对全类别都进行非极大值抑制。这样可以过滤更多置信度较低检测结果。

Head端有大中小三种不同尺度的初始锚框,初始锚框的设定源自COCO[16]数据集,COCO数据集是非常大且多样的数据集,默认的三种锚框大小也源自于COCO的数据,虽然具有普遍性,但是不具有强的针对性。本文根据K-means聚类方法重新计算了初始锚框,使锚框更具有针对性。本文由于针对的对象是人的一般危险驾驶行为,所以数据没有COCO数据集丰富,采用针对性的初始锚框可以提高算法的准确性。下面为实验的初始锚框大小。[10, 13, 16, 30, 33, 23]用于8倍采样,小目标检测,[30, 61, 62, 45, 59, 119]用于16倍采样,中目标检测,[116, 90, 156, 198, 373, 326],用于32倍采样,大目标检测。

3. 数据预处理

3.1. 数据采集

本文的数据集一部分来源于公开数据集,如开车玩手机、开车打电话两类危险驾驶行为数据来源于Kaggle开源数据集。另一部分数据集来源于网络搜集,如开车抽烟数据来自于网上自己收集,三类图片一共收集了8684张照片。以下分别为玩手机、打电话、吸烟的部分样本,如图2所示。



图2 部分样本。

在深度学习的学习过程中,添加背景图片通常也可以丰富数据集,增加模型可靠性,降低误识别的概率。一般的情况下,数据集中应该有(1%-10%)的背景图片,本文收集了500张司机正常驾驶的背景图片,约占总共数据集的5.4%。在yolov5模型中,背景图片不需要标签。

3.2. 数据标注

数据集的标注是图像目标检测不可或缺的步骤,只有将检测部分标出,计算机才能对目标区域的信息进行学习,从而根据已知的数据集,不断优化目标检测算法。本文所有数据都是使用Labelimg标注工具自己标注,标注的类别总共分为三类。标签名为Classes,用0, 1, 2分别表示玩手机、打电话、抽烟三种类,标签文件内容为类别标号,加上归一化后的预设框坐标,其中X, Y, W, H, 表示为自身图片尺寸的归一化值,其中X, Y表示为标注框中心点的坐标,W, H代表标注框的宽度和高度。这五个数值通过空格隔开并储存在一个txt格式的文本文件中,这五个数的值可以定位出一个唯一类别的矩形框。其具体格式如表1所示。

表1 标签样式。

Classes	X	Y	W	H
0	0.626562	0.488542	0.153125	0.235417
1	0.29375	0.315625	0.36875	0.435417
2	0.32698	0.590476	0.130158	0.123809

在标注完所有数据后，本文将数据按8:1:1的比例进行了随机分割。训练集、验证集、测试集分别为7033张、782张和869张。

3.3. 实验平台

本机的实验平台是基于Windows 10的操作系统，使用python编程语言，在Pytorch架构中实现的。具体实验环境如下表2所示。

表2 实验环境。

名称	配置
操作平台	Windows 10
CPU版本	AMD 锐龙5 3600X
GPU版本	NVIDIA GeForce GTX 1070
GPU加速	CUDA 10.2
编程语言	Python 3.8
实验平台	Pytorch 1.8.0

3.4. 模型评估指标

本文选用，*CIOW_Loss*损失值、精准率（P）、召回率(R)、*mAP@0.5*和每秒模型检测的图片数量（FPS）来衡量模型的好坏。

准确率就是查准率，即成功预测为正和全部预测为正的比值。查全率越高说明，错误检测的比例反而越低。召回率就是查全率，即正确预测为正和全部实际为正的比值。召回率高说明，漏检的比例反而越低。具体的计算公式如下：

$$Precision = \frac{TP}{TP + FP}$$
 (4)

$$Recall = \frac{TP}{TP + FN}$$
 (5)

其中，*TP*、*TN*、*FP*、*FN*分别指正类数、负类数、正样本识别成负类数、负样本识别成正类数。*mAP*指平均精度均值，是评估目标检测结果的一个重要指标。它能较综合的反应一个目标检测算法的好坏。*mAP@0.5*是指在*IOU*阈值为0.5时的*AP*值然后做平均。

$$mAP = \frac{\sum AP}{N}$$
 (6)

其中，*AP*指Precision-Recall曲线下围成的面积，*N*表示所有检测类别的总和。

4. 实验结果

4.1. YOLOV5训练过程

网络模型训练阶段，本文设置了150个迭代周期，迭代批次的大小为16，图片输入大小为640，动量可以加快

模型的训练过程，本文设置为0.937，学习率为0.01，置信度为0.5，非极大抑制值为0.35，如表3所示。

表3 初始训练参数。

参数	初始数据
迭代次数	200
批次大小	16
动量大小	0.937
学习率	0.01
置信度	0.5
全局非极大抑制阈值	0.3

由于本文数据集属于小数据集，在模型的初始权重上本文并没有从头开始训练，而是选择了在COCO数据集上取得较好效果的默认权重，这样做能加强模型的特征提取能力，加快训练速度。YOLOV5模型的训练损失值和平均精度度如图3所示。从图3中可以看出，由于使用的是COCO数据集的初始权重，模型的收敛速度很快。从开始迭代到曲线基本收敛的过程中，Loss损失值是一直减少的，在初始的30次数迭代下，损失值下降到了0.04，下降速度很快，在此期间平均准确度也开始激增，增加到0.85左右，在接下来30到150次迭代下，增加速度相对平缓，损失值下降到0.02往上，平均准确度达到0.95左右，在接下来的50次迭代下，损失值稳定到0.02，平均准确度达到0.97，已经没有继续下降的趋势。综上所述，可以认为模型训练过程是收敛可靠的，能达到训练预期。

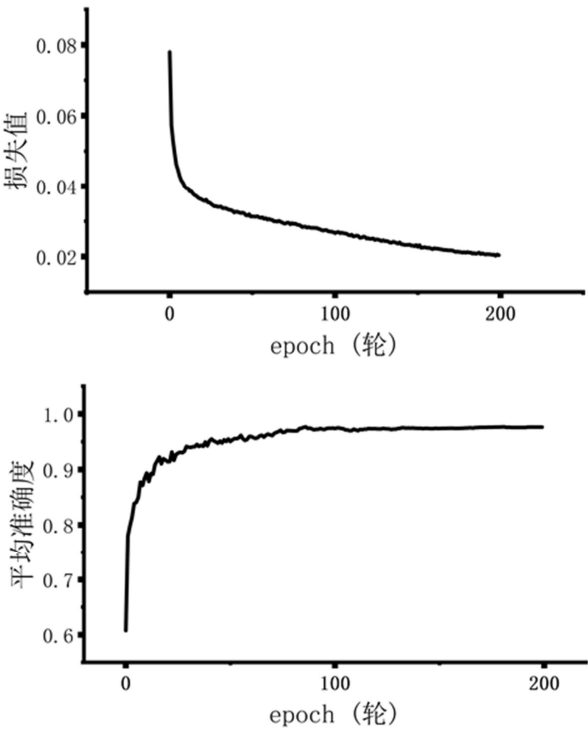


图3 训练结果。

4.2. 模型的验证

测试集总共占有数据集的10%，总共为868张图片，其中开车玩手机的194张，开车打电话的198张，开车吸烟的为476张。

将测试集的数据放入训练好的模型中，其结果如表4所示。

表4 测试集检测结果。

Class	Images	Labels	P	R	mAP
All	868	868	0.976	0.972	0.974
玩手机	868	194	0.998	0.999	0.995
打电话	868	198	0.993	0.999	0.92
吸烟	868	476	0.935	0.916	0.932

通过测试集检测结果可知，全类别的平均准确度为97.4%，准确率为97.6%，召回率为97.2%，识别开车打电话的平均准确度为99.5%，准确率为99.8%，召回率为99.9%，识别开车玩手机的平均准确度为92%，准确率为99.3%，召回率为99.9%，识别开车吸烟的平均准确度为93.2%，准确率为93.5%，召回率为93.2%。

由此可知，该算法对开车玩手机和开车打电话的识别效果最佳，开车抽烟检测相对来说较低，这是由于香烟数据集中含有大量小目标检测，香烟目标较小，在有些像素较低的情况下会出现误检测和漏检测的情况，但是平均准确度也达到了93.5%也符合实际要求。在检测速度上该模型检测速度可达到122FPS，即每秒钟检测122张图片。而且模型的大小仅仅为13.7MB。可见模型训练结果是令人满意的，也可以很好的满足现实需求。如图4是该算法部分检测结果。

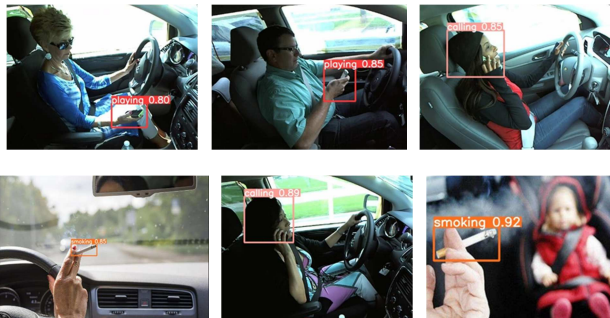


图4 实验检测结果。

4.3. 同类算法的比较

为了进一步对比和验证本文算法的性能和可靠性，本文对比了该领域内的主流的目标检测算法，其中包括SSD、YOLOV3、YOLOV4算法，在保证数据集和实验设备一致的条件各个算法的实际表现如表5所示。

表5 对比实验。

算法	mAP	模型大小(MB)	FPS
SSD	0.838	91.6	43
YOLOV3	0.927	123.5	33
YOLOV4	0.95	200.6	65
YOLOV5s	0.978	13.7	122

由表5的实验结果可知，YOLOV5的帧数为122FPS，是SSD算法的三倍，对比YOLOV3和YOLOV4算法，分别快了89FPS和57FPS，模型大小仅为13.7MB，约为YOLOV4算法的1/15，模型算法的平均准确度是97.8%，是所有算

法中准确率最高的，比较于SSD、YOLOV3、YOLOV4算法精度有较大提升，分别提升了14%、5.1%和3%。

由上述可知，对比主流的危险驾驶危险检测算法，本文算法的检测速度、模型大小、和检测精准度好过其他几种主流算法。在对危险驾驶检测时中，本文算法模型可以实现更快的检测速度，且更小的模型大小可以实现更高的精准度。

5. 结论

在实际的驾驶人危险驾驶行为检测上，由于要求检测的实时性，检测速度这个指标就显得十分重要，一个实时的检测系统才能带来更好、更安全驾驶体验。本文通过和其他主流算法进行对比，结果表明，YOLOV5算法的检测速度达到了122FPS，可以很好的满足实时性这样要求。又因为该算法要移植在车上，所以首先就要保证其经济性，本文算法体积只有13.7MB，这在实际应用过程中非常具有优势，在保证平均准确度高于主流算法的情况下，一个轻量化的检测算法是减少成本和增加检测速度的必要条件。

综上所述，YOLOV5算法在利用视频精准捕获驾驶人的危险驾驶行为，是一种低成本且高效率的有效方式，这为减少交通事故的发生提供了新思路，具有现实意义。在下一步的计划中，将计划收集更多的危险驾驶行为的图片扩展数据集，保证算法模型有更强的泛化能力。

致谢

本文为国家自然科学基金项目《基于深度学习与多模态融合方法的情感识别技术研究》(62066025)的阶段性成果之一。

参考文献

- [1] 中华人民共和国国家统计局. 中国统计年鉴[M]. 北京: 中国统计出版社, 2019.
- [2] 孔令铮. 交通事故致因中的人为因素分析[J]. 中国安全科学学报, 2013, 23 (1): 28-34.
- [3] Singh A., Kaur J.. Driver fatigue detection using machine vision approach [C] // IEEE International Advance Computing Conference, 2013.
- [4] 赵敏慧. 基于图像的驾驶员姿态识别方法研究[D]. 东南大学, 2018.
- [5] 王坤. 基于深度学习的驾驶员分心和疲劳驾驶行为监测方法[D]. 武汉理工大学, 2020.
- [6] Xiaoxi Ma, Lap-Pui Chau, Kim-Hui Yap. Depth video-based two-stream convolutional neural networks for driver fatigue detection [C] // International Conference on Orange Technologies, 2017.

- [7] Xiaoxi Ma, Lap-Pui Chau, Kim-Hui Yap, Guiju Ping. Convolutional Three-Stream Network Fusion for Driver Fatigue Detection from Infrared Videos [C] // IEEE International Symposium on Circuits and Systems, 2019.
- [8] Cao G, Xie X, Yang W, et al. Feature-fused SSD: Fast detection for small objects [C]. Ninth International Conference on Graphic and Image Processing (ICGIP 2017). International Society for Optics and Photonics, 2018, 10615: 106151E.
- [9] Hurtik P, Molek V, Vlasanek P (2020) YOLO-ASC: you only look once and see contours, accepted. In: Proceedings of IEEE-WCCI conference.
- [10] Nan Xiang, Zhao, Cao, Yuedong, Wang, Qianqian Jia. A Real-Time Vehicle TrafficLight Detection Algorithm Based on Modified YOLOv3 [C] // International Conference on Electronics Technology. 2021.
- [11] 李昭慧, 张玮良. 基于改进YOLOv4算法的疲劳驾驶检测[J]. 电子测量技术, 2021, 44 (13): 73-78.
- [12] 张上, 王恒涛, 冉秀康. 基于YOLOv5的轻量化交通标志检测方法[J]. 电子测量技术, 2022, 45 (08).
- [13] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun Seong Joon Oh, Youngjoon Yoo, Junsuk Choe. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features [C] // International Conference on Computer Vision, 2019.
- [14] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition [C] // 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.
- [15] ZHENG Z H, WANG P, LIU W, et al. Distance-IoU loss: faster and better learning for bounding box regression [J]. arXiv: 1911.08287, 2019.
- [16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context [C] // European conference on computer vision, 2014.