

基于改进 SSD 的道路交通标志检测

黄 桥¹, 胡绍林², 张彩霞¹

(1. 佛山科学技术学院 机电工程与自动化学院, 广东 佛山 528000;
2. 广东石油化工学院 自动化学院, 广东 茂名 525000;

摘要: 针对复杂环境下交通标志检测精度低的问题, 设计了一种检测精度更高的目标检测算法, 对 SSD 深度学习目标检测算法进行了优化改进; 将深度特征表征能力较强的 Resnet50 网络模型融入于 SSD 算法中; 采用 K-means++ 聚类算法确定 SSD 中先验框的尺寸, 提高交通标志的检测率; 分别利用 SSD 模型和改进的 SSD 模型做检测对比实验, 结果表明, 改进算法对各类交通标志的检测精度比原 SSD 算法更高; 改进的 SSD 方法对交通标志进行检测能取得较好效果, 弥补了原算法的不足。

关键词: 交通标志; 无人驾驶; SSD 算法; K-means++ 聚类

Road Traffic Sign Detection Based on Improved SSD

HUANG Qiao¹, HU Shaolin², ZHANG Caixia¹

(1. School of Mechatronic Engineering and Automation, Foshan University, Foshan 528000, China;
2. School of Automation, Guangdong University of Petrochemical Technology, Maoming 525000, China)

Abstract: Aiming at the problem of low detection accuracy of traffic signs in complex environments, a target detection algorithm with higher detection accuracy is designed, and the SSD deep learning target detection algorithm is optimized and improved. Incorporate the Resnet50 network model, which has strong deep feature characterization capabilities, into the SSD algorithm. The K-means++ clustering algorithm is used to determine the size of a priori box in the SSD to improve the detection rate of traffic signs. The SSD model and the improved SSD model were used to do detection comparison experiments. The results show that the improved algorithm has higher detection accuracy for various types of traffic signs than the original SSD algorithm. The improved SSD method can achieve better results in the detection of traffic signs, which makes up for the deficiencies of the original algorithm.

Keywords: traffic sign; driverless; SSD algorithm; K-means++ clustering

0 引言

交通标志是自动驾驶和导航等领域的重要研究对象, 其中交通标志检测是目标检测领域一个重要的方向, 在无人驾驶和自主导航等领域有重要应用。应用最广的交通标志检测方法是基于视觉的方法。

交通标志种类繁多, 我国常见的交通标志主要分为提示、禁止、警告等 3 种类型, 本文以这 3 类交通标志为研究对象, 进行自动检测。由于这 3 种类型的交通标志外观上有一定的差异, 指示标志和禁令标志虽然大多都是圆形, 但颜色有所不同; 警告标志外观和颜色二者都不同。因此, 可以根据这些共同特征, 进行简单的分类。

传统的目标检测算法, 通过形态学变换来提取交通标志边缘、RGB 空间转 HSV 空间, 根据颜色特征提取交通标志位置以及人工预设感兴趣区域等方法, 定位交通标志目标^[1]。例如, 基于 Cascade 分类器的目标检测通过快速遍历目标滑动窗口, 达到提高检测效率的目的。但是, 在实际

场景中, 采集到的图像通常存在噪声多、光线强弱和目标不明确等问题^[2], 利用机器学习目标检测方法会对图像中目标的定位有很大影响。

传统的目标检测算法通常是先进行候选框的提取, 候选框的提取通常采用滑动窗口的方法进行, 然后是对每个窗口中的局部信息进行特征提取, 最后对候选区域提取出的特征进行分类判定^[3]。其中分类器包括 Adaboost^[4]、SVM^[5]、Decision Tree^[6]和 DPM^[7]特征等。对候选区域提取的特征进行训练分类操作需要人为设计, 人工设计特征描述的方法需要巨大的工作量, 比如 DPM 算法用来检测人的激励模板不能拿去检测小猫或者小狗, 所以在每做一种物件的探测的时候, 都需要人工来设计激励模板, 为了获得比较好的探测效果, 需要花大量时间去做一些设计。

综合上述分析, 传统的目标检测方法更加适用于有明显特征, 背景简单等理想的目标, 且由于该方法是基于滑动窗口的区域选择策略, 因此传统的目标检测方法针对性不足、算法耗时长、算法计算过程产生的冗余窗口太多、

收稿日期: 2021-03-05; 修回日期: 2021-04-08。

基金项目: 国家自然科学基金(61973094); 广东省基础与应用基础研究基金粤港澳应用数学中心项目(2020B151531003)。

作者简介: 黄 桥(1995-), 男, 四川广安人, 硕士研究生, 主要从事深度学习、目标检测方向的研究。

胡绍林(1964-), 男, 陕西西安人, 教授, 主要从事过程监控、系统仿真、大数据与人工智能方向的研究。

引用格式: 黄 桥, 胡绍林, 张彩霞, 等. 基于改进 SSD 的道路交通标志检测[J]. 计算机测量与控制, 2021, 29(10): 60-65.

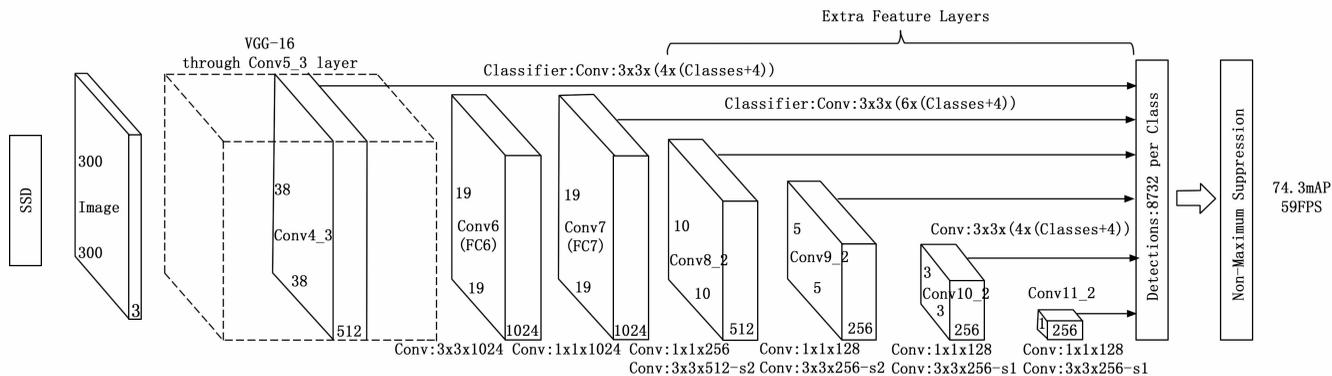


图 1 SSD 网络结构

难以普及和对多样性变化的鲁棒性弱^[5]等缺点。

在目标检测领域中,当前应用较广的交通标志检测是基于深度学习的方法,该方法的优点是获取的信息量大、成本低、容易普及^[8]。相比传统的目标检测方法,深度学习目标检测算法适合对复杂多变下的抽象特征目标的检测^[9]。目前,主流的深度学习目标检测算法主要分为两类:一类是 two stage 的方法,即先通过算法产生候选区域(region proposals),然后通过卷积神经网络对候选区域分类;另一类是 one stage 的方法^[10],该方法不需要先产生候选区域再分类,它只需要一步就可以预测出目标的边界框,从而直接给出检测结果^[11]。

1 SSD 检测算法

1.1 SSD 特征网络

SSD (single shot multibox detector) 是 Wei Liu^[12] 在 ECCV2016 上提出的一种基于卷积神经网络的目标检测算法。SSD 算法相比 Faster RCNN 有明显的速度优势,相比 YOLO 又有明显的 mAP (mean average precision) 优势^[13]。SSD 作为 one-stage 类目标检测算法之一,它只用一个全卷积网络就完成了目标分类和定位任务^[14]。其结构如图 1 所示。从图 1 中可以看出,SSD 使用 6 个不同特征图检测不同尺度的目标。低层预测小目标,高层预测大目标。

目标检测算法中常见的特征提取网络有 VGG16、DarkNet、MobileNet 和 ResNet。SSD 采用了 VGG16 中的前 5 层网络,并把 VGG16 网络结构的两个全连接层 FC6 和 FC7 分别改成 3×3 和 1×1 的两个卷积层,这增加了新的卷积层来获取更多的特征图^[15]。输入图像经过特征提取网络进行特征提取,并在各个卷积层中进行卷积和下采样等处理。最后通过非极大值抑制 (NMS) 得到算法的检测结果。SSD 算法将输出设为 300×300 和 512×512 两种,用于不同尺寸图像的输入。本文讨论 300×300 为输入的情况。其中输入图像大小的规格应是长宽比例应为 $1:1$,若使用其他尺寸的图像训练模型,会引起图片畸变,模型自动缩放,影响精度。

1.2 SSD 的 Anchor 机制

SSD 结合多尺度特征图共同检测,其中大尺度特征图

用来检测小目标,小尺度特征图用来检测大目标,尽量能够保证大小不一的目标都能够被检测到。因此 SSD 检测算法的结果理论上能达到较好的效果,主要有两个原因:第一是多尺度;第二是设置多种宽高比的锚框(anchor),即在特征图的每个像素点处,生成不同宽高比的 anchor,根据具体情况设置宽高比^[16]。假设每个像素点有 k 个 anchor,需要对每个 anchor 进行分类和回归,其中用于分类的卷积核个数为 ck (c 表示类别数),回归的卷积核个数为 $4k$ 。设置 anchor 机制的原因是每一层实际响应的区域其实是有效感受野区域,但在训练过程中并不知道有效感受野的大小,为了解决这个问题,提出了 anchor 技术。

2 改进的 SSD 目标检测模型

SSD 目标检测算法中,虽然理论上它的检测精度会随着网络的深度增加而提高,但在反向传播时梯度爆炸会导致简单堆叠的卷积层不能很好地训练网络^[17]。换言之,随着卷积神经网络层数的增加,训练集上的目标检测的准确率反而下降。交通标志复杂多变,存在不同环境下的背景,且采集的图像距离交通标志远近不一,当较远处的目标检测时,SSD 算法并不一定能够检测出来。

2.1 特征提取网络的改进

为了提高对输入图像的特征提取能力,本文将残差网络引入到 SSD 模型中,将 VGG16 部分替换成 ResNet50,构建 SSD-ResNet 网络结构,用 50 层网络拟合一个残差映射可以解决深层网络精度退化问题。用多个累积的非线性层拟合一个映射 $F(X)$, $H(X)$ 是目标最优解的映射,关系如式 (2) 所示:

$$F(X) := H(X) + X \quad (2)$$

此时,最优解映射 $H(X)$ 与 $F(X) + X$ 是等价的,图 2 所示为一个残差学习模块,根据该方式能将快捷连接实现。

图 2 中的 X 代表模块的输入向量, Y 代表模块的输出向量, W_i 代表权重层参数,当输入和输出维度一致时,需要增加一个线性投影 W_s 来匹配维度^[18]。

残差学习模块往往需要两层以上,在实际计算中,为

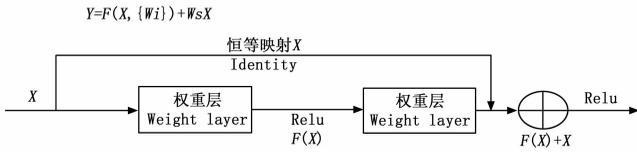


图 2 残差学习模块

了考虑计算成本,对残差学习模块做了计算优化,即将残差学习模块中的两个权重层替换为 3 个不同的权重层。这两种不同的残差学习模块如图 3 所示。ResNet 网络采用图 3 中这种跳跃结构来作为基本的网络结构,该跳跃结构又被称作 Bottleneck,其中,图 3 (a) 是使用于小网络的残差模块,图 3 (b) 结构的 1×1 卷积用于降维,该形式的跳跃结构是适用于大网络的残差模块,本文采用的 ResNet50 网络属于大网络。

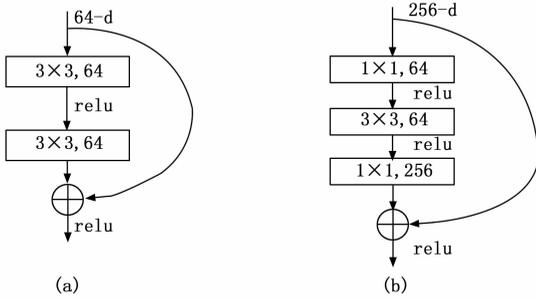


图 3 两种残差学习模块

ResNet50 总共有 16 个 Bottleneck,每个 Bottleneck 包含 3 个卷积层,最后加上输入层的卷积层和最后的全连接层构成了 50 层的残差网络。添加深度残差网络的目的,是克服传统 CNN 网络随着层数加深到一定程度之后越深的网络反而效果更差的缺陷,避免因过深的网络引起分类准确率下降。如图 4 为残差网络替换 VGG16 后的网络结构图。

以 ResNet50 作为提取网络,删除掉 ResNet50 后面的全连接层,再增添几层额外的卷积层提取特征,得到不同尺度的特征图,然后我们让这些不同层次的特征图分别预测不同大小的目标,浅层卷积层提取到的是比较细小的特征,越深层的卷积提取到的信息会越丰富,因此我们让浅层的卷积特征图去检测小的目标,让深层的卷积特征图去检测大的目标。该网络总共有 65 层,包括 59 个卷积层和 5

个池化层,每个经过池化层后输出的特征层,它的大小都会减小为上层输入大小的一半,最后的特征层输出为 1×1 。选择模型中 Conv4_6、Conv5_3、Conv6_2、Conv7_2、Conv8_2、Conv9_2、Conv10_2,将这 7 种不同尺度的特征层用于预测交通标志,输出交通标志类别置信度和检测框偏移量,本文使用 Jaccard Overlap 策略匹配目标框,其中 Jaccard Overlap 的阈值设置为 0.5,最后使用非极大值抑制^[19]去除多余的预测框。

2.2 先验框的设计

原 SSD 算法对每一张特征图,按照不同大小和长宽比生成多个默认框,是基于多尺度的方法得到目标检测结果。对于不同尺度特征图,就会设置不同大小和宽高比的先验框,其计算见式 (3):

$$S_k = S_{\min} + \frac{S_{\max} - S_{\min}}{m - 1}(k - 1) \quad (3)$$

其中: k 的取值范围为 $[1, m]$, m 为特征层数, S_{\min} 和 S_{\max} 分别代表最小和最大特征层尺度,对应的默认值一般为 0.2 和 0.9。中间特征层尺度均匀分布。

SSD 算法默认的 anchor 的宽高比设定为: $a_r \in \{1, 2, 3, 1/2, 1/3\}$ 。先验框的宽、高见式 (4) 和 (5):

$$w = S_k \times \sqrt{a_r} \quad (4)$$

$$h = \frac{S_k}{\sqrt{a_r}} \quad (5)$$

先验框生成之后,把不同尺度的特征输入到预测网络,然后进行预测计算,最后将输出的预测结果分为两部分。一部分是预测框的修正值,另一部分就是框内目标的概率。

在实际训练中, S_{\min} 和 S_{\max} 的值往往需要多次尝试,本质上是盲目搜索的,这样会导致检测效率变差。为了更符合交通标志数据的特点,本文使用 K-means++ 方法对目标框统计数据进行聚类,代替原 SSD 中生成的默认框基准大小 S_k 。

K-means++ 算法是随机从当前数据中选取一个样本作为初始聚类中心,然后计算每个样本与当前已有聚类中心之间的最短距离。本文先获取训练集样本,然后按照轮盘法选择总共 K 个聚类中心,针对目标框中每个样本,计算该样本到 K 个聚类中心的距离并将其分到距离最小的聚类中心对对应的类中,最后针对每个类别重新计算它的聚类

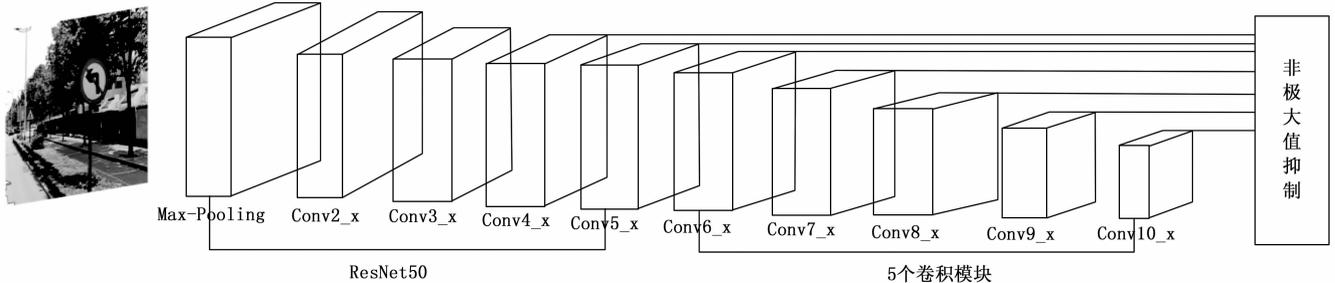


图 4 SSD-ResNet50 网络模型

中心直到位置不再改变。本文采用曼哈顿距离^[20]来衡量相似度, 依次选出它能更快地找到合理的先验框长宽比, 对先验框重新设计, 提高检测精度。

先验框的设置从尺度和长宽比等方面来考虑。先验框的尺寸有一个线性增加规律: 随着特征图尺寸的减小, 先验框尺度线性增加。图 5 为交通标志数据集的真实宽和高的聚类, 交通标志的宽、高聚类中心大多聚集在 $[62, 86]$, $[108, 150]$, $[174, 243]$ 附近, 宽高比大多在 0.72 左右, 因此本文将特征图上每个滑动窗口的尺度生成宽高比为 18 : 25, 在各个卷积层使用该比例大小的尺寸作为先验框的固定尺度, 第一个先验框单独设置, 为第二个先验框尺度的一半。另外, 对先验框重新设计后产生数量比原来的先验框数量更少, 提高了检测精度。

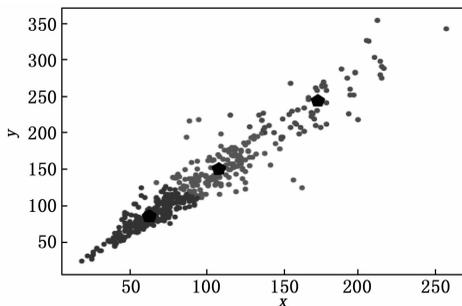


图 5 真实框宽高聚类结果图

3 模型训练及结果对比分析

本文的交通标志数据集来源于现有的中国交通标志检测数据集 (CCTSDB), 图像的分辨率为 600×600 。用 labelImg 工具对其进行标注, 生成 PASCAL VOC 数据格式。根据实际需求划分训练集、验证集和测试集, 将数据集的 60% 作为训练集, 剩下的数据集分为测试集和验证集, 且两者的比例相同, 因此测试集和验证集是相互独立的, 这样可以确保后面的评价标准是可信的。利用验证集评估模型, 找到训练过程中模型的最佳参数, 然后采用该参数在测试集上进行测试。

为了提高关于目标大小和形状的鲁棒性, SSD 采用了数据增强的随机采样策略。数据增强是为了在不降低检测精度的情况下提高算法的鲁棒性, 不同数据集用到的数据增强的方法不同, 特定场景和实际应用中, 一般很难收集满足各种条件的数据用于训练和测试, 因此许多目标检测算法都采用数据增强的方式来评估所设计的算法泛化能力, 常用的数据增强方法包括水平翻转、平移、裁剪和颜色抖动等。本文对采集的自然环境下的图像数据进行数据增强操作以验证模型的泛化能力。

3.1 实验环境

本文使用 PyCharm 软件训练网络和测试模型, 配置运行环境, 计算机的配置为: Windows10 64 位操作系统, 处理器 (CPU) 型号为 Intel i7, 内存 (RAM) 为 8 GB, 显卡 (GPU) 是 NVIDIA GeForce GTX。开发环境的软件配置及

版本: Anaconda3、CUDA10.0、cuDNN7.6.3、Python3.7.6、TensorFlow-GPU1.14.0。

3.2 模型训练

为验证本文方法的有效性, 训练改进前后的算法, 并对比其模型评价指标结果。准备好交通标志数据集, 为了提高模型的性能, 对数据集进行数据增强和尺寸调整等预处理。然后利用 labelImg 图像标注工具, 对图像中的目标画矩形框, 得到 xml 格式文件, 最后利用得到的文件制作所需要的数据集进行训练。经过反复训练, 迭代次数达到 10 000 次左右, 总损失值能达到最小并收敛, 因此本文将训练次数设定为 10 000 次, 初始学习率设为 0.001。在训练前, 加载 VGG16 或 ResNet50 的预训练模型, 训练模型参数初始化, 加载预训练模型是因为防止数据量过小出现过拟合现象^[21], 因为模型参数在开始都分布在一个性能强的范围之内, 所以可以缓解过拟合, 能够加快模型的收敛速度。

通过 Tensorboard 观察训练过程中每个类别的平均准确率 (mAP, mean average precision) 的变化, 如图 6 所示, 改进 SSD 算法的 mAP 值上升较快, 且 mAP 值较高, 都在 0.9 附近。

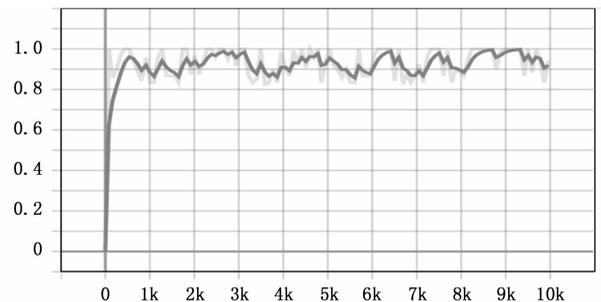


图 6 改进算法 mAP 值变化图

3.3 检测结果对比

测试改进的 SSD 算法与原 SSD 算法可以比较它们的检测效果, 其中有三组交通标志检测图像, 分别是 SSD 算法检测结果, 改进的 SSD 算法与原算法对比和改进算法的小目标测试效果。

图 7 为采用 SSD 算法训练模型检测到的结果, 可以看出, 直接采用 SSD 算法训练交通标志数据集得到的交通标志检测模型, 各类别的交通标志图像检测的准确率较低。

由图 8 可见, 改进后的交通标志检测效果相较于未改进的算法定位更加准确。原算法中的定位与目标位置相差较大, 而改进后的算法与目标位置吻合度较高。

图 9 为改进算法对较小的交通标志的检测, 从图中的检测案例可以看出, 较远距离的交通标志比较模糊且目标很小, 原算法几乎不能检测出来, 改进的 SSD 算法能够将小目标检测出来, 且在准确率方面取得了较好的效果。

上述实验结果显示, 改进的 SSD 算法在检测精度和对小目标检测方面相比原算法效果更优。

3.4 检测精度对比

目标检测任务中, 需要对模型进行评估, 性能评估使



图 7 SSD 检测示例图像

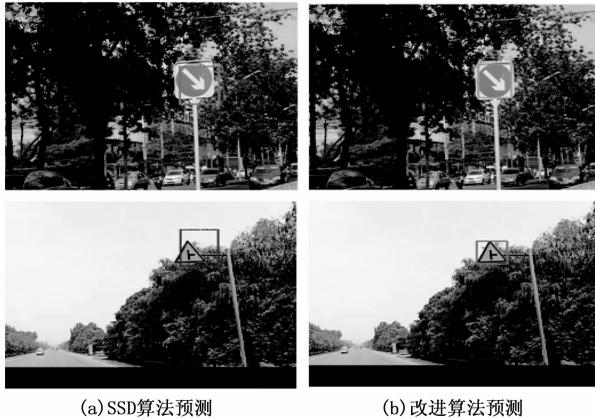


图 8 改进的 SSD 算法与原算法效果对比



图 9 小目标检测实例图像

用各种统计量如准确率 (accuracy), 精确率 (precision), 召回率 (recall) 等, 其计算方法如式 (6) 和 (7):

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

准确率即正确预测的目标数量除以总的目标数量; 精确率是指在所有预测结果中, 预测正确的数量和总的预测数量之间的比值; 召回率则是测试集中所有的正样本被正

确识别为正样本的比例。其中: TP (true positive) 为算法预测的为真的正样本; FP (false positive) 为算法预测为真的负样本; FN (false negative) 为算法预测为假的正样本。

本文使用每个类别的平均准确率 mAP 作为算法性能的评价指标, mAP 可以用来度量模型预测框类别和位置是否准确的指标。表 1 为改进 SSD 算法与其他算法的 mAP 值对比。其中 SSD* 表示改进的 SSD 算法。

表 1 改进算法与其他算法的 mAP 值对比

Model	Backbone	mAP	速度(s/张)
SSD	VGG16	0.667	0.36
Faster-RCNN	VGG16	0.793 6	0.84
YOLO V3	Darknet-53	0.905 4	0.27
SSD*	ResNet50	0.974 9	0.24

从表 1 可知, 改进的 SSD 检测算法的精度和速度均高于其他算法, 因此可以说明提出的改进方案是有效的, 且改进的算法适合对自然场景下交通标志的检测。

4 结束语

本文针对道路交通标志检测精度较低的问题, 提出了一种改进的 SSD 目标检测方法, 将原 SSD 算法中的 VGG16 特征网络替换为特征表达能力更强的 ResNet50, 并采用 K-means++ 确定先验框大小, 解决了原 SSD 模型随机确立先验框大小的缺点。

实验结果表明, 本文改进的 SSD 算法训练的交通标志检测模型, 在检测精度方面有所提高, 弥补了原 SSD 算法在自然环境下对道路交通牌目标检测准确率低的缺点, 提高了对远距离交通牌等小目标检测的识别率。本文所建立的道路场景下的交通标志检测方法具有理论参考和实际应用价值。

参考文献:

- [1] 欧阳维力. 基于单目视觉的交通标志检测与识别算法研究 [D]. 长沙: 湖南大学, 2014.
- [2] GANGADHAR H, SRINIVASAN E. Performance comparison of ROAD statistic based nonlinear filters for image denoising [C] // 2008 IEEE Region 10 and the Third International Conference on Industrial and Information Systems, IEEE (S2164-7011), 2008: 1-5.
- [3] 叶润, 张成, 李旭, 等. 基于计算机视觉的客机舱门识别与定位技术研究 [J]. 计算机测量与控制, 2021, 29 (3): 224-229.
- [4] FREUND Y, SCHAPIRE R E. A decision-theoretic generalization of on-line learning and an application to boosting [J]. Journal of Computer and System Sciences, 1997, 55 (1): 119-139.
- [5] OPTIMIZATION S M. A fast algorithm for training support vector machines [J]. CiteSeerX, 1998, 10 (1): 4376.
- [6] KE G, MENG Q, FINLEY T, et al. Lightgbm: a highly effi-

