

基于深度学习的交通信号灯快速检测与识别

钱弘毅¹ 王丽华¹ 牟宏磊²

(北京航空航天大学软件学院 北京 100191)¹ (西北工业大学自动化学院 西安 710072)²

摘要 交通信号灯检测与识别技术能够辅助司机做出正确的驾驶决策,减少交通事故的发生,为无人驾驶的实现提供安全保障。针对交通信号灯检测场景复杂多变、目标通常占检测数据集图片的比例极小等技术难点,提出了一种基于深度学习的交通信号灯快速检测与识别算法。整体框架包括如下 3 部分:基于启发式的图像预分割,用于缩小搜索范围,提升信号灯面板在输入图像中的相对大小和检测精度;基于深度学习的检测与识别,利用卷积神经网络准确地检测与识别信号灯;利用 NMS(Non-Maximum Suppression)算法去除上一阶段中重复的检测框。提出的 Split-CS-Yolo 模型在 LISA 数据集上取得了 96.08% 的 *mAP* 和 2.87% 的漏检率,相比 Yolo 系列的其他方法,其不仅有更高的准确率和更低的漏检率,还将模型大小缩小到原始 YOLOv2 的 8.6%,使得检测速度提升了 63%。

关键词 交通信号灯检测与识别,图像预分割,深度学习,NMS,快速检测

中图分类号 TP391 文献标识码 A DOI 10.11896/jsjx.190400026

Fast Detection and Identification of Traffic Lights Based on Deep Learning

QIAN Hong-yi¹ WANG Li-hua¹ MOU Hong-lei²

(College of Software, Beihang University, Beijing 100191, China)¹

(School of Automation, Northwestern Polytechnical University, Xi'an 710072, China)²

Abstract Traffic light detection and recognition technology can help drivers make correct driving decisions, reduce traffic accidents, and provide security for unmanned driving. Aiming at the technical difficulties such as the complex and variable traffic light detection scene, and targets typically account for a very small percentage of the dataset images, a fast detection and recognition algorithm for traffic light based on deep learning was proposed. The overall framework consists of three parts: heuristic-based image pre-segmentation, which is used to narrow the search range and improve the relative size and detection accuracy of the traffic light panel in the input images; detection and recognition based on deep learning, using convolutional neural networks to detect and identify traffic lights accurately; NMS (Non-Maximum Suppression) algorithm, which is used to remove the repeated detections of the previous stage. The proposed Split-CS-Yolo model achieves 96.08% *mAP* and 2.87% miss detection rate on the LISA dataset. Compared with other methods of the Yolo series, it not only has higher accuracy and lower missed detection rate, but also reduces the model size to 8.6% of the original YOLOv2, thus increasing the detection speed by 63%.

Keywords Traffic light detection and recognition, Image pre-segmentation, Deep learning, NMS, Fast detection

1 引言

交通信号灯是交通系统的重要设施,作为一种视觉信息,它可以辅助司机正确驾驶。但受疲惫、情绪、天气、光照、路况等因素的多重影响,司机有时无法依据交通信号灯做出正确决策,这不仅会影响交通系统的正常运转,同时也存在巨大的安全隐患。如果有一种能够准确检测并识别交通信号灯的技术来辅助司机驾驶,就可以有效提高行车效率和安全驾驶系数。

目前,国内外已经有很多学者展开了关于交通信号灯的检测与识别的研究,研究方法大体可分为基于数字图像处理的传统方法和基于卷积神经网络的深度学习方法。

在传统方法领域,许明文等^[1]利用交通信号灯的颜色、亮度和边缘特征生成融合显著图^[2],提取感兴趣区域 ROI 并对其进行形态学处理,利用形状特征和颜色特征进行目标分割,采用 HOG 特征提取算法并利用支持向量机 SVM 进行交通信号灯的分类识别。

Shi 等^[3]利用 HSV (Hue, Saturation, Value) 颜色模型和

到稿日期:2019-04-08 返修日期:2019-07-12

钱弘毅(1993-),男,硕士生,主要研究方向为深度学习、无人驾驶,E-mail:13861408913@163.com;王丽华(1961-),女,教授,主要研究方向为集成电路与物联网、软件工程项目管理与开发、基于视觉的机器人,E-mail:wanglihua@buaa.edu.cn(通信作者);牟宏磊(1981-),男,硕士生,主要研究方向为无人系统。

交通信号灯的变化规律对图像中的红色、绿色、黄色和黑色区域进行标记,以锁定候选区域;Ji 等^[4]用视觉选择注意^[5-7](Visual Selective Attention,VSA)模型获取交通信号灯候选区域;Jang 等^[8]基于整合低曝光技术和正常曝光技术,提出了高曝光技术,并以此为工具进行候选区域的选取。以上研究均利用 HOG 算法提取候选区域特征,并利用 SVM 进行交通信号灯的最终分类识别。

Omachi 等^[9]利用交通信号灯的颜色和边缘特征进行检测。首先预定义了交通信号灯的数学模型;得到给定输入图像后,依据 RGB 通道颜色将图像的每个像素聚类成 5 种类型,具有交通灯颜色的像素被用于 sobel 边缘检测;最后,通过对模型和检测边缘进行投票来检测交通信号灯。

Kim 等^[10]利用颜色阈值分割对交通信号灯进行检测;ZHANG 等^[11]利用多特征融合算法对图像中检测到的点和块进行打分,以确定最终的候选区域。这些方法对环境变化敏感度较高,受光照影响较大,因此实用性较差。2009 年中国智能车辆挑战赛中利用 Haar 特征结合 Adaboost 分类识别的方法进行交通信号灯的检测与识别^[12]。

在使用深度学习的算法领域,李子康等^[13]提出了一种渐变上下文特征检测算法,以随机点为中心逐步扩散,同时,根据引入不同尺寸的上下文特征检测每一步扩散结果,每一步扩散结果均涵盖目标周围的上下文信息,具有上下文信息的交通信号灯更易于与汽车尾灯区分。但该算法用到的模型过于简单,参数量少,特征提取能力较差,检测结果并不理想。

Kim 等^[14]将原始的 SSD^[15]应用于信号灯检测领域。检测分为两个阶段,第一阶段使用一个粗粒度的 CNN 尽可能多地检测有可能是信号灯的位置。第二阶段使用一个细粒度的 CNN 和基于物体时空位置关系的过滤方法,把第一阶段中 false positive(假阳性)的样本去除。该方法的流程较为复杂,且在自制的数据集上对红色圆型信号灯仅取得了 68.0% 的查准率和 29.5% 的召回率,难以在实际场景中应用。

Müller 等^[16]改进了 SSD 的结构,提出 TL-SSD,能够准确地检测小物体。TL-SSD 使用 Inception-v3^[17] 替换 VGG^[18] 作为基础网络,将背景上下文信息和本地信息结合起来用于信号灯的检测与识别。但该方法在使用背景信息降低误减率的同时,会降低信号灯位置的检测准确性。于是其将高低层的特征结合起来以缓解这一问题。虽然算法的识别准确率有所提高,在其自制的数据集上检测到了 95% 的信号灯,但是在使用 Nvidia Titan Xp 显卡的情况下其依然只能做到每秒检测 10 帧图像,难以保证实时性。

Han 等^[19]对 Faster-RCNN^[20]进行了改进,将原始主干网 VGG16 中的 pool4 层移除,增加了特征图尺寸,RPN(区域生成器)可以更方便地提取出小物体的特征。另外,其引入了在线难样本挖掘策略,使得算法进一步提升了对于训练数据较少的信号灯类型的检测鲁棒性,在其自制数据集上的识别准确率较原始 Faster-RCNN 提高了 12.1%。但是该方法计算量较大,需要使用两块 NVIDIA GeForce GTX 1080 显卡,难以保证实时性。

Lu 等^[21]借鉴了视觉注意力机制的概念,使用了两个级联的 Faster-RCNN 进行信号灯检测。第一个网络输入的是低分辨率的图像,负责预测信号灯所在的大致区域。然后将对应位置的高分辨率图片输入第二个网络,并由之确定信号灯的最终位置。这种做法大大减少了在高分辨率图像上的搜索空间,提升了检测速度,但识别准确率一般,最终在自制的数据集上取得了 87.0% 的 *mAP*。

Niu 等^[22]提出了一种结合传统方法与深度学习的信号灯检测框架。该框架包含了一个启发式的候选区域选择模块来确认所有可能是信号灯的区域,以及一个轻量级的 CNN 分类器来对结果进行分类。该框架检测速度非常快,在 LISA^[23](Laboratory for Intelligent & Safe Automobiles)数据集上的测准率较高,达到 96.6%。但是其漏检十分严重,有 33.3% 的信号灯被漏检。

综上,传统的非深度学习方法利用人工提取的信号灯颜色和形状信息进行候选区域检测时达到的效果较好。但信号灯所在的场景复杂多变,很难用单一算法对其特征进行提取,故传统算法只对背景简单的场景有效,实用性不高。

虽然深度学习方法不需要设计复杂的人工提取特征方法,对不同场景的泛化能力较强,可移植性较好。但是,目前的深度学习方法在应用于信号灯检测场景时仍然存在两个重要问题。

首先,信号灯面板在数据集图片中所占比例极小。图 1 展示了 LISA 数据集中一些标注的样例图片。它们包含了不同时间、地点、光照下的路面场景,信号灯在整幅图中占比很小,比如,在分辨率为 1280×960 的图片中占 10×18 个像素,仅 0.015%。常用的深度学习目标检测算法如 Faster-RCNN,SSD,Yolo^[24]对于此类小目标的检测效果不佳。



图 1 LISA 中的样例图片

Fig. 1 Sample pictures in LISA

其次,常用的深度学习模型参数太多,即使是较快的 one-stage 目标检测网络,Yolov2 及 Yolov3 单个模型大小都在 200 M 以上,导致计算量太大,在算力不强的情况下难以直接在实际场景中应用。

针对以上两个问题,本文提出了一种基于深度学习的交通信号灯快速检测与识别算法。检测框架由 3 部分组成,分别为启发式的图像预分割处理、基于深度学习的检测与分类、去除重复框的后处理步骤。在 LISA 数据集上的实验结果显

示,所提出的 Split-CS-Yolo 模型鲁棒性强,检测与识别准确率高, mAP 达到了 96.08%,漏检率仅为 2.87%;同时,模型大小缩小到 Yolov2 的 8.6%,仅 17.5M,检测速度提升了 63%。

2 交通信号灯的识别与检测

2.1 图像预处理

2.1.1 数据集选择

使用 LISA 白天的数据集,该数据集的分辨率大小为 1280×960 ,共包含 20248 张标注图片和 55830 个标注实例。将数据集中的图片按照 4:1 的比例随机分为训练集和测试集。

原始标注共分为红色圆形信号灯、红色左向箭头信号灯、黄色圆形信号灯、黄色左向箭头信号灯、绿色圆形信号灯、绿色前向箭头信号灯、绿色左向箭头信号灯 7 类。但由于各类别的样本分布极不均衡,故参考文献[22],只设置 go 和 stop 两种类别,最终标注数如表 1 所列。

表 1 LISA 数据集标注样本的分布

Table 1 Distribution of labeling samples in LISA dataset

	go	stop
标注数	22398	33432

2.1.2 预分割方法

数据输入网络前需要进行预处理,即依据先验信息不对整张图片进行处理,而只搜索信号灯有可能出现的上半部分区域,以提高检测精度。如图 2 所示,为了让信号灯在单张图片中所占的面积比例增加,从而使网络提取出的信号灯特征尽可能清晰,将图片分割成大小为 736×544 的两个区域,分别输入网络训练,从而使信号灯在整幅图像中所占的面积变为原来的 3.07 倍。

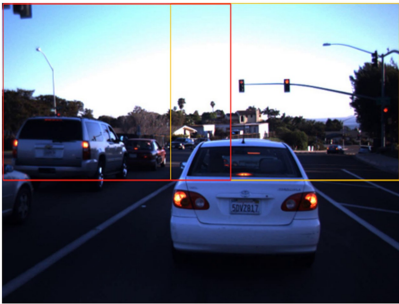


图 2 图像预分割

Fig. 2 Image pre-segmentation

为了确保检测到图像中的信号灯,设置了相对于原图宽度 15% 的重叠部分,在测试阶段会综合两者计算出最终的检测结果。

2.1.3 数据增强

与 Yolov2 的做法一致,在训练阶段在线对图片使用随机缩放、扣取、HSV 变换来进行数据增强,以防止过拟合,提升网络性能。

其中,设置缩放尺度为 32 的倍数,比例因子为 10~19 之间的整数。扣取范围为原始图像大小的 70%~130%。HSV 变换则是在 HSV 颜色空间中以相对于初始值 $2/3 \sim 3/2$ 的倍

率随机调整图像的曝光和饱和度。数据增强后,实际输入网络的图片有包含信号灯的,也有不包含信号灯的。图 3 是包含了信号灯的图片示例。



图 3 数据增强图片示例

Fig. 3 Examples of data augmentation image

2.2 卷积神经网络的训练

2.2.1 特征提取网络结构设计

网络整体框架基于 Yolov2^[25], 本文将其中的 backbone 网络 Darknet19 替换为性能更好的 ShuffleNet v2^[26]。为加速收敛和避免过拟合,每个卷积层后会跟着一个批标准化^[27] (Batch Normalization)层和非线性激活函数层。非线性激活函数层使用 leaky ReLU。相比于 Yolov2 设置网络输入大小为 416×416 ,本模型设置其大小为 544×544 ,网络结构如表 2 所列。

表 2 ShuffleNet v2 的网络结构

Table 2 Network architecture of ShuffleNet v2

Layer	Output size	Kernel size	Stride	Repeat	Output channels
Image	544×544				3
Conv1	272×272	3×3	2	1	24
MaxPool	136×136	3×3	2		
Stage2	68×68		2	1	116
	68×68		1	3	
Stage3	34×34		2	1	232
	34×34		1	7	
Stage4	17×17		2	1	464
	17×17		1	3	
conv5	17×17	3×3	1	1	464
conv6	17×17	1×1	1	1	35

ShuffleNet v2 的两个基本组成模块如图 4 所示,其中,图 4(a)和图 4(b)是两种不同 block 结构,两者的差别在于后者缩小了特征图尺寸。ShuffleNet v2 利用分组卷积(group convolution)与通道随机分配(channel shuffle)压缩模型参数大幅减少了计算时间。

原始的 ShuffleNet v2 仅使用了最后一层的特征,最后一层的特征经过了若干次步长大于 1 的池化或卷积操作,分辨率较低的小目标物体特征容易消失,而位于网络低层的更加细粒度的特征更好地保存了小目标物体的信息,故将 ShuffleNet v2 不同高度层的特征结合起来。为了不过多地增加计算量进而降低检测速度,本文并没有使用传统的级联式特

征金字塔,或类似 SSD 根据不同层特征分别预测结果,而是在维数变化后直接拼接来自高、中、低层的特征,再将其输入到后面的卷积层。依据此特性,将网络命名为 Concentrate-ShuffleNet(CSNet),其整体框架命名为 CS-Yolo。

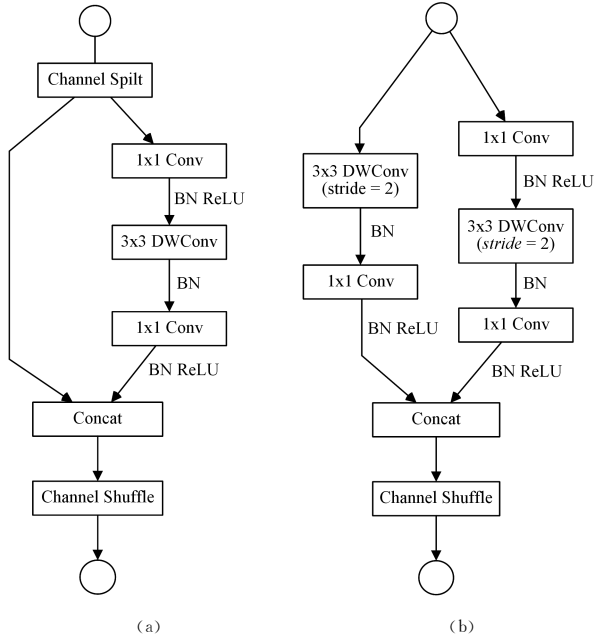


图 4 ShuffleNet v2 的两个基本组成模块

Fig. 4 Two basic components of ShuffleNet v2

如图 5 所示,图像特征在经过 ShuffleNet v2 的 stage2 后,生成的特征图 F_2 的分辨率为 68×68 ,通道数为 116。 F_2 在输入 stage3 的同时,还有另外一个分支,该分支经过一个 1×1 的卷积核,保持特征图分辨率不变,通道数减少为 16,再经由一个步长为 4 的张量重组的操作,最终生成通道数为 256、大小为 17×17 的特征图。

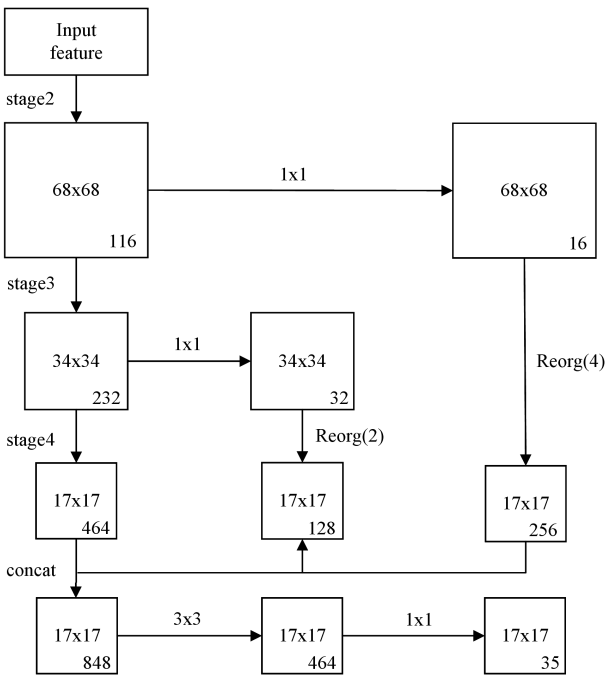


图 5 特征图的变化示意图

Fig. 5 Schematic diagram of change of feature map

具体地,张量重组对输入的特征张量在维度 2,3 方向以步幅 4 切片,然后在维度 1 方向上堆叠。与经由 stage4 的特征图大小保持一致,对经由 stage3 的特征图采取类似的操作,最终对 3 个相同大小的特征图直接拼接,得到一个通道数为 $464 + 128 + 256 = 848$ 的张量。

2.2.2 检测框预测

为了提升召回率,本文并非使用全连接层直接预测检测框位置,而是同 YoloV2 一样使用预选框(anchor boxes)机制,CSNet 最终输出的特征图大小为 17×17 ,即图像被分为 17×17 个网格。

每个 anchor box 只负责周围的网格,这有利于收敛网络和提升效率。为了让参数就更容易学习,模型更加稳定,对定位预测值进行归一化。其中 t_x, t_y, t_w, t_h 用于预测 anchor box 的中心点位置以及宽度和高度。 t_o 则表示该 anchor box 的目标性评分(Objectness Score),即该 anchor box 内包含物体的置信度高,通过 logistic 回归计算。目标性评分标签为 1 的情况有两种:

- (1)该 anchor box 与某个真实标注的重合度比其他 anchor box 都高;
- (2)该 anchor box 与某个真实标注的重合度不是最大的,但是超过某个阈值(本文设置为 0.5,即只有目标性评分在 0.5 以上的检测框才有机会输出)。

最终的检测框位置用下式计算:

$$b_x = \sigma(t_x) + c_x \tag{1}$$

$$b_y = \sigma(t_y) + c_y \tag{2}$$

$$b_w = P_w e^{t_w} \tag{3}$$

$$b_h = P_h e^{t_h} \tag{4}$$

其中, c_x 和 c_y 是所在网格与图片左上角的偏移值; P_w 和 P_h 是 anchor prior 的先验宽与长。

为了增强检测网络对不同尺度物体的泛化能力,需要设置多种长宽比和尺寸的 anchor box, Faster-RCNN 和 SSD 采用人工选择的方式,但是效果并不是最优的。为了让 anchor box 的尺寸尽可能与所使用的数据集的目标物体尺寸接近,本文通过 K 均值(K-means)聚类算法分析标注文件数据获得。K-means 是一种迭代求解的聚类分析算法,其工作原理可以分为如下 4 个步骤:

- (1)随机在样本中选取 K 个对象,并以此作为初始的聚类中心;
- (2)分别计算每个对象与当前聚类中心之间的距离,并将其分配给与它距离最近的聚类中心,这些分配的对象和聚类中心一起组成了一个聚类;
- (3)当所有对象分配完毕时,每个聚类中心会根据当前聚类中所有的对象重新计算;
- (4)不停重复步骤(2)和(3),直到聚类中心保持稳定为止。

本文 K-means 算法聚类的对象就是 anchor box 的宽与高,其距离函数 d 使用重叠度 IOU(Intersection over Union)

定义,计算公式如下:

$$d(box, centroid) = 1 - IOU(box, centroid) \quad (5)$$

如图6所示,随着聚类维数的增加,anchor prior与真实的平均IOU也会增加。但设置太多的anchor box会导致检测速度下降。当聚类维数 k 设置为5时,准确率和速度会达到一个比较均衡的状态,平均IOU达到83.6%。

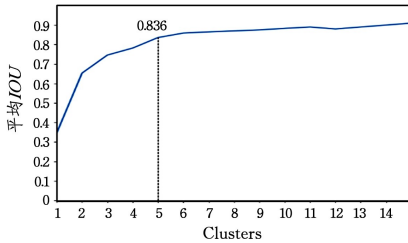


图6 LISA标注框尺度聚类

Fig. 6 Clustering box dimensions on LISA

每个网格设置的anchor box数量为5,整个特征图上共有 $17 \times 17 \times 5 = 1445$ 个anchor box。图2中最终的特征图通道数 C 的计算公式为:

$$C = (n_{\text{output}} + n_{\text{cls}}) * n_{\text{anchor}} \quad (6)$$

其中, n_{cls} 为标签种类数目,计算可得 $C = 35$ 。

对预分割处理后的图像标注数据聚类后发现,信号灯检测框的宽长比大致在0.44~0.55之间,符合在LISA数据集中信号灯面板大部分是竖直的情况。

CS-Yolo在训练阶段对每个anchor包含的区域进行目标性评分(Objectness Score),即包含目标的可能性大小。对于同一个目标,只选择目标性评分最高的anchor prior进行预测,而不是输出所有anchor prior,这样可以减少计算量。同时设置检测阶段目标性评分阈值为0.5,即只有目标性评分在0.5以上的检测框才有机会输出。我们尝试使用更小的置信度阈值,比如0.005,此时 mAP 还会提升2%,但是实际的检测结果会出现“叠框”现象,即对一个目标的重复检测,造成 mAP 的“虚高”,故并未实际使用。

2.2.3 损失函数

类似于Yolo,网络的损失函数由3个种类的loss加权相加得到,分别为坐标偏差损失 L_{coord} 、置信度损失 L_{conf} 、类别损失 L_{class} ,其权重比例设置为2:1:1。其中, L_{coord} 包括使用二值交叉熵(Binary cross-entropy) loss的 $L_{\text{coord}_{xy}}$ 和使用平滑L1 loss的 $L_{\text{coord}_{wh}}$; L_{conf} 包括均使用二值交叉熵 loss的 $L_{\text{conf}_{obj}}$ 和 $L_{\text{conf}_{noobj}}$; L_{class} 则使用softmax层和交叉熵 loss,而非Yolov3^[28]中使用的逻辑回归(Logistic regression)和二值交叉熵 loss。实验表明这样网络更容易收敛。

2.2.4 超参数的选择与训练策略

单次训练的批次大小设置为64,共迭代160000次。优化方法使用随机梯度下降法(SGD),由于初始时负样本居多,为了保持网络训练稳定,使用warm-up策略,即初始阶段设置一个较小的学习率0.0001来“预热”网络,短时间内逐步增大学习率。之后随着loss的下降与准确率的提升,学习率会分段逐渐下降。学习率的具体变化如表3所列。

表3 学习率变化

Table 3 Change of learning rate

学习率变化点(batch)	学习率
0	0.0001
400	0.0005
700	0.001
900	0.002
1000	0.02
40000	0.004
60000	0.001
80000	0.0002
120000	0.00004
140000	0.00001

训练使用finetune^[29]策略,先在清华大学计算机学院公布的TTTL^[30]交通信号灯数据集上进行预训练,接着使用训练好的backbone网络的权重外加其他权重对其随机初始化,之后将其应用在LISA数据集上继续训练,预训练的超参数与训练参数保持一致。

2.3 去除重复检测框

网络在测试阶段无须进行卷积神经网络的反向传播,但是对同一个目标可能有多个检测结果。本文使用非极大值抑制算法NMS(Non Maximum Suppression)将对于同一个物体的检测框数量降至1,NMS的IOU阈值 N_t 设置为0.45。

NMS算法单次仅作用于同一类别的目标,这里令所有待处理的检测框集合为 B ,最终的检测结果集合为 D 。首先选择 B 中具有最大置信度的检测框 M ,将其从 B 中移除,并加入到 D 中。再将 B 中剩余检测框中与 M 的IOU大于阈值 N_t 的框从 B 中移除。重复上述移除-添加-移除的过程,直到集合 B 为空。然后再切换到下一个类别的目标,继续用NMS对其进行处理。

图7的第一列展示了交通信号灯原图,第二、三列分别是使用NMS算法之前与之后的检测结果图。为方便观察,对后两列信号灯的所在区域进行了局部放大。可以发现,在经过NMS算法处理之后,同一信号灯仅剩余单一的检测框。

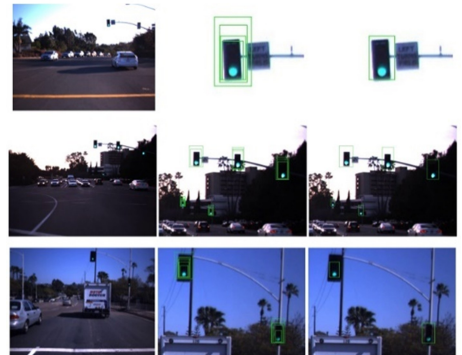


图7 NMS前后效果对比

Fig. 7 Comparison of effects before and after using NMS

3 实验结果与分析

实验中,CPU使用的是Intel(R) Core(TM) i7-8700 CPU @ 3.20 GHz,GPU使用单块NVIDIA GeForce GTX 1070Ti。实

验在 Ubuntu 操作系统下使用 pytorch 来搭建卷积神经网络。

Split-CS-Yolo 与其他模型的检测结果如表 4 所列,为保证公平性,仅选取针对 LISA 数据集进行检测的方法进行对比。其中,Shuffle-Yolo 只将 Yolov2 的 backone 网络替换为 ShuffleNet v2; CS-Yolo 进一步增强了 Shuffle-Yolo,拼接 ShuffleNet v2 的不同高度层的特征,检测小目标物体的能力更强,且只增加极少的额外计算时间;Split-CS-Yolo 将训练数据改为预分割图片,检测速度相比 CS-Yolo 降低了一半,但仍然快于 Yolov2 和 Yolov3,且检测 *mAP* 到与 Yolov3 相当。虽然其测准率和 Niu^[22]的方法还有微小差距,但是漏检率最低。Split-CS-Yolo 在检测准确率和速度之间取得了较好的平衡。

表 4 各模型检测结果的对比

Table 4 Comparison of test results of each model

Models	<i>mAP</i> /%	单张图片前向传播耗时/ms	Size/MB	漏检率/%
文献[13]	74.34	—	—	—
文献[22]	—	—	0.8	33.30
Yolov2	93.48	17.719	202.4	5.34
Yolov3	96.01	29.624	246.4	3.29
Shuffle-Yolo	89.94	5.101	11.1	8.24
CS-Yolo	92.46	5.339	17.5	5.47
Split-CS-Yolo	96.08	10.846	17.5	2.87

定义漏检率为经信号灯检测模型处理后,信号灯待识别区域中漏检信号灯数占实际信号灯总数的比例,计算公式为:

$$\text{漏检率} = \frac{\text{未检测数}}{\text{全部待检测数}} \quad (7)$$

图 8 是 Split-CS-Yolo 的检测 *mAP* 值随着 batch 数增加而变化的曲线图。实际实验中,迭代到第 139 300 次 batch 时,检测 *mAP* 达到最大值 96.08%。

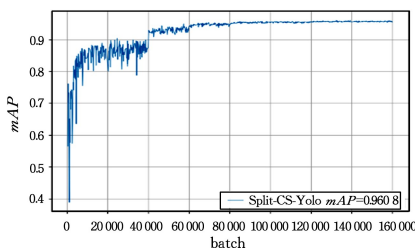


图 8 训练 batch 与 *mAP* 曲线

Fig. 8 Batch-*mAP* curve

Yolov2, Yolov3, Split-CS-Yolo 的准确率-召回率 (Precision-Recall) 曲线如图 9 所示。

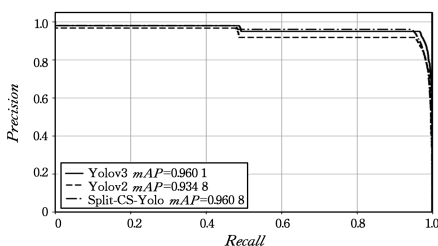


图 9 准确率-召回率曲线对比

Fig. 9 Precision-recall curve comparison

单类别目标的准确率(Precision)的计算公式如下:

$$\text{准确率} = \frac{\text{正确检测数}}{\text{检测总数}} \quad (8)$$

单类别目标的召回率(Recall)的计算公式如下:

$$\text{召回率} = \frac{\text{正确检测数}}{\text{全部待检测数}} \quad (9)$$

几个模型的检测效果对比如图 10 所示,第一列展示的是原图像,第二、三、四列分别是 Yolov2, Yolov3, Split-CS-Yolo 的检测结果,红绿框分别表示 go, stop 类别。为了方便观察,后三列对信号灯所在位置的附近区域进行了局部放大。相比另外两个模型, Yolov2 漏检较为严重,很多远距离的交通信号灯检测不到。而 Yolov3 与 Split-CS-Yolo 的检测效果则各有长短。

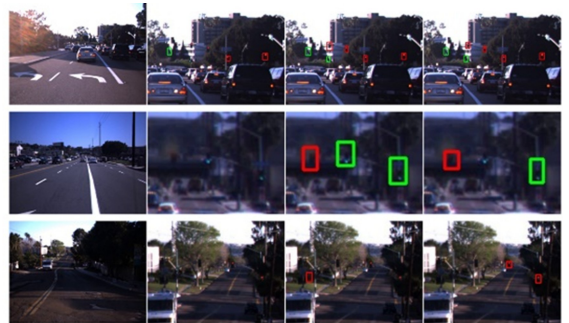


图 10 检测结果对比(电子版为彩色)

Fig. 10 Comparison of test results

结束语 针对 LISA 交通信号灯数据集提出了一种基于深度学习的具有良好准确性和较快速度的检测算法。该算法利用改进的 ShuffleNet v2 作为 backbone 网络,大幅减少了计算量;将待检测图片分割为两块分别检测并汇总检测结果,提高了准确率;最后使用 NMS 算法去除重复检测框,完成了信号灯的检测与识别。实验结果表明,Split-CS-Yolo 模型鲁棒性强,检测与识别准确率高,*mAP* 达到了 96.08%,漏检率仅为 2.87%。相比于 Yolov2, Split-CS-Yolo 模型的大小缩小到原来的 8.6%,仅 17.5M,检测速度提升了 63%。该模型未来可应用于辅助驾驶和无人驾驶等领域。但其仍存在很大的提升空间,下一步的研究方向是针对黄色信号灯或数字信号灯进行更精确的识别。

参 考 文 献

[1] XU M W, ZHANG C Y. Traffic Light Detection and Recognition Based on Saliency Map[J]. Computer and Digital Engineering, 2017, 45(7): 1397-1401. (in Chinese)
许明文,张重阳. 基于显著性特征的交通信号灯检测和识别[J]. 计算机与数字工程, 2017, 45(7): 1397-1401.

[2] KOCH C, ULLMAN S. Shifts in selective visual attention: towards the underlying neural circuitry[J]. Hum Neurobiol, 1987, 4(4): 219-227.

[3] SHI X M, ZHAO N, XIA Y J. Detection and classification of traffic lights for automated setup of road surveillance systems [J]. Multimedia Tools & Applications, 2016, 75(20): 12547-12562.

- [4] JI Y, YANG M, LU Z, et al. Integrating visual selective attention model with HOG features for traffic light detection and recognition[C]//Intelligent Vehicles Symposium. Korea: IEEE, 2015:280-285.
- [5] ZHANG Q R, GU G C, XIAO H M. Computational Model of Visual Selective Attention[J]. Robot, 2009, 31(6):574-580.
- [6] YANG M, LU Z C, GUO L D, et al. Vision-Based Environmental Perception and Navigation of Micro-Intelligent Vehicles [C]//Advances in Intelligent Systems and Computing. 2014: 653-665.
- [7] LIU Q, QIN S Y, LI Z C. The Modeling of Visual Selective Attention and Its Applications[J]. Science & Technology Review, 2010, 28(1):107-115. (in Chinese)
刘琼, 秦世引, 李志成. 视觉选择性注意的模型化计算及其应
用前景[J]. 科技导报, 2010, 28(1):107-115.
- [8] JANG C H, KIM C, KIM D, et al. Multiple exposure images based traffic light recognition[C]// IEEE Intelligent Vehicles Symposium Proceedings. USA: IEEE, 2014:1313-1318.
- [9] OMACHI M, OMACHI S. Detection of traffic light using structural information[C]// IEEE 10th International Conference on Signal Processing. Beijing: IEEE, 2010:809-812.
- [10] KIM Y K, KIM K W, YANG X L. Real Time Traffic Light Recognition System for Color Vision Deficiencies[C]// International Conference on Mechatronics and Automation. Harbin: IEEE, 2007:76-81.
- [11] ZHANG Y, XUE J, ZHANG G, et al. A multi-feature fusion based traffic light recognition algorithm for intelligent vehicles[C]// Proceedings of the 33rd Chinese Control Conference. 2014:4924-4929.
- [12] KASTNER R, MICHALKE T, BURBACH T, et al. Attention-based traffic sign recognition with an array of weak classifiers [C]// Intelligent Vehicles Symposium. San Deigo, CA: IEEE, 2010:333-339.
- [13] LI Z K, XU G Z, GUO M M. Traffic signal recognition method based on gradient content feature[J]. Transducer and Microsystem Technologies, 2018, 37(10):38-40. (in Chinese)
李子康, 徐桂芝, 郭苗苗. 基于渐变上下文特征的交通灯识别方法[J]. 传感器与微系统, 2018, 37(10):38-40.
- [14] KIM J, CHO H, HWANGBO M, et al. Deep Traffic Light Detection for Self-driving Cars from a Large-scale Dataset[C]// IEEE 2018 21st International Conference on Intelligent Transportation Systems (ITSC). USA: IEEE, 2018:280-285.
- [15] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector[C]// Proceedings of European Conference on Computer Vision. 2016:21-37.
- [16] MÜLLER J, DIETMAYER K. Detecting Traffic Lights by Single Shot Detection[C]// IEEE International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018.
- [17] SZEGEDY C, VANHOUCKE V, IOFFE S, et al. Rethinking the Inception Architecture for Computer Vision[C]// 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2016:2818-2826.
- [18] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition [C]// Proceedings of ICLR. 2015.
- [19] HAN C, GAOG Y, ZHANG Y. Real-time small traffic sign detection with revised faster-RCNN[J]. Multimedia Tools and Applications, 2018, 78(10):13263-13278.
- [20] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149.
- [21] LU Y F, LU J M, ZHANG S H, et al. Traffic signal detection and classification in street views using an attention model[J]. Computational Visual Media, 2018, 4(3):253-266.
- [22] NIU J W, LIU Y, GUIZANI M, et al. Deep CNN-based Real-time Traffic Light Detector for Self-driving Vehicles[C]// IEEE Transactions on Mobile Computing. 2019.
- [23] PHILIPSEN M P, JENSEN M B, MOGELMOSE A, et al. Traffic light detection: a learning algorithm and evaluations on challenging dataset[C]// 2015 IEEE 18th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2015: 2341-2345.
- [24] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016:779-788.
- [25] REDMON J, FARHADI A. YOLO9000: Better, Faster, Stronger [C]// 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2017:6517-6525.
- [26] MA N, ZHANG X, ZHENG H T, et al. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design[C]// Lecture Notes in Computer Science (ECCV 2018). Cham: Springer, 2018.
- [27] IOFFE S, SZEGEDY C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift[C]// Proceedings of the 32nd International Conference on International Conference on Machine Learning (ICML'15). 2015:448-456.
- [28] REDMON J, FARHADI A. YOLOv3: An Incremental Improvement[J]. arXiv:1804.02767.
- [29] YOSINSKI J, CLUNE J, BENGIO Y, et al. How transferable are features in deep neural networks? [C]// Proceedings of the 27th International Conference on Neural Information Processing Systems. 2014:3320-3328.
- [30] LU Y, LU J, ZHANG S, et al. Traffic signal detection and classification in street views using an attention model[J]. Computational Visual Media, 2018, 4(3):59-72.