

# 基于改进 YOLOv3 和立体视觉的园区障碍物检测方法

胡丹丹, 张莉莎, 张忠婷

(中国民航大学 电子信息与自动化学院, 天津 300300)

**摘要:** 为了解决无人驾驶障碍物检测在园区场景中准确率低、实时性不足等问题, 提出一种基于改进 YOLOv3 (You Only Look Once) 和立体视觉的障碍物检测方法: YOLOv3-CAMPUS; 通过改进特征提取网络 Darknet-53 的结构减少前向推断时间, 进而提升模型检测速度, 通过增加特征融合尺度提升检测精度和目标定位能力; 通过引入 GIOU (Generalized Intersection over Union) 改进目标定位损失函数, 通过改进 k-means 算法降低初始聚类点造成的聚类偏差, 进而提高模型检测精度; 通过立体视觉相机获得预测边界框中心点的深度信息, 确定障碍物与无人车的距离; 实验结果表明, 提出的方法较原模型在园区混合数据集 (KITTI+PennFudanPed) 上平均精度提升了 4.19%, 检测速度提升了 5.1 fps; 在自建园区数据集 (HD-Campus) 上平均精度达到 98.57%, 均能满足实时性要求。

**关键词:** 园区障碍物; YOLOv3; 特征提取; GIOU; 立体视觉

## Obstacle Detection Method in Campus Based on Improved YOLOv3 and Stereo Vision

Hu Dandan, ZHANG Lisha, ZHANG Zhongting

(Civil Aviation University of China, Tianjin 300300, China)

**Abstract:** In order to solve problems of low accuracy and lack of real-time performance of unmanned vehicle obstacle detection in the campus scene, an obstacle detection method based on improved YOLOv3 (You Only Look Once) and stereo vision was proposed: YOLOv3-Campus. The forward inference time was reduced and the model detection speed was faster by improving the structure of the feature extraction network Darknet-53. The detection accuracy and target location accuracy were improved by increasing the feature fusion scale. Meanwhile, by using GIOU (Generalized Intersection Over Union), the target location loss function was improved. Enhanced K-means algorithm could reduce the cluster deviation caused by the initial clustering point, then the model detection accuracy was ameliorated. In additional, the depth information of the predicted boundary frame's center point was obtained by stereo vision camera. Then the distance between the obstacle and the unmanned vehicle could be measured. Experimental results show the proposed method increases the average accuracy by 4.19% and the detection speed increases by 5.1 fps compared with the original model on the campus mixed data set (KITTI+PennFudanPED). On the self-built campus data set (HD-Campus), average accuracy could reach 98.57%, and it could satisfy the real-time requirements by using improved method.

**Keywords:** campus obstacle; YOLOv3; feature extraction; GIOU; stereo vision

### 0 引言

由于封闭园区环境具有结构化程度高、道路环境简单、行驶路线固定等特点<sup>[1]</sup>, 无人驾驶技术在园区物流、园区巡逻等场景的应用越来越广泛。障碍物的准确检测是园区无人车安全行驶的基础, 也是当前无人驾驶技术研究的重难点之一。其中基于视觉的障碍物检测方法具有成本较低、检测内容直观、检测范围广泛等优点, 是目前最常用的检测方法。

传统的视觉障碍物检测算法以基于人工特征提取方法为主<sup>[2]</sup>, 存在计算复杂度高和检测速度慢等问题<sup>[3]</sup>。基于深

度学习的障碍物检测算法主要分为基于区域建议和基于逻辑回归两类。其中, 以 R-CNN<sup>[4-6]</sup> 系列模型为代表的基于区域建议算法生成样本候选框, 通过卷积神经网络进行样本分类<sup>[7]</sup>, 此类算法检测精度较高, 但计算复杂度大, 无法满足实时性要求; 而以 YOLO<sup>[8-10]</sup> 系列模型为代表的基于逻辑回归算法不需要生成候选框, 网络直接输出边界框信息, 检测速度和精度均有所提升, 但仍存在少量漏检情况。

因而提升检测精度和速度成为近几年障碍物检测算法改进的重点。针对文献 [10] 存在的问题, 文献 [11] 采用数据增强、引入 CIOU (Complete Intersection Over Union)

收稿日期: 2021-03-01; 修回日期: 2021-03-31。

基金项目: 天津市科技计划项目 (17ZXHLGX00120)。

作者简介: 胡丹丹 (1979-), 女, 吉林人, 硕士, 副教授, 主要从事机器人环境感知、多传感器数据融合方向的研究。

引用格式: 胡丹丹, 张莉莎, 张忠婷. 基于改进 YOLOv3 和立体视觉的园区障碍物检测方法[J]. 计算机测量与控制, 2021, 29(9): 54-60.

损失函数等方式, 平均精度提升的同时检测速度有所牺牲; 其优化版本 YOLOv5 减小模型体量, 检测速度有较大提升, 但适用场景和精度均受到影响。

综上发现, 文献 [10] 采用的 YOLOv3 算法在检测精度和速度上均具有较好的表现。近几年, 国内外诸多学者针对基于 YOLOv3 的检测方法作出了不同程度的改进。文献 [12] 通过采用 Gaussian-YOLOv3 与变形卷积技术, 有效解决了小目标的检测问题; 文献 [13] 结合 Tiny-YOLOv3 和目标跟踪算法, 提升了无人机针对地面目标的检测速度; 文献 [14] 通过改进特征提取网络和定位损失函数, 降低了非机动车类目标的漏检率; 文献 [15] 考虑融入平滑标签, 改进特征提取网络结构, 一定程度解决了行人检测实时性和精度不能同时兼顾的问题。

基于以上研究, 为实时准确获得园区障碍物的类别与位置信息, 本文提出一种结合改进 YOLOv3 和立体视觉的园区无人车障碍物检测方法: 通过优化特征提取网络、引入 GIOU 损失函数、采用 k-means++ 聚类方法对 YOLOv3 算法进行改进, 并结合立体视觉测距原理获得无人车与障碍物的距离, 实验结果表明在公开数据集与园区自建数据集上均取得较好的检测效果。

### 1 YOLOv3 算法

如图 1 所示, YOLOv3 的网络结构主要包括基础网络和 YOLOv3 分支。其中 Darknet-53 负责特征提取, YOLOv3 分支负责目标检测。YOLOv3 网络将图片尺寸方缩到  $416 \times 416$  作为网络输入, 输入的障碍物图像经过 darknet-53 特征提取网络后得到 3 个尺度 ( $13 \times 13$ 、 $26 \times 26$ 、 $52 \times 52$ ) 的特征图。接着将输入的图像分为  $S \times S$  个网格, 每个网格单元负责预测 3 个边界框。预测信息包含边界框的位置及尺寸信息 (目标中心坐标  $(x, y)$  与边界框尺寸  $(w, h)$ )、置信度得分及其属于某类别的条件概率。最后设置得分阈值, 通过非极大值抑制过滤分数较低的边界框, 其余边界框信息将作为最终的障碍物检测结果输出。

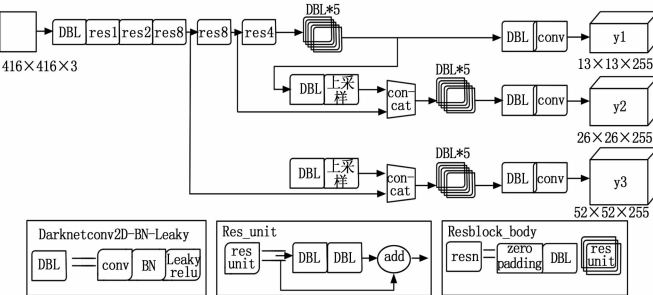


图 1 YOLOv3 网络结构图

### 2 结合立体视觉的改进 YOLOv3 算法

目前基于 YOLOv3 的障碍物检测研究中普遍存在检测实时性差、具体场景检测准确率低等问题, 本文针对园区无人车障碍物检测场景, 对 YOLOv3 算法进行特征提取网络结构、损失函数和聚类方法三方面的改进。

#### 2.1 特征提取网络结构改进

考虑到无人车障碍物检测的实时性要求以及园区检测目标多为行人和非机动车的中小型目标的情况, 对 YOLOv3 采用的 Darknet-53 特征提取网络结构进行以下改进。

##### 1) 网络结构简化:

如图 1 所示, DBL 是 YOLOv3 的基本组成部件, 在 DBL 单元结构中, 批次归一化 (batch normalization, BN) 层一般置于卷积层之后, 非线性处理单元 (Leaky RELU) 之前。其计算过程<sup>[16]</sup>如公式 (1) 所示, 首先计算输入数据均值  $\mu$  和数据方差  $\sigma$ , 然后对输入数据进行标准化操作, 训练缩放因子  $\gamma$  与偏置参数  $\beta$ , 经  $\gamma$ 、 $\beta$  线性变换后得到新的输出值  $y$ 。

$$\begin{aligned} \mu_B &\leftarrow \frac{1}{m} \sum_{i=1}^m x_i \\ \sigma_B^2 &\leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \\ x_i &\leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \\ y_i &\leftarrow \gamma x_i + \beta \equiv BN_{\gamma, \beta}(x_i) \end{aligned} \quad (1)$$

对输入数据进行标准化操作之后, 能够有效解决梯度消失与梯度爆炸的问题, 在模型训练时起能加速网络收敛, 并减少过拟合现象出现的可能性。但同时 BN 层会在网络前向推理时增加模型运算量, 影响检测性能, 占用更多存储空间, 导致检测速度变慢。本文将 BN 层的参数合并到卷积层, 进而减少模型运算量。合并后的权值参数  $\omega$ 、偏置  $\beta$  与缩放因子  $\gamma$ 、均值  $\mu$  呈正相关, 进而提升模型前向推断速度。

其中 YOLOv3 中卷积计算结果如公式 (2) 所示,  $x_i$ 、 $\omega_i$  分别为输入数据和权值参数, 输出为二者卷积之和。公式 (3) 中  $\mu$ 、 $\sigma$  分别为数据均值和方差,  $\gamma$ 、 $\beta$  分别为缩放因子和偏置参数,  $x_{out}$  即为 BN 计算结果。

$$x_{conv} = \sum_{i=0}^n (x_i * \omega_i) \quad (2)$$

$$x_{out} = \frac{\gamma(x_{conv} - \mu)}{\sqrt{\sigma^2 + 0.000001}} + \beta \quad (3)$$

将 BN 层参数与卷积层合并后的计算结果如公式 (4) 所示, 变量含义与上文相同。合并后权值  $\omega_i$  和偏置  $\beta$  如公式 (5) ~ (6) 所示。

$$\begin{aligned} x_{out} &= \sum_{i=0}^n \left( x_i * \frac{\gamma * \omega_i}{\sqrt{\sigma^2 + 0.000001}} \right) - \\ &\frac{\gamma * \mu}{\sqrt{\sigma^2 + 0.000001}} + \beta \end{aligned} \quad (4)$$

合并后权值  $\omega_i$  为:

$$\omega'_i = \frac{\gamma * \omega_i}{\sqrt{\sigma^2 + 0.000001}} \quad (5)$$

偏置  $\beta$  为:

$$\beta' = \beta - \frac{\gamma * \mu}{\sqrt{\sigma^2 + 0.000001}} \quad (6)$$

故合并后的计算结果为：

$$x_{out} = \sum_{i=0}^n (x_i * \omega'_i) + \beta' \quad (7)$$

卷积层与 BN 层合并后，可以共用 Blob 数据，进而降低内存的占用，提升速度。

2) 多尺度检测改进

将输入图像尺寸统一放缩到 416×416，特征提取网络对输入图像进行下采样 (subsampling) 操作，每次下采样对应得到一个尺度的特征图。具有较大分辨率的浅层特征图涵盖更多的位置相关信息，具有小分辨率的深层特征图涵盖更多的语义相关信息。将深层残差块采样得到的特征图与浅层残差块采样得到的特征图进行融合，可以增强浅层特征的语义信息，同时在细节信息最多的特征图上给待检测障碍物分配最精准的边界框，提高对小目标物体的检测性能。

如图 2 所示，本文在 YOLOv3 原有 3 个检测尺度的基础上进行扩展，添加一个尺度的特征图。经过 2 倍上采样后，特征尺度输出为 104×104，将第 11 层特征提取层与第 109 层 route 层、第 85 层与第 61 层，第 97 层与第 36 层进行融合，充分利用各层特征。特征图大小依次变为了 208、104、52、26、13；4 个尺度大小为 104×104、52×52、26×26 和 13×13。4 个特征尺度分别为：104×104，52×52，26×26 和 13×13。最后借鉴特征金字塔网络 (FPN, feature pyramid network) 结构进行特征融合，其中新增加的 104×104 特征图采用更精准的锚框 (anchor box)，提高算法对小目标的检测精度，同时缓解了在密集车辆和行人环境下待检测目标的错检和漏检问题。

网络首先依次对 4 个尺度 (13×13、26×26、52×52、104×104) 大小的特征图进行处理；然后将前 3 个尺度 (13×13、26×26、52×52) 的特征图上采样后与较大一个尺度的特征图拼接后分别送至检测大、中、小目标的 YOLO 层；最后与新增加的 104×104 的特征图进行拼接操作后送到剩余 YOLO 层，实现 4 个尺度的融合操作。由此可以获得一个强语义信息，并且在每个融合后的特征层上单独进行预测。

预测的输出张量中包括 3 部分内容，分别是障碍物边界框、障碍物目标和障碍物类别。改进后的 YOLOv3 将融合后生成的特征图细分为 S×S 的网格 (根据具体特征图包含 13×13、26×26、52×52、104×104)，每个网格需要预测 3 个不同的边界框，进而输出张量应当表示为 S×S× [3×(4+1+3)]，包含 4 个障碍物边框偏移参数，1 个障碍物目标和 3 个障碍物类型。

2.2 损失函数改进

交并比 (IOU, intersection over union) 是比较两个任意形状之间相似性最常用的指标，IOU 定义如公式 (8) 所示，其中 A、B 表示预测框与真实框。

$$IOU = \frac{|A \cap B|}{|A \cup B|} \quad (8)$$

Type	Filters	Size	Output
Convolutional	32	3×3	416×416
Convolutional	64	3×3/2	208×208
Convolutional	32	1×1	1×
Convolutional	64	3×3	
Residual			
Convolutional	128	3×3/2	104×104
Convolutional	64	1×1	2×
Convolutional	128	3×3	
Residual			
Convolutional	256	3×3/2	52×52
Convolutional	128	1×1	8×
Convolutional	256	3×3	
Residual			
Convolutional	512	3×3/2	26×26
Convolutional	256	1×1	8×
Convolutional	512	3×3	
Residual			
Convolutional	1024	3×3/2	13×13
Convolutional	512	1×1	4×
Convolutional	1024	3×3	
Residual			13×13

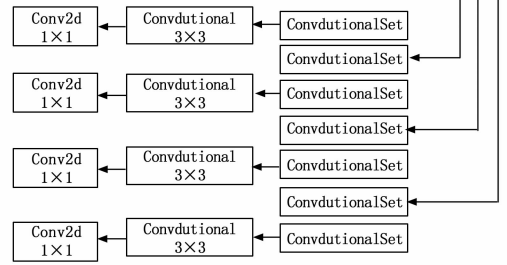


图 2 多尺度 YOLOv3 网络结构图

YOLOv3 原始算法采用 IOU 作为性能指标和边框损失函数。但物体重叠时 IOU 值为零，无法反映两物体之间的距离；物体不重叠且梯度为零时，无法进行参数优化。因此本文选用推广 IOU (generalized intersection over union, GIOU)<sup>[17]</sup> 代替 IOU。GIOU 与 IOU 定义一致，并且维持尺度不变性，在重叠情况下与 IOU 具备强相关性。GIOU 的计算公式如公式 (9) 所示，C 为可以将 A 和 B 包围在内的最小封闭形状。

$$GIOU = IOU - \frac{|C \setminus (A \cup B)|}{|C|} \quad (9)$$

在 YOLO 和 YOLOv2 的基础上，YOLOv3 将分类损失调整为二分交叉熵。如公式 (10) 所示，其损失函数依次由五部分组成：边框中心位置误差、边框宽度与高度误差、边框内包含检测目标时的置信度误差、边框内不包含检测目标时的置信度误差和目标分类误差。

$$Loss = \lambda_{coord} \sum_{i=0}^S \sum_{j=0}^B I_{ij}^{obj} [(x_i - \hat{x}_i^j)^2 + (y_i - \hat{y}_i^j)^2] + \lambda_{coord} \sum_{i=0}^S \sum_{j=0}^B I_{ij}^{obj} [(\sqrt{w_i^j} - \sqrt{\hat{w}_i^j})^2 + (\sqrt{h_i^j} - \sqrt{\hat{h}_i^j})^2] - \sum_{i=0}^S \sum_{j=0}^B I_{ij}^{obj} [\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j)] - \lambda_{noobj} \sum_{i=0}^S \sum_{j=0}^B I_{ij}^{noobj} [\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j)] - \sum_{i=0}^S \sum_{j=0}^B I_{ij}^{obj} \sum [ \hat{P}_i^j \log(P_i^j) + (1 - \hat{P}_i^j) \log(1 - P_i^j) ] \quad (10)$$

若值为 1，则第 i 个网格的第 j 个 anchor box 负责预测

该目标边界框。本文采用 GIOU 作为边框损失函数, 首先输入预测边界框与真实边界框的坐标值, 并将预测值进行排序, 然后分别计算、及二者交集的面积, 随后找出最小包围框  $I$ , 其中:

$$\begin{aligned} B^p &= (x_1^p, y_1^p, x_2^p, y_2^p) \\ B^g &= (x_1^g, y_1^g, x_2^g, y_2^g) \\ B^c &= (x_1^c, y_1^c, x_2^c, y_2^c) \end{aligned} \quad (11)$$

最小包围框的面积为:

$$A^c = (x_2^c - x_1^c) \times (y_2^c - y_1^c) \quad (12)$$

最后计算 GIOU、 $L_{GIOU}$ , 其计算公式如公式 (13) ~ (14) 所示。

$$IOU = \frac{I}{U} (U = A^p + A^g - I) \quad (13)$$

$$GIOU = IOU - \frac{A^c - U}{A^c} \quad (13)$$

$$L_{GIOU} = 1 - GIOU \quad (14)$$

### 2.3 聚类方法优化

YOLOv3 采用 anchor box 机制进行边界框预测, 并选用 k-means 聚类方法对 anchor box 进行初始化, 获得一组固定尺寸的候选框。特征图的尺度与 anchor box 的尺寸信息成反比, 较大尺度的特征图使用较小的 anchor box 进而获得小目标的检测信息, 较小尺度的特征图使用较大的 anchor box 进而获得小目标的检测信息。例如, YOLOv3 在 COCO 数据集上, 可得 9 种 anchor box 的尺寸信息如表 1 所示。

表 1 YOLOv3 的先验框尺寸

特征图尺度	anchor box 尺寸
13×13	(373×326)、(156×198)、(116×90)
26×26	(59×119)、(62×45)、(30×61)
52×52	(33×23)、(16×30)、(10×13)

考虑到 KITTI、PennFudanPed 等数据集中图像尺寸过大, 且 k-means 聚类方法在初始聚类点选择时具备过强的随机性和较大的时间复杂度, 本文采用 k-means++ 算法<sup>[18]</sup>对数据重新进行聚类, 获得更恰当的聚类中心。

k-means++ 的基本思想是使所有被选择的初始聚类中心之间的距离值尽量大。首先从输入的众多数据点中任意选择一个数据点作为首个聚类中心  $x$ , 然后针对数据集中的每一个点  $a$ , 计算其与最近聚类中心 (当前已选择的聚类中心) 的距离:  $d(a)$ ; 接着对聚类中心点重新进行选择 (依据是当前  $d(a)$  最大的点), 重复以上操作, 直至  $k$  个聚类中心点均被选出; 最后用选出的  $k$  个初始聚类中心运行原始 k-means 算法, 进而获得  $k$  个 anchor box 的尺寸信息。

考虑到本文数据集与公开数据集的差异, 原始参数会影响模型训练时间和准确性, 因此本文采用 k-means++ 算法对园区障碍物数据重新进行聚类分析, 针对设置的 4 种尺度, 基于本文数据集最终得到如表 2 所示的 12 种 anch

or box 大致尺寸信息。

表 2 k-means++ 聚类的先验框尺寸

特征图尺度	anchor box 尺寸
13×13	(399×265)、(346×265)、(329×265)
26×26	(241×316)、(356×206)、(281×212)
52×52	(212×273)、(177×324)、(174×220)
104×104	(145×155)、(77×110)、(34×57)

### 2.4 结合立体视觉的障碍物检测流程

针对封闭园区场景内无人车障碍物实时检测需求, 首先对立体视觉相机左侧图像使用改进 YOLOv3 网络模型进行障碍物目标检测, 得到其类别及预测框信息; 然后结合立体视觉相机获取左右两侧图像, 进行立体匹配处理, 通过深度计算获取障碍物与相机的距离信息, 最终实现障碍物的类别与位置检测。检测流程如图 3 所示。

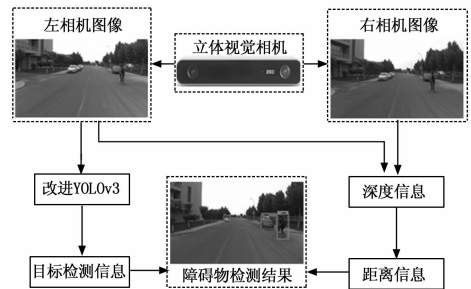


图 3 障碍物检测流程

## 3 实验

### 3.1 实验设置与数据集制作

本文实验选用 NVIDIA GTX1060 GPU, 操作系统为 Ubuntu 16.04, 软件环境为 CUDA9.0、Open CV3.4。ZED 双目立体视觉相机用于获取深度信息, 其基线距离为 120 mm, 帧率最高可达 100 fps (Frames Per Second, 每秒传输帧数)。

实验选用 KITTI 公开数据集中的校园场景图像数据, 包含车辆、行人等园区内主要类别障碍物, 由于该数据集中车辆类别占比较大, 导致首次训练模型平均精度均值 (mean Average Precision, mAP) 仅为 33%。因此, 实验增加复旦大学 PennFudanPed 行人数据集和自制园区数据集 HD-campus 并进行数据扩充。最终根据实际场景将园区内主要障碍物分为 Car、Pedestrian、Cyclist 三类, 共包含 11 050 张图像, 按照 6: 2: 2 的比例分为训练集、验证集和测试集, 将测试效果最好的模型用于实时场景检测。

### 3.2 模型训练

本文将训练集中图像统一转换为 416×416 大小输入网络, 训练时设置 32 张图像为一个批次, 其它参数设置如表 3 所示。

经过 20 000 次迭代后的损失函数变化情况如图 4 所示, 横坐标为迭代次数, 纵坐标为损失函数值。可以看出, 前

表 3 模型训练参数设置表

数据类型	设置数值
最大迭代次数	20 000
初始学习率	0.001
16 000 次后学习率	0.000 1
18 000 次后学习率	0.000 01
衰减系数	0.000 5
动量	0.9

2 000次迭代中平均损失下降速度较快, 2 000~10 000 次迭代平均损失缓慢下降, 10 000 次以后趋于稳定, 只有小幅振荡, 说明模型训练效果较好。

结果显示, 迭代 20 000 次的收敛效果最为稳定, 所以选取对应的权重文件作为障碍物检测实验的网络模型。

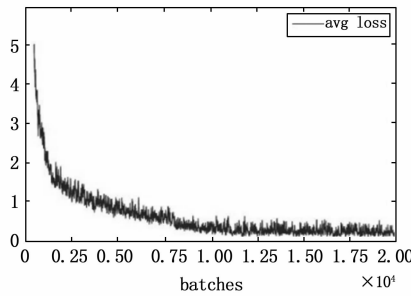


图 4 模型损失函数变化图

### 3.3 实验结果与分析

#### 1) 分步改进实验:

为了提升模型的检测速度, 本文将 BN 层参数合并到卷积层, 并随机选取测试集中的 10 张图片, 在其他实验设置相同情况下进行测试, 对比合并前后的单帧图像检测速度。如表 4 所示, 参数合并后, 在本文实验平台上单帧图像检测时间平均减少 25 ms, 录制视频数据检测帧率提升 50%, 且合并后检测准确程度不受影响。

表 4 单帧图像检测速度对比表

图像编号	合并前/ms	合并后/ms
000001	80.05	58.00
000002	89.68	58.02
000003	81.80	61.57
000004	81.30	58.92
000005	80.03	57.39
000006	81.07	56.60
000007	86.08	61.31
000008	81.84	59.31
000009	84.54	55.82
000010	84.20	54.88
Average	83.06	58.18

合并参数后, 为了降低较小障碍物的漏检率, 本文在 YOLOv3 的基础上增加了一个检测尺度。如表 5 所示, 四尺度的特征提取网络结构较之前具有更高的检测精度。其

中, Car 类提升效果最为明显, mAP 为 98.55%, 检测帧率达到 40 fps。其它数据类型也有明显改善, 平均检测精度提升 2.42%, 平均帧率提升 6 fps。数据表明, 增加至 4 尺度的 YOLOv3 模型针对每一类数据的特征提取更精准, 分类识别能力更强。

表 5 多尺度改进检测结果

数据类型	YOLOv3(3 尺度)		YOLOv3(4 尺度)	
	mAP/%	FPS/fps	mAP/%	FPS/fps
Car	96.75	34	98.55	40
Pedestrian	93.26	32	96.37	37
Cyclist	92.81	29	95.49	36
Average	94.38	31.70	96.80	37.67

如图 6 所示, 引入 GIOU 代替 IOU 作为边框损失函数后 mAP 值较改进前提升了 6.3%。其中, Car 类检测准确率最高, 而 Cyclist 类检测准确率提升效果最为明显。实验表明, 采用 GIOU 能有效提升模型检测精度与目标定位能力。

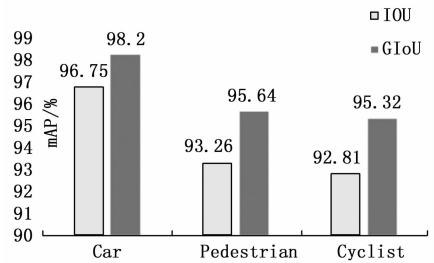


图 5 改进前后 mAP 对比图

本文将改进后的模型与双目测距算法结合, 使用 ZED 双目相机在校园 (封闭园区) 场景进行实时检测。改进后的模型检测平均帧率可达 40 fps, 满足实时要求。图 6 为实时检测效果截图, 图 (a)、(b) 和 (c) 中类别检测未出现错检、漏检情况, 模型目标检测效果较好。

本文将改进后的模型与双目测距算法结合, 使用 ZED 双目相机在校园 (封闭园区) 场景进行实时检测。

本文根据公式 (15) 计算测距误差  $e$ , 其中  $d_m$  和  $d_t$  分别为测量距离和真实距离。

$$e = \left| \frac{d_m - d_t}{d_t} \right| \times 100\% \quad (15)$$

表 6 为图 6 中障碍物测距实验数据, 参考相机参数及数据可知校园场景中 3~20 m 内障碍物测距检测结果较为准确, 图 (b) 与图 (c) 中距离较近或较远的障碍物测距误差稍大, 平均测距误差为 4.67%, 效果较好。

#### 2) 消融实验:

本文从特征网络改进、改进多尺度融合、定位损失函数、聚类方法四方面对 YOLOv3 进行改进, 为单独分析每一项对检测效果的影响, 设置消融实验。如表 7 所示, 设置五组实验, 确保其它实验设置一致的情况下, 从原始 YOLOv3 算法逐一添加改进项。实验 1 只对特征网络进行

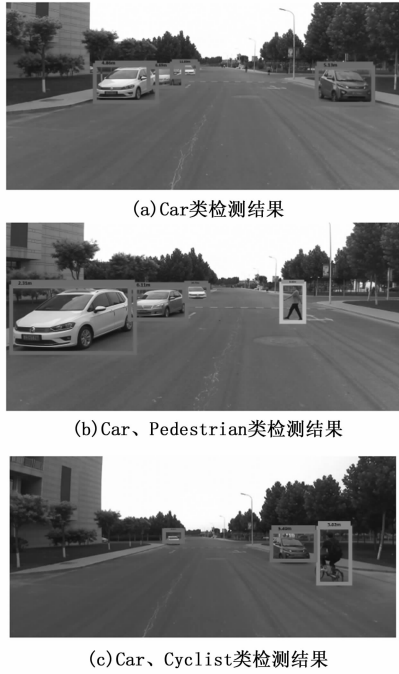


图 6 实时检测效果截图

改进, 实验 2 在实验 1 的基础上增加改进多尺度融合操作, 实验 3 在实验 2 的基础上采用 GIOU 作为边框损失函数, 最后一组即为本文改进后算法。

表 6 实际场景测距结果

图像编号	类别检测	实际类别	$dm/m$	$dt/m$	$e/\%$
(a)	Car	Car	4.86	4.97	2.21
	Car	Car	5.13	5.16	0.58
	Car	Car	6.69	6.52	2.61
	Car	Car	11.09	10.78	2.87
(b)	Car	Car	2.31	2.49	7.23
	Car	Car	6.11	6.53	6.43
	Pedestrian	Pedestrian	8.43	9.22	8.56
	Car	Car	14.73	13.79	6.82
(c)	Cyclist	Cyclist	3.02	3.16	4.43
	Car	Car	5.49	5.58	1.61
	Car	Car	18.03	19.60	8.01

表 7 消融实验结果

实验	特征网络改进	改进多尺度融合	GIOU	k-means++	mAP/%	FPS/fps
YOLOv3	×	×	×	×	94.38	31.7
实验 1	√	×	×	×	96.97	39.5
实验 2	√	√	×	×	97.28	38.7
实验 3	√	√	√	×	97.69	37.4
改进后	√	√	√	√	98.57	36.8

对比 YOLOv3 与实验 1 可得, 特征网络改进不仅提升了检测精度, 检测帧率也有明显改善; 对比实验 1、2、3 和本文改进算法实验, 可以发现改进多尺度融合、采用 GIOU

边框损失函数和优化聚类方法均能有效提升检测精度, 但会增加计算量, 使帧率稍有下降; 最终改进算法 mAP 达到 98.57%, 较原始算法提升 4.19%, 帧率提升 5.1 fps。由此可见, 本文的改进方案对园区障碍物检测具有实际意义。

3) 检测效果对比实验:

改进后模型检测效果如图 7 所示, 进行数据扩充后训练得到的权重文件更容易检测到 Pedestrian 和 Cyclist 类别。图 (a)、(b) 为 KITTI 园区数据集改进前后检测效果对比, 原模型未检测出的障碍物, 改进后模型均能有效检测 (图 (a)、图 (b) 白色框选部分); 图 (c) 选取了不同光照情况下的真实园区场景图像数据进行检测, 原始 YOLOv3 模型无法检测出障碍物, 改进后模型可准确检测, 对强光及阴影等环境适应性较好; 图 (d) 选取了多类型障碍物场景, 原始模型存在漏检现象, 改进后模型使障碍物漏检率降低, 且帧率仍满足实时检测要求, 可见本文算法可以提升园区环境中障碍物检测效果。

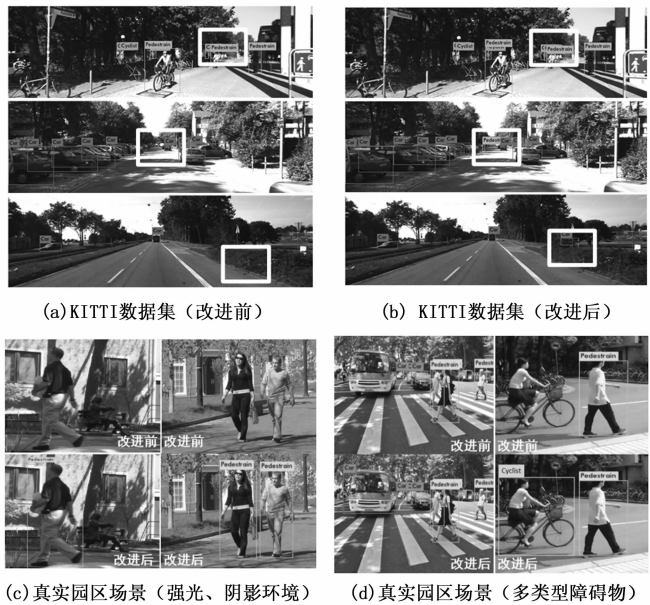


图 7 改进前后检测效果

将 SSD、Faster R-CNN、近两年改进算法、YOLOv3 以及本文选用算法检测结果进行比较分析。如表 8 所示, 与 YOLOv3 原始算法相比, 本文算法帧率增加了 8 fps, mAP

表 8 多种算法检测对比结果

算法	主干网络	FPS/fps	mAP/%
SSD	VGG16	17.86	91.57
Faster R-CNN	VGG16	6.53	96.23
文献[14]采用算法	Darknet-53	30.3	98.52
文献[15]采用算法	ResNet-101	32.25	97.20
YOLOv3	DarkNet-53	28.57	94.38
YOLOv4	CSPDarkNet-53	21.13	96.58
YOLOv5l	CSP1_3	25.52	93.62
本文改进后算法	DarkNet-53	35.71	98.57

提升了 4.19%，具有明显改进效果；而 SSD 与 Faster R—CNN 在检测精度或检测速度存在明显不足；与文献 [14—15] 的改进算法相比，本文算法的 mAP 与帧率也有一定提升；与 YOLO 的较新版本 YOLOv4 和 YOLOv5 (YOLOv5 中较为完善的版本 YOLOv5l) 相比，本文的算法能同时兼顾检测精度与速度，可以较好适应园区环境障碍物检测。

#### 4 结束语

针对园区无人车障碍物检测存在的问题，本文提出结合改进 YOLOv3 和立体视觉的检测方法。实验表明，改进后的模型单张图像检测速度和实时检测帧率均有提升，满足实时性要求；针对阴影及强光照等恶劣环境的漏检情况有较好改善；测距误差稳定维持在厘米级。可见，本文的方法能更好地适用于封闭园区无人车障碍物检测。

在接下来的工作中，将针对存在的问题进一步研究，丰富真实场景数据集和实验平台，继续改进相关算法。

#### 参考文献:

- [1] 金楠. 封闭园区自动驾驶横向控制系统设计与实现 [D]. 成都: 西南交通大学, 2019.
- [2] 李明熹, 林正奎, 曲毅. 计算机视觉下的车辆目标检测算法综述 [J]. 计算机工程与应用, 2019, 55 (24): 20—28.
- [3] 曹凯宁. 基于深度学习的车辆目标检测 [D]. 长春: 吉林大学, 2019.
- [4] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C] // Proceedings of the IEEE conference on computer vision and pattern recognition, 2014: 580—587.
- [5] GIRSHICK R. Fast R—CNN [C] // IEEE International Conference on Computer Vision, IEEE, 2015: 1440—1448.
- [6] 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.
- [7] 杨薇, 王洪元, 张继, 等. 一种基于 Faster—RCNN 的车辆实时检测改进算法 [J]. 南京大学学报 (自然科学), 2019, 55 (2): 231—237.
- [8] 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.
- [9] REDMON J, FARHADI A. YOLO9000: better, faster, stronger [C] // IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2017: 6517—6525.
- [10] REDMON J, FARHADI A. YOLOv3: An incremental improvement [C] // IEEE Conference on Computer Vision and Pattern Recognition, Washington D. C., USA: IEEE Press, 2018: 1—6.
- [11] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. YOLOv4: Optimal Speed and Accuracy of Object Detection [Z]. arxiv: Computer Vision and Pattern Recognition, 2020: 1—17.
- [12] 文生平, 李超贤. 基于 Gaussian—YOLOv3 的铝型材表面缺陷检测 [J]. 计算机测量与控制, 2020, 28 (9): 88—93.
- [13] 张兴旺, 刘小雄, 林传健, 等. 基于 Tiny—YOLOv3 的无人机地面目标跟踪算法设计 [J]. 计算机测量与控制, 2021, 29 (2): 76—81.
- [14] 叶佳林, 苏子毅, 马浩炎, 等. 改进 YOLOv3 的非机动车检测与识别方法 [J]. 计算机工程与应用, 2020, 12 (15): 1—9.
- [15] 景亮, 王瑞, 刘慧, 等. 基于双目相机的果园行人检测与定位 [J]. 农业机械学报, 2020, 8 (3): 1—8.
- [16] IOFFE S, SZEGEDY C. Batch normalization: Accelerating deep network training by reducing internal covariate shift [C] // International Conference on Machine Learning, PMLR, 2015: 448—456.
- [17] REZATOFIGHI H, TSOI N, GWAK J Y, et al. Generalized intersection over union: a metric and a loss for bounding box regression [C] // Proceedings of the IEEE International Conference on Computer Vision, 2019: 658—666.
- [18] ARTHUR, DAVID, VASSILVITSKII, et al. K—Means++: The Advantages of Careful Seeding [C] // Proceedings of the Eighteenth Annual ACM—SIAM Symposium on Discrete Algorithms, 2007: 1035—1057.
- [17] 江海军, 陈力. 激光扫描热波成像技术及在航空领域的应用 [J]. 红外技术, 2018, 40 (6): 110—115.
- [18] 王纪武, 鱼鹏飞, 罗海保, 等. 基于多项式拟合的激光线分段光顺处理方法的研究 [J]. 激光与光电子学进展, 2018, 55 (9): 216—222.
- [19] 马龙, 钱瑞婕, 刘宇哲, 等. 航空发动机叶片 HSI 模型三维快速测量方法 [J]. 激光与光电子学进展, 2020, 57 (19): 144—153.
- [20] 刘洁, 杨晓菊, 廖宜涛, 等. 基于线结构激光传感器的土壤表面粗糙度测量方法研究 [J]. 农业工程学报, 2019, 35 (3): 134—140.

(上接第 53 页)

- [13] 张程, 刘晓光, 倪继锋. 基于 ZYNQ 的航空发动机传感器仿真系统设计 [J]. 计算机测量与控制, 2019, 27 (4): 230—234.
- [14] 李晖, 常永乐, 张体南, 等. 基于激光扫描法的纤维增强复合薄板损伤定位研究 [J]. 机械工程学报, 2019, 55 (18): 24—30.
- [15] 马龙, 徐泓悦, 胡艳敏, 等. 基于投影条纹级次识别的发动机叶片三维测量方法 [J]. 激光与光电子学进展, 2018, 55 (4): 217—228.
- [16] 郑震, 查冰婷, 张合. 基于 DHGF 算法的激光扫描成像引信目标识别方法 [J]. 中国激光, 2018, 45 (7): 147—154.