

# 一种改进 YOLO v3 的货车车轮检测算法\*

周睿璇, 汪俊霖, 孙宏, 何秋生

(太原科技大学, 山西 太原 030024)

**摘要:** 针对 YOLO v3 算法在检测货车车轮中准确度不高的问题, 提出一种改进 YOLO v3 算法的货车车轮检测方法。利用 K-means 算法对数据集进行聚类, 实现先验框的自定义, 从而使之符合特定物体的尺寸; 采用 Loss<sub>CIOU</sub> 和 Mish 激活函数提高精度并应用在货车车轮识别中进行验证。实验结果表明: 改进后的 mAP 提高了 6.67%, 漏检率降低了 36.34%。

**关键词:** YOLOv3; 货车车轮识别; K-means; Loss<sub>CIOU</sub>; Mish

**中图分类号:** TP391.4 **文献标识码:** A

## 0 引言

传统识别货车超重的方法主要是检测人员用眼睛观察, 根据经验判断是否超重, 但是这样误差大。现在可以用视觉识别的方法来解决这一问题。根据国家货车车型对应标准(GB1589), 货车车轴的数量以及分布与货车的型号相关联, 可以通过检测货车车轴的数量和分布分辨货车车型判断其核载重量, 再结合货车实际载重量判断是否超重。YOLO 系列算法是比较成熟的神经网络算法, 但也有不足之处, 比如在检测特定的物体时算法中原先设定的先验框有的就不适用了, 本项目以识别货车的车轮为例, 对 YOLO v3 算法进行改进。

## 1 YOLO V3 原理

### 1.1 检测过程

经典的 YOLO 算法中输入的图片经过 Darknet-53 输出三种尺度特征, 分别用于检测小、中、大的物体, 每个特征层有 3 个先验框。每个网格点预测出与所在层三个先验框对应的三个预测框, 每个预测框包含 C+5 个信息, 其中 C 是物体的类别数, 5

包括中心点的偏移量  $t_x$  和  $t_y$ 、长宽  $t_h$  和  $t_w$ 、置信度<sup>[1]</sup>。这时的  $t_x, t_y, t_h, t_w$  并不能直接反应预测框的空间信息, 还需要经过下面公式获取预测框的长宽  $b_h, b_w$  和中心位置  $b_x, b_y$ 。

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= a_t^{t_w} \\ b_h &= a_h e^{t_h} \end{aligned} \quad (1)$$

其中:  $c_x, c_y$  是网格点在网格中的坐标;  $a_h$  和  $a_w$  是先验框的长和宽<sup>[2]</sup>。iou 是预测框和真实框的交集和并集的比, 可以反映出两个框的重合程度, 计算每个预测框和真实框的 iou, 与真实框 iou 值最大的预测框为最终的检测结果。

### 1.2 损失函数

损失函数可以作为深度神经网络对误检测样本惩罚的根据, YOLO v3 的损失函数采用二交叉熵的形式<sup>[3]</sup>。损失包括坐标损失、置信度的损失和类的损失, 其中坐标损失包括中心点的损失、宽高的损失。损失函数公式如下:

$$\begin{aligned} loss = & \sum_{i=0}^{K_x K} \sum_{j=0}^M I_{ij}^{obj} (2 - w_i^j * h_i^j) (\hat{x}_i^j * \log(x_i^j) + (1 - \hat{x}_i^j) * \log(1 - x_i^j)) + \sum_{i=0}^{K_x K} \sum_{j=0}^M I_{ij}^{obj} (2 - w_i^j * h_i^j) \\ & (\hat{y}_i^j * \log(y_i^j) + (1 - \hat{y}_i^j) * \log(1 - y_i^j)) + \sum_{i=0}^{K_x K} \sum_{j=0}^M I_{ij}^{obj} (2 - w_i^j * h_i^j) * 0.5 * [(w_i^j - \hat{w}_i^j)^2 + \end{aligned}$$

收稿日期: 2021-03-21 修回日期: 2021-04-20

基金项目: 山西省研究生教育改革研究课题(2018JG73); 太原科技大学创新创业计划训练项目(XJ2020033); 国家级大学生创新创业计划训练项目(202010109012); 山西省研究生教育创新项目(2020SY421)

作者简介: 周睿璇(2000-), 女, 江西赣州人, 本科, 研究方向: 自动化。

$$\begin{aligned}
 & (h_i^j - \hat{h}_i^j)^2] - \sum_{i=0}^{K_x K} \sum_{j=0}^M I_{ij}^{obj} [\hat{C}_i^j \log(C_i^j) + (1 - \hat{C}_i^j) \log(1 - C_i^j)] - \sum_{i=0}^{K_x K} \sum_{j=0}^M I_{ij}^{noobj} [\hat{C}_i^j \log(C_i^j) + \\
 & (1 - \hat{C}_i^j) \log(1 - C_i^j)] - \sum_{i=0}^{K_x K} \sum_{j=0}^M I_{ij}^{obj} \sum_{c \in classes} [\hat{P}_{i,c}^j \log(P_{i,c}^j) + (1 - \hat{P}_{i,c}^j) \log(1 - P_{i,c}^j)] \quad (2)
 \end{aligned}$$

其中:  $\hat{x}_i^j, \hat{y}_i^j, \hat{w}_i^j, \hat{h}_i^j$  是真实框经过变换后的新的中心点位置和长宽;  $x_i^j, y_i^j, w_i^j, h_i^j$  是预测框的中心点位置和长宽;  $I_{ij}^{obj}$  为第  $(i, j)$  个网格是否检测目标, 是为 1, 不是为 0;  $C_i^j$  为某个预选框含有目标物体的得分;  $\hat{C}_i^j$  为第  $(i, j)$  个网格是否有目标物体, 是为 1, 不是为 0;  $\hat{P}_{i,c}^j$  是第  $(i, j)$  个网格里的物体是否属于类别  $c$ , 是为 1, 不是为 0;  $P_{i,c}^j$  是第  $(i, j)$  个预测框属于类别  $c$  的概率<sup>[4]</sup>。真实框的中心点位置和长宽需要经过如下公式解码再代入公式:

$$\begin{aligned}
 \hat{t}_x &= g_x - c_x \\
 \hat{t}_y &= g_y - c_y \\
 \hat{t}_w &= \log(g_w / a_w) \\
 \hat{t}_h &= \log(g_h / a_h) \quad (3)
 \end{aligned}$$

其中:  $g_x, g_y, g_w, g_h$  是真实框的中心点的偏移量和长宽;  $c_x, c_y$  是真实框所在网格点的坐标。

坐标损失函数的不足之处在于: 中心点位置并不是 0 或 1 而是 0 和 1 之间的某个数, 但当预测结果与实际结果相同时坐标损失并不为 0。

### 1.3 激活函数

激活函数能加入非线性因素, 提高神经网络对模型的表达能力, YOLO v3 采用的是 Leaky ReLU 激活函数, 其公式为:

$$\text{Leaky Relu}(x) = \begin{cases} x, & x > 0 \\ leak * x, & x \leq 0 \end{cases} \quad (4)$$

其中  $leak$  是一个很小的正数。

Leaky ReLU 激活函数的特点是在输入为负值时只有很小的坡度且并不光滑, 可能导致基于梯度

$$dis = 1 - iou = 1 - \frac{(x_m - x_n)(y_m - y_n)}{(x_r^g - x_l^g)(y_b^g - y_t^g) + (x_r^a - x_l^a)(y_b^a - y_t^a) - (x_m - x_n)(y_m - y_n)} \quad (5)$$

其中:  $(x_r^g, y_b^g)$  是真实框右下角的横纵坐标;  $(x_r^a, y_t^a)$  是真实框左上角的横纵坐标;  $(x_r^g, y_b^g)$  是先验框右下角的横纵坐标;  $(x_l^a, y_t^a)$  是先验框左上角的横纵坐标;  $x_m$  为  $x_r^g$  和  $x_r^a$  中最小的;  $x_n$  是  $x_l^g$  和  $x_l^a$  中最大的;  $y_m$  是  $y_b^g$  和  $y_b^a$  中最小的;  $y_n$  是  $y_t^g$  和  $y_t^a$  中最大的。 $dis$  越小, 相似程度越高。先随机选取真实框中的  $k$  个做为初始的先验框, 把距离某个中心值差值最小的框分在一起, 从小到大排列, 分成  $k$  个类  $C_i = \{[a_{wi1}, a_{hi1}], [a_{wi2}, a_{hi2}] \dots, [a_{win}, a_{hin}]\} i = \{1, 2, 3, 4, 5 \dots k\}$ , 通过如下公式计算新的先验框:

的学习很慢, 使得检测的效率降低。

## 2 改进算法

### 2.1 优化先验框参数

传统的 YOLO v3 算法的先验框尺寸是固定的, 可能导致先验框与数据集中物体的大小并不匹配, 从而影响检测精度。以货车车轮的检测为例, 数据集中车轮的体积偏小, 如果先验框设定过大可能导致预测框与真实框的  $iou$  的太小从而造成物体的漏检。图 1 是先验框和数据集中真实框尺寸的分布图。

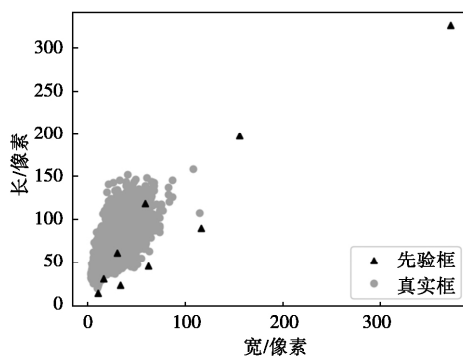


图 1 改前先验框和真实框尺寸的分布

从图 1 可以看出有的先验框并不在真实框的集中范围内, 先验框并不能代表数据集中车轮的总体特征。所以对先验框的尺寸进行改进。根据实际场景中货车车轮特征, 本文采用 K-means 算法确定代表样本特征的几个先验框。

K-means 聚类能够在的一组数中找到具有相似特征的数并把它们分在一起从而得到聚类的中心点, 这些中心点能代表这些数的总体情况。本文用  $iou$  来衡量两个框的相似程度, 其度量公式如下:

$$\begin{aligned}
 a_w^* &= \begin{cases} \frac{a_{wi\frac{n}{2}} + a_{wi(\frac{n}{2}+1)}}{2} & n = 2, 4, 6 \dots \\ \frac{a_{wi\frac{n+1}{2}}}{2} & n = 1, 3, 5 \dots \end{cases} \\
 a_h^* &= \begin{cases} \frac{a_{hi\frac{n}{2}} + a_{hi(\frac{n}{2}+1)}}{2} & n = 2, 4, 6 \dots \\ \frac{a_{hi\frac{n+1}{2}}}{2} & n = 1, 3, 5 \dots \end{cases} \quad (6)
 \end{aligned}$$

重复操作, 直到新的先验框不变。预选框的生成和真实框的解码受先验框变化的影响, 公式(1)

变为:

$$\begin{aligned} b_x &= \sigma(t_w) + c_x \\ b_y &= \sigma(t_h) + c_y \\ b_w &= a_w^* e^{t_w} \\ b_h &= a_h^* e^{t_h} \end{aligned} \quad (7)$$

公式(3)变为:

$$\begin{aligned} \hat{t}_x &= g_x - c_x \\ \hat{t}_y &= g_y - c_y \\ \hat{t}_w &= \log(g_w / a_w^*) \\ \hat{t}_h &= \log(g_h / a_h^*) \end{aligned} \quad (8)$$

## 2.2 改进损失函数

传统的 YOLO v3 采用的二交叉熵并不适用于中心点偏移量处于 0 到 1 之间的情况,考虑到本文检测对象的特殊性,将坐标损失函数替换为  $Loss_{CIOU}$ 。 $Loss_{CIOU}$  采用的是预测框和真实框的中心点的欧式距离与包含预测框和真实框的最小闭包区域的对角线距离的平方之比来衡量中心点的误差,为保证真实框和先验框长宽比的一致性, $Loss_{CIOU}$  定义  $\alpha$  和  $v$  两个参数以衡量长宽的损失,还把两个框没有重叠的部分纳入考量范围。 $Loss_{CIOU}$  的公式如下:

$$Loss_{CIOU} = 1 - iou + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha v \quad (9)$$

其中:  $\rho^2(b, b^{gt})$  为预测框和真实框的中心点的欧式距离;  $c$  是能够同时包含预测框和真实框的最小闭包区域的对角线距离;  $\alpha$  和  $v$  的公式如下:

$$\begin{aligned} \alpha &= \frac{v}{1 - iou + v} \\ v &= \frac{4}{\pi} \left( \arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \end{aligned} \quad (10)$$

其中:  $w^{gt}$ 、 $h^{gt}$  是真实框的宽和长;  $w$ 、 $h$  是预测框的宽和长。

## 2.3 更改激活函数

由于 Leaky ReLU 是单调、不光滑的函数,在检测过程中容易导致梯度学习缓慢。Mish 是一种光滑、非单调的函数。非单调性有助于负值的时候允许较小的负梯度流入, Mish 函数不仅能加快学习速率还能得到更好的准确性和泛化能力。Mish 函数公式为:

$$Mish(x) = x \times \tanh(\ln(1 + e^x)) \quad (11)$$

## 3 实验结果及分析

### 3.1 实验环境及实验结果

实验对本文提出的改进后的算法与原 YOLO v3

算法进行对比分析。配置有英特尔 E5 - 2660 v CPU、三星 64G 内存和三星 500G 固态硬盘,还搭建了 Keras、python、OpenCV - python 等常用环境。实验采用 1140 张货车图片作为测试集,检测效果如图 2 和图 3 所示。



图2 改进前检测效果



图3 改进后检测效果

从图 2 和图 3 可知:改进前检测出的三个预测框的平均得分为 0.847,更改后的三个预测框平均得分为 0.94,可见更改后的检测效果更好。

### 3.2 结果分析

对于改进前后的检测效果从模型的收敛速度,先验框与真实框的匹配程度,检测精度这几个角度进行分析。

1) 从模型的收敛速度看,图 4 可以看出更改后的 Loss 值比更改前的 Loss 值小,说明模型收敛地更快,这是由于先验框相当于把统计上的先验经验加入到模型中,对预测的对象范围进行约束,有助于模型快速收敛,  $Loss_{CIOU}$  使得目标框回归变得更加稳定, Mish 函数使得权重能继续更新。

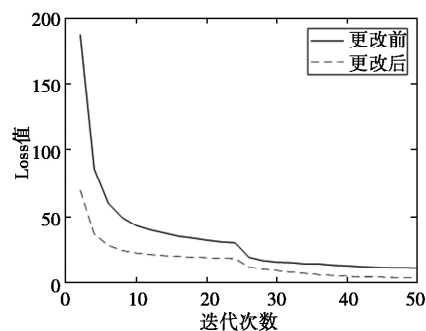


图4 Loss 曲线图

2) 从先验框与真实框的匹配程度看,通过 K - means 聚类得到与车轮大小特征相符的三个先验框 分别是  $[39, 97]$ ,  $[53, 102]$ ,  $[39, 67]$  将这三个框代 替原来第二层的三个框,由图 5 可以看出更改后的三 个先验框在真实框主要集中的范围内均匀分布,与图 3 作比较可以看出更改后的框更适合识别货车车轮。

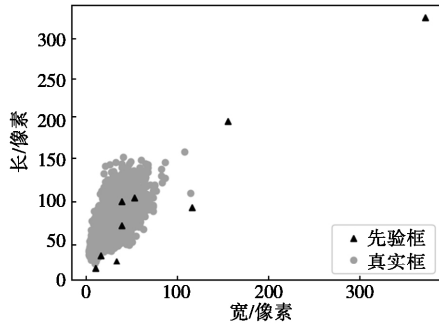


图 5 更改后先验框和真实框分布

3) 从检测精度看,图 6 中更改后的 PR 曲线在 更改前的上方,即更改后 mAP 的结果会更高,最后 表 1 的检测结果也表明更改后的 mAP 提高到了 90.16%,漏检率降低到了 9.23%,说明更改后的检 测结果更精确。

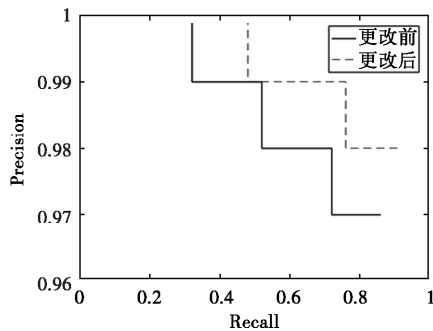


图 6 PR 曲线

表 1 检测结果对比

| 网络          | 漏检率    | mAP    |
|-------------|--------|--------|
| 原 YOLO v3   | 14.50% | 84.52% |
| 改进后 YOLO v3 | 9.23%  | 90.16% |

## 4 结语

本文提出了一种改进 YOLO v3 算法的货车车轮 检测方法。通过 K - means 聚类,采用更好的位置损 失函数和激活函数的方法改进 YOLO v3 网络。在 1140 张测试集上进行验证,结果表明改进后模型的收 敛速度更快、准确度提升、漏检情况也有所改善,说明 改进后的 YOLO v3 算法更适合检测货车车轮。

## 参考文献

- [1] 郭进祥,刘立波,徐峰,等. 基于 YOLO v3 的 机场场面飞机检测方法[J]. 激光与光电子学 进展, 2019( 19): 111 - 119.
- [2] 张富凯,杨峰,李策. 基于改进 YOLO v3 的快 速车辆检测方法[J]. 计算机工程与应用, 2019( 2): 12 - 20.
- [3] 王兵,李文璟,唐欢. 改进 YOLO v3 算法及其 在安全帽检测中的应用[J]. 计算机工程与应 用, 2020( 9): 33 - 40.
- [4] 崔文靓,王玉静,康守强,等. 基于改进 YOLO v3 算法的公路车道线检测方法[J/OL]. 自动 化学报, [2021 - 02 - 23]. <http://kns.cnki.net/kcms/detail/11, 2019. TP. 20190605. 1051.001.html>.

## An Improved YOLO v3 Algorithm for Truckwheel Detection

Zhou Ruixuan , Wang Junlin , Sun Hong , He Qiusheng

( Taiyuan University of Science and Technology , Taiyuan Shanxi 030024 , China)

**Abstract:** Aiming at the problem of low accuracy of YOLO v3 algorithm in the detection of truck wheel , an im- proved YOLO v3 algorithm for truck wheel detection is proposed. Using the K - means algorithm , the data set is clustered to realize the customization of the prior bounding box , so as to make it conform to the size of the specific object.  $Loss_{CIoU}$  and Mish are used to activate the teaching function to improve the accuracy and applied it in the wheel recognition for verification. The experimental results show that the improved mAP is increased by 6.67% and the missed detection rate decreased by 36.34% .

**Key words:** YOLO v3; truck wheel identification; K - meams;  $Loss_{CIoU}$ ; Mish