

基于改进 YOLOv3 的安全帽佩戴识别算法

许凯, 邓超*

河南理工大学物理与电子信息学院, 河南 焦作 454003

摘要 针对传统安全帽佩戴识别算法检测精度低、鲁棒性差的问题, 提出了一种基于深度学习的安全帽佩戴检测方法。该方法以 YOLOv3 检测算法为基础, 对其网络结构和损失函数加以改进。首先, 通过增加特征图弥补原 YOLOv3 算法对小目标检测效果不佳的问题; 然后在增加特征图的基础上, 使用 K -means 聚类算法对收集的安全帽数据集进行聚类, 选择出合适的先验锚框; 最后, 采用 GIoU Loss 作为边界框损失, 在损失函数中加入 Focal Loss, 减少正负样本不均衡带来的误差。实验结果表明, 相较于 YOLOv3 检测算法, 改进后的算法在平均精确率上提高了 3.47%, 在安全帽识别精确率上提高了 4.23%, 在安全帽识别上具有一定的先进性和有效性。

关键词 机器视觉; 目标检测; YOLOv3; 安全帽检测; K -means 算法

中图分类号 TP391.4

文献标志码 A

doi: 10.3788/LOP202158.0615002

Research on Helmet Wear Identification Based on Improved YOLOv3

Xu Kai, Deng Chao*

School of Physics & Electronic Information Engineering, Henan Polytechnic University, Jiaozuo, Henan 454003, China

Abstract To address the problem of low detection accuracy and poor robustness of the traditional helmet wearing recognition algorithm, we propose a deep learning-based helmet wearing detection method. The method is based on the YOLOv3 detection algorithm, and its network structure and loss function are improved. Firstly, the poor detection of small targets by the original YOLOv3 algorithm is compensated by adding feature maps. Then, on the basis of adding feature maps, the K -means clustering algorithm is used to cluster the collected helmet datasets and select the appropriate *a priori* anchor frames. Finally, GIoU Loss is adopted as the boundary frame loss, and Focal Loss is added to the loss function to reduce the errors due to positive and negative sample imbalance. Experimental results show that, compared with the YOLOv3 detection algorithm, the improved algorithm improves the average accuracy by 3.47% and the accuracy of helmet identification by 4.23%, which is advanced and effective in helmet identification.

Key words machine vision; target detection; YOLOv3; helmet detection; K -means algorithm

OCIS codes 150.0155; 330.1880; 150.1135; 150.3040

1 引言

在建筑工地的安全生产规范中, 不允许未佩戴安全帽进入施工现场。佩戴安全帽可以有效降低安全事故发生后因坠落、物体坠击对头部造成的损伤, 最大限度地保护施工人员的生命安全。早期的安全

帽佩戴检测主要是通过人工巡检和视频监控系统等方式, 人工巡检效率低、成本高, 且存在漏检情况, 视频监控依靠人力在海量的监控视频中检索异常信息, 难度较大, 且难以长时间、不间断地发现违规作业并及时制止。为了降低施工人员不按规定佩戴安全帽而造成的事故发生率, 有必要对施工人员是否

收稿日期: 2020-07-24; 修回日期: 2020-08-11; 录用日期: 2020-08-25

基金项目: 河南省重点研发与推广专项(科技攻关)(202102310560)、淮阴工学院重点实验室开放课题(JSLERS-2018-005)、河南省高校基本科研业务费专项资金资助(NSFRF180411)

* E-mail: super@hpu.edu.cn

戴安全帽进行实时检测。

传统的基于视觉的安全帽检测算法通过手工设计的特征对人员上半身进行特征提取,再对提取的特征进行分类判断是否佩戴安全帽。如刘晓慧等^[1]采用肤色检测的方法定位人脸区域,再利用人脸与安全帽位置关系对安全帽进行识别;蔡利梅^[2]则利用安全帽外部特性的方向信息建立安全帽模型,用来解决复杂环境下矿井视频的安全帽识别;Rubaiyat 等^[3]首先将结合图像的频域信息与基于梯度方向直方图(HOG)特征的人体检测算法用于工人的检测,然后使用基于颜色特征和圆形霍夫变换的方法来检测工人安全帽的佩戴情况。这些基于传统的手工特征设计的方法,在实际的安全帽佩戴检测应用中难以保证其识别率和实时性。

近年来随着深度学习的发展,许多学者也提出了一系列基于卷积神经网络(CNN)的目标检测算法。Girshick 等^[4]提出区域卷积神经网络(RCNN)及后面的改进版本 Fast RCNN^[5]和 Faster RCNN^[6];Redmon 等^[7-9]提出 YOLO(You Only Look Once)系列算法;Liu 等^[10]提出 SSD(Single Shot MultiBox Detector)算法。在诸多算法中 YOLOv3 在检测速度上领先于其他算法,取得了较好的检测效果。但随着交并比(IoU)的增加,YOLOv3 算法检测性能下降,为此,诸多学者提出一系列改进措施。鞠默然等^[11]提出小目标检测算法,李汉冰等^[12]提出车辆检测算法,Zhang 等^[13]提出行人目标检测算法,这些针对特定场景优化后的 YOLOv3 算法都有不错的检测效果。

本文选用 YOLOv3 模型为基础,针对施工现场人员佩戴安全帽检测任务中存在的密集目标、小尺度目标等问题,对 YOLOv3 算法进行了改进和优化。首先通过增加尺度层提高模型检测小目标的能力,其次重新聚类了先验锚框,将广义交并比(GIoU)损失作为边界框损失,并将 Focal Loss 加入损失函数中,最后进行训练得到安全帽佩戴情况检测模型。

2 YOLOv3 网络结构

YOLOv3 目标检测算法将检测问题视为回归问题,算法将输入的一张图像划分为 $s \times s$ 的网格,如果一个检测对象的中心点在某个网格中,则该网格负责检测这个对象,每个网格输出 B 个预测框,包含预测框的位置信息(包括中心点坐标 x 和 y 、宽度 w 、高度 h)和置信度。根据置信度的大小,模型将置信度低于阈值的预测框置为 0,最后采用非极大值抑

制(NMS)算法去除重复的边界框并输出预测结果。

YOLOv3 模型如图 1 所示,使用 DarkNet-53 网络作为特征提取网络。同时,为了提高模型检测小目标的能力,还采用了类似特征金字塔网络(FPN)的方法,将上采样^[15]后的深层语义特征与浅层的位置特征进行融合^[16]。YOLOv3 分别从特征图 y_1 、 y_2 、 y_3 上提取特征进行检测,实现了对大、中、小尺度目标的预测。

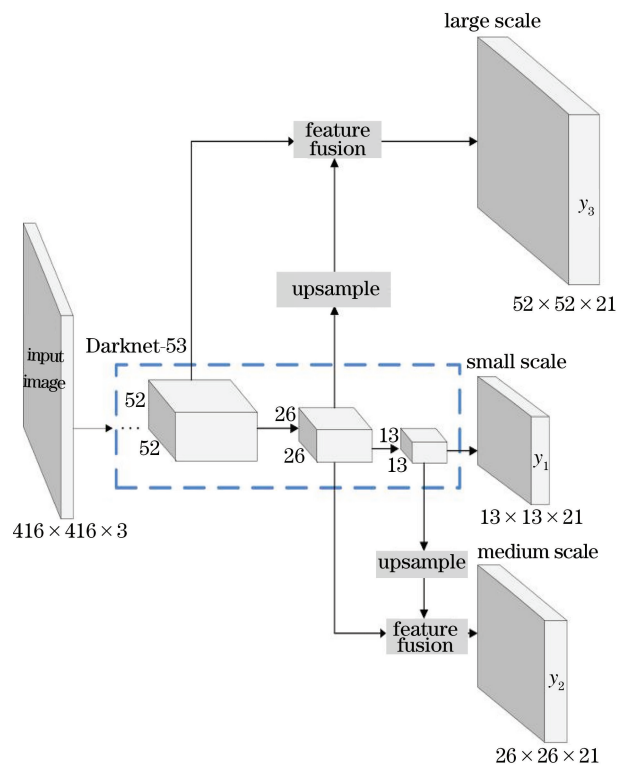


图 1 YOLOv3 模型结构图

Fig. 1 YOLOv3 model structure

但直接将 YOLOv3 应用于施工现场人员安全帽佩戴检测任务仍有不足:1)YOLOv3 虽然提高了对小目标的检测精度,但仍存在浅层特征提取不充分的问题;2)YOLOv3 预测过程的准确性依赖于 IoU,但随着 IoU 增大,预测精度会有所下降;3)针对施工场景下存在的密集目标及尺度变化等问题,YOLOv3 仍有很大提升空间。针对上述问题,本文对 YOLOv3 加以改进和优化。

3 改进的 YOLOv3 安全帽识别算法

安全帽识别问题可以简单看成一个二分类问题,即把佩戴安全帽的人看作是一类目标,未佩戴安全帽的人视作另一类目标,整个算法流程如图 2 所示。为了适用于安全帽佩戴检测任务,本文针对 YOLOv3 算法主要进行以下改进工作:1)增加融合

的尺度层,提高检测密集小目标的能力;2)使用 K-means 算法聚类出更适合的先验锚框;3)使用

GIoU Loss 代替边界框损失,在损失函数中使用 Focal Loss,解决正负样本分布不均衡的问题。

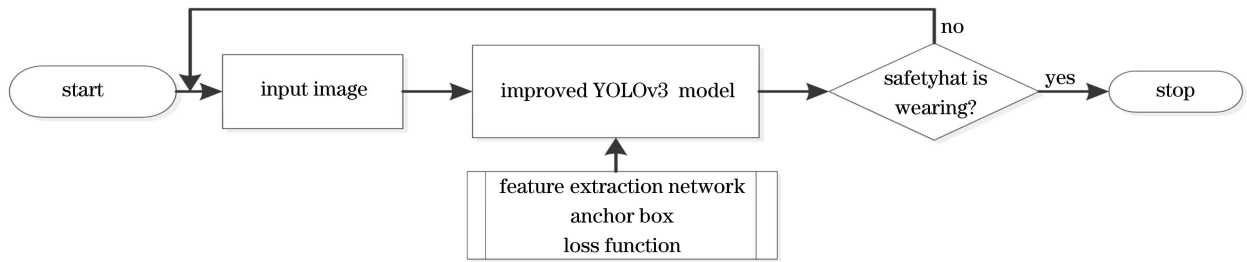


图 2 安全帽佩戴识别流程图

Fig. 2 Safety helmet wear identification process

3.1 特征提取网络的改进

YOLOv3 算法使用的最小尺度特征图尺寸为 13×13 ,较小的特征图包含较多的语义信息,但对位置信息预测较粗劣。对于要识别的安全帽目标,由于拍摄距离远近的不同和目标自身尺寸的原因,存在较小的安全帽目标,YOLOv3 算法对此类小目标检测效果不好,会发生误检、漏检等现象。为进一步提升对小目标的检测能力,在原有的 3 个尺度上增加 1 个尺度,即在原输出特征图的基础上增加 1 个特征图,并与其他特征图进行融合,从 4 个尺度上对安全帽目标进行检测。

具体操作是将 Darknet-53 网络的 52×52 尺寸

特征图 y_3 的输入进行 2 倍上采样操作,并与网络第 11 层 104×104 的特征图进行连接,再经过系列卷积操作进行特征融合,输出特征图 y_4 。由于要检测的类别只有 2 个,故输出的尺度维度为 $13 \times 13 \times 21$ 、 $26 \times 26 \times 21$ 、 $52 \times 52 \times 21$ 、 $104 \times 104 \times 21$,减少了网络参数运算量,加快了运算速度。 104×104 尺寸特征图 y_4 是输入图像的 4 倍下采样,其感受野较小,适合检测较小的安全帽目标。在 4 个尺度上训练出的安全帽佩戴检测模型,即使画面中目标大小发生变化,也能在多个尺度上准确检测。增加尺度后的模型如图 3 所示,为方便叙述,记为 YOLOv3-4l 模型。

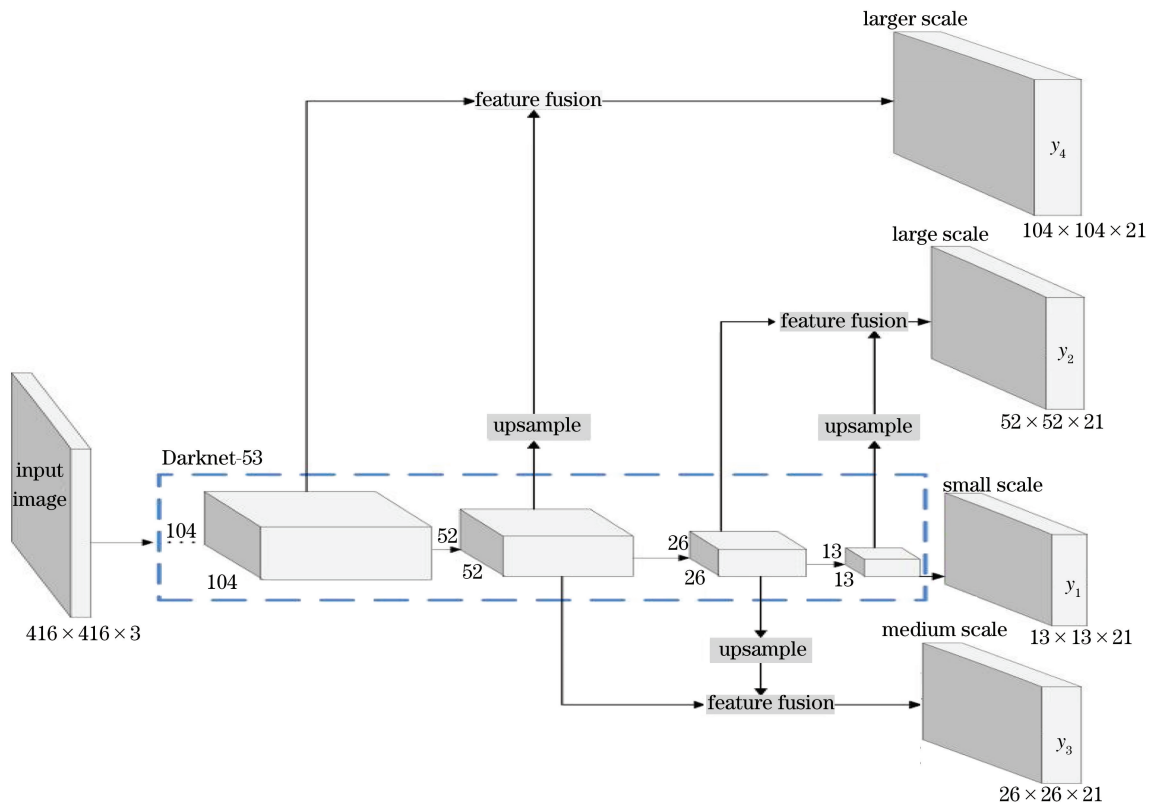


图 3 YOLOv3-4l 模型图

Fig. 3 Diagram of YOLOv3-4l model

3.2 初始候选框的优化

在 YOLO 系列之前的目标检测算法都是采用手工挑选的候选框尺寸,对不同数据的应用带有较强的主观性。YOLOv3 算法延续了 v2 版本使用先验框 anchor 来预测边界框坐标的方法,anchor 是针对不同尺度网络层而确定的初始候选框。

YOLO 算法中 anchor 值是用 K-means 聚类方法计算得出的。K-means 算法通过随机选取 K 个对象作为初始聚类中心,然后计算每个对象与聚类中心的距离,再按照一定规则将其分到该聚类中心下,重复迭代直至所有对象分类完成。重新设计的 YOLOv3-4l 相较于 YOLOv3 多了一个尺度,因此原有的 3×3 的锚框已不适用,需要重新聚类。经过多次实验,权衡平均交并比 (Avg IoU) 与聚类中心 K 个数,本文的聚类中心数 K 设为 12,实验过程如图 4 所示。

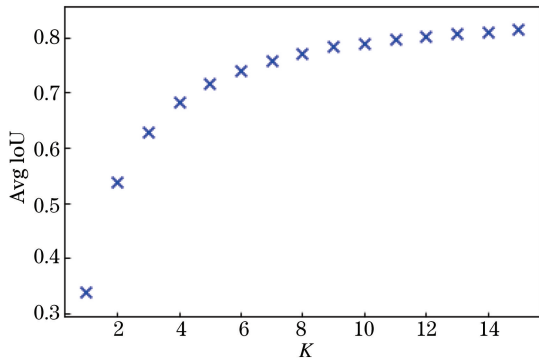


图 4 K-means 聚类分析结果图

Fig. 4 K-means cluster analysis results

聚类后候选框分布如表 1 所示。增加的 104×104 尺寸特征图主要侧重于对小尺寸目标进行检测,而 52×52、26×26 和 13×13 尺寸特征图则侧重于对中等尺寸以上目标进行检测。通过分配大小不同的候选框,进一步增强了网络对不同尺寸安全帽的检测能力。

表 1 聚类后先验候选框分布

Table 1 Distribution of a priori candidate box after clustering

Feature map size	Candidate box size		
104×104	8, 17	11, 23	16, 31
52×52	24, 41	34, 56	46, 80
26×26	66, 105	82, 152	150, 157
13×13	112, 225	185, 304	335, 427

3.3 损失函数的改进

YOLOv3 在边界框回归过程中采用的是均方差误差 (MSE) 损失函数,在类别和置信度上使用了

交叉熵作为损失函数,其损失函数表示为

$$\begin{aligned}
 L_{\text{loss}} = & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} [(x_i - \hat{x}_i^j)^2 + (y_i - \hat{y}_i^j)^2] + \\
 & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} [(\sqrt{w_i^j} - \sqrt{\hat{w}_i^j})^2 + (\sqrt{h_i^j} - \sqrt{\hat{h}_i^j})^2] + \\
 & \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{obj}} [\hat{C}_i^j \ln(C_i^j) + (1 - \hat{C}_i^j) \ln(1 - C_i^j)] + \\
 & \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{ij}^{\text{noobj}} [\hat{C}_i^j \ln(C_i^j) + (1 - \hat{C}_i^j) \ln(1 - \\
 & C_i^j)] + \sum_{i=0}^{S^2} I_{ij}^{\text{obj}} \sum_{c \in c_{\text{classes}}} [\hat{P}_i^j \ln(P_i^j) + (1 - \hat{P}_i^j) \\
 & \ln(1 - P_i^j)] \quad (1)
 \end{aligned}$$

式中: $x_i, y_i, w_i^j, h_i^j, C_i^j, P_i^j$ 分别表示预测框的坐标 (包括中心点坐标 x 和 y 、宽度 w 、高度 h)、置信度和类别概率; $\hat{x}_i^j, \hat{y}_i^j, \hat{w}_i^j, \hat{h}_i^j, \hat{C}_i^j, \hat{P}_i^j$ 分别表示真实框的坐标、置信度和类别概率; λ_{coord} 表示坐标损失权重; I_{ij}^{obj} 表示第 i 个网格的第 j 个 anchor box 是否负责该 object (0 或 1); λ_{noobj} 表示不包含 object 的置信度损失权重。

文献[17]指出目标检测算法在计算边界框损失时采用的是预测框与真实框的 L1 范数或 L2 范数来计算位置回归损失,但在评测时却使用 IoU 来判断是否检测到目标,但是两者并不等价。如图 5(a) 和 (b) 所示,当 L2 范数或 L1 范数相等时,即使预测框与真实框距离相同, IoU 的值也不同。

若直接使用 IoU 作为边界框损失,则当预测框和边界框无重合时, IoU 值为 0, 导致优化损失函数梯度为 0, 无法优化。所以本文使用 GIoU 来代替 IoU 作为坐标误差损失函数,使网络朝着预测框和真实框重叠度较高的方向优化。A 和 B 分别表示预测框和真实框,若 C 是能包含 A 和 B 的最小凸集,则 $R_{\text{GIoU}} = R_{\text{IoU}} - \frac{|C \setminus (A \cup B)|}{|C|}$, 其中 $R_{\text{IoU}} = \frac{|A \cap B|}{|A \cup B|}$, 使用 GIoU 度量指标的坐标误差损失函数 $L_{\text{GIoU}} = 1 - R_{\text{GIoU}}$ 。

同时,在置信度损失函数上使用 Focal Loss^[18] 代替交叉熵损失。Focal Loss 是基于交叉熵损失的改进,通过降低大量简单背景在训练过程中所占的权重,可以有效解决 one-stage 目标检测中正负样本比例严重失衡的问题。Focal Loss 定义为

$$L_{\text{FL}}(p_t) = -\alpha_t (1 - p_t)^\gamma \ln(p_t), \quad (2)$$

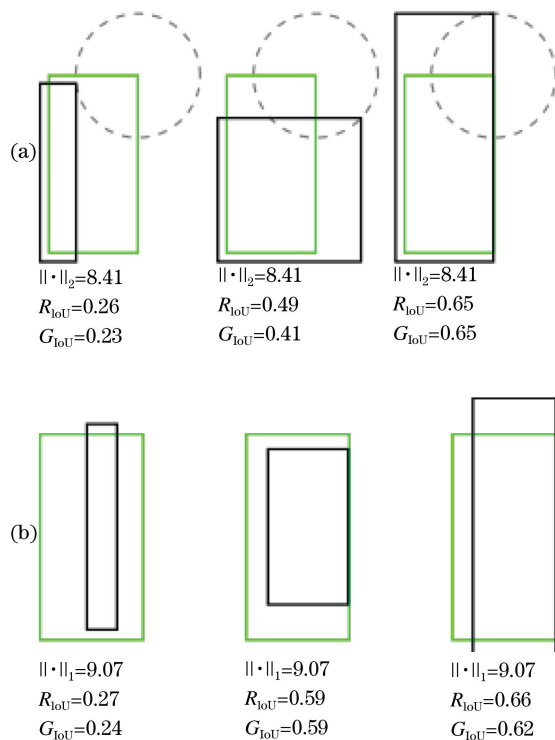


图 5 IoU 标准与 GIoU 标准对比图。(a) L2 范数；
(b) L1 范数

Fig. 5 Comparison chart of IoU standard and GIoU standard. (a) L2 norm; (b) L1 norm

式中: $p_i = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{otherwise} \end{cases}$; α_i 为控制正负样本的权重参数; γ 为控制难易分类样本的参数。本文中 α_i 设为 0.5。

改进后的损失函数使用 GIoU Loss 作为预测框损失, Focal loss 作为置信度损失, 类别损失仍使用原来的交叉熵损失, 将融合了 GIoU Loss 和 Focal Loss 的 YOLO-4l 模型记作 GF-YOLOv3 模型。

4 实验结果与分析

本文实验在 Darknet 框架下进行, 操作系统为 Ubuntu18.04, 硬件环境为: Intel(R) Xeon(R) E5-3650v4 @ 2.42 GHz CPU, 内存为 8 GB, NVIDIA Tesla V100 GPU, 显存为 16 GB。

4.1 数据集制作与预处理

实验数据集来自于网络收集^[19]及监控视频截取, 总共包括 6057 张图片。在标注前对图像进行预处理, 主要包括剔除部分不含目标图像及图像类型的归一化。然后使用标注工具 LabelImg 进行标注, 把佩戴安全帽人员的整个头部区域标记为 hat,

未佩戴安全帽人员头部区域标记为 person, 留取了 600 张图片作为测试集。经标记处理后, 标记出的目标总数为 92879, 其中佩戴安全帽的目标数为 6917, 未佩戴安全帽的目标数为 85962。数据集具体分布如表 2 所示。本文使用在线增强的方式扩大数据集, 在训练之前, 训练集图片会随机进行旋转、翻转、剪切、调整对比度以及饱和度等操作。

表 2 数据集分布表

Table 2 Data set distribution

Class	Marked number	Train set number	Test set number
Hat	6917	6343	571
Person	85962	78104	7858

4.2 网络模型训练

采用 YOLO 预训练模型和微调的方法进行训练, 其中 YOLOv3-4l 和 GF-YOLOv3 采用了 K-means 算法聚类的 12 个锚框。初始学习率、衰减系数以及动量分别设置为 0.001、0.0005 和 0.9, 迭代 150 次后使用初始学习率进行训练, 8000 次迭代后学习率降为 0.0001, 15000 次迭代后学习率降为 0.00001, 使损失函数进一步收敛。图 6 为训练过程中优化后的 GF-YOLOv3 的损失值变化, 随着训练次数增加, 损失值不断下降, 最终模型损失值稳定在 1.65 左右。

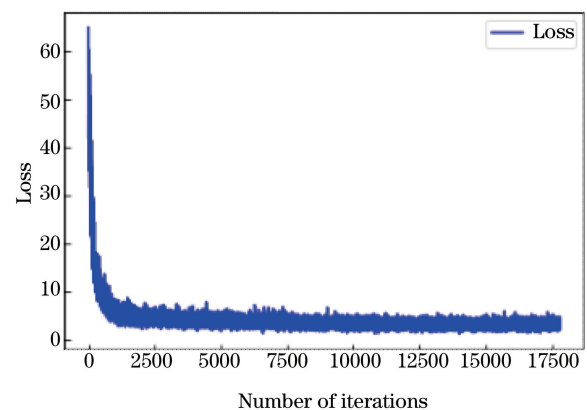


图 6 损失值变化

Fig. 6 Change of loss value

4.3 模型结果评价

为了验证本文改进思路的有效性, 使用消融实验方法分析不同改进措施对整个模型的影响。将本文算法裁剪成五组分别进行训练, 第一组为原始 YOLOv3; 第二组为增加一个尺度的 YOLOv3, 即 YOLOv3-4l; 第三组为第二组基础上使用损失函数 GIoU, 即 YOLOv3 + GIoU Loss; 第四组为第二组

基础上使用 Focal 损失函数,即 YOLOv3 + Focal Loss;第五组为第二组基础上同时使用 GIoU 和 Focal 损失函数,即本文提出的 GF-YOLOv3 算法。

在安全帽测试集上实验结果如表 3 所示,其中 mAP 是平均精度值,表示所有类别识别精度的平均值,本文中为 hat 与 person 识别精度的平均值。

表 3 不同模型检测精度比较

Table 3 Comparison of detection accuracy of different models

Group	Algorithm	Hat	Person	mAP	Time /s	Speed /((frame · s ⁻¹))
1	YOLOv3	0.8757	0.8568	0.8662	0.022	45
2	YOLOv3-4l	0.8972	0.8439	0.8705	0.025	40
3	YOLOv3-4l+GIoU Loss	0.9206	0.8122	0.8664	0.029	34
4	YOLOv3-4l+Focal Loss	0.8930	0.8183	0.8556	0.030	33
5	GF-YOLOv3(proposed)	0.9180	0.8837	0.9009	0.029	34

从表 3 可以看出,原 YOLOv3 算法检测速度最快,但检测精度不高;增加检测尺度和单独使用 GIoU Loss 或 Focal Loss 都能提高安全帽检测精度,其中仅使用 GIoU Loss 的检测精度最高达到 92.06%。由于未使用重新聚类后的先验候选框,导致平均检测精度略微下降,而本文改进后的算法对安全帽检测精度和平均检测精度都有了提升。改进后的模型检测单张图片所需时间为 29 ms,检测安全帽精度达到 91.80%,与 YOLOv3 模型相比,平均精度提高了 3.47%,在安全帽识别精确率上提高了 4.23%,检测单张图片所需时间增加了 7 ms,检测速度每秒降低了 11 frame,但仍大于 25 frame/s,依旧满足实时性要求。

为了公平评估算法的有效性,将测试集随机分为 5 组,每组测试数量相同为 500 张,分别使用 YOLOv3、YOLOv3-4l、YOLOv3-4l + GIoU Loss、YOLOv3-4l + Focal Loss 和 GF-YOLOv3 算法进行

检测。本文使用准确率^[20]和召回率^[21]两个参考指标来衡量测试算法的性能。准确率又称查准率,表示检测出正样本数量占所有被检测出来正样本数量的比率, $P_{\text{precision}} = P_{\text{TP}} / (P_{\text{TP}} + P_{\text{FP}})$;召回率又称查全率,表示检测出的正样本数量占真正的正样本数量的比例, $P_{\text{recall}} = P_{\text{TP}} / (P_{\text{TP}} + P_{\text{FN}})$ 。其中 P_{TP} 表示经过算法运行后正确识别到未戴安全帽的建筑工人的数量, P_{FP} 表示错误识别到未戴安全帽的建筑工人数量, P_{FN} 表示未戴安全帽的建筑工人被误判的数量。按照测试识别结果统计了 P_{TP} 、 P_{FP} 、 P_{FN} 、准确率和召回率,统计结果如表 4 所示。

从表 4 可以看出,增加尺度后召回率有了较大提升;单独使用 GIoU Loss 作为边界框损失,对准确率和召回率提升不明显;而单独采用 Focal Loss 对准确率有较大的提升,本文优化后的 GF-YOLOv3 算法相比 YOLOv3 在准确率和召回率上均有不错的提升。

表 4 佩戴识别结果统计

Table 4 Wear identification results statistics

Group	Algorithm	Total	P_{TP}	P_{FP}	P_{FN}	Precision	Recall
1	YOLOv3	500	6742	1007	1097	0.87	0.86
2	YOLOv3-4l	500	6635	1080	499	0.86	0.93
3	YOLOv3-4l+GIoU Loss	500	6718	994	1236	0.87	0.84
4	YOLOv3-4l+Focal Loss	500	6641	447	1313	0.94	0.83
5	GF-YOLOv3(proposed)	500	6728	429	354	0.94	0.95

图 7 为模型在不同场景下(密集单类/多类目标、单类/多类小目标)的检测效果图,左列为原图,中间列为 YOLOv3 模型检测效果图,右列为本文改进后算法检测效果图。从结果可以看出,YOLOv3 算法均存在不同程度的漏检情况,图 7(a)场景中 YOLOv3 算法未能识别后排被遮挡目标,改进后的算法则正确区分,说明本文改进算法对遮挡具有一定的鲁棒性;图 7(b)场景中存在多处密集目标,

YOLOv3 对此识别效果差,预测框没有准确包含目标,而改进后算法目标预测框更加准确,说明对损失函数的优化使得预测位置和精度更加准确;图 7(c)和(d)场景中 YOLOv3 算法均不能识别较小尺度目标,而本文改进算法可以准确识别,说明改进后算法更进一步提高了对小目标的检测效果。实验结果显示,本文优化后的 YOLOv3 算法较原 YOLOv3 算法更适合安全帽佩戴检测任务。

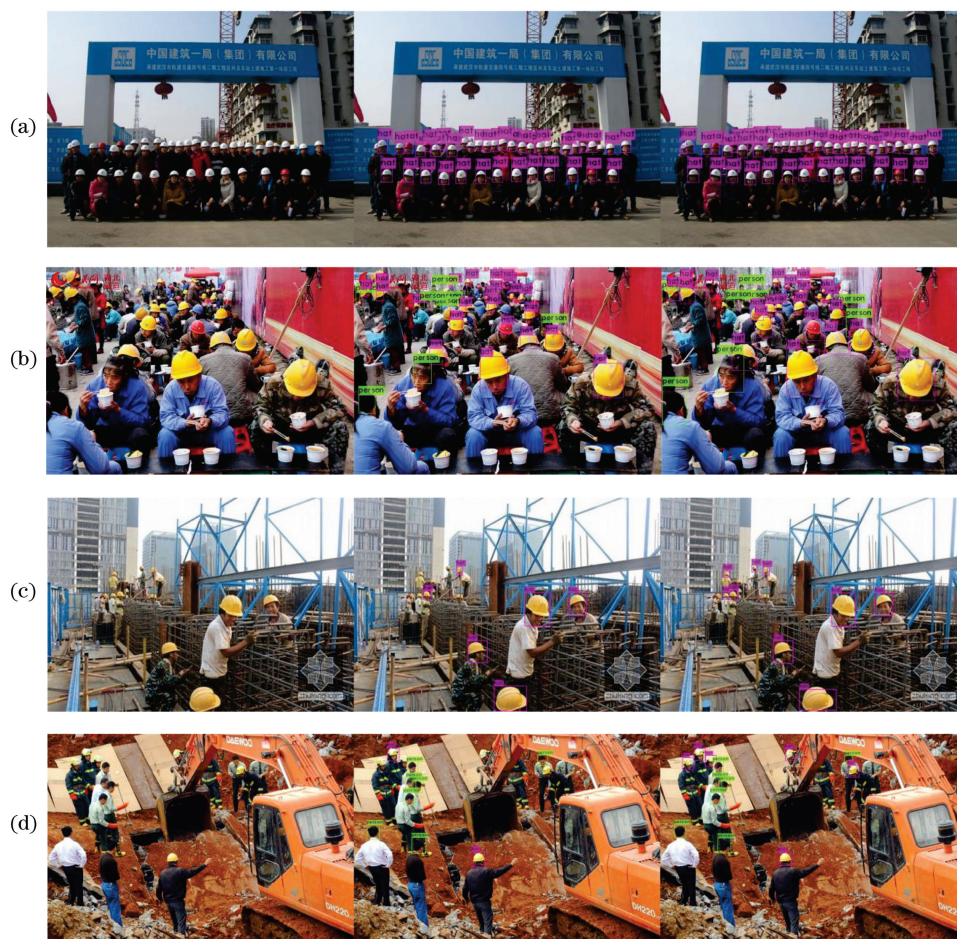


图 7 检测效果图。(a) (b) 密集目标；(c)(d) 小目标

Fig. 7 Test results. (a)(b) Intensive targets; (c)(d) small targets

5 结 论

针对传统安全帽佩戴识别算法检测效果差、精度低的问题,本文提出了一种基于 YOLOv3 改进的安全帽佩戴识别检测算法。首先通过增加 104×104 特征图来提高模型对较小安全帽的检测率,其次使用 K-means 算法在自制的安全帽数据集上聚类出了候选锚框尺寸,最后通过在预测框坐标回归损失上使用 GIoU Loss,提高算法检测精度,同时将 Focal Loss 加入置信度损失中,缓解数据类别分布不均衡对算法带来的影响。实验结果表明,相较于 YOLOv3 算法,本文改进后的 GF-YOLOv3 模型,在多个场景下对施工人员佩戴安全帽识别均有较高的检测精度。本文算法由于模型结构变得复杂,在检测速度上略有下降,同时从理论上来说对其他类似小目标也具有一定适用性,但未进行实验验证,后续工作可以从这两方面考虑对模型进一步优化。

参 考 文 献

- [1] Liu X H, Ye X N. Skin color detection and hu moments in helmet recognition research [J]. Journal of East China University of Science and Technology (Natural Science Edition), 2014, 40(3): 365-370.
刘晓慧, 叶西宁. 肤色检测和 Hu 矩在安全帽识别中的应用 [J]. 华东理工大学学报(自然科学版), 2014, 40(3): 365-370.
- [2] Cai L M. Research on human detection and tracking in underground coal mine videos [D]. Xuzhou: China University of Mining and Technology, 2010: 74-80.
蔡利梅. 基于视频的煤矿井下人员目标检测与跟踪研究 [D]. 徐州: 中国矿业大学, 2010: 74-80.
- [3] Rubaiyat A H M, Toma T T, Kalantari-Khandani M, et al. Automatic detection of helmet uses for construction safety [C] // 2016 IEEE/WIC/ACM International Conference on Web Intelligence Workshops (WIW), October 13-16, 2016, Omaha, NE, USA. New York: IEEE Press, 2016: 135-142.

- [4] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C] // 2014 IEEE Conference on Computer Vision and Pattern Recognition, June 23-28, 2014, Columbus, OH, USA. New York: IEEE Press, 2014: 580-587.
- [5] Girshick R. Fast R-CNN [C] // 2015 IEEE International Conference on Computer Vision (ICCV), December 7-13, 2015, Santiago, Chile. New York: IEEE Press, 2015: 1440-1448.
- [6] Ren S Q, He K M, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.
- [7] Redmon J, Divvala S, Girshick R, et al. You only look once: unified, real-time object detection[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 27-30, 2016, Las Vegas, NV, USA. New York: IEEE Press, 2016: 779-788.
- [8] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 21-26, 2017, Honolulu, HI, USA. New York: IEEE Press, 2017: 6517-6525.
- [9] Redmon J, Farhadi A. YOLOv3: an incremental improvement[EB/OL]. (2018-04-08)[2020-07-24]. <https://arxiv.org/abs/1804.02767>.
- [10] Liu W, Anguelov D, Erhan D, et al. SSD: single shot MultiBox detector [M] // Computer Vision - ECCV 2016. Cham: Springer International Publishing, 2016: 21-37.
- [11] Ju M R, Luo H B, Wang Z B, et al. Improved YOLO V3 algorithm and its application in small target detection [J]. Acta Optica Sinica, 2019, 39(7): 0715004.
鞠默然, 罗海波, 王仲博, 等. 改进的 YOLO V3 算法及其在小目标检测中的应用[J]. 光学学报, 2019, 39(7): 0715004.
- [12] Li H B, Xu C Y, Hu C C. Improved real-time vehicle detection method based on YOLOV3 [J]. Laser & Optoelectronics Progress, 2020, 57(10): 101507.
李汉冰, 徐春阳, 胡超超. 基于 YOLOV3 改进的实时车辆检测方法[J]. 激光与光电子学进展, 2020, 57(10): 101507.
- [13] Zhang Y, Shen Y L, Zhang J. An improved tiny-yolov3 pedestrian detection algorithm [J]. Optik, 2019, 183: 17-23.
- [14] He K M, Zhang X Y, Ren S Q, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904-1916.
- [15] Li D J, Li R H. Mug defect detection method based on improved faster RCNN [J]. Laser & Optoelectronics Progress, 2020, 57(4): 041515.
李东洁, 李若昊. 基于改进 Faster RCNN 的马克杯缺陷检测方法[J]. 激光与光电子学进展, 2020, 57(4): 041515.
- [16] Zhu M M, Xu Y L, Ma S P, et al. Airplane detection based on feature fusion and soft decision in remote sensing images[J]. Acta Optica Sinica, 2019, 39(2): 0210001.
朱明明, 许悦雷, 马时平等. 基于特征融合与软判决的遥感图像飞机检测[J]. 光学学报, 2019, 39(2): 0210001.
- [17] Rezatofighi H, Tsoi N, Gwak J, et al. Generalized intersection over union: a metric and a loss for bounding box regression [C] // 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA. New York: IEEE, 2019: 658-666.
- [18] Lin T Y, Goyal P, Girshick R, et al. Focalloss for dense object detection [J]. 2017 IEEE International Conference on Computer Vision (ICCV), 2017: 2999-3007.
- [19] Wu J X, Cai N, Chen W J, et al. Automatic detection of hardhats worn by construction personnel: a deep learning approach and benchmark dataset[J]. Automation in Construction, 2019, 106:102894.
- [20] Gupta S, Agrawal A, Gopalakrishnan K, et al. Deep learning with limited numerical precision [EB/OL]. (2015-09-04)[2020-07-24]. <https://arxiv.org/abs/1502.02551>.
- [21] Lee H, Grosse R, Ranganath R, et al. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations [C] // Proceedings of the 26th Annual International Conference on Machine Learning - ICML'09, June 14-18, 2009. Montreal, Quebec, Canada. New York: ACM Press, 2009: 609-616.