

基于 YOLOv5 改进的铁路工人安全帽检测算法研究

周 瑶¹, 周 石^{2,3}

(1. 中国移动通信集团湖北有限公司 武汉分公司, 武汉 430100;

2. 武汉纺织大学 机械工程与自动化学院, 武汉 430073;

3. 武汉纺织大学 湖北省数字化纺织装备重点实验室, 武汉 430073)

摘要: 目前铁路上普遍采用人工监督方式来检测工人是否佩戴安全帽, 但监督范围过大, 在实践中不能及时跟踪和管理所有工作人员; 因此针对该问题, 采用深度学习目标检测的方法, 通过改进 YOLOv5s 目标检测算法来实现铁路工人是否佩戴安全帽和穿戴背心; 具体来说, 以 YOLOv5s 算法为基础, 采用 GhostNet 模块替换原始网络中的卷积 Conv, 提高模型的实时检测速度; 采用更高效简单的多尺度特征融合 BiFPN, 使特征融合方式更加简单高效, 以提高检测速度和降低模型复杂度; 把原始的 CIOU 损失函数替换为 SIOU 损失函数, 以提高模型精度; 研究结果表明, 改进的 YOLOv5s-GBS 算法的准确率和识别效率可达到 95.7% 和每秒 45 帧, 并且模型大小减少了一半, 准确率提高了 4.5%。

关键词: 安全帽; 深度学习; BiFPN; SIOU 损失函数; YOLOv5s-GBS 算法

Research on the Detection Algorithm of Railway Worker's Hard Hat Based on YOLOv5 Improvement

ZHOU Yao¹, ZHOU Shi^{2,3}

(1. Wuhan Branch, China Mobile Co., Ltd., Wuhan 430100, China;

2. School of Mechanical Engineering and Automation, Wuhan Textile University, Wuhan 430073, China;

3. Hubei Provincial Key Laboratory of Digital Textile Equipment, Wuhan Textile University, Wuhan 430073, China)

Abstract: At present, manual supervision is generally used on railways to detect whether workers wear safety helmets, but the supervision scope is too large, and it is impossible to timely track and manage all workers in practice. Therefore, in response to this problem, the deep learning target detection method is adopted, and the YOLOv5s target detection algorithm is improved to realize whether railway workers wear hard hats and vests. Specifically, based on the YOLOv5s algorithm, the GhostNet module is used to replace the convolution Conv in the original network to improve the real-time detection speed of the model; the more efficient and simple multi-scale feature fusion BiFPN is used to make the feature fusion method simpler and more efficient to improve the detection speed and reduce the complexity of the model; The original CIOU loss function is replaced by the SIOU loss function to improve the accuracy of model. The research results show that the accuracy and recognition efficiency of the improved YOLOv5s-GBS algorithm can reach 95.7% and 45 fps, respectively, and the model size is reduced by half, and the accuracy rate of the model is increased by 4.5%.

Keywords: hard hat; deep learning; BiFPN; SIOU loss function; YOLOv5s-GBS algorithm

0 引言

铁路工人佩戴安全帽和穿戴背心, 对工作人员的安全至关重要。然而, 由于缺乏安全意识, 安全帽和背心的重要性往往被忽视^[1]。在铁路上普遍采用人工监督方式, 但监督范围过大, 在实践中不可能及时跟踪和管理所有工作人员^[2]。因此, 基于图像处理^[3]的智能监控下的安全帽佩戴和背心穿戴检测正逐渐成为铁路实施管理的主要手段。图像处理又可分为深度学习算法和传统检测算法。由于基于深度学习的检测方法可以有效避免检测精度和效率低的问题, 且泛化性强。因此, 基于深度学习的铁路工人安全帽检测方法成为近年来的研究热点。

随着深度学习和检测的结合, 基于深度学习的检测算法可以分为两大类, 两阶段网络和单阶段网络。最具代表性的两阶段网络包括 R-CNN^[4]和 Faster R-CNN^[5], 单阶段网络包括 SSD^[6]和 YOLO^[7]。同时, 无数学者使用深度学习目标检测算法对安全帽“是否正确佩戴”作了大量的研究, 并取得了一定的研究成果。陈松波^[8]将 K-means++ 算法融入 Faster R-CNN 中, 更好的适配小尺寸头盔, 提高准确

收稿日期: 2023-04-19; 修回日期: 2023-05-29。

作者简介: 周 瑶 (1993-), 女, 硕士研究生。

引用格式: 周 瑶, 周 石. 基于 YOLOv5 改进的铁路工人安全帽检测算法研究[J]. 计算机测量与控制, 2024, 32(3): 71-78, 175.

率,但该方法检测速度较慢,无法满足实时要求。饶诚^[9]通过在 YOLOv3-Tiny 的特征提取网络中添加改进的空间金字塔池(SPP)模块,以提取具有丰富语义信息的局部和全局特征,该方法可以对安全帽佩戴实时性检测,但检测精度较低,漏检率较高无法满足实际铁路上的检测要求。范武^[10]利用 Densenet 在模型参数和技术成本上的优势,替代 YOLOv3 骨干网络进行特征提取,该方法可以有效检测是否佩戴安全帽,但该方法模型过大,检测速度较慢。陈俊华^[11]在 YOLOv4 的骨干网络 3 个输出特征层中嵌入坐标注意力机制模块以增强特征信息,并设计了改进的特征融合结构来融合目标信息,该方法可以有效检测出是否佩戴安全帽,但检测速度较慢,且检测精度较低。戴斌^[12]对 SSD 模型进行改进,使用了轻量级网络结构,减少了模型的参数量和计算量,该模型检测速度较快,但精度较低。谭石磊^[13]等人通过在 YOLOv5s 的基础上增加一层检测层,该方法可以有效降低小目标漏检率,但该方法模型过大,检测速度较慢。白培瑞等人^[14]提出 DS-yolov5 模型,在 YOLOv5s 的基础上加入了简化版的 Transformer 模块,加强对全局语义信息的特征学习能力,并在 Neck 层引入双向特征金字塔融合多尺度的特征,最后模型在公开数据集 SHWD 上的验证,Map 达到 95.5%,更好适应了工地复杂的生产环境,但该方法模型过大,检测速度较慢。

基于上述相关文献研究成果,为了更好地平衡铁路工人安全帽检测任务的检测精度、检测速度和网络模型体积,在 YOLOv5s 模型的基础上进行相关改进,提出一种新的网络模型。本文的主要贡献如下:

1) 采用 GhostNet 模块替换原始网络中的卷积 Conv,这样可以减少冗余,使参数量变少,更利于模型的实时检测和最终部署。

2) 采用更高效简单的多尺度特征融合 BiFPN,来对原始 FPN 进行改进,BiFPN 它舍弃了最顶层和底层的两个神经元,同时在中间各特征层加入一条与输入输出特征相连的残差结构,相较于之前的特征融合方式更加简单高效。

3) 把原始的 CIOU 损失函数替换为 SIOU 损失函数,这样可以充分考虑到所需回归之间的向量角度,提高收敛速度以及检测精度。

1 基于融入 GhostNet 模块、BiFPN、SIOU 损失函数的 YOLOv5s 算法

1.1 原始 YOLOv5 算法的结构

YOLOv5 算法是由 Ultralytics 公司于 2020 年在 YOLOv4 发布不久后推出的。虽然没有相关的论文,但 YOLOv5 在前者基础上借鉴了其他算法的创新和改进思路,因此获得了一个更高的检测精度。YOLOv5 网络结构可以简单地描述为三大部分:特征提取网络,Neck 网络和检测头,其结构如图 1 所示。该模型根据输入通道数 C 和模块的复用次数不同可分为 4 个版本,分别是 YOLOv5s、YOLOv5m、YOLOv5l、YOLOv5x,这些不同尺度的模型让算法工程师更好的选择不同的模型,权衡精度和速度,

简化了工程的相关落地与部署。一般来说,YOLOv5s 模型最小,具有更高的检测速度;YOLOv5x 模型最大,具有更高的检测准确率,YOLOv5m 和 YOLOv5l 是在检测速度与检测准确率中均衡的结构模型。为了满足对铁路工人实时检测,选用 YOLOv5s 模型进行检测。

1.2 融入 GhostNet 模块

由于原始的 YOLOv5s 骨干网络过于复杂,参数量多,不利于模型的实时检测和最终部署,而采用 GhostNet^[15]模块替换原始网络中的卷积 Conv,这样可以减少冗余,使参数量变少,更利于模型的实时检测和最终部署。

GhostNet 是华为诺亚方舟实验室于 2020 年^[15]提出的轻量级特征提取网络,它采用端到端神经网络架构,外表为 MoblieNetv3^[16],GhostNet 的核心是 Ghost 模块,如图 2 所示显示了 Ghost 模块之间的卷积过程。Ghost 模块引入了线性运算而不是部分卷积,与标准卷积相比,它分为 3 个步骤:1) 使用标准卷积生成少量的本征图;2) 在第一步生成的特征图的基础上,使用深度卷积或移位等线性操作,用少量参数获得更多的 Ghost 特征图;3) 将前两步生成的特征映射进行合并,得到 Ghost 模块的输出特征映射。在输入和输出特征映射大小相同的情况下,Ghost 模块的计算量远低于普通卷积,实现了用更少的计算获取更多的特征信息,并且不会对模型的性能产生负面影响。

基于 Ghost 模块的轻量级优势,Ghost-BottleNeck 由两个 Ghost 模块堆叠构成。当步长为 1 时,第一个 Ghost 模块是扩展层,参考 MobileNetV2 的结构,在第二个 Ghost 模块之后没有使用 ReLU 激活函数,在彼此层之后引入 BN 和 ReLU 激活函数。当步长为 2 时,捷径由下采样层和步长=2 的深度可分离卷积组成,其中深度可分离卷积可以减少通道数。

1.3 融入 BiFPN 模块

通常,由于铁路工人检测过程中,背景较为复杂,采用原始的 YOLOv5s 会导致深浅层的信息会互相干扰,对检测的精度会产生影响^[17]。因此,对于多尺度的受到影响,对颈部网络进行改进,以提高检测精度。

虽然 YOLOv5s 中的 PANet 通过上采样和下采样在多尺度融合方面取得了很好的效果,但其计算量很大^[18]。然而,双向特征金字塔网(BiFPN)^[19]可以实现简单快速的多尺度特征融合。它采用跨尺度连接去除 PANet 中对特征融合贡献较小的节点,并在同一级别的输入和输出节点之间添加额外的连接。本次研究我们使用 BiFPN 对 Neck 模块进行改进,利用一层结构的 BiFPN 来提高模型的训练效率,如图 3 所示。

利用 BiFPN 改进 YOLOv5s 的颈部,可以实现更简单、更快的多尺度特征融合。此外,在 BiFPN 中引入可学习权重,使其能够学习不同输入特征的重要性,并反复应用自顶向下和自底向上的多尺度特征融合。与 YOLOv5s 的颈部网络 PANet 相比,BiFPN 具有更少参数和 FLOPS^[20]的性能。因此,它可以更好地实时检测铁路工人。

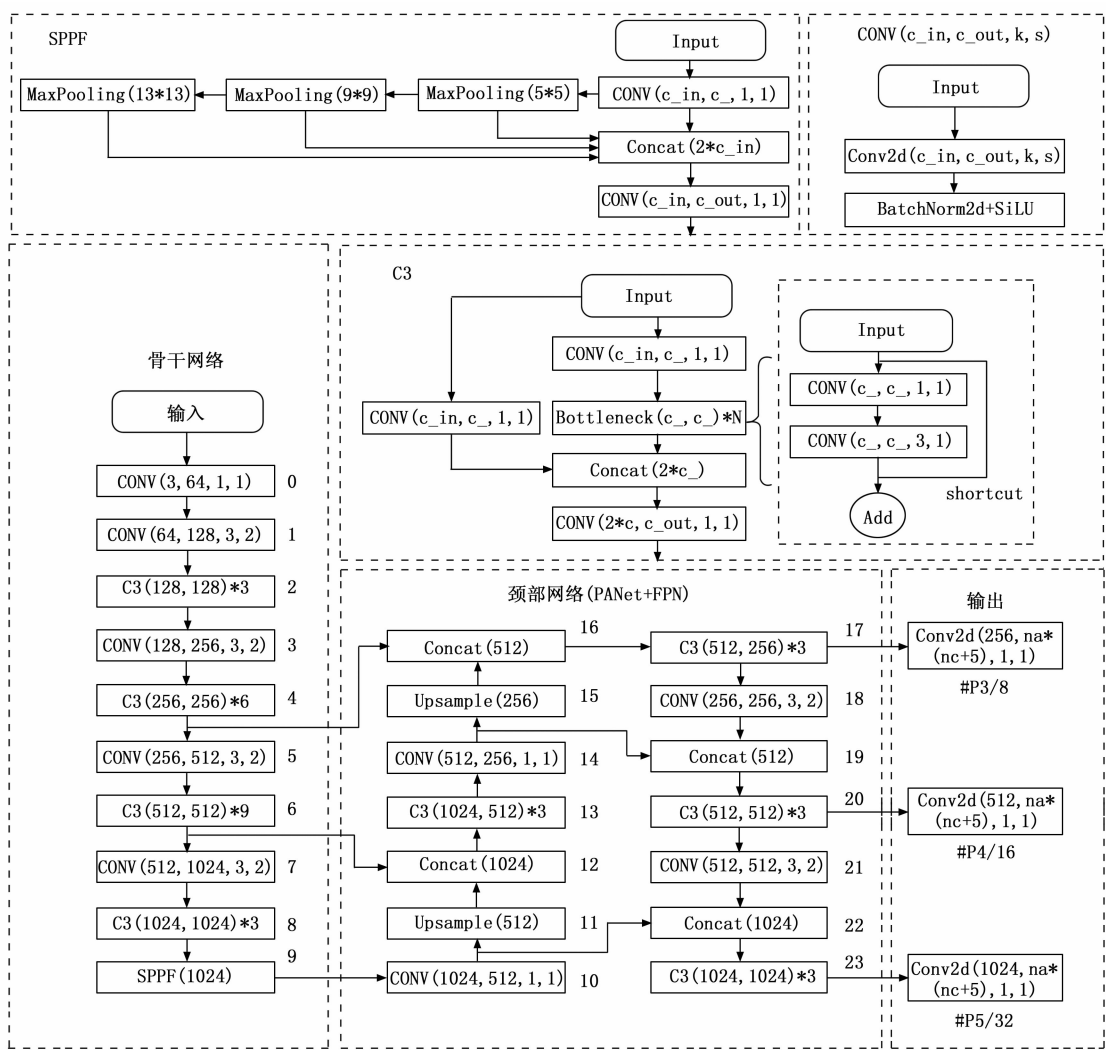


图 1 YOLOv5s 模型结构图

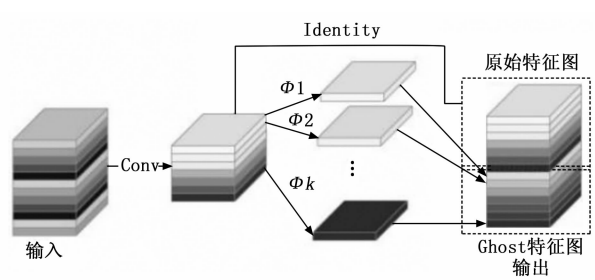


图 2 Ghost 模块的卷积过程

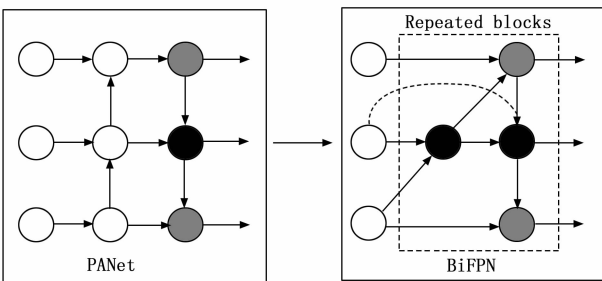


图 3 BiFPN 结构图

1.4 SIOU 损失函数

损失函数在模型性能中起着重要作用^[21],它是衡量深度卷积神经网络预测值与真实值之间差异的重要指标,对于目标检测算法尤为重要。因此,损失函数决定了模型参数训练的好坏。

原始的 YOLOv5s 中的 CIOU Loss^[22]在之前损失函数的基础上增加了对纵横比的考虑,但是没有考虑需要回归的矢量角度,即所需真实框和预测框之间的方向不匹配。此外,采用 CIOU Loss 会导致预测框在训练过程中出现游走的问题,降低实验训练速度。文献 [23] 在 CIOU Loss 的基础上提出了一种新的损失函数 SIOU Loss,它不仅依赖边界框回归指标的聚合,而且在损失函数 cost 中引入方向性,重新定义惩罚指标。如图 4 所示,虚线为 YOLOv5s-SIOU 训练效果,实线为 YOLOv5s-CIOU 训练效果。实验证明,与 CIOU 相比, SIOU 收敛速度更快,训练效果更好,且精确率更高。

SIOU 损失包括 4 个成本:角度、距离、形状和 IOU。角度将真实框和预测框之间的角度差表示为一个变量。角

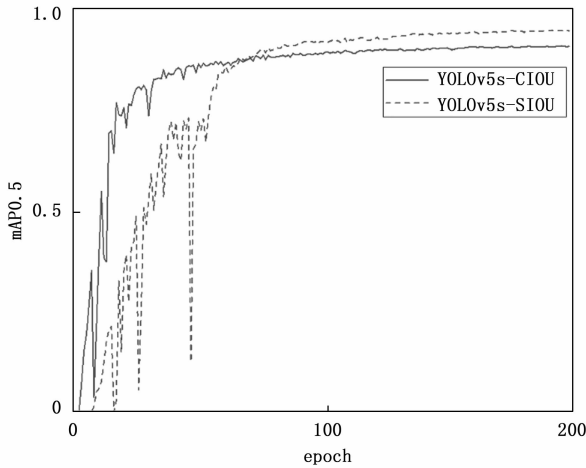


图4 YOLOv5s 模型的 SIOU 和 CIOU 训练效果对比

度成本对损失函数的贡献为图 5，定义如式 (1) 所示。

$$\Delta = 1 - 2 \times \sin^2(\arcsin(x) - \frac{\pi}{4}) \quad (1)$$

$$x = \frac{c_h}{\sigma} = \sin(\alpha)$$

$$\sigma = \sqrt{(b_{Cx}^{gt} - b_{Cx})^2 + (b_{Cy}^{gt} - b_{Cy})^2} \quad (2)$$

$$c_h = \max(b_{Cx}^{gt}, b_{Cx}) - \min(b_{Cx}^{gt}, b_{Cx})$$

在等式 (2) 中， $(b_{Cx}^{gt}, b_{Cy}^{gt})$ 和 (b_{Cx}, b_{Cy}) 分别表示真实框和预测框的中心点坐标。将角度损失作为距离损失的变量可以更好地计算角度造成的损失。

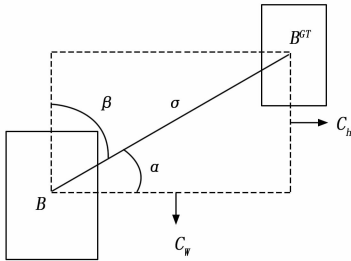


图5 角度成本对损失函数的贡献

距离成本的定义如式 (3) 所示。

$$\Delta = \sum_{l=x,y} (1 - e^{-\gamma \rho_l}) \quad (3)$$

ρ_x 和 ρ_y 都测量了真实框和预测框之间的距离，它们的定义如等式 (4) 和 (5) 所示。更进一步， Δ 表示角度代价，将角度代价作为一个变量融入到距离损失中，可以更准确地表达损失。

$$\rho_x = \left(\frac{b_{Cx}^{gt} - b_{Cx}}{c_w} \right)^2 \quad (4)$$

$$\rho_y = \left(\frac{b_{Cy}^{gt} - b_{Cy}}{c_h} \right)^2 \quad (5)$$

$$\gamma = 2 - \Delta \quad (6)$$

形状成本的定义如式 (7) 所示。

$$\Omega = \sum_{l=w,h} (1 - e^{-\theta l})^\theta,$$

$$\tau_w = \frac{|w - w^{gt}|}{\max(w, w^{gt})},$$

$$\tau_h = \frac{|h - h^{gt}|}{\max(h, h^{gt})} \quad (7)$$

其中： (w, h) 和 (w^{gt}, h^{gt}) 分别表示预测框和真实框的宽度和高度。 θ 表示控制损失函数对形状成本的关注程度。

IOU 表示真实框和预测框之间联合比率的交集，IOU 成本的定义如式 (8) 所示。

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

其中： B 和 B^{gt} 分别代表预测框区域和真实框区域。

SIOU loss 的定义如式 (9) 所示。

$$L_{box} = 1 - IOU + \frac{\Delta + \Omega}{2} \quad (9)$$

对于方程 (9)， Δ 表示修改后的距离成本，而 Ω 和 IOU 表示形状成本和距离成本。

1.5 本文改进算法 YOLOv5s-GBS

将融入 GhostNet 模块、BiFPN、SIOU 损失函数的 YOLOv5s 算法称为 YOLOv5s-GhostNet-BiFPN-SIOU 算法 (简称 YOLOv5s-GBS 算法)，其算法结构如图 6 所示。

2 实验与结果分析

2.1 实验环境及步骤

本文采用的操作系统为 Windows10，模型训练采用 Pytorch 深度学习框架，采用 GPU 加速，型号为 NVIDIA GeForce GTX 1070Ti，本文实验所使用的计算机配置具体如表 1 所示。

表1 实验环境配置

硬件配置		软件配置	
显卡	NVIDIA GeForce GTX 1070Ti	系统	Win10
硬盘	SSD NVMe 1TB	Python 版本	3.9.0
内存	128 GB RAM	框架	Pytorch1.8.1 + cudnn8.0.5
处理器	Intel Core i7 11900H	CUDA	CUDA11.1

实验步骤如下：1) 数据集增强。对铁路工人数据集进行数据增强处理，提高模型泛化能力。2) 实验评价指标的确定。确定一系列指标，用来判断模型的好坏。3) 模型训练及参数设置。加载模型配置文件以及超参数，将输入的数据进行预处理。4) 将数据集数据输入到模型进行训练。随着训练次数的不断增加，将使用 SGD 进行网络参数的更新优化。5) 选出最优的模型权重，通过检测检测脚本，对测试集进行预测。

2.2 数据集介绍与增强

2.2.1 数据集介绍

本文数据集采用 kaggle^[24] 的铁路工人检测，该数据集一共分为三类，有工人、安全帽、背心。Kaggle 成立于 2010 年，是一个进行数据发掘和预测竞赛的在线平台。将

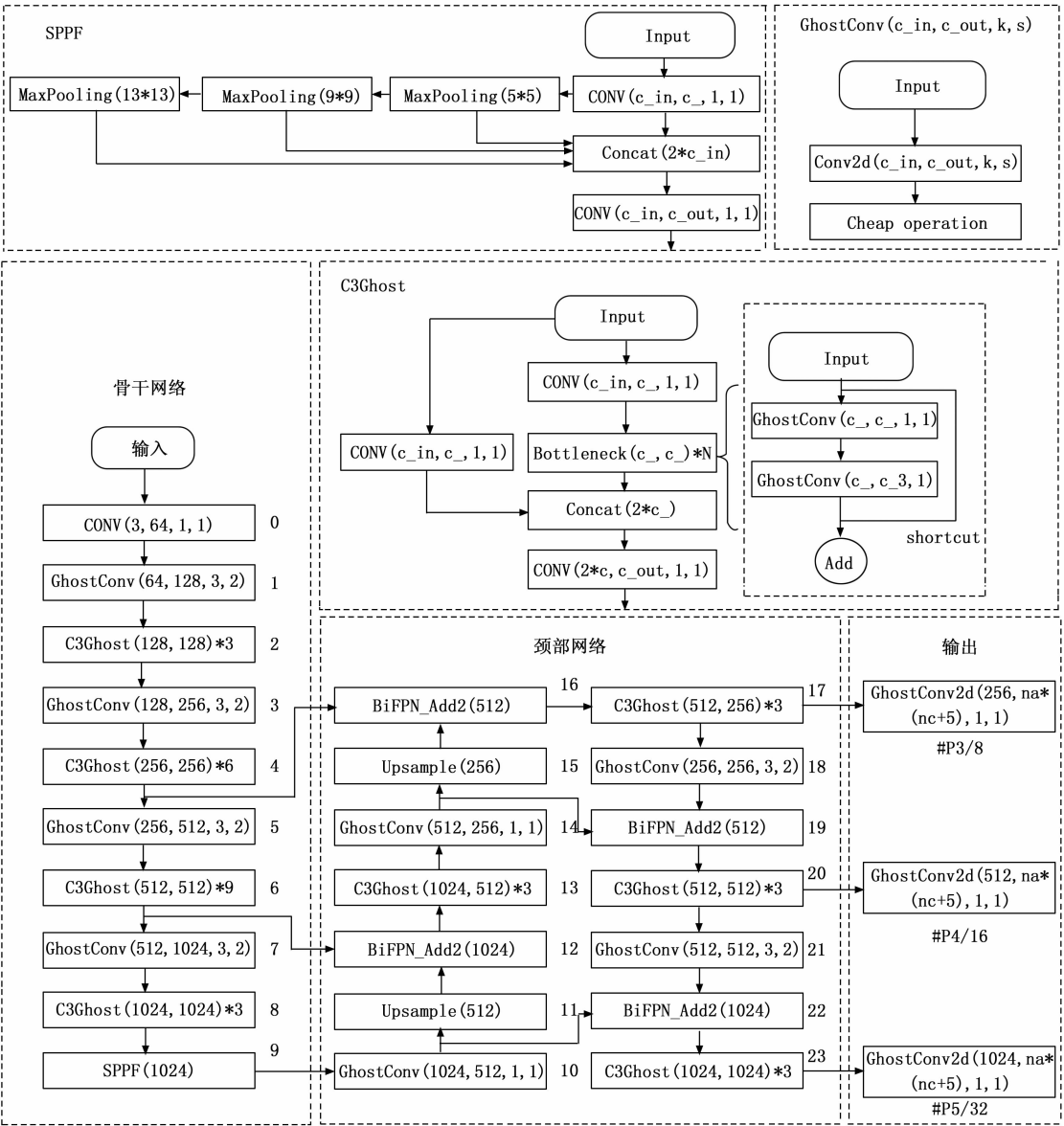


图 6 本文改进算法 YOLOv5s-GBS 结构图

数据集分成训练集、验证集和测试集，其中训练集含有 2 609 张图片，验证集含有 483 张图片，测试集含有 130 张图片。如图 7 所示为数据集可视化分析图，(a) 为目标框位置分布图，横纵坐标代表中心位置；(b) 为目标框大小分布图。横纵坐标代表标注的宽和高。

从图 7 (a) 可以看出目标框中心位置分布均匀，从图 7 (b) 可以看出目标框大小较小。因此，分析可知数据集中的大部分数据为小目标。

2.2.2 Copy-Pasting 数据增强

在对小目标检测过程中，导致小目标漏检率高，除了网络本身可能存在缺陷，还有一大原因就是训练数据中小目标的再现性较差。因此，本文将采用 Copy-Pasting^[25] 数据增强策略，来对数据集的小目标进行过采样、复制、粘贴，以此来提供足够的小目标来和 anchor 进行匹配，以

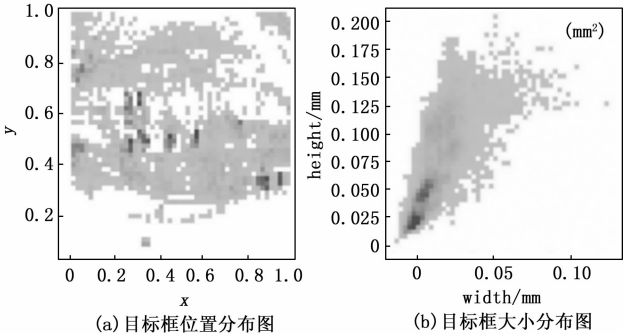


图 7 数据集可视化图

此实现提高小目标检测的性能。

Copy-Pasting 数据增强由 Fang 等人^[25] 在 2017 年提出，

该增强方法通过对小目标样本进行 Oversampling（过采样），然后在样本里的小目标进行 Copy-Pasting（复制粘贴）操作来提高小目标检测的性能。

对小目标的图像进行 Oversampling（过采样），来改善小目标图像较少的问题，及将含有小目标的图片通过多次反复训练，采用这种方法不仅简单，还很有效果。其中，复制的次数就是过采样率。

在进行过采样操作之后，然后通过混合粘贴（Copy-Pasting），利用以下公式（10）将两幅图混合在一起。

$$I_1 \times \alpha + I_2 \times (1 - \alpha) \tag{10}$$

式中， I_1 代表粘贴对象图像， I_2 代表被粘贴对象图像， α 代表 mask 掩模。通俗的意思就是，将 I_1 中的 mask 部分像素扣出来，然后随机粘贴在 I_2 中。

2.3 评价指标

本文采用 Precision、Recall、F1-score 和 mean Average Precision (mAP) 作为评价指标，设置 0.5 IOU 阈值。具体计算公式如下：

$$P_{\text{recision}} = \frac{TP}{TP + FP} \times 100\% \tag{11}$$

$$R_{\text{ecall}} = \frac{TP}{TP + FN} \times 100\% \tag{12}$$

$$F_1 = \frac{2 \times P_{\text{recision}} \times R_{\text{ecall}}}{P_{\text{recision}} + R_{\text{ecall}}} \tag{13}$$

$$mAP = \frac{1}{c} \sum_{M=i}^N P(k) \Delta R(k) \tag{14}$$

其中： TP 是正确识别铁路工人， FP 是误检铁路工人， FN 是漏检铁路工人的数量。 c 是代表种类数； M 和 N 表示 IOU 阈值和 IOU 阈值的数量； $P(k)$ 和 $R(k)$ 是准确率和召回率。 F_1 是 Precision 和 Recall 之间的权衡，用来表示模型的性能， mAP 是不同置信度阈值下的总体性能。参数和浮点运算 (FLOPs) 用于测量模型的网络复杂性，较小的值表示模型的网络复杂度较低。

2.4 模型训练超参数设置

模型训练过程中，对超参数进行设置，超参数的设置好坏会对模型的好坏产生一定的影响，具体设置参数如表 2 所示。

表 2 超参数设置

优化器	SGD
动量因子 (momentum)	0.937
优化器权重衰减 (weight_decay)	0.000 5
学习率 (learning_rate)	0.01
迭代次数 (epoch)	200
批量尺寸 (batch_size)	16

2.5 实验结果与分析

2.5.1 本文 YOLOv5s-GBS 算法与原 YOLOv5s 算法的 p 、 r 对比

本文提出的 YOLOv5s-GBS 算法是基于 YOLOv5s 进行改进的，因此为了更好的看到改进前后算法性能之间的差

异，将 YOLOv5s 算法和本文改进 YOLOv5s-GBS 算法在训练过程中的 P 、 R 进行比较，如图 8 所示。

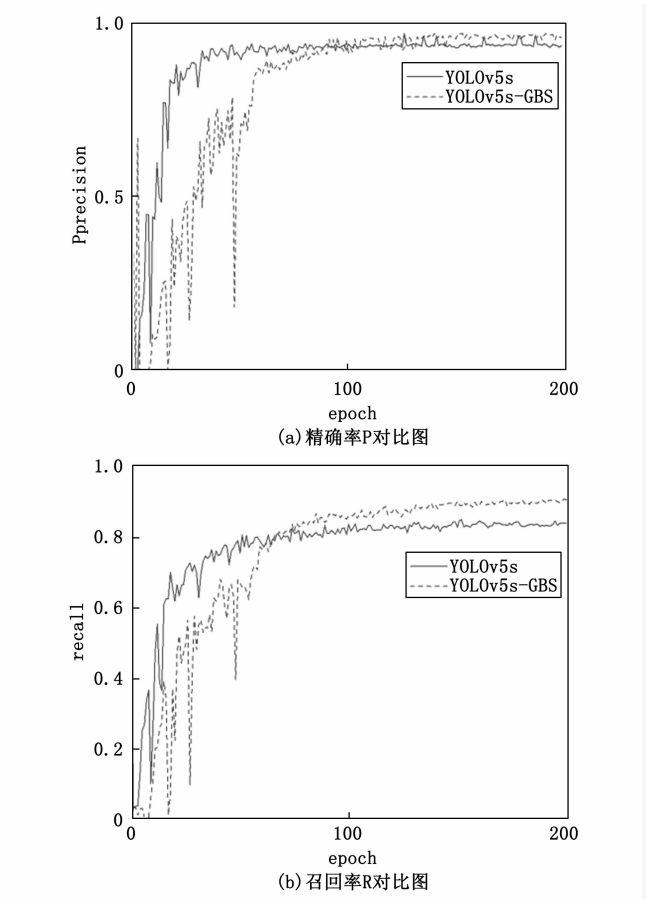


图 8 改进算法 YOLOv5s-GBS 与原始 YOLOv5s 算法 P 、 R 对比图

从图 8 (a)、(b) 可以看出 YOLOv5s-GBS 与 YOLOv5s 算法收敛速度相差不大，但精确率 P 、召回率 R 的值更高。因此，说明本文提出的 YOLOv5s-GBS 算法在保证收敛速度的情况下，且精确率 P 、召回率 R 更高。

2.5.2 消融实验

为了分析每个模块的作用和有效性，采用铁路工人数据集进行消融实验。通过设计三组实验对不同的改进进行分析，每组实验均在相同训练参数，不同算法内容上进行测试。消融实验的具体结果如表 3 所示。“√”是指使用相应的策略来改进模型，以及“—”意思是不使用相应的策略。原始 YOLOv5s 模型大小为 13.7 MB，当采用 Ghost-Net 模块后及 YOLOv5s-G 算法，虽然 F_1 和 mAP 略微有所降低，但是 YOLOv5s-G 模型大小减小将近一半，且网络的参数和浮点数也有大幅度降低，训练速度和检测速度都有一定上升。当在 YOLOv5s-G 模型的基础上，融入 BiFPN 模块对 Neck 模块进行改进及 YOLOv5s-G-B 算法，其模型大小基本持平， F_1 提升了 0.9%， mAP 提升了 0.5%，训练速度提升了 0.012 s · 张⁻¹，检测速度提升了 4 fps。当在 YOLOv5s-G-B 模型的基础上，将 CIOU 损失函数替换为

SIOU 损失函数及 YOLOv5s-GBS 算法, 其模型大小、训练速度和检测速度一致, 但 F_1 提升了 5.1%, mAP 提升了 6%。综上所述, 本文提出 YOLOv5s-GBS 算法与原始的 YOLOv5s 相比, 各项性能都有大幅提升。

2.5.3 不同算法性能对比

为验证所提算法具有更好的效果, 在相同配置条件下采用相同数量的训练集, 并使用现阶段流行的几种目标检测网络进行对比实验: Faster RCNN、SSD、YOLOv3、YOLOv5s、文献 [26]、文献 [27]、YOLOv5s-GBS 算法。其中, SSD、YOLOv3、YOLOv5s、文献 [26]、文献 [27]、YOLOv5s-GBS 算法均属于单阶段检测算法, Faster RCNN 属于两阶段检测算法。使用 mAP 、 F_1 、训练速度和检测速度 4 个评价指标对实验结果进行评价, 实验结果如表 4 所示。

表 3 消融实验对比结果

名称	GhostNet	BiFPN	SIOU 损失函数	模型大小 (M)	F_1 / %	mAP / %	训练速度 / (s · 张 ⁻¹)	检测速度 / (f · s ⁻¹)
YOLOv5s	×	×	×	13.7	89	91.2	0.186	35
YOLOv5s-G	✓	×	×	7.42	87	89.2	0.159	41
YOLOv5s-G-B	✓	✓	×	7.57	87.9	89.7	0.147	45
本文改进 YOLOv5s-GBS	✓	✓	✓	7.57	93.0	95.7	0.147	45

表 4 不同算法性能对比结果

名称	模型大小 / M	F_1 / %	mAP / %	训练速度 / (s · 张 ⁻¹)	检测速度 / (f · s ⁻¹)
Faster RCNN	625	60.2	64.3	0.457	10
SSD	90.1	40.5	49.7	0.214	26
YOLOv3	240	67.3	74.2	0.201	30
YOLOv5s	13.7	89.4	91.2	0.169	35
文献[26]	41.9	86.0	90.0	—	35.2
文献[27]	7.7	89.7	93.50	—	37
本文改进 YOLOv5s-GBS	7.57	93.0	95.7	0.147	45

1) 多种不同的算法在相同的数据集下, YOLOv5s-GBS 算法最优权重大小最小为 7.57 M, 其余的按从小到大的顺序进行排列分别是文献 [27]、YOLOv5s、文献 [26]、SSD、YOLOv3、Faster RCNN, 它们权重大小分别为: 7.7、13.7、41.9、90.1、240、625 M。由此可知, 本文的改进算法 YOLOv5s-GBS 的复杂度最低, 可以很好的部署。Faster RCNN 算法模型复杂度最高, 需要较高算力的计算机来运行。

2) YOLOv5s-GBS 算法与 Faster RCNN 算法相比, F_1 提升了 32.8%, mAP 提升了 31.4%, 且 Faster R-CNN 检测速度 FPS、训练速度过慢, 无法满足实时要求。SSD 算法精度过低, 无法满足精度要求。相比 YOLOv3 算法, F_1 提升了 25.7%, mAP 提升了 21.5%, 训练速度提升了

0.054 s · 张⁻¹, 检测速度提升了 15 fps。相比 YOLOv5s, F_1 提升了 3.6%, mAP 提升了 4.5%, 训练速度提升了 0.022 s · 张⁻¹, 检测速度提升了 10 fps。相比文献 [26], F_1 提升了 7%, mAP 提升了 5.7%, 检测速度提升了 9.8 fps。相比文献 [27], F_1 提升了 3.3%, mAP 提升了 2.2%, 检测速度提升了 8 fps。通过这些结果可以看出, 本文提出的改进算法 YOLOv5s-GBS 相比其他主流算法来说, 在性能上均有较大的提升, 更加满足实际检测要求。

2.6 检测结果对比分析

为了更直观地看出 YOLOv5s-GBS 算法与原 YOLOv5s 算法的检测差距, 将测试集对改进算法进行测试。使用 YOLOv5s-GBS 算法和 YOLOv5s 算法分别进行测试, 部分检测结果如下图 9 原 YOLOv5s 检测结果、图 10 YOLOv5s-GBS 算法检测结果所示。

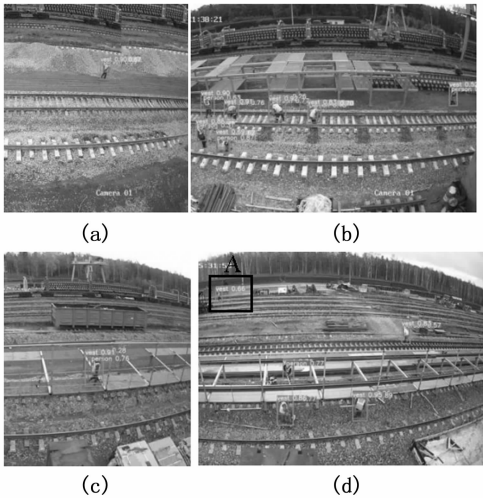


图 9 原 YOLOv5s 检测结果

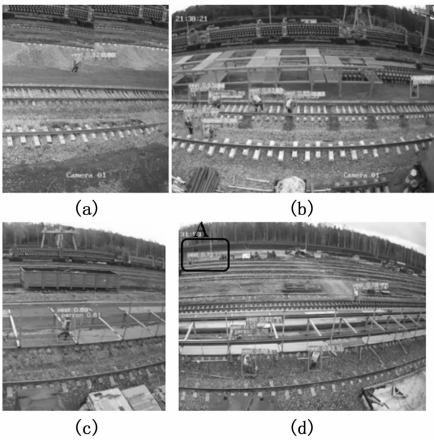


图 10 本文 YOLOv5s-GBS 算法检测结果

通过图 10 (a) 与图 9 (a) 对比可知, YOLOv5s-GBS 算法检测精度相比 YOLOv5s 有所提升。图 10 (b) 与图 9 (b) 对比可知, YOLOv5s-GBS 算法检测精度相比 YOLOv5s 有所提升, 且对小目标检测更好。图 10 (c) 与

图 9 (c) 对比可知, YOLOv5s-GBS 算法检测精度相比 YOLOv5s 有所提升。图 10 (d) 与图 9 (d) 对比可知, YOLOv5s-GBS 算法检测精度相比 YOLOv5s 有所提升, 且图 9 (d) A 位置处, 出现了漏检, 而图 10 (d) 图上 A 位置处未出现漏检。实验表明, 本文改进算法 YOLOv5s-GBS 相比原始 YOLOv5s 算法, 在检测精度上有明显提升, 对实际场景中工人检测效果更好。

3 结束语

本文提出的基于融入 GhostNet 模块、BiFPN、SIOU 的改进 YOLOv5s 铁路工人佩戴安全帽和穿戴背心检测算法及 (YOLOv5s-GBS 算法)。针对原 YOLOv5s 算法网络过于复杂, 参数量多, 不利于模型的实时检测和最终部署等问题, 首先采用 GhostNet 模块替换原始网络中的卷积 Conv, 这样可以减少冗余, 使参数量变少, 更利于模型的实时检测和最终部署。其次采用更高效简单的多尺度特征融合 BiFPN, 来对原始 FPN 进行改进, BiFPN 它舍弃了最顶层和底层的两个神经元, 同时在中间各特征层加入一条与输入输出特征相连的残差结构, 相较于之前的特征融合方式更加简单高效。最后把原始的 CIOU 损失函数替换为 SIOU 损失函数, 这样可以充分考虑到所需回归之间的向量角度, 提高收敛速度以及检测精度。研究表明, 本文提出的 YOLOv5s-GBS 算法的调和平均数 F_1 、均值平均精度 mAP 均有较为显著的提升, 同时模型大小相比原 YOLOv5s 减少了一半, 更利于模型的实时检测和最终部署, 更适合对铁路工人佩戴安全帽和穿戴背心进行检测。

参考文献:

- [1] 许 逵, 李鑫卓, 张 历, 等. 自然场景下配电网施工安全帽佩戴检测算法 [J/OL]. 计算机工程与应用, 2023, 4 (1): 1-11. [2023-05-20] <http://kns.cnki.net/kcms/detail/11.2127.TP.20230328.1059.016.html>
- [2] 乔 炎, 甄 彤, 李智慧. 改进 YOLOv5 的安全帽佩戴检测算法 [J]. 计算机工程与应用, 2023, 4 (1): 1-12.
- [3] 梁 循, 翁小林, 李嘉伟, 等. 基于改进 YOLOv5 的安全帽检测算法 [J]. 科技创新与应用, 2023, 13 (6): 81-84.
- [4] GIRSHICK R. Fast r-cnn [C] //Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.
- [5] REN S, HE K, GIRSHICK R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks [J]. Advances in neural information processing systems, 2015, 9199 (10.5555): 2969239-2969250.
- [6] LIU W, ANGUELOV D, ERHAN D, et al. Ssd: Single shot multibox detector [C] //Computer Vision-ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016: 21-37.
- [7] 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.

- [8] CHEN S, TANG W, JI T, et al. Detection of safety helmet wearing based on improved faster R-CNN [C] //2020 International joint conference on neural networks (IJCNN). IEEE, 2020: 1-7.
- [9] RAO C, HE X N, ZHENG Z L, et al. Multi-scale safety helmet detection based on SAS-YOLOv3-tiny [J]. Applied Sciences, 2021, 11 (8): 3652.
- [10] WE F, JIN G, GAO M, et al. Helmet detection based on improved YOLO V3 deep model [C] //2019 IEEE 16th International conference on networking, sensing and control (ICNSC). IEEE, 2019: 363-368.
- [11] CHEN J, DENG S, WANG P, et al. Lightweight Helmet Detection Algorithm Using an Improved YOLOv4 [J]. Sensors, 2023, 23 (3): 1256.
- [12] DAI B, NIE Y, CUI W, et al. Real-time safety helmet detection system based on improved SSD [C] //Proceedings of the 2nd International Conference on Artificial Intelligence and Advanced Manufacture. 2020: 95-99.
- [13] TAN S, LU G, JIANG Z, et al. Improved YOLOv5 network model and application in safety helmet detection [C] //2021 IEEE International Conference on Intelligence and Safety for Robotics (ISR). IEEE, 2021: 330-333.
- [14] SUN C, ZHANG S, QU P, et al. MCA-YOLOV5-Light: A faster, stronger and lighter algorithm for helmet-wearing detection [J]. Applied Sciences, 2022, 12 (19): 9697.
- [15] HAN K, WANG Y, TIAN Q, et al. Ghostnet: More features from cheap operations [C] //Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 1580-1589.
- [16] LI L, GAO Z, HUANG L, et al. A dual-modal face anti-spoofing method via light-weight networks [C] //2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID). IEEE, 2019: 70-74.
- [17] WANG K, LIEW J H, ZOU Y, et al. Panet: Few-shot image semantic segmentation with prototype alignment [C] //proceedings of The IEEE/CVF international conference on computer vision. 2019: 9197-9206.
- [18] TAN M, PANG R, LE Q V, et al. Efficientdet: Scalable and efficient object detection [C] //Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020: 10781-10790.
- [19] WANG K, LIEW J H, ZOU Y, et al. Panet: Few-shot image semantic segmentation with prototype alignment [C] //proceedings of the IEEE/CVF international conference on computer vision. 2019: 9197-9206.
- [20] KOLLÁR J. Flops [J]. Nagoya Mathematical Journal, 1989, 113: 15-36.

(下转第 175 页)