

控制与决策

Control and Decision

YOLOv5预测边界框分簇自适应损失权重改进模型

聂鹏, 肖欢, 喻聪

引用本文:

聂鹏,肖欢,喻聪. YOLOv5预测边界框分簇自适应损失权重改进模型[J]. *控制与决策*, 2023, 38(3): 645–653.

在线阅读 View online: <https://doi.org/10.13195/j.kzyjc.2021.1612>

您可能感兴趣的其他文章

Articles you may be interested in

[基于ResNet34_D改进YOLOv3模型的行人检测算法](#)

Pedestrian detection based on developed YOLOv3 with ResNet34_D

控制与决策. 2022, 37(7): 1713–1720 <https://doi.org/10.13195/j.kzyjc.2021.0136>

[基于深度学习的复杂背景下目标检测](#)

Target detection under complex background based on deep learning

控制与决策. 2022, 37(12): 3115–3121 <https://doi.org/10.13195/j.kzyjc.2021.0686>

[特征响应权重自适应的IoU网络跟踪算法改进](#)

Improvement of IoU network tracking with adaptive weighted characteristic responses

控制与决策. 2022, 37(7): 1752–1762 <https://doi.org/10.13195/j.kzyjc.2021.0148>

[基于两阶段深度网络的输电线路异常目标检测方法](#)

Transmission line abnormal object detection method based on deep network of two-stage

控制与决策. 2022, 37(7): 1873–1882 <https://doi.org/10.13195/j.kzyjc.2020.1840>

[多目标小尺度车辆目标检测方法](#)

Multi-target and small-scale vehicle target detection method

控制与决策. 2021, 36(11): 2707–2712 <https://doi.org/10.13195/j.kzyjc.2020.0635>

YOLOv5 预测边界框分簇自适应损失权重改进模型

聂鹏[†], 肖欢, 喻聪

(江西财经大学 软件与物联网工程学院, 南昌 330013)

摘要: 目标检测的精确程度是计算机视觉识别任务的主要影响因素. 针对单阶段目标检测模型 YOLOv5 存在的检测精度问题, 从多任务损失优化角度, 提出一种在不同分辨率特征图上基于同一目标的预测边界框分簇自适应损失权重改进模型. 该模型由 GT(ground true) 目标边界框 UID 分配器、GT 目标边界框 UID 匹配器、边界框位置及分类损失权重算法构成, 通过改善 YOLOv5 的位置精度和分类精度实现模型整体精度的提升. 实验结果表明, 改进模型的平均精度均值 (mean average precision, mAP) 较 YOLOv5.6 标准模型相对提升 5.23%; 相较于更为复杂的 YOLOv5x6 标准模型, 改进模型 mAP 取得 8.02% 的相对提升.

关键词: 计算机视觉; 目标检测; 深度学习; YOLOv5; 自适应损失权重; 多任务损失

中图分类号: TP301 文献标志码: A

DOI: 10.13195/j.kzyjc.2021.1612

引用格式: 聂鹏, 肖欢, 喻聪. YOLOv5 预测边界框分簇自适应损失权重改进模型 [J]. 控制与决策, 2023, 38(3): 645-653.

Enhanced self-adaptive loss weight YOLOv5 model based on predicted bounding boxes in clusters

NIE Peng[†], XIAO Huan, YU Cong

(School of Software and Internet of Things Engineering, Jiangxi University of Finance and Economics, Nanchang 330013, China)

Abstract: Object detection precision plays a critical role in computer vision tasks. Aiming at the precision problem available in the one-stage object detection model of the YOLOv5, this paper proposes an enhanced self-adaptive loss weight YOLOv5 model based on predicted bounding boxes in clusters presenting the individual targets in multi-resolution feature maps to optimize the multi-task loss. The enhanced model consists of the GT (ground true) target bounding box UID distributor, the GT target bounding box UID matcher, the bounding box position loss weight algorithm and the classification loss weight algorithm. The overall detection precision is improved by the enhancements of both position precision and classification precision in the YOLOv5. The experimental results present that compared with the YOLOv5.6, the mean average precision(mAP) is promoted relatively by 5.23% on average by the enhanced model which achieves the relative performance of 8.02% compared with the more complex model of YOLOv5x6.

Keywords: computer vision; object detection; deep learning; YOLOv5; self-adaptive loss weight; multi-task loss

0 引言

在计算机视觉研究中, 目标检测是一项关键性研究内容, 在文本检测、人脸识别、遥感目标识别等应用领域中发挥着重要的作用^[1-2], 目标检测的精确程度是影响上述计算机视觉识别任务的主要因素. 近年来, 随着卷积神经网络 (convolutional neural networks, CNN) 的兴起, 目标检测模型的研究也发生了显著的变化, 由传统的 Viola-Jones 算法 (Viola Jones, VJ)^[3]、HOG 算法 (histogram of oriented gradients, HOG)^[4]、DPM 算法 (deformable part model,

DPM)^[5] 演变到以 CNN 为核心的现代模型. CNN 现代目标检测模型一般可分为以 RCNN^[6]、SPPNet^[7]、FPN (feature pyramid networks)^[8] 为代表的两阶段和以 YOLO (you only look once) 系列^[9-12] 为代表的单阶段检测模型两个类别. 不同于两阶段模型将物体分类与位置进行分步求解, YOLO 基于图片中的细分网格将分类与位置求解问题统一为单一的回归问题, 显著提升了检测速度. 但相较于两阶段法, YOLO 牺牲了部分检测准确率, 原因可归纳为 4 个方面: 1) 数据集的类别失衡^[13-14]; 2) 数据集的尺度失衡^[15]; 3) 数据

收稿日期: 2021-09-15; 录用日期: 2022-01-11.

基金项目: 国家自然科学基金项目 (61866014).

[†]通讯作者. E-mail: niepeng@jxufe.edu.cn.

集的空间失衡^[16]; 4) 多任务损失优化失衡^[17].

针对YOLOv5模型中存在的检测精度问题,从多任务损失优化角度,本文对YOLOv5^[18]的6号升级版(简称YOLOv5.6)^[19]提出一种在不同分辨率特征图上基于同一目标的预测边界框自适应损失权重改进模型(ALW-YOLOv5, enhanced self-adaptive loss weight YOLOv5),通过改善YOLOv5.6的位置精度和分类精度实现模型整体精度的提升.

YOLOv5.6在训练过程中,随训练次数的增加, \mathcal{L}_{box} 和 \mathcal{L}_{cls} 损失相较于训练早期趋于平坦,而YOLOv5.6依赖于梯度计算更新自身网络模型. \mathcal{L}_{box} 或 \mathcal{L}_{cls} 在模型训练的中后期趋于平坦导致梯度过小,使得YOLOv5.6网络更新变慢,在给定训练周期内精度受限.基于上述分析,ALW-YOLOv5通过引入新的权重算法对 \mathcal{L}_{box} 与 \mathcal{L}_{cls} 损失分别进行调节,增加梯度力度,从而提升网络的整体预测精度.

ALW-YOLOv5主要由GT目标边界框UID(universal identifier)分配器、GT目标边界框UID匹配器、ALW-YOLOv5模型 \mathcal{L}_{box} 权重算法、ALW-YOLOv5模型 \mathcal{L}_{cls} 权重算法构成.GT目标边界框UID分配器负责对GT目标边界框进行全局唯一标记;GT目标边界框UID匹配器负责将不同分辨率特征图产生的预测边界框绑定到所对应的UID上; \mathcal{L}_{box} 权重算法用于分析不同分辨率下同一目标的预测边界框位置距离,对位置损失函数权重进行自适应调整,提升预测边界框的位置精度; \mathcal{L}_{cls} 权重算法用于分析不同分辨率下的目标分类距离,对分类损失函数权重进行自适应调整,提升预测边界框的分类精度.

在VOC2007标准数据集上进行模型对比实验验证,所得结果表明,ALW-YOLOv5模型较YOLOv5s6、YOLOv5m6、YOLOv5l6、YOLOv5x6四个YOLOv5.6模型在mAP50指标上取得了4.3%、4.9%、4.8%、6.7%的平均性能改善.

1 相关工作

1.1 YOLOv5模型

YOLO模型由Redmon等^[9]于2016年首次提出,并于2017年^[10]、2018年^[11]以及Bochkovskiy等^[12]于2020年分别提出YOLOv2~YOLOv4三个改进版本;2020年,Jocher等^[18]提出了YOLOv4的改进模型YOLOv5,2021年又提出了YOLOv5.6^[19].YOLOv5.6模型工作过程分为四个部分:端输入、backbone、neck和prediction,具体模型框架如图1所示.

YOLOv5.6模型中:端输入负责对输入数据进行预处理,包括自适应图片缩放、Mosaic数据增强、自适

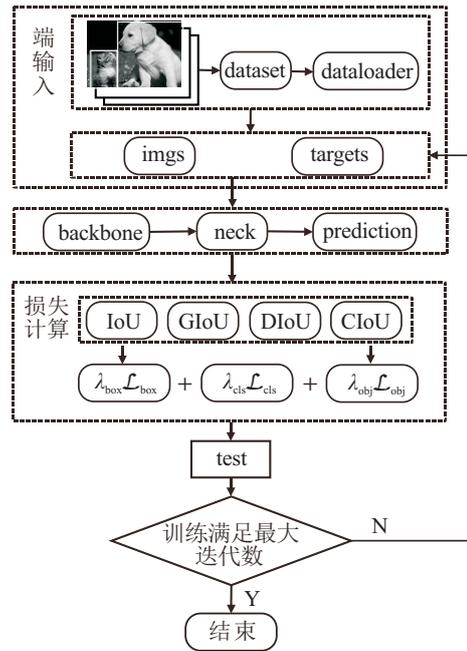


图1 YOLOv5.6标准模型框架

应锚框计算、输出训练图片数据集及目标;backbone采用CSP^[20]+SPPF结构;neck为FPN+PAN结构,输出4组不同分辨率的融合特征图;prediction基于4组融合特征图进行卷积,输出不同分辨率特征图所对应的预测边界框;损失计算则根据预测边界框,通过多任务损失函数 $\mathcal{L} = \lambda_{\text{box}}\mathcal{L}_{\text{box}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}} + \lambda_{\text{obj}}\mathcal{L}_{\text{obj}}$,选取IoU、GIoU、DIoU、CIoU不同策略进行损失计算;最后对模型进行测试.

1.2 \mathcal{L}_{box} 损失函数

YOLOv5.6模型中 \mathcal{L}_{box} 损失函数为

$$\mathcal{L}_{\text{box}} = \sum_{l=1}^L (1 - \text{mean}(f_{\text{xIoU}}(\text{box}_{\text{pre}}^{l,1}, \text{box}_{\text{gt}}^{l,1}); \dots, f_{\text{xIoU}}(\text{box}_{\text{pre}}^{l,M}, \text{box}_{\text{gt}}^{l,M}))). \quad (1)$$

其中: L 为金字塔层数, M 为金字塔第 l 层所对应的边界框数.

基于IoU及其变体计算各层预测边界框与真实边界框之间的边框回归损失,并向CNN网络提供位置学习任务的梯度变化.IoU的变体为

$$f_{\text{xIoU}} = \begin{cases} \frac{\text{box}_{\text{pre}}^{l,m} \cap \text{box}_{\text{gt}}^{l,m}}{\text{box}_{\text{pre}}^{n,m} \cup \text{box}_{\text{gt}}^{n,m}}, & \text{xIoU} = \text{IoU}; \\ f_{\text{IoU}} - \frac{|C|/(\text{box}_{\text{pre}}^{l,m} \cup \text{box}_{\text{gt}}^{l,m})|}{|C|}, & \text{xIoU} = \text{GIoU}; \\ f_{\text{IoU}} - \frac{\rho^2(\text{box}_{\text{pre}}^{l,m}, \text{box}_{\text{gt}}^{l,m})}{c^2}, & \text{xIoU} = \text{DIoU}; \\ f_{\text{IoU}} - \left(\frac{\rho^2(\text{box}_{\text{pre}}^{l,m}, \text{box}_{\text{gt}}^{l,m})}{c^2} + \alpha v \right), & \text{xIoU} = \text{CIoU}. \end{cases} \quad (2)$$

其中 $m \in [1, M]$.

1.3 \mathcal{L}_{cls} 损失函数

YOLOv5.6模型中 \mathcal{L}_{cls} 损失函数如下式所示:

$$\text{loss}_{cls}^{l,m} = -(\text{cls}_{gt}^{l,m} \times \log(\delta(\text{cls}_{pre}^{l,m})) + (1 - \text{cls}_{gt}^{l,m}) \times \log(1 - \delta(\text{cls}_{pre}^{l,m}))), \quad (3)$$

$$\mathcal{L}_{cls} = \sum_{l=1}^L (\text{mean}(\text{loss}_{cls}^{l,1}, \text{loss}_{cls}^{l,2}, \dots, \text{loss}_{cls}^{l,M})). \quad (4)$$

基于预测分类向量 sigmoid 值计算预测分类与真实分类的交叉熵损失,并向 CNN 网络提供分类学习任务的梯度变化.

1.4 \mathcal{L}_{obj} 损失函数

YOLOv5.6对预测边界框内是否存在物体进行概率判断,即预测边界框置信度. YOLOv5.6模型的 \mathcal{L}_{obj} 损失函数如下式所示:

$$\text{loss}_{obj}^{l,m} = -(\text{obj}_{gt}^{l,m} \times \log(\delta(\text{obj}_{pre}^{l,m})) + (1 - \text{obj}_{gt}^{l,m}) \times \log(1 - \delta(\text{obj}_{pre}^{l,m}))), \quad (5)$$

$$\mathcal{L}_{obj} = \sum_{l=1}^L (\text{mean}(\text{loss}_{obj}^{l,1}, \text{loss}_{obj}^{l,2}, \dots, \text{loss}_{obj}^{l,M})). \quad (6)$$

利用式(5)和(6)所示的 \mathcal{L}_{obj} 损失函数, YOLOv5.6计算预测边界框与真实边界框的置信度 sigmoid 交叉熵损失,并向网络返回 \mathcal{L}_{obj} , CNN 基于 \mathcal{L}_{obj} 获取网络更新所需的梯度变化.

2 ALW-YOLOv5模型

本文提出一种在不同分辨率特征图上基于同一目标的预测边界框分簇自适应损失改进模型 ALW-

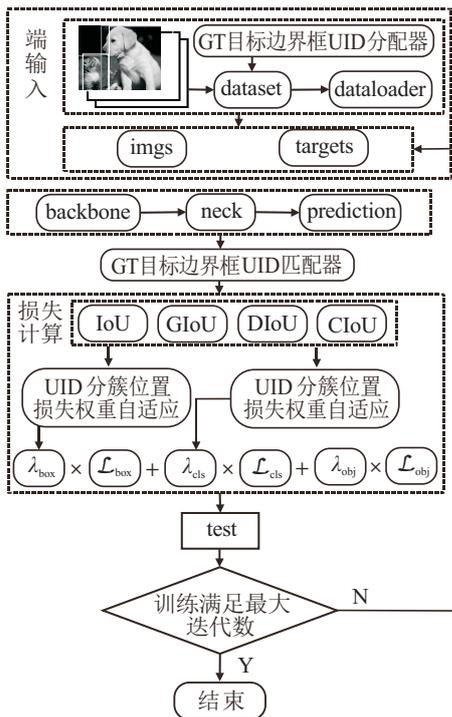


图2 ALW-YOLOv5改进模型框架

YOLOv5,其框架如图2所示.

ALW-YOLOv5在YOLOv5.6模型的基础上构建了GT目标边界框UID分配器、GT目标边界框UID匹配器、ALW-YOLOv5模型 \mathcal{L}_{box} 权重算法和ALW-YOLOv5模型 \mathcal{L}_{cls} 权重算法. 基于GT边界框,ALW-YOLOv5对不同分辨率特征图的预测边界框进行分簇,并在簇内进行自适应损失权重调节,从而进一步改善YOLOv5.6模型的位置精度和分类精度.

2.1 符号说明

box, cls, obj: 边界框位置信息、边界框分类信息、边界框物体置信信息;

$\lambda_{box}, \lambda_{cls}, \lambda_{obj}$: 多任务损失函数权重;

$\mathcal{L}_{box}, \mathcal{L}_{cls}, \mathcal{L}_{obj}$: 多任务损失函数分量;

f_{xIoU} : IoU, GIoU, DIoU, CIoU;

pre, gt, cluster: 预测值、真实值、聚集簇;

UID: GT目标边界框标记全局唯一标识;

L, l : 金字塔层数, $l \in [1, L]$;

M, m : M 为金字塔第 l 层所对应的边界框数, $m \in [1, M]$;

loss, δ : 预测框损失, sigmoid函数;

K, k : K 为与uid相关的预测框 $\text{box}_{pre}^{l,k}$ 的数量, $K \leq M, k \in [0, K]$;

$\text{label}_{img}^{\text{index}}$: 图像集合 dataset 中第 img 个图片内的第 index 个真实框标记信息;

$\text{uid}_{img}^{\text{index}}, \text{img} \in [1, \text{IMG}], \text{index} \in [1, N]$: gt目标边界框标记全局唯一标识UID, IMG为数据集图片数, N 为单张图片所含真实框数量;

$\text{box}_{gt}^{\text{index, img}}$: 图像集合 dataset 中第 img 个图片内的第 index 个真实框信息;

$(cx, cy, w, h), (\text{left}, \text{top}, \text{right}, \text{bottom})$: 边界框中心点 x 轴、中心点 y 轴、宽度、高度坐标与边界框左上角、右下角坐标;

$(cx_{gt}^{\text{index, img}}, cy_{gt}^{\text{index, img}}, w_{gt}^{\text{index, img}}, h_{gt}^{\text{index, img}})$: 第 img 个图片内第 index 个 gt 目标边界框的相对中心点坐标与相对宽高;

$\text{pair}_{uid}^{l,m}$: 预测框与真实框UID配对集合;

$xIoU_{l,n}^{\text{uid}}$: 预测框的 f_{xIoU} 值.

2.2 GT目标边界框UID分配器

在ALW-YOLOv5模型中,预测边界框在簇内的 f_{xIoU} 聚集度是 \mathcal{L}_{box} 与 \mathcal{L}_{cls} 自适应权重的计算基础,而该框聚集度对于每个GT目标边界框均是个性化的,因此,必须对GT目标边界框进行全局唯一UID标记,以便于预测边界框根据集合 $\text{box}_{cluster}^{l, \text{uid}} =$

$\{\text{box}_{\text{pre}}^{l,k} | k \in [1, K]\}$ 对特定分辨率特征图的预测边界框簇进行识别. 其中: $\text{uid} \in \text{UID}$, $l \in L$, K 为与 uid 相关的预测边界框 $\text{box}_{\text{pre}}^{l,k}$ 的数量. GT 目标边界框 UID 分配器算法如算法 1 所示.

算法 1

initialize gtLabels by imagePath and batchSize

initialize $\text{UID}_{\text{box}} = \{\}$

foreach $\text{label}_{\text{img}}^{\text{index}}$ in gtLabels do

analyze $\text{label}_{\text{img}}^{\text{index}}$

and create ground true bounding box $\text{box}_{\text{gt}}^{\text{index, img}}$

create universal identifier $\text{uid}_{\text{img}}^{\text{index}}$

of $\text{box}_{\text{gt}}^{\text{index, img}}$ by hash(normalize($\text{box}_{\text{gt}}^{\text{index, img}}$))

$\text{UID}_{\text{box}} = \bigcup_{\text{img}=1}^{\text{IMG}} \bigcup_{\text{index}=1}^N (\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}})$

GT 目标边界框 UID 分配算法首先对 imagePath 路径中的 gt 标签进行加载, 并按照 batchSize 对标签进行样本分组, 生成与 gt 输入图片数据集批次相匹配的标签集合 $\text{gtLabel} = \bigcup_{\text{img}=1}^{\text{IMG}} \{\text{label}_{\text{img}}^1, \text{label}_{\text{img}}^2, \dots, \text{label}_{\text{img}}^N\}$, 并初始化 UID_{box} ; 随后算法对 $\text{label}_{\text{img}}^{\text{index}}$ 进行解析, 构建边界框 $\text{box}_{\text{gt}}^{\text{index, img}}$.

算法再对 $\text{box}_{\text{gt}}^{\text{index, img}} = ((\text{cx}|\text{left})_{\text{gt}}^{\text{index, img}}, (\text{cy}|\text{top})_{\text{gt}}^{\text{index, img}}, (\text{w}|\text{right})_{\text{gt}}^{\text{index, img}}, (\text{h}|\text{bottom})_{\text{gt}}^{\text{index, img}})$ 进行归一化处理, 生成 $(\text{cx}\%_{\text{gt}}^{\text{index, img}}, \text{cy}\%_{\text{gt}}^{\text{index, img}}, \text{w}\%_{\text{gt}}^{\text{index, img}}, \text{h}\%_{\text{gt}}^{\text{index, img}})$, 并基于此计算归一化向量的 hash 值得到 $\text{uid}_{\text{img}}^{\text{index}}$. 最后, 将 $\text{uid}_{\text{img}}^{\text{index}}$ 与 $\text{box}_{\text{gt}}^{\text{index, img}}$ 联合构成向量 $(\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}})$, 完成 UID 分配.

2.3 GT 目标边界框 UID 匹配器

YOLOv5.6 在 neck 中首先使用 backbone 下采样生成不同分辨率的特征图; 然后利用特征金字塔 FPN 上采样, 自顶向下将高层特征信息传递到下层; 最终得到多分辨率下的融合特征图, 并生成不同分辨率下的预测边界框 $\text{box}_{\text{pre}}^{l,m}$. ALW-YOLOv5 进一步将 $(\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}})$ 匹配到 $\text{box}_{\text{pre}}^{l,m}$ 上, 为自适应损失权重计算提供依据. GT 目标边界框 UID 匹配器算法如算法 2 所示.

算法 2

load $\bigcup_{l=1}^L \bigcup_{m=1}^M \text{box}_{\text{pre}}^{l,m}$,

$\text{UID}_{\text{box}} = \bigcup_{\text{img}=1}^{\text{IMG}} \bigcup_{\text{index}=1}^N (\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}})$

expand UID_{box} to $\{\text{UID}_{\text{box}}^{\text{layer1}}, \dots, \text{UID}_{\text{box}}^{\text{layer4}}\}$,

adapting to the FPN layers

foreach FPN layer $l \in L$ do

foreach $\text{box}_{\text{pre}}^{l,m}$ in layer l do

filter $(\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}})$ in UID_{box} with aspect ratio

assign $\text{box}_{\text{gt}}^{\text{index, img}}$ to

$(\text{box}_{\text{gt},1}^{\text{index, img}}, \text{box}_{\text{gt},2}^{\text{index, img}}, \dots, \text{box}_{\text{gt},4}^{\text{index, img}})$

expand $(\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}})$ to

$(\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}})$ expand =

$\bigcup_{p=1}^4 (\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}})_p$

match $(\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}})$ expand to

$\text{box}_{\text{pre}}^{l,m}$ by cell coordinate (gj, gi)

GT 目标边界框 UID 匹配器首先加载预测边界框与 gt 目标边界框的全局唯一标识 UID; 然后将 UID 扩展至 FPN 结构各层, 与各层中的预测边界框进行匹配. $(\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}})$ 在匹配过程中剔除真实边界框中的极端长宽比, 增强有效数据比重. 由于 YOLOv5.6 将 gt 边界框扩展至邻居网格坐标, $(\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}})$ 需进一步进行适配扩展, 得到与扩展后 $\text{box}_{\text{gt}}^{\text{index, img}}$ 数量一致的 $(\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}})$ expand. 最后, 通过网格坐标 (gj, gi) 将 $(\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}})$ 匹配到 FPN 不同分辨特征图所对应的预测边界框 $\text{box}_{\text{pre}}^{l,m}$ 上, 构建出 UID 配对集合 $\text{pair}_{\text{uid}}^{l,m}((\text{box}_{\text{gt}}^{\text{index, img}}, \text{uid}_{\text{img}}^{\text{index}}), \text{box}_{\text{pre}}^{l,m})$.

2.4 \mathcal{L}_{box} 权重算法

ALW-YOLOv5 模型改进了原 YOLOv5.6 中基于金字塔特征图尺寸和目标边界框数量的权重调节算法, 提出一种利用目标边界框 UID 对预测边界进行分簇, 并基于簇内预测边界框 f_{xIoU} 聚集度自动调节损失权重的自适应算法, 具体如算法 3 所示.

算法 3

fetch $f_{\text{xIoU}}, \text{xIoU} \in \{\text{IoU}, \text{GIoU}, \text{DIoU}, \text{CIoU}\}$

attach $\text{pair}_{\text{uid}}^{l,m}$ to f_{xIoU}

foreach FPN layer $l \in L$ do

create $\text{cluster}_l^{\text{uid}} = \bigcup_{l=1, m=1}^{L, M} \text{box}_{\text{pre}}^{l,m}, \text{box}_{\text{pre}}^{l,m} \in$

$\text{pair}_{\text{uid}}^{l,m}$,

$\text{uid} = \text{uid}_{\text{img}}^{\text{index}}$

foreach $\text{cluster}_l^{\text{uid}}$ in layer l do

rank each $\text{xIoU}_{l,n}^{\text{uid}}$ in $\text{cluster}_l^{\text{uid}}$ by formula (7)

calculate loss weight $\lambda_{\text{box}}^{l,m}$ by formula (8)

$\lambda_{\text{box}} = \bigcup_{l=1, m=1}^{L, M} \lambda_{\text{box}}^{l,m}$,

$\text{rank}_{\text{box}}^{l,m} =$

$$2 - \text{sigmoid}(\text{order}_{\text{desc}}(\text{xIoU}_{l,n}^{\text{uid}} \text{ in cluster}_l^{\text{uid}})), \quad (7)$$

$$\lambda_{\text{box}}^{l,m} = \text{rank}_{\text{box}}^{l,m} \times \exp\left(\frac{\text{epoch}}{\text{epochs}} \times \text{layerIndex}\right),$$

$$\text{layerIndex} \in \{1, 2, 3, 4\}. \quad (8)$$

算法首先提取 YOLOv5.6 的 f_{xIoU} 结果, 并将 f_{xIoU} 与 $\text{pair}_{\text{uid}}^{l,m}$ 进行关联; 然后, 基于 UID 对预测边界框 $\text{box}_{\text{pre}}^{l,m}$ 按所属特征图层进行分簇, 在具体簇 $\text{cluster}_l^{\text{uid}}$ 内部对 $\text{box}_{\text{pre}}^{l,m}$ 基于 $\text{xIoU}_{l,n}^{\text{uid}}$ 排序, 并根据式 (7) 计算预测边界框的位置聚集度 $\text{rank}_{\text{box}}^{l,m}$; 最后, 结合网络学习迭代进度将权重自动适配到不同的特征图层的位置损失上, 如式 (8) 所示。

2.5 \mathcal{L}_{cls} 权重算法

不同于 YOLOv5.6 在第 i 次训练迭代中无差别地对金字塔同层特征图得到的预测边界框应用统一的 \mathcal{L}_{cls} 分类损失权重, ALW-YOLOv5 模型基于 UID 对预测边界框进行分簇, 并在簇内计算分类损失 $\text{loss}_{\text{cls}}^{l,m}$, 然后根据簇内 $\text{loss}_{\text{cls}}^{l,m}$ 大小向网络反馈自适应权重 \mathcal{L}_{cls} , 具体如算法 4 所示。

算法 4

```
reuse clusterluid =  $\bigcup_{l=1, m=1}^{L, M} \text{box}_{\text{pre}}^{l,m}$ 
foreach FPN layer  $l \in L$  do
  foreach clusterluid in layer  $l$  do
    rank each  $\text{loss}_{\text{cls}}^{l,m}$  in clusterluid by formula (9)
    calculate loss weight  $\lambda_{\text{cls}}^{l,m}$  by formula (10)
 $\lambda_{\text{cls}} = \bigcup_{l=1, m=1}^{L, M} \lambda_{\text{cls}}^{l,m}$ ,
rankclsl,m =
0.5 + sigmoid(orderasc( $\text{loss}_{\text{cls}}^{l,m}$  in clusterluid)), (9)
 $\lambda_{\text{cls}}^{l,m} = \text{rank}_{\text{cls}}^{l,m} \times \exp\left(\frac{\text{epoch}}{\text{epochs}} \times \text{layerIndex}\right)$ ,
layerIndex  $\in \{1, 2, 3, 4\}$ . (10)
```

算法复用 $\text{cluster}_l^{\text{uid}}$, 在具体簇 $\text{cluster}_l^{\text{uid}}$ 内部对 $\text{box}_{\text{pre}}^{l,m}$ 基于 $\text{loss}_{\text{cls}}^{l,m}$ 排序并根据式 (9) 计算预测边界框的分类聚集度 $\text{rank}_{\text{cls}}^{l,m}$, 将其通过 sigmoid 激活后调整至范围 [1, 1.5], 最终结合网络学习迭代进度将权重自动适配到不同的特征图层上, 如式 (10) 所示。

3 实验与结果

3.1 实验设置

将 ALW-YOLOv5 模型在目标检测公开数据集 VOC2007 上进行分组实验, 对比了标准模型 YOLOv5s6、YOLOv5m6、YOLOv5l6、YOLOv5x6 与 ALW-YOLOv5 模型在采用 IoU、GIoU、DIoU 与 CIoU

等不同计算策略时的指标差异。

VOC2007 包含 9963 张标注图片, 含 20 个类别, 24 640 个已标注物体, 由 train、val、test 三部分组成, 分别含图片 2 501 张、2 510 张与 4 952 张。

YOLOv5s6、YOLOv5m6、YOLOv5l6、YOLOv5x6 为 YOLOv5.6 模型中的 4 个公开标准版本, 其共同拥有的网络基本特征为 6 个 Concat、12 个 Conv、3 个 Upsample、1 个 SPPF、1 个 Detect, 共有特征不对各结构内部的子结构进行重复计数 (如 BottleneckCSP 中的 Conv 数量不再重复统计到网络基本特征中), 差异性特征如表 1 所示。

表 1 YOLOv5s6、YOLOv5m6、YOLOv5l6、YOLOv5x6 差异性特征表

	YOLOv5s6	YOLOv5m6	YOLOv5l6	YOLOv5x6
深度系数	0.33	0.67	1.0	1.33
宽度系数	0.50	0.75	1.0	1.25
BottleneckCSP 数量	14	28	42	56

实验环境为 Ubuntu 20.04、CUDA 11.0.207、GeForce RTX 2080 Ti。

3.2 评价指标

YOLOv5.6 属于多目标检测任务, 其分类标签不止一个, 因此, 不能使用普通单标签图像分类的标准 mean accuracy; 其次, 仅使用 accuracy 时, 当不同类别样本比例极端失衡时, 占比大的类别将主导 accuracy, 从而导致得分较高的模型实际效果很差。

平均精度均值 (mean average precision, mAP) 综合考虑了目标检测的精确率 (precision) 与召回率 (recall)。精确率与召回率之间存在博弈关系, 较好的模型可以同时具备较高的精确率和召回率。基于 P-R (precision-recall) 曲线求取所有目标类别的平均精度均值 mAP, 可对目标检测模型进行更全面的评价。

ALW-YOLOv5 模型也采用 mAP 和 P-R 曲线作为本模型与标准模型的主要对比评价指标, 有

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (11)$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (12)$$

$$P_c^{\text{smooth}}(r) = \max_{r' \geq r} P(r'), \quad (13)$$

$$\text{AP}_c = \sum_{n=0}^{\text{interpoints}-1} I_c(n) / \text{interpoints}, \quad (14)$$

$$\text{mAP} = \sum_{c=1}^{\text{NC}} \text{AP}_c / \text{NC}. \quad (15)$$

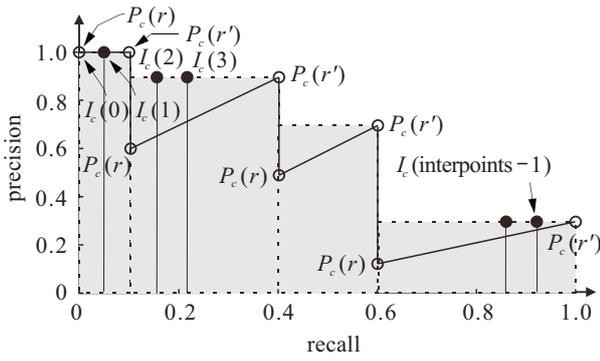
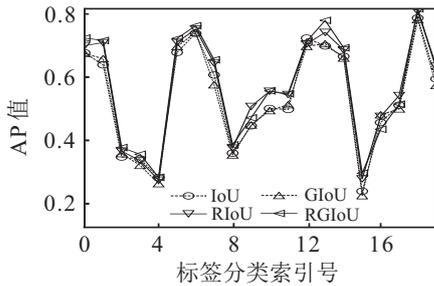


图3 P-R曲线平滑处理示意

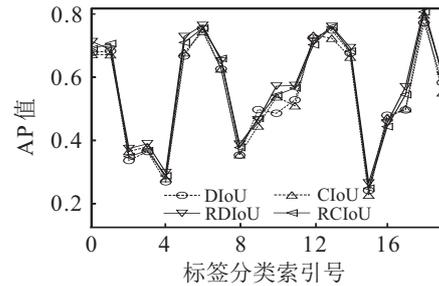
mAP指标首先在具体类别 C 中根据 precision 和 recall 计算 P-R 曲线;然后依据式(13)对曲线进行如图3所示的平滑处理,并在此基础上根据式(14)计算分类 C 的 AP_c 。其中: interpoints 为插值数量, mAP 为全部分类 AP_c 的均值, NC 为数据集包含的标签类别数。

3.3 实验结果分析

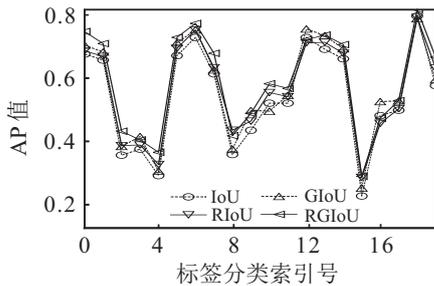
VOC2007数据集所含20个分类以标签分类索引号的形式标记为0~19,图4展示了ALW-YOLOv5模型(R字头代表的曲线)与YOLOv5.6模型在使用不同IoU计算策略时在各分类上的AP指标值。



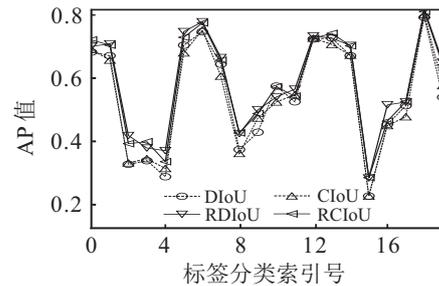
(a) 基于IoU、GIoU标准模型与改进模型对比(YOLOv5s6)



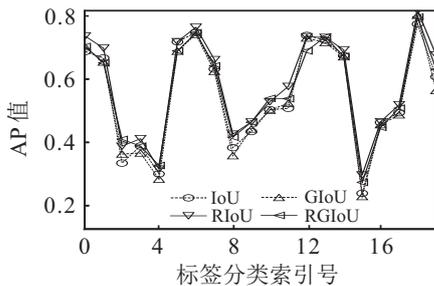
(b) 基于DIoU、CIoU标准模型与改进模型对比(YOLOv5s6)



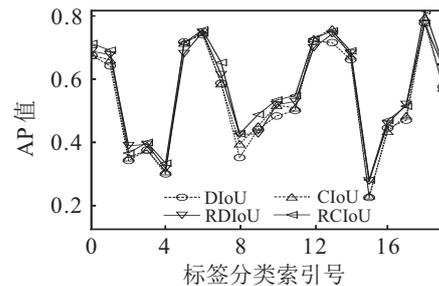
(c) 基于IoU、GIoU标准模型与改进模型对比(YOLOv5m6)



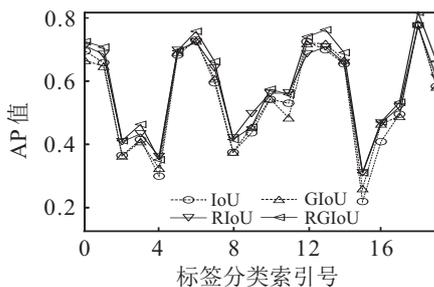
(d) 基于DIoU、CIoU标准模型与改进模型对比(YOLOv5m6)



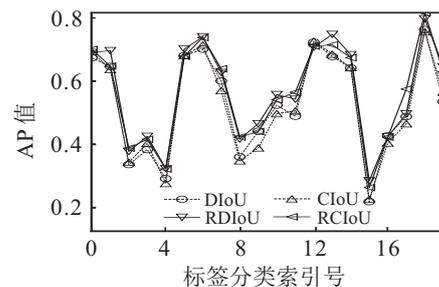
(e) 基于IoU、GIoU标准模型与改进模型对比(YOLOv5l6)



(f) 基于DIoU、CIoU标准模型与改进模型对比(YOLOv5l6)



(g) 基于IoU、GIoU标准模型与改进模型对比(YOLOv5x6)



(h) 基于DIoU、CIoU标准模型与改进模型对比(YOLOv5x6)

图4 YOLOv5.6与ALW-YOLOv5模型各分类AP对比

实验表明: ALW-YOLOv5 模型有 14 个类别优于 YOLOv5s6-IoU 和 YOLOv5s6-DIoU, 16 个类别优于 YOLOv5s6-GIoU, 15 个类别优于 YOLOv5s6-CIoU; 17 个类别优于 YOLOv5m6-IoU, 17 个类别优于 YOLOv5m6-GIoU 和 YOLOv5m6-DIoU, 16 个类别优于 YOLOv5m6-CIoU; 18 个类别优于 YOLOv5l6-IoU, 17 个类别优于 YOLOv5l6-GIoU, 20 个类别优于 YOLOv5l6-DIoU 和 YOLOv5l6-CIoU; 20 个类别优于 YOLOv5x6-IoU 和 YOLOv5x6-CIoU, 19 个类别优于 YOLOv5x6-GIoU, 18 个类别优于 YOLOv5x6-DIoU.

ALW-YOLOv5 模型在采用 IoU 计算策略时,

在 bottle 分类中取得了最大相对优势, AP 值提升 39.23%; 在 YOLOv5m6 采用 Diou 计算策略时, 在 cow 分类中取得了最大绝对优势, AP 值由 0.379 提升至 0.472. 总体而言, ALW-YOLOv5 模型对各分类 AP 提升大于 10% 的占 26.42%, 较标准模型有显著提升.

表 2 给出了 ALW-YOLOv5 模型与 YOLOv5.6 模型在全部分类的 mAP50 指标实验对比结果, 其中 RIoU 至 RCIoU 各值表达形式为 $\text{value}_{\text{lcls}}^{\text{lbox}}$. value 表示 \mathcal{L}_{box} 与 \mathcal{L}_{cls} 权重算法联合工作时获得的 mAP50 精度, lbox、lcls 分别表示 \mathcal{L}_{box} 与 \mathcal{L}_{cls} 权重算法独立工作时获得的精度.

表 2 YOLOv5.6 模型与 ALW-YOLOv5 模型 mAP50 对比表

模型	标准模型				ALW-YOLOv5 改进模型							
	IoU	GIoU	DIoU	CIoU	RIoU	RGIoU	RDIoU	RCIoU				
YOLOv5s	0.517 1	0.521 5	0.528 3	0.525 3	0.542 2	$\frac{0.524}{0.534}$	0.545 5	$\frac{0.528}{0.537}$	0.553 1	$\frac{0.537}{0.542}$	0.542 1	$\frac{0.532}{0.533}$
YOLOv5m	0.525 6	0.548 9	0.542 0	0.543 1	0.554 2	$\frac{0.536}{0.542}$	0.574 2	$\frac{0.556}{0.566}$	0.571 0	$\frac{0.552}{0.559}$	0.566 3	$\frac{0.552}{0.555}$
YOLOv5l	0.543 6	0.534 8	0.532 8	0.541 6	0.570 6	$\frac{0.555}{0.559}$	0.558 6	$\frac{0.542}{0.551}$	0.557 6	$\frac{0.542}{0.548}$	0.571 1	$\frac{0.551}{0.561}$
YOLOv5x	0.521 9	0.523 8	0.533 0	0.534 8	0.552 8	$\frac{0.534}{0.539}$	0.565 7	$\frac{0.543}{0.544}$	0.573 1	$\frac{0.550}{0.554}$	0.565 4	$\frac{0.544}{0.553}$

当 ALW-YOLOv5 模型仅采用 \mathcal{L}_{box} 权重算法时, 在 32 组对比实验中, \mathcal{L}_{box} 策略的 mAP50 最小提升 0.7% (相对提升 1.2%), 最大提升 1.9% (相对提升 3.6%), 平均提升 1.0%, 从而表明了 \mathcal{L}_{box} 策略的有效性.

当 ALW-YOLOv5 模型仅采用 \mathcal{L}_{cls} 权重算法时, \mathcal{L}_{cls} 策略的 mAP50 最小提升 0.8% (相对提升 1.5%), 最大提升 2.1% (相对提升 3.9%), 平均提升 1.6%, 同样表明了 \mathcal{L}_{cls} 策略的有效性, 同时显示出分类损失调节更有益于模型精度的提升.

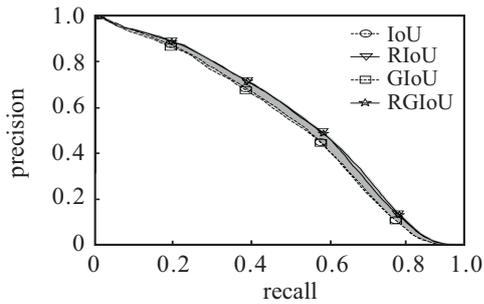
在采用 \mathcal{L}_{box} 、 \mathcal{L}_{cls} 权重策略联合工作的 32 组对比实验中, ALW-YOLOv5 较标准模型在 mAP50 上最小提升 1.7% (相对提升 3.2%), 最大提升 4.2% (相对提升 8.0%), 平均提升 2.8%, 表明 \mathcal{L}_{box} 与 \mathcal{L}_{cls} 权重算法联合工作时较独立实施时能更有效地改善模型 mAP50 精度.

在整体指标提升区段中: $1\% \leq \text{mAP50} < 2\%$ 区段为 1 个 YOLOv5s6 模型; $2\% \leq \text{mAP50} < 4\%$ 区段为 3 个 YOLOv5s6 模型, 4 个 YOLOv5m6, 4 个 YOLOv5l6, 2 个 YOLOv5x; $4\% \leq \text{mAP50} < 5\%$ 区段为 2 个 YOLOv5x. 实验数据表明, ALW-YOLOv5 模型伴随 YOLOv5 模型复杂度的提高, 显示出更好的准确性.

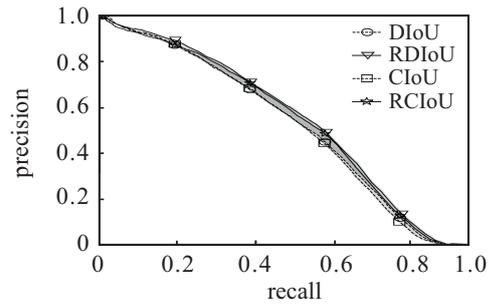
图 5 为 YOLOv5.6 模型与 ALW-YOLOv5 模型的 P-R 曲线对比, 可明显看出, 从简单 YOLOv5s6 到复杂 YOLOv5x6, ALW-YOLOv5 模型较 YOLOv5.6 模型的 P-R 曲线面积差异逐渐扩大, 显示出 ALW-YOLOv5 模型对于更加复杂的 YOLOv5.6 结构具有更优的性能提升.

4 结论

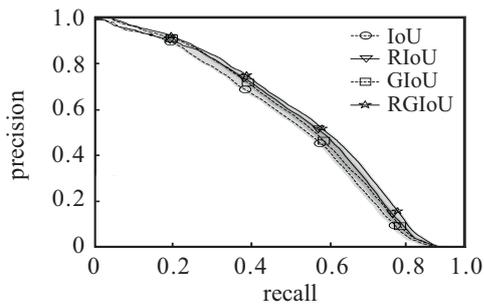
为了进一步提升 YOLOv5.6 模型的检测精度, 本文针对多任务损失权重进行自适应优化, 提出了改进的 ALW-YOLOv5 模型. ALW-YOLOv5 首先使用 GT 目标边界框 UID 分配器对 GT 目标边界框进行全局唯一标记, 并在网络生成预测张量之后, 通过 GT 目标边界框 UID 匹配器将不同分辨率特征图产生的预测边界框绑定到对应的 UID 上. 在此基础上, \mathcal{L}_{box} 权重算法与 \mathcal{L}_{cls} 权重算法分别负责针对不同分辨率下同一目标的预测边界框基于 UID 进行分簇, 并生成簇内位置损失权重与分类损失函数权重, 自适应调节预测边界框的位置与分类反馈梯度, 总体改善了 YOLOv5.6 的目标检测精度. 通过分组实验对比, ALW-YOLOv5 的目标检测精度较标准 YOLOv5.6 有了普遍的提升, 并随模型复杂度的增加, ALW-YOLOv5 表现出更显著的精度优势.



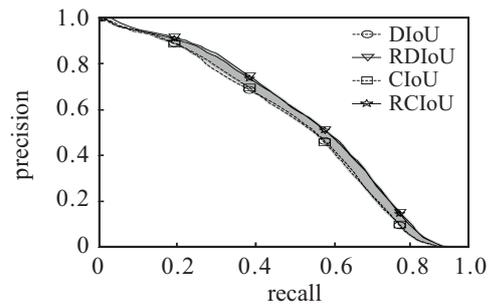
(a) YOLOv5s6(IoU/RIoU/GIoU/RGIoU) P-R 对比



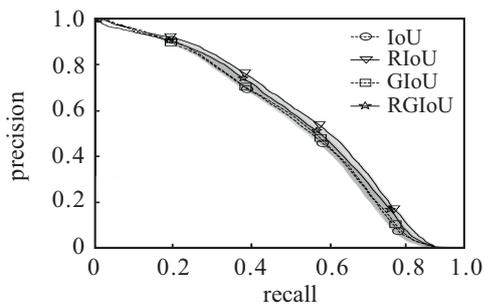
(b) YOLOv5s6(DIoU/RDIoU/CIoU/RCIoU) P-R 对比



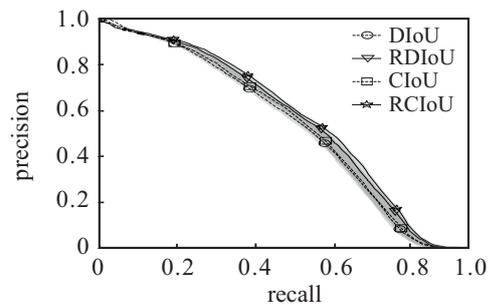
(c) YOLOv5m6(IoU/RIoU/GIoU/RGIoU) P-R 对比



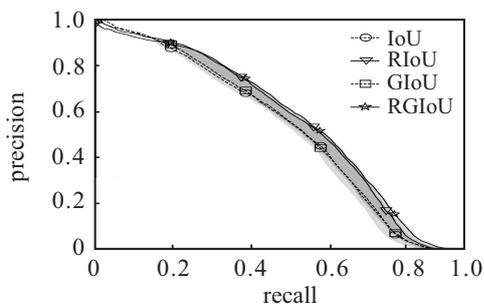
(d) YOLOv5m6(DIoU/RDIoU/CIoU/RCIoU) P-R 对比



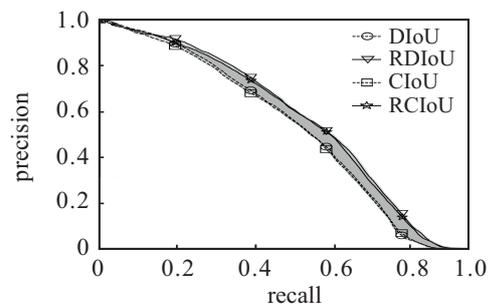
(e) YOLOv516(IoU/RIoU/GIoU/RGIoU) P-R 对比



(f) YOLOv516(DIoU/RDIoU/CIoU/RCIoU) P-R 对比



(g) YOLOv5x6(IoU/RIoU/GIoU/RGIoU) P-R 对比



(h) YOLOv5x6(DIoU/RDIoU/CIoU/RCIoU) P-R 对比

图5 YOLOv5.6与ALW-YOLOv5模型P-R曲线对比

参考文献(References)

- [1] 刘杰, 朱旋, 宋密密. 改进 YOLOv2 的端到端自然场景中文字检测[J]. 控制与决策, 2021, 36(10): 2483-2489.
(Liu J, Zhu X, Song M M. End-to-end Chinese character detection in natural scene based on improved YOLOv2[J]. Control and Decision, 2021, 36(10): 2483-2489.)
- [2] 安珊, 林树宽, 乔建忠, 等. 基于生成对抗网络学习被遮挡特征的目标检测方法[J]. 控制与决策, 2021, 36(5): 1199-1205.
(An S, Lin S K, Qiao J Z, et al. Object detection via learning occluded features based on generative adversarial networks[J]. Control and Decision, 2021, 36(5): 1199-1205.)
- [3] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]. Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Kauai, 2001: 511-518.
- [4] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]. 2005 IEEE Computer Society

- Conference on Computer Vision and Pattern Recognition. San Diego, 2005: 886-893.
- [5] Maji S, Berg A C, Malik J. Classification using intersection kernel support vector machines is efficient[C]. 2008 IEEE Conference on Computer Vision and Pattern Recognition. Anchorage, 2008: 1-8.
- [6] 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. Columbus, 2014: 580-587.
- [7] 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.
- [8] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, 2017: 936-944.
- [9] 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. Las Vegas, 2016: 779-788.
- [10] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, 2017: 6517-6525.
- [11] Redmon J, Farhadi A. YOLOv3: An incremental improvement[J/OL]. 2018, arXiv: 1804.02767.
- [12] Bochkovskiy A, Wang C Y, Liao H Y M. YOLOv4: Optimal speed and accuracy of object detection[J/OL]. 2020, arXiv: 2004.10934.
- [13] Li B Y, Liu Y, Wang X G. Gradient harmonized single-stage detector[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33: 8577-8584.
- [14] Oksuz K, Cam B C, Akbas E, et al. Generating positive bounding boxes for balanced training of object detectors[C]. 2020 IEEE Winter Conference on Applications of Computer Vision. Snowmass, 2020: 883-892.
- [15] Singh B, Davis L S. An analysis of scale invariance in object detection SNIP[C]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, 2018: 3578-3587.
- [16] Zhang X S, Wan F, Liu C, et al. Learning to match anchors for visual object detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(6): 3096-3109.
- [17] Chen J, Liu D, Xu T, et al. Is heuristic sampling necessary in training deep object detectors? [J]. IEEE Transactions on Image Processing, 2021, 30: 8454-8467.
- [18] Jocher G, Nishimura K, Mineeva T, et al. YOLOv5 (minor version 1.0) [EB/OL]. (2020-05-27) [2020-09-18]. <https://github.com/ultralytics/yolov5/releases/tag/v1.0>.
- [19] Jocher G, Nishimura K, Mineeva T, et al. YOLOv5 (minor version 6.0) [EB/OL]. (2021-10-12) [2021-11-05]. <https://github.com/ultralytics/yolov5/releases/tag/v6.0>.
- [20] Wang C Y, Mark Liao H Y, Wu Y H, et al. CSPNet: A new backbone that can enhance learning capability of CNN[C]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Seattle, 2020: 1571-1580.

作者简介

聂鹏(1977—),男,副教授,博士,从事计算机视觉、数字图像处理、目标检测等研究, E-mail: niepeng@jxufe.edu.cn;

肖欢(1997—),男,硕士生,从事计算机视觉、目标检测的研究, E-mail: 1767064404@qq.com;

喻聪(1996—),男,硕士生,从事计算机视觉、目标检测的研究, E-mail:1021988613@qq.com.

(责任编辑:李君玲)