doi:10.6041/j.issn.1000-1298.2025.06.024

基于 GSD - YOLO 的复杂场景仔猪检测和计数方法

曾志雄[1]　黄迎辉[1]　林　锴[2]　罗毅智[3,4]　吴灶铭[5]　吕恩利[1,6]

(1. 华南农业大学工程学院, 广州 510642; 2. 华南农业大学数学与信息学院, 广州 510642;
3. 广东省农业科学院设施农业研究所, 广州 510642; 4. 农业装备技术全国重点实验室, 广州 510642;
5. 华南农业大学电子工程学院, 广州 510642; 6. 猪禽种业全国重点实验室, 广州 510645)

摘要: 哺乳期仔猪因体型小、生性好动、易被遮挡且聚集重叠,人工盘点效率低且易出错。现有方法在复杂猪场环境和频繁遮挡下难以平衡检测准确率与资源受限部署需求,增加了漏检和误检风险。为此,本文提出了一种基于 YOLO v8n 的轻量化仔猪检测模型 GSD - YOLO。模型通过引入柔性非极大值抑制边界框交并操作和 Inner - MPDIoU 损失函数,优化边界框回归以降低误检和漏检率;并嵌入坐标注意力机制(Coordinate attention for efficient mobile network design,CA),增强了目标区域的特征表达能力,有效解决长程依赖问题。为实现嵌入式设备的高效部署,模型引入 GhostNet 模块优化特征提取和融合,减少通道间特征冗余的同时降低模型参数量。模型重构了一种轻量化的检测头 Detect_DG,在模型体积缩减 18.48% 的同时,进一步提升了检测精度。与 YOLO v8n 相比,GSD - YOLO 的 F1 分数和平均精度分别提升 1.0、0.6 个百分点,参数量降低 61.28%,帧率提高 12.5%。GSD - YOLO 在综合检测性能上优于 YOLO v11 等 4 种主流模型。结果表明,该模型在不同遮挡、重叠和光照下检测仔猪目标的准确率更优,且模型内存占有量较小,仅有 2.6 MB。将 GSD - YOLO 部署到边缘计算设备 Jetson Orin NX 和安卓(Android)移动端,为实际应用中的仔猪检测提供了有效的技术支撑。

关键词: 仔猪计数; 目标检测; YOLO v8n; 轻量化; 重度遮挡

中图分类号: TP391　　**文献标识码**: A　　**文章编号**: 1000-1298(2025)06-0247-11　　**OSID**:

Advanced Piglet Detection and Counting in Complex Scenarios Using GSD - YOLO Architecture

ZENG Zhixiong[1]　HUANG Yinghui[1]　LIN Kai[2]　LUO Yizhi[3,4]　WU Zaoming[5]　LÜ Enli[1,6]

(1. *College of Engineering*, *South China Agricultural University*, *Guangzhou* 510642, *China*
2. *College of Mathematics and Informatics*, *South China Agricultural University*, *Guangzhou* 510642, *China*
3. *Institute of Facility Agriculture*, *Guangdong Academy of Agricultural Sciences*, *Guangzhou* 510642, *China*
4. *State Key Laboratory of Agricultural Equipment Technology*, *Guangzhou* 510642, *China*
5. *College of Electronic Engineering*, *South China Agricultural University*, *Guangzhou* 510642, *China*
6. *State Key Laboratory of Animal and Poultry Breeding and Nutrition Research*, *Guangzhou* 510645, *China*)

Abstract: In modern pig farming, the detection and counting of piglets presents a significant challenge due to their small size, high mobility, tendency to be occluded, and habit of clustering. Manual counting is not only time-consuming and labor-intensive but also prone to errors. To address these issues, a novel piglet detection algorithm, GSD - YOLO, was proposed based on YOLO v8n, which was designed to provide fast and accurate detection. The algorithm tackled the inherent difficulty of detecting piglets by introducing flexible non-maximum suppression (FNMS) for bounding box intersection handling and employing the Inner - MPDIoU loss function to optimize the bounding box regression mechanism.

收稿日期: 2025 - 04 - 15　修回日期: 2025 - 04 - 29

基金项目: 广东省乡村振兴战略专项(2025TS - 3)、农业农村部华南现代农业智能装备重点实验室开放课题项目(HNZJ202209)、广州市重点研发计划项目(2023B03J1363)、广州市基础与应用基础研究项目(2023A04J0752)、汕尾市科技计划项目(2023A009)、猪禽种业全国重点实验室 PI 团队自主选题研究项目(2023QZ - NK16、GDNKY - ZQQZ - K13)、广东省农业科学院科技创新战略计划项目(ZX202402)、农业装备技术全国重点实验室(华南农业大学)开放课题项目(SKLAET - 202407)和广东省现代农业产业技术体系创新团队建设项目(2024CXTD02)

作者简介: 曾志雄(1989—),男,高级实验师,博士,主要从事智慧养殖技术装备研究,E-mail: zhixzeng@scau.edu.cn

通信作者: 罗毅智(1995—),男,助理研究员,博士,主要从事智慧农业装备、渔业多源信息获取和融合技术研究,E-mail: yizhi_luo@yeah.net

Additionally, a coordinate attention (CA) mechanism was integrated to enhance the feature representation of target areas, effectively resolving issues related to long-range dependencies. For efficient deployment on embedded devices, the model was further optimized for lightweight performance. Specifically, the backbone and neck components of the original model were replaced with the GhostNet module, which reduced both model parameters and feature redundancy in the channels. Furthermore, a lightweight detection head, Detect_DG, was introduced, which reduced the overall model size by 18.48% while simultaneously improving detection accuracy. Compared with YOLO v8n, GSD-YOLO achieved improvements of 1.0 and 0.6 percentage points in F1-score and average precision (AP), respectively, while reducing the model size by 61.28% and improving the frame rate by 12.5%. In comprehensive detection performance, GSD-YOLO outperformed four widely used models, including YOLO v5. Experimental results demonstrated that the proposed model can rapidly and accurately detect piglets under challenging conditions, such as occlusion, overlap, and varying lighting environments. Moreover, with a compact memory footprint of only 2.6 MB. The deployment of GSD-YOLO on edge computing devices such as the Jetson Orin NX and Android mobile devices provided effective technological support for piglet counting and detection in practical applications.

Key words: piglet counting; object detection; YOLO v8n; lightweight optimization; severe occlusion

0 引言

在猪场规模化养殖过程中,由于猪只死亡、出栏等因素,不同阶段的猪舍内猪只数量经常发生变化,尤其是哺乳断奶周期短,仔猪数量频繁波动,使得盘点任务繁琐且易出错。仔猪盘点对养殖管理至关重要,因为精确的数量监控不仅关系到养殖效益,也能及时发现仔猪的健康问题,进行有效干预。现代计算机视觉技术因其高效、非接触和实时性等优势,正逐步取代人工成为主流趋势[1]。但养殖场内的光线昏暗和摄像头污染常导致图像质量受损,且猪只动态行为和频繁遮挡,为计算机视觉技术的应用带来诸多挑战。由于仔猪体型小且活动频繁,相互间易遮挡,这对检测的精度和鲁棒性提出更高要求。

早期研究尝试结合形态学和区域生长法[2-3]来分割猪只与背景以完成检测任务,但在背景干扰和复杂遮挡场景下的效果不佳,且工作量大。近年来,深度学习技术凭借其强大的特征提取能力和高效的自动化处理效率,已在该领域取得显著成果。例如,高云等[4]基于卷积神经网络构建了一个仔猪识别模型 PigNet,对不同形态、粘连严重的大白仔猪进行分割,从而识别出猪只个体,但其分割准确率为 86.15%,且每幅图像处理时间为 2.12 s;沈明霞等[5]使用 YOLO v3,采取阈值分割与前景椭圆拟合的办法,根据椭圆长轴推理出猪只缺失部位,实现猪只定位并分辨其头尾位置。不过该方法需实施于 Matlab 平台上,再加上模型体积较大,限制了其实时嵌入应用;杨秋妹等[6]提出了一种基于 YOLO v5n 的猪只盘点算法,通过添加 SE-Net 注意力机制和对检测层进行多尺度特征融合处理,提高在复杂遮挡重叠场景下的精度与鲁棒性,实现了猪只盘点的功能。但未融合可视化数字化表达,用户无法迅速掌握栏中猪只总数;丁奇安等[7]通过改进 YOLO v5 的内部结构实现轻量化,并在 Jetson Nano 设备上实现仔猪目标检测,在光线比较差的条件下,仍然有漏检的问题;AHN 等[8]在猪只检测与计数邻域,通过改进 YOLO v4-Tiny 算法并应用在嵌入式系统上,该研究侧重于解决曝光问题,对数据集要求庞大,且检测速度有待提升。

针对猪只的检测与计数研究,仍面临以下问题:①现有研究多集中于保育期猪只,针对哺乳期仔猪检测计数的研究较少。而哺乳期的仔猪体型小,更容易被栏杆等设备遮挡,且遮挡面积大,外形上与母猪腿部类似,增加了漏检和误检的风险。②部分研究的数据集对于环境要求比较单一,通常仅在光照较好的条件下采集,忽略了实际养殖场复杂光照变化对模型泛化能力的影响。③部分研究在进行模型轻量化时,未能有效实现检测准确率与受限资源部署的平衡,导致检测性能无法满足实际需求。④部分研究在计数可视化效果上有待优化。

针对上述问题,本文以产房哺乳期仔猪为研究对象,在 YOLO v8n 的基础上,提出一种基于 YOLO v8n 的改进轻量化目标检测模型 GSD-YOLO。将 GSD-YOLO 分别部署到边缘计算设备 Jetson Orin NX 和安卓(Android)移动端,以满足养殖场对实时、高效检测系统的需求,确保仔猪检测模型能够在不同设备上高效稳定运行,提升管理效率。

1 数据集构建

1.1 数据来源

本文数据集来源于云南省的一个商业猪舍。如图 1 所示,通过在猪舍的正上方架设深度相机来进

行数据采集。相机采用英特尔的 D435i 型实感深度摄像头，采集图像分辨率为 640 像素 × 480 像素。猪舍整体设计为 6 列，每列均分为 10 个小单元，即最大容纳量为 60 头分娩母猪。每个小单元划分为区域 1 和区域 2，分别为母猪活动区与仔猪活动区。分娩母猪在区域 1 可以饮食、活动和哺乳，仔猪则可以在区域 2 进行锻炼和哺乳。床架采用热镀锌管焊接，配备食槽和仔猪食盘，且设有可移动环境监测节点。

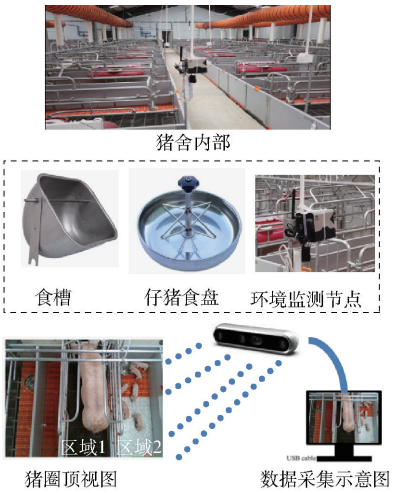

图 1　数据采集环境

Fig. 1　Data collection environment

1.2　数据集搭建

去除数据集中部分相似数据，最终获得 900 幅图像。利用 LabelImg 标注软件对图像中的仔猪进行人工标注，标签为“spig”。为了丰富数据集，让模型能学习到更多的特征，增强模型的泛化能力以及减少过拟合现象，增强模型的鲁棒性，进行了数据增强。具体方法包括从原始图像中随机抽取一定比例，对数据集进行添加高斯噪声、旋转、亮度调整等。部分数据增强的图像如图 2 所示。

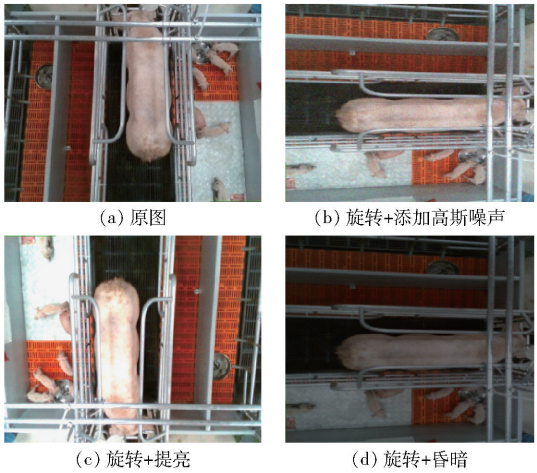

图 2　仔猪数据扩增图像

Fig. 2　Piglet data augmentation images

最终将 900 幅图像扩增到 2 500 幅。并按照比例 8∶1∶1 划分为训练集、验证集和测试集，分别为 2 000、250、250 幅图像。

2　GSD－YOLO 网络设计

2.1　整体架构

基于 YOLO v8 架构提出了一种改进的轻量化仔猪检测模型，改进后的模型结构如图 3 所示。首先在 Backbone 和 Neck 部分，引入 GhostConv 和 C3Ghost 模块优化传统的 Conv 和 C2f 模块。通过引入 GhostNet 网络的线性变换生成幻影特征图（Ghost features），有效降低了模型体积和计算复杂度。其次，在骨干网络末端的 3 个 C3Ghost 模块后嵌入 3 个 Coordinate Attention 模块。该注意力机制模块增强了模型对关键信息的关注能力。在参数占比较高

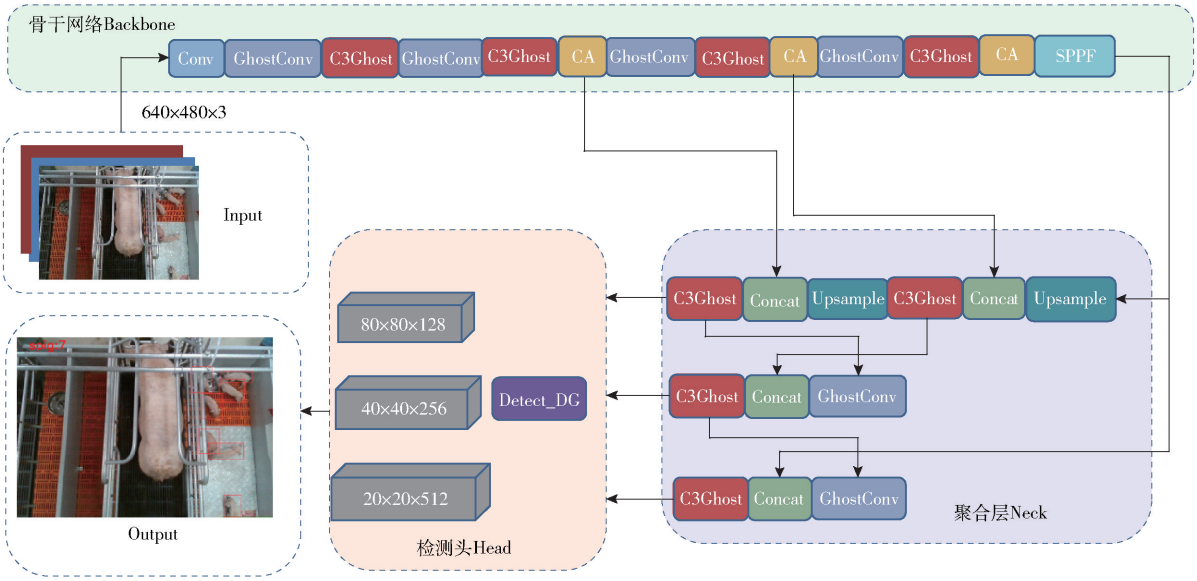

图 3　改进模型结构图

Fig. 3　Improved model structure diagram

的 Head 部分,通过重构 detect 模块,剔除解耦检测头的冗余信息,大量降低了检测头的参数量与计算量,同时提高了模型的检测精度。

2.2 幽灵卷积模块

幽灵卷积网络 GhostNet[9] 旨在通过简单的操作生成更多的特征图,从而显著降低计算成本。该卷积过程主要分为 2 个步骤:①利用 1×1 的卷积将输入的特征图降维生成原始特征图,然后对原始特征图逐一通过线性变换生成部分冗余的特征图,称为 Ghost features。②通过通道维度上的堆叠生成最终的特征图。这里的线性变换,指的是仅进行普通的卷积操作,而不包含非线性变换(如激活函数)以及归一化等复杂操作。在效果良好的卷积神经网络 ResNet-50[10] 中,存在特征冗余是成功的一个重要特征,即存在许多的特征相似对。因此,通过一系列的线性操作就可以生成特征的 Ghost features,不仅可以有效保留网络性能,还能大幅减少计算量。图 4 为GhostNet 的操作示意图。

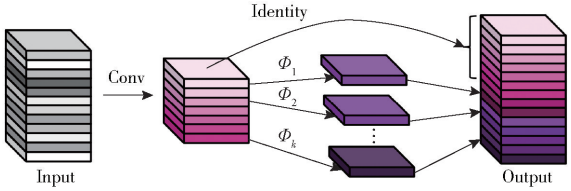

图 4 幽灵卷积模块

Fig. 4 Ghost module

凭借 Ghost 模块的优异特性,研究者开发了一种适用于小型卷积神经网络的 Ghost 瓶颈结构(G-bneck)。其结构类似于 ResNet 中的基本残差块。Ghost 瓶颈主要由 2 个堆叠的 Ghost 模块组成。如图 5 所示,在 2 个模块间添加步长为 1 的深度卷积 DWConv[11],不会改变特征图的空间分辨率,保留足够的细节信息。若改变步长为 2,则进行下采样操作,减少特征图尺寸,从而减少计算量,增大感受野,有助于关注整体的语义信息。使用步长为 1 的瓶颈结构用于搭建 C3Ghost 模块,契合其增强网络特征提取的能力。

2.3 坐标注意力机制 CA

CBAM 注意力机制具备对通道及空间信息的感知力,在经过 3 次卷积之后,只能捕获局部信息,坐标注意力机制(Coordinate attention for efficient mobile network design,CA)通过精确的位置信息对通道关系与长程依赖性进行编码[12],能够捕捉精确位置信息的远程空间交互。其过程主要分为坐标信息嵌入(Coordinate information embeding)和通道注意力加权(Coordinate attention generation),如图 6 所示,图中 H、W 和 C 分别表示图像的高度、宽度和通道数。CA 将通道注意力分解成 2 个 1D 编码的过程,分别沿 2 个空间方向聚合特征。通过这种方式,可以沿一个空间方向捕获长距离依赖关系,同时可以沿另一个空间方向保留精确的位置信息。然后将生成的特征图分别编码为一对方向感知和位置敏感的注意力图,这些图可以互补地应用于输入特征图,以增强感兴趣对象的表示。

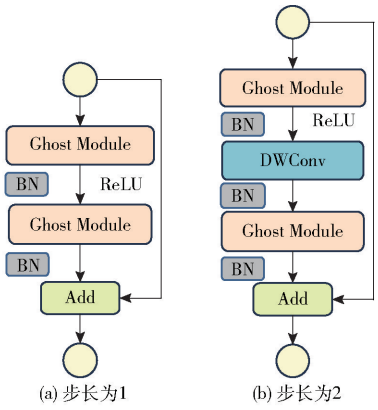

图 5 幽灵瓶颈

Fig. 5 Ghost bottleneck

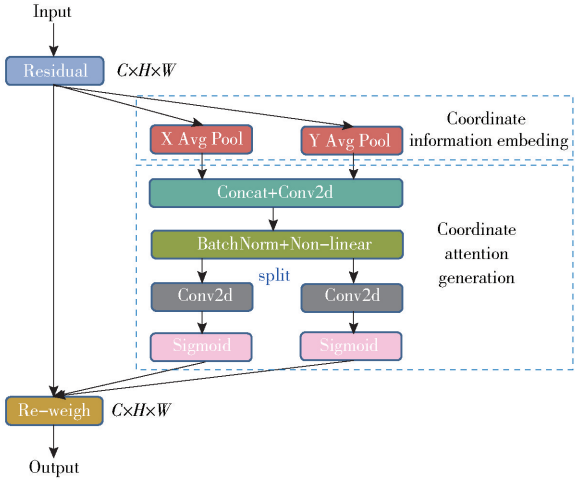

图 6 CA 注意力机制结构

Fig. 6 Structure of coordinate attention

2.4 自定义检测头 Detect_DG

相较以往的版本,YOLO v8[13] 采用解耦头结构(Decoupled-Head),将分类头与检测头进行分离。这一结构的变更虽然大幅增加了计算量,但提升了精度,并加速了模型收敛。检测头中计算量的增加来源于 2 个卷积核(kernel)为 3×3 的卷积操作。通过将第 2 个卷积模块的 kernel 设置为 1,实现通道信息的整合,突出重要特征信息。此外,利用 GhostNet 生成幻影特征图,以挖掘潜在特征并丰富特征多样性。针对本文单类别的任务,对分类支路的首个卷积模块进行优化,采用深度可分离卷积 DSC[14] 技术替代,在保证空间特征提取和融合性能的同时,显著降低了计算复杂度。具体结构如图 7

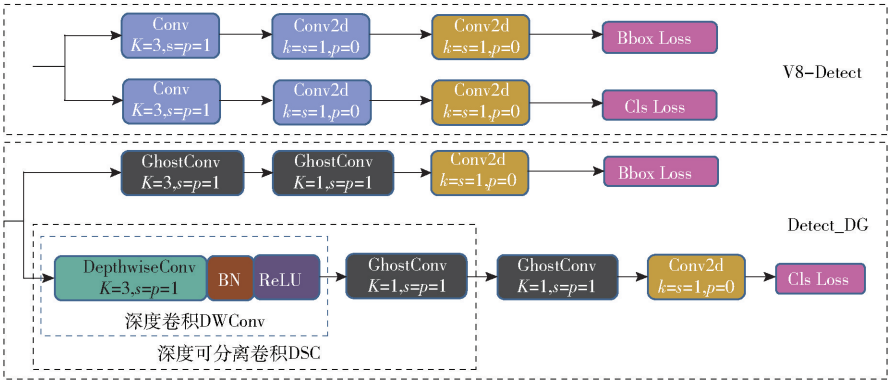

图 7　Detect_DG 与 YOLO v8 检测头结构对比

Fig. 7　Architectural comparison between Detect_DG and YOLO v8 detection head

所示。

2.5　Inner－MPDIoU 损失函数

损失函数是目标检测算法的重要组成部分。目标检测算法 YOLO v8 默认的损失函数是 CIoU，主要是计算预测框与真实框的损失。CIoU 虽然考虑了边界框重叠和中心点距离，但对数据集标注质量要求高。本文设计 Inner－MPDIoU 边界框损失函数，进一步提高边界框回归的准确率。该函数采用辅助(Inner)[15]边界框计算损失，引入 ratio 尺寸因子来控制辅助边界框的大小来检测不同尺寸缺陷。对应公式为

$$\begin{cases} b_l^{gt} = x_c^{gt} - \dfrac{w^{gt} S_{\text{ratio}}}{2} \\ b_r^{gt} = x_c^{gt} + \dfrac{w^{gt} S_{\text{ratio}}}{2} \end{cases} \tag{1}$$

$$\begin{cases} b_t^{gt} = y_c^{gt} - \dfrac{h^{gt} S_{\text{ratio}}}{2} \\ b_b^{gt} = y_c^{gt} + \dfrac{h^{gt} S_{\text{ratio}}}{2} \end{cases} \tag{2}$$

$$\begin{cases} b_l = x_c^{p} - \dfrac{w^{p} S_{\text{ratio}}}{2} \\ b_r = x_c^{p} + \dfrac{w^{p} S_{\text{ratio}}}{2} \end{cases} \tag{3}$$

$$\begin{cases} b_t = y_c^{p} - \dfrac{h^{p} S_{\text{ratio}}}{2} \\ b_b = y_c^{p} + \dfrac{h^{p} S_{\text{ratio}}}{2} \end{cases} \tag{4}$$

$$I_{\text{inter}} = \left(\left(b_r^{gt}, b_r\right)_{\min} - \left(b_l^{gt}, b_l\right)_{\max}\right) \cdot \left(\left(b_b^{gt}, b_b\right)_{\min} - \left(b_t^{gt}, b_t\right)_{\max}\right) \tag{5}$$

$$IoU_{\text{inner}} = \frac{I_{\text{inter}}}{w^{gt} h^{gt} S_{\text{ratio}}^2 + w^{p} h^{p} S_{\text{ratio}}^2 - I_{\text{inter}}} \tag{6}$$

式中　b_l^{gt}、b_r^{gt}、b_t^{gt}、b_b^{gt}——真实框左、右、下、上边界中心点

b_l、b_r、b_t、b_b——锚框左、右、下、上边界中心点

x_c^{gt}、y_c^{gt}——真实框、辅助边界框中心点

x_c^{p}、y_c^{p}——锚框、内锚框中心点

w^{gt}、h^{gt}——真实框高度、宽度

w^{p}、h^{p}——锚框高度、宽度

I_{inter}——真实框、锚框的区域交集

IoU_{inner}——Inner 交并比

MPDIoU[16]通过直接最小化预测框和真实框之间的关键点距离来进行计算，能够综合考虑重叠区域、中心点距离以及宽度和高度偏差。对应公式为

$$\text{MPDIoU} = IoU - \frac{\left(x_1^{p} - x_1^{gt}\right)^2 + \left(y_1^{p} - y_1^{gt}\right)^2}{w^2 + h^2} - \frac{\left(x_2^{p} - x_2^{gt}\right)^2 + \left(y_2^{p} - y_2^{gt}\right)^2}{w^2 + h^2} \tag{7}$$

$$L_{\text{MPDIoU}} = 1 - \text{MPDIoU} \tag{8}$$

式中　(x_1^{gt}, y_1^{gt})、(x_2^{gt}, y_2^{gt})——真实边界框左上和右下坐标

(x_1^{p}, y_1^{p})、(x_2^{p}, y_2^{p})——预测边界框左上和右下坐标

w、h——输入图像宽、高

MPDIoU——多尺度距离交并比

IoU——交并比

L_{MPDIoU}——MPDIoU 损失函数

综上，结合 Inner－IoU 和 MPDIoU，提出 Inner－MPDIoU 损失函数

$$L_{\text{Inner-MPDIoU}} = L_{\text{MPDIoU}} + IoU - IoU_{\text{inner}} \tag{9}$$

式中　$L_{\text{Inner-MPDIoU}}$——Inner－MPDIoU 损失函数

使用该损失函数可以更好地计算边界框回归损失，提高模型精度和收敛速度。

2.6　Soft－NMS

YOLO v8n 目标检测算法采用非极大值抑制算法(Non－maximum suppre，NMS)[17]对目标边界框进行后处理来获得最终的边界框。该算法通过将置信度最高的框(M)与其他框进行 IoU 操作，滤除掉超过与 M 重叠度阈值的候选框。养殖场环境下仔

猪目标存在栏杆遮挡和尺寸小等特点,这一操作容易导致仔猪漏检。柔性非极大值抑制算法 Soft - NMS[18]为 NMS 的改进算法,不会直接舍弃重叠的低置信度候选框,而是根据高斯函数利用这些 IoU 大于阈值边界框的置信度计算出新的置信度,计算公式为

$$S_i = \begin{cases} S_i & (IoU(M,b_i) < N_t) \\ S_i(1 - IoU(M,b_i)) & (IoU(M,b_i) \geqslant N_t) \end{cases} \tag{10}$$

式中 S_i——边界框置信度

b_i——候选框

$IoU(M,b_i)$——置信度最高的边界框与当前边界框的交并比计算值

N_t——阈值边界框阈值

2.7 试验环境与参数设置

本试验使用的操作系统是 Linux(Ubuntu 20.04),CPU 为 12 vCPU Intel(R) Xeon(R) Platinum 8255C CPU @ 2.50 GHz,GPU 为 RTX 3080Ti(11GB)。使用 Python 作为编程语言,神经网络框架选择 Pytorch 2.0,CUDA 版本为 11.8。训练时的迭代次数为 300,批次为 16,初始学习率设为 0.01,动量设为 0.937,优化器选择 Adam。

2.8 评价指标

采用 F1 分数(F1-score)、平均精度(Average precision)、模型参数量(Params)、模型内存占用量(Model memory)、帧率(Frams per second)、浮点计算量(Floating point operations)来作为模型性能评价指标。

3 试验结果分析

3.1 消融试验

为了验证改进模型 GSD - YOLO 的优越性,以 YOLO v8n 为基准模型,评估所有改进方法对基准模型性能的提升。所有试验皆用自制的仔猪数据集进行训练验证。消融试验结果如表 1 所示。表中“√”表示使用该改进方法。

表 1 消融试验结果

Tab.1 Ablation test results

模型序号	GhostNet	CA	Inner - MPDIoU	Detect_DG	参数量	浮点计算量	模型内存占用量/MB	F1 分数/%	平均精度/%	帧率/(f·s^{-1})
1					3.006×10^6	8.1×10^9	6.2	95.8	97.9	32
2	√				1.714×10^6	5.1×10^9	3.8	96.0	98.0	36
3		√			3.017×10^6	8.2×10^9	6.2	96.1	98.1	34
4			√		3.011×10^6	8.2×10^9	6.2	96.1	98.1	36
5				√	2.443×10^6	5.9×10^9	5.1	96.0	97.9	33
6	√	√			1.730×10^6	5.2×10^9	3.8	96.4	98.3	38
7	√		√		1.719×10^6	5.1×10^9	3.8	96.1	98.2	33
8		√	√		3.028×10^6	5.2×10^9	3.8	96.2	98.2	38
9		√		√	2.453×10^6	6.0×10^9	5.1	96.3	98.0	34
10	√			√	1.142×10^6	2.6×10^9	2.6	96.2	98.2	35
11			√	√	2.448×10^6	6.0×10^9	5.1	96.3	98.3	35
12	√	√	√		1.726×10^6	5.2×10^9	3.7	96.7	98.4	34
13		√	√	√	2.452×10^6	6.0×10^9	5.2	96.4	98.3	34
14	√	√	√	√	1.158×10^6	2.7×10^9	2.6	96.8	98.5	36

表 1 结果显示,模型 1 是 YOLO v8n 原始算法,以该模型的性能指标作为评价改进效果的基准。

由表 1 可得,为了实现模型轻量化,在模型 2 引入 GhostNet,参数量和浮点计算量分别降低 42.80%、37.04%,模型内存占用量降至 3.8 MB。F1 分数和 AP 分别提高 0.2、0.1 个百分点,帧率提升 4 f/s。在模型 3 和模型 4 中分别嵌入 CA 注意力模块和改进替换了 Inner - MPDIoU 损失函数,增强了网络末端对关键部位的关注能力,优化了边界框的回归机制。虽然二者对模型内存占用量及计算量等影响较小,但 F1 分数和平均精度分别提升 0.3、0.2 个百分点,帧率分别提升 2、4 f/s。模型 5 采用自定义的 Detect_DG 检测头,虽然 F1 分数、平均精度以及帧率收效不显著,但参数量和浮点计算量分别降低 18.73%、27.17%。模型 6 到模型 11 的结果表明,不同改进方法的组合使用提升效果比之单一优化方法的效果更佳。模型 12、13 则结合了所用 3 个改进方法的优点,不仅实现了轻量化,还有效提升了检测的速度与精度。在模型 12 的基础上,模型 14 引入自定义的轻量化检测头 Detecr_DG。相比于基准模型,其参数量、浮点计算量和模型内存占用量分别降低 61.28%、66.66% 和 58.07%,实现大

幅轻量化。F1 分数、平均精度分别提升 1.0、0.6 个百分点，帧率提升 12.5%，显著提升了模型的检测精度与速度。由此，选择模型 14 作为本文模型 GSD－YOLO。

3.2　不同注意力机制对比试验

为了验证本文嵌入的 CA 注意力模块的性能，在基准模型的相同位置中分别嵌入 CBAM[19]、ECA[20]和 SimAm[21]模块，并与本文嵌入模块进行对比，试验结果如表 2 所示。

表 2　不同注意力机制模型性能对比

Tab. 2　Performance comparison of different attention mechanism models　%

注意力机制	F1 分数	平均精度
CBAM	95.8	98.0
ECA	95.8	98.0
SimAm	95.9	97.9
CA	96.1	98.1

由表 2 可知，对比其他 3 种注意力机制，本文嵌入的 CA 注意力机制，F1 分数和平均精度最高。相比之下，凭借其解决长依赖问题的能力，更有效地增强了关键区域的表达能力。试验结果表明，CA 注意力机制对本文任务是最适合的。

3.3　不同损失函数对比试验

为了验证 Inner－MPDIoU 损失函数对本文模型性能的提升能力，分析了不同损失函数对模型性能的影响。基准模型 YOLO v8n 的默认损失函数为 CIoU[22]，分别引入 GIoU[23]、MPDIoU、Inner－GIoU、Inner－MPDIoU 这 4 个损失函数进行对比试验，结果如表 3 所示。

表 3　不同损失函数对模型性能影响

Tab. 3　Impact of different loss functions on model performance　%

损失函数	F1 分数	平均精度
GIoU	95.8	98.0
MPDIoU	96.0	97.9
Inner－GIoU	95.7	98.0
Inner－MPDIoU	96.1	98.1

结合表 3 和图 8 的试验结果，相较于 CIoU 损失函数，所提出的改进损失函数在训练过程中对边界框位置和分类参数的回归精度有显著进步。尽管图 8 中Inner－GIoU 与 Inner－MPDIoU 2 种损失函数在边界框、分类回归的精度上并无明显差异，然而在 F1 分数和平均精度等关键性能指标上，Inner－MPDIoU 损失函数表现更为出色。因此，本文选择 Inner－MPDIoU 损失函数替代原有的 CIoU 损失函数，以期获得更优的检测性能。

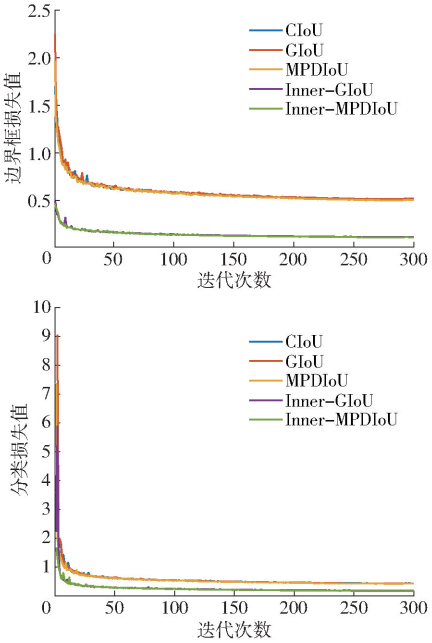

图 8　不同损失函数损失值

Fig. 8　Loss for different loss functions

3.4　热力图可视化分析

为了验证上述改进方法对仔猪检测的影响，对基准模型和本文模型进行了热力图可视化，对比了模型间对重叠、遮挡区域的敏感程度。结果如图 9 所示，在使用本文改进方法后，模型对仔猪目标的关注更多，感知区域更准确。



图 9　热力图可视化对比

Fig. 9　Heatmap visualization comparison

3.5　不同算法试验

为对 GSD－YOLO 模型进行性能测试，选择 Faster R－CNN、CenterNet、YOLO v5s[24]、YOLO v6s[25]、YOLO v7－tiny、YOLO v8n、YOLO v11n 等 7 种主流模型进行比较，模型对比结果如表 4 所示。

由表 4 可以看出，GSD－YOLO 模型计算量、模型内存占用量显著优于 Faster R－CNN、SDD、YOLO v5s、YOLO v6s、YOLO v7－tiny、YOLO v8n 和 YOLO v11n 等主流模型。此外，该模型 F1 分数、平均精度以及帧率 3 项关键性能指标均达到最优水平。与这些模型相比，GSD－YOLO F1 分数分别提升 4.8、

表 4 不同算法对比结果

Tab. 4 Algoritmic performance benchmarking results

算法	参数量	浮点计算量	模型内存占用量/MB	F1 分数/%	平均精度/%	帧率/(f·s^{-1})
Faster R-CNN	2.150×10^{8}	2.187×10^{11}	137.6	92.0	92.7	4.6
SSD	2.400×10^{7}	1.014×10^{11}	97.7	92.8	93.6	15
YOLO v5s	7.010×10^{6}	1.580×10^{10}	14.3	95.4	97.6	33
YOLO v6s	1.850×10^{7}	4.520×10^{10}	36.76	94.5	97.2	32
YOLO v7-tiny	6.010×10^{6}	1.300×10^{10}	12.2	95.5	97.8	33
YOLO v8n	3.006×10^{6}	8.200×10^{9}	6.2	95.8	97.9	32
YOLO v11n	2.405×10^{6}	5.800×10^{9}	4.8	95.0	97.8	32
GSD-YOLO	1.158×10^{6}	2.700×10^{9}	2.6	96.8	98.5	36

4.0、1.4、2.3、1.3、1.0、1.8 个百分点，AP 则分别提升 5.8、4.9、0.9、1.3、0.7、0.6、0.7 个百分点。尽管本文模型在 YOLO 系列上帧率优势并不突出，但仍实现了帧率的稳步提升，分别提升 9.09%、12.5%、9.09%、12.5%、12.5%。虽然 YOLO v11n 作为当前最先进的版本，但其设计可能侧重于降低计算负担，在公共数据集上的效果表现比以往版本更加出色。针对自制的仔猪数据集，GSD-YOLO 模型检测准确性、检测速度均优于其他 7 种模型。

4 检测结果分析

4.1 目标检测与计数分析

为进一步给后续研究提供参考，对 GSD-YOLO 模型和基准模型 YOLO v8n 的检测结果进行对比分析，并添加目标计数来验证准确性。检测图像从测试集中筛选，结果如图 10 所示，图像左上角为仔猪目标的计数结果：蓝圈表示误检，黄圈表示漏检，红圈表示检测框重叠。

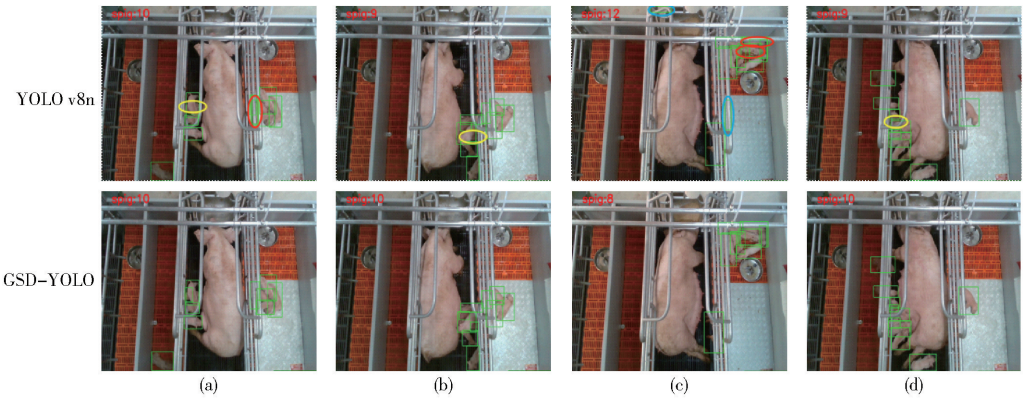

图 10 改进前后检测效果对比

Fig. 10 Comparison of detection effect before and after improvement

从图 10 可知，GSD-YOLO 在解决遮挡、猪群聚集问题上显示出比 YOLO v8n 更优越的性能。在遮挡问题上，图 10a、10b、10d 均出现了漏检问题，图 10b、10d 的仔猪遮挡比例达到 80%。同时图 10a、10c 出现的检测框重叠问题在本文提出的模型得到了有效解决。此外，图 10c 中将栏杆误检为仔猪的现象并没有在本文模型的预测结果中发生。综上可得，本文提出的模型在遮挡和猪群聚集条件下检测仔猪的效果更为出色。

4.2 泛化性试验

为了展示改进模型 GSD-YOLO 在仔猪目标检测领域的显著优势，将其与 YOLO v8n 模型在全新数据上进行了泛化性测试。鉴于养殖场内部环境光线条件复杂多变，加之卫生状况的挑战，图像质量常受昏暗光线或摄像头污染的影响。为模拟实际养殖环境，采用图像增强技术，降低图像亮度以模拟光线昏暗条件。此外，通过应用高斯模糊和二值化处理，模拟了灰尘和油污等常见污染，以进一步考验模型的鲁棒性。最终生成了 500 幅包含昏暗光照以及油污灰尘遮挡的图像，专门用于评估 2 个模型在复杂数据集上的泛化能力。以上数据仅用于测试泛化性，未参与模型训练。表 5 给出了二者在复杂数据集的性能指标。图 11 展示了泛化性测试的对比结果：黄圈表示漏检，绿圈表示检测框重叠，α 为亮度影响因子。

表 5 复杂数据集的对比试验结果

Tab. 5 Comparison experiment on a complex dataset

模型	F1 分数/%	平均精度/%	帧率/(f·s^{-1})
YOLO v8n	87.7	90.3	40
GSD-YOLO	88.3	91.1	39

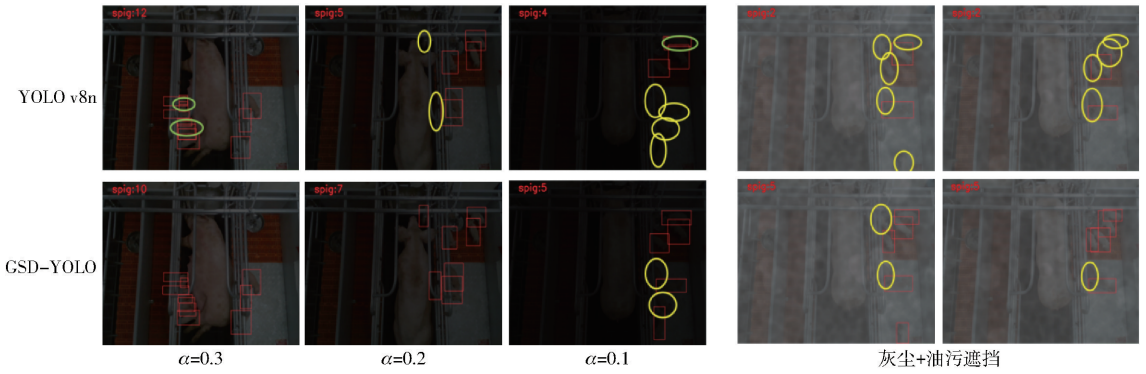

图 11　泛化性检测结果对比

Fig. 11　Comparison of generalization results employs

由表 5 可知，GSD－YOLO 模型在复杂数据集上的性能优于 YOLO v8n。图 11 结果表明，在亮度调整因子 α 设定于 0.2 及 0.3 的条件下，GSD－YOLO 模型几乎没有出现漏检和检测框重叠的情况。本研究使用的训练数据集，其亮度调整因子 α 设定是 0.6。即便在 α 降至 0.1 的极端条件下，相较于 YOLO v8n 模型，本文模型依然展现出较低的漏检率。尤其在灰尘和油污等复杂情况下，GSD－YOLO 模型表现出的检测效果优于 YOLO v8n 模型。宋怀波等[26]通过调整 γ 变换参数来控制图像亮度，并观察到当 γ 改变至 2.5 时（模拟养殖场在20:00 至 21:00 时段的光照条件），肉牛检测过程中出现了漏检和误检的问题。由于监控视角较大，图像边缘的肉牛难以标注，故而在亮度变化较大时模型难以提取其特征，突出了数据质量对检测性能的重要性。SCAILLIEREZ 等[27]则通过收集 07:00 至 18:00 不同时间段的 10 种室内光照数据进行训练，成功将猪只目标的正确检测率提升至 97%。同时试验中发现摄像头污染导致系统将不明遮挡物误检为猪只。在未来研究应提高模型鲁棒性，或通过图像预处理减轻污染物对检测结果的影响。与现有研究相比，本文在试验设计上进行了更为全面的考虑，不仅涵盖了光照条件的泛化试验，还特别关注了摄像头污染对检测结果的影响。本文的研究目标更加细化，且所提出的模型在应对不同亮度条件及污染物干扰方面展现出了较强的鲁棒性。

5　不同设备部署试验

5.1　Jetson Orin NX 部署试验结果

为验证本研究所提模型的可行性，将 GSD－YOLO 模型部署在 Jetson Orin NX 边缘计算平台，并选择 TensorRT 优化框架进行推理加速，比较其部署前后的性能，以评估其在实际复杂场景下的有效性。Jetson Orin NX 平台部署运行情况如图 12 所示。

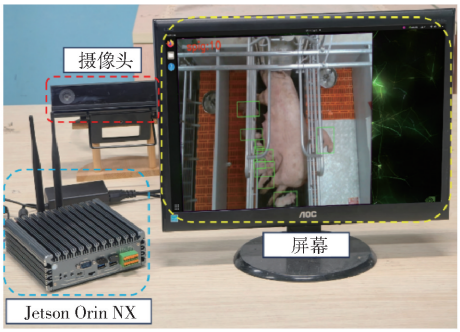

图 12　Jetson Orin NX 平台模型部署情况

Fig. 12　Model deployment on Jetson Orin NX platform

模型部署试验结果如表 6 所示。YOLO v8n 模型在主机平台和 Jetson Orin NX 平台上的检测帧率分别为 32、3.4 f/s。GSD－YOLO 模型在主机平台和 Jetson Orin NX 平台上实现的检测帧率分别为 36、5.8 f/s，相较于原始 YOLO v8 模型，改进 GSD－YOLO 模型部署前后分别提升 4、2.4 f/s。加速前，由于边缘设备计算资源有限，模型检测速度较低，经 TensorRT 加速后，改进模型检测帧率达到 16.8 f/s，提升约 2.9 倍。上述结果表明，GSD－YOLO 模型在部署推理速度方面优于基线模型，具有良好的边缘设备部署前景，为模型实现仔猪端到端的检测提供技术支撑。

表 6　模型部署检测帧率对比

Tab. 6　Comparison of detection frame rates before and after model deployment　　f/s

模型	主机平台	Jetson Orin NX 平台	
		无加速	TensorRT 加速
YOLO v8n	32	3.4	9.1
GSD－YOLO	36	5.8	16.8

5.2　安卓移动端部署

本文所使用的软件开发环境为 Android Studio Chipmunk 2021.2.1，安卓系统为 iQOO OS 12.1。手机屏幕尺寸为 16.865 6 cm，分辨率是 2 400 像素 × 1 080 像素。将 GSD－YOLO 模型最终导出成 ncnn

格式并部署到该智能手机上。仔猪检测软件功能包括:①从相册中调取图像。②实现检测模型加载。③检测结果的可视化。④实现调用摄像头模块加载。⑤模型以及 CPU、GPU 的切换。

部署结果如图 13 所示,该软件可通过“选图”从相册调取图像,“识别”实现目标的检测并更新目标个数,对每个猪舍内的仔猪数量进行盘点。通过“视频识别”可进行实时检测,在相机的右上角显示帧率。本文模型在安卓移动端上帧率达33 f/s 左右,基本满足实时检测的性能需求。

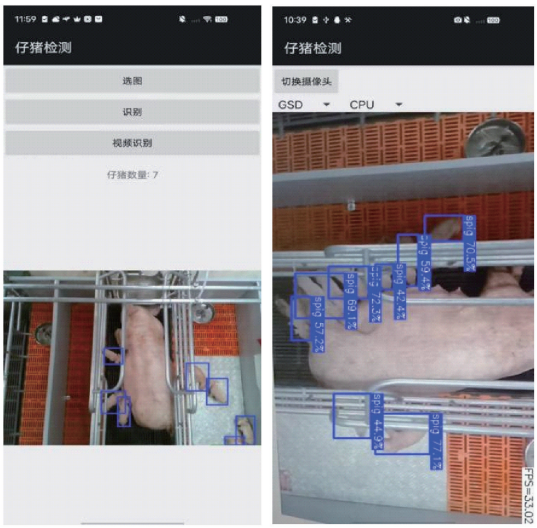

图 13 安卓部署效果

Fig. 13 Android deployment effects

6 结论

(1)针对养殖场复杂环境下轻量化仔猪目标检测模型的精度要求,在 YOLO v8n 的基础架构上进行了改进。引入 GhostNet 模块、重构检测头 Detect_DG 实现了模型的轻量化,模型参数和浮点计算量分别下降 61.28%、66.66%;在骨干 Backbone 网络末端嵌入 CA 注意力模块,整合 Inner - MPDIoU 损失函数与 Soft - NMS 后处理算法,F1 分数、平均精度分别提升 1.0、0.6 个百分点。最终开发了一种体积小、计算高效和检测精度高的快速检测模型。

(2)为验证 GSD - YOLO 模型的性能,选择 7 种网络进行了消融试验。试验结果表明,与基准模型 YOLO v8n 相比,GSD - YOLO 在模型内存占用量、计算量、检测精度与速度上均展示出显著优势。

(3)在相同的试验环境中,GSD - YOLO 在复杂环境下对仔猪目标的检测效果优异。特别是在昏暗、灰尘和油污遮挡等恶劣条件下,本模型依然能保持出色的检测效果,充分证明其优越性。

(4)将 GSD - YOLO 模型部署在 Jetson Orin NX 平台上,选择 TensorRT 优化框架进行推理加速。此外设计了一款基于 Android 的仔猪检测软件,操作简便,具备相册检测、实时检测和统计仔猪数量的功能,对用户的设备要求低,有较高的实用性,能满足仔猪目标的实时检测的需求。

参 考 文 献

[1] KAMILARIS A, PRENAFETA-BOLDÚ F X. Deep learning in agriculture: a survey[J]. Computers and Electronics in Agriculture, 2018, 147(1): 70 - 90.

[2] ZHENG Z, WANG P, REN D, et al. Enhancing geometric factors in model learning and inference for object detection and instance segmentation[J]. IEEE Transactions on Cybernetics, 2021, 52(8): 8574 - 8586.

[3] HUANG E, MAO A, GAN H, et al. Center clustering network improves piglet counting under occlusion[J]. Computers and Electronics in Agriculture, 2021, 189: 106417.

[4] 高云,郭继亮,黎煊,等. 基于深度学习的群猪图像实例分割方法[J]. 农业机械学报,2019,50(4):179 - 187.
GAO Yun, GUO Jiliang, LI Xuan, et al. Instance-level segmentation method for group pig images based on deep learning[J]. Transactions of the Chinese Society for Agricultural Machinery, 2019, 50(4): 179 - 187. (in Chinese)

[5] 沈明霞,太猛,CEDRIC Okinda,等. 基于深层卷积神经网络的初生仔猪目标实时检测方法[J]. 农业机械学报,2019,50(8):270 - 279
SHEN Mingxia, TAI Meng, CEDRIC Okinda, et al. Real-time detection method of newborn piglets based on deep convolution neural network[J]. Transactions of the Chinese Society for Agricultural Machinery, 2019, 50(8): 270 - 279. (in Chinese)

[6] 杨秋妹,陈淼彬,黄一桂,等. 基于改进 YOLO v5n 的猪只盘点算法[J]. 农业机械学报,2023,54(1):251 - 262.
YANG Qiumei, CHEN Miaobin, HUANG Yigui, et al. Pig counting algorithm based on improved YOLO v5n[J]. Transactions of the Chinese Society for Agricultural Machinery, 2023, 54(1): 251 - 262. (in Chinese)

[7] 丁奇安,刘龙申,陈佳,等. 基于 Jetson Nano + YOLO v5 的哺乳期仔猪目标检测[J]. 农业机械学报,2022,53(3):277 - 284.
DING Qi'an, LIU Longshen, CHEN Jia, et al. Object detection of suckling piglets based on Jetson Nano and YOLO v5[J]. Transactions of the Chinese Society for Agricultural Machinery, 2022, 53(3): 277 - 284. (in Chinese)

[8] AHN H, SON S, KIM H, et al. EnsemblePigDet: Ensemble deep learning for accurate pig detection[J]. Applied Sciences, 2021, 11(12): 5577.

[9] 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.

[10] SIFRE L, MALLAT S. Rigid-motion scattering for texture classification[J]. arXiv Preprint, arXiv:1403.1687, 2014.
[11] HOU Q, ZHOU D, FENG J. Coordinate attention for efficient mobile network design[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021: 13713－13722.
[12] TERVEN J, CÓRDOVA-ESPARZA D M, ROMERO-GONZÁLEZ J A. A comprehensive review of yolo architectures in computer vision: from yolov1 to YOLOv8 and YOLO-nas[J]. Machine Learning and Knowledge Extraction, 2023, 5(4): 1680－1716.
[13] GOYAL P. Accurate, large minibatch SG D: training imagenet in 1 hour[J]. arXiv Preprint, arXiv:1706.02677, 2017.
[14] ZHANG H, XU C, ZHANG S. Inner-IoU: more effective intersection over union loss with auxiliary bounding box[J]. arXiv Preprint, arXiv:2311.02877, 2023.
[15] MA S, XU Y. Mpdiou: a loss for efficient and accurate bounding box regression[J]. arXiv Preprint, arXiv:2307.07662, 2023.
[16] BODLA N, SINGH B, CHELLAPPA R, et al. Soft-NMS—improving object detection with one line of code[C]//Proceedings of the IEEE International Conference on Computer Vision, 2017: 5561－5569.
[17] FELZENSZWALB P F, GIRSHICk R B, MCALLESTER D, et al. Object detection with discriminatively trained part-based models[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2009, 32(9): 1627－1645.
[18] WOO S, PARK J, LEE J Y, et al. Cbam: convolutional block attention module[C]//Proceedings of the European Conference on Computer Vision (ECCV), 2018: 3－19.
[19] WANG Q, WU B, ZHU P, et al. ECA－Net: Efficient channel attention for deep convolutional neural networks[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020: 11534－11542.
[20] YANG L, ZHANG R Y, LI L, et al. Simam: A simple, parameter-free attention module for convolutional neural networks[C]//International Conference on Machine Learning. PMLR, 2021: 11863－11874.
[21] 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/CVF Conference on Computer Vision and Pattern Recognition,2019: 658－666.
[22] ZHOU X, WANG D, KRÄHENBÜHL P. Objects as points[J]. arXiv Preprint, arXiv:1904.07850, 2019.
[23] JIANG P, ERGU D, LIU F, et al. A review of Yolo algorithm developments[J]. Procedia Computer Science, 2022, 199: 1066－1073.
[24] LI C, LI L, JIANG H, et al. YOLOv6: a single-stage object detection framework for industrial applications[J]. arXiv Preprint, arXiv:2209.02976, 2022.
[25] HUSSAIN M. YOLO－v1 to YOLO－v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection[J]. Machines, 2023, 11(7): 677.
[26] 宋怀波,李嵘,王云飞,等. 基于 ECA－YOLO v5s 网络的重度遮挡肉牛目标识别方法[J]. 农业机械学报,2023,54(3):274－281.
SONG Huaibo,LI Rong,WANG Yunfei,et al. Recognition method of heavily occluded beef cattle targets based on ECA－YOLO v5s[J]. Transactions of the Chinese Society for Agricultural Machinery,2023,54(3):274－281. (in Chinese)
[27] SCAILLIEREZ A J, IZQUIERDO G T, BROERS H, et al. Determining posture and location of pigs using an object detection model under different lighting conditions[J]. Translational Animal Science, 2024,8:167.