计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(11):48−61 [doi: 10.15888/j.cnki.csa.009306]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于改进 YOLOv5 和 Bytetrack 的牦牛跟踪①

王建文[1], 张玉安[1], 朱海鹏[1], 宋仁德[2]

[1](青海大学 计算机技术与应用系, 西宁 810016)
[2](玉树州动物疫病预防控制中心, 玉树 815099)
通信作者: 张玉安, E-mail: 2011990029@qhu.edu.cn

摘　要: 目前, 我国青藏高原地区的牦牛养殖方式以传统的人工放牧为主. 为解决人力养殖方式无法快速跟踪统计牦牛数量的问题, 本文提出了一种改进 YOLOv5 和 Bytetrack 的牦牛跟踪方法, 以实现在视频输入情况下快速检测跟踪牦牛. 采用基于深度学习的 YOLOv5 目标检测网络, 结合 CA 注意力、跨尺度特征融合和空洞卷积池化金字塔等优化方法, 减少牦牛检测中因遮挡而导致检测难度大、误检漏检的问题, 实现对视频中牦牛更精确的检测; 使用 Bytetrack 跟踪器通过卡尔曼滤波和匈牙利算法实现帧间目标关联, 并为目标匹配 ID; 使用 ImageNet 中的部分牦牛数据和青海玉树地区采集的牦牛样本图像来训练模型. 实验结果表明: 本文改进模型的平均检测精确度为 98.7%, 比原 YOLOv5s、SSD、YOLOX 和 Faster RCNN 模型分别提高 1.1、1.89、8.33、0.4 个百分点, 能快速收敛, 检测性能最优; 改进的 YOLOv5s 和 Bytetrack 跟踪结果最优, MOTA 提高了 7.164 6%. 本研究改进的模型能够更加快速准确地检测和跟踪统计牦牛, 为青海地区畜牧业的智慧化发展提供技术支持.

关键词: 牦牛; 目标检测; 注意力机制; Swin Transformer; 多目标跟踪; Bytetrack

引用格式: 王建文,张玉安,朱海鹏,宋仁德.基于改进 YOLOv5 和 Bytetrack 的牦牛跟踪.计算机系统应用,2023,32(11):48–61. http://www.c-s-a.org.cn/1003-3254/9306.html

Yak Tracking Based on Improved YOLOv5 and Bytetrack

WANG Jian-Wen[1], ZHANG Yu-An[1], ZHU Hai-Peng[1], SONG Ren-De[2]

[1](Department of Computer Technology and Applications, Qinghai University, Xining 810016, China)
[2](Yushu Prefecture Animal Disease Prevention and Control Center, Yushu 815099, China)

Abstract: At present, the yak breeding method in the Qinghai-Tibet Plateau region of China is mainly based on traditional manual grazing. To solve the problem that human breeding methods cannot quickly track and count the number of yaks, an improved YOLOv5 and Bytetrack yak tracking method is proposed in this study to achieve the fast detection and tracking of yaks under video input. The YOLOv5 object detection network based on deep learning, combined with optimization methods such as coordinate attention, cross-scale feature fusion, and atrous spatial pyramid pooling pyramid, is adopted to reduce the difficulty of detection and misdetection caused by occlusion in yak detection, so as to accurately detect yak targets in videos. The Bytetrack tracker is used to implement the inter-frame object association through Kalman filtering and Hungarian algorithm, and the IDs are matched to the targets. The model is trained by using part of the yak data in ImageNet Dataset and yak sample images collected from the Yushu region of Qinghai. The experimental results show that the average detection accuracy of the improved model proposed in this study is 98.7%, which is 1.1, 1.89, 8.33, and 0.4 percentage points higher than the original YOLOv5s, SSD, YOLOX, and Faster RCNN models, respectively. It can converge quickly and has the best detection performance. The improved YOLOv5s and Bytetrack tracking results are

① 基金项目: 青海省科技计划 (2020-QY-218); 国家现代农业产业技术体系 (CARS-37); 青海省“昆仑英才•高端创新创业人才”
收稿时间: 2023-04-28; 修改时间: 2023-05-29; 采用时间: 2023-06-28; csa 在线出版时间: 2023-09-19
CNKI 网络首发时间: 2023-10-07

the best, with MOTA increased by 7.164 6%. The improved model developed in this study can detect and track yaks more quickly and accurately, providing technical support for the intelligent development of animal husbandry in the Qinghai region.

Key words: yak; object detection; attention mechanism; Swin Transformer; multi-object tracking; Bytetrack

牦牛作为我国高寒地区的特色畜牧业品种之一, 主要分布在青藏高原地区. 其产业规模呈现上升态势, 中国的牦牛产量约占世界产量的 95% 左右[1]. 牦牛养殖业逐步成为青海省的支柱性产业和区域特色优势产业, 以牦牛和藏羊养殖为主的现代畜牧养殖业已经成为当前乡村振兴战略的重要驱动力. 高效发展牦牛养殖业对畜牧经济的可持续性发展和增加农牧民群众经济收入有着重要作用. 以青海玉树地区为例, 大多数牧民的牦牛养殖规模达到上百头, 牧场区域面积大, 传统的放牧方式难以快速准确地跟踪计数牦牛数量, 人力投入多且管理效率较低. 而计算机图像处理技术与畜牧业的结合, 可实现快速检测和跟踪统计牦牛数量, 提高畜牧业生产效率的同时, 推动畜牧业向着更加科技化的方向发展.

目前, 国内外学者们已经将深度卷积神经网络算法应用于动物检测和跟踪中. 文献 [2] 提出一种基于视频数据的牦牛统计方法, 使用分辨率高的牦牛视频, 人工设计牦牛检测的外观特征信息, 结果表明模型的泛化能力不是很好. 文献 [3] 利用 YOLOv3 检测猫、老鼠和鸟类等动物, 检测平均精度为 75.2%. 文献 [4] 利用生成对抗网络模型, 检测野生动物的夜间红外图像, 使检测精确度得到提升. 文献 [5] 将 RFID 技术应用到动物检测和跟踪管理中, 它把 RFID 标签固定在动物身上, 这种方法虽然提升了统计精度, 但损害动物福利, 且标签容易受到外界干扰而脱落, 无法大范围展开应用. 文献 [6] 提出一种基于参数迁移策略的再训练源模型的方法, 用神经网络检测识别水产动物, 检测精度为 97.4%. 文献 [7] 通过改进 YOLOv3 进行猪脸检测识别, 模型检测精度有一定的提升, 但是仍存在小目标检测边界定位不准的问题. 文献 [8] 提出 Siamese-FC 算法, 将全卷积网络嵌入到跟踪算法中, 提升了跟踪效果和检测速度. 文献 [9] 提出 Siamese-RPN 算法, 通过结合 Siamese 跟踪算法和 RPN 网络, 将多尺度测试跟踪任务转变为 one-shot 检测任务. 文献 [10] 提出基于 YOLOv4_tiny 的网络模型, 通过结合迁移学习和权重加权使模型能在数据集较少时提高检测精度, 但平均精度为 61.18%. 文献 [11] 提出一种基于 SSD 的网络模型, 利用 DenseNet-169 网络提取特征, 然后联合训练中心损失函数和归一化指数来加快模型的收敛速度, 但降低了模型的检测准确率. 上述算法的应用可以提高检测准确率, 并且在不同的应用场景下都取得了较好的性能表现. 然而, 这些方法仍然存在一些问题, 如模型泛化能力不足、计算量大、检测速度慢和误检漏检率高等缺点. 因此, 本文通过改进 YOLOv5 和 Bytetrack 算法, 实现快速检测和跟踪统计牦牛, 在兼顾推理速度和跟踪准确度的同时, 提高模型的泛化能力, 帮助牧民更加高效的监测牦牛, 为牦牛养殖业的可持续发展和乡村振兴提供有力支持.

1 实验数据

1.1 目标检测数据集

本研究将牦牛目标检测定义为二分类问题, 简化了原 YOLOv5 网络对 80 类物体进行分类检测的问题. 数据集来源于 ImageNet 数据集[12] 的部分图像和在青海省玉树藏族自治州使用 GoPro8 拍摄的牦牛视频数据. 处理视频数据时, 保留 80% 的牦牛躯干出现在视野中的视频段. 利用 FFmpeg 工具将视频分割为图片, 筛选去除帧间相似度过高的图片, 得到牦牛样本图像 3 164 张, 使用 Labelimg 工具标注得到 3 164 个 XML 文件.

为了提升模型泛化能力, 使用随机旋转、裁剪、平移、镜像、增加噪点和调整亮度的数据增广技术, 扩充牦牛检测数据, 增强后得到 19 704 张图片. 其中, 对图像进行随机旋转可以扩大数据集的规模, 以获得理想的训练效果; 改变图像的色调和亮度可以模拟光照情况变化对图像的干扰, 在一定程度上消除光环境的影响[13]. 然而, 增强后的样本数据中背景重复率高, 如果全部用于学习, 会降低训练速度, 且容易导致模型过拟合. 故本研究设计两种实验方案. 实验 1: 从数据集中抽取 7 020 张图片做消融实验, 用于测试各模块对模型性能的影响. 按照比例 7:1.5:1.5 随机划分为训练

集、验证集和测试集. 其中训练集 4 900 张, 验证集和测试集各 1 060 张. 实验 2: 抽取全部数据的 80% 用于训练以评估模型的整体性能. 数据集划分和实验 1 相同, 其中训练集 11 034 张, 验证集和测试集各 2 364 张. 每个图像样本按顺序编号, 训练集和验证集的样本编号互斥. 部分数据集图片如图 1 所示.

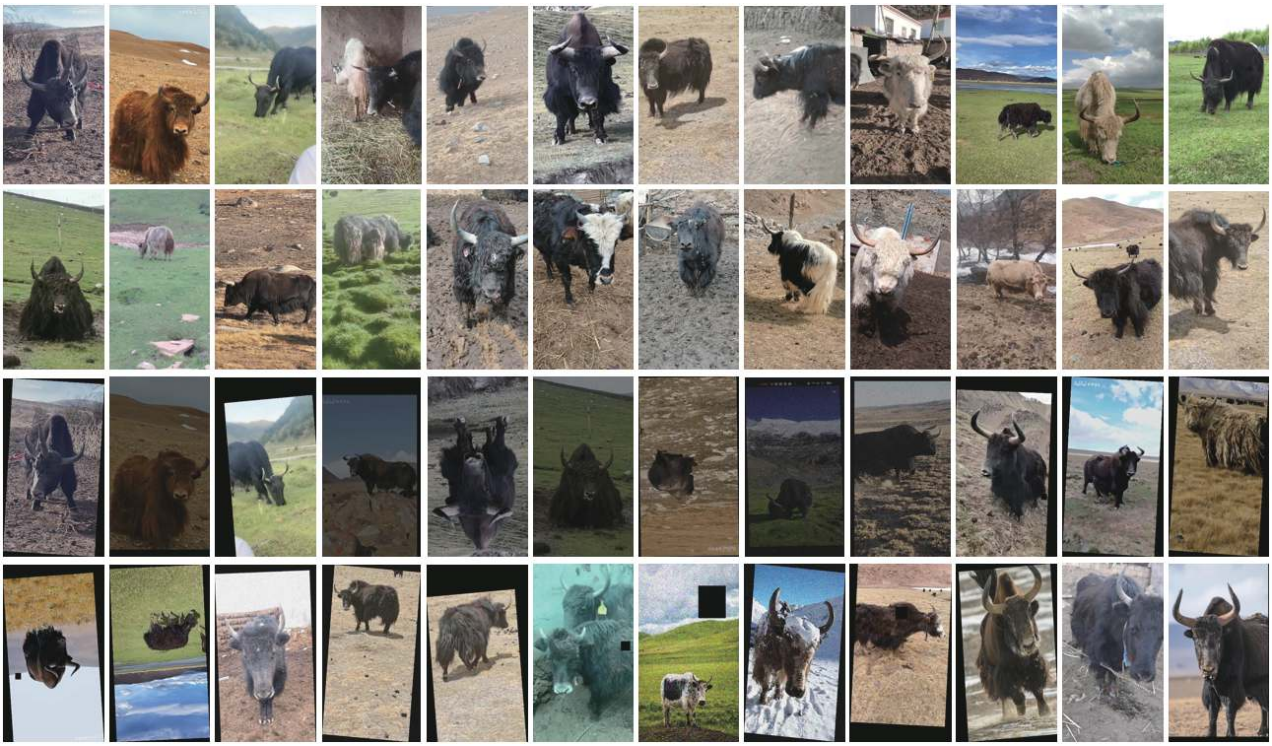


图 1 牦牛目标检测部分数据集

1.2 跟踪评价数据集

为了全面评估跟踪算法在实际放牧环境下的性能, 选取 10 段不同条件下的视频段, 记为 video01–10. 每段视频的时长均在 10 s 以上, 分辨率为 1920 像素×1080 像素. 数据集包含牦牛活动频繁与较少场景、目标拥挤与稀疏场景. 使用 Darklabel 软件标注, 得到 10 个 CSV 文件, 内容包括所有帧中实际牦牛的 ID、位置和大小等信息, 用来评估跟踪算法的准确度和鲁棒性.

2 YOLOv5

2.1 算法结构及原理

YOLO (you only look once)[14] 最初由 Redmon 等提出. 相比于 Faster RCNN[15] 算法的两阶段检测, YOLO 接收整张图片作为输入, 经过推理后直接输出目标框位置、类别信息和检测置信度大小. YOLOv5 有 4 个版本的检测网络, 分别是 YOLOv5s, YOLOv5m, YOLOv5l 和 YOLOv5x[16–18]. 其中最小、最浅的是 YOLOv5s, 其余 3 种都是在此基础上不断加深加宽的. YOLOv5s 模型文件大小只有 14.1 MB, 计算参数少, 故本文选择在此基础上进行改进和提升, 以达到更好的训练效果.

YOLOv5s 网络由 4 个通用模块组成, 分别是输入端 (Input), 骨干网络 (Backbone), Neck 网络和 Head 预测输出层.

Input: 通常包括图像预处理操作, 如将图像缩放到适应网络的输入大小并进行归一化处理等. 该模块使用包括随机缩放、裁剪和排布等操作的 Mosaic 数据增强方式, 以此提高模型的预测精度. 此外, YOLOv5 使用一种自适应锚框计算方法来减少冗余信息并加快网络的训练速度.

Backbone: 实质上是卷积神经网络, 用于在不同图像粒度上提取特征. YOLOv5s 网络使用 Focus 和 CSP 结构. Focus 结构的关键操作是切片, 例如将 4×4×3 的特征图经过切片后, 尺寸变成 2×2×12. 值得注意的是, YOLOv5s 网络中的 Focus 结构使用 32 个卷积核进行卷积操作, 而其他 3 种网络的卷积核数量均有所增加.

YOLOv5s 中有两种 CSP 结构, CSP1_X 位于骨干网络中, CSP2_X 位于 Head 预测输出层. 在骨干网络中加入 CSP, 可以增强网络的学习能力, 降低计算复杂度, 使网络更轻量化, 同时提高查准率.

Neck: 位于骨干网络和预测输出层之间, 用于加工特征信息. 该模块使用特征金字塔网络 (feature pyramid network, FPN)[19] 和路径聚合网络 (path aggregation network, PANet)[20] 多尺度的融合特征, 结构如图 2 所示. FPN 是自顶向下的, 通过上采样向低层传递高层的强语义特征, 增强特征金字塔的语义信息. PANet 则相反, 是自底向上的, 通过下采样融合低层特征和高层特征, 以增强高层特征的定位信息. 经过 FPN 和 PANet 融合的特征, 不同尺寸的特征图都包含图像的语义信息和位置信息, 以此保证对不同尺寸图片的准确预测.

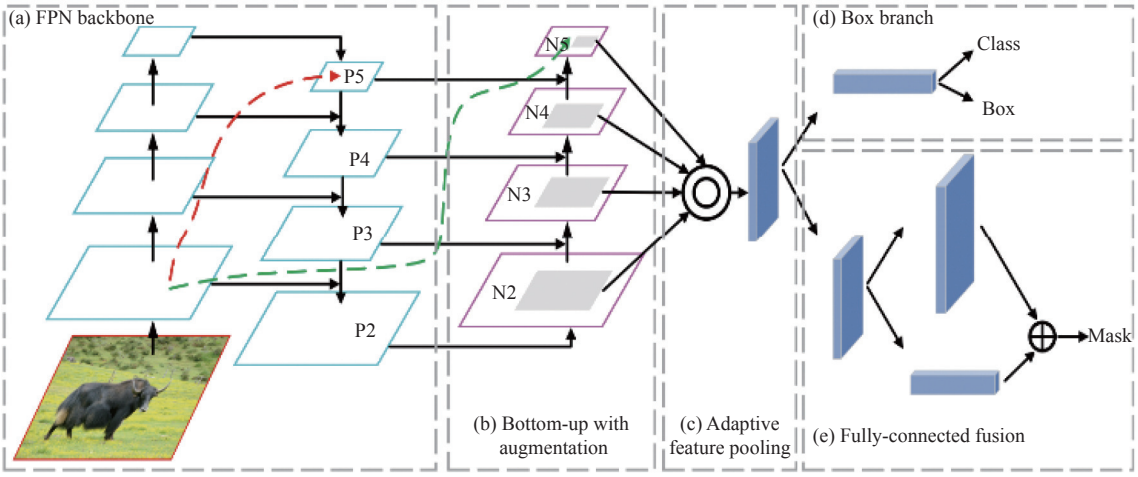


图 2 FPN 和 PANet 的网络架构

Head: 输出目标检测结果. 该层沿用之前 YOLOv3 的检测头. 对于不同的网络结构, 输出层的分支个数不尽相同, 但通常都包含一个分类分支和一个回归分支.

2.2 模型的改进与优化

针对原 YOLOv5s 模型在牦牛检测任务上误检漏检率高、小目标检测效果不好等问题, 通过改进其骨干网络和 Neck 网络, 实现更精确的牦牛检测. 改进的 YOLOv5s 网络如图 3 所示. 在骨干网络中加入改进的 Swin Transformer 模块, 并使用空洞空间卷积池化金字塔 (atrous spatial pyramid pooling, ASPP) 以多比例提取图片的上下文信息, 增强网络对小目标的检测效果. 颈部使用双向特征金字塔网络跨尺度融合特征图, 通过增加同层级网络间的跳转连接, 以保留原始节点的未融合信息; 同时加入改进的协同注意力 (coordinate attention, CA) 机制, 以获取较多的远程依赖关系. 在预测输出层, 从上到下分别是融合特征图的 1/8、1/16、1/32、1/64 倍下采样后的特征信息. 使用二元交叉熵计算置信度损失 (obj_loss) 和分类损失 (cls_loss)、EIoU loss (efficient intersection over union) 计算定位损失, 采用非极大值抑制算法筛除冗余目标框. 改进普通卷积模块的激活函数为 *FReLU* (funnel ReLU). 相较于 SiLU, *FReLU* 有更快的收敛速度、更好的泛化能力和稀疏性, 同时减少计算量, 从而提高模型的鲁棒性.

$$FReLU = \max(0, x) + \min(a \times (x - m)) \quad (1)$$

其中, x表示输入, a和m为可学习的参数.

2.2.1 D-STB 模块

Transformer[21] 是一种基于自注意力机制的深度神经网络模型, 在计算机视觉领域应用广泛. 基于 Transformer 的网络模型在目标检测领域取得了显著的性能提升, 因为它能提取图像的全局信息并关注重要的区域. 但在像素级别上进行预测的视觉任务的自注意力计算复杂度是图像大小的二次方, 这限制了 Transformer 在高分辨率图像处理任务中的应用. 而 Swin Transformer[22] 有效解决了 Transformer 的应用缺陷, 它将自注意力计算限制在窗口区域内, 并允许跨窗口进行信息交互. 因此, 本研究中在 YOLOv5s 骨干网络添加 Swin Transformer block (STB), 以此增强骨干网络的特征提取能力.

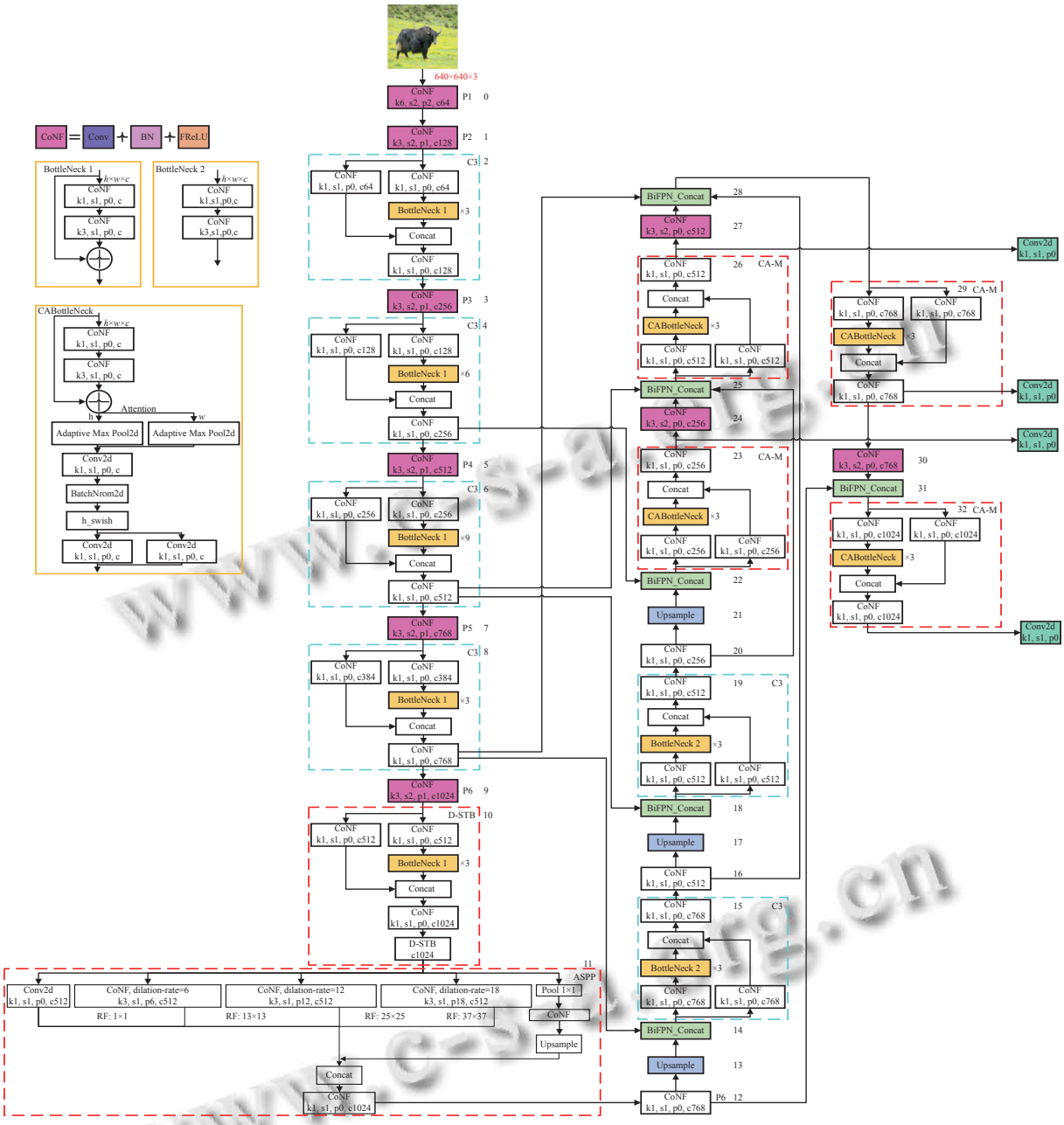


图 3　改进的 YOLOv5s 网络模型结构图

Swin Transformer 网络结构如图 4 所示. 首先, 输入为 [H×W×3] 的图像会被传入图块分割层 (patch partition), 该层将每个大小为 M×M 的像素块划分为一个图块, 并在通道方向展开, 从而将图像维度变为 [H/4, W/4, 48]. 接下来, 线性嵌入层对通道数做线性变换, 将图像维度进一步变为 [H/4, W/4, C], 然后将其输入 STB 进行自注意力计算, 提取图像特征. 模块的输出将成为下一阶段的输入. 阶段 2–4 的操作相同, 先使用图块拼接层 (patch merging) 将上一个阶段的输出特征图中相邻的大小为 M×M 的窗口合并, 然后将结果送入 STB 构建分层特征图. 其中, 图块拼接层实现下采样和维度变换.

Swin Transformer 在每个阶段间使用图块拼接 (patch merging) 实现图像下采样来构建分层特征图, 如图 5 所示. 当对特征图进行 4×或 8×的下采样时, 图像被分割成多个小尺寸的图块. 然后将图块输入到

STB 中提取特征. 考虑到模型可能存在过度参数化问题而导致模型过拟合, 故在 STB 模块的残差连接层后添加一层 DropBlock 层, 对卷积层提取的特征图中移除相邻区域, 以此提高模型的泛化能力, 使卷积神经网络可以更好地提取有用信息. 同时减少模型参数量, 降低模型的计算复杂度. 实验结果表明使用 block_size=7 时可以获得最佳准确度. 改进的 STB 结构 (D-STB) 如图 6(a) 所示, 由 4 个 LN 层、1 个 W-MSA (windows multi-head self-attention) 层、1 个 SW-MSA (shifted windows multi-head self-attention) 层、2 个二层的多层感知机 (multi layer perceptron, MLP)、4 个残差连接层和 4 个 DropBlock 层组成. 其中, W-MSA 和 SW-MSA 交替组成基于窗口的 multi-head 自注意力模块. 输入到 D-STB 的特征z^{l-1}先经过 LN 层进行归一化, 然后将结果送入 W-MSA 层提取特征. 接着进行残差层和 DropBlock 层计算得到$\hat{z}^l$, 然后再次归一化后输入到 1 个使用 GELU 非线性激活函数的 MLP 中做通道维度的线性变换, 最后通过残差连接层和 DropBlock 层得到经过 W-MSA 处理的输出特征$\hat{z}^l$. 把$\hat{z}^l$输入到包含 SW-MSA 层的相似模块中计算[23]. D-STB 引入了残差连接层, 以解决神经网络的退化问题.

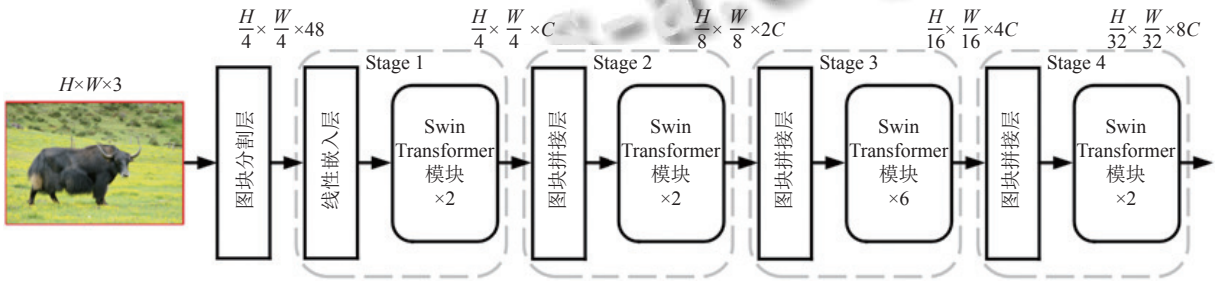


图 4 Swin Transformer 网络架构图

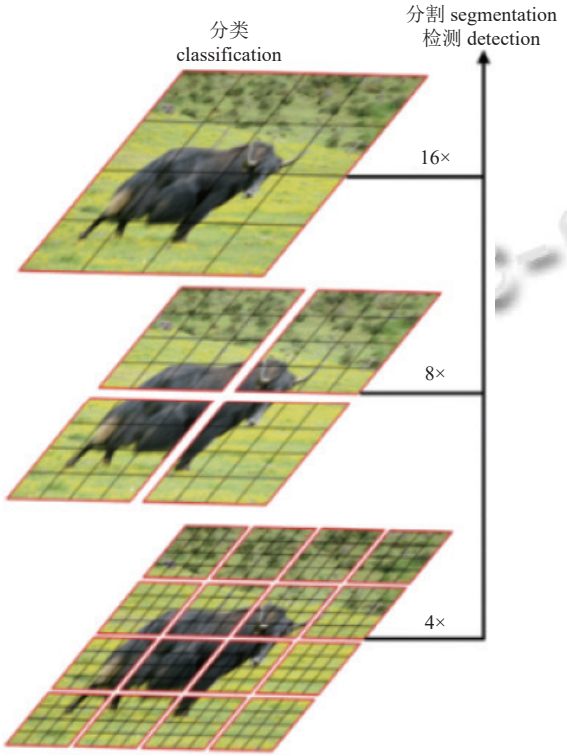
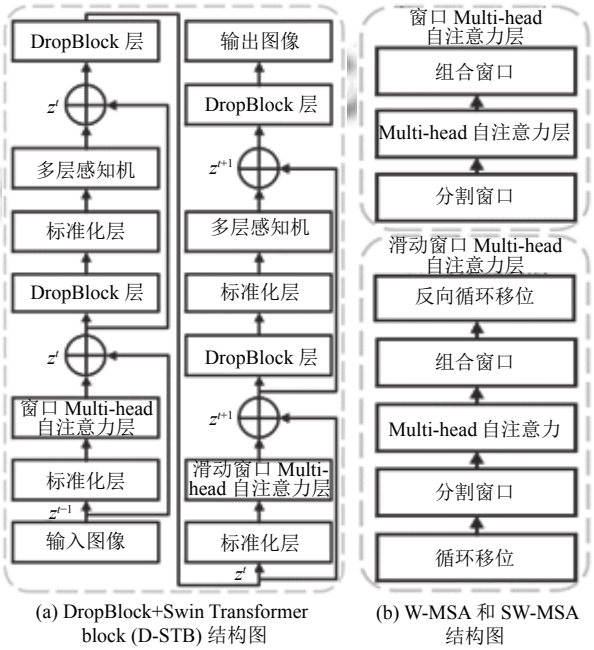


图 5 Swin Transformer 构建的分层特征图



(a) DropBlock+Swin Transformer block (D-STB) 结构图 (b) W-MSA 和 SW-MSA 结构图

图 6 Swin Transformer 模块结构

D-STB 中的 W-MSA 和 SW-MSA 结构如图 6(b) 所示. 其中, W-MSA 包括分割窗口 (window partition)、

组合窗口 (window reverse) 和 MSA 计算. 分割窗口是指将特征图从左上角像素划分为多个 $M\times M$ 的互不重叠的独立窗口. 组合窗口用于将 W-MSA 计算的特征拼接还原为完整的 multi-head 自注意力特征图. MSA 用于在窗口内部进行 multi-head 自注意力计算, 使计算复杂度与图像大小成线性关系, 从而降低模型的训练成本. 但也隔绝了窗口间的信息交互, 从而导致全局特征缺失的问题. SW-MSA 通过移位操作解决了该问题, 它将原本不相邻的像素点组成独立窗口计算 MSA, 实现跨窗口间的信息传递, SW-MSA 窗口移位如图 7 所示.

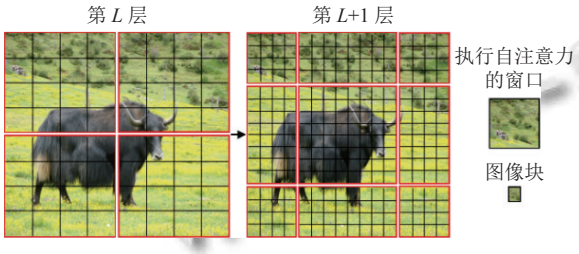


图 7 SW-MSA 中的窗口移位方法

2.2.2 ASPP 金字塔

空洞卷积也叫扩张卷积或膨胀卷积, 其原理是在卷积核元素之间填充一些空格 (零) 来扩大卷积核感受野, 以此来获取图像更多的特征信息. 改进的 YOLOv5s 网络中存在池化采样操作, 导致部分特征信息丢失. 同时 Swin Transformer 在进行 4×和 8×倍率下采样时, 相较于 Vision Transformer 一直不变的 16×倍率下采样, 会使卷积核感受野变小; 且在 D-STB 提取特征时, 使用 DropBlock 层移除了特征图的部分信息. 而空洞卷积通过在不同尺度下使用不同的空洞卷积核来获取上下文信息, 得到不同尺寸的特征图. 这样就可以在不增加网络参数的情况下增大卷积核感受野来提取更多的全局特征, 同时不丢失空间分辨率, 保持像素点的空间位置不变.

对于空洞数为 d 的膨胀卷积, 卷积结果为:

$$S(i,j)=\sum_m\sum_n I[i+m(d+1)+1,j+n(d+1)+1]K(m,n) \tag{2}$$

其中, K 为当前卷积核大小, $(d+1)K+1$ 等价于一个新的卷积核, $d+1$ 为膨胀比. 该卷积核的首行、首列、尾行、尾列权重均是零, 每间隔 d 个像素点的权重非零, 否则权重为零.

ASPP 结构如图 8 所示, 对输入特征图使用 6、12、18 扩张率 (dilation rate) 的多个并行空洞卷积层并行采样. 同时, 经过 1×1 的池化层、1×1 的卷积层和上采样后, 将得到的特征图连接到一起扩大通道数. 最后使用 1×1 的卷积将通道数降低到预期的数值. 注意要谨慎选择扩张率, 因为过大的扩张率可能会产生无意义的权重.

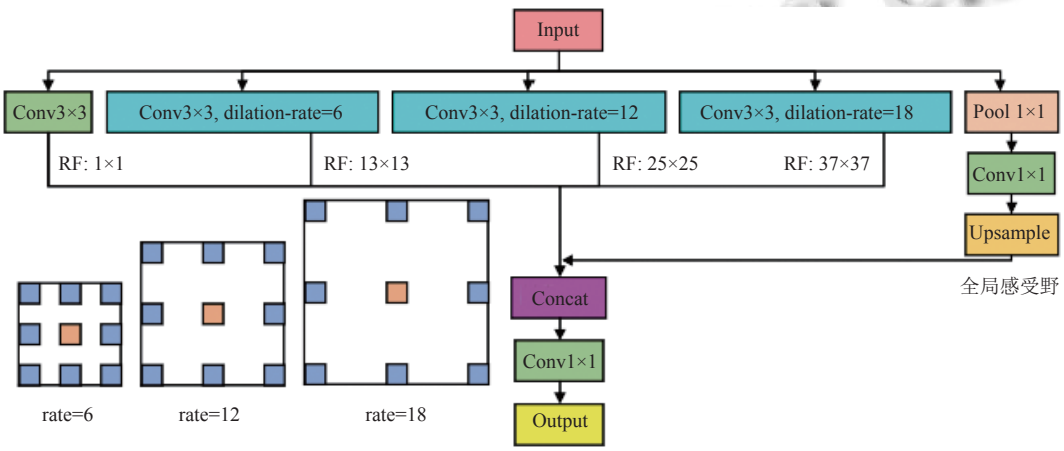


图 8 空洞空间卷积池化金字塔结构图

2.2.3 跨尺度特征融合

目标检测查准率的高低受特征图信息表达多样性的影响, 融合多尺度特征是提高准确率的重要手段. YOLOv5 使用 FPN + PANet 实现高层语义信息和低层细节信息的交流. 为了进一步增强多尺度信息的融合能力, 本研究借鉴了双向特征金字塔网络 (bi-directional feature pyramid network, BiFPN)[24] 的结构优势, 并将其思想迁移到 YOLOv5s 的特征融合网络中, 结构如图 9

所示. 通过添加同一层级的输入和输出结点之间的跳跃连接, 使同层网络的特征图可以互相共享特征信息, 增强了特征图表达的多样性.

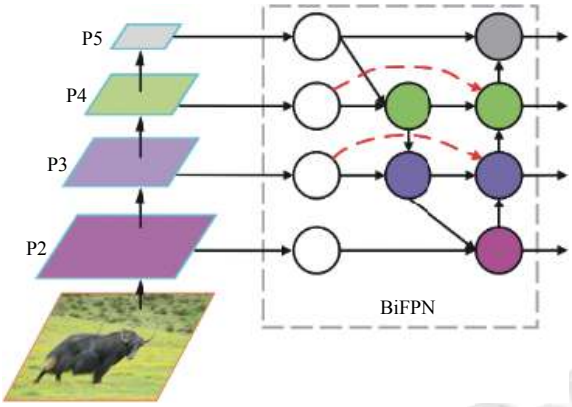


图 9 跨尺度特征融合模块结构图

2.2.4 CA-M 注意力

为了提高神经网络的检测性能, 注意力机制被广泛应用. 然而, 考虑到 SE (squeeze-and-excitation network)[25] 注意力仅关注构建通道之间的依赖关系, 忽略了位置信息. CBAM (convolutional block attention module)[26] 注意力引入大尺度的卷积核提取空间特征, 但普通卷积操作只能提取局部空间关系, 无法获取大范围空间依赖关系. 此外, 大多数注意力机制的计算开销较大. 鉴于本文待检测数据的目标是牦牛, 其中部分目标较大且分布疏密不均. 为了增强网络对牦牛特征提取的能力, 本文引入 “协同注意力机制”[27]. 该机制的核心思想是将候选框的位置信息编码到信道注意力中, 避免二维池化将特征张量转化为单个特征向量而造成信息丢失, 进而使网络可以关注大范围的位置信息. 考虑到牦牛有较多显著的特征, 如牦牛头大、肩部隆起、耳朵较小、黑色犄角等, 原 CA 注意力的全局平均池化可以有效减少参数量, 但无法提取上述显著特征. 因此, 为了帮助网络更好地捕捉牦牛的显著特征, 改进后的 CA 注意力用全局最大池化层替换全局平均池化层. 改进的 CA-M 注意力结构如图 10 所示. 先将输入的特征分解为两个一维特征, 分别沿着 x 和 y 方向聚合特征进行全局最大池化操作. 这样在捕获一个空间方向远程依赖关系的同时, 获得另一个方向的位置结构信息. 具体地, 高度 h 处的第 c 通道的输出为:

$$z_c^h(h)=\max_{0\leqslant i<W} x_c(h,i) \tag{3}$$

同理, 宽度 w 处的第 c 通道的输出为:

$$z_c^w(w)=\max_{0\leqslant j<H} x_c(j,w) \tag{4}$$

然后通过 Concat 操作聚合 x 方向和 y 方向生成的 attention map, 使通道叠加, 接着使用卷积 Conv2d 和非线性函数变换, 将位置信息映射编码到各自的 x 和 y 方向的中间特征, 结果进行归一化处理后将新特征沿着 x 和 y 方向分割, 再分别进行卷积 Conv2d 和 Sigmoid 激活, 最后进行 Re-weight 重新加权, 至此得到一个基于空间维度的注意力机制[28,29].

最后, CA-M 注意力的数学表达式为:

$$y_c(i,j)=x_c(i,j)\times g_c^h(i)\times g_c^w(j) \tag{5}$$

通过引入 CA-M 注意力, 网络不仅获取远程依赖相关性, 还得到了的位置信息. 同时, CA-M 将获取的特征图编码为方向感知和位置感知的注意图, 然后一起输入特征图中, 从而使网络更专注于位置和语义信息, 抑制不重要的特征, 进而提高网络的预测准确率[28]. 在改进的 YOLOv5s 结构中, 跨尺度特征融合实现了高层语义信息和低层强定位信息间的交互, 同时融合了原输入结点的特征信息. 基于这一思路, 本文将 CA-M 添加在跨尺度特征融合模块之后. 实验证明, 改进的 CA-M 注意力可以有效提升模型的检测性能, 且不增加计算开销.

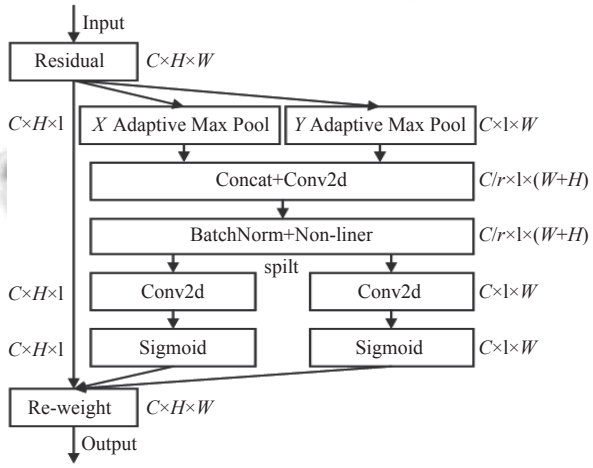


图 10 CA-M 注意力结构图

3 Bytetrack

3.1 算法结构及原理

Bytetrack 跟踪器的轻量级网络架构可实现牦牛自然放牧场景下的多目标实时跟踪和计数. Bytetrack 是一种基于 Tracking-By-Detection 范式的多目标跟踪方

法, 通过引入检测器, 可以在目标跟踪过程中进行目标的重新检测和更新, 从而更加准确地跟踪目标. Bytetrack 算法流程图如图 11 所示. 首先, 使用一个目标检测器在第 1 帧图像中检测牦牛. 然后提取特征, 如颜色、形状、纹理等. 在接下来的帧中, 使用卡尔曼滤波算法来估计牦牛在图像中的运动, 并微调牦牛目标的位置. 然后在微调后的位置周围, 使用检测器重新检测牦牛目标, 并提取更新后的特征. 其次, 进行目标匹配, 使用匈牙利算法将更新后的特征与前一帧中的目标进行匹配来确定当前帧中的目标与前一帧的目标是否为同一个目标. 匹配会产生 3 种结果: 1) 匹配成功, 跟踪目标轨迹并更新状态; 2) 未匹配到检测框, 为其新建一个跟踪轨迹; 3) 匹配失败, 删除轨迹. 匹配失败的跟踪轨迹会保留 30 帧的生命线, 如果在 30 帧以内重新匹配目标成功, 则转结果 1); 如果超过 30 帧仍没有匹配到目标, 则删除轨迹. 通过不断重复上述过程, 实现视频实时多目标跟踪.

Bytetrack 和其他跟踪算法的区别在于, 它不仅保留高分检测结果, 也保留了低分检测框. 通过去除低分框中的背景信息, 挖掘出真正的目标 (遮挡、模糊等困难目标), 从而降低漏检并提高轨迹的连贯性. 然而, Bytetrack 算法没有采用外观特征匹配, 因此跟踪器的性能很大程度上取决于检测器的性能. 如果检测器性能好, 跟踪效果会很好; 如果检测性能不佳, 将严重影响跟踪器的性能. 因此, 在实际应用中, 需要综合考量检测器和跟踪器, 以达到更好的跟踪效果.

在数据关联部分, Bytetrack 只使用卡尔曼滤波来预测当前帧中的轨迹在下一帧的位置. 然后通过预测框和实际检测框的 IoU 来计算两次匹配的相似度, 并用匈牙利匹配算法完成匹配. 数据关联的具体步骤如下.

(1) 根据检测框得分, 把结果分为高分框和低分框, 分别进行处理.

(2) 用高分框和之前的跟踪轨迹进行匹配.

(3) 用低分框和步骤 (2) 与高分框匹配失败的跟踪轨迹进行匹配.

(4) 对未匹配成功且得分高的检测框, 为其新建一个跟踪轨迹. 对未匹配到检测框的跟踪轨迹, 为其保留 30 帧的生命线.

3.2 模型的改进与优化

Bytetrack 算法使用非极大值抑制 (non-maximum suppression, NMS) 去除冗余的检测框, 保留最有可能的目标框. NMS 选择得分最高的检测框 (假设为 A), 计算 A 与剩余目标框的 IoU 值, 当 IoU 值超过阈值时进行抑制, 即将目标框得分设置为 0. 处理一轮后, 在剩下的框中继续寻找得分最高的, 再抑制与其 IoU 值超过阈值的检测框, 直到处理完所有的检测框. 然而, 该算法需要手动设置阈值, 而阈值的大小会影响筛选重叠目标框的准确性, 可能会导致误检. 此外, 抑制低于阈值的目标框的方法过于简单, 忽略了一些可能是遮挡、模糊等原因导致得分低的目标. 因此, 可以改进算法来重新挖掘被错过的目标.

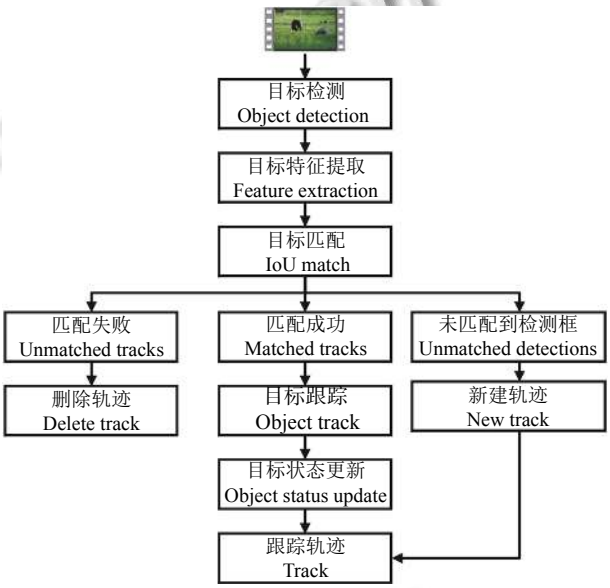


图 11 Bytetrack 算法流程图

将 IoU 改为 *DIoU* 来评估是否进行抑制操作, 这样可以避免 IoU 评估方法所导致的误判, 尤其是在目标框相互包含时, *DIoU* 能够更好地确定冗余框和保留框.

$$DIoU = \frac{\rho^2(A,B)}{c^2} \tag{6}$$

其中, $d=\rho^2(A,B)$是 A 框与 B 框中心点坐标的欧氏距离, c是包含两个目标框的最小方框的对角线长度. *DIoU* 在目标框不重叠时, 仍可为边界框提供移动方向. 对于包含两个框在水平或垂直方向时, *DIoU* 能加快回归速度. 且保持 NMS 阈值不变时, 可以得到更高的精确度和召回率.

4 实验与结果分析

4.1 实验环境

本实验使用 PyTorch 1.13.1 深度学习框架、Python 3.7.12 环境, 操作系统为 Linux 5.4, CPU 型号为因特尔 Core(TM) i5-3470、8 GB 内存, 使用 GeForce GTX

1080Ti 图形处理器 (graphics processing unit, GPU)、CUDA 11.5 和 CUDNN 7.6.5 加速.

在训练过程中使用统一超参数配置. 输入图片大小为 640×640, 消融实验迭代 300 个 epoch, 模型性能评估实验迭代 100 个 epoch, batch size 为 64. 采用 Adam (adaptive moment estimation, 自适应矩估计) 优化器, 初始化学习率为 0.001. 开始阶段使用 warmup 预热学习率, 预热阶段前 3 个 epoch 采用一维线性插值调整学习率, 预热阶段的动量因子为 0.8, 随后使用余弦退火算法更新学习率.

4.2 评价指标

4.2.1 目标检测评价指标

为了科学分析各模型在牦牛检测任务上的性能表现, 使用经典的目标检测评价指标[30].

$F1$ 分数 ($F1$-$score$): 指精确率和召回率的调和平均数.

$$F1\text{-}score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{7}$$

检测精确度 ($Precision$): 指目标检测模型预测出的真正例的概率.

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

召回率 ($Recall$): 指在模型判断为正例且真实类别也是正例的数量占所有真实正例的比例.

$$Recall = \frac{TP}{TP + FN} \tag{9}$$

其中, n表示类别总数, TP表示真正例, FP表示假正例, FN表示假反例.

平均精确度均值 (mean average precision, mAP): 指所有类别AP的均值. mAP大小在 [0, 1] 区间, 越大越好. 其中 mAP@0.5 表示置信度阈值 IoU>0.5 的mAP, mAP@0.5:0.95 表示在不同 IoU 阈值 (从 0.5 到 0.95, 步长为 0.05) 上的平均mAP.

$$mAP = \frac{\sum_{i=1}^{n} AP(i)}{n} \tag{10}$$

4.2.2 目标跟踪评价指标

本实验采用多目标跟踪基准[31] 中使用的指标来评价牦牛跟踪算法的性能, 其中包括衡量单摄像头下多目标跟踪准确度的指标$MOTA$ (multiple object tracking accuracy)、识别IDF_1分数 (identification $F1$-$score$)、召回率$Recall$、实际目标框数量GT、未命中目标总数FN和帧率FPS.

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \tag{11}$$

$$IDF_1 = \frac{2IDTP}{2IDTP + IDFP + IDFN} \tag{12}$$

$$IDR = \frac{IDTP}{IDTP + IDFN} \tag{13}$$

其中, t为时间索引, FP为误检数量. $IDSW$表示目标被遮挡再次被检测到时, 如果ID发生变化, 则定义为发生一次$IDSW$. $IDTP$、$IDFP$、$IDFN$分别代表真正ID数、假正ID数和假负ID数.

4.3 实验结果与分析

4.3.1 目标检测

本文实验在 YOLOv5s 网络的基础上, 添加具有 4 个检测尺度的检测层, 其将更低层级的特征图引入到特征融合网络中, 使网络能捕获更丰富的细粒度特征, 从而提高牦牛检测精度. 消融实验结果如表 1 所示, 其中“CA”是指使用原 CA 模块替换颈部的部分 C3 模块; “CA-M”是指使用本文改进的 CA 注意力替换颈部的部分 C3 模块; “跨尺度特征融合”是指使用本文的跨尺度特征融合模块替换原特征融合网络; “STB”是指在骨干网络中加入原 Swin Transformer 模块; “D-STB”是指在骨干网络中加入改进的 Swin Transformer 模块.

表 1 消融实验结果对比

Model	$F1$-$score$ (%)	$Precision$ (%)	$Recall$ (%)	mAP@0.5 (%)	mAP@0.5:0.95 (%)	参数量 (M)	模型大小 (MB)
YOLOv5s	90	90.4	90.5	93.8	70.8	7.02	14.3
YOLOv5s+CA	92	90.8	92.8	95	77.2	7.15	14.8
YOLOv5s+CA-M	92	93.3	91.2	95.7	82.2	7.12	14.7
YOLOv5s+STB	90	89.4	90.4	92.6	68.1	7.28	15.0
YOLOv5s+D-STB	91	90.1	91.7	93.3	81.2	7.25	15.0
YOLOv5s+D-STB+CA-M	92	93.5	91.3	96.2	84.1	7.40	15.4
YOLOv5s+D-STB+CA-M+跨尺度特征融合	93	93.3	93.3	96.5	83.1	12.84	26.4
本文改进模型	**94**	**94.7**	**93.4**	**96.9**	**83.2**	**21.08**	**42.9**

由表 1 可知, YOLOv5s 网络模型的参数量为 7.02M、模型大小为 14.3 MB, YOLOv5s+CA 模型的参数量为 7.15M、模型大小为 14.8 MB. 保持参数量和模型大小在较小范围内变化时, YOLOv5s+CA 模型相比 YOLOv5s 的精度提升了 0.4%, 召回率提升 2.3%, 说明 CA 注意力机制可以有效提升牦牛的检测精度. 对比 YOLOv5s+CA-M 和 YOLOv5s+CA 模型可知, 在 CA 注意力中将全局平均池化层替换为全局最大池化层后, 可以更好地帮助网络提取牦牛的显著特征, 使检测精确度提升 2.5%. YOLOv5s+STB 模型比 YOLOv5s 的检测性能低, 本文认为 STB 在构建分层特征图时进行 4×和 8×下采样时卷积核感受野变小, 网络无法提取足够多的全局特征信息, 且它提取的是图像初步的、细节性的信息, 特征表达单一, 导致检测精确度下降. 但对比 YOLOv5s+STB 和 YOLOv5s+D-STB 模型可知, 在加入 DropBlock 层后, 网络的检测精确度提升 0.7%, 因为 DropBlock 层随机移除特征图的相邻区域, 从而提升了模型的泛化性能, 且加入 DropBlock 层后, 减少了模型的参数量, 从而降低了模型部署的硬件要求. YOLOv5s+D-STB+CA-M 比 YOLOv5s+D-STB 模型的精确度提升 3.4%, 平均检测精度 *mAP*@0.5 提升 2.9%. YOLOv5s+D-STB+CA-M+跨尺度特征融合模型相比 YOLOv5s 模型, 精确度提升 2.9%, 召回率提升 2.8%. 本文模型比 YOLOv5s 的精确度提升 4.3%, 召回率提升 2.9%, 平均精确率 *mAP*@0.5 提升 3.1%, 它具备 Swin Transformer 强大的特征提取能力; 空洞空间卷积池化金字塔多比例的提取图像上下文信息, 且不丢失空间分辨率; 跨尺度特征融合更有效的融合来自 D-STB 的细节信息和深层的语义特征, 使网络更深层的颈部和预测层的特征图有更丰富的高级语义信息; CA-M 注意力机制同时获得远程依赖相关性和空间结构信息; 各个模块协同增强网络特征提取和表达的多样性, 进而提升模型的检测性能.

值得注意的是, YOLOv5s+CA-M、YOLOv5s+D-STB+CA-M 模型相比本文模型, 在平均检测精确度上分别降低了 1.4% 和 1.2%, 本研究认为原本的特征融合网络使用简单的卷积和池化操作, 导致网络特征表达受限. 而本文改进模型中的跨尺度特征融合不仅融合全局特征和空间位置信息, 还融合输入图像特征, 增强了特征图的多样性, 进而提高了检测效果.

为验证改进后模型的有效性, 本文使用原 YOLOv5s、YOLOX[32]、以 VGG 为骨干特征提取网络的一阶段检测模型 SSD[33] 和以 ResNet101 为骨干特征提取网络的二阶段检测模型 Faster RCNN 进行对比. 从表 2 可以看出, 本文模型在牦牛检测任务上的检测精度、召回率和平均精确度都优于其他模型. 相比 SSD 模型, 本文改进模型在参数量较少的情况下提升了检测性能. 相比 Faster RCNN 模型大小, 本文模型只有 42.9 MB, 有效降低了模型在实际应用场景中的硬件要求.

表 2 模型性能评估实验结果

Model	*F*1-*score* (%)	*Precision* (%)	*Recall* (%)	*mAP*@0.5 (%)	*mAP*@0.5:0.95 (%)	参数量 (M)	模型大小 (MB)
YOLOv5s	94	92.2	95.2	97.6	87.2	7.02	14.4
SSD	94	92.48	95.06	—	—	26.29	90.5
YOLOX	92	93	91.02	90.37	86.22	8.94	68.5
Faster RCNN	85.25	90.1	80.9	98.3	75.5	60.11	460.23
本文改进模型	**95**	**95.7**	**94.6**	**98.7**	**88.8**	**21.08**	**42.9**

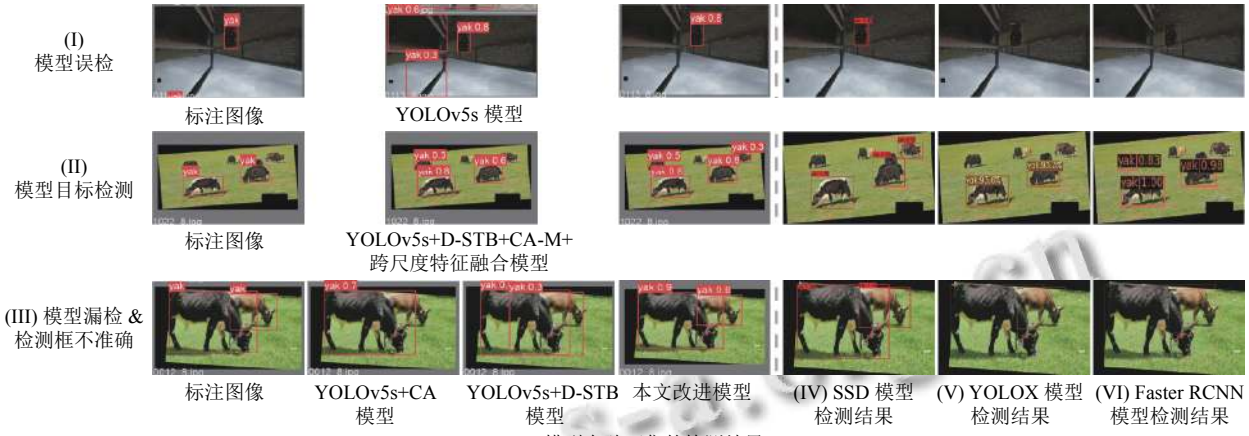
注: SSD仅统计了mAP结果, mAP=96.81%

为了更直观地展示出改进模型的优势, 使用改进前后的各个模型在验证集上进行检测, 结果如图 12(a). 对比可知原 YOLOv5s 模型存在误检漏检的情况, 如图 12(a)(I). 当牦牛躯干被遮挡时, YOLOv5s+CA 模型会漏检目标, YOLOv5s+D-STB 模型检测锚框定位不精准, 如图 12(a)(III). 相比之下, YOLOv5s+D-STB+CA-M+跨尺度特征融合模型和本文模型的检测效果较好. 且加入 D-STB 模块后, 提升了模型对小目标的检测效果. 然而在检测相同图片时, 本文模型的检测置信度比 YOLOv5s+D-STB+CA-M+跨尺度特征融合模型的高, 检测性能更好, 如图 12(a)(II). SSD、YOLOX 和 Faster RCNN 模型在验证集上的检测结果如图 12(a)中的 (IV)、(V) 和 (VI), 可看出各个模型的检测性能表现良好, 但牦牛密集且较小时的检测效果略逊于本文模型. 由上述分析可知, 本文模型可以有效解决牦牛检测中遮挡而导致检测难度大、误检漏检的问题, 模型在测试集上的检测结果如图 12(b) 所示.

4.3.2 多目标跟踪

Bytetrack 跟踪器的检测器先采用 YOLOv5s 和本文改进模型做跟踪对比实验. 然后使用本文模型对改

进前后的 Bytetrack 算法做对比实验. 选取 3 个评估视频测试跟踪器性能, 包括运动幅度从低到高的牦牛吃草 video01、行走 video02 和打架 video03, 实验结果如表 3 所示.



(a) 模型在验证集的检测结果

(b) 本文改进模型在测试集的检测结果

图 12 牦牛检测结果

表 3 目标跟踪结果对比

Bytetrack	det_model	Video sequence	GT	IDF_1 (%)	*Recall* (%)	*FN*	MOTA (%)	*FPS*
改进Bytetrack前	YOLOv5s	video01	1 185	85.893 1	75.274 3	293	75.274 3	29
		video02	743	81.481 5	68.75	185	68.75	33
		video03	766	84.862	74.282	197	73.498 7	47
		mean	898	84.078 9	72.768 8	225	75.507 7	36
	本文模型	video01	1 185	90.379 3	82.447 3	208	82.447 3	27
		video02	743	85.473 7	74.632 4	169	74.632 4	22
		video03	766	85.671 9	75.326 4	189	74.804 2	19
		mean	898	87.175 0	77.468 7	189	77.294 6	23
改进Bytetrack后	本文模型	video01	1 185	96.732	93.670 9	75	93.670 9	20
		video02	743	92.063 5	85.294 1	140	85.294 1	22
		video03	766	85.373 1	74.673 6	194	74.412 5	20
		mean	**898**	**91.389 5**	**84.546 2**	**136**	**84.459 2**	**21**

由表 3 可知, Bytetrack 跟踪模型的检测器采用本文模型相比原 YOLOv5s, 使 MOTA 提升 1.786 9%, IDF_1 提升 3.096 1%, 未命中目标数 *FN* 降至 189, 牦牛跟踪准确率提升明显. 这进一步说明本文模型在牦牛检测任务上表现良好, 然而推理速度相比采用 YOLOv5s 的推理速度降低了 13 f/s. 因为本文模型中的 D-STB 和 ASPP 金字塔的计算耗时较长. 在推理速度降低 2 f/s 时, 改进 Bytetrack 算法后的 MOTA 提升 7.164 6%, IDF_1 提升 4.214 5%, 召回率 *Recall* 提升 7.077 5%, 跟踪器性能表现更好. 因为改进的 *DIoU* 在牦牛被遮挡或检测锚框重合时, 会综合考虑目标框的中心点距离和重叠情况, 计算出更适合抑制的 IoU 阈值, 从而获得更高的 IDF_1 和召回率, 提高 Bytetrack 模型低分框匹配高分框失配轨迹的成功率.

为了更加直观地显示出本文改进模型的优势, 使用改进前后的 Bytetrack 模型在跟踪评价数据集的测试集视频上来检验牦牛的跟踪结果, 选取两段视频进行测试, 分别为牦牛吃草 video01 和打架 video02, 结果如图 13 所示. 将牦牛跟踪结果视频按照 1 秒 30 帧来切帧, 然后取其第 1, 31, …, 301 秒的图片. 对

比图 13(I) 中第 2 秒和第 3 秒以及图 13(II) 中第 1 秒和第 4 秒的结果可知, 改进后的 Bytetrack 跟踪模型, 提高了对低得分检测框的复用率, 从而提高模型跟踪的准确率. 同时, 结果图的左上角会实时统计当前跟踪牦牛目标的数量并显示, 达到快速检测统计牦牛数量的目的.

(I) 牦牛吃草

(II) 牦牛打架 video02

图 13 牦牛跟踪结果对比

5 结论

(1) YOLO 是目前应用广泛的目标检测模型, 本文通过向原 YOLOv5s 网络的骨干网络中添加改进的 Swin Transformer 模块和空洞空间卷积池化金字塔, 以多尺度的方式提取图像上下文信息. 并改进融合特征的 Neck 网络, 提高了特征表达的多样性, 同时添加改进的 CA 注意力机制以获取大范围空间位置信息. 本文改进模型实现了部分遮挡状态下或小目标牦牛的检测, 并可以减少漏检和误检情况的发生. 结果表明本文模型的平均检测精确度 *mAP*@0.5 为 98.7%, 检测性能优于原 YOLOv5s、SSD 和 Faster RCNN 等模型.

(2) Bytetrack 跟踪器采用轻量级的模型架构, 有助于实现实时的牦牛多目标跟踪. Bytetrack 跟踪器只采用运动模型, 在检测器性能较好的情况下, 卡尔曼滤波的预测准确性能替代 ReID 模型进行长时间目标关联. Bytetrack 的检测器使用本文模型和原 YOLOv5s 进行实验, 结果表明采用本文模型的 MOTA 为 77.294 6%, 跟踪效果优于使用原 YOLOv5s 模型的跟踪器. 改进 Bytetrack 的抑制标准为 *DIoU*, MOTA 提升了 7.164 6%. 改进后的 Bytetrack 提高了从低分检测框中重新挖掘出牦牛目标和轨迹匹配成功的比例, 从而降低漏检并提高多目标跟踪的准确度, 同时实现快速检测统计牦牛数量, 可有效提高人工养殖牦牛的管理效率. 本研究为青藏高原地区牦牛检测和跟踪提供了技术支持.

参考文献

1 吉林农业大学. 我国牦牛市场与产业调查分析报告. 农产品市场周刊, 2021, (23): 54–55.
2 赵洪文, 罗晓林, 安添午. 一种基于视频数据的牦牛计数方法: 中国, 107330403B. 2020-11-17.
3 Gabriel M, Cha S, Al-Nakash NYB, *et al*. Wildlife detection and recognition in digital images using YOLOv3: Extended abstract. Proceedings of the 2020 IEEE Cloud Summit. Harrisburg: IEEE, 2020. 170–171.
4 何嘉. 基于深度学习的野生动物智能检测与识别 [硕士学位论文]. 深圳: 深圳大学, 2019. [doi: 10.27321/d.cnki.gszdu.2019.000334]
5 张海峰, 沈媛萍. RFID 技术在动物识别与跟踪管理中的应用. 青海畜牧兽医杂志, 2012, 42(3): 36–38.
6 王柯力, 袁红春. 基于迁移学习的水产动物图像识别方法. 计算机应用, 2018, 38(5): 1304–1308, 1326.
7 陈占琦, 张玉安, 王文志, 等. 基于迁移学习的多尺度特征融合牦牛脸部识别算法. 智慧农业 (中英文), 2022, 4(2): 77–85.
8 Bertinetto L, Valmadre J, Henriques JF, *et al*. Fully-convolutional Siamese networks for object tracking. Proceedings of the 2016 European Conference on Computer Vision. Amsterdam: Springer, 2016. 850–865.
9 Li B, Yan JJ, Wu W, *et al*. High performance visual tracking with Siamese region proposal network. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 8971–8980.
10 蔡前舟, 郑伯川, 曾祥银, 等. 结合长尾数据解决方法的野生动物目标检测. 计算机应用, 2022, 42(4): 1284–1291.
11 韩家臣. 基于深度学习的野生动物图像识别方法研究 [硕士学位论文]. 兰州: 西北师范大学, 2021. [doi: 10.27410/d.cnki.gxbfu.2021.001828]
12 Deng J, Dong W, Socher R, *et al*. ImageNet: A large-scale

hierarchical image database. Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition. Miami: IEEE, 2009. 248–255.

13 魏贤哲, 卢武, 赵文彬, 等. 基于改进 Mask R-CNN 的输电线路防外破目标检测方法研究. 电力系统保护与控制, 2021, 49(23): 155–162. [doi: 10.19783/j.cnki.pspc.210482]

14 Redmon J, Divvala S, Girshick R, *et al*. You only look once: Unified, real-time object detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 779–788.

15 Ren SQ, He KM, Girshick R, *et al*. Faster RCNN: Towards real-time object detection with region proposal networks. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2015. 91–99.

16 Glenn JK, Gold MH. Purification and characterization of an extracellular Mn(II)-dependent peroxidase from the lignin-degrading basidiomycete, Phanerochaete chrysosporium. Archives of Biochemistry and Biophysics, 1985, 242(2): 329–341. [doi: 10.1016/0003-9861(85)90217-6]

17 宁纪锋, 林靖雅, 杨蜀秦, 等. 基于改进 YOLOv5s 的奶山羊面部识别方法. 农业机械学报, 2023, 54(4): 331–337.

18 葛云飞, 祁云嵩, 孟祥宇. YOLOv5 改进的轻量级口罩人脸检测. 计算机系统应用, 2023, 32(3): 195–201. [doi: 10.15888/j.cnki.csa.009021]

19 Lin TY, Dollár P, Girshick R, *et al*. Feature pyramid networks for object detection. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 936–944.

20 Liu S, Qi L, Qin HF, *et al*. Path aggregation network for instance segmentation. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 8759–8768.

21 Vaswani A, Shazeer N, Parmar N, *et al*. Attention is all you need. Proceedings of the 31st International Conference on Information Processing Systems. Long Beach: Curran Associates Inc., 2017. 6000–6010.

22 Liu Z, Lin YT, Cao Y, *et al*. Swin Transformer: Hierarchical vision transformer using shifted windows. Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision. Montreal: IEEE, 2021. 9992–10002.

23 郑楚伟, 林辉. 基于 Swin Transformer 的 YOLOv5 安全帽佩戴检测方法. 计算机测量与控制, 2023, 31(3): 15–21. [doi: 10.16526/j.cnki.11-4762/tp.2023.03.003]

24 Tan MX, Pang RM, Le QV. EfficientDet: Scalable and efficient object detection. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 10778–10787.

25 Hu J, Shen L, Sun G. Squeeze-and-excitation networks. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 7132–7141.

26 Woo S, Park J, Lee JY, *et al*. CBAM: Convolutional block attention module. Proceedings of the 15th European Conference on Computer Vision. Munich: Springer, 2018. 3–19.

27 Hou QB, Zhou DQ, Feng JS. Coordinate attention for efficient mobile network design. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 13708–13717.

28 朱高兴, 于璨. 基于 YOLOv5-CA 算法的野生动物目标检测研究. 信息技术与信息化, 2022, (6): 32–35.

29 沈杰, 黄晓华. 基于改进 YOLOv5s 算法的危险区域入侵报警. 计算机系统应用, 2023, 32(3): 157–162. [doi: 10.15888/j.cnki.csa.009019]

30 Abbas A, Jain S, Gour M, *et al*. Tomato plant disease detection using transfer learning with C-GAN synthetic images. Computers and Electronics in Agriculture, 2021, 187: 106279. [doi: 10.1016/j.compag.2021.106279]

31 Ristani E, Solera F, Zou R, *et al*. Performance measures and a data set for multi-target, multi-camera tracking. Proceedings of the 2016 European Conference on Computer Vision. Amsterdam: Springer, 2016. 17–35.

32 Ge Z, Liu ST, Wang F, *et al*. YOLOX: Exceeding YOLO series in 2021. arXiv:2107.08430, 2021.

33 Liu W, Anguelov D, Erhan D, *et al*. SSD: Single shot multibox detector. Proceedings of the 14th European Conference on Computer Vision. Amsterdam: Springer, 2016. 21–37.

(校对责编: 牛欣悦)