计算机科学与探索
Journal of Frontiers of Computer Science and Technology
1673-9418/2024/18(09)-2221-18
doi: 10.3778/j.issn.1673-9418.2402044

YOLO系列目标检测算法综述

徐彦威[1,2],李　军[1,2+],董元方[3],张小利[4]
1. 吉林财经大学 管理科学与信息工程学院,长春 130117
2. 吉林财经大学 人工智能研究中心,长春 130117
3. 长春理工大学 经济管理学院,长春 130022
4. 吉林大学 计算机科学与技术学院,长春 130015
+ 通信作者 E-mail: lijun@jlufe.edu.cn

摘　要:近年来,基于深度学习的目标检测算法是计算机视觉研究热点,YOLO算法作为一种优秀的目标检测算法,其发展历程中网络架构的改进,对于提高检测速度和精度起到了重要作用。对YOLOv1~YOLOv9的整体框架进行了横向分析,从网络架构(骨干网络、颈部层、头部层)、损失函数方面进行了对比分析,充分讨论了不同改进方法的优势和局限性,具体评估了改进方法对模型精度的提升效果。讨论了数据集的选择与构建方法、不同评价指标的选择依据,及其在不同应用场景中的适用性和局限性,深入研究了在五个应用领域(工业、交通、遥感、农业、生物)YOLO算法的具体改进,并对检测速度、检测精度及复杂度之间的平衡进行探讨。分析了YOLO在各领域的发展现状,通过具体实例总结YOLO算法研究中存在的问题,并结合应用领域的发展趋势,展望YOLO系列算法的未来,详细探讨了YOLO算法的四个研究方向(多任务学习、边缘计算、多模态结合、虚拟和增强现实技术)。

关键词:YOLO算法;目标检测;计算机视觉;特征提取;卷积神经网络
文献标志码:A　**中图分类号:**TP391

Survey of Development of YOLO Object Detection Algorithms

XU Yanwei[1,2], LI Jun[1,2+], DONG Yuanfang[3], ZHANG Xiaoli[4]
1. School of Management Science and Information Engineering, Jilin University of Finance and Economics, Changchun 130117, China
2. Center for Artificial Intelligence, Jilin University of Finance and Economics, Changchun 130117, China
3. School of Economics and Management, Changchun University of Science and Technology, Changchun 130022, China
4. College of Computer Science and Technology, Jilin University, Changchun 130015, China

Abstract: In recent years, deep learning-based object detection algorithms have been a hot topic in computer vision research, with the YOLO (you only look once) algorithm standing out as an excellent object detection algorithm. The evolution of its network architecture has played a crucial role in improving detection speed and accuracy. This paper conducts a comprehensive horizontal analysis of the overall frameworks of YOLOv1 to YOLOv9, comparing the network architecture (backbone network, neck layers and head layers) and loss functions. The strengths and limitations of different improvement methods are thoroughly discussed, with a specific evaluation of the impact of these improvements on model accuracy. This paper also delves into discussions on dataset selection and construction

基金项目:国家自然科学基金(61801190)。
This work was supported by the National Natural Science Foundation of China (61801190).
收稿日期:2024-02-26　**修回日期:**2024-05-23

methods, the rationale behind choosing different evaluation metrics, and their applicability and limitations in various application scenarios. It further explores specific improvement methods for YOLO algorithm in five application domains (industrial, transportation, remote sensing, agriculture, biology), and discusses the balance among detection speed, accuracy, and complexity in these application domains. Finally, this paper analyzes the current development status of YOLO in various fields, summarizes existing issues in YOLO algorithm research through specific examples, and in conjunction with the trends in application domains, provides an outlook on the future of the YOLO algorithm. It also offers detailed explanations for four future research directions of YOLO (multi-task learning, edge computing, multimodal integration, virtual and augmented reality technology).

Key words: YOLO algorithm; object detection; computer vision; feature extraction; convolutional neural network

目标检测是计算机视觉领域的一项关键任务，其核心目标是从图像或视频序列中准确地确定物体的存在、种类及其空间位置。目标检测任务既包含分类问题，即将物体划分到预定义的类别中，又包含回归问题，即定位物体的准确位置。图1展示了目标检测任务的处理结果，目标检测算法可以取得目标物体的中心点坐标及其宽度和高度，从而绘制矩形选择框，并且得到目标物体类别。目标检测算法通常可以分为两大类：传统检测方法和深度学习检测方法。传统检测方法依赖于手工设计的特征提取和相对浅层的可训练模型。其基本工作流程如下：首先通过在输入图像上滑动不同大小的窗口，产生一系列候选框。然后采用基于颜色、纹理、形状等特性，以及中等或高层次的语义特征，如尺度不变特征转换（scale-invariant feature transform，SIFT）[1]、哈尔特征（Haar）[2]、方向梯度直方图（histogram of oriented gradients，HOG）[3]等，提取候选框中的局部信息。然后通过分类器，如支持向量机（support vector machine，SVM）[4]、自适应增强（adaptive boosting，AdaBoost）[5]、随机森林（random forest）[6]等对提取的特征进行分类。最后采用非极大值抑制（non-maximum suppression，NMS）[7]算法，以去除冗余的检测框，从而获得最优的目标检测位置。传统方法通过有效的候选框生成、特征提取和分类策略，实现目标物体的可靠检测。尽管在资源受限或对解释性要求较高的情景中，传统方法呈现一定的优势，然而当前阶段的图像识别系统通常针对特定任务，数据规模有限，且泛化能力较差，难以实现精确的实际识别效果。近年来，深度学习迅速发展，并引入能学习语义、高层次、深度特征的工具以解决传统体系结构的问题，进一步提升模型在网络架构、训练策略和优化方面的性能。2012年，深度卷积神经网络首次被应用于大规模图像分类任务。目前，主流的深度学习目标检测算法分为双阶段检测算法和单阶段检测算法。双阶段检测算法的典型代表为Fast R-CNN[8]系列，由于对候选区域进行细粒度处理，通常检测精度较高。并且在复杂场景下具有较强的鲁棒性。但是该类方法需要进行候选区域生成和特征分类两阶段处理，故模型结构复杂，计算复杂度高，实时性较差。单阶段检测算法的典型代表为YOLO（you only look once）[9-15]系列算法。YOLO算法将检测和分类任务合并处理，模型结构简单，实时性强。YOLO算法设计了输入端、骨干网络、颈部网络、输出端等模块。从YOLOv1到YOLOv9，及其多个改进版本，已被广泛应用于医学、工业、农业等多个领域。然而，目前对于YOLO目标检测算法的综述[16-20]中，缺乏将YOLO系列的骨干网络、颈部层、头部层、锚框、损失函数等部分进行拆分对比，并对其改进点及改进原因进行阐述的研究。本文旨在系统分析YOLO算法骨干网络、颈部层、头部层、锚框、损失函数等模块、阐明改进策略及作用，并探讨研究YOLO算法在工业检测、交通检

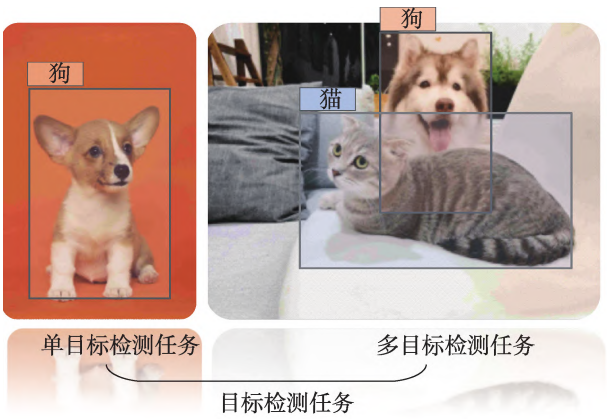

图1 目标检测任务

Fig.1 Object detection task

测、遥感检测、农业检测和生物检测等五个领域的应用。旨在为研究者改进YOLO模型、促进YOLO算法在各行各业的应用提供帮助。

1 相关工作

本文梳理并研究了现有YOLO目标检测算法综述，其中，Terven等人[16]对YOLOv1到YOLOv8[9-14]的改进点进行了研究，按照时间线讨论了网络架构的主要变化以及模型的训练技巧，并研究了增强实时目标检测系统的潜在研究方向。Diwan等人[17]详细地回顾了单阶段检测算法，研究了YOLO算法的网络结构、损失函数及算法性能，并对比了单阶段检测算法的标准性能。Hussain[18]从工业制造的角度研究了YOLOv1至YOLOv8的改进。Sirisha等人[20]研究了YOLO系列算法的性能指标、损失函数以及YOLO算法变体版本的网络架构设计、性能和应用。王琳毅等人[19]回顾了YOLOv1~YOLOv7目标检测算法改进，并且总结了输入、特征提取和输出三阶段下的轻量化网络构建和交并比（intersection over union，IoU）损失优化的改进。现有YOLO算法综述对YOLO主要模块的对比分析不够充分，多为按时间线的讨论，本文对YOLO系列主流算法YOLOv1~YOLOv9进行了深入的剖析，将YOLO算法模块拆解，详细阐述了各版本算法在骨干网络、颈部层、头部层、锚框、损失函数等几个方面的改进及其原因。同时，本文还列举了YOLO系列算法常用的数据集及其评价指标。针对YOLO算法在五个细分领域的应用，对近期相关的改进文章进行了分析[21-29]。这些领域包括工业检测（钢表面缺陷检测、输电线路高巡检）、交通检测（小目标检测、车辆目标检测）、遥感检测（无人机巡检）、农业检测（麦穗检测、受灾树木实时检测）和生物检测（奶山羊个体识别检测、蛋鸡啄羽异常行为检测）。本文分析各领域算法要求，指出潜在改进策略，如骨干网络重构、颈部层引入注意力机制等，并探讨YOLO算法在各个行业的应用前景。尽管YOLO算法在许多方面表现出色，但存在不同场景适应性、小目标精准定位、正负样本均衡等问题。本文对其进行了深入剖析，并展望了YOLO算法的未来发展。

2 YOLO算法系列横向对比

近年来，随着更深层、更复杂的网络架构的提出，YOLO算法的特征提取能力不断加强，YOLO算法经过长时间的发展，已迭代到v9版本。图2展示了YOLO算法的时间发展线。新的激活函数的更新和发展，也使得YOLO算法训练得更加高效、稳定。注意力机制的发展使得基于YOLO算法的目标检测精度大大提高。除此之外，随着YOLO算法不断地被用于各种领域，如医疗、交通、工业、农业、遥感，针对不同领域改进的YOLO算法如雨后春笋般涌现出来，这些改进算法吸纳了其他算法的灵感思路，也进一步地促进了YOLO系列算法的发展。

2.1 YOLO网络结构比较

YOLO算法的流程是，首先将图片划分为 $S\times S$ 的网格[9]，每个网格会预测生成两个长宽比不同的边界框 B，并且每个边界框都有5个参数，分别表示边界框 B 中心点的横坐标、纵坐标、高度和宽度以及边界框 B 内包含物体的概率，每个网格预测（$B\times 5+C$）个值，其中 C 为类别数。然后根据阈值去除置信度比较低的边界框，最后利用非极大值抑制（NMS）[7]算法去除重合的边界框，得到最终的预测结果。

为了实现上述操作，YOLO算法主要由四个部分组成：输入侧（Input）、骨干网络（Backbone）、颈部（Neck）和头部（Head）。这些组成部分共同构成了YOLO系列模型的核心结构，使得模型高效、准确。

输入侧（Input）：该部分主要负责接收输入数据，同时应用数据增强算法和特殊的预处理操作，以确

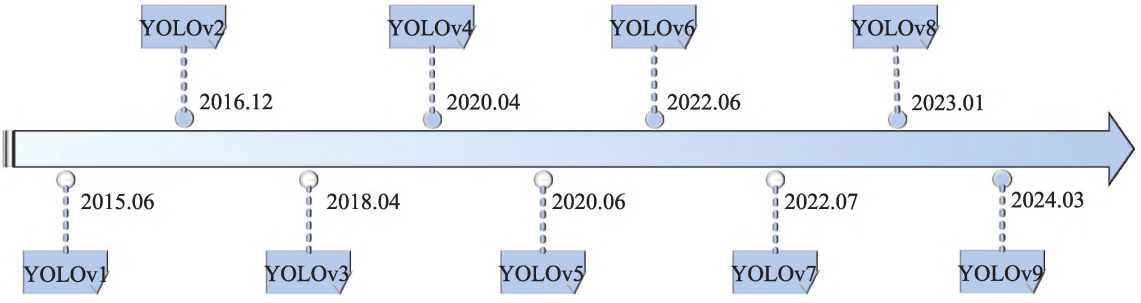

图2 YOLO算法发展线

Fig.2 Timeline of YOLO algorithms

保模型对各种输入数据具有良好的适应性。

骨干网络(Backbone):骨干网路采用一种通用的神经网络结构,可以广泛应用于不同的计算机视觉细分领域。通过优化网络结构和参数设置,骨干网络能够有效地从输入数据中学习并提取有用的特征表示。

颈部(Neck):构建在骨干网络与头部之间的颈部,用于汇集和融合来自不同层的特征图。在YOLOv3中首次引入,并成为后续系列模型不可或缺的一部分。通过在颈部对特征图进行细粒度调整和处理,模型能够更好地适应不同任务和数据集。

头部(Head):该部分负责预测对象类别和边界框,包含检测头结构、损失函数部分以及优化策略。

图3展示了YOLO算法重要版本的网络架构图。表1对比了YOLO各版本主要网络结构。近年来,随着深度学习的不断发展。新型网络结构,如残

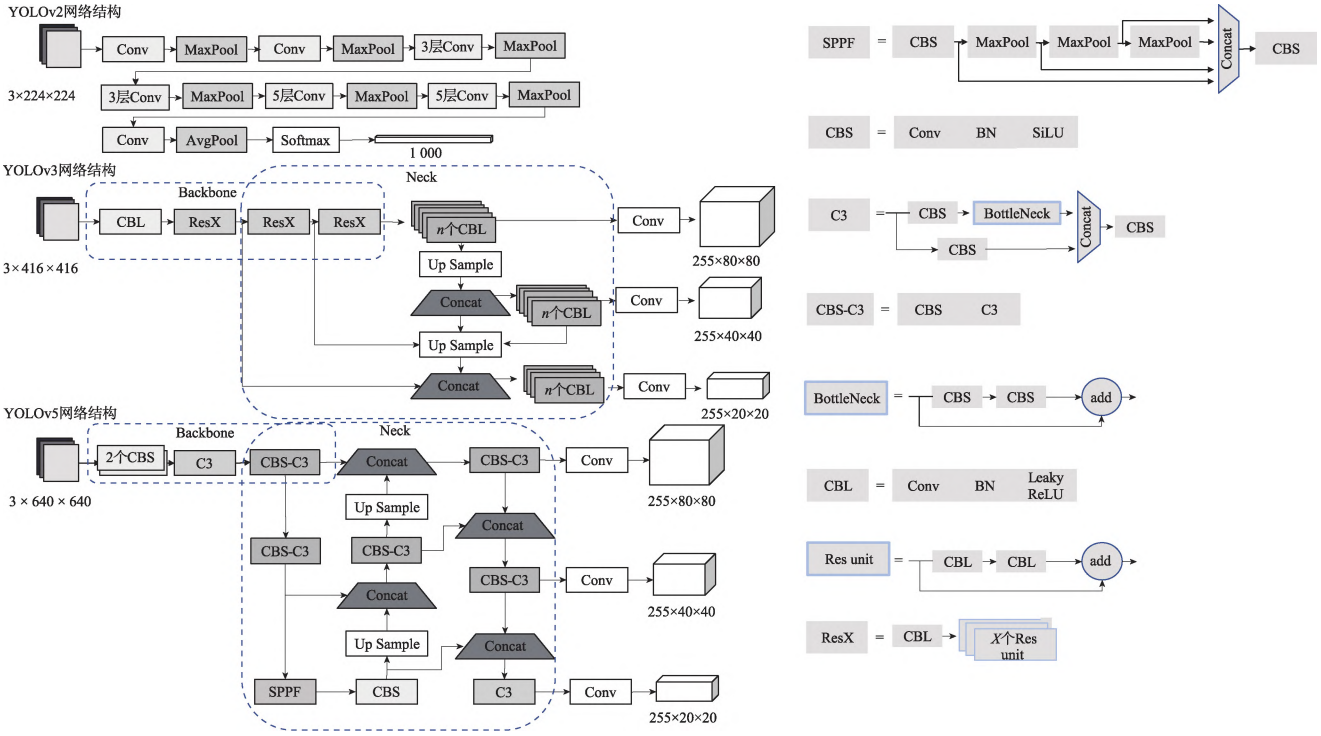

图3 YOLO重要版本网络架构图

Fig.3 Network architecture of important YOLO versions

表1 YOLO各版本主要网络结构及锚框对比

Table 1 Comparison of main network structures and anchor boxes of YOLO versions

版本	骨干网络	颈部网络	头部网络	锚框
v1	GoogleNet提取特征,Dropout防止过拟合	—	全连接网络	Bounding box
v2	Darknet-19替代GoogleNet减少计算量,BN、ReLU替代Dropout提高收敛速度	—	Coupled head	Anchor-based
v3	Darknet-53加深网络提高特征提取能力,Leaky ReLU修复“神经元死亡”问题	FPN融合低层和高层特征信息,实现多尺度预测	Coupled head	Anchor-based
v4	Darknet-53加入CSP丰富梯度信息,Mish、DropBlock改善梯度消失和梯度爆炸	SPP改善范围野,PAN丰富FPN多尺度特征	Coupled head	Anchor-based
v5	CSPDarknet-53引入Focus减少信息损失	C3替换CBL,残差丰富特征	Coupled head	Anchor-based
v6	EfficientRep提取特征,减少训练和推理时间	Rep-PAN提高推理速度	Decoupled head	Anchor-free
v7	ELAN网络控制梯度距离,进一步加强特征提取能力	SPP基础上融合更多特征,E-ELAN提高特征学习能力	Decoupled head	Anchor-based
v8	CSPDarknet-53引入C2f丰富梯度流	C2f替换C3,细化特征融合	Decoupled head	Anchor-free
v9	GELAN网络梯度路径规划保留重要特征	GELAN保留更多特征信息	Decoupled head	Anchor-free

差、跨阶段局部（cross stage partial，CSP）等模块不断引入到骨干网络中，YOLO算法的特征提取能力也越来越强。自从YOLOv4引入路径聚合网络（path aggregation network，PAN）以来，颈部网络的设计逐渐趋向于成熟。为了进一步提升算法实时性，头部网络发展过渡到解耦合头（decoupled head）设计。锚框设计的主流也变为了无锚框（anchor-free）设计。

2.1.1 骨干网络的改进

在目标检测领域，YOLO算法以其高效性和准确性备受关注。其中，模型的Backbone结构起着核心特征提取的作用。YOLOv1[9]使用GoogLeNet[30]作为骨干网络，利用不同大小的卷积核提取多样的特征。虽然GoogLeNet可以准确识别目标物体，但是其计算复杂度相对较高，会导致算法检测速度较低。YOLOv1引入了Dropout[31]随机屏蔽神经元，防止过拟合，但在训练过程中，使得损失函数收敛速度变慢。Darknet-19网络精度与VGG-16[32]相近，计算复杂度更低，YOLOv2[13]引入Darknet-19[13]网络很好地解决了GoogLeNet网络计算复杂度高的问题，使其在PASCAL VOC07数据集上，平均精度均值（mean average precision，mAP）提高了0.4个百分点[13]。虽然Darknet-19较少的参数量可以提高算法速度，但在复杂情景下，由于层数较少，该网络无法充分提取深层次图像特征，导致算法精度降低。为了弥补Dropout[31]在收敛速度上的不足，YOLOv2采用了批归一化（batch normalization，BN）[33]和ReLU激活函数来防止过拟合，使模型在PASCAL VOC07数据集上，mAP值提高了2.4个百分点[13]。YOLOv3[10]采用Darknet-53[10]网络，弥补了Darknet-19深层特征提取不充分的劣势，并且大量残差结构解决了层数过深导致的退化问题，进一步增强了模型对于小目标、复杂场景的检测能力，使其在ImageNet数据集精度提高了3.1个百分点[10]。虽然Darknet-53网络加深网络层数能够提高网络深层特征提取能力，但是也降低了算法推理速度。YOLOv3在每个卷积层后使用BN层和Leaky ReLU[34]激活函数，修复了ReLU函数“神经元死亡”的问题。但是，Leaky ReLU激活函数不够平滑，可能会出现梯度爆炸和梯度消失的问题。

CSPNet[35]是一种可以将梯度变化从头到尾集成到特征图中的一种网络结构。YOLOv4[14]将Darknet-53网络与CSPNet网络相结合，提出了新的网络结构CSPDarknet-53[10,14]，该网络梯度组合更加丰富，并且优化了网络重复梯度信息，相比Darknet-53网络减少了计算量，提高了算法推理速度。但是该网络仍存在一些局限性，如模型缩放时深度模型收敛性逐渐恶化的问题，计算开销仍然较大。YOLOv4采用改进的BN层和Mish[36]激活函数，有利于避免梯度爆炸和梯度消失的问题，提高网络训练速度和稳定性，使模型在ImageNet数据集精度提高了0.9个百分点[14]。YOLOv4使用DropBlock[37]解决Dropout信息仍可能传递至下一层的问题，进一步防止过拟合。此外，YOLOv4引入空间注意力模块（spatial attention module，SAM）[38]提高模型精度，SAM更关注重要图形特征，使其平均精度值（average precision，AP）提高了0.3个百分点[14]。YOLOv5引入Focus模块对图片进行切片操作，减少下采样带来的信息损失，进一步提高模型的精度。YOLOv6[11]使用EfficientRep[11]网络有效利用GPU计算资源，减少计算开销。YOLOv7引入双分支网络结构ELAN（effective long-range aggregation network）[12]模块，控制梯度路径距离，使网络学习到更多特征，在MS COCO17数据集上，为模型的AP值带来3.6个百分点的提高，并且提升了21%的推理速度。YOLOv7[12]卷积层后使用BN[33]层和Sigmoid线性单元（Sigmoid linear unit，SiLU）激活函数防止过拟合。YOLOv8将C3替换为C2f模块，为网络带来了更加丰富的梯度流，提高了算法精度。YOLOv9[15]引入通用高效层聚合网络（generalized efficient layer aggregation network，GELAN），梯度路径规划可以保留更多重要特征，提高算法精度。

2.1.2 颈部层的改进

在YOLO系列模型中，Neck颈部结构设计在汇聚和融合多层特征方面起到了关键作用。在早期的YOLOv1[9]和YOLOv2[13]版本中，并未引入Neck结构，采用卷积的方式虽然可以提取图像特征，但是会存在像素错位问题。从YOLOv3开始，Neck结构被引入，并采用特征金字塔网络（feature pyramid network，FPN）[39]，充分利用低层特征的高分辨率特性和高层特征的丰富语义信息。并且，上采样还原特征图可以缓解像素错误的问题。此外，还可以实现多尺度特征的独立预测，从而显著提升了小物体的检测效果。但是，FPN在特征融合过程中可能会造成信息丢失的问题。在YOLOv4[14]的Neck中，使用空间金字塔池化（spatial pyramid pooling，SPP）[40]模块替代常规池化层，SPP将特征图进行三次不同的卷积。不同尺度卷积的融合扩大了感受野。YOLOv4还采用PAN[41]自底向上地融合多尺度特征，保留了更多浅层位

置特征,优化了FPN信息不完整的问题。在YOLOv5的Neck中,采用C3结构替换CBL(Conv BN Leaky_ReLU),C3将特征图一分为二,将进行多层卷积和无变化的两路特征图融合。残差连接的卷积方式可以增强网络特征融合的能力。虽然C3可以减少参数量,但是梯度流信息不够丰富。YOLOv6[11]引入Rep-PAN,Rep-PAN基于RepVGG,是一种可重参数化的网络,将PAN中的CSP-Block替换为RepBlock,降低算法在硬件上的延时,提高推理速度。YOLOv7[12]将Neck层与Head层合并,引入SPPCSPC(spatial pyramid pooling cross stage partial connection)模块,SPPCSPC在SPP的基础上增加了Concat操作,与其之前的特征图进行融合。同时,PAN模块引入扩展高效层聚合网络(extended efficient layer aggregation networks,E-ELAN)[12]结构,在不破坏原始梯度路径的情况下提高了网络的学习能力。YOLOv8将YOLOv5Neck层中的C3模块替换成C2f模块,C2f模块是一种基于C3模块的多尺度信息提取融合技术,在进行卷积同时采用PAN方式融合卷积特征。更细粒度的特征融合可以获得更加丰富的梯度流信息。在YOLOv7Neck层的基础上,YOLOv9[15]将PAN模块的E-ELAN结构改进为通用高效层聚合网络(GELAN),进一步保留特征提取中更多特征信息,GELAN模块可以为模型带来0.2个百分点的AP值提升[15]。

2.1.3 头部层的改进

YOLO系列中的检测头是用来进行目标检测的部分,包括一些卷积层、池化层和全连接层等。主要负责对骨干网络提取的特征图进行多尺度目标检测。YOLOv1[9]没有严格意义上的Head层,最后的全连接层相当于Head层。由于使用卷积对特征图进行下采样会损失很多细粒度特征,导致小物体的识别效果不佳。YOLOv2在YOLOv1[9]的基础上去掉了最后的全连接层,并引入Passthrough结构,Passthrough是一种将特征图一分为四,并进行Concat操作的结构。多特征图融合的方法保留了更多细粒度特征。引入Passthrough层可以在PASCAL VOC07数据集上,为模型mAP值带来1.0个百分点的提升[13]。为了提高对于中小目标的检测精度,YOLOv3[10]的Head检测头在YOLOv2的基础上引入多尺度检测逻辑和多标签分类思想,加强对小目标的检测,减少漏检,提高检测精度。YOLOv4[14]的Head检测头沿用YOLOv3的整体架构。为了丰富不同尺度的特征信息,YOLOv5的Head检测头在YOLOv4的基础上采用多层级特征融合的方法,首先将骨干网络输出的特征图通过Conv模块进行通道数降维和特征图缩放,再将不同层级的特征图进行融合,从而得到更加丰富的特征信息,提高检测性能。YOLOv6[11]的Head检测头改进了解耦检测头结构,将分类任务和回归任务分离,从而提高模型检测精度,解耦检测头可以将AP值提高1.4个百分点。YOLOv7[12]的Head检测头使用了辅助头(auxiliary head)训练,以浅层网络权重的辅助损失为指导,用于辅助网络的训练,辅助头的设计,给模型带来0.5个百分点的AP值提升[12]。YOLOv8、YOLOv9[15]参考YOLOv6,使用了Decoupled-Head,使用两个卷积分别完成分类和回归任务。为解决数据通过深度网络传输时丢失的问题,即“信息瓶颈和可逆函数”,保留前馈阶段的重要信息,YOLOv9提出可编程梯度信息(programmable gradient information,PGI),包含三部分:主分支用于推理,辅助可逆分支解决信息瓶颈问题,多级辅助信息避免传统多路径特征融合深度监督过程造成的语义损失。YOLOv9的PGI可看作v7中辅助头的改进,在不增加额外推理成本的前提下,可以进一步提高模型的特征提取能力,增加PGI模块可以为模型带来0.6个百分点的AP值提升[15]。

2.2 YOLO系列锚框设置对比

锚框(anchor boxes)是指在训练过程中,通过先验信息选择固定框的尺寸和比例,以适应训练集中的目标类别和形状。在YOLOv1[9]中,每个网格会预测固定数量的锚框,锚框的尺寸和比例与训练集中的目标类别和形状有关。通过与真实标记框进行匹配,YOLO可以确定每个锚框负责预测哪个目标。锚框可以帮助模型更好地对不同大小和形状的目标进行预测。由于锚框是根据训练集中目标的分布进行选择的,可以在模型中提供多个尺寸和比例的预选框,使得模型能够更好地适应各种目标,提高模型的预测能力和泛化能力。其次,通过将每个锚框与真实标记框进行匹配,模型可以更好地学习目标的位置和形状,从而减少对背景的误判,提高预测的准确性。

严格来讲,YOLOv1[9]是没有锚框的。v1版本的Bounding box起到了锚框的作用,YOLOv1会将输入图片分成7×7个单元格,每个单元格预测输出2个Bounding box,每个Bounding box包含5个值(4个坐标+1个置信度)。这种设计方式使得模型在预测时能够更加关注目标的位置和大小。但是,由于没有考虑到不同尺寸和比例的目标,在某些场景下可能会出现预测偏差。为了弥补当同一类物体出现不常

见的长宽比时，v1版本泛化能力偏弱的问题，YOLO-v2[13]将输入图片分成13×13个单元格，每个单元格预测输出5个锚框，并使用*k*-means聚类算法替代手工设计获得锚框的先验尺寸。这种设计方式可以使模型更好地适应不同尺寸和比例的目标，提高预测的准确性。YOLOv3[10]、YOLOv4[14]、YOLOv5、YOLOv7[12]将输入图片分成13×13个单元格，每个单元格预测输出9个锚框，分别来自3种不同的尺度、3种不同的宽高比。这种设计方式可以使模型适应不同尺寸和比例的目标，同时提高模型的泛化能力。为了使模型更好地适应不同形状的目标，提高模型的泛化能力，YOLOv6、YOLOv8、YOLOv9[15]采用了无锚框设计，通过消除锚框的使用，可以减少模型的复杂性，并提高预测速度。无锚框设计可以在MS COCO17数据集上，使模型AP值提高1.3个百分点的同时，将算法推理速度提高51%[11]。然而，无锚框设计可能会影响模型对特定形状目标的预测准确性，需要根据具体应用场景进行选择。

2.3 YOLO系列损失函数设计

损失函数是YOLO算法的重要组成部分，通常由三部分组成，分别为预测框坐标损失、置信度损失以及类别损失。

2.3.1 YOLOv1损失函数

YOLOv1[9]的损失函数涵盖了预测框损失、目标置信度损失和目标类别损失，并采用均方误差进行度量，其定义如式(1)所示：

$$L_{\text{all}} = L_{\text{box}} + L_{\text{obj}} + L_{\text{class}} \tag{1}$$

其中，L_{box} 表示模型预测框损失，L_{obj} 表示模型置信度损失，L_{class} 表示模型分类损失。

模型预测框损失 L_{box} 如式(2)所示：

$$\begin{aligned} L_{\text{box}} = {} & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \\ & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{obj}} [(\sqrt{w}_i - \sqrt{\hat{w}}_i)^2 + (\sqrt{h}_i - \sqrt{\hat{h}}_i)^2] \end{aligned} \tag{2}$$

其中，λ_{coord} 表示正样本损失权重系数，I_{ij}^{obj} 表示挑选出负责检测该目标的预测框，$\hat{x}_i$、$\hat{y}_i$、$\hat{w}_i$、$\hat{h}_i$ 分别表示负责检测物体预测框中心点的横坐标、纵坐标、宽度、高度，x_i、y_i、w_i、h_i 分别表示真实物体标注框中心点的横坐标、纵坐标、宽度、高度。

模型置信度损失 L_{obj} 如式(3)所示：

$$L_{\text{obj}} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} I_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \tag{3}$$

其中，I_{ij}^{obj} 表示挑选出负责检测该目标的预测框，I_{ij}^{noobj} 表示挑选出不负责检测该目标的预测框，λ_{noobj} 表示惩罚项系数，用来惩罚不负责检测目标的预测框，C_i 表示预测框与真实框的IoU，$\hat{C}_i$ 表示模型正向推断出的维向量中预测框的置信度得分。

模型分类损失 L_{class} 如式(4)所示：

$$L_{\text{class}} = \sum_{i=0}^{S^2} I_i^{\text{obj}} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \tag{4}$$

其中，$\hat{p}_i(c)$ 表示预测单元格为该类别物体的概率，$p_i(c)$ 表示真实概率。

YOLOv1损失函数仍存在局限性，例如，在训练的早期，许多单元格不存在目标物体，将会导致这些单元格内Bounding box的置信度过早置0，从而导致早期的训练发散。

2.3.2 YOLOv2损失函数

相较于v1，YOLOv2增加了损失函数，使得负责预测物体的锚框能在训练早期，学习所有锚框的形状特征，从而加快收敛速度。YOLOv2整体损失函数共有5项构成，如式(5)所示：

$$\begin{aligned} loss_t = {} & \sum_{i=0}^{W} \sum_{j=0}^{H} \sum_{k=0}^{A} I_{Max\text{-}IoU < Thresh} \lambda_{\text{noobj}} \times (-b_{ijk}^{o})^2 + \\ & I_{t < 12\,800} \lambda_{\text{prior}} \times \sum_{r \in (x,y,w,h)} (prior_k^r - b_{ijk}^r)^2 + \\ & I_k^{\text{truth}} (\lambda_{\text{coord}} \times \sum_{r \in (x,y,w,h)} (truth^r - b_{ijk}^r)^2 + \\ & \lambda_{\text{obj}} \times (IoU_{\text{truth}}^k - b_{ijk}^o)^2 + \\ & \lambda_{\text{class}} \times \left(\sum_{c=1}^{c} (truth^c - b_{ijk}^c)^2 \right) \Bigg) \end{aligned} \tag{5}$$

其中，第1项是让不负责预测物体的锚框损失越小越好，*Thresh* 表示置信度阈值，$I_{Max\text{-}IoU < Thresh}$ 用来找到置信度低的锚框，λ_{noobj} 是惩罚项系数，$(-b_{ijk}^{o})^2$ 表示预测框的置信度。第2项衡量训练早期定位误差，$I_{t < 12\,800}$ 是选出前12 800次迭代，也就是训练的早期。第3项衡量真实框和负责预测物体锚框的定位误差，I_k^{truth} 用来找到负责预测物体的锚框。第4项计算置信度误差，IoU_{truth}^k 表示负责预测物体锚框和真实值的交并比，b_{ijk}^o 表示预测物体锚框的置信度。第5项衡量真实框和负责预测物体锚框的分类误差，$truth^c$ 表示真实框的类别，b_{ijk}^c 表示预测物体锚框的类别。

2.3.3 YOLOv3损失函数

在YOLOv2中，Softmax层用于单标签分类，即每个目标仅属一个类别。但在复杂场景中，目标可

能属多个类别，如汽车既属于车辆又属于交通工具。为实现多标签分类，YOLOv3[10]采用Sigmoid函数，将输出限制在0到1，若某特征图输出值大于阈值，则目标属于该类。由于使用Sigmoid函数，损失函数引入了二元交叉熵损失，最终的损失函数与式(1)一致，也由边界框损失、目标置信度损失和目标类别损失三项组成。

式(6)表示边界框损失 L_{box}：

$$L_{\text{box}}=\lambda_{\text{coord}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}I_{ij}^{\text{obj}}(2-w_i^j\times h_i^j)\times [(w_i^j\times\hat{w}_i^j)^2+(h_i^j\times\hat{h}_i^j)^2] \tag{6}$$

其中，示性函数 I_{ij}^{obj} 表示单元格的锚框是否负责预测物体，该式用来衡量锚框的定位误差。

YOLOv3引入了交叉熵损失来计算置信度误差，式(7)表示目标置信度损失 L_{obj}：

$$L_{\text{obj}}=\sum_{i=0}^{S^2}\sum_{j=0}^{B}I_{ij}^{\text{obj}}[\hat{C}_i^j\ln C_i^j+(1-\hat{C}_i^j)\ln(1-C_i^j)]-\lambda_{\text{noobj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}I_{ij}^{\text{noobj}}[\hat{C}_i^j\ln C_i^j+(1-\hat{C}_i^j)\ln(1-C_i^j)] \tag{7}$$

其中，$\hat{C}_i^j$ 表示模型拟合值，C_i^j 表示真实值，I_{ij}^{noobj} 用来挑选出不负责检测该目标的预测框。

目标类别损失 L_{class} 如式(8)所示，同样采用交叉熵，只有当单网格的某个锚点负责预测真实目标物体时，该锚点才被用来计算分类误差。

$$L_{\text{class}}=\sum_{i=0}^{S^2}I_{ij}^{\text{obj}}\sum_{c\in class}[\hat{P}_i^j\ln P_i^j+(1-\hat{P}_i^j)\ln(1-P_i^j)] \tag{8}$$

其中，$\hat{P}_i^j$ 表示单元格预测为该类别物体的概率，P_i^j 表示真实概率。

2.3.4 YOLOv4损失函数

YOLOv4[14]定位误差损失函数采用CIoU(complete intersection over union)，CIoU在IoU的基础上进行了补充，考虑了重叠面积、中心点距离和长宽比，从而能够更好地度量预测框和真实框重合程度。式(9)表示IoU损失函数：

$$L_{\text{IoU}}=1-IoU=1-\frac{|B\cap B^{\text{gt}}|}{|B\cup B^{\text{gt}}|} \tag{9}$$

其中，B 为预测框，B^{gt} 为真实框。

CIoU如式(10)所示：

$$L_{\text{CIoU}}=L_{\text{IoU}}+\frac{\rho^2(b,b^{\text{gt}})}{c^2}+\alpha v \tag{10}$$

$$v=\frac{4}{\pi^2}\left(\arctan\frac{w^{\text{gt}}}{h^{\text{gt}}}-\arctan\frac{w}{h}\right)^2 \tag{11}$$

其中，α 为参数，c 表示同时包含预测框和真实框的最小闭包区域的对角线距离，v 衡量宽高比的一致性。

YOLOv4整体损失函数共有4项构成，如式(12)所示：

$$loss=\lambda_{\text{coord}}\sum_{i=0}^{K\times K}\sum_{j=0}^{M}I_{ij}^{\text{obj}}(2-w_i\times h_i)(1-L_{\text{CIoU}})-\sum_{i=0}^{K\times K}\sum_{j=0}^{M}I_{ij}^{\text{obj}}[\hat{C}_i\ln C_i+(1-\hat{C}_i)\ln(1-C_i)]-\lambda_{\text{noobj}}\sum_{i=0}^{K\times K}\sum_{j=0}^{M}I_{ij}^{\text{noobj}}[\hat{C}_i\ln C_i+(1-\hat{C}_i)\ln(1-C_i)]-\sum_{i=0}^{K\times K}\sum_{j=0}^{M}I_{ij}^{\text{obj}}\sum_{c\in classes}[\hat{p}_i(c)\ln p_i(c)+(1-\hat{p}_i(c))\ln(1-p_i(c))] \tag{12}$$

其中，第1项是定位误差损失，衡量负责预测物体的锚框与真实框的重叠程度。第2项是正样本置信度误差损失，采用交叉熵损失衡量负责预测物体锚框的置信度。第3项是负样本置信度误差，采用交叉熵损失衡量不负责预测物体锚框的置信度。第4项是正样本分类损失，即衡量负责预测物体锚框的分类误差。

3 YOLO算法数据集及评价指标

3.1 典型数据集

数据集在模型训练与评估中扮演着基准的重要角色。利用多样性且广泛涵盖的图像数据集进行训练有助于模型在图像识别、分类和分割等任务上获得更为强大的能力。在目标检测领域，对图像进行标注，包括目标位置和类别信息，有助于模型准确地识别和定位图像中的各种物体。本节旨在概述目标检测任务中一些典型的数据集，包括PASCAL VOC数据集、ImageNet数据集、Google Open Images数据集、MS COCO数据集以及DOTA数据集，如表2所示。这些数据集在不同场景下，提供了丰富的标注信息，为YOLO算法的训练和评估提供了重要的基准和实践验证。

3.1.1 PASCAL VOC数据集

PASCAL VOC[42]代表了国际计算机视觉领域的一项挑战赛。目前，PASCAL VOC[42]在目标检测领域应用最广泛的版本为PASCAL VOC07和PASCAL VOC12，两者均包含20个类别。PASCAL VOC07包含9 963个图像，涵盖超过12 000个标记对象；而PASCAL VOC12则涵盖22 531个图像和超过27 000个标记对象。该数据集是YOLO算法常用标准测试数据集之

表2 YOLO算法常用数据集

Table 2 Datasets commonly used by YOLO algorithms

数据集	类别	图像数量	图像尺寸/像素	起始年份	特点
PASCAL VOC12	20	22 531	470×380	2005	数据集规模较小，部分类别相似度较高
ImageNet	20 000	14 000 000	500×400	2009	数据集规模大
Google Open Images	600	1 900 000	尺寸不一致	2017	数据集规模较大，图像背景较为复杂
MS COCO17	91	162 888	640×480	2014	数据集适中，部分目标存在遮挡
DOTA	15	188 282	4 000×4 000	2017	数据集图像尺寸大，多应用于航空遥感领域

一，虽然该数据集相对其他几种数据集规模和复杂度较小，但由于其高质量的标注信息及多种常见目标类别，仍能较为充分地检验模型性能。

3.1.2 ImageNet数据集

ImageNet[43]数据集在计算机视觉领域得到了广泛的应用，几乎成为深度学习图像算法性能评估的“标准”数据集。ImageNet数据集拥有超过14 000 000幅图像，覆盖了超过20 000个类别，其中有超过百万张图片被准确标注了类别和物体位置信息。然而ImageNet数据集数据量庞大，图像类别多，并不适用于目标检测训练，主要用于分类任务。但ImageNet数据集的一些子集通常可以用于目标检测任务，如ILSVRC（ImageNet large scale visual recognition challenge）。

3.1.3 Google Open Images数据集

Google Open Images[44]数据集由谷歌团队发布。最新版本的Open Images V4涵盖了约1 900 000张图像，包含了600个不同的物体类别，标注了大约15 400 000个物体边界框，是当前拥有物体位置标注信息的最大规模数据集。这些边界框大部分由经过专业训练的注释人员手工绘制，以确保其准确性和一致性。该数据集涵盖了各种各样的图像类别，如人、动物、物品、车辆，可以使模型学习到更加广泛的图像特征。此外，该数据集通常包含复杂场景中多个对象的视觉信息，可以使研究者训练更具鲁棒性的模型。

3.1.4 MS COCO数据集

MS COCO[45]数据集由微软团队发布，收集了大量涵盖日常场景中的物体图像。该数据集广泛应用于机器视觉领域的诸多核心任务，包括目标检测、实例分割、关键点识别和图像注释。相较于ImageNet，COCO数据集更加倾向于收集目标物体与其周围场景共同出现的图像。这类图像具有更强的视觉语义，更符合图像理解任务的需求。其中，MS COCO17作为目标检测领域中常用的数据集，涵盖了91个物体类别（但在检测任务中使用了80类）。该数据集包含117 266个训练图像、4 952个验证图像和40 670个测试图像。标记的目标数量超过89 000个。相较于PASCAL VOC，COCO数据集包含图像类别更多，且数据集图像背景复杂，内容更加丰富。表3列举了YOLO各版本算法在相关数据集上的测试结果。从YOLOv4版本开始，算法速度和精度较之前版本有很

表3 各版本YOLO算法性能对比

Table 3 Performance comparison of different YOLO algorithms

算法模型	图像尺寸/像素	mAP/%	AP/%	FPS	数据集	GFLOPs
YOLOv1[9]	448	57.9	—	—	PASCAL VOC12	—
YOLOv2[13]	544	73.4	—	—	PASCAL VOC12	—
YOLOv3[10]	416	—	31.0	35	MS COCO17	—
YOLOv4[14]	416	—	41.2	38	MS COCO17	—
YOLOv5-M[11]	640	—	45.4	182 (batch size=1)	MS COCO17	49.0
YOLOv6-M[11]	640	—	49.5	179 (batch size=1)	MS COCO17	82.2
YOLOv7[12]	640	—	51.2	161 (batch size=1)	MS COCO17	104.7
YOLOv8-M[15]	640	—	50.2	—	MS COCO17	78.9
YOLOv9-M[15]	640	—	51.4	—	MS COCO17	76.3
DAMO YOLO-M[46]	640	—	49.2	233	MS COCO17	61.8
Gold YOLO-M[47]	640	—	49.8	152 (batch size=1)	MS COCO17	87.5

大提升。

3.1.5 DOTA数据集

DOTA[48]包含2 806张航空图像，是遥感航空图像检测领域广泛使用的数据集，通常用于YOLO算法针对无人机领域的目标识别检测任务。这些图像包含了15个类别，合计188 282个实例，其中包括14个主要类别。与传统数据集不同的是，航空图像在尺度上具有更大的变化性，给遥感图像检测算法的训练和评估带来了挑战，因其特殊的尺度变化性，需要更为灵活和复杂的算法模型，需要对YOLO算法进行合理改进，如小目标检测、多尺度融合等，以应对不同尺度下的目标检测任务。

3.1.6 数据集的选择与构建

在目标检测任务数据集的选择上，明确具体应用场景需求至关重要。处理不同的目标检测数据集对于YOLO算法不同场景的检测性能有着显著的影响。通常情况下，检测日常生活中常见的物体，如人、车辆、动物等，可以选择现有的数据集，如PASCAL VOC12、MS COCO17数据集。

PASCAL VOC数据集的场景简单、数据量较小，使用该数据集训练和评估模型更为迅捷。然而，场景覆盖范围相对单一，主要聚焦于一些常见物体和环境，这在一定程度上限制了其对于复杂多变场景的适用性。因此，PASCAL数据集更适合那些追求检测速度的轻量化版本YOLO算法。相较而言，MS COCO数据集具有丰富的场景，包含了更多种类的物体，使得COCO数据集在提升模型泛化能力方面更具优势。然而，其庞大的数据量导致训练和评估过程相对较慢。因此，MS COCO数据集适合追求检测速度和检测精度相平衡的YOLO算法。DOTA数据集是专门为航空影像中的目标检测任务设计的，具有独特的航拍视角和复杂场景，如城镇、山区、平原、河流。因此，DOTA数据集适用于飞行设备目标检测任务的训练与评估，如无人机目标检测任务。

通常情况下，具体的目标检测任务，如果园苹果检测、稻穗生长情况检测，需构建专门的数据集，数据集质量好坏直接影响到YOLO算法训练模型实际应用中的性能。在构建数据集的过程中，需广泛收集涵盖目标类别的图像数据，以确保数据的多样性。数据收集可以通过多种途径获取，如采用公开的数据集资源，通过自行拍摄整理获取特定场景的目标图像。收集到图像后，需对图像进行数据清洗，剔除重复模糊或与目标类别不相关的图像，以确保数据集的有效性。最后，需对图像数据集进行标注工作，精确绘制目标边界框，并标注正确的类别标签。

3.2 评价指标

目标检测领域建立了一套完整的评价体系[49]，如表4所示，用于全面衡量算法的性能，包括每秒帧率（frames per second，FPS）、精确率（precision，P）、召回率（recall，R）、交并比（IoU）、平均精度值（AP）、平均精度均值（mAP）。平均精度均值作为一个综合性指标，能够综合反映精确率、召回率、交并比以及平均精度值这四个度量指标的性能。在YOLO系列算法中，通常采用FPS值和AP值作为评价模型性能的核心指标，其中FPS值度量算法速度，AP值度量算法精度。

表4 YOLO算法评价度量

Table 4 Evaluation metrics of YOLO algorithms

评价指标	指标作用
FPS	评价YOLO算法模型推理速度
P	评价YOLO算法预测的精确率
R	评价YOLO算法预测的漏检率
IoU	度量预测框和真实框的重叠程度
AP	评价YOLO算法预测单一类别的平均精确率
mAP	评价YOLO算法预测所有类别的平均精确率

FPS是用于评价YOLO算法处理速度的指标，表示每秒处理的图像帧数。对于实时性要求高的场景，如自动驾驶、道路异常检测等领域，需要快速响应突发情况。更侧重于FPS这一性能指标，高FPS值意味着YOLO算法模型能够更快速地检测目标物体。

对于二分类问题，混淆矩阵是一个2×2矩阵，如表5所示，其中行代表预测值，列代表真实值。真阳性（true positive，TP）指YOLO算法正确分类的正样本个数，真阴性（true negative，TN）指YOLO算法正确分类的负样本个数，假阳性（false positive，FP）指YOLO算法错误地标记为正样本的负样本个数，假阴性（false negative，FN）指YOLO算法错误地标记为负样本的正样本个数。

表5 混淆矩阵

Table 5 Confusion matrix

混淆矩阵		真实值	
		True	False
预测值	True	TP	FP
	False	FN	TN

Precision如式(13)所示，指被预测为正例的样本中，准确预测出的真实正例的占比。Recall如式(14)所示，指被准确预测出的真实正例占实际正例的占比。这两个指标共同衡量了算法精度。对于高精度要求的场景，如医学影像分析、工业伤损检测等领域，即使检测速度稍慢，也要尽可能保证算法准确识别目标。故在这一场景下，更加侧重于算法精度。

$$P=\frac{TP}{TP+FP} \tag{13}$$

$$R=\frac{TP}{TP+FN} \tag{14}$$

IoU如式(15)所示，IoU是度量预测边界框和真实边界框之间重叠的度量。其计算方法为两个方框的相交面积与合并面积之比。高IoU值意味着YOLO算法模型的预测目标位置越接近真实目标位置，预测框更加精确。

$$IoU=\frac{|B\cap B^{gt}|}{|B\cup B^{gt}|} \tag{15}$$

AP值如式(16)所示，AP是物体检测中广泛使用的度量指标，用来衡量模型在不同精度水平下检测物体的精度。AP计算不同阈值下的PR曲线下方面积。

$$AP=\sum_{i=1}^{n}(R_i-R_{i-1})P_i \tag{16}$$

mAP值如式(17)所示，mAP是在不同精度水平上计算的AP的平均值。它是用来测量模型跨所有类的总体性能。

$$mAP=\frac{1}{N}\sum_{i=1}^{N}AP_i \tag{17}$$

AP和mAP作为综合性评价指标，能够全面反映算法在单一类别和所有类别目标上的算法精度。高AP或mAP值意味着YOLO算法模型检测到的目标物体置信度更高。

4 YOLO算法应用领域

YOLO算法广泛应用于各个领域的目标检测任务，特别是在工业、交通、农业、生物以及遥感等领域，如图4所示。表6分析了YOLO算法在各应用领域常见改进方法，并且对于各种方法的优势和局限性作出探讨。相对于传统方法，YOLO算法取得了显著的效果。

4.1 工业检测

可变形卷积网络(deformable convolution network，DCN)是一种通过为卷积操作添加偏置，使其具有形变能力的网络。在钢表面缺陷检测这一行业，卢俊哲等人[23]提出了轻量级DCN-YOLO模型。引入DCN模块提高了模型对不同尺寸和形状缺陷的灵敏度。并且引入深度可分离卷积[58]DSConv有效减少计算量和参数量。最后引入通道注意力机制提升网络对重要特征的感知能力。

贾晓芬等人[21]设计了轻量化网络DE-YOLO，以适用于移动终端设备，从而实现输电线路高精度和速度巡检。首先，提出NewC3模块，降低网络参数，同时强化网络提取有效信息的能力。最后借助通道数成倍增长策略和通道注意力机制SE(squeeze-and-excitation)设计了轻量化模块DC-SE，减少复杂背景对目标的干扰，增强浅层网络提取能力。李想等人[22]提出TCS-YOLO(Transformer-CBAM-SIoU YOLO)模型，该模型在骨干网络添加基于Transformer[59]架构

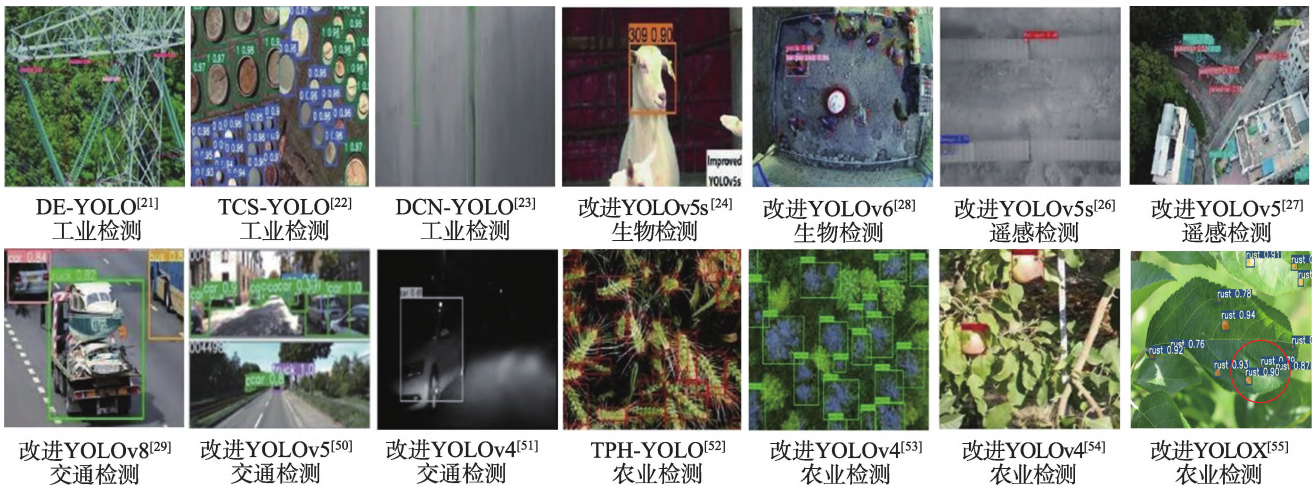

图4 YOLO算法在各个领域的应用

Fig.4 Application of YOLO algorithm in various domains

表6 不同应用领域中YOLO算法改进对比分析

Table 6 Comparative analysis of YOLO algorithm improvement in different application fields

领域	改进方法	优势	局限性	各领域算法要求
工业	数据预处理，如旋转、缩放、平移	增加训练数据多样性，提升模型复杂场景下泛化能力	扩充数据集质量一般，在特殊场景精度较低	工业应用场景复杂，干扰因素多，该领域检测更关注算法精度
	借鉴最新网络结构[29,56]	网络结构可以强化算法特征提取能力	添加新模块通常会导致算法计算量增大	
	添加注意力机制[21-22]	可以增强特征提取网络对重要图像特征的感知能力，提升检测精度	加深网络层数，导致参数量增多	
	改进空间金字塔[56]	可以提高模型多尺度融合能力，提升精度	不重要的特征信息融入特征图，产生计算量	
交通	对主干网络进行重构[29]以轻量化模型	可以减少模型参数，提高检测速度	轻量化模型的同时通常会降低算法精度，研究者需要对二者进行权衡	交通领域的检测通常要求实时性，更关注算法检测速度
遥感	添加注意力机制[21-22]	可以加强特征提取网络对小目标的关注度，从而提升算法对小目标的检测精度	网络层数加深，参数量增多会降低检测速度	遥感图像分辨率高，图像尺寸大，背景通常较为复杂，对于算法精度和速度均有较高的要求，其中遥感领域的检测更关注小目标检测的精度
	特征金字塔改进[57]	可以更好地融合多尺度语义信息，从而提升算法对于小目标的检测精度	存在融入的不重要特征信息，降低检测速度	
农业	重构主干网络	减少不必要层级连接可以提升处理速度	轻量化模型的同时通常会降低算法精度	农业检测领域通常将算法模型部署在AI边缘计算设备，需要在尽可能保证算法精度的情况下，降低复杂度，提升检测速度
	引入其他卷积方式	可以加强特征提取能力和减少计算量，从而有效提升算法性能	部分卷积方式会大幅度减少卷积参数，从而导致算法精度的降低	
生物	添加注意力机制	可以提升特征提取网络对不同尺度图像的关注度，从而提升模型泛化性能	网络层数的加深会增加计算复杂度，降低算法检测速度	生物行为状态十分丰富，对于YOLO算法的泛化性能要求高
	Neck层融入其他改进[28]	可以提高算法模型范围野，以提升检测精度	增加其他模块通常会增加计算复杂度，降低算法检测速度	

的C3TR层。此外，引入卷积块注意力模块（convolutional block attention module，CBAM）[60]，提升YOLO算法对重要特征的捕获能力。最后使用SIoU Loss激活函数，解决CIoU Loss大损失值点周围梯度趋于平缓的问题。

总体而言，这些算法的提出与实践在很大程度上推动了工业发展的智能化进程。然而，这些算法在解决特定检测任务时，其性能与效果存在一定的差异。因此，针对特定的工业检测任务，需要根据实际需求选择合适的解决方案。在某些场景下，快速高效的检测是关键，而在另一些场景下，提高模型的检测精度则更为重要。工业应用场景复杂、干扰因素多，通常会进行数据预处理，如旋转、缩放、平移等操作来增加训练数据的多样性，提升模型复杂场景下的泛化能力。一般情况下，工业领域的检测更关注算法精度。为了提高精度，通常的改进方法有：引入更强大的网络以强化算法特征提取能力，如使用可逆连接的多列网络（reversible column networks，RevColNet），对主干网络进行重构以提升算法检测性能[29,56]，添加注意力机制[21-22]增强网络对重要特征的感知能力从而提升算法检测精度，改进网络空间金字塔结构[56]以提高模型多尺度融合能力从而提升算法检测性能。为了进一步推动工业检测算法的发展和应用，未来的研究需要在数据采集与处理、特征提取网络的设计与改进、模型训练参数的优化等诸多方面进行深入研究。同时，需要不断探索和创新，以期在工业检测领域取得更为出色的成果。

4.2 交通检测

牛为华等人[50]在道路小目标检测行业，改进了YOLOv5算法。首先，增加小目标检测层，利用浅层特征层中包含的丰富的语义及位置信息精确定位识别小目标，模型设计ConvFocus模块，将Focus模块和

卷积操作残差连接，以优化小目标漏检问题。最后，采用双线性插值上采样操作，引入CBAM[60]注意力机制模块，聚焦图像局部信息，减少小目标特征丢失，增强小语义及位置信息。

郭克友等人[51]提出了基于YOLOv4[14]的Dim Env-YOLO车辆目标检测算法，解决夜间场景车辆识别干扰因素较多的问题，利用MobileNetV3网络替换主干网络，以减少参数量。使用图像暗光增强方法，提高车辆目标在昏暗环境中的可识别性。引入注意力机制的同时，利用深度可分离卷积来降低网络计算量。张利丰等人[29]提出了RBT-YOLO算法，使用可逆连接的多列网络（RevColNet）对主干网络进行重构，轻量化模型。此外，在CBiFPN（concat bi-directional feature pyramid network）基础上增加卷积操作，删改连接层数。最后引入注意力机制和SoftNMS，提升模型特征提取能力和算法精度。

交通检测是目标检测的一个重要应用领域，在众多细分领域都有所应用，如监测道路异常及违法事件、自动驾驶领域等。在交通检测中，实时性尤为重要，因此交通领域的检测更关注算法检测速度。为了提升速度，通常的改进方法有：对主干网络进行重构、剪枝处理，轻量化模型以提升模型处理速度[29]。为了推动交通检测算法的发展和应用，未来需要在模型压缩与优化等方面进行深入研究。

4.3 遥感检测

谢椿辉等人[27]提出了改进YOLOv5的算法Drone-YOLO，通过增加检测头，提高模型在多尺度下的检测能力，设计基于多尺度通道注意力机制的特征融合模块，并且设计多层次信息聚合的特征金字塔网络结构，实现了跨层次信息的融合，提高了小目标关注度。最后使用Alpha-IoU优化损失函数解决了当两个框没有交集面积时IoU值为0的问题。孙建波等人[26]提出了YOLO v5s故障检测算法，以提升无人机在光伏组件巡检任务中的检测速度和精度。该算法采用自适应调节置信度损失平衡系数以提升模型训练效果。随后，在每个检测层前分别添加InRe特征增强模块，丰富特征表达，以增强目标特征提取能力。苏志威等人[25]对YOLOv8进行改进，用来提高航空铝合金焊接缺陷数字射线成像自动检测效率和准确度。首先，采用Retinex图像增强算法对数据集进行数据增强，提高模型泛化能力。GhostBottleneck模块设计思想来源于GhostNet[61]，即特征相似的特征图仅做线性变化，改进算法使用GhostBottleneck模块替换Bottleneck模块，将原本大量非线性运算改为简单的线性运算，从而轻量化模型。最后引入空间注意力机制，增强空间位置关联，提高模型检测精度。

总体而言，改进的遥感检测算法显著提高了遥感图像的检测准确率。然而，由于遥感图像分辨率高，图像尺寸大，处理这些图像需要大量的计算资源。此外，由于遥感图像的背景通常较为复杂，遥感领域对于算法精度和速度均有较高的要求。但是遥感领域的检测更关注小目标检测的精度，通常的改进方法是添加注意力机制[21-22]加强特征提取网络对小目标的关注度，从而提升算法对于小目标的检测精度。特征金字塔网络结构可以实现特征的多尺度融合，通过进一步改进[57]该模块，如多层次聚合的特征金字塔[27]，可以提升算法对于小目标的检测精度。总而言之，构建优秀的网络模型是推动遥感检测领域发展的重要方向。

4.4 农业检测

鲍文霞等人[52]研究设计了基于TPH-YOLO的麦穗检测模型，旨在提高无人机图像麦穗计数的精度。首先，添加坐标注意力机制（coordinate attention, CA），强化麦穗信息提取能力。其次，Transformer[59]预测头（Transformer prediction heads，TPH）具有多头注意力机制。最后，采用迁移学习策略，提高模型泛化能力。林文树等人[53]改进了YOLOv4，提出了一种基于受灾树木实时检测模型，通过重构CSPDarknet53网络，在CSPNet中加入SENet增加感受野信息，提升了无人机搭载的边缘计算设备速度。基于RGB与Depth双输入图像，郝鹏飞等人[54]提出了YOLO-RD-Apple果园异源图像遮挡果实检测模型。该模型使用MobileNetV2-Lite分别作为RGB和Depth图像的特征提取器，保证特征提取能力的同时降低网络的计算量。同时将CSPNet[35]与深度可分离卷积结合，并引入SE注意力模块，提出全新的SE-DWCSP3模块。此外，对PANet结构进行改进，提升网络对于残缺苹果目标的特征提取能力。最后，引入Soft NMS算法，以减少模型对密集目标错误抑制现象，降低被遮挡苹果漏检率。

尽管YOLO算法在农业检测领域取得了一定的进展，但农业场景的复杂性和独特性使得该算法在实际应用中仍面临一些挑战。由于生物的变异性，导致不同作物之间存在明显的差异性，这使得构建数据集的难度加剧，从而导致模型的识别精度下降。此外，天气和光照条件的变化也会对检测效果

产生不利影响,例如暴雨天、夜晚等情况下,图像的清晰度和对比度可能会受到影响,导致模型进行推理预测时的检测精度下降。此外,在农业检测领域,通常将算法模型部署在AI边缘计算设备,如搭载英伟达芯片的无人机、机器人。AI边缘计算设备在本地实时离线处理数据图片,具有低延迟、隐私性好等优点。但是,受限于自身体积、芯片功耗等影响因素,AI边缘设备的计算能力和存储容量有限,无法处理大规模数据或复杂的计算。故对于部署在AI边缘设备的算法性能要求较高,需要尽可能降低算法复杂度,减轻模型体积,降低处理器功耗,并且还需要同时保证算法精度和算法速度较高。通常的改进方法是减少不必要的层级连接轻量化模型以提升模型处理速度[55],引入其他卷积方式,如深度可分离的卷积可以有效提升算法效率。总之,针对农业场景的特殊需求,需要不断探索和研究新的算法和技术,以进一步提高农业检测的精度和速度。这有助于实现智能化的农业管理和监测,提高农业生产效率。

4.5 生物检测

宁纪锋等人[24]提出改进YOLOv5s的奶山羊个体识别方法,通过迁移学习,以及引入基于相似性的注意力模块(similarity-based attention module,SimAM)和CARAFE(content-aware reassembly of features)的上采样模块,提升模型泛化能力及算法精度。杨断利等人[28]提出改进YOLOv6-tiny[11]模型,以提高蛋鸡啄羽异常行为识别精度,通过引入DenseBlock结构和融入CSP结构的SPP模块,以增强特征提取能力并扩大感受野,从而提高检测精度。

在生物检测领域,YOLO算法的应用可以看作一种革命性的变革。该应用不仅在学术研究中具有重要意义,更在实践应用中展示了其巨大的价值。YOLO算法在动物行为学等方面展示了其广泛的应用前景。通过对动物行为的监测和分析,揭示动物的行为习性和生态适应机制。在这一领域中,生物的行为状态十分丰富。不同行为状态对应着不同姿态,生物姿态的多样性使得数据集的采集变得复杂且庞大,这对于YOLO算法的泛化性能提出了更高要求。通常的改进方法是对网络的Neck层进行改进。如Neck层添加注意力机制[24]提升算法识别精度,Neck层融入其他改进结构[28]以提高算法范围野,从而提升模型检测精度。尽管YOLO算法在生物检测领域的应用已经取得了一定的进展,但仍然存在许多挑战和问题需要进一步研究和解决。例如,如何进一步提高模型的泛化能力,使其能够适应不同环境下的识别,识别到不同的生物体形态和行为特征。

5 难点与展望

YOLO算法发展至今,提出了众多版本,本章对YOLO及其他较新目标检测算法进行综合评价,如表7所示。讨论YOLO算法发展面临的问题,并对YOLO算法未来的发展作出展望。

5.1 YOLO发展面临的问题

YOLO算法在目标检测领域表现出色,采用端到端的架构简化了网络结构,减少了不同模块之间的复杂连接,将目标检测问题转化为回归和分类问题,从而在实时性和检测速度上优于两阶段算法。具体而言,YOLO算法的骨干网络有效地提取了整张图片的特征,并且通过聚类算法得到的锚框能够快速准确地捕获目标物体。YOLO算法的颈部网络采用相应算法有效地提取多尺度特征,从而实现对于大、中、小三种不同尺度物体的识别。此外,YOLO算法的不同通道之间汇聚了不同类别物体的特征模型,使得该算法能够在使用较小模型的基础上预测复杂情境下的多种类别的物体。然而,与其他优秀的算

表7 目标检测算法分析

Table 7 Analysis of target detection algorithms

算法	机制	优势	局限性	适用场景
YOLO	单阶段检测	端到端架构降低了算法复杂度,检测速度优于两阶段算法	检测精度略逊于两阶段算法,小目标的精准定位仍然不足,存在正负样本均衡问题	工业、交通、遥感、农业、生物
ThunderNet[62]	双阶段检测	超轻量级双阶段检测算法,在保证较高检测精度的同时提升了检测速度	与新版YOLO算法相比,其检测速度仍稍显不足	工业、遥感、农业、生物
DETR[63]	单阶段检测	无锚框设计减少了算法复杂度,并且Transformer架构,能够充分利用输入图像的全局上下文信息,从而提高目标检测的准确性	相较于YOLO算法,DETR训练时间更久,并且难以处理高分辨率的特征,从而导致小目标检测精度较差	工业、交通、农业、生物

法一样，YOLO算法也存在一些不足之处。首先，由于感受野较大，会影响小目标的定位精度。如遥感领域中，一张包含车辆和行人高分辨率照片，YOLO算法容易把较小的行人识别为背景。其次，YOLO算法正负样本不均衡。例如，在目标检测任务中，一张只有单个目标物体的图片，由于图片中目标物体数量有限，只有少数锚框预测为目标物体，即正样本，大部分锚框预测为负样本。正负样本不均衡会影响模型训练效果。最后，YOLO算法的训练需要大量的图片以确保模型的泛化能力。例如，在交通检测中，需要包含不同天气状况、不同时间段、不同交通场景的图像数据集，才能保证模型的检测精度。综上所述，YOLO算法在目标检测领域表现出色，具有许多优点，但也存在一些不足之处。在未来的研究中，可以针对这些不足之处进行改进和完善，以进一步提高YOLO算法的性能。

5.2 展望YOLO的未来

多任务学习：多任务学习是YOLO算法十分具有潜力的发展方向。语义分割可以对图片中的像素进行类别上的分类，将目标检测、语义分割等多个计算机视觉任务集成到同一模型中联合训练，共享YOLO算法骨干网络层，从而不同视觉任务可以共享底层网络的特征提取能力。并且，将不同任务的特征在颈部层进行融合，使模型找到不同视觉任务中的内在联系，学习更加全面的图像特征，从而提高模型对多个任务的综合理解能力和泛化能力。

边缘计算：随着物联网技术的高速发展，YOLO算法在边缘计算领域展现出巨大的潜力。将轻量化的YOLO算法部署到嵌入式边缘计算设备具有实时性高、安全性好的优势。例如，使用剪枝技术减少模型冗余参数，以及通过模型蒸馏技术，训练简化网络的学生YOLO模型来学习具有复杂网络的教师YOLO模型的预测能力。在资源有限的嵌入式设备上高效运行YOLO算法，可以有效降低功耗和成本。

多模态结合：随着YOLO算法的不断完善和改进，该算法有望将语言和声音等不同模态融入YOLO算法中，通过语言和声音等多元化信息来辅助目标检测。例如，设计新的骨干特征提取网络，多分支地处理图像、语言和声音等不同模态的特征信息，每个分支处理一种模态，并通过注意力等融合机制动态地融合不同模态的特征，从而使模型全面感知和理解周围环境，使YOLO算法能够应用于更多的实际场景。

虚拟、增强现实技术：YOLO算法有望与虚拟现实（virtual reality，VR）、增强现实（augmented reality，AR）技术相结合，通过目标检测和语义分割等任务，为用户提供更加精准的物体定位。例如，将YOLO算法应用于手机地图App中，可以为用户提供导航和定位的功能。通过实时分析用户周围环境中的物体，如建筑物、地标和路标等，并识别标记周围的建筑物，以增强现实的方式添加到地图应用中，可以给用户提供更准确和直观的导航体验。

综上所述，随着YOLO算法的不断迭代创新与应用发展，该算法将在未来的众多领域充分展现出其独特的应用价值和潜力。

6 结束语

本文详细剖析了YOLO系列算法在构成骨干网络的各个组成部分，以及该算法的损失函数。同时，对YOLOv1至YOLOv9各个版本[9-15]之间的改进进行了横向梳理，并列出了该算法常用的数据集以及评价指标。此外，本文还结合YOLO算法在各个细分领域的最新应用文章，深入分析了YOLO算法在各个领域的具体应用、发展前景以及所面临的挑战，并且对于YOLO算法未来的发展，凝练出四个具体的发展方向。通过深入的分析和探讨，本文为读者提供了一个全面而深入的视角来理解YOLO算法的发展历程和应用场景。

参考文献：

[1] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2004, 60(2): 91-110.

[2] LIENHART R, MAYDT J. An extended set of Haar-like features for rapid object detection[C]//Proceedings of the 2002 International Conference on Image Processing, Rochester, Sep 22-25, 2002. Piscataway: IEEE, 2002: 900-903.

[3] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection[C]//Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2005: 886-893.

[4] CRISTIANINI N, SHAWE TAYLOR J. An introduction to support vector machines and other kernel-based learning methods[M]. Cambridge: Cambridge University Press, 2000.

[5] FREUND Y, SCHAPIRE R E. Experiments with a new boosting algorithm[C]//Proceedings of the 13th International Conference on Machine Learning, Bari, Jul 3-6, 1996. San Francisco: Morgan Kaufmann, 1996: 148-156.

[6] LIAW A, WIENER M. Classification and regression by randomforest[J]. R News, 2002, 2/3: 18-22.

[7] NEUBECK A,VAN GOOL L. Efficient non-maximum suppression[C]//Proceedings of the 18th International Conference on Pattern Recognition. Washington: IEEE Computer Society, 2006: 850-855.

[8] GIRSHICK R. Fast R-CNN[C]//Proceedings of the 2015 IEEE International Conference on Computer Vision. Washington: IEEE Computer Society, 2015: 1440-1448.

[9] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified, real-time object detection[C]//Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Jun 27-30, 2016. Washington: IEEE Computer Society, 2016: 779-788.

[10] REDMON J, FARHADI A. YOLOv3: an incremental improvement[EB/OL]. [2023-12-15]. https://arxiv.org/abs/1804.02767.

[11] LI C, LI L, JIANG H, et al. YOLOv6: a single-stage object detection framework for industrial applications[EB/OL]. [2023-12-15]. https://arxiv.org/abs/2209.02976.

[12] WANG C Y, BOCHKOVSKIY A, LIAO H-Y M. YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors[C]//Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2023: 7464-7475.

[13] REDMON J, FARHADI A. YOLO9000: better, faster, stronger [C]//Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2017: 7263-7271.

[14] BOCHKOVSKIY A,WANG C Y, LIAO H Y M. YOLOv4: optimal speed and accuracy of object detection[EB/OL]. [2023-12-15]. https://arxiv.org/abs/2004.10934.

[15] WANG C, YEH I, LIAO H. YOLOv9: learning what you want to learn using programmable gradient information[EB/OL]. [2023-12-15]. https://arxiv.org/abs/2402.13616.

[16] TERVEN J, CORDOVA-ESPARZA D. A comprehensive review of YOLO: from YOLOv1 to YOLOv8 and beyond [EB/OL]. [2023-12-15]. https://arxiv.org/abs/2304.00501.

[17] DIWAN T, ANIRUDH G, TEMBHURNE J V. Object detection using YOLO: challenges, architectural successors, datasets and applications[J]. Multimedia Tools and Applications, 2023, 82(6): 9243-9275.

[18] 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.

[19] 王琳毅, 白静, 李文静, 等. YOLO系列目标检测算法研究进展[J]. 计算机工程与应用, 2023, 59(14): 15-29.
WANG L Y, BAI J, LI W J, et al. Research progress of YOLO series target detection algorithms[J]. Computer Engineering and Applications, 2023, 59(14): 15-29.

[20] SIRISHA U, PRAVEEN S P, SRINIVASU P N, et al. Statistical analysis of design aspects of various YOLO-based deep learning models for object detection[J]. International Journal of Computational Intelligence Systems, 2023, 16(1): 126.

[21] 贾晓芬, 吴雪茹, 赵佰亭. 绝缘子自爆缺陷的轻量化检测网络DE-YOLO[J]. 电子测量与仪器学报, 2023, 37(5): 28-35.
JIA X F, WU X R, ZHAO B T. Lightweight detection network for insulator self-detonation defect DE-YOLO[J]. Journal of Electronic Measurement and Instrument, 2023, 37(5): 28-35.

[22] 李想, 特日根, 仪锋, 等. 针对全球储油罐检测的TCS-YOLO模型[J]. 光学精密工程, 2023, 31(2): 246-262.
LI X, TE R G, YI F, et al. TCS-YOLO model for global oil storage tank inspection[J]. Optics and Precision Engineering, 2023, 31(2): 246-262.

[23] 卢俊哲, 张铖怡, 刘世鹏, 等. 面向复杂环境中带钢表面缺陷检测的轻量级DCN-YOLO[J]. 计算机工程与应用, 2023, 59(15): 318-328.
LU J Z, ZHANG C Y, LIU S P, et al. Lightweight DCN-YOLO for strip surface defect detection in complex environments[J]. Computer Engineering and Applications, 2023, 59(15): 318-328.

[24] 宁纪锋, 林靖雅, 杨蜀秦, 等. 基于改进YOLO v5s的奶山羊面部识别方法[J]. 农业机械学报, 2023, 54(4): 331-337.
NING J F, LIN J Y, YANG S Q, et al. Face recognition method of dairy goat based on improved YOLO v5s[J]. Transactions of the Chinese Society of Agricultural Machinery, 2023, 54(4): 331-337.

[25] 苏志威, 黄子涵, 邱发生, 等. 基于改进YOLOv8的航空铝合金焊缝缺陷检测方法[J]. 航空动力学报, 2024, 39(6): 20230414.
SU Z W, HUANG Z H, QIU F S, et al. Weld defect detection of aviation aluminum alloy based on improved YOLOv8 [J]. Journal of Aerospace Power, 2024, 39(6): 20230414.

[26] 孙建波, 王丽杰, 麻吉辉, 等. 基于改进YOLOv5s算法的光伏组件故障检测[J]. 红外技术, 2023, 45(2): 202-208.
SUN J B, WANG L J, MA J H, et al. Photovoltaic module fault detection based on improved YOLOv5s algorithm[J]. Infrared Technology, 2023, 45(2): 202-208.

[27] 谢椿辉, 吴金明, 徐怀宇. 改进YOLOv5的无人机影像小目标检测算法[J]. 计算机工程与应用, 2023, 59(9): 198-206.
XIE C H, WU J M, XU H Y. Small object detection algorithm based on improved YOLOv5 in UAV image[J]. Computer Engineering and Applications, 2023, 59(9): 198-206.

[28] 杨断利, 王永胜, 陈辉, 等. 基于改进YOLO v6-tiny的蛋鸡啄羽行为识别与个体分类[J]. 农业机械学报, 2023, 54(5): 268-277.
YANG D L, WANG Y S, CHEN H, et al. Feather pecking

abnormal behavior identification and individual classification method of laying hens based on improved YOLO v6-tiny [J]. Transactions of the Chinese Society of Agricultural Machinery, 2023, 54(5): 268-277.

[29] 张利丰, 田莹. 改进YOLOv8的多尺度轻量型车辆目标检测算法[J]. 计算机工程与应用, 2024, 60(3): 129-137.
ZHANG L F, TIAN Y. Improved YOLOv8 multi-scale and lightweight vehicle object detection algorithm[J]. Computer Engineering and Applications, 2024, 60(3): 129-137.

[30] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions[C]//Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2015: 1-9.

[31] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting [J]. The Journal of Machine Learning Research, 2014, 15(1): 1929-1958.

[32] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[EB/OL]. [2023-12-15]. https://arxiv.org/abs/1409.1556.

[33] IOFFE S, SZEGEDY C. Batch normalization: accelerating deep network training by reducing internal covariate shift [C]//Proceedings of the 32nd International Conference on Machine Learning, Lille, Jul 6-11, 2015: 448-456.

[34] MAAS A L, HANNUN A Y, NG A Y. Rectifier nonlinearities improve neural network acoustic models[C]//Proceedings of the 30th International Conference on Machine Learning, Atlanta, Jun 16-21, 2013: 3.

[35] WANG C Y, LIAO H Y M, WU Y H, et al. CSPNet: a new backbone that can enhance learning capability of CNN[C]//Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2020: 390-391.

[36] MISRA D. Mish: a self regularized non-monotonic activation function[EB/OL]. [2023-12-15]. https://arxiv.org/abs/1908.08681.

[37] GHIASI G, LIN T Y, LE Q V. Dropblock: a regularization method for convolutional networks[C]//Advances in Neural Information Processing Systems 31, Montréal, Dec 3-8, 2018: 10750-10760.

[38] ZHU X, CHENG D, ZHANG Z, et al. An empirical study of spatial attention mechanisms in deep networks[C]//Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2019: 6688-6697.

[39] LIN T-Y, DOLLÁR P, GIRSHICK R, et al. Feature pyramid networks for object detection[C]//Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2017: 2117-2125.

[40] HE K, ZHANG X, REN S, 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.

[41] WANG W, XIE E, SONG X, et al. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network [C]//Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2019: 8440-8449.

[42] EVERINGHAM M, ESLAMI S A, VAN GOOL L, et al. The Pascal visual object classes challenge: a retrospective [J]. International Journal of Computer Vision, 2015, 111: 98-136.

[43] DENG J, DONG W, SOCHER R, et al. ImageNet: a large-scale hierarchical image database[C]//Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2009: 248-255.

[44] KUZNETSOVA A, ROM H, ALLDRIN N, et al. The open images dataset V4: unified image classification, object detection, and visual relationship detection at scale[J]. International Journal of Computer Vision, 2020, 128(7): 1956-1981.

[45] LIN T Y, MAIRE M, BELONGIE S, et al. Microsoft COCO: common objects in context[C]//Proceedings of the 13th European Conference on Computer Vision, Zurich, Sep 6-12, 2014. Cham: Springer, 2014: 740-755.

[46] XU X, JIANG Y, CHEN W, et al. Damo-YOLO: a report on real-time object detection design[EB/OL]. [2023-12-15]. https://arxiv.org/abs/2211.15444.

[47] WANG C, HE W, NIE Y, et al. Gold-YOLO: efficient object detector via gather-and-distribute mechanism[C]//Advances in Neural Information Processing Systems 36, New Orleans, Dec 10-16, 2023.

[48] XIA G S, BAI X, DING J, et al. DOTA: a large-scale dataset for object detection in aerial images[C]//Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2018: 3974-3983.

[49] ZAIDI S S A, ANSARI M S, ASLAM A, et al. A survey of modern deep learning based object detection models[J]. Digital Signal Processing, 2022, 126: 103514.

[50] 牛为华, 殷苗苗. 基于改进YOLO v5的道路小目标检测算法[J]. 传感技术学报, 2023, 36(1): 36-44.
NIU W H, YIN M M. Road small target detection algorithm based on improved YOLO v5[J]. Journal of Transduction Technology, 2023, 36(1): 36-44.

[51] 郭克友, 王苏东, 李雪, 等. 基于Dim Env-YOLO算法的昏暗场景车辆多目标检测[J]. 计算机工程, 2023, 49(3): 312-320.
GUO K Y, WANG S D, LI X, et al. Multi-target detection of vehicles in dim scenes based on Dim Env-YOLO algorithm [J]. Computer Engineering, 2023, 49(3): 312-320.

[52] 鲍文霞, 谢文杰, 胡根生, 等. 基于TPH-YOLO的无人机图像麦穗计数方法[J]. 农业工程学报, 2023, 39(1): 155-161.
BAO W X, XIE W J, HU G S, et al. Wheat ear counting method in UAV images based on TPH-YOLO[J]. Transactions of the Chinese Society of Agricultural Engineering, 2023, 39(1): 155-161.
[53] 林文树, 张金生, 何乃磊. 基于改进YOLO v4的落叶松毛虫侵害树木实时检测方法[J]. 农业机械学报, 2023, 54(4): 304-312.
LIN W S, ZHANG J S, HE N L. Real-time detection method of dendrolimus superans-infested larix gmelinii trees based on improved YOLO v4[J]. Transactions of the Chinese Society for Agricultural Machinery, 2023, 54(4): 304-312.
[54] 郝鹏飞, 刘立群, 顾任远. YOLO-RD-Apple果园异源图像遮挡果实检测模型[J]. 图学学报, 2023, 44(3): 456-464.
HAO P F, LIU L Q, GU R Y. YOLO-RD-Apple orchard heterogenous image obscured fruit detection model[J]. Journal of Graphics, 2023, 44(3): 456-464.
[55] 盛帅, 段先华, 胡维康, 等. Dynamic-YOLOX: 复杂背景下的苹果叶片病害检测模型[J]. 计算机科学与探索, 2024, 18(8): 2118-2129.
SHENG S, DUAN X H, HU W K, et al. Dynamic-YOLOX: detection model for apple leaf disease in complex background [J]. Journal of Frontiers of Computer Science and Technology, 2024, 18(8): 2118-2129.
[56] 王春梅, 刘欢. YOLOv8-VSC:一种轻量级的带钢表面缺陷检测算法[J]. 计算机科学与探索, 2024, 18(1): 151-160.
WANG C M, LIU H. YOLOv8-VSC: lightweight algorithm for strip surface defect detection[J]. Journal of Frontiers of Computer Science and Technology, 2024, 18(1): 151-160.
[57] 聂源, 赖惠成, 高古学. 改进YOLOv7+Bytetrack的小目标检测与追踪[J]. 计算机工程与应用, 2024, 60(12): 189-202.
NIE Y, LAI H C, GAO G X. Improved small target detection and tracking with YOLOv7+Bytetrack[J]. Computer Engineering and Applications, 2024, 60(12): 189-202.
[58] CHOLLET F. Xception: deep learning with depthwise separable convolutions[C]//Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2017: 1251-1258.
[59] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems 30, Long Beach, Dec 4-9, 2017: 5998-6008.
[60] WOO S, PARK J, LEE J Y, et al. CBAM: convolutional block attention module[C]//Proceedings of the 15th European Conference on Computer Vision. Cham: Springer, 2018: 3-19.
[61] HAN K, WANG Y, TIAN Q, et al. GhostNet: more features from cheap operations[C]//Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2020: 1580-1589.
[62] QIN Z, LI Z, ZHANG Z, et al. ThunderNet: towards real-time generic object detection on mobile devices[C]//Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision. Piscataway: IEEE, 2019: 6718-6727.
[63] CARION N, MASSA F, SYNNAEVE G, et al. End-to-end object detection with transformers[C]//Proceedings of the 16th European Conference on Computer Vision. Cham: Springer, 2020: 213-229.

徐彦威(1999—),男,硕士研究生,CCF学生会员,主要研究方向为目标检测、人工智能安全等。
XU Yanwei, born in 1999, M.S. candidate, CCF student member. His research interests include target detection, artificial intelligence security, etc.

李军(1974—),男,博士,教授,硕士生导师,CCF高级会员,主要研究方向为深度学习、人工智能等。
LI Jun, born in 1974, Ph.D., professor, M.S. supervisor, CCF senior member. His research interests include deep learning, artificial intelligence, etc.

董元方(1975—),女,博士,副教授,硕士生导师,CCF高级会员,主要研究方向为数据挖掘、自然语言处理等。
DONG Yuanfang, born in 1975, Ph.D., associate professor, M.S. supervisor, CCF senior member. Her research interests include data mining, natural language processing, etc.

张小利(1987—),男,博士,副教授,博士生导师,CCF高级会员,主要研究方向为计算机视觉、图像融合等。
ZHANG Xiaoli, born in 1987, Ph.D., associate professor, Ph.D. supervisor, CCF senior member. His research interests include computer vision, image fusion, etc.