第44卷第10期 电 子 与 信 息 学 报 Vol. 44No. 10
2022年10月 Journal of Electronics & Information Technology Oct. 2022

基于深度学习的YOLO目标检测综述

邵延华*[①] 张 铎[①] 楚红雨[①] 张晓强[①] 饶云波[②]
[①](西南科技大学信息工程学院 绵阳 621010)
[②](电子科技大学 成都 610054)

摘 要：目标检测是计算机视觉领域的一个基础任务和研究热点。YOLO将目标检测概括为一个回归问题，实现端到端的训练和检测，由于其良好的速度-精度平衡，近几年一直处于目标检测领域的领先地位，被成功地研究、改进和应用到众多不同领域。该文对YOLO系列算法及其重要改进、应用进行了详细调研。首先，系统地梳理了YOLO家族及重要改进，包含YOLOv1-v4, YOLOv5, Scaled-YOLOv4, YOLOR和最新的YOLOX。然后，对YOLO中重要的基础网络，损失函数进行了详细的分析和总结。其次，依据不同的改进思路或应用场景对YOLO算法进行了系统的分类归纳。例如，注意力机制、3D、航拍场景、边缘计算等。最后，总结了YOLO的特点，并结合最新的文献分析可能的改进思路和研究趋势。

关键词：目标检测；YOLO；深度学习；卷积神经网络
中图分类号：TN911.73 **文献标识码**：A **文章编号**：1009-5896(2022)10-3697-12
DOI: 10.11999/JEIT210790

A Review of YOLO Object Detection Based on Deep Learning

SHAO Yanhua[①] ZHANG Duo[①] CHU Hongyu[①]
ZHANG Xiaoqiang[①] RAO Yunbo[②]
[①](*School of Information Engineering, Southwest University of Science and Technology, Mianyang* 621010, *China*)
[②](*University of Electronic Science & Technology of China, Chengdu* 610054, *China*)

Abstract: Object detection is one of the basic tasks and research hotspots in the field of computer vision. The YOLO (You Only Look Once) frames object detection is a regression problem to implement end-to-end training and detection. YOLO becomes the leading object detector due to its good speed-accuracy balance, which has been successfully studied, improved, and applied to many different fields. YOLO series and its important improvements and applications are investigated in detail. Firstly, the YOLO family and important improvements are systematically summarized, including YOLOv1-v4, YOLOv5, Scaled-YOLOv4, YOLOR, and the latest YOLOX. Then, important backbone and loss functions in YOLO are analyzed and summarized in detail. Next, the application of YOLO is systematically classified and summarized according to different improvement ideas or scenarios, such as attention mechanisms, three-dimensional scenes, aerial scenes, edge computing, etc. Finally, the characteristics of the YOLO series are summarized and the possible improvement ideas and research trends are analyzed in combination with the latest literature.

Key words: Object detection; YOLO; Deep learning; Convolutional Neural Network (CNN)

收稿日期：2021-08-06；改回日期：2022-01-22；网络出版：2022-02-19
*通信作者： 邵延华 syh@cqu.edu.cn
基金项目：国家自然科学基金(61601382)，四川省科技计划(2019YJ0325, 2020YFG0148, 2021YFG0314)
Foundation Items: The National Natural Science Foundation of China (61601382), Sichuan Provincial Science and Technology Project (2019YJ0325, 2020YFG0148, 2021YFG0314)

1 引言

1.1 背景和意义

目标检测是计算机视觉中的一个研究热点，在很多领域都有应用需求，例如监控安全、自动驾驶、交通监控和机器人视觉等场景[1]。目标检测一般是检测一些预定义类别的目标实例(例如人和车等)。

传统目标检测依赖精巧的手工特征设计与提取[1,2]，例如方向梯度直方图(Histogram of Oriented Gradient, HOG)[3]。2012年，基于深度卷积神经网络(Convolutional Neural Network, CNN)的AlexNet[4]以显著优势夺得ImageNet图像识别比赛冠军，从此深度学习开始受到广泛的关注[5]。目标检测也逐步进入深度学习时代[1,2]。

基于深度学习的目标检测依据检测方式被分为两类[6]：两阶段检测和单阶段检测，前者是一个“从粗到细”的过程；而后者端到端“一步完成”[2]。通常，两阶段检测的定位和目标识别精度较高，单阶段检测速度较快[6]。限于篇幅，两阶段检测算法的分析可参见文献[1,6]。

通常，单阶段检测尝试直接将每个感兴趣区域分类为背景或目标对象[7]。即只通过一个阶段便可直接给出物体的类别概率和位置坐标值。典型代表有YOLO(You Only Look Once)[8]等。

2015年的YOLOv1[8]直接将图像划分为若干区域，并同时预测每个区域的边界框和概率，检测速度得到极大提高。但与当时的两阶段检测器相比，定位精度有所欠缺，特别是对小目标。

YOLO[8]是一种先进的单阶段目标检测框架，经历了v1～v4的演变，到目前为止已发展到结合传统压缩感知的YOLOR(You Only Learn One Representation)[9]和不依赖锚框的YOLOX[10]。如图1所示，简要展示了YOLO检测模型的发展历程[8–15]。

1.2 YOLO的基本思想

YOLOv1[8]的基本框架如图2所示，首先调整输入图片大小到448×448，送入CNN提取特征，然后处理网络预测结果，实现端到端的目标检测。

YOLOv1放弃传统的滑动窗口技术，其CNN将输入图片划分成$S\times S$网格，然后每个单元格负责去检测那些中心点落在该网格内的目标，每个单元格会预测B个边界框及边界框的置信度。置信度包含该边界框含有目标的可能性大小和该边界框的准确度。每个边界框预测5个元素：(x,y,w,h,c)，分别表示边界框的位置、大小与置信度。每个单元格预测$(B\times 5+C)$个值，其中C为类别数。之后利用非极大值抑制(Non-Maximum Suppression, NMS)算法进行网络预测。

后续YOLO系列均传承该基本思想。另外，基于PyTorch开发的YOLOv5[14]尚处于快速发展的实验阶段，且没有论文公开发表，其特点是代码更容易理解和扩展。因此，本文后续进行相关模块的介绍，重点是YOLOv1[8]和里程碑的YOLOv3[12]，以及在COCO(Common Objects in COntext)数据集[16]上表现更佳的最新的YOLOv4[13,15]和YOLOX[10]。

1.3 综述对比

本文调研大量目标检测技术论文和综述，进而对YOLO算法做出更全面的总结。Wu等人[7]系统地分析了现有的目标检测框架，并将其总结为检测组件、学习策略及应用和数据集3个主要部分。

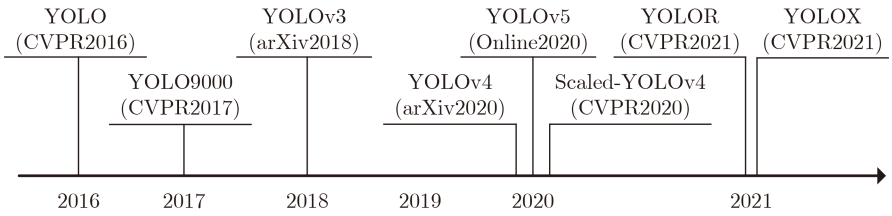


图 1 YOLO检测模型的发展历程

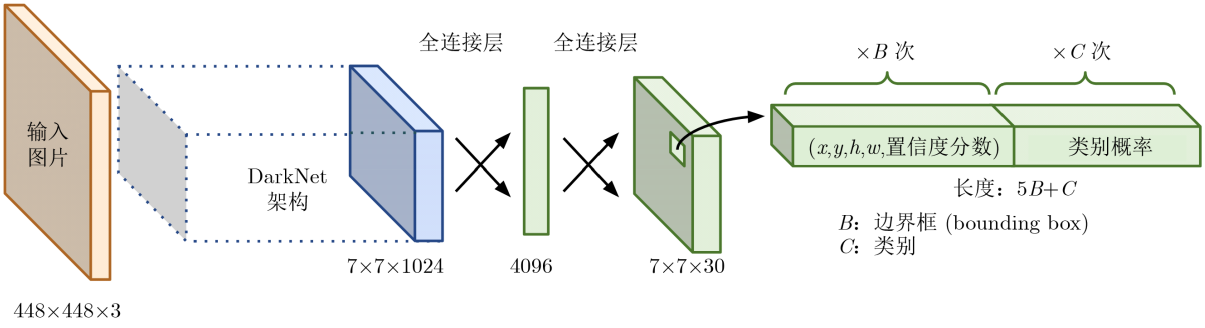


图 2 YOLOv1的网络结构

Zou等人[2]对检测器、数据集和指标等都进行了分析，还对行人、人脸和文本等检测进行了概括综述，并分析了面临的挑战。Liu等人[1]的研究主要包括检测框架、目标特征表示、目标建议生成、上下文建模、训练策略和评估指标。Jiao等人[6]对各类检测都进行了说明。同时，参考国内最新出版的综述[17]，发现检测领域发展迅速，但限于篇幅和侧重点，上述综述对YOLOv3之后的发展基本尚未提及。因此，业界急需对YOLO更专注、更细致和更新的调研分析。

1.4 文章的结构

本文结构如下。第2节讨论了YOLO家族的网络结构。第3节对目标检测领域最具代表性的损失函数进行了阐述。第4节描述了常用的数据集、指标和性能评估。然后，根据5个典型领域，在第5节对YOLO系列的几个主流的改进和应用进行了分析。最后，给出了本文总结，并对今后的工作和发展趋势进行了讨论。

2 YOLO系列的网络结构

检测器通常由两部分组成，一个是提取特征的主干网络(backbone)，即基础网络，一般在ImageNet数据集上进行预训练。另一个是预测对象类别和边界框的头部(Head)[12,13]。近几年，颈部(Neck)被构建在主干与头部之间，用于汇集不同的特征图，YOLOv4对该部分进行了详细的对比实验。

下面将对YOLO的基础网络进行详细分析。其中关于颈部和头部的更多描述可参见文献[13]。

2.1 YOLO家族的基础网络演变

2.1.1 YOLOv1的网络结构

YOLOv1[8]使用了类似GoogleNet[18]的主干网络，有24个卷积层和2个全连接层。在ImageNet上进行预训练，然后迁移到检测任务，在VOC(Visual Object Classes)数据集[19]上进行验证。

YOLOv1将输入图片分为7×7的网格，每个网格预测两个边界框，因此有7×7×2个边界框。最多识别49个目标。因此YOLOv1不利于识别密集型目标和小目标。

2.1.2 YOLOv2的网络结构

YOLOv2[11]在v1基础上，借鉴了VGG网络，构建新的主干网络Darknet-19。

YOLOv1利用全连接层直接预测边界框，丢失空间信息较多，引发定位不准。因此YOLOv2引入锚框代替v1的全连接层来预测边界框。同时，YOLOv2将输入尺寸修为416×416，最后得到13×13的特征图，维数为奇数，所得特征图恰好只有一个中心。利用这个中心位置预测中心点落入该位置的目标，对该类目标的检测会更容易。图3展现了其边界框的预测方法。

图3虚线矩形框为先验框，实线矩形框为通过网络预测的偏移量得到的预测边界框。其中(c_x,c_y)表示单元网格左上角的坐标，(p_w,p_h)为先验框的宽和高，(t_x,t_y)和(t_w,t_h)分别为网络预测边界框的中心偏移量和宽高缩放比；特征图中真值坐标为(G_x,G_y,G_w,G_h)。(b_x,b_y,b_w,b_h)为最终预测的目标边界框，从预设边界框到预测边界框的转换过程如图3右侧公式所示，其中σ函数是Sigmoid函数，其目的是将预测偏移量缩放到0～1，加快网络收敛。

YOLOv2[11]开创性地提出联合使用分类和检测的训练方法，扩展目标检测到缺乏检测样本的对象。明显提升预测准确性的同时保持了推理快的优势。

2.1.3 YOLOv3的网络结构

YOLOv3[12]的基础网络为Darknet-53，它借鉴了ResNet[20]的残差结构，加深网络结构的同时，又防止了网络梯度爆炸引发的网络难以收敛的问题。前向传播过程中，移除池化层和全连接层，通过改变卷积核的步长来改变张量的尺寸。与v2类似，Darknet-53会将输出特征缩小到输入的1/32，因此通常要求输入图片分辨率是32的倍数。

同时YOLOv3采用张量拼接扩充张量的维度，以提取更多信息，具体操作是将Darknet-53中间层和后面的某一层经过上采样之后进行拼接。

Darknet-53从第0～74层，共有53个卷积层，其余为残差层[12]。第75～105层为YOLOv3的特征融合层，其中YOLOv3增加了多尺度检测(相当于颈部)，使用了3种尺度，其输出分别是52×52，26×26，13×13用于检测小、中、大目标，每种尺度预测3个锚框。

总之，YOLOv3的预测框较YOLOv2增加了10多倍，且它们是在不同尺度上进行的，所以整体检测精度以及对小物体的检测准确率都有很大的提升，故成为单阶段检测中的里程碑算法之一。

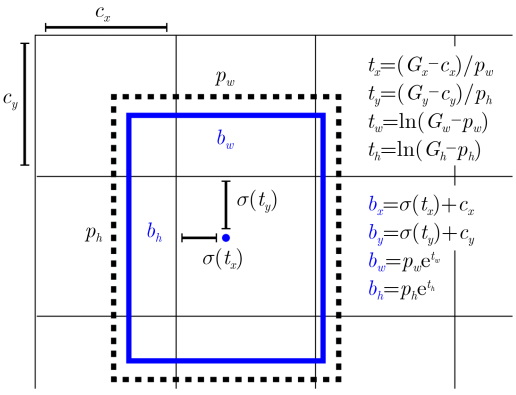


图 3 具有尺寸先验和位置预测的边界框

2.1.4 YOLOv4的网络结构

YOLOv4[13]将v3之后的各类改进方法进行总结，分为免费包和特价包。其中，前者表示提升训练而对推理速度没有影响的模块，后者表示对推理时间影响较小而性能回报较高的模块，例如主干网络中采用的局部跨阶段结构 (Cross Stage Partial, CSP)[21]在保持较高推理速度的同时，仍具有较高的精度。同时，YOLOv4更适合在单个显卡上训练[13]。

Bochkovskiy等人[13]发现模型分类最优时，其检测未必是最佳的，例如CSPResNeXt-50的分类精度比CSPDarknet-53要高，但后者检测精度更高。同时使用免费包和Mish[22]提高了检测的准确率。因此YOLOv4选用CSPDarknet-53为主干网络。

基础网络部分，YOLOv4的整体架构与YOLOv3相同，但对各个子结构都进行了改进。图4展现了Darknet-53与CSPDarknet-53[21]两种网络结构，其中黑色表示Darknet-53，CSPDarknet-53网络只需换为红色框中的结构，同时将滤波器的数值换为括号中的红色数值即可，YOLOv4删去了最后的池化层、全连接层以及Softmax层，其主干网络有5个CSP模块[13]。

	类别	通道数	尺寸	输出
	卷积	32	3×3	256×256
	卷积	64	3×3/2	128×128
	局部跨阶段结构			
1×	卷积	32	1×1	
	卷积	64	3×3	
	残差			128×128
	卷积	128	3×3/2	64×64
	局部跨阶段结构			
2×	卷积	64	1×1	
	卷积	128(64)	3×3	
	残差			64×64
	卷积	256	3×3/2	32×32
	局部跨阶段结构			
8×	卷积	128	1×1	
	卷积	256(128)	3×3	
	残差			32×32
	卷积	512	3×3/2	16×16
	局部跨阶段结构			
8×	卷积	256	1×1	
	卷积	512(256)	3×3	
	残差			16×16
	卷积	1024	3×3/2	8×8
	局部跨阶段结构			
4×	卷积	512	1×1	
	卷积	1024(512)	3×3	
	残差			8×8
	池化		全局	
	连接		1000	
	Softmax			

图 4 Darknet-53与CSPDarknet-53

颈部模块，YOLOv4引入空间金字塔池化(Spatial Pyramid Pooling, SPP)与路径聚合网络(Path Aggregation Network, PANet)[23]模块。其中，SPP显著增加了感受野，在不降低运行速度的情况下分离了重要的上下文特征。PANet[23]代替YOLOv3中的特征金字塔网络(Feature Pyramid Network, FPN)[24,25]进行参数聚合，并使用张量连接代替原来的短连接。

头部模块，YOLOv4继承了YOLOv3的多尺度思想来进行预测。

2.1.5 YOLOv5的网络结构

YOLOv5[14]基本结构与YOLOv4类似，最大不同为根据不同通道的尺度缩放，依据模型从小到大构建了YOLOv5-N/S/M/L/X 5种模型。

2.1.6 YOLOX的网络结构

YOLOX[10]依据YOLOv3[12]和YOLOv5[14]，使用了CSPNet[21]，SiLU[26]激活函数以及PANet[23]，并遵循缩放规则设计了YOLOX-S/M/L/X 4种模型；然后将模型进一步缩小构建了YOLOX-Tiny和面向移动边缘设备的YOLOX-Nano。

2.2 轻量化

部分检测算法的计算量较大或模型较大，对计算能力和功耗受限的边缘计算设备或物联网场景的部署提出了巨大的挑战。

当前，轻量化主要有两个方式：一种是设计轻量化的基础网络，比如MobileNet系列[27]、ShuffleNet系列[28]等。主要通过深度可分离卷积、分组卷积、可调超参数降低空间分辨率和减少通道数。另一种是压缩整体网络参数，减少卷积层等设计一些“小而薄”的网络，例如YOLO的tiny系列。李成跃等人[29]基于YOLOv3，保留对提取特征有较大帮助的全卷积结构、FPN以及ResNet的同时，尽可能减少每层的参数量和残差层数，并加入密集连接网络和SPP。最终检测速度大幅度优化，且检测精度优于YOLOv3-tiny。

2.3 注意力

YOLO还可通过引入注意力机制来提高算法的检测精度。注意力机制[30–32]被应用于许多计算机视觉任务，如图像分类[30]等。从原理上来讲，注意力模型可分为空间型、通道型和混合型3种。

文献[31]通过添加全局上下文(Global Context, GC)，提升检测效果，其GC对来自空间和通道维度的输入特征分配不同的权重，以突出有用的信息。

YOLOv4尝试了多种注意力机制[13,30,32]，并最终选用空间注意力机制[32]。对新场景和新算法的实验有一定的指导意义。

3　损失函数与激活函数

大多数深度学习算法都涉及某种形式的优化[6]。目标检测也是如此，它不仅定位目标，还需要分类。损失函数、交并比(Intersection over Union, IoU)损失[33,34]和激活函数这3部分对YOLO至关重要。下面对它们分别进行分析和总结。

3.1 YOLO的损失函数

YOLOv1的一个主要创新是设计了一个巧妙的损失函数，涵盖了目标检测中必要的几个损失函数模块，如边界框、目标置信度和目标类别。定义为

$$\mathcal{L}_{\text{all}}=\mathcal{L}_{\text{bbox}}+\mathcal{L}_{\text{obj}}+\mathcal{L}_{\text{class}} \tag{1}$$

其中，$\mathcal{L}_{\text{class}}$表示模型的预测目标类别损失，$\mathcal{L}_{\text{obj}}$表示模型的目标置信度损失，$\mathcal{L}_{\text{bbox}}$表示模型的边界框损失，定义为

$$\mathcal{L}_{\text{bbox}}=\mathcal{L}_{xy}+\mathcal{L}_{wh} \tag{2}$$

3.1.1 YOLOv1的损失函数

YOLOv1[8]中

$$\mathcal{L}_{xy}=\lambda_{\text{coord}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{\text{obj}}\left[\left(x_i-\widehat{x}_i\right)^2+\left(y_i-\hat{y}_i\right)^2\right] \tag{3}$$

$$\begin{aligned}\mathcal{L}_{wh}=&\lambda_{\text{coord}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{\text{obj}}\left[\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2\right.\\&\left.+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2\right]\end{aligned} \tag{4}$$

该模块用于坐标预测，$\mathbb{1}_{ij}^{\text{obj}}$判断第$i$个网格中第$j$个边界框是否负责该目标；与目标真值框的IoU最大的边界框负责检测该目标

$$\begin{aligned}\mathcal{L}_{\text{obj}}=&\lambda_{\text{obj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{\text{obj}}\left(C_i-\hat{C}_i\right)^2\\&+\lambda_{\text{noobj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{\text{noobj}}\left(C_i-\hat{C}_i\right)^2\end{aligned} \tag{5}$$

该模块表达目标置信度的预测，第1项表示含有目标的边界框的置信度预测，第2项表示不含目标的边界框的置信度预测。$\mathbb{1}_{ij}^{\text{noobj}}$表示第$i$个网格的第$j$个边界框不预测目标。

$$\mathcal{L}_{\text{class}}=\sum_{i=0}^{S^2}\mathbb{1}_{ij}^{\text{obj}}\sum_{c\in\text{classes}}\left(p_i(c)-\hat{p}_i(c)\right)^2 \tag{6}$$

该部分表示类别预测，$\mathbb{1}_{i}^{\text{obj}}$判断是否有目标的中心落在网格$i$中，若网格中含有目标的中心，就负责预测该目标的类别概率。由于模型中3种类别的损失在整体损失函数中贡献不同，因此被赋予不同的比例因子，λ_{coord}, λ_{noobj}和λ_{obj}。

YOLOv1的损失函数中，大小物体的IoU误差在网络训练中对损失函数贡献值接近。因此，对于小物体，小的IoU误差也会对网络优化过程造成很大的影响，从而降低物体检测的定位准确性。

3.1.2 YOLOv3的损失函数

YOLOv3[12]的损失函数如式(7)—式(9)所示，其$\mathcal{L}_{\text{obj}}$采用了二分交叉熵。由于YOLOv3在3种不同尺度下进行预测，所以最终的损失函数为3种尺度下损失函数之和

$$\begin{aligned}\mathcal{L}_{wh}=&\lambda_{\text{coord}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{\text{obj}}\left(2-w_i^j*h_i^j\right)\\&*\left[\left(w_i^j*\hat{w}_i^j\right)^2+\left(h_i^j-\hat{h}_i^j\right)^2\right]\end{aligned} \tag{7}$$

在训练过程中，利用网格本身满足条件的最佳锚框来回归宽高的坐标偏差

$$\begin{aligned}\mathcal{L}_{\text{obj}}=&\sum_{S^2}^{i=0}\sum_{B}^{j=0}\mathbb{1}_{ij}^{\text{obj}}\left[\widehat{C}_i^j\ln(C_i^j)+(1-\widehat{C}_i^j)\ln(1-C_i^j)\right]\\&-\lambda_{\text{noobj}}\sum_{S^2}^{i=0}\sum_{B}^{j=0}\mathbb{1}_{ij}^{\text{noobj}}\\&\cdot\left[\widehat{C}_i^j\ln(C_i^j)+(1-\widehat{C}_i^j)\ln(1-C_i^j)\right]\end{aligned} \tag{8}$$

式(8)采用交叉熵，此外，不管网格是否负责某个目标物体，都会计算置信度误差，但输入图像的大部分空间不包含目标物体，只有少部分空间包含了目标物体，因此需要引入权重对不包含目标物体的置信度进行约束。$\hat{C}_i^j$表示真实值，由网格的边界框是否负责预测某个对象来决定，负责为1，否则为0。C_i^j表示拟合值

$$\begin{aligned}\mathcal{L}_{\text{class}}=&\sum_{S^2}^{i=0}\mathbb{1}_{ij}^{\text{obj}}\sum_{c\in\text{classes}}\\&\cdot\left[\widehat{P}_i^j\ln(P_i^j)+(1-\widehat{P}_i^j)\ln(1-P_i^j)\right]\end{aligned} \tag{9}$$

该损失同样采用交叉熵，只有当第i个网格的第j个锚点负责某个真实目标物体时，该锚点相应的真值框才被用来计算分类损失精度。

目标检测是在多个尺度上进行的，对于每个网格，目标的宽度$\hat{w}$，高度$\hat{h}$，目标的中心坐标$(\hat{x},\hat{y})$以及目标类别都被预测出来。最后，和YOLOv1类似，采用非极大值抑制滤除低置信度的检测框。

YOLOv3训练中忽略了大量背景产生的损失，在一定程度上提升了正样本的作用。

3.1.3 YOLOv4的损失函数

YOLOv4[13]认为(x,y,w,h)并不独立，需要一个损失函数表达它们之间的相互关系，采用CIoU (Complete-IoU)[33]取得较好效果。关于IoU[33,34]的内容将在下节进行分析。

3.2 IoU

IoU[33]是目标检测的一个重要评价指标，可衡量预测框和真值之间的距离。常用来计算$\mathcal{L}_{\text{bbox}}$，多个检测框可能有相同大小的损失，但IoU可能差异很大，为此需引入IoU损失。其定义为

$$\mathcal{L}_{\text{IoU}}=1-\text{IoU}=1-\frac{|B\cap B^{\text{gt}}|}{|B\cup B^{\text{gt}}|} \tag{10}$$

其中，B为预测框，B^{gt}为目标框。

3.2.1 GIoU (Generalized-IoU)

若两框无交集，则$\text{IoU}=0$，此时因损失为0，没有梯度回传，学习训练失败。因此提出GIoU[34]

$$\mathcal{L}_{\text{GIoU}}=\mathcal{L}_{\text{IoU}}+\frac{|C-B\cup B^{\text{gt}}|}{|C|} \tag{11}$$

其中，C为两个框的最小闭包区域面积。GIoU，可显著提高YOLOv3的定位精度[34]。

3.2.2 DIoU (Distance-IoU)

当目标框完全包裹预测框时，IoU和GIoU的值相等，此时GIoU退化为IoU，即GIoU严重依赖IoU，因此在两个垂直方向，误差大，难收敛。由此提出DIoU[33]

$$\mathcal{L}_{\text{DIoU}}=1-\text{IoU}+\frac{\rho^2(b,b^{\text{gt}})}{c^2} \tag{12}$$

其中，b和b^{gt}表示B和B^{gt}的中心点，$\rho(\cdot)$是欧氏距离，c是覆盖这两个框的最小封闭框的对角线长度。

DIoU的惩罚项直接最小化两个中心点之间的距离，对包含两个框在水平方向和垂直方向上的情况，可加速模型回归。

同时，NMS是目标检测的主流后处理方法，可过滤预测同一物体、但效果差的边界框，仅保留较好的边界框。基于DIoU的NMS，其结果更加合理和有效[35]。

3.2.3 CIoU (Complete-IoU)

目标框损失应考虑3个重要的几何因素：重叠面积、中心点距离和长宽比。而DIoU未考虑到检测框的长宽比。长宽比更接近的边框应有更低的损失，因此提出了CIoU[33]

$$\mathcal{L}_{\text{CIoU}}=\mathcal{L}_{\text{IoU}}+\frac{\rho^2(b,b^{\text{gt}})}{c^2}+\alpha v \tag{13}$$

其中，α是一个正的度量参数，同时v用来度量宽高比的一致性，具体定义参见文献[33]。通过这种方式，尤其是在非重叠情况下，重叠区域因子在回归上具有更高的优先级。

文献[30]对各种IoU在YOLOv3上用VOC 2007数据集进行了丰富的消融实验，可知各IoU对YOLOv3的提升程度。YOLOv3引入CIoU结合DIoU-NMS，可带来5.91%平均精度(Average Precision, AP)的巨大改进[33]。

3.2.4 其他

面向具有高“长宽比”和复杂背景的对象时，PIoU(Pixels-IoU)[36]可显著提高方向边界框检测性能，优于GIoU。

3.3 激活函数

YOLOv1-v2的最后一层使用Softmax激活函数，对每个框分配一个类别(得分最大的一个)，不适合目标可能存在重叠类别标签的数据集。同时，Softmax可被多个独立的Logistic替代，且准确率不会下降。因此v3采用Logistic回归，用于对锚框包围的部分进行目标性评分，即该位置是目标的可能性(得分)有多大。这一步是在预测之前进行的，可去掉不必要锚框，减少计算量。

YOLOv4[13]在主干网络中采用了Mish激活函数[23]，其特点有低成本、平滑、非单调、无上界、有下界等，与线性整流函数(Rectified Linear Unit, ReLU)、Swish等常用函数相比，提高了性能[23]。

YOLOv5[14]尚处于快速发展的实验阶段，中间层使用Leaky ReLU激活函数，最后的检测层使用Sigmoid激活函数。

YOLOR[9]与YOLOX[10]选用SiLU激活函数[26]。

4 数据集与性能评估

4.1 数据集

大量可靠的带标注数据集是深度学习成功的前提之一。数据集不仅是衡量算法性能的基础，还极大地推动了目标检测的发展，不同数据集特点不同[1,2,6,7]。本节将分析VOC[19], COCO[16]和VisDrone[37]等目标检测领域中的部分典型数据集。

4.1.1 VOC数据集

VOC数据集[19]是计算机视觉中常用的数据集之一，主要任务有分类、检测和分割。针对检测任务，典型的VOC2007由train/val/test 3部分组成，而VOC2012则将训练集和验证集组合起来分为trainval/test两部分。

VOC中物体可分4大类，亦可细粒度的分为20类，其中，“person”类实例最多，“sheep”是数据实例最少的类，此外，VOC还存在一定的检测难度，例如“cat”和“dog”在视觉上较相似。

4.1.2 COCO数据集

随着深度学习的发展，VOC数据集的规模略显不足，业界需要新的数据集，其中COCO[16]是最

著名的一个。其官网为：http://mscoco.org/。可用来图像识别、分割、检测等任务。分为训练集、验证集和测试集。具体来讲，目标检测赛道的主要特点如下：(1)多目标；(2)部分对象存在遮挡与噪声；(3)包含各种尺寸的物体；(4)超过300000张图片；(5)超过2000000个实例；(6)80个目标类别[16]。

COCO中单图的平均目标数是VOC的3倍左右，且小目标居多。在种类方面，COCO平均每幅图片包含3.5个类别。

4.1.3 VisDrone数据集

截至目前，各领域对于无人机的应用需求非常广泛。VisDrone数据集[37]发布于2018年，并在2019年由天津大学等进行扩展。为目标检测提供了10209张图片，其中6471张图像用于训练，548张用于验证，3190张用于测试，同时还提供了96个用于目标检测的视频剪辑，包括56个用于训练(共计24201帧)，7个用于验证(共计2819帧)和33个用于测试(共计12968帧)[37]。相对而言，VisDrone数据集场景复杂，遮挡情况严重，同时标签密集，对算法的要求也随之上升。其部分典型场景如图5所示。

4.1.4 其他

除了上述典型数据集，各行业的具体需求催生一些检测比赛，使智能检测应用至各个领域。如中国模式识别与计算机视觉学术会议(chinese conference on Pattern Recognition and Computer Vision, PRCV)对火焰、后厨老鼠、安全帽的检测比赛，以及Kaggle比赛对小麦的检测等[38]，其数据集示例如图6所示。YOLO在这些比赛中通常都会占有一席之地。促进了YOLO的进步和落地实战，更加验证了其优势。

4.2 指标

如何评估检测器的有效性？不同的数据集给出了不同的指标答案。

4.2.1 VOC数据集的指标

对于VOC数据集，使用插值平均精度来评估分类和检测。其设计是为了在出现重检测、误检测以及目标丢失时对算法进行惩罚。主要指标为[6]

$$\text{recall}(t)=\frac{\sum_{ij}1\left[s_{ij}\geq \text{t}\right]z_{ij}}{N} \tag{14}$$

$$\text{precision}(t)=\frac{\sum_{ij}1\left[s_{ij}\geq \text{t}\right]z_{ij}}{\sum_{ij}1\left[s_{\text{ij}}\geq \text{t}\right]} \tag{15}$$

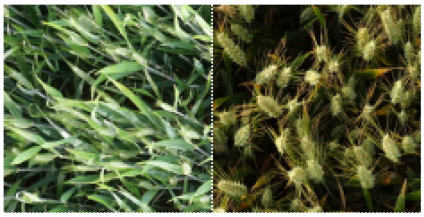
其中，t是一个常数为0.5，表示预测框与真值框之间IoU的阈值。i为第i个图像的索引，j为第j个目标的索引。N为真值框的个数。当$1\left[s_{ij}\geq \text{t}\right]=1$为真时该值取1，否则取0。若根据阈值标准，第$i$个图像的第$j$个目标的检测结果与真值框相匹配，则$z_{ij}$取1，否则取0。

精确度(precision)即预测为真的真实正例与所有预测正例的比值，召回率(recall)即预测为真与所有真实正例的比值[1,6]。几乎所有的目标检测论文中，基于0.5IoU的平均精度均值(mean AP, mAP)已成为多年来目标检测问题最重要的实际度量标准之一[2]。在判断检测速度方面，每秒处理帧数(Frames Per Second, fps)是公认的度量方法。

4.2.2 COCO数据集的指标

COCO的指标比VOC更严格，采用多IoU评估方法更关注边界框的准确性。在阈值设置方面，不同于VOC的常值0.5，COCO将其分为以0.5为起始值，0.95为终止值，间隔为0.05的10个值[16]。通过计算不同阈值的平均精度，来确定度量标准，更能反映检测算法的综合性能。此外，COCO还分别计算了大、中、小3类物体的平均精度，来衡量检测

图 5 VisDrone2019数据集示例[37]



(a) Kaggle小麦检测数据集示例

(b) PRCV比赛数据集示例

图 6 Kaggle小麦检测数据集与PRCV比赛数据集示例

器检测不同尺度目标的性能。在COCO评价指标中，所有的AP默认为mAP[16]。

VisDrone数据集采用COCO的度量指标。

4.3 表现

4.3.1 VOC数据集

表1列举了不同YOLO在VOC2012上的检测结果。其中，当输入尺寸增大时，精度会随之增加，相应的检测速度会随之降低。

由于YOLOv4在速度和精度方面相较之前版本都有了重大提升，因此YOLOv4之后的模型一般在规模更大的COCO数据集上进行测试。

4.3.2 COCO数据集

表2列举了几种典型的YOLO算法在COCO数据集的结果，并给出了其使用的图形处理器(Graphics Processing Unit, GPU)类别。可见COCO数据集对目前的检测器来说仍是一个巨大的挑战。

4.3.3 VisDrone数据集

尽管在通用目标检测场景有了巨大进步，但是面向航拍场景的目标检测，精度依然较低[41]。另外，无人机的机载计算能力通常受限，因此航拍目标检测仍是一个巨大的挑战[37]。

历年VisDrone比赛中YOLO被许多团队采用并改进，详情可在VisDrone官方网址：http://aiskyeye.com/visdrone-2020-leaderboard/中查询。

4.4 YOLO程序框架

YOLO的代码实现主要基于Darknet和PyTorch。Darknet[8]是一种基于CUDA (Compute Unified Device Architecture) 和 C的较PyTorch与TensorFlow轻量的开源框架。其支持中央处理器(Central Processing Unit, CPU)与GPU，具有易安装，编译速度快、依赖少和易部署等优点，因此其更适合用来研究底层。YOLOv1-v4的文献开源代码均基于Darknet。YOLOv5代码基于PyTorch，且代码不再适配Darknet的数据组织方式。

5 改进与应用

除常规场景外，YOLO还被应用到其他场景中。例如3D场景[42–44]、边缘计算[45,46]、航拍场景[47–49]等。针对不同的场景和需求，众多研究者对YOLO提出了不同的改进策略。

5.1 3D场景

对于目标抓取这一典型应用场景，不仅需要物体的3D空间位置(x,y,z)，同时还需要物体的旋转状态。Tekin等人[42]提出一种仅使用一张2D图片来预测物体6D姿态的方法。除此之外，结合视觉语义[43]和引入3D-IoU[44]被证明是有效的手段。

5.2 移动端边缘计算

边缘计算需要更好的平衡精度和网络体积[40]。其中自动售货机也是边缘计算检测的一种典型应用场景，例如Lee等人[46]通过消除对不感兴趣区域的计算，在精度下降2.81%的情况下，将YOLOv3的运算速度提高了3.29倍。基于现场可编程逻辑门阵

表 1 YOLO系列在VOC2012的检测结果

检测框架	mAP(%)	fps	GPU
YOLO[8]	57.9	–	TitanX
YOLOv3 416[12]	79.3	39	1080Ti
SPP-YOLO 416[39]	77.5	65.2	1080Ti
DC-SPP-YOLO 416[39]	78.4	56.3	1080Ti
GC-YOLOv3 544[31]	83.7	31	1080Ti

表 2 各类YOLO算法在COCO test2017上的表现

检测框架	主干网络	尺寸	fps	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	GPU
YOLOv3[12], arXiv2018	Darknet-53	416	35	31.0	55.3	32.3	15.2	33.2	42.8	Maxwell GPU
YOLOv3-tiny[12], arXiv2018	Darknet Ref	416	330	–	33.1	–	–	–	–	GTX 1080Ti
GC-YOLOv3[31], MDPI2020	Darknet 53	416	28	–	55.5	–	–	–	–	GTX 1080Ti
YOLOv4-CSP[13], arXiv2020	CSPDarknet-53	640	70	47.5	66.2	51.7	28.2	51.2	59.8	Volta GPU
YOLOv5-S[14]	Modified CSP v5	640	156.3	36.7	55.4	–	–	–	–	Volta GPU
YOLOv5-X[14]	Modified CSP v5	640	82.6	50.4	68.8	–	–	–	–	Volta GPU
PP-YOLOv2[40], arXiv2021	ResNet50-vd-dcn[28]	640	68.9	49.5	68.2	54.4	30.7	52.9	61.2	Volta GPU
YOLOR-P6[9], arXiv2021	–	1280	49	52.6	70.6	57.6	34.7	56.6	64.2	Volta GPU
YOLOX-X[10], arXiv2021	Modified CSP v5	640	57.8	51.2	69.6	55.7	31.2	56.1	66.1	Volta GPU

列的YOLO加速实现了体积小、运行速度快、功耗小的需求[45]。

5.3 模型缩放

为了应对不同平台的部署需求，研究人员提出了尺度缩放网络，基于基础结构，通过加深、加宽网络或者改变网络分辨率从而实现神经网的尺度缩放[15]。典型的YOLOv4-tiny适合部署在低端设备。最新的YOLOX[10]更是推出了6个不同大小的模型面对不同的需求。

5.4 航拍场景

随着无人机的广泛应用，航拍目标检测应运而生。其主要问题有：目标尺度变化大、背景复杂以及场景明暗变化等问题[47–49]。针对上述问题，通道随机混合、分组卷积、可变形卷积模块[47]和剪枝[49]被证明是有效的。自Tijtgat等人[48]在NVIDIA Jetson TX2上成功部署YOLOv2以后，NVIDIA Jetson TX2已成为航拍算法验证的标准平台之一。

5.5 其他

5.5.1 交通

当前，面向交通场景的YOLO应用越来越多。例如，汽车牌照检测[50]、卡车的盲区检测[51]、汽车类型检测[52]和车辆徽标检测[53]。可进一步提升车辆行驶安全以及交通管制的智能化程度。

5.5.2 农业

YOLO目标检测在农业方面也有重要需求和应用。例如农业温室检测[54]、苹果花实时检测[55]和青芒果检测[56]。

5.5.3 医学

在医学领域，2020年便有多篇应用YOLO的论文发表[57,58]。例如，基于YOLOv2的白细胞定位[57]和基于YOLO心脏矢量流映射分析与评价方法[58]。上述研究证明了YOLO在医学领域的适应性。

5.5.4 行人检测

行人检测是智能自动驾驶中的重要领域[59,60]。针对红外感知场景，在U-FOV红外行人数据集上，通过构建特征金字塔和注意力模块，对比YOLOv3提升了26.49%[60]。Krišto等人[61]使用YOLOv3在热图像中自动检测人，同时给出了人体和动物在热图像中的识别结果。

5.5.5 工业

YOLO算法在电力[62]、建筑足迹[63]以及航空工业[64]等方面也有一定的应用。例如，Liu等人[62]基于改进的YOLOv3进行了输电线路检测；Xie等人[63]基于YOLO实现检测小目标与密集分布的建筑足迹；Luo等人[64]基于YOLOv3-tiny的改进模型，使得涡旋区域检测有了更好更快的提升。

6 总结与展望

本文根据YOLO的模型结构、损失函数、交并比等的改进，对YOLO系列的几个重要版本进行了详细的分析与总结。并对经注意力、轻量化等方式改进后的YOLO算法进行了对比分析，体现了其被广泛使用的原因。之后对常用的典型数据集进行了介绍，展现了YOLO系列算法目前所能达到的效果。最后，分析了YOLO在不同场景下的应用。

综合当前目标检测的研究现状，对今后的研究做出如下展望：

(1) 尽管YOLO系列是目标检测领域中速度-精度均衡的佼佼者，但其主要工作是面向电脑端。当前，边缘计算已成为人工智能(Artificial Intelligence, AI)发展的重要趋势之一。面向Nvidia Jetson TX2，Nano和树莓派等嵌入式AI计算设备，如何使YOLO更轻更快是一个值得深思的问题。

(2) 技术角度讲，自里程碑式的YOLOv3到集成创新大成者YOLOv4，再到最新的结合传统压缩感知的YOLOR[9]和不依赖锚框的YOLOX[10]。融合各种先进的算法成为YOLO算法发展的重要方式，例如探索融入注意力聚合FPN，变结构IoU损失以及广义焦点损失等。

(3) 经过数年的发展，YOLO算法虽取得了显著的进步，但在解决更多现实问题时仍需要进一步的研究，例如旋转矩形框、3D目标、少样本、航拍场景以及如何将所研究算法进行TensorRT等优化部署等。

参 考 文 献

[1] LIU Li, OUYANG Wanli, WANG Xiaogang, *et al*. Deep learning for generic object detection: A survey[J]. *International Journal of Computer Vision*, 2020, 128(2): 261–318. doi: 10.1007/s11263-019-01247-4.

[2] ZOU Zhengxia, SHI Zhenwei, GUO Yuhong, *et al*. Object detection in 20 years: A survey[J]. arXiv preprint arXiv: 1905.05055, 2019.

[3] DALAL N and TRIGGS B. Histograms of oriented gradients for human detection[C]. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, USA, 2005: 886–893. doi: 10.1109/CVPR.2005.177.

[4] KRIZHEVSKY A, SUTSKEVER I, and HINTON G E. ImageNet classification with deep convolutional neural networks[C]. The 25th International Conference on Neural Information Processing Systems, Lake Tahoe, USA, 2012: 1097–1105.

[5] LECUN Y, BENGIO Y, and HINTON G. Deep learning[J]. *Nature*, 2015, 521(7553): 436–444. doi: 10.1038/nature14539.

[6] JIAO Licheng, ZHANG Fan, LIU Fang, *et al*. A survey of deep learning-based object detection[J]. *IEEE Access*, 2019, 7: 128837–128868. doi: 10.1109/access.2019.2939201.

[7] WU Xiongwei, SAHOO D, and HOI S C H. Recent advances in deep learning for object detection[J]. *Neurocomputing*, 2020, 396: 39–64. doi: 10.1016/j.neucom.2020.01.085.

[8] REDMON J, DIVVALA S, GIRSHICK R, *et al*. You only look once: Unified, real-time object detection[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 2016: 779–788. doi: 10.1109/CVPR.2016.91.

[9] WANG C Y, YEH I H, and LIAO H Y M. You only learn one representation: Unified network for multiple tasks[J]. arXiv preprint arXiv: 2105.04206, 2021.

[10] GE Zheng, LIU Songtao, WANG Feng, *et al*. YOLOX: Exceeding YOLO series in 2021[J]. arXiv preprint arXiv: 2107.08430, 2021.

[11] REDMON J and FARHADI A. YOLO9000: Better, faster, stronger[C]. 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, 2017: 6517–6525. doi: 10.1109/CVPR.2017.690.

[12] REDMON J and FARHADI A. YOLOv3: An incremental improvement[J]. arXiv preprint arXiv: 1804.02767, 2018.

[13] BOCHKOVSKIY A, WANG C Y, and LIAO H Y M. YOLOv4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv: 2004.10934, 2020.

[14] JOCHER G, STOKEN A, BOROVEC J, *et al*. Ultralytics/YOLOv5: V3.1 - bug fixes and performance improvements[EB/OL].https://doi.org/10.5281/zenodo.4154370, 2020. doi: 10.5281/zenodo.4154370,2020.

[15] WANG C Y, BOCHKOVSKIY A, and LIAO H Y M. Scaled-YOLOv4: Scaling cross stage partial network[C]. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, USA, 2021: 13024–13033. doi: 10.1109/CVPR46437.2021.01283.

[16] LIN T Y, MAIRE M, BELONGIE S, *et al*. Microsoft COCO: Common objects in context[C]. 13th European Conference on Computer Vision, Zurich, Switzerland, 2014: 740–755. doi: 10.1007/978-3-319-10602-1_48.

[17] 罗会兰, 陈鸿坤. 基于深度学习的目标检测研究综述[J]. 电子学报, 2020, 48(6): 1230–1239. doi: 10.3969/j.issn.0372-2112.2020.06.026.

LUO Huilan and CHEN Hongkun. Survey of object detection based on deep learning[J]. *Acta Electronica Sinica*, 2020, 48(6): 1230–1239. doi: 10.3969/j.issn.0372-2112.2020.06.026.

[18] SZEGEDY C, LIU Wei, JIA Yangqing, *et al*. Going deeper with convolutions[C]. 2015 IEEE Conference on Computer Vision and Pattern Recognition, Boston, USA, 2015: 1–9. doi: 10.1109/CVPR.2015.7298594.

[19] EVERINGHAM M, ESLAMI S M A, VAN GOOL L, *et al*. The PASCAL visual object classes challenge: A retrospective[J]. *International Journal of Computer Vision*, 2015, 111(1): 98–136. doi: 10.1007/s11263-014-0733-5.

[20] HE Kaiming, ZHANG Xiangyu, REN Shaoqing, *et al*. Deep residual learning for image recognition[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 2016: 770–778. doi: 10.1109/CVPR.2016.90.

[21] WANG C Y, LIAO H Y M, WU Y H, *et al*. CSPNet: A new backbone that can enhance learning capability of CNN[C]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, USA, 2020: 1571–1580. doi: 10.1109/CVPRW50498.2020.00203.

[22] MISRA D. Mish: A self regularized non-monotonic activation function[J]. arXiv preprint arXiv: 1908.08681, 2019.

[23] LIU Shu, QI Lu, QIN Haifang, *et al*. Path aggregation network for instance segmentation[C]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, 2018: 8759–8768. doi: 10.1109/CVPR.2018.00913.

[24] LIN T Y, DOLLÁR P, GIRSHICK R, *et al*. Feature pyramid networks for object detection[C]. The IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, USA, 2017: 936–944. doi: 10.1109/CVPR.2017.106.

[25] GHIASI G, LIN T Y, and LE Q V. NAS-FPN: Learning scalable feature pyramid architecture for object detection[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, USA, 2019: 7029–7038. doi: 10.1109/CVPR.2019.00720.

[26] ELFWING S, UCHIBE E, and DOYA K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning[J]. *Neural Networks*, 2018, 107: 3–11. doi: 10.1016/j.neunet.2017.12.012.

[27] HOWARD A, SANDLER M, CHEN Bo, *et al*. Searching for MobileNetV3[C]. 2019 IEEE/CVF International Conference on Computer Vision, Seoul, Korea (South), 2019: 1314–1324. doi: 10.1109/ICCV.2019.00140.

[28] MA Ningning, ZHANG Xiangyu, ZHENG Haitao, *et al*. ShuffleNet V2: Practical guidelines for efficient CNN architecture design[C]. 2018 15th European Conference on Computer Vision, Munich, Germany, 2018: 122–138. doi: 10.1007/978-3-030-01264-9_8.

[29] 李成跃, 姚剑敏, 林志贤, 等. 基于改进YOLO轻量化网络的目标检测方法[J]. 激光与光电子学进展, 2020, 57(14): 141003. doi: 10.3788/LOP57.141003.

LI Chengyue, YAO Jianmin, LIN Zhixian, *et al*. Object detection method based on improved YOLO lightweight

network[J]. *Laser & Optoelectronics Progress*, 2020, 57(14): 141003. doi: 10.3788/LOP57.141003.

[30] HU Jie, SHEN Li, and SUN Gang. Squeeze-and-excitation networks[C]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, 2018: 7132–7141. doi: 10.1109/CVPR.2018.00745.

[31] YANG Yang and DENG Hongmin. GC-YOLOv3: You only look once with global context block[J]. *Electronics*, 2020, 9(8): 1235. doi: 10.3390/electronics9081235.

[32] WOO S, PARK J, LEE J Y, *et al*. CBAM: Convolutional block attention module[C]. 2018 15th European Conference on Computer Vision, Munich, Germany, 2018: 3–19. doi: 10.1007/978-3-030-01234-2_1.

[33] ZHENG Zhaohui, WANG Ping, LIU Wei, *et al*. Distance-IoU loss: Faster and better learning for bounding box regression[C]. The 34th 2020 AAAI Conference on Artificial Intelligence, New York, USA, 2020: 12993–13000. doi: 10.1609/aaai.v34i07.6999.

[34] REZATOFIGHI H, TSOI N, GWAK J Y, *et al*. Generalized intersection over union: A metric and a loss for bounding box regression[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, USA, 2019: 658–666. doi: 10.1109/CVPR.2019.00075.

[35] BODLA N, SINGH B, CHELLAPPA R, *et al*. Soft-NMS--improving object detection with one line of code[C]. 2017 IEEE International Conference on Computer Vision, Venice, Italy, 2017: 5562–5570. doi: 10.1109/ICCV.2017.593.

[36] CHEN Zhiming, CHEN Kean, LIN Weiyao, *et al*. PIoU loss: Towards accurate oriented object detection in complex environments[C]. 16th European Conference on Computer Vision, Glasgow, UK, 2020: 195–211. doi: 10.1007/978-3-030-58558-7_12.

[37] DU Dawei, ZHU Pengfei, WEN Longyin, *et al*. VisDrone-DET2019: The vision meets drone object detection in image challenge results[C]. 2019 IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea (South), 2019: 213–226. doi: 10.1109/ICCVW.2019.00030.

[38] University of Saskatchewan. Kaggle competition: Global wheat detection[EB/OL]. https://www.kaggle.com/c/global-wheat-detection, 2020.

[39] HUANG Zhanchao, WANG Jianlin, FU Xuesong, *et al*. DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection[J]. *Information Sciences*, 2020, 522: 241–258. doi: 10.1016/j.ins.2020.02.067.

[40] HUANG Xin, WANG Xinxin, LV Wenyu, *et al*. PP-YOLOv2: A practical object detector[J]. arXiv preprint arXiv: 2104.10419, 2021.

[41] DING Jian, XUE Nan, XIA Guisong, *et al*. Object detection in aerial images: A large-scale benchmark and challenges[J]. arXiv preprint arXiv: 2102.12219, 2021.

[42] TEKIN B, SINHA S N, and FUA P. Real-time seamless single shot 6D object pose prediction[C]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, USA, 2018: 292–301. doi: 10.1109/CVPR.2018.00038.

[43] SIMON M, AMENDE K, KRAUS A, *et al*. Complexer-YOLO: Real-time 3D object detection and tracking on semantic point clouds[C]. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Long Beach, USA, 2019: 1190–1199. doi: 10.1109/CVPRW.2019.00158.

[44] TAKAHASHI M, JI Y, UMEDA K, *et al*. Expandable YOLO: 3D object detection from RGB-D images[C]. 2020 21st International Conference on Research and Education in Mechatronics (REM), Cracow, Poland, 2020: 1–5. doi: 10.1109/REM49740.2020.9313886.

[45] DING Caiwen, WANG Shuo, LIU Ning, *et al*. REQ-YOLO: A resource-aware, efficient quantization framework for object detection on FPGAs[C]. 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, Seaside, USA, 2019: 33–42. doi: 10.1145/3289602.3293904.

[46] LEE Y, LEE C, LEE H J, *et al*. Fast detection of objects using a YOLOv3 network for a vending machine[C]. 2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS), Hsinchu, China, 2019: 132–136. doi: 10.1109/aicas.2019.8771517.

[47] AZIMI S M. ShuffleDet: Real-time vehicle detection network in on-board embedded UAV imagery[C]. 2018 European Conference on Computer Vision Workshops, Munich, Germany, 2019: 88–99. doi: 10.1007/978-3-030-11012-3_7.

[48] TIJTGAT N, VAN RANST W, VOLCKAERT B, *et al*. Embedded real-time object detection for a UAV warning system[C]. 2017 IEEE International Conference on Computer Vision Workshops, Venice, Italy, 2017: 2110–2118. doi: 10.1109/ICCVW.2017.247.

[49] ZHANG Pengyi, ZHONG Yunxin, and LI Xiaoqiong. SlimYOLOv3: Narrower, faster and better for real-time UAV applications[C]. 2019 IEEE/CVF International Conference on Computer Vision Workshops, Seoul, Korea (South), 2019: 37–45. doi: 10.1109/ICCVW.2019.00011.

[50] HENDRY and CHEN R C. Automatic license plate recognition via sliding-window darknet-YOLO deep learning[J]. *Image and Vision Computing*, 2019, 87: 47–56. doi: 10.1016/j.imavis.2019.04.007.

[51] TU Renwei, ZHU Zhongjie, BAI Yongqiang, *et al*. Improved YOLO v3 network-based object detection for blind zones of heavy trucks[J]. *Journal of Electronic Imaging*, 2020, 29(5): 053002. doi: 10.1117/1.JEI.29.5.053002.

[52] YANG Shuo, ZHANG Junxing, BO Chunjuan, *et al*. Fast vehicle logo detection in complex scenes[J]. *Optics & Laser Technology*, 2019, 110: 196–201. doi: 10.1016/j.optlastec.2018.08.007.

[53] YANG Fan, YANG Deming, HE Zhiming, *et al*. Automobile fine-grained detection algorithm based on multi-improved YOLOv3 in smart streetlights[J]. *Algorithms*, 2020, 13(5): 114. doi: 10.3390/a13050114.

[54] LI Min, ZHANG Zhijie, LEI Liping, *et al*. Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of faster R-CNN, YOLO v3 and SSD[J]. *Sensors*, 2020, 20(17): 4938. doi: 10.3390/s20174938.

[55] WU Dihua, LV Shuaichao, JIANG Mei, *et al*. Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments[J]. *Computers and Electronics in Agriculture*, 2020, 178: 105742. doi: 10.1016/j.compag.2020.105742.

[56] XU Zhifeng, JIA Ruisheng, SUN Hongmei, *et al*. Light-YOLOv3: Fast method for detecting green mangoes in complex scenes using picking robots[J]. *Applied Intelligence*, 2020, 50(12): 4670–4687. doi: 10.1007/s10489-020-01818-w.

[57] SHARIF M, AMIN J, SIDDIQA A, *et al*. Recognition of different types of leukocytes using YOLOv2 and optimized bag-of-features[J]. *IEEE Access*, 2020, 8: 167448–167459. doi: 10.1109/access.2020.3021660.

[58] ZHUANG Zhemin, LIU Guobao, DING Wanli, *et al*. Cardiac VFM visualization and analysis based on YOLO deep learning model and modified 2D continuity equation[J]. *Computerized Medical Imaging and Graphics*, 2020, 82: 101732. doi: 10.1016/j.compmedimag.2020.101732.

[59] KYRKOU C. YOLOpeds: Efficient real-time single-shot pedestrian detection for smart camera applications[J]. *IET Computer Vision*, 2020, 14(7): 417–425. doi: 10.1049/iet-cvi.2019.0897.

[60] 赵斌, 王春平, 付强. 显著性背景感知的多尺度红外行人检测方法[J]. 电子与信息学报, 2020, 42(10): 2524–2532. doi: 10.11999/JEIT190761.
ZHAO Bin, WANG Chunping, and FU Qiang. Multi-scale pedestrian detection in infrared images with salient background-awareness[J]. *Journal of Electronics & Information Technology*, 2020, 42(10): 2524–2532. doi: 10.11999/JEIT190761.

[61] KRIŠTO M, IVASIC-KOS M, and POBAR M. Thermal object detection in difficult weather conditions using YOLO[J]. *IEEE Access*, 2020, 8: 125459–125476. doi: 10.1109/access.2020.3007481.

[62] LIU Peng, SONG Changlin, LI Junmin, *et al*. Detection of transmission line against external force damage based on improved YOLOv3[J]. *International Journal of Robotics and Automation*, 2020, 35(6): 460–468. doi: 10.2316/J.2020.206-0479..

[63] XIE Yiqun, CAI Jiannan, BHOJWANI R, *et al*. A locally-constrained YOLO framework for detecting small and densely-distributed building footprints[J]. *International Journal of Geographical Information Science*, 2020, 34(4): 777–801. doi: 10.1080/13658816.2019.1624761.

[64] LUO Yanyang, SHAO Yanhua, CHU Hongyu, *et al*. CNN-based blade tip vortex region detection in flow field[C]. SPIE 11373, Eleventh International Conference on Graphics and Image Processing (ICGIP 2019), Hangzhou, China, 2020: 113730P. doi: 10.1117/12.2557248.

邵延华：男，讲师，研究方向为计算机视觉.
张 铎：男，硕士生，研究方向为计算机视觉.
楚红雨：男，副研究员，研究方向为机器人技术.
张晓强：男，讲师，研究方向为合成孔径成像和计算机视觉.
饶云波：男，副教授，研究方向为虚拟现实、互联网和计算机视觉.

责任编辑：余 蓉