文章编号:1671-6833(2024)03-0038-08

基于注意力与多级特征融合的 YOLOv5 算法

王 瑜,毕 玉,石健彤,肖洪兵,孙 梅

(北京工商大学 计算机与人工智能学院,北京 100048)

摘 要:针对复杂场景下目标检测与识别精度较低的问题,提出了一种基于注意力与多级特征融合的 YOLOv5 目标检测与识别算法。该算法在传统 YOLOv5s 模型的主干网络中引入双空间方向的金字塔切分注意力机制,增强对特征空间和通道信息的学习能力,同时在瓶颈网络中采用多级特征融合结构,对不同分支的特征进行融合,增加特征的丰富性,提升应对复杂场景的能力。此外,利用 C3Ghost 模块和深度可分离卷积分别替换 C3 模块和普通卷积,降低网络参数量和复杂度。结果表明:与传统的 YOLOv5s 算法相比,所提算法在 VOC2007+2012 数据集的均值平均精度高达 85%,在智能零售柜商品识别数据集的均值平均精度高达 97.2%,表现出较好的性能。

关键词:深度学习; YOLOv5s; 目标检测; 多级特征融合; 注意力机制

中图分类号:TP391 **文献标志码**:A **doi**:10.13705/j.issn.1671-6833.2023.06.009

目标检测与识别是计算机视觉的一个重要研究方向,在实际生活中具有广泛的应用,例如智能零售柜中的商品检测与识别、智能移动端上的人脸检测与识别以及监控系统中的目标检测与识别等。然而,这些实际应用环境多为复杂场景,该类场景存在同类目标差异大、不同类目标相似度高以及目标不同程度遮挡等特点。因此,如何在复杂场景下保证检测与识别算法既能够解决上述难题,同时满足准确和快速的需求,具有十分重要的意义。

目标检测与识别技术的发展经历了传统方法和深度学习方法两个阶段,传统方法一般应用于简单场景下的目标检测与识别,而复杂场景更适合采用深度学习方法。深度学习方法一般分为两大类,即双阶段方法和单阶段方法[1-3]。

双阶段的目标检测与识别方法包括目标候选区域选择、卷积神经网络特征提取、候选区域分类和目标识别结果优化等步骤。2013 年,Girshick 等[4]提出区域卷积神经网络(region convolution neural network, R-CNN),初次使用深度学习方法进行目标检测与识别。2014 年,He 等[5]提出空间金字塔池化网络(spatial pyramid pooling networks, SPPNet),消除了网络对输入图像尺寸的限制,避免了卷积特征的重复计算。2015 年,Girshick[6]提出 Fast R-CNN 算法,对 R-CNN 和 SPPNet 进行了融合,使网络性能进一步提高。同年,Ren 等[7]又提出 Faster R-CNN 算法,引入候选区域生成网络(region proposal network, RPN),自动生成目标候选区域。2017 年,He 等[8]提出 Mask R-CNN 算法,解决了原图与特征图的特征位置不匹配问题。

单阶段的目标检测与识别方法是将目标检测与识别问题转换为回归问题,优化网络中候选区域的产生过程,直接在图像中预测目标的类别概率和位置坐标[9]。2016 年,Liu 等[10]引入多尺度识别技术,提出了单次多边框检测器(single shot multibox detector, SSD),可以一次完成定位和分类。Redmon 等[11]提出了 YOLOv1,运行速度快,但定位准确率较低。Redmon 等[12]对 YOLOv1 进行改进,提出 YOLO9000,提升了目标检测与识别的定位准确率和召回率。Lin 等[13]提出了 RetinaNet 算法,解决了正负样本之间不均衡的问题。2018 年,Redmon 等[14]在改进基础网络的同时,结合金字塔结构,提出了 YOLOv3 算法,以获取更多小目标的有效信息。2019 年,Zhao 等[15]提出了 M2Det 算法,用来解决目标尺度变化的问题。Tan 等[16]设计了一种多维度混合的模型放缩

收稿日期:2023-05-08;**修订日期**:2023-06-09

基金项目:北京市教委-市自然科学基金联合资助项目(KZ202110011015)

作者简介:王瑜(1977—),女,吉林长春人,北京工商大学教授,博士,博士生导师,主要从事图像处理、模式识别、计算机视觉研究,E-mail:wangyu@btbu.edu.cn。

引用本文:王瑜,毕玉,石健彤,等. 基于注意力与多级特征融合的 YOLOv5 算法[J]. 郑州大学学报(工学版),2024,45(3):38-45,95. (WANG Y, BI Y, SHI J T, et al. Object detection and recognition algorithm based on YOLOv5 and the fusion of attention and multistage features[J]. Journal of Zhengzhou University (Engineering Science),2024,45(3):38-45,95.)

方法，即 EfficientNet 算法，并在此基础上改进，提出了 EfficientDet 算法[17]。此外，在 YOLOv3 的基础上，YOLOv4 模型[18]和 YOLOv5 模型[19]在保持运行效率优势的同时提高了检测与识别的准确率，因此 YOLO 系列模型得到了广泛使用[20-22]。

与双阶段的方法相比，单阶段的目标检测与识别方法检测与识别速度更快，更能满足实际应用场景的需要。

为进一步提升模型应对复杂场景的能力，在保持运行速度的同时提升准确率，本文提出了一种基于注意力与多级特征融合（attention and multistage feature fusion，AMFF）的 YOLOv5 目标检测与识别算法，具体改进思想包括：①提出双空间方向的金字塔切分注意力（dual space directions pyramid split attention，DSD-PSA）机制，并将其添加在 YOLOv5s 的主干网络阶段，以便更好地提取图像空间方向的位置信息和通道重要性的加权信息，对无关信息加以抑制；②提出多级特征融合（multistage feature fusion，MFF）结构，并将其添加在模型的瓶颈网络阶段，获取上下文信息，提升模型的精度；③利用 C3Ghost 模块替换 C3 模块，并使用深度可分离卷积，大幅减少模型的参数量，降低模型权重所需的存储空间，提高检测与识别的速度，满足实际应用中对实时性的需求。

1　YOLOv5 算法介绍

YOLOv5 包括 4 种模型，分别为 YOLOv5s、YOLOv5m、YOLOv5l 以及 YOLOv5x。利用模型配置文件中的网络深度与网络宽度调节系数进行模型规模的选择，由于 YOLOv5s 模型的参数量最小，所以本文以 YOLOv5s 作为基线模型。模型分为输入端、主干网络、瓶颈网络和输出端。在输入端，该模型利用 Mosaic 方法[19]进行数据增强。在主干网络阶段，模型利用空间金字塔池化结构，融合各个尺度的特征信息。在瓶颈网络阶段，模型采用特征金字塔网络（feature pyramid networks，FPN），增加金字塔自注意力网络（pyramid attention network，PAN），进一步提高特征融合的能力。在输出端，模型采用非极大值抑制（non-maximum suppression，NMS）的方法，筛选保留最佳目标框。

2　算法优化

2.1　YOLOv5-AMFF 的整体网络架构

本文提出的 YOLOv5-AMFF 模型结构如图 1 所示，分为输入端、主干网络、瓶颈网络和输出端。相较于原始的 YOLOv5s 模型，本文模型在主干网络阶段加入 DSD-PSA 机制，以增强模型对图像空间信息和通道重要性加权信息的学习能力。在瓶颈网络阶段采用 MFF 结构，以增强模型对特征的融合能力，并利用 C3Ghost 模块替换 C3 模块，引入深度可分离卷积，以降低网络的参数量。

2.2　DSD-PSA 机制

在复杂场景下，目标的分布更加密集，需要提取更多的空间信息，以提高模型检测与识别的精度。本文提出的 DSD-PSA 机制总体结构如图 2 所示。DSD-PSA 机制通过级联方式将 X、Y 这 2 个空间方

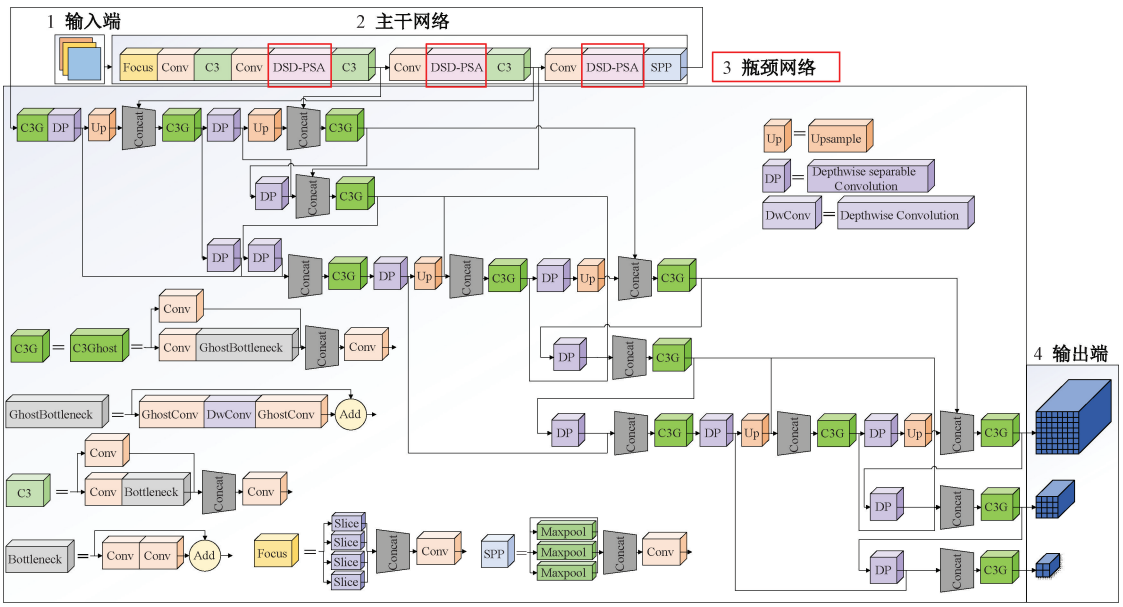


图 1　YOLOv5-AMFF 结构图

Figure 1　Structure diagram of the YOLOv5-AMFF

向的池化卷积模块与 PSA 机制[23]相结合。X、Y 池化卷积模块有利于提取 X、Y 空间方向上的图像特征,并对全局信息进行交互学习。PSA 机制可以有效融合不同尺度的上下文信息,并产生像素级的注意力。首先,输入张量进入 X、Y 池化卷积模块;然后,通过 SEWeight 模块,判断每组通道的重要性,以便获得不同尺度特征图的注意力权值;最后,进行 Softmax 运算,将软分配权重与原特征图进行乘积[23],得到输出特征图。

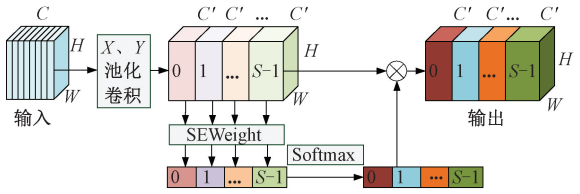


图 2 DSD-PSA 机制结构图

Figure 2 Structure diagram of the DSD-PSA mechanism

该机制的 X、Y 池化卷积模块如图 3 所示。输入数据为 $\boldsymbol{U}\in\mathbf{R}^{C\times H\times W}$,其中 C、H 和 W 分别表示特征图的通道数、高度和宽度。特征图利用池化核(H, 1)和(1, W)分别进行 X 和 Y 空间方向上的池化操作,得到特征图 $\boldsymbol{U}_w\in\mathbf{R}^{C\times 1\times W}$ 和 $\boldsymbol{U}_h\in\mathbf{R}^{C\times H\times 1}$,以获取精确位置信息的远程空间交互。$\boldsymbol{U}_w\in\mathbf{R}^{C\times 1\times W}$ 和 $\boldsymbol{U}_h\in\mathbf{R}^{C\times H\times 1}$ 扩展张量到 $C\times H\times W$,并与 $\boldsymbol{U}\in\mathbf{R}^{C\times H\times W}$ 进行乘积操作,得到特征图 $\boldsymbol{V}\in\mathbf{R}^{C\times H\times W}$。然后,通道数 C 切分为 S 组,得到特征图 $\boldsymbol{V}_i\in\mathbf{R}^{C'\times H\times W}$,$i=0,1,\cdots,S-1$。每组分别进行卷积核 $\boldsymbol{K}_0$ 为 1×1、$\boldsymbol{K}_1$ 为 5×5、$\boldsymbol{K}_2$ 为 9×9、$\boldsymbol{K}_3$ 为 13×13 的卷积操作,得到特征图 $\boldsymbol{F}_i\in\mathbf{R}^{C'\times H\times W}$,再进行 Concat 操作,输出特征图 $\boldsymbol{F}\in\mathbf{R}^{C\times H\times W}$,该步骤可以获取特征图不同尺度的感受野。

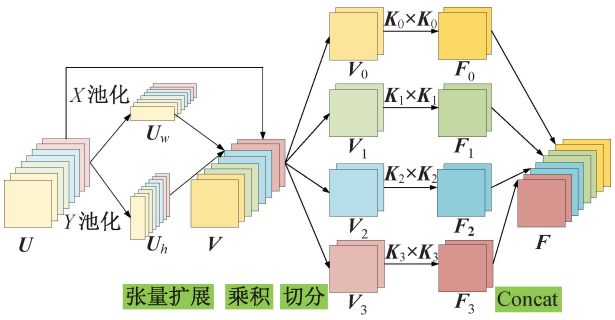


图 3 X、Y 池化卷积模块结构图

Figure 3 Structure diagram of the X and Y pooled convolution module

特征图 $\boldsymbol{F}_i\in\mathbf{R}^{C'\times H\times W}$ 经过 SEWeight 模块得到注意力权重 $\boldsymbol{Z}_i\in\mathbf{R}^{C'\times 1\times 1}$,其计算过程如式(1)所示:

$$\boldsymbol{Z}_i=\mathrm{SEWeight}(\boldsymbol{F}_i),i=0,1,\cdots,S-1。\quad(1)$$

SEWeight 模块允许网络有选择地对每个通道的重要性进行加权,从而产生更多的信息。

SEWeight 模块包括两个步骤。第 1 步为特征压缩,利用全局平均池化生成通道信息,得到特征图 $\boldsymbol{G}_C\in\mathbf{R}^{C\times 1\times 1}$,其计算过程为

$$\boldsymbol{G}_C=\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\boldsymbol{x}_C(i,j)。\quad(2)$$

式中:C、H 和 W 分别表示特征图的通道数、高度和宽度。

第 2 步为特征激励,用于自适应地重新校准信道关系。该步骤有效地组合通道间的线性信息,有利于高维和低维的信息交互。第 C 个通道的权重计算过程为

$$\boldsymbol{w}_C=\sigma(\boldsymbol{W}_1\delta(\boldsymbol{W}_0(\boldsymbol{G}_C)))。\quad(3)$$

式中:σ 和 δ 分别表示 Sigmoid 和 Relu 激活函数;$\boldsymbol{W}_0\in\mathbf{R}^{C\times\frac{C}{r}}$、$\boldsymbol{W}_1\in\mathbf{R}^{\frac{C}{r}\times C}$ 均表示全连接层,其中 r 为用来降维的超参数。

软分配权重 $\boldsymbol{att}_i$[23]的计算过程为

$$\boldsymbol{att}_i=\mathrm{Softmax}(\boldsymbol{Z}_i)=\frac{\exp(\boldsymbol{Z}_i)}{\sum_{i=0}^{S-1}\exp(\boldsymbol{Z}_i)}。\quad(4)$$

输出特征图 $\boldsymbol{Y}_i$ 的计算如式(5)所示:

$$\boldsymbol{Y}_i=\boldsymbol{F}_i\otimes\boldsymbol{att}_i,i=1,2,\cdots,S-1。\quad(5)$$

式中:⊗表示卷积。将 $\boldsymbol{Y}_i$ 进行 Concat 操作后,得到最终的输出特征图 $\boldsymbol{Y}$。

DSD-PSA 机制的伪代码如下所示。

输入:输入特征图 $\boldsymbol{U}$;

输出:输出特征图 $\boldsymbol{Y}$。

① $\boldsymbol{U}_w$ = self.pool_x($\boldsymbol{U}$)/*X 方向池化操作*/;

② $\boldsymbol{U}_h$ = self.pool_y($\boldsymbol{U}$)/*Y 方向池化操作*/;

③ $\boldsymbol{V}=\boldsymbol{U}\boldsymbol{U}_h$.expand_as($\boldsymbol{U}$)$\boldsymbol{U}_w$.expand_as($\boldsymbol{U}$)/*张量扩展,并与原特征图乘积*/;

④ $\boldsymbol{F}_0$ = self.Conv_0($\boldsymbol{V}$)/*1×1 分组卷积*/;

⑤ $\boldsymbol{F}_1$ = self.Conv_1($\boldsymbol{V}$)/*5×5 分组卷积*/;

⑥ $\boldsymbol{F}_2$ = self.Conv_2($\boldsymbol{V}$)/*9×9 分组卷积*/;

⑦ $\boldsymbol{F}_3$ = self.Conv_3($\boldsymbol{V}$)/*13×13 分组卷积*/;

⑧ $\boldsymbol{F}$ = torch.Concat($\boldsymbol{F}_0$, $\boldsymbol{F}_1$, $\boldsymbol{F}_2$, $\boldsymbol{F}_3$);

⑨ $\boldsymbol{F}_0\_SE$ = self.SEWeight($\boldsymbol{F}_0$);

⑩ $\boldsymbol{F}_1\_SE$ = self.SEWeight($\boldsymbol{F}_1$);

⑪ $\boldsymbol{F}_2\_SE$ = self.SEWeight($\boldsymbol{F}_2$);

⑫ $\boldsymbol{F}_3\_SE$ = self.SEWeight($\boldsymbol{F}_3$);

⑬ $\boldsymbol{F}\_SE$ = torch.Concat($\boldsymbol{F}_0\_SE$, $\boldsymbol{F}_1\_SE$, $\boldsymbol{F}_2\_SE$, $\boldsymbol{F}_3\_SE$)/*Concat 共 4 组特征图*/;

⑭ $\boldsymbol{att}$ = self.softmax($\boldsymbol{F}\_SE$)/*Softmax 操作*/;

⑮ $\boldsymbol{Y}$ = torch.Concat($\boldsymbol{F}_0\boldsymbol{att}_0$, $\boldsymbol{F}_1\boldsymbol{att}_1$, $\boldsymbol{F}_2\boldsymbol{att}_2$, $\boldsymbol{F}_3\boldsymbol{att}_3$)/*特征图与软分配权重相乘*/;

⑯ 结束。

2.3　MFF 结构

复杂场景下目标检测与识别的难点在于无法准确提取形状、颜色等差异较大的同类目标的特征,而易将相似度较高的不同类目标划分为同种类别。因此,如何提取更加全面、有效的信息是提高算法性能的关键。

深度学习的浅层特征中包含更多位置、细节信息,但是经过的卷积少,其语义信息较少,噪声更多。而深层特征包含更多的语义信息,但是分辨率较低,对细节的表达能力较差。因此,本文提出了 MFF 结构,可以有效融合深层特征与浅层特征,充分利用深层和浅层特征的优势,得到包含丰富语义和位置信息的特征,从而提升目标检测与识别的准确性。

MMF 结构如图 4 所示,具有三级结构,通过每一级结构中包含的 4 个 Concat 操作,充分融合上下文信息,在模型反复卷积得到深层信息时,以跳跃连接的方式融合浅层信息,保留更多高分辨率的细节信息,从而将提取得到的图像浅层、深层以及更深层的特征信息进行加强。

Concat 操作分为两种结构。第 1 种结构由 3 幅输入特征图叠加获得,包括 2 个前层的较浅特征与本层特征。第 2 种结构由 2 幅输入特征图叠加获得。两种结构的 Concat 操作采用了两类连接方式:一类为级内连接,即同级的前层特征与本层特征进行连接;另一类为跨级连接,即前级的特征与本级的特征进行连接,例如,三级结构跨级连接二级结构的特征、二级结构跨级连接一级结构的特征、一级结构跨级连接至主干网络中的浅层特征等,获得更多位置信息。这两类连接方式将深层信息与浅层信息进行有效融合,提升了模型检测与识别的精度。加入 MFF 结构之后,网络参数量大幅增加,因此,需要对网络参数进行调整。

MFF 结构可根据需求调整不同的系数,该结构的伪代码如下。

输入:设主干网络中的 Focus 模块是第 0 层,输入特征图是第 5 层 $\boldsymbol{M}_1$、第 8 层 $\boldsymbol{M}_2$、第 11 层 $\boldsymbol{M}_3$;

输出:输出特征图 $\boldsymbol{P}_1$、$\boldsymbol{P}_2$、$\boldsymbol{P}_3$。

① $\boldsymbol{x}_{13}$ = self. C3Ghost($\boldsymbol{M}_3$);
② $\boldsymbol{x}_{14}$ = self. DPConv($\boldsymbol{x}_{13}$)
③ $\boldsymbol{x}_{15}$ = self. Upsample($\boldsymbol{x}_{14}$);
④ $\boldsymbol{x}_{16}$ = torch. Concat($\boldsymbol{M}_2$, $\boldsymbol{x}_{15}$);
⑤ $\boldsymbol{x}_{17}$ = self. C3Ghost($\boldsymbol{x}_{16}$);
⑥ $\boldsymbol{x}_{18}$ = self. DPConv($\boldsymbol{x}_{17}$);
⑦ $\boldsymbol{x}_{19}$ = self. DPConv($\boldsymbol{x}_{17}$);
⑧ $\boldsymbol{x}_{20}$ = self. Upsample($\boldsymbol{x}_{18}$);
⑨ $\boldsymbol{x}_{21}$ = torch. Concat($\boldsymbol{x}_{20}$, $\boldsymbol{M}_1$);
⑩ $\boldsymbol{x}_{22}$ = self. C3Ghost($\boldsymbol{x}_{21}$);
⑪ $\boldsymbol{x}_{23}$ = self. DPConv($\boldsymbol{x}_{22}$);
⑫ $\boldsymbol{x}_{24}$ = torch. Concat($\boldsymbol{x}_{23}$, $\boldsymbol{x}_{18}$, $\boldsymbol{M}_2$);
⑬ $\boldsymbol{x}_{25}$ = self. C3Ghost($\boldsymbol{x}_{24}$);
⑭ $\boldsymbol{x}_{26}$ = self. DPConv($\boldsymbol{x}_{25}$);
⑮ $\boldsymbol{x}_{27}$ = torch. Concat($\boldsymbol{x}_{26}$, $\boldsymbol{x}_{19}$, $\boldsymbol{x}_{14}$);
⑯ $\boldsymbol{x}_{28}$ = self. C3Ghost($\boldsymbol{x}_{27}$);

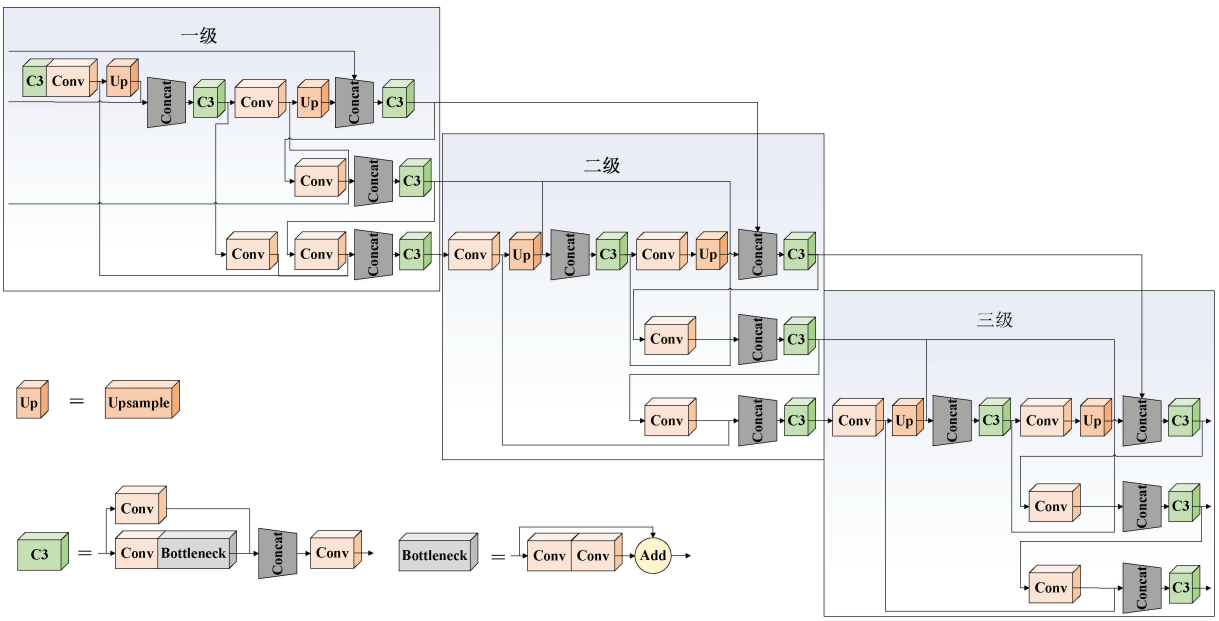


图 4　MFF 结构图

Figure 4　Structure diagram of the MFF

⑰ $\boldsymbol{x}_{29}$ = self. DPConv($\boldsymbol{x}_{28}$);
⑱ $\boldsymbol{x}_{30}$ = self. Upsample($\boldsymbol{x}_{29}$);
⑲ $\boldsymbol{x}_{31}$ = torch. Concat($\boldsymbol{x}_{30}$, $\boldsymbol{x}_{25}$);
⑳ $\boldsymbol{x}_{32}$ = self. C3Ghost($\boldsymbol{x}_{31}$);
㉑ $\boldsymbol{x}_{33}$ = self. DPConv($\boldsymbol{x}_{32}$);
㉒ $\boldsymbol{x}_{34}$ = self. Upsample($\boldsymbol{x}_{33}$);
㉓ $\boldsymbol{x}_{35}$ = torch. Concat($\boldsymbol{x}_{34}$, $\boldsymbol{x}_{22}$);
㉔ $\boldsymbol{x}_{36}$ = self. C3Ghost($\boldsymbol{x}_{35}$);
㉕ $\boldsymbol{x}_{37}$ = self. DPConv($\boldsymbol{x}_{36}$);
㉖ $\boldsymbol{x}_{38}$ = torch. Concat($\boldsymbol{x}_{37}$, $\boldsymbol{x}_{32}$, $\boldsymbol{x}_{25}$);
㉗ $\boldsymbol{x}_{39}$ = self. C3Ghost($\boldsymbol{x}_{38}$);
㉘ $\boldsymbol{x}_{40}$ = self. DPConv($\boldsymbol{x}_{39}$);
㉙ $\boldsymbol{x}_{41}$ = torch. Concat($\boldsymbol{x}_{40}$, $\boldsymbol{x}_{29}$);
㉚ $\boldsymbol{x}_{42}$ = self. C3Ghost($\boldsymbol{x}_{41}$);
㉛ $\boldsymbol{x}_{43}$ = self. DPConv($\boldsymbol{x}_{42}$);
㉜ $\boldsymbol{x}_{44}$ = self. Upsample($\boldsymbol{x}_{43}$);
㉝ $\boldsymbol{x}_{45}$ = torch. Concat($\boldsymbol{x}_{44}$, $\boldsymbol{x}_{39}$);
㉞ $\boldsymbol{x}_{46}$ = self. C3Ghost($\boldsymbol{x}_{45}$);
㉟ $\boldsymbol{x}_{47}$ = self. DPConv($\boldsymbol{x}_{46}$);
㊱ $\boldsymbol{x}_{48}$ = self. Upsample($\boldsymbol{x}_{47}$);
㊲ $\boldsymbol{x}_{49}$ = torch. Concat($\boldsymbol{x}_{48}$, $\boldsymbol{x}_{36}$);
㊳ $\boldsymbol{x}_{50}$ = self. C3Ghost($\boldsymbol{x}_{49}$);
㊴ $\boldsymbol{x}_{51}$ = self. DPConv($\boldsymbol{x}_{50}$);
㊵ $\boldsymbol{x}_{52}$ = torch. Concat($\boldsymbol{x}_{51}$, $\boldsymbol{x}_{46}$, $\boldsymbol{x}_{39}$);
㊶ $\boldsymbol{x}_{53}$ = self. C3Ghost($\boldsymbol{x}_{52}$);
㊷ $\boldsymbol{x}_{54}$ = self. DPConv($\boldsymbol{x}_{53}$);
㊸ $\boldsymbol{x}_{55}$ = torch. Concat($\boldsymbol{x}_{54}$, $\boldsymbol{x}_{43}$);
㊹ $\boldsymbol{x}_{56}$ = self. C3Ghost($\boldsymbol{x}_{55}$);
㊺ $\boldsymbol{P}_3 = \boldsymbol{x}_{56}$, $\boldsymbol{P}_2 = \boldsymbol{x}_{53}$, $\boldsymbol{P}_1 = \boldsymbol{x}_{50}$;
㊻ 结束。

2.4 网络参数调整

由于模型的网络参数量成倍增加,且单幅图像的处理时间变长,为了进一步优化网络,将瓶颈网络中的 C3 模块替换为 C3Ghost 模块,并引入深度可分离卷积。

C3Ghost 模块的核心部分为 GhostBottleneck 模块,GhostBottleneck 模块依次进行 GhostConv[24]、逐通道卷积(depthwise convolution)以及第 2 个 GhostConv 操作,再与原特征图进行 Add 操作。

GhostConv 使用计算量更低的操作,去除冗余的特征图,降低参数和运行内存,其结构如图 5 所示。该结构输入特征图尺寸为 $c\times h\times w$,利用 1×1 卷积将特征图通道数 c 压缩至 o,得到尺寸为 $o\times h'\times w'$的基础特征图。每个基础特征对应 1 个基础特征和 $s-1$ 个 Ghost 特征。Ghost 特征是由基础特征进行线性操作 $\Phi_i, i=1,2,\cdots,k$,即逐通道卷积得到的。最后将每个基础特征对应的 s 个特征在通道维度上进行融合,输出特征 $p\times h'\times w'$,其中 $p=s\times o$。普通卷积与 GhostConv 参数量比 r 的计算过程为

$$r=\frac{p\times m\times m\times c}{\frac{p}{s}\times m\times m\times c+(s-1)\times\frac{p}{s}\times d\times d}\approx\frac{s\times c}{s+c-1}\approx s。\tag{6}$$

式中:$m\times m$ 和 c 分别表示普通卷积中卷积核的尺寸和通道数;p 为输出特征图的通道数;d 为线性操作的卷积核,令 $d\approx m$。

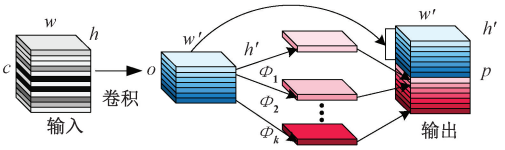


图 5 GhostConv 结构图

Figure 5 Structure diagram of the GhostConv

2.5 YOLOv5-AMFF 算法流程

本文提出的 YOLOv5-AMFF 算法步骤:①将数据集划分为训练集、验证集以及测试集,并将所有数据调整为 VOC 格式;②利用 Mosaic 数据增强方法,增强训练集数据;③设计 YOLOv5-AMFF 目标检测与识别模型,模型包括 4 部分,分别是输入端、主干网络、瓶颈网络以及输出端;④设置训练模型的相关参数;⑤用测试集数据验证模型的性能,并采用客观评测方法评估模型。

3 实验结果与分析

本文实验的硬件条件为 Intel(R)Xeon(R) CPU E5-2620 v3@2.40 GHz、Nvidia TITAN X GPU 和 32 GB 内存。运行环境为 Torch1.7.1+cu101、Torchvision 0.8.2+cu101 和 Python3.7。

3.1 数据集与数据增强

VOC2007+2012 数据集为目标检测与识别算法评测时常用的数据集之一,其场景复杂,单幅图像中同类目标差异大,包含训练集 16 385 幅图像,验证集 166 幅图像,测试集 4 952 幅图像。

为了进一步验证提出模型在其他常见复杂场景的适用性,采用阿里天池的智能零售柜商品识别数据集,并结合部分自己收集的商品图像。该数据集场景复杂,商品之间相互遮挡,且不同类别商品的相似度高,包含 113 类商品,训练集 4 495 幅图像,验证集 1 284 幅图像,测试集 643 幅图像。

数据增强部分首先利用 Mosaic 数据增强方法,将 4 幅图像进行随机裁剪,再拼接为一幅图,然后粘

贴部分标签,进行随机偏移,进一步增强输入数据的场景复杂度。

3.2　评价指标

本文实验使用精度(*Precision*)、召回率(*Recall*)、均值平均精度 *mAP*(mean average precision)、*F*1 分数(*F*1*Score*)以及单幅图像处理时间 *t*,来验证提出算法的有效性和可行性。

Precision 是精确性的度量,表示所有检测出的目标中检测正确的概率,计算过程为

$$Precision = \frac{TP}{TP + FP}。\tag{7}$$

式中:真阳性 *TP* 表示样例是真实正例,同时模型对其的预测结果也是正例的数量;真阴性 *TN* 表示样例是真实负例,同时模型对其的预测结果也是负例的数量;假阳性 *FP* 表示样例是真实负例,同时模型对其的预测结果却为正例的数量;假阴性 *FN* 表示样例是真实正例,同时模型对其的预测结果却为负例的数量。

Recall 是覆盖面的度量,表示所有的正样本中正确识别的概率,计算过程为

$$Recall = \frac{TP}{TP + FN}。\tag{8}$$

平均精度 *AP*(average precision)表示 *Precision* 和 *Recall* 所形成的曲线的面积。均值平均精度 *mAP* 是所有目标检测类别 *AP* 的平均值,@ 后面的数表示 *IoU* 阈值,例如,*mAP*@ 0.5 表示 *IoU* 阈值大于 0.5 的 *mAP*,*mAP*@ 0.5:0.95 表示不同 *IoU* 阈值(从 0.5 到 0.95,步长 0.05)上的 *mAP*。

Precision 和 *Recall* 通常是一对矛盾的性能度量指标。一般来说,*Precision* 越高时,*Recall* 往往越低。*F*1*Score* 是精确率和召回率的调和平均数,最大为 1,表示最好,最小为 0,表示最差,计算式为

$$F1Score = \frac{2 \times Precision \times Recall}{Precision + Recall}。\tag{9}$$

3.3　模型训练

本文模型训练的初始学习率为 0.01,最终学习率为 0.002,设置总迭代次数为 500,采用早停法,即在验证集损失不再减小的时候停止训练。YOLOv5-AMFF 模型在 VOC 数据集和智能零售柜商品识别数据集的损失函数没有剧烈震荡,逐渐趋于平稳,模型训练效果很好。

3.4　实验结果和分析

为了验证本文算法中添加模块的有效性,在 VOC 数据集上对 YOLOv5s 模型进行消融实验,结果如表 3 所示。

从表 3 可以看出,在 YOLOv5s 模型的特征提取部分添加 DSD-PSA 机制后,由于提取的特征图包含了更加全面的空间和通道信息,因此,*mAP* 提升了 1.1 百分点。在瓶颈网络部分采用 MFF 结构后,本文算法对图像的特征融合能力有显著提升,评价指标 *mAP* 提升了 2.6 百分点,*F*1*Score* 提升了 0.02,但是网络参数增至 YOLOv5s 的 2.05 倍。在添加 MFF 模块的基础上,利用 C3Ghost 模块和深度可分离卷积优化 MFF 结构,模型参数量降低了 45%,模型大小降低了 44%,同时 *mAP* 提升了 1.5 百分点,*F*1*Score* 提升了 0.02,证明了 C3Ghost 模块和深度可分离卷积的有效性。以此为基础,在特征提取阶段添加 DSD-PSA 机制后,模型 *F*1*Score* 不变,*mAP* 提升了 0.8 百分点。从实验结果看,本文算法在复杂场景下目标的检测与识别的精度提升,表明 DSD-PSA 机制、MFF 结构和网络降参结构的改进十分有效。

为了验证本文算法检测与识别的效果,在 VOC 数据集和智能零售柜商品识别数据集对 YOLOv5-AMFF、YOLOv4、YOLOv5s、YOLOv5m、SSD(VGG)以及 Faster R-CNN 模型进行了多元对比分析,具体结果如表 4 所示。

由表 4 可知,在 VOC 数据集上,本文算法的 *mAP*@ 0.5:0.95 高达 66.7%,比基线模型 YOLOv5s 提升了 9.4 百分点,并且高于 YOLOv4,YOLOv5m、SSD 和 Faster R-CNN。本文算法的 *mAP*@ 0.5 比 YOLOv4 提升了 5 百分点,比基线模型 YOLOv5s 提高了

表 3　不同改进的性能对比

Table 3　Performance comparison of different improvements

DSD-PSA 机制	MFF 结构	网络降参结构	*mAP*@ 0.5/%	*mAP*@ 0.5:0.95/%	*F*1*Score*	*Recall*/%	参数量	数据大小/MB	*t*/ms
			80.1	57.3	0.76	74.3	7 105 153	13.8	14.7
√			81.2	59.7	0.77	76.0	7 572 253	14.7	14.9
	√		82.7	62.6	0.78	74.5	14 532 121	28.2	16.3
	√	√	84.2	66.4	0.80	78.7	7 925 833	15.8	15.0
√	√	√	85.0	66.7	0.80	80.3	10 160 129	19.9	15.2

注:√表示在基线模型 YOLOv5s 中引入选中部分。

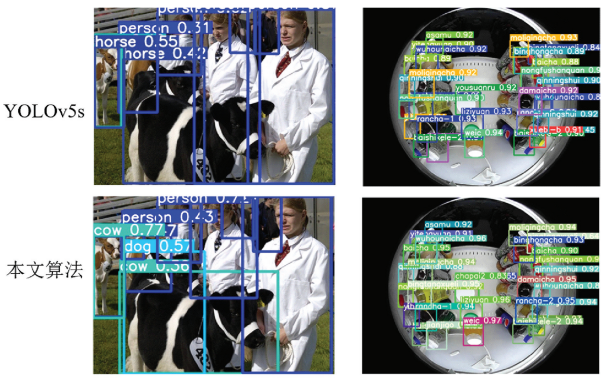
表4 不同目标检测与识别算法的性能对比

Table 4 Performance comparison of different object detection and recognition algorithms

方法	数据集	*mAP*@0.5/%	*mAP*@0.5:0.95/%	*F*1*Score*	*Recall*/%	参数量	数据大小/MB	*t*/ms
YOLOv4	VOC2007+2012	80.0	49.2	0.79	89.7	64 040 001	245.0	39.2
	智能零售柜商品识别数据集	96.7	81.1	0.94	95.1	64 487 874	246.0	41.3
YOLOv5s	VOC2007+2012	80.1	57.3	0.76	74.3	7 105 153	13.8	14.7
	智能零售柜商品识别数据集	96.9	89.1	0.95	95.6	7 552 062	14.6	15.3
YOLOv5m	VOC2007+2012	85.5	66.6	0.78	80.9	21 114 417	40.6	18.3
	智能零售柜商品识别数据集	97.4	91.3	0.96	96.0	21 490 230	43.3	19.8
SSD	VOC2007+2012	78.5	47.1	0.77	87.6	2 285 486	100.0	19.7
	智能零售柜商品识别数据集	95.3	79.1	0.95	95.8	26 732 395	101.0	20.8
Faster R-CNN	VOC2007+2012	77.8	46.2	0.76	86.8	28 786 578	108.0	83.3
	智能零售柜商品识别数据集	94.6	78.3	0.94	94.9	28 998 090	109.0	84.5
本文算法	VOC2007+2012	**85.0**	**66.7**	**0.80**	**80.3**	**10 160 129**	**19.9**	**15.2**
	智能零售柜商品识别数据集	**97.2**	**91.0**	**0.96**	**96.8**	**10 410 950**	**20.8**	**16.7**

4.9百分点,比SSD提升了6.5百分点,比Faster R-CNN提升7.2百分点。在模型参数量、权重存储空间和单幅图像处理时间3项指标中,本文算法仅大于YOLOv5s。本文算法的*F*1*Score*比基线模型YOLOv5s提升了0.04,*Recall*提升了6百分点。在智能零售柜商品识别数据集上,本文算法的评价指标*mAP*@0.5、*mAP*@0.5:0.95、*F*1*Score*和*Recall*均优于基线模型YOLOv5s、YOLOv4、SSD和Faster R-CNN,虽然模型精度略低于YOLOv5m,但是*Recall*提升了0.8百分点,模型大小仅为其48%,单幅图像的处理时间缩短了3.1 ms。本文算法单幅图像的处理时间为16.7 ms,明显优于YOLOV4、YOLOV5m、SSD和Faster R-CNN。

为了更好地验证本文算法对复杂场景下目标的检测与识别性能,在VOC数据集和智能零售柜商品识别数据集上进行了可视化实验,结果如图6所示。



(a) VOC2007+2012数据集 (b) 智能零售柜商品识别数据集

图6 算法检测与识别结果

Figure 6 Algorithm detection and identification results

从图6可以看出,在面对相似物体干扰、遮挡等复杂场景时,本文算法比基线模型YOLOv5s检测与识别出了更多的目标,且精度更高,验证了本文算法的优良性能。

4 结论

为了更准确地对复杂场景下的目标进行检测与识别,本文在传统YOLOv5s模型的主干网络部分加入了DSD-PSA机制,获得了双空间方向的信息交互,加强了通道信息的学习,得到了更加鲁棒的特征。在瓶颈网络部分,采用MFF结构,提高了模型的精度,同时使用C3Ghost模块和深度可分离卷积进行网络降参,降低了权重所需的存储空间和单幅图像的处理时间。在VOC2007+2012和智能零售柜商品识别数据集的相关实验验证了所提方法的有效性和可行性。未来工作将聚焦于研究如何在无数据标签的情况下将本文算法拓展到更多复杂场景中,提升模型的泛化能力。

参考文献:

[1] 李柯泉,陈燕,刘佳晨,等. 基于深度学习的目标检测算法综述[J]. 计算机工程,2022,48(7):1-12.
LI K Q, CHEN Y, LIU J C, et al. Survey of deep learning-based object detection algorithms[J]. Computer Engineering, 2022, 48(7): 1-12.

[2] 包晓敏,王思琪. 基于深度学习的目标检测算法综述[J]. 传感器与微系统,2022,41(4):5-9.
BAO X M, WANG S Q. Survey of object detection algorithm based on deep learning[J]. Transducer and Microsystem Technologies, 2022, 41(4): 5-9.

[3] 赵永强,饶元,董世鹏,等. 深度学习目标检测方法综述[J]. 中国图象图形学报,2020,25(4):

629-654.
ZHAO Y Q, RAO Y, DONG S P, et al. Survey on deep learning object detection [J]. Journal of Image and Graphics, 2020, 25(4): 629-654.

[4] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]// 2014 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway: IEEE, 2014: 580-587.

[5] HE K M, ZHANG X Y, REN S Q, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904-1916.

[6] GIRSHICK R. Fast R-CNN[C]//2015 IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE, 2016: 1440-1448.

[7] REN S Q, HE K M, GIRSHICK R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[C]//IEEE Transactions on Pattern Analysis and Machine Intelligence. Piscataway: IEEE, 2016: 1137-1149.

[8] HE K M, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN[C]// 2017 IEEE International Conference on Computer Vision (ICCV). Piscataway: IEEE, 2017: 2980-2988.

[9] CHEN L K, YE F Y, RUAN Y D, et al. An algorithm for highway vehicle detection based on convolutional neural network[J]. EURASIP Journal on Image and Video Processing, 2018, 2018(1): 1-7.

[10] LIU W, ANGUELOV D, ERHAN D, et al. SSD: single shot multibox detector [C]// European Conference on Computer Vision. Cham: Springer, 2016: 21-37.

[11] 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 (CVPR). Piscataway: IEEE, 2016: 779-788.

[12] REDMON J, FARHADI A. YOLO9000: better, faster, stronger[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2017: 6517-6525.

[13] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 318-327.

[14] REDMON J, FARHADI A. YOLOv3: an incremental improvement [EB/OL]. (2018-04-08)[2022-12-23]. https://arxiv.org/pdf/1804.02767v1.

[15] ZHAO Q J, SHENG T, WANG Y T, et al. M2Det: a single-shot object detector based on multi-level feature pyramid network[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33(1): 9259-9266.

[16] TAN M X, LE Q V. EfficientNet: rethinking model scaling for convolutional neural networks [EB/OL]. (2020-09-11)[2022-12-23]. https://arxiv.org/pdf/1905.11946v5.

[17] TAN M X, PANG R M, LE Q V. EfficientDet: scalable and efficient object detection[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 10778-10787.

[18] BOCHKOVSKIY A, WANG C Y, LIAO H M. YOLOv4: optimal speed and accuracy of object detection [EB/OL]. (2020-04-23)[2022-12-23]. https://arxiv.org/abs/2004.10934.

[19] JOCHER G. YOLOv5 [EB/OL]. (2020-06-17)[2022-12-23]. https://github.com/ultralytics/YOLOv5.

[20] ZHANG C X, KANG F, WANG Y X. An improved apple object detection method based on lightweight YOLOv4 in complex backgrounds[J]. Remote Sensing, 2022, 14(17): 4150.

[21] HONG W W, MA Z H, YE B L, et al. Detection of green asparagus in complex environments based on the improved YOLOv5 algorithm[J]. Sensors, 2023, 23(3): 1562.

[22] 贾云飞, 郑红木, 刘闪亮. 基于 YOLOv5s 的金属制品表面缺陷的轻量化算法研究[J]. 郑州大学学报(工学版), 2022, 43(5): 31-38.
JIA Y F, ZHENG H M, LIU S L. Lightweight surface defect detection method of metal products based on YOLOv5s[J]. Journal of Zhengzhou University (Engineering Science), 2022, 43(5): 31-38.

[23] ZHANG H, ZU K K, LU J, et al. EPSANet: an efficient pyramid squeeze attention block on convolutional neural network[C]// Asian Conference on Computer Vision. Cham: Springer, 2023: 541-557.

[24] HAN K, WANG Y H, TIAN Q, et al. GhostNet: more features from cheap operations[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway: IEEE, 2020: 1577-1586.

(下转第 95 页)

Geological Named Entity Recognition Based on MacBERT and R-Drop

LIU Xin[1], XU Hongzhen[1,2], LIU Aihua[2], DENG Dejun[1]

(1. School of Information Engineering, East China University of Technology, Nanchang 330013, China; 2. School of Software, East China University of Technology, Nanchang 330013, China)

Abstract: The commonly used deep learning methods based on BERT pre-trained model in geological named entity recognition were character-based approaches, and could not utilize word-level information. Additionally, the dropout mechanism in neural networks might cause inconsistency between the training and inference stage. To address this issue, a geological named entity recognition model MBCR based on MacBERT and R-Drop was proposed. Firstly, MacBERT was used to learn text feature representations, which could fully utilize character and word information. Then, BiGRU was employed to encode context features, effectively extracting complete semantic information. Subsequently, CRF was adopted to capture dependencies between labels and generate the optimal label sequence. Moreover, R-Drop was introduced during the training process to further enhance the model's generalization capabilities. Compared with BiLSTM-CRF, BERT-BiLSTM-CRF, and other models, the proposed MBCR model improved the *F*1-score on the NERdata dataset by 2.08-4.62 percentage points and on the Boson dataset by 1.26-17.54 percentage points.

Keywords: named entity recognition; geology; MacBERT; BiGRU; R-Drop

---

(上接第 45 页)

Object Detection and Recognition Algorithm Based on YOLOv5 and the Fusion of Attention and Multistage Features

WANG Yu, BI Yu, SHI Jiantong, XIAO Hongbing, SUN Mei

(School of Computer and Artificial Intelligence, Beijing Technology and Business University, Beijing 100048, China)

Abstract: To tackle the problem of low accuracy of detection and recognition for object in complex scenes, YOLOv5 object detection and recognition algorithm based on attention and multistage feature fusion(AMFF) was proposed in this study. The main ideas included adding the proposed dual space directions pyramid split attention (DSD-PSA) mechanism to the backbone network of the traditional YOLOv5s model to enhance the learning of the feature map space and channel information, adopting multistage feature fusion(MFF) structure in the bottleneck network to fuse the features of different branches, increasing richness of the feature and improving the ability to cope with complex scenes. In addition, C3Ghost module and depthwise separable convolution were used to replace C3 module and common convolution to reduce the number of parameters and the complexity of network. Compared with the traditional YOLOv5s algorithm, the mean average accuracy of the proposed algorithm in the VOC2007+2012 data set reached 85%, and the mean average accuracy of the smart retail cabinet commodity identification data set reached 97.2%, which verified the effectiveness and feasibility of the proposed algorithm.

Keywords: deep learning; YOLOv5s; object detection; multistage feature fusion; attention mechanism