控制与决策
Control and Decision

基于深度学习的复杂背景下目标检测

王红梅, 王晓鸽, 王晓燕

引用本文:
王红梅,王晓鸽,王晓燕. 基于深度学习的复杂背景下目标检测[J]. 控制与决策, 2022, 37(12): 3115−3121.

在线阅读 View online: https://doi.org/10.13195/j.kzyjc.2021.0686

基于深度学习的复杂背景下目标检测

王红梅[1†], 王晓鸽[2], 王晓燕[1]

(1. 西北工业大学 航天学院，西安 710072；2. 航空工业西安航空计算技术研究所，西安 710065)

摘 要: 目标检测是计算机视觉领域的重要研究方向. 传统的目标检测方法在特征设计上花费了大量时间,且手工设计的特征对于目标多样性的问题并没有好的鲁棒性,深度学习技术逐渐成为近年来计算机视觉领域的突破口. 为此,对现有的基础神经网络进行研究,采用经典卷积神经网络VGGNet作为基础网络,添加部分深层网络,结合SSD (single shot multibox detector)算法构建网络框架. 针对模型训练中出现的正负样本不均衡问题,根据困难样本挖掘原理,在原有的损失函数中引入调制因子,将背景部分视为简单样本,减小背景损失在置信损失中的占比,使得模型收敛更快速,模型训练更充分,从而提高复杂背景下的目标检测精度. 同时,通过构建特征金字塔和融合多层特征图的方式,实现对低层特征图的语义信息融合增强,以提高对小目标检测的精度,从而提高整体的检测精度. 仿真实验结果表明,所提出的目标检测算法(feature fusion based SSD, FF-SSD)在复杂背景下对各种目标均可取得较高的检测精度.

关键词: 目标检测；深度学习；SSD算法；复杂背景；困难样本；特征融合

中图分类号: TP391 **文献标志码:** A

DOI: 10.13195/j.kzyjc.2021.0686

引用格式: 王红梅,王晓鸽,王晓燕. 基于深度学习的复杂背景下目标检测[J]. 控制与决策, 2022, 37(12): 3115-3121.

Target detection under complex background based on deep learning

WANG Hong-mei[1†], *WANG Xiao-ge*[2], *WANG Xiao-yan*[1]

(1. School of Astronautics，Northwestern Polytechnical University，Xi'an 710072，China；2. AVIC Computing Technique Research Institute，Xi'an 710065，China)

Abstract: Object detection is an important research in the field of computer vision. Traditional target detection methods spend a lot of time on feature extraction, and manual features are not robust to the problem of diverse targets. Deep learning technology has gradually become a breakthrough in computer vision in recent years. By using the classical convolutional neural network VGGNet as a basic network, a network framework for target detection is built by adding some deep networks and combining with the SSD (single shot multi-box detector) algorithm, and an algorithm of feature fusion based SSD (FF-SSD) is proposed. Aiming at the problem of sample imbalance during the training of the model, the original loss function is modified according to the principle of hard example mining. The background is regarded as a simple sample and a modulation factor is introduced to reduce the proportion of background loss to the confidence loss, which makes the model be trained more fully and converge faster, and the target detection accuracy under the complex background is promoted as a result. Meanwhile, for poor detection effect to small targets of the SSD algorithm, the feature pyramid is constructed according to the feature maps extracted from each convolutional layer. Appropriate feature maps are selected and fused to form a new feature map for the prediction. The semantic information fusion is strengthened to enhance the detection accuracy of small targets in order to improve the overall detection accuracy. Experimental results show that the proposed target detection algorithm can achieve high detection accuracy for all kinds of targets in the complex background.

Keywords: target detection；deep learning；SSD algorithm；complex background；hard example；feature fusion

0 引 言

复杂背景下的目标检测是计算机视觉领域中一个十分重要的课题. 传统的目标检测方法面临以下两个问题:一是基于滑动窗口的区域选择策略容易产生窗口冗余;二是手工设计的特征对于目标多样性的变化并没有好的鲁棒性. 因此,基于深度学习的

收稿日期: 2021-04-21；**录用日期:** 2021-07-30.

基金项目: 国家自然科学基金项目(61771400)；陕西省重点研发计划项目(2020GY-014).

责任编委: 易建强.

†通讯作者. E-mail: haipw@nwpu.edu.cn.

目标检测方法开始受到人们的广泛关注. 深度学习方法能克服传统人工选取特征的缺点,自适应地学习表征目标的最佳特征,且抗干扰性能优异,可以有效提高目标识别的准确性和鲁棒性[1].

在深度学习目标检测模型中,具有代表性的是Girshick等[2]提出的一系列目标检测算法,其开山之作是R-CNN (region-convolutional neural network). 针对R-CNN训练时间过长的问题,Girshick[3]又提出了Fast R-CNN. 与R-CNN类似,Fast R-CNN依然采用selective search[4]生成候选区域,但是,与R-CNN提取出所有候选区域并使用SVM分类的方法不同,Fast R-CNN在整张图片上使用CNN,然后使用特征映射提取感兴趣区域(region of interest, RoI); 同时,利用反向传播网络进行分类和回归. 该方法不仅检测速度快,而且具有RoI集中层和全连接层,使得模型可求导,更容易训练. Ren等[5]又提出了Fast R-CNN的升级版本Faster R-CNN算法. Faster R-CNN是第一个真正意义上端到端的、准实时的深度学习目标检测算法. Faster R-CNN最大的创新点在于设计了候选区域生成网络(region proposal network, RPN),并设计了anchor机制. 从R-CNN到Fast R-CNN再到Faster R-CNN,候选区域生成、特征提取、候选目标确认以及边界框坐标回归被逐渐统一到同一个网络框架中.

同样是基于深度学习的目标检测方法,另一个发展分支是基于回归的目标检测方法. 华盛顿大学的Redmon等[6]提出了YOLO (you only look once)算法,其核心思想是使用整张图像作为网络输入,直接在输出层中输出边界框的位置及其所属的类别. 它的训练和检测均在单独的网络中进行,取得了较好的实时检测效果. YOLO方法舍弃了区域备选框阶段,加快了速度,但是定位和分类精度较低,尤其对小目标以及比较密集的目标群检测效果不够理想,召回率较低. 2017年,Redmon等[7]又提出了具有检测速度更快、检测精度更高和稳健性更强的YOLO v2. Ju等[8]则以YOLO v3[9]为基础,提出了一种改进的多尺度目标检测算法,PASCAL VOC和KITTI数据集上的实验结果均表明了该算法的有效性. 针对现有网络模型在实时性方面存在的不足,He等[10]提出了实时的目标检测模型TF-YOLO (tiny fast YOLO),仿真结果表明,该算法在多种设备上都可实现实时目标检测.

针对YOLO算法定位精度低的问题,Liu等[11]提出了SSD算法,该算法先根据锚点(anchor)提取备选框,然后再进行分类. SSD算法将YOLO的回归思想与Faster R-CNN的锚点机制相结合,一次即可完成网络训练,并且定位精度和分类精度相比YOLO都有大幅度提高. Bosquet等[12]提出了一种基于改进SSD模型的SAR (synthetic aperture radar)目标检测算法,仿真结果表明,该算法可以实现复杂背景下SAR目标的检测.

尽管SSD算法在特定数据集上已经取得了较高的准确率和较好的实时性,但是,该算法损失函数的设计未考虑正负样本不均衡所带来的问题,也存在因网络结构的缺陷而引起的小目标检测精度不高的问题. 针对模型中出现的正负样本失衡问题,本文基于困难样本挖掘原理,在损失函数中引入调制因子;针对因低层语义信息缺乏所导致的小目标检测结果欠佳的问题,采取多层特征融合的结构加以解决,即进行预测之前先进行浅层特征图的融合,增强其低层的语义信息,以期能够提高小目标的检测精度.

1 网络模型

1.1 SSD网络结构

本文检测模型以传统的基础网络VGG16 (visual geometry group)为基础,并添加深层卷积网络而构成. 前部分浅层网络采用卷积神经网络提取图像特征[10],包括输入层、卷积层和下采样层;后部分深层网络用卷积层代替原始的全连接层. 卷积层尺寸逐层递减,分类和定位回归在多尺度特征图上完成.

1.2 先验框设计

SSD网络能够识别多个物体,其核心是预测固定集合的类别分数和位置偏移,并使用应用于特征映射的小卷积滤波器的默认边界框. SSD借鉴了Faster R-CNN中anchor的理念[5],在特征图上通过卷积计算产生若干覆盖全图的候选区域,形成了先验框机制. 通过为每个单元设置尺度或者长宽比不同的先验框(预测的边界框是以这些先验框为基准的偏移系数),在一定程度上减少了训练难度. 对于每个单元的每个先验框,都输出一套独立的检测值,其对应的边界框由两部分描述: 第1部分是各个类别的置信度;第2部分是边界框的位置,包含4个值$(\mathrm{c}x, \mathrm{c}y, w, h)$,分别表示边界框的中心坐标以及宽和高. 由于先验框在模型训练之前就已确定,很难与真实的标注区域完全重合. 为解决此问题,SSD算法使用位置回归层来输出4个位置校正参数$(\mathrm{d}x, \mathrm{d}y, \mathrm{d}w, \mathrm{d}h)$. 先验框经过适当变换后,便能与真实的标注区域基本吻合.

1.3 引入调制因子的损失函数

损失函数用来计算模型预测值与真实值的不一致程度. 对于样本集合(x, y),本文采用多任务损失函数(multi-task loss function),可以在损失函数中完成

置信度判别和位置回归,两者加权求和,得到最终的损失函数[11],即

$$L(x,c,l,g)=\frac{1}{N}[L_{\rm conf}(x,c)+\lambda\, L_{\rm loc}(x,l,g)], \tag{1}$$

$$L_{\rm conf}(x,c)=-\Big(\sum_{i\in{\rm Pos}}^{N}x_{ij}^{P}\log(\hat{c}_i^{P})+\sum_{i\in{\rm Neg}}^{N}\log(\hat{c}_i^{N})\Big), \tag{2}$$

$$L_{\rm loc}(x,l,g)=\sum_{i\in{\rm Pos}}^{N}\sum_{m\in(cx,cy,w,h)}x_{ij}^{P}{\rm smooth}_{L1}(l_i^m-\hat{g}_j^m). \tag{3}$$

SSD算法在损失计算中,所有的候选框可以分为正样本和负样本两类,即在所有的先验框中,与每个标注框有最大重叠率的被视为正样本,或者是与标注框的重叠大于某一阈值时,被视为正样本,其他为负样本. 然而,在大多数图像中,目标所占的比例通常远小于背景所占比例. 尽管人们对阈值选择以及正负样本的判断标准有所放松,但是仍然存在正负样本不均衡的问题,也就是“类别失衡”[13]. 负样本过多时,容易造成负样本损失占比过大,进而导致正样本的误差损失被忽略,不利于模型的收敛.

为解决上述问题,本文首先将所有的待训练先验框进行排序,按照置信度得分情况从大到小排列,取前四分之一为正样本,其余为负样本,以减少负样本比重;然后,在原损失函数中引入调制因子,增加困难样本对参数的贡献值.

对于二分类的逻辑回归问题,损失函数为

$$L_{ce}=-\log(p). \tag{4}$$

其中:$p\in(0,1)$且$L_{ce}\in(0,+\infty)$,它代表预测框相对于标注框的置信度. p越大,L_{ce}越小,说明所训练的样本越容易,该样本越容易被正确识别,从而对损失值的贡献也越小;反之,p越小,L_{ce}越大,说明所训练的样本越困难,该样本越不容易被正确识别,从而对损失值的贡献也越大. 由于大量背景样本都是容易样本,这些样本叠加,损失值之和较大,就有可能造成“类别失衡”. 因此,可将$(1-p)$作为调制因子,加入到原有的交叉熵损失函数中. 原有的损失函数[14]变为

$$L'_{ce}=-(1-p)\log(p). \tag{5}$$

当样本为容易样本时,$(1-p)$越小,损失值会在原基础上进一步被降低,该分类越容易,被降低的程度也越大;相反,当样本为困难样本时,$(1-p)$越大,分类越困难,也有可能被误判,这时的调制因子相应较大,损失值在一定程度上会被保持. 如此便实现了困难样本的挖掘.

对于多分类问题,仍然采用交叉熵损失函数,区别在于p的取值不再由sigmod激活函数的输出值所定义,而是采用softmax函数来定义该变量,这时p为某一类的回归结果,即

$$\begin{cases}L'_{ce}=-(1-p)\log(p),\\ p={\rm softmax}(x)={\rm e}^{x_j}\Big/\sum\limits_{j}{\rm e}^{x_j}.\end{cases} \tag{6}$$

1.4　引入调制因子后前向传播函数和反向传播函数的推导

为了让引入调制因子后的损失函数能够替换原有的损失函数,下面进行损失函数的前向和反向传播推导. 损失函数的前向传播计算公式如下:

$$L'_{ce}=-(1-p)\log(p). \tag{7}$$

令t表示目标的类别($t\in[0,\,20]$),则损失函数为

$$L'_{ce}=-(1-p_t)\log(p_t), \tag{8}$$

$$\frac{\partial L'_{ce}(x,t)}{\partial x_i}=-\frac{\partial(-p_t)}{\partial x_i}\cdot\log(p_t)-(1-p_t)\cdot\frac{\partial\log(p_t)}{\partial x_i}=\log(p_t)\cdot\frac{\partial(p_t)}{\partial x_i}-(1-p_t)\cdot\frac{\partial\log(p_t)}{\partial x_i}. \tag{9}$$

下面计算$\dfrac{\partial p_t}{\partial x_i}$和$\dfrac{\partial\log(p_t)}{\partial x_i}$,有

$$\frac{\partial p_t}{\partial x_i}=\frac{\partial}{\partial x_i}\frac{{\rm e}^{x_i}}{\sum\limits_{j}{\rm e}^{x_j}}=\begin{cases}\dfrac{{\rm e}^{x_i}\sum\limits_{j}{\rm e}^{x_j}-{\rm e}^{x_t}\cdot{\rm e}^{x_t}}{\sum\limits_{j}{\rm e}^{x_j}\sum\limits_{j}{\rm e}^{x_j}}=p_t-p_t^2,\\ \dfrac{-{\rm e}^{x_t}\cdot{\rm e}^{x_t}}{\sum\limits_{j}{\rm e}^{x_j}\sum\limits_{j}{\rm e}^{x_j}}=-p_i\cdot p_t;\end{cases} \tag{10}$$

$$\frac{\partial\log(p_t)}{\partial x_i}=\frac{1}{p_t}\cdot\frac{\partial p_t}{\partial x_i}. \tag{11}$$

将式(10)代入(11),可得

$$\frac{\partial\log(p_t)}{\partial x_i}=\frac{1}{p_t}\cdot\frac{\partial p_t}{\partial x_i}=\frac{1}{p_t}\cdot\frac{\partial}{\partial x_i}\frac{{\rm e}^{x_i}}{\sum\limits_{j}^{N}{\rm e}^{x_j}}=\begin{cases}\dfrac{1}{p_t}(p_t-p_t^2)=1-p_t,\ i=t;\\ \dfrac{1}{p_t}(-p_i\cdot p_t)=-p_i,\ i\neq t.\end{cases} \tag{12}$$

将式(10)和(12)代入(9),可得

$$\frac{\partial L'_{ce}(x,t)}{\partial x_i}=\log(p_t)\cdot\frac{\partial(p_t)}{\partial x_i}-(1-p_t)\cdot\frac{\partial\log(p_t)}{\partial x_i}=$$

$$\begin{cases}\log(p_t)(p_t-p_t^2)-(1-p_i)(1-p_t),\ i=t\\ \log(p_t)(-p_ip_t)-(1-p_t)(-p_i),\ i\neq t\end{cases}=$$
$$\begin{cases}-(1-p_t)(1-p_t-p_t\log(p_t)),\ i=t;\\ p_i(1-p_t)(1-p_t-p_t\log(p_t)),\ i\neq t.\end{cases}\tag{13}$$

2 多层特征融合

SSD网络参与分类和定位回归的是多层特征图,这些特征图呈金字塔结构. 下面先简单介绍特征金字塔和图像反卷积,进而给出本文所设计的多层特征融合模型.

2.1 图像金字塔与特征金字塔

在目标检测中,经常遇到多尺度问题,通常采用图像金字塔[15]和特征金字塔[16-17]的方法. 特征金字塔是由图像金字塔发展而来,它利用卷积特性,在提取特征的同时也减小了图像尺寸. 一个卷积神经网络在不同的特征层,其语义信息是不同的[18]. 特征金字塔中每一层特征都有丰富的语义信息,但是,如果使用金字塔中的全部特征图,无疑会加大运算量,并且产生较多冗余信息. 经过对特征图的分析,实验确定使用conv4-3之后的部分特征层用于目标检测.

2.2 图像反卷积

不同卷积层的特征图有着不同的尺寸,因此,在进行特征融合之前,需要对相融合的特征图进行尺寸变换,这就需要用到反卷积结构[19]. 反卷积,可以简单理解为卷积的逆过程. 即卷积层的反向传播就是反卷积的前向传播,卷积层的前向传播就是反卷积的反向传播.

2.3 多层特征融合结构

SSD网络分别在conv4_3至conv11的6层特征图上进行分类回归,即使用conv4_3、conv7、conv8_2、conv9_2、conv10_2和conv11_2这6层特征图进行检测,较大的特征图用来检测相对较小的目标,而较小的特征图负责检测较大的目标[11].

通过对卷积层可视化结构图可以看出:特征层conv3_3由于深度较浅,边缘信息以及非目标干扰信息较为明显;conv4_3和conv5_3两层特征图,除了有大致的轮廓信息以外,还包含了更多的抽象语义信息;对于更深的conv8_2和conv9_2特征层,基本的轮廓信息以及细节信息都丢失了,这对于小目标的检测效果不是很明显. 如果加以融合,则不仅增加了计算量,而且对于融合后所带来的信息增益并不明显.

综上,针对SSD仅利用少量浅层特征图来检测目标,缺少足够的语义信息所导致的小目标检测精度低的问题,本文提取并融合浅层特征图,加强浅层特征图的语义信息,即选取conv4_3到conv7之间的特征图进行特征融合,多层特征融合结构如图1所示.

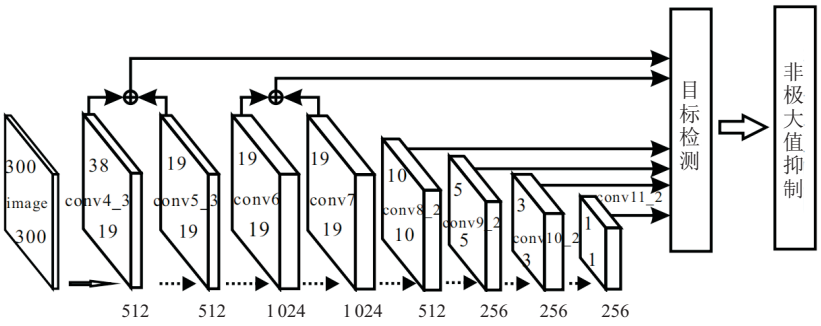

图1 多层特征融合结构

特征图的融合方式主要有两种:通道级联和同位置元素相加[20]. 通道级联法增加了原有的通道数,即描述图像本身的特征数(通道数)增加了,而每一特征下的信息没有增加. 同位置元素相加法将所对应的特征图相加,再进行下一步的卷积操作. 该方法并未改变图像的维度,只是增加了每一维下的信息量,这对最终的图像分类显然是有益的. 此外,同位置元素相加法所需要的内存和参数量小于通道级联法,故计算量也小于通道级联法. 所以,本文选择同位置元素相加法进行特征图融合.

3 仿真实验

3.1 实验数据集

本文采用PASCAL VOC数据集(VOC2007和VOC2012)[21-22]进行训练和测试,该数据集组成为:目标真值区域、类别标签、包含目标的图像、标注像素类别和标注像素所属的物体. 该数据集总共分4个大类:vehicle、household、animal和person,共计21个小类(包括1个背景类). 实验统一图片规格为300×300.

3.2 检测模型评价指标

在对目标检测模型进行分析评价中,本文使用公共评价指标:平均精确度均值(mean average preci-

sion, mAP) 对模型进行评价[23]. 下面先给出准确率(precision)和召回率(recall)的定义,进而给出mAP的定义.

准确率是指在所有正样本中,正确目标所占的比例,衡量的是查准率;召回率是指在所有真实的目标中,被模型正确检测出来的目标所占的比例,衡量的是查全率. 其计算公式分别为

$$\text{precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \tag{14}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}. \tag{15}$$

其中: TP为模型正确检测的目标个数, FP为模型错误检测的目标个数, FN为模型漏检的正确目标个数.

以召回率为横坐标,以准确率为纵坐标,二者形成的曲线称为*p-r*曲线,用来显示检测模型在准确率与召回率之间的平衡. *p-r*曲线下的面积为该类别的平均精度(average precision, AP). 在多类别分类中,通过求取各个类别AP的均值来计算模型整体的检测性能指标,其计算方法如下:

$$\text{AP} = \int_0^1 \boldsymbol{P}(R)\mathrm{d}R, \tag{16}$$

$$\text{mAP} = \frac{1}{c}\sum_{i=1}^{c}\text{AP}_i = \frac{1}{c}\sum_{i=1}^{c}\int_0^1 \boldsymbol{P}(R)\mathrm{d}R. \tag{17}$$

其中: c为目标检测的类别数, AP_i为第i类目标的平均精确度.

3.3　实验结果与分析

为加快网络模型的收敛速度并提升网络性能,本文将已训练好的VGG16作为预训练模型,后续目标检测只需在其基础上进行微调训练即可. 本文采用随机梯度下降法进行模型优化,设定初始学习速率为0.001,权值衰减为0.000 5,动量为0.9;卷积核大小为3×3, IOU设置为0.5;采用Pytorch深度学习框架, Python版本为Anaconda 3.6,实验统一图片规格为300×300.

表1给出了Fast R-CNN[3]、Faster R-CNN[5]、YOLO[6]、YOLO v3[9]、SSD300[11]、DSSD321[20]以及本文算法的目标检测精度.

表1　不同目标检测算法检测结果对比

目标检测算法	基础网络	训练集	测试集	输入大小	mAP/%	预测框数量
Fast R-CNN	VGGNet	VOC07+12	VOC 2007	–	70.0	–
Faster R-CNN	VGGNet	VOC07+12	VOC 2007	–	73.2	6 000
YOLO	GoogleNet	VOC07+12	VOC 2007	448×448	63.4	98
YOLO v3	DarkNet-53	VOC07+12	VOC 2007	416×416	75.4	10 647
SSD300	VGGNet	VOC07+12	VOC 2007	300×300	74.3	8 732
DSSD321	ResNet-101	VOC07+12	VOC 2007	321×321	78.6	17 080
本文算法	VGGNet	VOC07+12	VOC 2007	300×300	78.1	8 732

本文算法以VGGNet为基础网络,其在检测精度方面较Fast R-CNN、Faster R-CNN、YOLO、YOLO v3和SSD300均有优势,但是对比基础网络为ResNet-101的DSSD算法而言,精度稍有下降. 主要原因是, VGGNet网络较浅,而ResNet-101是非常深的网络,网络越深,目标特征越能够更好地被提取出来,因此检测精度越高.

除了检测精度外,时间复杂度也是算法设计时需要考虑的问题. 因Fast R-CNN、Faster R-CNN、YOLO、SSD300、DSSD321算法的运行平台与本文算法不同,所以本文用基础网络的层数、基础网络所占内存的大小(网络参数)和预测框的数量来衡量不同算法的时间复杂度. GoogleNet[24]、VGGNet[25]、DarkNet-53[7]和ResNet-101[26]的层数分别为22层、19层、53层和101层,它们所占的内存分别为99.8 M、82.1 M、30.8 M和170 M.

一般而言,层数越多,所占内存越大,预测框数量越多,则认为算法的时间复杂度越高. 从表1和上述基础网络参数可以看出, YOLO算法中基础网络的层数和所占内存略高于VGGNet,但是预测框数量较少,所以其计算复杂度较低. YOLO v3使用的基础网络是DarkNet53,其性能可以与最先进的分类器媲美,但是因DarkNet53需要更少的浮点运算,所以时间复杂度较低. Fast R-CNN、Faster R-CNN、SSD300和本文算法都使用VGGNet作为基础网络, Faster R-CNN的预测框数量相对较少,所以时间复杂度也较低. Fast R-CNN采用的是选择性搜索算法,其计算复杂度要高于采用候选框生成算法的Faster R-CNN. DSSD算法所使用的基础网络ResNet-101的层数远多于本文所采用的VGGNet,所占用的内存高出87.9 MB,在预测框的数量上, DSSD网络比本文算法多8 348个,因此, DSSD算法计算复杂度最高.

图2给出了不同算法在20个种类的测试集上的目标检测结果. 从实验结果可以看出,本文算法对于bicycle、bus、car、cat、dog、horse、motorbike、train这8类目标检测效果较好,都已达到了85%以上.

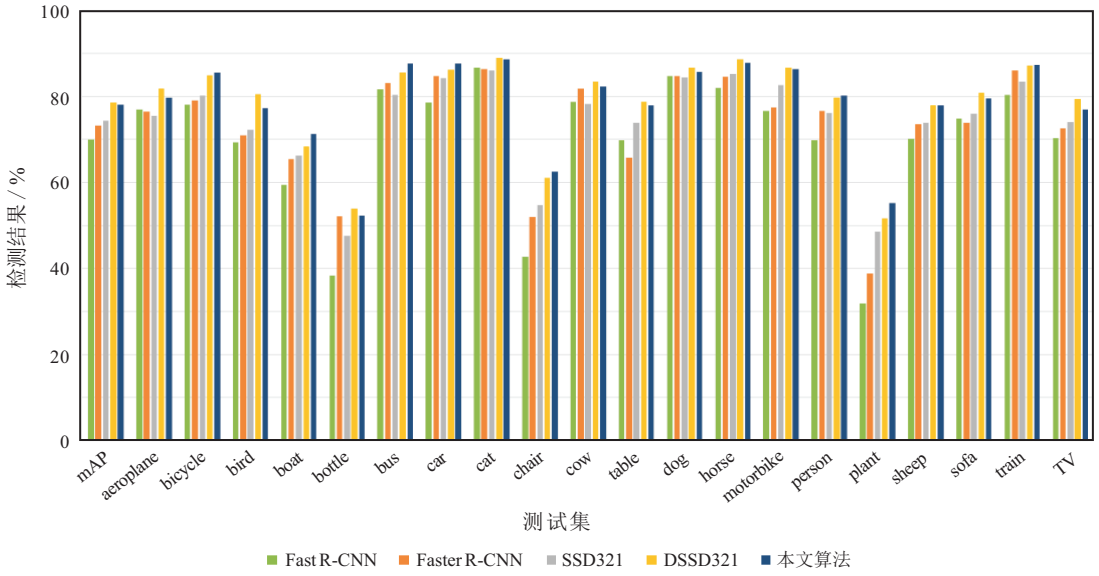

图 2 PASCAL VOC2007上各类别检测结果对比

综合表1和图2可以看出,本文算法在多数类别上的检测精度均能获得较好表现,尤其是对于bicycle、bus、car、person等复杂背景下的目标,相比于SSD网络提升较为明显,mAP分别提高了5.4%、7.3%、3.5%、4%. 但相比于DSSD网络在bird、bottle、cow、table、sofa、TV这些类别上,检测精度稍有下降,其原因可能是基础网络的不同而导致的特征提取信息不足.

为验证本文算法对不同大小目标的检测精度,实验中随机选取100张图片,其中包含198个目标,将其分为大、中、小三类. 由于该网络的输入图像尺寸为300×300,将图像中的检测目标按照其面积占图像总面积的比例分为三类: 目标面积占图像总面积5%以下的认为是小目标,目标面积占图像面积5%～25%的是中等目标,目标面积占图像总面积20%以上的是大目标. 表2给出了SSD算法和本文算法的检测结果(其中: A方法为SSD算法, B方法为本文算法).

表 2 随机检测结果对比

测试目标尺寸	小目标		中目标		大目标	
目标总数	36		54		108	
方法	A	B	A	B	A	B
检测数	16	21	41	43	89	91
检测率/%	47.1	58.3	75.9	79.6	82.4	84.2

由表2可知,本文算法对于不同尺寸的目标检测精度均有不同程度的提高,尤其是对于小目标的检测率由原来的47.1%增加到58.3%.

图3给出了不同情况下的目标检测结果,可以看出,本文算法对小目标的检测、存在遮挡物的检测以及在云雾天气和夜间的检测都有不错的效果.

(a) 小目标 (b) 雨雾天气

(c) 遮挡情况 (d) 夜间场景

(e) 复杂背景大目标 (f) 复杂背景小目标

图 3 不同条件下的目标检测结果

4 结 论

针对正负样本不均衡所导致的低分类精度等问题,本文在原SSD算法的损失函数中引入调制因子,减小简单样本的损失权值,增加困难样本的损失值所占比重,以达到提高复杂背景下目标检测精度的目的. 同时,调制因子的引入可以减少原模型交叉熵损失函数浪费在容易样本上的计算力,使得损失函数可以更快地跳过原有容易样本的简单数据,更快地进入后面困难样本的计算,从而加快训练阶段的收敛速度. 其次,针对因网络结构的缺陷而引起的小目标检

测精度欠佳问题,本文采取一种基于特征金字塔的多层特征检测结构,以增强用于检测小目标的浅层特征图语义信息. 实验结果表明,本文算法在多种类别目标的检测精度上都较SSD算法有了不同程度的提高,尤其是在小目标检测识别方面,检测精度显著提高.

参考文献(References)

[1] Wu X W, Sahoo D, Hoi S C H. Recent advances in deep learning for object detection[J]. Neurocomputing, 2020, 396: 39-64.

[2] Girshick R, Donahue J, Darrell T, et al. Region-based convolutional networks for accurate object detection and segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2016, 38(1): 142-158.

[3] Girshick R. Fast R-CNN[C]. IEEE International Conference on Computer Vision. Santiago, 2015: 1440-1448.

[4] Uijlings J R R, Sande K, Gevers T, et al. Selective search for object recognition[J]. International Journal of Computer Vision, 2013, 104(2): 154-171.

[5] Ren S Q, He K M, Girshick R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[6] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]. IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016: 779-788.

[7] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger[C]. IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, 2017: 6517-6525.

[8] Ju M R, Luo H B, Wang Z B, et al. The application of improved YOLO v3 in multi-scale target detection[J]. Applied Sciences, 2019, 9(18): 3775.

[9] Redmon J, Farhadi A. YOLO v3: An incremental improvement[C]. IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, 2018: 1-6.

[10] He W P, Huang Z, Wei Z F, et al. TF-YOLO: An improved incremental network for real-time object detection[J]. Applied Sciences, 2019, 9(16): 3225.

[11] Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multibox detector[C]. European Conference on Computer Vision, Amsterdam, 2016, 9905: 21-37.

[12] Bosquet B, Mucientes M, Brea V M. STDnet: Exploiting high resolution feature maps for small object detection[J]. Engineering Applications of Artificial Intelligence, 2020, 91: 103615.

[13] Buda M, Maki A, Mazurowski M A. A systematic study of the class imbalance problem in convolutional neural networks[J]. Neural Networks, 2018, 106: 249-259.

[14] 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.

[15] 曹晓明. 基于多图像特征金字塔的车辆检测[D]. 北京: 北京交通大学, 2016.
(Cao X M. Vehicle detection based on a multi-channel image feature pyramid[D]. Beijing: Beijing Jiaotong University, 2016.)

[16] 李春伟, 于洪涛, 高超, 等. 结合快速特征金字塔计算的可变形部件模型[J]. 小型微型计算机系统, 2016, 37(11): 2532-2536.
(Li C W, Yu H T, Gao C, et al. Deformable part model with fast computation of feature Pyramids[J]. Journal of Chinese Computer Systems, 2016, 37(11): 2532-2536.)

[17] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]. IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, 2017: 936-944.

[18] Du L, Li L, Wei D, et al. Saliency-guided single shot multibox detector for target detection in SAR images[J]. IEEE Transactions on Geoscience and Remote Sensing, 2020, 58(5): 3366-3376.

[19] Zeiler M D, Krishnan D, Taylor G W, et al. Deconvolutional networks[C]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. San Francisco, 2010: 2528-2535.

[20] Fu C Y, Liu W, Ranga A, et al. DSSD: Deconvolutional single shot detector[J/OL]. 2017, arXiv: 1701.06659.

[21] Kong T, Yao A B, Chen Y R, et al. HyperNet: Towards accurate region proposal generation and joint object detection[C]. IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016: 845-853.

[22] Everingham M, Gool L, Williams C K I, et al. The pascal visual object classes (VOC) challenge[J]. International Journal of Computer Vision, 2010, 88(2): 303-338.

[23] 孙南. 基于改进SSD模型面向中小目标的检测研究[D]. 开封: 河南大学, 2020.
(Sun N. Detection of small and medium targets based on improved SSD model[D]. Kaifeng: Henan University, 2020.)

[24] Szegedy C, Liu W, Jia Y Q, et al. Going deeper with convolutions[C]. IEEE Conference on Computer Vision and Pattern Recognition. Boston, 2015: 1-9.

[25] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J/OL]. 2014, arXiv: 1409.1556.

[26] He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016: 770-778.

作者简介

王红梅(1977−), 女, 教授, 博士, 从事深度学习、目标识别和图像融合等研究, E-mail: haipw@nwpu.edu.cn;

王晓鸽(1995−), 女, 助研, 硕士, 从事航空计算技术的研究, E-mail: 565794086@qq.com;

王晓燕(1996−), 女, 硕士, 从事深度学习、小样本识别的研究, E-mail: Wang_Anita@outlook.com.

(责任编辑: 李君玲)