计算机科学
COMPUTER SCIENCE

基于改进拆分注意力网络的目标检测算法

潘毅，王丽萍

引用本文

潘毅，王丽萍. 基于改进拆分注意力网络的目标检测算法[J]. 计算机科学, 2022, 49(10): 198-206.

PAN Yi, WANG Li-ping. Object Detection Algorithm Based on Improved Split-attention Network[J]. Computer Science, 2022, 49(10): 198-206.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.210800214

基于改进拆分注意力网络的目标检测算法

潘 毅 王丽萍
浙江工业大学计算机科学与技术学院 杭州 310023
(735442196@qq.com)

摘 要 当前,以卷积神经网络为基础的目标检测算法大多存在缺少对有价值的上下文信息的合理利用以及易对困难目标漏检等问题。针对这些问题,提出了一种基于改进拆分注意力网络的目标检测算法。首先,引入拆分注意力机制,将多通道结构与注意力机制相结合,提升其特征表示。然后,在网络的卷积层中使用多尺度卷积取代传统的卷积操作,增强了神经网络对尺度变化的敏感性。最后,将改进的网络应用于 Faster R-CNN 中,并在 Pascal VOC 数据集和 MS COCO 数据集上进行实验。所提算法在不增加超参数量及计算复杂度的情况下,其 *mAP* 相较于原始算法分别提升了 1.6%和 2.4%,且对比其他算法也有所优势,验证了所提算法的良好性能。

关键词:卷积神经网络;上下文信息;目标检测;拆分注意力;多尺度卷积
中图法分类号 TP391

Object Detection Algorithm Based on Improved Split-attention Network

PAN Yi and WANG Li-ping
College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

Abstract Recently, most object detection algorithms based on convolutional neural network have the problems of lacking of reasonable use of meaningful contextual information and are easy to miss the detection of hard targets. In order to solve these problems, this paper proposes an object detection algorithm based on improved split-attention networks. Firstly, the split attention mechanism is introduced, and the multi-path structure is combined with feature-map attention mechanism to improve its feature representations. Then, in the convolution layer, poly-scale convolution is used to replace the vanilla convolution to enhance the scale-sensitivity of the neural network. Finally, the proposed algorithm is applied to Faster R-CNN. Experiments are carried out on Pascal VOC and MS COCO datasets. Compared with the original algorithm, the *mAP* of the proposed algorithm has improved 1.6% and 2.4% respectively without introducing additional parameters and computational complexities, and the *mAP* of the proposed algorithm is also higher than that of other algorithms, which verifies its good performance.

Keywords Convolutional neural network, Contextual information, Object detection, Split-attention, Poly-scale convolution

1 引言

目标检测是计算机视觉领域中的热门研究方向之一,被广泛应用于行人检测[1]、自动驾驶[2]、视频监控[3]等方面。目标检测也成为了许多其他计算机视觉任务的基础,如实例分割[4]、目标跟踪[5]等。近年来,基于深度学习的目标检测算法逐渐成为主流[6]。深度学习的方法又可分为单阶段法(One-Stage)和双阶段法(Two-Stage)。单阶段法不需要产生候选框,直接将目标框定位的问题转化为回归问题处理。其特点是速度快。代表性算法有 YOLO[7]和 RetinaNet[8]等。双阶段法首先需要通过候选区域(Region Proposal)算法产生目标的候选框,然后对候选框做分类与回归。相较于单阶段法,双阶段算法准确率高,但速度较慢。代表性算法有 Faster R-CNN[9]和 R-FCN[10]等。

目前,许多目标检测的最新工作仍使用残差网络[11](Residual Network,ResNet)或其变体作为特征提取网络即骨干网络。起初,ResNet 结构是为图像分类而设计的,由于感受野的大小有限且缺乏跨通道交互作用,其不能完全适用于目标检测、实力分割等下游应用[12]。若要提高相关计算机视觉任务的性能,则需要对骨干网络进行修改,使其更有效地处理对应的任务。

对于卷积神经网络而言,从各种大小的目标中收集信息并了解有价值的上下文背景信息至关重要。但是,流线型架构的卷积神经网络通常具有固定大小的感受野,缺乏解决此类问题的能力,这限制了其在视觉识别任务上的性能表现,尤其是对尺度敏感的密集预测问题。FCN 等[13]证明了多尺度

到稿日期:2021-08-24 返修日期:2022-03-04
基金项目:浙江省重点研发计划(2018C01080)
This work was supported by the Key Technologies Research and Development Program of Zhejiang Province, China(2018C01080).
通信作者:王丽萍(wlp@zjut.edu.cn)

表示法具有感知不同感受野的能力，并且使算法的性能得到了显著改善。后续据此提出的许多方法主要是探索或改进具有更复杂的跳跃连接或并行流的多尺度特征融合。但是，现有的大多数方法都以分层或逐层的方式捕获多尺度特征，将重点放在整个网络的体系构建上，这样无疑会引入更多的参数，增加了额外计算的复杂性。

本文针对上述研究问题，提出了一种基于改进拆分注意力网络（Split-Attention Networks）的目标检测算法。本文的贡献主要如下：

（1）在特征提取网络中引入拆分注意力（Split-Attention）模块，采用多通道结构与注意力机制，丰富了特征图的多样性，并加强了特征图之间的联系，提高了目标检测的性能；

（2）引入多尺度感受野融合模块，在卷积层采用一系列规则的膨胀因子，通过在卷积滤波器不同通道中循环地采用一个膨胀率谱来有效聚合多尺度特征[14]，进一步增强了卷积神经网络缩放尺度变化的鲁棒性；

（3）将改进的网络分别应用在 Faster R-CNN 和 RetinaNet 上，并在 Pascal VOC 数据集与 MS COCO[15] 数据集上对本文提出的改进算法与原始算法进行对比实验。结果表明，本文方法的目标检测的平均精度均值 *mAP* 在 Pascal VOC 数据集上平均提高了 1.6%，在 MS COCO 数据集上提高了 2.4%。此外，相比其他检测算法，本文算法表现优异。

2　相关工作

2.1　特征提取网络

通常，目标检测器的精度在很大程度上取决于其特征提取网络，即骨干网络。2012 年，Krizhevsky 等提出了 AlexNet[16]，网络一共 8 层，由 5 层卷积层及 3 层全连接层组成。AlexNet 一经提出，便赢得了当年 ImageNet 竞赛的冠军。自此，深层卷积神经网络开启了在图像分类领域的主导趋势。2014 年，牛津视觉几何组（VGG）提出了 VGG-Net[17]，其展现了模块化的网络设计策略，将模型的深度增加至 16～19 层，并使用 3×3 的卷积核取代在 AlexNet 中使用的 5×5 和 7×7 的卷积核，通过将相同类型的网络块重复堆叠，简化了网络设计的工作。GoogLeNet[18-19] 是由 Google 团队提出的一系列基于 Inception 模块的深度神经网络模型，除增加了 CNN 的宽度和深度外，该系列的主要贡献是引入了分解卷积（Factorizing Convolution）和批归一化（Batch Normalization，BN），加快了模型的收敛速度，并且在一定程度上缓解了深层网络中特征分布较散的问题。2014 年，Girshick 等提出 R-CNN[20] 这种全新的目标检测框架。R-CNN 首选通过选择性搜索（Selective Search）提取一组对象候选框，然后将每个候选框重新缩放为固定大小的图像，使用神经网络提取特征，接着用线性 SVM 分类器预测每个区域内对象的类别，最后使用回归器修正候选框的位置。但是，其冗余的特征计算导致了极慢的检测速度。次年，Girshick 等[21] 以 R-CNN 为基础，基于 R-CNN 中存在的训练耗时长、训练速度慢、训练所需空间大等问题进行改进，提出了 Fast R-CNN。Ren 等[22] 于 2016 年提出 Faster R-CNN，并针对 Fast R-CNN 遗留下来的问题进行进一步改进，提出用区域生成网络（Region Proposal Network，RPN）代替选择性搜索候选框。Faster R-CNN 在基于深度神经网络检测算法中具有里程碑式的意义[6]，后续的很多算法都是基于此进行修改。同年，He 等[11] 又提出一种基于 ResNet 的框架，并以之取代使用 VGG-Net 或者 AlexNet 的传统检测框架。ResNet 比之前的卷积网络架构深得多，最多可达 152 层；同时，ResNet 建立在开拓性工作取得成功的基础上，引入了跳跃连接（Skip Connection），可缓解深度神经网络中因网络深度增加而出现的梯度消失或梯度爆炸等问题，并允许网络学习更深层的特征表示。目前，ResNet 作为最成功的 CNN 架构之一，被各类计算机视觉应用方案所采用。2017 年，Huang 等提出 DenseNet[23]，在网络中引入了紧密连接（Dense Connection）的块，该块以前馈的方式将每一层连接到其他每一层。DenseNet 加强了对特征的复用，相比 ResNet 拥有更少的参数数量，有效抑制了过拟合，使得网络更易训练。SE-Net（Squeeze-and-Excitation Networks）[24] 由 Hu 等于 2018 年提出，它的主要贡献是集成了全局池化和改组，以了解特征图在通道方面的重要性。上述的很多改进工作为了提高模型的准确率，基本都是加深网络层数或是加宽网络的结构，但是随着超参数量如通道数、卷积核大小等的增加，网络设计的难度和计算开销也会增加。对此，Zhang 等[12] 进一步提出 ResNeSt 结构，在不增加参数复杂度的前提下提升模型的准确率，还减少了超参数的数量。

2.2　多通道及特征图注意力

多通道的卷积操作已在 GoogleNet 中取得成功，其中每个 Inception 块均由不同尺寸的卷积核组成。Xie 等于 2017 年提出的 ResNeXt[25] 在 ResNet 的 Bottleneck 块中采用分组卷积，将多通道结构转换为统一操作。ResNeXt 打破了加深或者加宽网络以换得性能提高的传统思维桎梏，在网络中引入了基数（Cardinality），并用实验证明了增加基数组可取得比增加网络深度或宽度更有效的优点。SE-Net 通过引入通道注意力（Channel-attention）机制来自适应地重新校准通道特征响应。SK-Net[26] 则通过两个网络分支引入特征图注意力（Feature-map Attention）机制，来调整神经网络中神经元的自适应感受野尺寸的能力。ResNet-50 与 SE-Net 和 SK-Net 的 Block 的对比如图 1 所示。

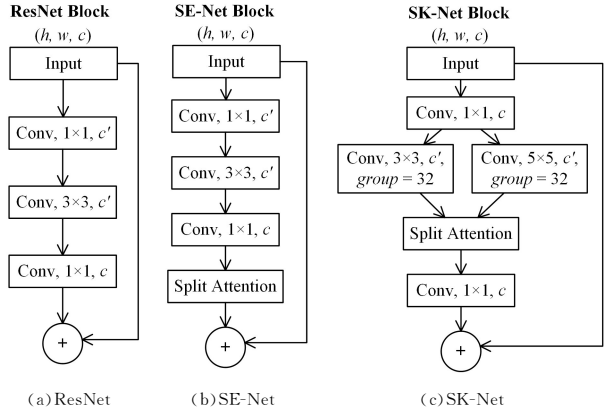


图 1　ResNet 与 SE-Net 和 SK-Net 的网络模块结构的对比

Fig. 1　Comparison of network module structure of ResNet, SE-Net and SK-Net

受先前方法的启发，本文采用的骨干网络 ResNeSt 基于 ResNet 的架构做了进一步的修改。ResNeSt 将多分支结构与通道注意力机制结合，将通道注意力概括为特征图组表示，然后对不同组的特征加权生成最终的特征图。相比 ResNeXt，ResNeSt 在基数组中又添加了 r 个分支，进一步将特征图的注意力分散到单个网络块中；相比 SE-Net，ResNeSt 将不同分支的卷积核操作大小统一成 3×3，相对减小了计算的复杂度。原生 ResNeSt 的一个 Block 模块结构如图 2 所示。

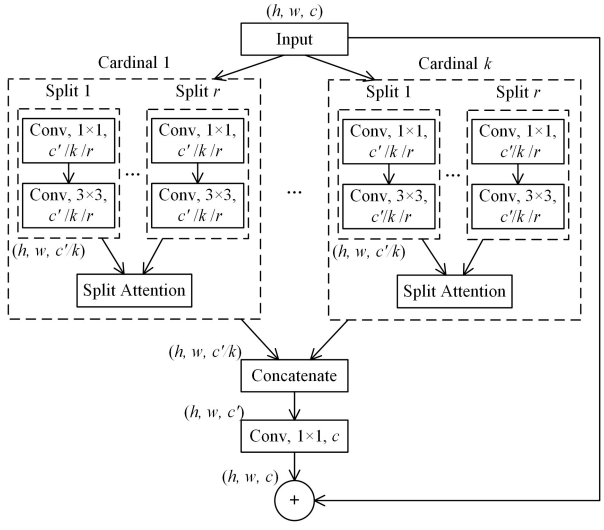


图 2 ResNeSt 网络层模块的结构图

Fig. 2 Structure diagram of ResNeSt network layer module

2.3 多尺度信息

当前的卷积神经网络虽具有强大的建模能力，但其通常对尺度较为敏感。为了增强 CNN 尺度变化的鲁棒性，将来自不同层或卷积核的多尺度特征融合在现有的解决方案中引起了极大的关注。2017 年，Lin 提出了特征金字塔网络(Feature Pyramid Networks，FPN)[27]。受 ResNet 的残差连接和 SSD[28]的检测策略的启发，通过自上而下和横向连接来融合多种分辨率的特征，在不同分辨率的特征图上分别做预测，结合了浅层特征和深层特征。浅层特征在图像中寻找诸如边缘、角和纹理之类的结构；而深层特征则是提取语义信息，如轮廓和类别。FishNet[29]在主干尾部堆叠了一个向上采样的主体和一个向下采样的头部，保留了多种分辨率的特征。大小网络(Big-Little Net，BL-Net)[30]由具有不同计算复杂度的分支组成，每个分支只有一个单一的图像尺度，其中将较少卷积层和通道用在具有高分辨率的分支以节省计算资源，并且将不同尺度的不同分支的特征通过线性组合进行合并。但是，以上算法应用至检测算法框架时，不可避免地需要通过选择新的超参数和卷积层的配置来调整原始的框架结构。另一项处理多尺度信息的方式，如 TridentNet[31]等，是利用膨胀卷积(Dilated Convolution)来扩大感受野，如图 3 所示。但是，单纯地使用膨胀卷积对识别大物体有利，对小物体的识别效果不佳。此外，膨胀卷积的核并不连续，膨胀卷积不能覆盖所有的图像特征，即栅格效应。

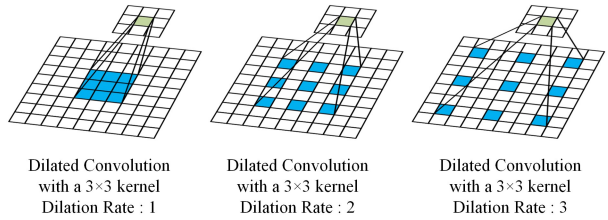


图 3 膨胀卷积效果

Fig. 3 Dilated convolutions with a kernel size of 3×3

文献[32]提出可形变卷积，在卷积核或特征图的每个采样位置学习偏移量，以使卷积过程中的感受野出现变形。将可形变卷积模块加入网络，并以自动化方式处理各种感受野，使得网络关注特征图中信息更重要的部分。文献[33]则从另一个角度利用多尺度信息，在单个深度可分离卷积通道内分组，每组使用不同大小的卷积核，最终将卷积结果 concat 连接，但这样无疑增加了计算量。

基于上述问题，Li 通过更精细地利用多尺度特征，提出可以直接取代传统卷积操作的多尺度卷积(Poly-Scale Convolution，PSConv)[14]。PSConv 混合了一系列的卷积膨胀率，并在每个卷积层的单个卷积内核中有规律地将它们分配给单个卷积层。这样既可以不损失分辨率和不增加计算复杂度，又可以更好地利用更细粒度的卷积核空间，在不引入其他参数和计算复杂度的情况下更有效地增强多尺度特征表示。

3 改进的目标检测算法框架

3.1 拆分注意力模块

针对当前 50 层 ResNet 模型，由于其感受野大小有限以及缺少跨通道信息交换而不适用于下游应用的问题，对 ResNet 的网络结构进行改动，融入了多分支和通道注意力的机制。通过借鉴 SK-Net 和 ResNeXt，本文引入拆分注意力块。拆分注意力块是一种计算单元，由特征图组和拆分注意力操作组成。图 4 给出了拆分注意力模块示意图。

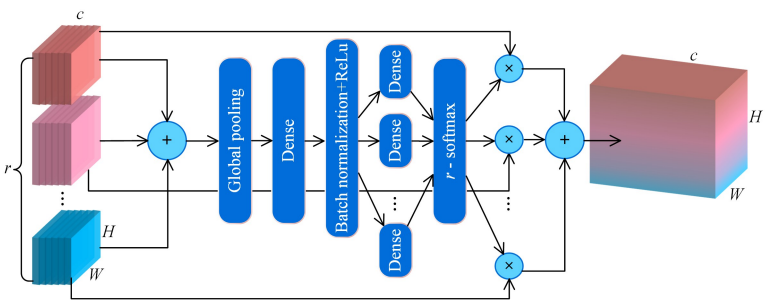


图 4 拆分注意力模块

Fig. 4 Split-attention module

首先，将特征图分为 K 组（K 为超参数），如图 2 所示，所得的特征图组被称为基数组。此外，在网络层中引入了一个新的超参数 R，即基数组内的拆分数。因此，特征图组的总数为 $G=K\times R$。最初，输入的特征图通过应用一系列映射变换 $\{F_1,F_2,\cdots,F_G\}$ 分配到每个单独的组，其中组序号 $i\in\{1,2,\cdots,G\}$，每个组的中间表示为 $(K,G,\Lambda,W,D,T_{\text{net}},\lambda_{\max},B,\mu,c)$。

每个基数组的组合表示可以在多个分段之间通过逐元素求和来融合获得。第 k 个基组的表示为：$\hat{U}^k=\sum_{j=R(k-1)+1}^{Rk}U_j$。其中 $\hat{U}^k\in\mathbb{R}^{H\times W\times C_{\text{out}}/K}$，$k\in\{1,2,\cdots,K\}$，$H$，$W$ 和 C_{out} 分别是块输出特征图的长、宽以及通道数。

然后，聚合通道维度的全局上下文信息 $s^k\in\mathbb{R}^{C_{\text{out}}/K}$ 由全局平均池化降采样操作实现。其中，第 c 个分量的计算式如式(1)所示：

$$s_c^k=\frac{1}{H\times W}\sum_{i=1}^{H}\sum_{j=1}^{W}\hat{U}_c^k(i,j) \tag{1}$$

其中，s_c^k 是基于全局纹理表达 s^k 第 c 个通道的每一组权值。

基数组表示 $V^k\in\mathbb{R}^{H\times W\times C_{\text{out}}/K}$ 的加权融合是按通道分类的软注意力聚合的。其中，每个特征图通道是使用拆分后的加权组合生成的。第 c 个通道的计算式如式(2)所示：

$$V_c^k=\sum_{i=1}^{R}a_i^k(c)U_{R(k-1)+i} \tag{2}$$

其中，$a_i^k(c)$ 表示特征图组的软分配权重，其权重的计算式如式(3)所示：

$$a_i^k(c)=\begin{cases}\dfrac{\exp(\mathcal{G}_i^c(s^k))}{\sum_{j=0}^{R}\exp(\mathcal{G}_i^c(s^k))}, & \text{if } R>1\\[2ex] \dfrac{1}{1+\exp(-\mathcal{G}_i^c(s^k))}, & \text{if } R=1\end{cases} \tag{3}$$

其中，映射 $\mathcal{G}_i^c$ 根据全局上下文表示 s^k 确定第 c 个通道的每个分割的权重。

3.2　多尺度感受野融合模块

得益于 ResNeSt 网络框架以及多尺度卷积的启发，本文提出一种多尺度感受野融合模块，将其应用在 ResNeSt 框架上并进行改进。在 ResNeSt 的 Bottleneck 第二个卷积层的操作中，本文将传统卷积替换成多尺度的卷积操作，以期获得更广的感受野，并扩展了传统的单尺度卷积运算的范畴，增强了神经网络尺度变化的鲁棒性。

在原生 ResNeSt 的 Bottleneck 中，第二层 3×3 卷积操作使用的是组卷积的形式，k 个基组再次被拆分成 r 个分支。在此基础上，为了进一步利用多尺度的特征，卷积核内部采用不同的膨胀率，并且配合输入通道数，在卷积核中采用依次循环的膨胀因子来达到多尺度特征融合的目的。

传统的卷积操作中，$\mathcal{F}\in\mathbb{R}^{C_{\text{in}}\times H\times W}$ 表示其形状为 $C_{\text{in}}\times H\times W$ 的输入特征图，其中 C_{in} 是通道数，H 和 W 分别是输入特征图的高度和宽度。将一组卷积核大小为 $N\times N$ 的 C_{out} 个卷积滤波器与输入的特征图分别进行卷积，以获得具有 C_{out} 个通道的所需输出特征图，其中每个滤波器都有 C_{in} 个卷积内核以匹配输入图中的通道。

引入了多尺度卷积的操作后，通过膨胀卷积扩大对空间域的采样间隔，从而覆盖更大尺寸的目标对象。具体来说，多尺度感受野融合模块是将有组织的膨胀率分散在一个卷积滤波器中的不同卷积内核上。为了通过线性求和从不同的输入通道收集多尺度信息，在一个卷积滤波器中，不同卷积内核的膨胀率会发生变化。其中，关于多尺度融合卷积的运算如式(4)所示：

$$\mathcal{H}_{o,x,y}=\sum_{e=1}^{C_{\text{in}}}\sum_{i=\frac{N-1}{2}}^{\frac{N-1}{2}}\sum_{j=\frac{N-1}{2}}^{\frac{N-1}{2}}\mathcal{G}_{o,e,i,j}\mathcal{F}_{e,x+i\boldsymbol{D}(o,e),y+j\boldsymbol{D}(o,e)} \tag{4}$$

其中，$\mathcal{F}$ 表示输入的特征图；$\mathcal{G}$ 表示集合中的 C_{out} 个卷积滤波器；$\boldsymbol{D}$ 是一个矩阵，由两个正交维度上的输出特征图的通道数 $o\in\{1,2,\cdots,C_{\text{out}}\}$ 和每个卷积滤波器内的通道数 $e\in\{1,2,\cdots,C_{\text{in}}\}$ 对应的膨胀率组成。

多尺度感受野融合模块在卷积核中重新设置了膨胀率。由于特征提取网络采用分组卷积操作，输入特征图组 C_{in} 被分为 r 个通道分支。在每个通道分支中，我们又将通道进一步划分为 P 个分区，每个分区有 $t=\left\lceil\frac{C_{\text{in}}/r}{P}\right\rceil$ 个通道，t 个通道分别对应膨胀率 $\{d_1,d_2,\cdots,d_t\}$，膨胀率沿输入通道的轴以周期性的方式变化，从而实现所需的多尺度特征融合。此外，为了赋予不同的卷积滤波器收集输入特征图中不同比例组合的能力，本文采用基于移位的膨胀率策略将前一个卷积滤波器与后一个滤波器的膨胀率模式移动一个通道。因此，矩阵 $\boldsymbol{D}_{(o,e)}$ 内对应的卷积核膨胀率大小的计算方式如下：

$$\boldsymbol{D}(o,e)=d_{\left[o \bmod \frac{C_{\text{out}}}{r}+\left(o \bmod \frac{C_{\text{in}}}{r}\right) \bmod P-1\right] \bmod t} \tag{5}$$

综上所述，在一个分支通道中，卷积核的膨胀率沿着以输出通道数作为垂直轴和输入通道数作为水平轴的矩阵网格交替出现。在传统的卷积滤波器中，即使采用膨胀卷积，每一组卷积滤波器都使用固定的膨胀率；而多尺度感受野融合模块通过不引入其他参数和计算复杂性的情况下，具有更细粒度和更强特征提取能力的特征，同时保持近似的计算量负荷来实现更好的表征学习。其操作示意图如图 5 所示，其中设定特定的颜色将一种类型的膨胀率与其他类型的膨胀率区分开，即相同的颜色代表相同的膨胀率。

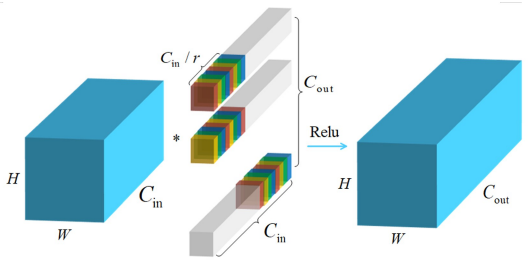


图 5　多尺度感受野模块示意图

Fig. 5　Schematic illustration of poly-scale receptive field module

3.3　改进的拆分注意力网络

根据 ResNeSt 中的 Block 结构，本文进行一定的修改。首先，设定基数组内的通道拆分数 r 为 2，即每个基数组内采用并行双通道的模式，取得计算复杂度与模型性能的平衡。在 k 个基数组内，多尺度感受野融合模块首先对特征图小组进行 1×1 卷积，将其拆分为 2 个小组；其次，进行 3×3 多尺度卷积操作，卷积核内的膨胀率设置为 1，2，1，4，依次有规律地进行循环，然后，使用 concat 将输出聚合；接着，经过拆分

注意力模块将特征图小组按对应元素相加，再进行全局平均池化操作，得到的 c' 维特征向量表示各个通道的权重；随后经过 BN＋Relu 操作以及两层全连接层；最后使用 softmax 操作，修正后的每个 channel 的权重向量再与原始的特征组进行对应元素相乘，得到输出。改进后的拆分注意力网络层结构如图 6 所示。

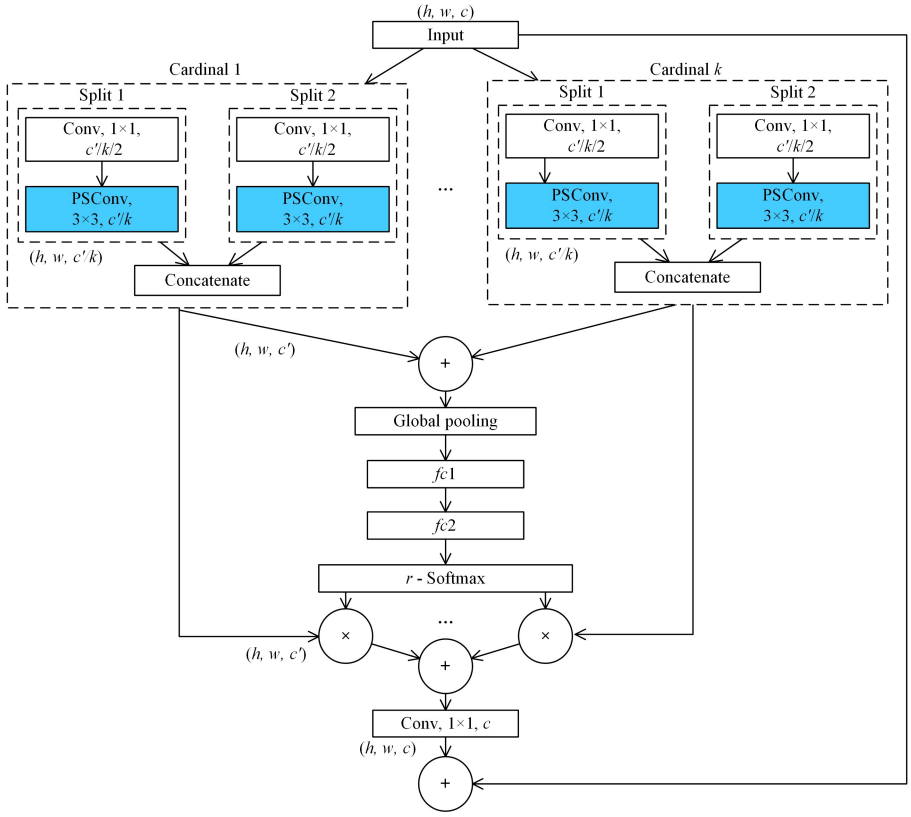


图 6 PS-ResNeSt 网络层模块图

Fig. 6 Diagram of PS-ResNeSt network layer module

我们将 50 层网络结构模块进行对比，其在残差网络层有所区别，如表 1 所列。对比的 4 种网络分别是 ResNet-50[22]，PS-ResNet-50[14]，ResNeSt-50[12]以及本文改进后的拆分注意力网络 PS-ResNeSt-50。其中，r 表示拆分组数量，C 代表基数组量。

表 1 4 种 50 层网络结构的对比

Table 1 Comparison of four 50-layer network architectures

Ouput	ResNet-50	PS-ResNet-50	ResNeSt-50	PS-ResNeSt-50
112×112	7×7, 64, stride 2			
	3×3 max pool, stride 2			
56×56	[Conv, 1×1, 64 Conv, 3×3, 64 Conv, 1×1, 256] ×3	[Conv, 1×1, 64 PSConv, 3×3, 64 Conv, 1×1, 256] ×3	[Conv, 1×1, 64 Conv, 3×3, 64, r, C Conv, 1×1, 256] ×3	[Conv, 1×1, 64 PSConv, 3×3, 64, r=2, C=32 Conv, 1×1, 256] ×3
28×28	[Conv, 1×1, 128 Conv, 3×3, 128 Conv, 1×1, 512] ×4	[Conv, 1×1, 128 PSConv, 3×3, 128 Conv, 1×1, 512] ×4	[Conv, 1×1, 128 Conv, 3×3, 128, r, C Conv, 1×1, 512] ×4	[Conv, 1×1, 128 PSConv, 3×3, 128, r=2, C=32 Conv, 1×1, 512] ×4
14×14	[Conv, 1×1, 256 Conv, 3×3, 256 Conv, 1×1, 1024] ×6	[Conv, 1×1, 256 PSConv, 3×3, 256 Conv, 1×1, 1024] ×6	[Conv, 1×1, 256 Conv, 3×3, 256, r, C Conv, 1×1, 1024] ×6	[Conv, 1×1, 256 PSConv, 3×3, 256, r=2, C=32 Conv, 1×1, 1024] ×6
7×7	[Conv, 1×1, 512 Conv, 3×3, 512 Conv, 1×1, 2048] ×3	[Conv, 1×1, 512 PSConv, 3×3, 512 Conv, 1×1, 2048] ×3	[Conv, 1×1, 512 Conv, 3×3, 512, r, C Conv, 1×1, 2048] ×3	[Conv, 1×1, 512PSConv, Conv, 1×1, 2048] ×3
1×1	Global average pool fc, softmax	Global average pool fc, softmax	Global average pool fc, softmax	Global average pool fc, softmax
Params	25.5×10^6	25.5×10^6	27.5×10^6	27.5×10^6

4　实验及结果分析

4.1　数据集与评估指标

为了验证本文改进模型算法在目标检测算法中的有效性，分别在 Pascal VOC 数据集与 MS COCO 2017 数据集上进行实验。

(1)Pascal VOC 数据集

Pascal VOC 为图像识别与分类提供了一整套标准化的数据集，从 2005 年到 2012 年每年都会举行一场图像识别的挑战赛。训练集均以带标签的图片的形式给出。数据集图像分为训练集(train)、验证集(val)、测试集(test)、训练与验证集(trainval)，图片数据集的分类及数量如表 2 所列。

表 2　Pascal VOC 数据集划分

Table 2　Division of Pascal VOC dataset

Dataset	train	val	trainval
VOC2007	2501	2510	5011
VOC2012	5717	5823	11540

本文使用 VOC2007 与 VOC2012 数据集进行训练，并使用 VOC2007 的测试集来测试数据。VOC 数据集主要分为 4 个大类：人、动物、交通车辆、室内家具用品。各大类又有多个分类，一共包括 20 个类别：飞机、单车、鸟、船、瓶、巴士、汽车、猫、椅子、牛、桌子、狗、马、摩托、人、盆栽、羊、沙发、火车、电视。

(2)MS COCO 数据集

COCO 的全称是 Common Objects in Context。MS COCO 2017 数据集是微软团队于 2017 年提供的一个可用于图像识别的数据集，此外还提供目标检测、分割和对图像的语义文本描述信息。目前大多数目标检测模型主要在 COCO 数据集上进行训练和检测。MS COCO 2017 数据集中的图像同样分为训练集、验证集和测试集，数据集划分方式如表 3 所列。数据集共有 80 个类别，如人、自行车、书、猫、手机等。

表 3　MS COCO 2017 数据集划分

Table 3　Division of MS COCO 2017 dataset

Dataset	train	val	test
MS COCO 2017	118287	5000	40670

(3)评估指标

实验采用的评估指标为平均精度均值(Mean Average Precision，mAP)[21]。平均精度均值被定义为数据集中所有类别的平均精度(Average Precision，AP)的均值，平均精度是根据精准率-召回率曲线(简称 PR 曲线)定义的。PR 曲线以召回率为 x 轴，精确率为 y 轴，精确率越高，召回率越高，意味着算法越高效。平均精度的计算如式(6)和式(7)所示：

$$AP=\sum_{n=1}^{N}(r_{n+1}-r_n)\rho_{\text{interp}}(r_{n+1}) \tag{6}$$

$$\rho_{\text{interp}}(r_{n+1})=\max_{\tilde{r}:\tilde{r}\geqslant r_{n+1}}\rho(\tilde{r}) \tag{7}$$

其中，r_n 代表第 n 个召回率(Recall)的取值，$\rho(\tilde{r})$ 代表召回率为 $\tilde{r}$ 时对应精准率(Precision)的取值。

召回率指正确检测出的目标框(True Positive，TP)数量占所有预测目标框(All Ground Truths)数量的比例。其中，所有预测目标框的数量大小或等于 TP 和漏检框(False Negative，FN)的和，漏检框 FN 表示没有检测出的物体。召回率的计算如式(8)所示：

$$Recall=\frac{TP}{all\_ground\_truths}=\frac{TP}{TP+FN} \tag{8}$$

精准率指 TP 数量占所有真实目标框(All Detections)数量的比例。所有真实目标框的数量等于 TP 与 FP 之和，FP(False Positive)表示错误分类的正例数量。精准率的计算如式(9)所示：

$$Precision=\frac{TP}{all\_detections}=\frac{TP}{TP+FP} \tag{9}$$

4.2　实验参数

本次实验以 Pytorch 深度学习框架为基础，使用 Python 3.7 进行编译，硬件信息为：Intel Core i7-8700K@3.70 GHz CPU，Nvidia RTX 2080 显卡，32 GB 内存。

实验将本文改进的模型(PS-ResNeSt-50)与骨干网络使用原生 ResNet-50 的算法、原生 ResNeSt-50 以及使用 PS-ResNet-50 的算法进行对比，并且本文所有网络主干都在 ImageNet1k 分类集上进行了预训练，然后在检测数据集上进行了微调。

在 Pascal VOC 数据集的实验中，将上述改进的网络应用于 Faster R-CNN 的骨干网络中。模型共训练了 30 万次，优化器采用随机梯度下降(Stochastic Gradient DescentSGD)，设置初始的学习率为 0.012，动量因子 momentum 设置为 0.9，权重衰减系数为 1×10^{-4}。学习率采用线性增加的策略，并且设置在初始的 500 次迭代中学习率逐渐增加，在 20 万次和 27 万次时将学习率分别调整为 1.2×10^{-3} 和 1.2×10^{-4}，并将 IoU 阈值设为 0.5。在实验过程中，将输入图像的尺寸统一归一化到 500 * 500 以进行数据的预处理。对分类进行评估的损失函数均设定为交叉熵损失函数(Cross Entropy Loss)，损失权重系数为 1；对回归框 Bbox 定位进行评估的损失函数为 L1Loss，损失权重系数为 1。

在 MS COCO 数据集上的模型训练实验中，将上述改进的网络分别应用于 Faster R-CNN 和 RetinaNet 的骨干网络中。Faster R-CNN 组和 RetinaNet 组采用同样的训练策略，分别训练 20 个 epoch，batch size 设置为 4，学习率为 0.012，动量因子 momentum 为 0.9，权重衰减因子为 1×10^{-4}。学习率采用线性增加的策略，并且设置在初始的 500 次迭代中学习率逐渐增加，在第 15 个 epoch 和第 18 个 epoch 时将学习率分别调整为 1.2×10^{-3} 和 1.2×10^{-4}。在实验过程中，我们将实验中输入图像的尺寸统一归一化到 1000 * 600 以进行数据的预处理。在 Faster R-CNN 组中分类进行评估的损失函数均设定为 Focal Loss 函数，损失权重系数为 1，对回归框 Bbox 定位进行评估的损失函数为 L1Loss，损失权重系数为 1；在 RetinaNet 组中分类进行评估的损失函数均设定为交叉熵损失函数，gamma 项为 2，alpha 值设为 0.25，损失权重系数为 1。

4.3　实验结果分析

4.3.1　Pascal VOC 数据集上的实验

首先，将本文提出的网络结构应用于 Faster R-CNN 的特征提取网络中，得到的 mAP 与各类对应的 AP 值的结果如

表 4 所列。检测的对比结果显示，本文提出的算法相比使用原生 ResNet-50、PS-ResNet-50 以及原生 ResNeSt-50 作为骨干网络的算法，绝大多数类别的 *AP* 值均有提升。其中，相较于原生 ResNet-50，本文算法的 *mAP* 提升了 1.6%；对比基于原生 ResNeSt-50 作为骨干网络的算法，本文算法在大多数类别的 *AP* 值均获得了提升，*mAP* 提升了 1.2%。

表 4 不同方法在 Pascal VOC 数据集上的比较

Table 4 Comparison of various methods on Pascal VOC dataset

（单位：%）

Backbone	*mAP*	*AP*																			
		aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv
ResNet-50	75.4	79.5	85.0	68.1	63.1	64.2	85.2	87.0	84.4	60.1	79.9	76.8	80.8	86.9	79.1	79.3	42.8	77.7	75.7	78.4	73.5
PS-ResNet-50	75.6	80.4	84.9	68.7	62.0	63.3	85.2	87.5	85.9	60.4	78.9	76.4	81.1	85.2	78.8	79.4	47.3	77.6	78.6	78.4	72.9
ResNeSt-50	75.8	80.4	80.4	77.4	63.5	64.4	79.4	88.1	87.9	57.2	79.0	71.3	86.5	86.9	80.6	79.8	50.1	73.8	73.7	79.4	76.9
Ours(PS-ResNeSt-50)	**77.0**	80.2	**87.4**	76.8	**66.9**	**66.8**	**87.1**	87.4	**87.9**	56.9	77.9	73.0	85.9	**88.3**	**80.9**	**79.8**	46.2	76.6	78.2	**80.9**	75.9

4.3.2 MS COCO 数据集上的实验

在 MS COCO 数据集上，将所提的改进算法分别应用于双阶段的目标检测算法 Faster R-CNN 与单阶段的算法 RetinaNet 中，并分别与其原始的算法进行对比。

不同方法在 COCO 数据集上的平均精度均值比较结果如表 5 所列。其中，mAP_{50} 表示 *IoU* 即交并比阈值为 0.5 时的 *mAP* 值，mAP_{75} 为 *IoU* 阈值为 0.75 时的测量值；mAP_S 为针对小目标即像素面积小于 32^2 的目标框的 *AP* 测量值，同理，mAP_M 为中等目标即像素面积在 $32^2 \sim 96^2$ 之间目标框的 *AP* 值，mAP_L 为大目标即像素面积大于 96^2 的目标框的 *AP* 值。

表 5 不同方法在 MS COCO2017 数据集上的比较

Table 5 Comparison of various methods on MS COCO2017 dataset

Detector	Backbone	*mAP*	mAP_{50}	mAP_{75}	mAP_S	mAP_M	mAP_L
Faster R-CNN	ResNet-50	34.0	53.2	37.0	19.1	37.1	43.8
	PS-ResNet-50	34.7	54.2	37.6	19.9	37.4	44.6
	ResNeSt-50	36.2	54.8	40.0	22.3	39.2	45.9
	Ours(PS-ResNeSt-50)	**36.4**	**54.9**	**40.2**	**22.8**	**40.0**	**46.7**
RetinaNet	ResNet-50	32.9	50.2	35.2	18.3	35.6	43.2
	PS-ResNet-50	33.1	51.1	36.0	19.2	36.8	44.2
	ResNeSt-50	35.0	53.0	37.2	21.1	38.6	44.3
	Ours(PS-ResNeSt-50)	**35.7**	**54.2**	**38.0**	**21.5**	**39.6**	**45.1**

在 Faster-RCNN 实验组中，每组对比实验中本文改进的网络模型在所有类别的 *AP* 值均有提升，*mAP* 提升了 2.4%。在大、中、小目标的检测上，本文算法相较于使用原生 ResNet 的算法平均提升了 3.1%，对比将原生 ResNeSt 作为骨干网络的算法平均提升了 0.7%。在 RetinaNet 实验组中，本文算法的 *mAP* 同样提升明显，提升了 2.8%。同时，对比将原生 ResNeSt 作为骨干网络的算法，本文算法也提升了 0.7%。这得益于多尺度感受野融合模块以及多通道注意力机制，其增强了卷积神经网络对尺度变化的鲁棒性，进一步提升了检测效果。

其次，将提出的改进算法与近年来提出的检测算法在 MS COCO 数据集上以相同的训练参数进行实验对比，对比的算法及结果如表 6 所列。其中，*Size* 代表输入图像的尺寸。可以看到，得益于特征提取网络对多个不同尺度特征感受野的增强，本文算法对目标的检测能力得到了有效提高。本文改进的骨干网络在 *mAP* 上有较明显的提升，且相比使用原生 ResNeSt 作为特征提取网络的检测方法 *mAP* 也有一定的提升。

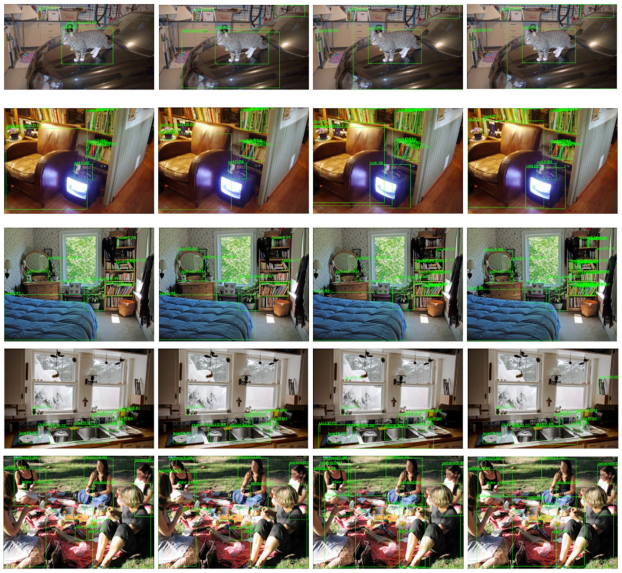
表 6 在 MS COCO 数据集上与不同算法的对比

Table 6 Comparison of different algorithms on MS COCO

	Backbone	*Size*	*mAP*
Faster R-CNN[9]	ResNet-50	1 000×600	34.0
Cascade R-CNN[34]	ResNet-50	1 000×600	35.8
CenterNet[35]	ResNet18+DCN	512×512	30.5
YOLOv3[36]	DarkNet-53	608×608	33.7
SSD512[28]	VGG16	512×512	29.7
RetinaNet[8]	ResNet-50	1 000×600	32.9
TridentNet[31]	ResNet-50	1 000×600	35.5
Faster R-CNN	ResNeSt-50	1 000×600	36.2
RetinaNet	ResNeSt-50	1 000×600	35.0
Faster R-CNN	Ours	1 000×600	36.4
RetinaNet	Ours	1 000×600	35.7

另外，为了更加直观地体现本文改进模型在检测物体精度方面的进步，用经过 Faster R-CNN 算法训练好的模型对

MS COCO 测试集中的图片进行测试，将 Bbox 分数的阈值设定为 0.6，并使用随机的图片展示对比效果，如图 7 所示。使用原生 ResNet-50 和 PS-ResNet-50 作为骨干网络存在多处漏检的情况，而原生的 ResNeSt-50 对于图片中的小物体存在部分错检和漏检的情况。相比之下，本文提出的算法得益于拆分注意力网络及多尺度感受野融合模块的加入，能较好地利用上下文信息，并且这也体现在其对复杂场景小物体的检测比原有算法具有更好的表现力。



(a)ResNet-50　(b)PS-ResNet-50　(c)ResNeSt-50　(d)PS-ResNeSt-50

图 7　不同算法训练模型的检测效果对比

Fig. 7　Comparison of detection effect of different models

结束语　本文基于残差网络结构，结合多通道结构、拆分注意力机制以及多尺度卷积提出了一种基于改进的拆分注意力网络的目标检测算法框架，在不增加额外超参数和计算复杂度的情况下提高了卷积神经网络缩放尺度变化的鲁棒性，并充分发挥了膨胀卷积的特性，进一步增强了特征图的全局感受野。最后进行实验评估，在 Pascal VOC 数据集训练的模型上，本文改进的算法模型在大多数类别的 AP 值均获得了提升；在 MS COCO 数据集训练的模型上，本文的算法模型与原始算法及其他检测算法相比取得了更好的检测结果，表明本文方法展现了改进算法框架的性能优势。但所提算法对小尺度物体的检测提升不明显，未来的工作将继续探索对小尺度物体的检测精度的提高以及目标旋转等问题，并将算法框架通过迁移学习应用于实例分割、姿态识别等其他计算机视觉领域的下游应用中。

参 考 文 献

[1] CHEN L, MA N, PANG G L, et al. Research on multi-view data fusion and balanced YOLOv3 for pedestrian detection[J]. CAAI Transactions on Intelligent Systems, 2021, 16(1): 57-65.

[2] YUAN Z H, SUN Q, LI G X, et al. Automatic Driving Target Detection Based on Yolov3[J]. Journal of Chongqing University of Technology(Natural Science), 2020, 34(9): 56-61.

[3] HE Z H, HUANG S, RAN G, et al. An Improved Visual Background Extractor Model for Moving Objects Detection Algorithm[J]. Journal of Chinese Mini-Micro Computer Systems, 2015, 36(11): 2559-2562.

[4] HE K, GKIOXARI G, DOLLÁRP, et al. Mask r-cnn[C] // Proceedings of the IEEE International Conference on Computer Vision. Venice, 2017: 2961-2969.

[5] LI J W, ZHOU X L, CHAN S X, et al. A Novel Video Target Tracking Method Based on Adaptive Convolutional Neural[J]. Journal of Computer-Aided Design & Computer Graphics, 2018, 30(2): 273-281.

[6] ZOU Z, SHI Z, GUO Y, et al. Object detection in 20 years: A survey[J]. arXiv: 1905.05055, 2019.

[7] REDMON J, DIVVALA S, GIRSHICK R, et al. You Only Look Once: Unified, Real-Time Object Detection[C] // Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016: 779-788.

[8] LIN T Y, GOYAL P, GIRSHICKR, et al. Focal loss for dense object detection[C] // Proceedings of the IEEE International Conference on Computer Vision. Venice, 2017: 2980-2988.

[9] REN S, HE K, GIRSHICKR, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6): 1137-1149.

[10] DAI J, LI Y, HE K, et al. R-fcn: Object detection via region-based fully convolutional networks[C] // Advances in Neural Information Processing Systems. Barcelona, 2016: 379-387.

[11] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas, 2016: 770-778.

[12] ZHANG H, WU C, ZHANG Z, et al. Resnest: Split-attention networks[J]. arXiv: 2004.08955, 2020.

[13] LONG J, SHELHAMER E, DARRELLT. Fully Convolutional Networks for Semantic Segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 39(4): 640-651.

[14] LI D, YAO A, CHEN Q. PSConv: Squeezing feature pyramid into one compact poly-scale convolutional layer[C] // Computer Vision-ECCV 2020. Glasgow, 2020: 615-632.

[15] LIN T Y, MAIRE M, BELONGIE S, et al. Microsoft coco: Common objects in context[C] // Computer Vision-ECCV 2014. Zurich, 2014: 740-755.

[16] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[J]. Advances in Neural Information Processing Systems, 2012, 25: 1097-1105.

[17] SIMONYAN K, ZISSERMANA. Very deep convolutional networks for large-scale image recognition[J]. arXiv: 1409.1556, 2014.

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

[19] SZEGEDY C, VANHOUCKE V, IOFFES, et al. Rethinking the inception architecture for computer vision[C] // Proceedings of the IEEE Conference on Computer Vision and Pattern Recogni-

tion. Las Vegas, 2016: 2818-2826.
[20] GIRSHICK R, DONAHUE J, DARRELLT, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Columbus, 2014: 580-587.
[21] GIRSHICK R. Fast r-cnn[C]// Proceedings of the IEEE International Conference on Computer Vision. Santiago, 2015: 1440-1448.
[22] REN S, HE K, GIRSHICKR, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6): 1137-1149.
[23] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely connected convolutional networks[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, 2017: 4700-4708.
[24] HU J, SHEN L, SUN G. Squeeze-and-excitation networks[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City, 2018: 7132-7141.
[25] XIE S, GIRSHICK R, DOLLÁR P, et al. Aggregated residual transformations for deep neural networks[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Ho-nolulu, 2017: 1492-1500.
[26] LI X, WANG W, HU X, et al. Selective kernel networks[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, 2019: 510-519.
[27] LIN T Y, DOLLÁR P, GIRSHICK R, et al. Feature pyramid networks for object detection[C]// Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Honolulu, 2017: 2117-2125.
[28] LIU W, ANGUELOV D, ERHAND, et al. Ssd: Single shot multibox detector[C]// Computer Vision-ECCV 2016. Amsterdam. 2016: 21-37.
[29] SUN S, PANG J, SHI J, et al. Fishnet: A versatile backbone for image, region, and pixel level prediction[J]. arXiv: 1901. 03495, 2019.
[30] CHEN C F, FAN Q, MALLINAR N, et al. Big-little net: An efficient multi-scale feature representation for visual and speech recognition[J]. arXiv: 1807. 03848, 2018.
[31] LI Y, CHEN Y, WANG N, et al. Scale-aware trident networks for object detection[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. Long Beach, 2019: 6054-6063.
[32] DAI J, QI H, XIONG Y, et al. Deformable convolutional networks[C]// Proceedings of the IEEE International Conference on Computer Vision. Venice, 2017: 764-773.
[33] TAN M, LE Q V. Mixconv: Mixed depthwise convolutional kernels[J]. arXiv: 1907. 09595, 2019.
[34] CAI Z, VASCONCELOS N. Cascade R-CNN: High Quality Object Detection and Instance Segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(5): 1483-1498.
[35] DUAN K, BAI S, XIE L, et al. Centernet: Keypoint triplets for object detection[C]// Proceedings of the IEEE/CVF International Conference on Computer Vision. Long Beach, 2019: 6569-6578.
[36] REDMON J, FARHADI A. Yolov3: An incremental improvement[J]. arXiv: 1804. 02767, 2018.

PAN Yi, born in 1996, postgraduate. His main research interests include object detection and multi-objective optimization.

WANG Li-ping, born in 1964, Ph. D, professor, Ph.D supervisor, is a member of China Computer Federation. Her main research interests include computing intelligence and decision optimization.

（责任编辑：柯颖）