计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(2):379−386 [doi: 10.15888/j.cnki.csa.008975]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

融合空洞卷积的轻量化目标检测①

李　洋, 苟　刚

(贵州大学 计算机科学与技术学院 公共大数据国家重点实验室, 贵阳 550025)
通信作者: 苟　刚, E-mail: 6706605@qq.com

摘　要: 为了轻量化模型, 便于移动端设备的嵌入, 对 YOLOv4 网络进行了改进. 首先, 用 MobileNetV3 作为主干网络, 并使用深度可分离卷积替换加强特征提取网络的普通卷积, 降低模型参数量; 其次, 在 104×104 特征图输出时融合空洞率为 2 的空洞卷积, 与 52×52 的特征层进行特征融合, 获取更多的语义信息和位置信息, 细化特征提取能力, 提升模型对极小目标的检测性能; 最后, 将原来的池化层使用 3 个 5×5 的 Maxpool 进行串联, 减少计算量, 提升检测速度. 实验结果表明, 在华为云 2020 数据集上, 改进算法的 *mAP* 比 YM 算法提高了 2.33%, 在公共数据集 VOC07+12 上, *mAP* 提高了 3.12%, FPS 比原来的 YOLOv4 算法提高了一倍多, 参数量降低至原来的 18%, 证明了改进算法的有效性.

关键词: MobileNetV3; YOLOv4; 空洞卷积; 轻量化; 深度可分离卷积

引用格式: 李洋,苟刚.融合空洞卷积的轻量化目标检测.计算机系统应用,2023,32(2):379–386. http://www.c-s-a.org.cn/1003-3254/8975.html

Lightweight Target Detection Based on Dilated Convolution

LI Yang, GOU Gang

(State Key Laboratory of Public Big Data, College of Computer Science and Technology, Guizhou University, Guiyang 550025, China)

Abstract: In order to make the model lightweight and facilitate the embedding of mobile devices, the YOLOv4 network is improved. Firstly, MobileNetV3 is used as the backbone network, and a deep separable convolution is adopted to replace the ordinary convolution of an enhanced feature extraction network, so as to reduce the number of model parameters. Secondly, when the feature map with a size of 104×104 is output, the dilated convolution with a dilated rate of 2 is fused, and it is then fused with a feature layer with a size of 52×52, so as to obtain more semantic and location information, which can refine the feature extraction ability and improve the detection performance of the model for minimal targets. Finally, the original pooling layer is connected in series with three Maxpools with a size of 5×5 to reduce the computational load and improve the detection speed. The experimental results show that on Huawei Cloud 2020 dataset, the *mAP* of the improved algorithm is improved by 2.33% compared with the YM algorithm, and on the public dataset VOC07 + 12, the *mAP* is improved by 3.12%, and the FPS has more than doubled compared with the original YOLOv4 algorithm, with the number of parameters reduced to 18% of the original one. As a result, the effectiveness of the improved algorithm is verified.

Key words: MobileNetV3; YOLOv4; dilated convolution; lightweight; depth-separable convolution

随着计算机视觉研究的不断发展, 目标检测在近年来发展迅速, 被应用于各行各业, 比如行人检测[1]、车辆检测、自动驾驶、农作物杂草识别等, 越来越多的研究者开始关注目标检测算法, 但是在目标物体遮

① 基金项目: 国家自然科学基金 (62162010); 贵州省科技支撑计划 (黔科合支撑 [2022] 一般 267)
收稿时间: 2022-07-13; 修改时间: 2022-09-07; 采用时间: 2022-09-16; csa 在线出版时间: 2022-12-23
CNKI 网络首发时间: 2022-12-27

挡、光照变化、图像位置变换等目标检测上, 检测效果差强人意, 且现有的网络结构复杂, 对计算能力要求高, 难以嵌入移动端设备广泛使用, 这值得我们进一步深入研究.

目标检测分为传统方法和深度学习方法. 传统的目标检测算法是利用手工特征和分类器, 以滑窗方式在图像金字塔上遍历所有位置和大小, 进行目标检测. 深度学习包括了无需锚框的关键点法、中心域法以及基于锚框的单阶段法、多阶段法, 其中单阶段算法包括 SSD[2]、YOLO[3–6] 系列算法、Retinanet[7] 等, 单阶段 (one stage) 算法是通过预设一系列不同大小的锚框, 将图像输入卷积神经网络, 利用区域生成网络 (region proposal network, RPN) 对 anchors 进行分类回归, 得到候选区域; 双阶段 (two stage) 算法则还需继续利用对图像的感兴趣区域 (region of interest, ROI) 池化提取候选区域的特征, 将提取的特征输入 R-CNN 网络, 进一步对候选区域分类回归, 如 R-CNN[8,9] 系列算法. 所以 one stage 算法的检测速度更快.

为进一步提高目标检测的精度和检测速度, 2020 年 Bochkovskiy 等[6] 提出 YOLOv4 算法, 在 YOLOv3 主干网络 Darknet53 的每个大残差块上加入跨阶段局部网络 (cross stage partial, CSP)[10] 结构, 并在 13×13 的输出引入空间金字塔池化 (spatial pyramid pooling, SPP)[11] 增加网络的感受野, 改进 YOLOv3 的特征图金字塔网络 (feature pyramid network, FPN), 用路径聚合网络 (path aggregation network, PANet)[12] 作为加强特征提取网络, 进行特征融合, 使用下采样的方法融合不同维度的语义信息特征.

文献 [13] 中, 构建以 MobileNetV2 为核心的轻量级特征提取网络, 利用通道和空间注意力机制增强网络对特征的细化能力; 多尺度特征融合结构, 增强网络对尺度的适应性, 提高模型精度. 文献 [14] 将 Darknet-53 的第 2 个残差块输出的特征图混合空洞卷积, 与 YOLOv3 中 8 倍下采样的特征图融合, 使用 Focal Loss 损失函数改进负样本的置信度公式, 在 VOC 数据集上, 精度达到 81.5%. 文献 [15] 为解决串联操作只是将通道维度上不同尺度特征融合, 不能反映通道间特征相关性的问题, 提出一种基于注意力机制的特征融合算法, 对通道特征进行权重的重新分配, 使用 Focal Loss 和 GIOU Loss 重新设计损失函数, 在 VOC 数据集上, 精度达到 82.69%.

由于发展的需要, 复杂的目标检测网络不适应当前社会的发展, 难以进行移动端设备的嵌入, 轻量级的网络应运而生. 本文借鉴前人的研究方法, 改进 YOLOv4 算法, 使用轻量级的 MobileNetV3[16] 网络替换 YOLOv4 的主干网络部分; 为提升网络模型的检测精度, 对特征不明显的小目标的检测能力, 引入空洞卷积, 扩大感受野, 将主干网络 104×104 的特征图卷积后添加空洞卷积, 与 52×52 输出的特征图进行特征融合, 得到的新特征包含更多语义信息和位置信息, 能提升模型的检测能力; 改进 SPP 结构, 将 3 个 Maxpool 串联后再特征融合, 降低计算量; 在 VOC07+12 的数据集上进行检测, 改进模型的精度达到 87.32%, 与使用 MobileNetV3 作为主干网络的轻量化 YM 模型相比 *mAP* 提升了 3.12%.

1 相关工作

1.1 YOLOv4 算法

YOLOv4 是在 YOLOv3 的基础上进行改进的, 在 Darknet-53 的每个 residual block 加上 CSP 结构, 取消 bottleneck 结构, 使模型更容易训练, CSP 的结构图如图 1 所示.

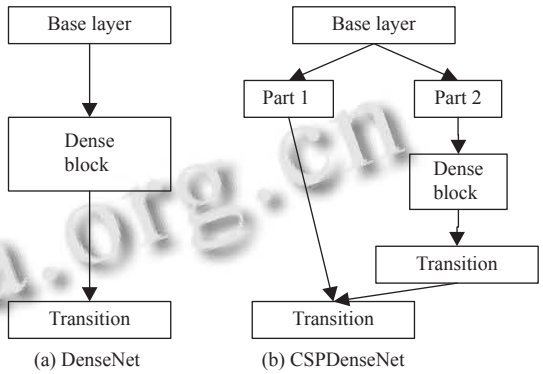


图 1 CSP 结构图

输出 3 个特征尺度检测层, 其中 52×52 用于检测小目标物体, 26×26 用于检测中目标物体, 13×13 用于检测大目标物体; 另外, 增加了 SPP 层, 增强网络的感受野; 利用 YOLO-Head 输出 3 个检测头, 该算法的每个 YOLO-Head 都包含了 3 个先验框, 每个先验框包含了中心点 (x, y)、宽 w、高 h、置信度 *confidence* 五个参数, 输出计算如式 (1) 所示, 通过调整这些参数生成最优值, *num_anchors* 代表先验框的个数, *num_classes* 代表数据集的种类.

$$out = num\_anchors \times (5 + num\_classes) \quad (1)$$

1.2 损失函数

YOLOv4 损失函数的计算包括 3 部分: 回归损失、置信度损失、分类损失.

IoU (intersection over union) 是目标检测算法中常用的指标, 用于计算预测框和真实框之间交集与并集的比值, 如式 (2) 所示, A 代表预测边框, B 代表真实边框.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

CIOU 计算如式 (3) 所示, ρ 表示预测框与真实框中心点间的欧氏距离, p、g 分别表示预测框和真实框的中心点, c 表示同时包含了预测框和真实框最小闭包区域对角线的距离. 影响因子 αv 拟合了预测框与真实框的纵横比.

$$CIOU = 1 - IoU + \frac{\rho^2(p,g)}{c^2} + \alpha v \tag{3}$$

YOLOv4 的损失函数如式 (4) 所示. $K \times K$ 表示对所有预测框遍历, M=3, 代表每层有 3 个先验框, λ_{coord} 表示正样本权重系数, I_{ij}^{obj}表示第 i 个网格第 j 个检测框是否存在目标, 存在则为正样本, 否则判定为负样本, w_i、h_i 表示预测框中心点的宽和高, $\widehat{C}_i$表示样本值, C_i 表示预测值 , λ_{noobj} 表示负样本的权重系数, $\widehat{p_i}(c)$表示预测框类别概率, $p_i(c)$表示真实框类别概率.

$$\begin{aligned} Loss =& \lambda_{\text{coord}} \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{\text{obj}}(2 - w_i \times h_i)(1 - CIOU) \\ & - \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{\text{obj}}\left[\hat{C}_i \log(C_i) + (1-\hat{C}_i)\log(1-C_i)\right] \\ & - \lambda_{\text{noobj}} \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{\text{noobj}}\left[\hat{C}_i \log(C_i) + (1-\hat{C}_i)\log(1-C_i)\right] \\ & - \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{\text{obj}} \sum_{c\in classes} [\hat{p}_i(c)\log(p_i(c)) \\ & + (1-\hat{p}_i(c))\log((1-p_i(c))] \end{aligned} \tag{4}$$

1.3 MobileNetV3 算法

2017 年, 针对手机等嵌入式设备, Google 提出了轻量级网络 MobileNet[17], 其基本组成是深度可分离卷积, 它将传统卷积分为了深度卷积和点卷积两步; 2018 年, 在 MobileNetV1 的基础上引入了逆残差块和线性瓶颈, 提出了 MobileNetV2 网络[18]. 2019 年, Google 团队提出了 MobileNetV3, 在 ImageNet 的分类任务中正确率提高了 3.2%, 计算延时 20%, 主要改进了以下几点: 激活函数更新, SE 模块的引入, 修改尾部结构.

在 bottleneck 结构中的 depthwise filter 之后加入了注意力机制, 使通道数变为了原本来的 1/4, 对通道重新分配权重, 不仅降低了时间损耗, 还提高了精度; 对第 1 层卷积层, 由于卷积核比较大, 所以将第 1 层卷积核个数从 32 改为 16, 虽然准确率没有改变, 但是大大降低了参数量, 节省了 2 ms 的时间, 之后将 MobileNetV2 中平均池化前的 1×1 卷积层移动到了平均池化之后, 节省了 7 ms 的时间, 提高了特征图的维度, 降低了计算量.

由于 MobileNetV3 具有更准确、高效, 参数量更少, 实时性更高的特点, 改进后的结构如表 1 所示, SE 表示在该块中是否存在 SE 注意力模型, NL 表示所使用的非线性的类型, HS 表示 *h-swish*, RE 表示 *ReLU*, NBN 表示没有批量规范化, *s* 表示步幅.

表 1 MobileNetV3 网络结构表

Input	Operator	exp size	*#out*	SE	NL	*s*
$224^2\times3$	conv2d	—	16	—	HS	2
$112^2\times16$	bneck, 3×3	16	16	—	RE	1
$112^2\times16$	bneck, 3×3	64	24	—	RE	2
$56^2\times24$	bneck, 3×3	72	24	—	RE	1
$56^2\times24$	bneck, 5×5	72	40	√	RE	2
$28^2\times40$	bneck, 5×5	120	40	√	RE	1
$28^2\times40$	bneck, 5×5	120	40	√	RE	1
$28^2\times40$	bneck, 3×3	240	80	—	HS	2
$14^2\times80$	bneck, 3×3	200	80	—	HS	1
$14^2\times80$	bneck, 3×3	184	80	—	HS	1
$14^2\times80$	bneck, 3×3	184	80	—	HS	1
$14^2\times80$	bneck, 3×3	480	112	√	HS	1
$14^2\times112$	bneck, 3×3	672	112	√	HS	1
$14^2\times112$	bneck, 5×5	672	160	√	HS	2
$7^2\times160$	bneck, 5×5	960	160	√	HS	1
$7^2\times160$	bneck, 5×5	960	160	√	HS	1

MobileNetV3 网络结构中, 激活函数, 使用 *h-swish* 函数替换原来的 *swish* 函数, 计算公式如式 (5), 能降低运算量, 提高模型性能. ReLU6 是普通的 ReLU 函数, 但是限制其最大值只能为 6, 能够使移动设备端在低精度时有很好的数值分辨率.

$$h\text{-}swish[x] = x \cdot \frac{\text{ReLU6}(x+3)}{6} \tag{5}$$

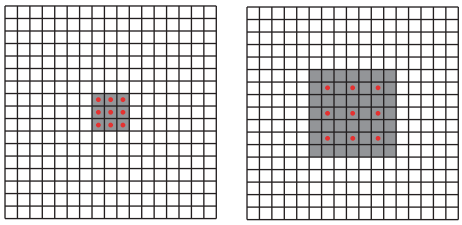
2 改进的轻量化模型

本文使用轻量化网络 MobileNetV3 作为模型的主

干特征提取网络, 并使用深度可分离卷积替换普通卷积, 极大降低了参数量; 引入空洞卷积增强浅层特征图的感受野, 与深层特征图的细粒度特征融合, 获取更多小目标特征, 提高模型对小目标的检测能力; 改进 SPP 层, 将原来的池化层并联改成串联方式, 降低了计算量, 提高了推理速度.

2.1 空洞卷积

由于在图像分类网络中池化和子采样整合多尺度上下文信息时会降低图像分辨率, 丢失部分特征信息, 2016 年, Yu 等人在文献 [19] 中提出空洞卷积 (dilated convolution), 利用在卷积层之间添加空洞增大感受野, 使原 3×3 的标准卷积, 在参数量和计算量不变的情况下, 拥有 5×5 (dilated rate =2) 或者更大的感受野, 在不丢失多尺度上下文信息的情况下系统地聚合多尺度上下文信息, 不需要再进行下采样操作. 而且空洞卷积不仅能增大感受野, 还能保证输入输出特征图的 W、H 不变. 图 2(a) 表示空洞率为 1 时的普通卷积, 图 2(b) 表示空洞率为 2 的卷积.



(a) dilated rate=1 (b) dilated rate=2

图 2 空洞率分别为 1 和 2 的空洞卷积

2.2 深度可分离卷积

在 PANet 网络中利用深度可分离卷积替换了原来的普通卷积, 深度可分离卷积分为深度卷积和点卷积, 如图 3 所示.

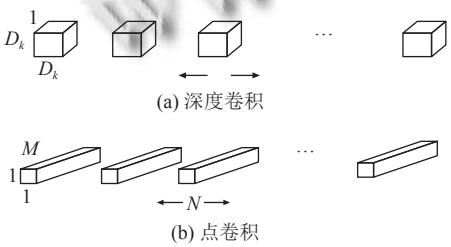


图 3 深度可分离卷积

先用 1×1 卷积升维, 再用 3×3 提取特征, 最后用 1×1 卷积降维, 虽然加深了网络层数, 但是减少了内存占用时间, 且保存了更多的特征信息, 用于提取特征图 (feature map), 卷积核的个数等于通道数, 相比于普通卷积参数数量更小, 计算速度快. 深度卷积的数学表达式如式 (6) 所示:

$$G_{i,j,m}=\sum_{w,h}^{W,H}K_{w,h,m}\times X_{i+w,j+h,m} \tag{6}$$

其中, G 表示输出的特征图, i 和 j 表示输出特征图在第 m 通道上的坐标, m 表示特征图的第 m 个通道, W 表示宽, H 表示高, w 和 h 表示第 m 通道的卷积核权重元素坐标, K 表示宽高的卷积核.

2.3 本文算法网络结构

为了使网络轻量化, 使用 MobileNetV3 作为主干网络, 同时在 PANet 网络使用深度可分离卷积替换普通卷积, 网络参数量降低了 82%; 其次为提高模型对小目标的检测能力, 获得更多语义信息, 使用 104×104 输出的特征信息, 如果直接进行特征融合, 104×104 特征图的感受野较小, 不利于目标检测, 为了扩大感受野, 同时保证所有信息的充分利用, 在 104×104 特征图输出时融合空洞卷积, 混合批归一化和 ReLU 激活函数使用, 将提取到的 104×104 特征图的浅层信息进行与 52×52 特征图的深层特征融合, 提升小目标的检测性能; 最后, 改进 SPP 结构, 将原来 Maxpool2d 的并联操作改为串行结构, 由于将两个核大小 (kernel size) 为 5×5 的 Maxpool2d 串联, 相当于一个 kernel size 为 9×9 大小的 Maxpool2d, 将 3 个 kernel size 为 5×5 大小的 Maxpool2d 串联, 相当于一个 kernel size 为 13×13 大小的 Maxpool2d, 而且 5×5 大小的明显比 9×9、13×13 的计算量更小, 所以, SPP 的 3 个最大池化改用 3 个 5×5 的最大池化串联, 改进的 YOLOv4 网络结构如图 4 所示, Conv 表示普通卷积, Batch Norm 表示批归一化, Leaky ReLU 表示激活函数, K 代表卷积核的大小, s 代表步距, DepthConv 代表深度可分离卷积.

3 实验结果与分析

3.1 实验环境与数据集

实验环境: 深度学习框架 TensorFlow, 操作系统 Windows 10. CPU 是 Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHz, GPU 是 GeForce RTX 2080Ti.

实验参数设置如表 2 示, 使用 Adam 优化器, 交叉熵损失函数, warmup_learning_rate 表示余弦退火函数的学习率, 冻结层的 batch_size 设置为 8, 非冻结层的 batch_size 设置为 4.

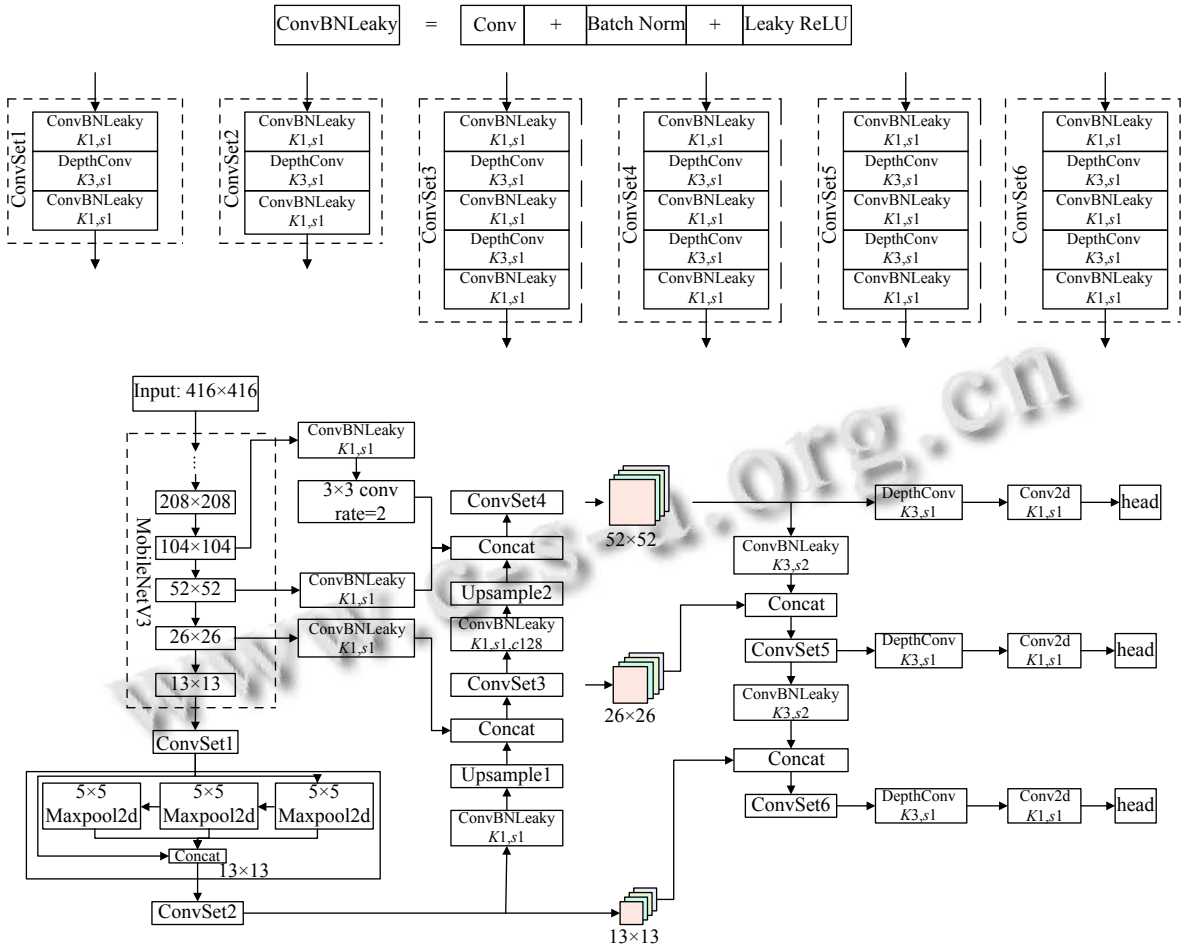


图 4 本文算法结构图

表 2 训练参数

参数	值
size	416×416
optimizer_type	Adam
warmup_learning_rate	1E–4
ignore_thresh	0.5
learning_rate	1E–3

数据集: 为了验证本文算法的有效性, 使用了两种数据集验证本文算法: (1) 华为云 2020 数据集; (2) VOC07+12 的数据集; 数据集训练集与测试集的划分按照 8:2 展开.

3.2 评价指标

Precision 表示预测正确的样本占预测结果的比例, *Recall* 表示预测正确的样本占检测框的比例, *TP* 表示正样本被正确分为正类, *FP* 表示负样本被错误分为正类, *FN* 表示正样本被错误分为负类, *TN* 表示负样本被正确分为负类, *AP* 表示计算一个类别的结果, *mAP* 表示计算所有类的均值. *FPS* (frame per second) 表示每秒帧数, 准确率和召回率的计算公式如式 (7) 和式 (8) 所示:

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

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

3.3 模型训练

本文采用迁移学习的方式, 使用原来的预训练权重进行训练. 分为两个阶段进行训练, 第 1 阶段冻结主干网络进行训练, 第 2 阶段解冻主干网络, 对所有网络训练. 损失值的大小用于判断模型是否收敛, 当验证集损失逐渐趋于稳定, 表示模型基本收敛.

本文算法在华为云数据集上的损失曲线如图 5 所示, 在 VOC 数据集上的损失曲线如图 6 所示, 横轴 Epoch 表示迭代批次, 纵轴 *Loss* 表示验证损失值. YM 表示 YOLOv4+MobileNetV3 算法, our 表示本文算法.

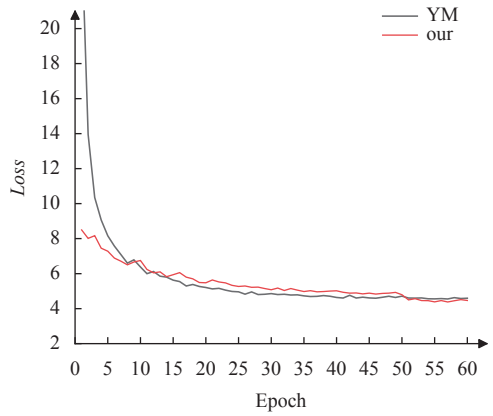


图 5 华为云数据集的损失曲线图

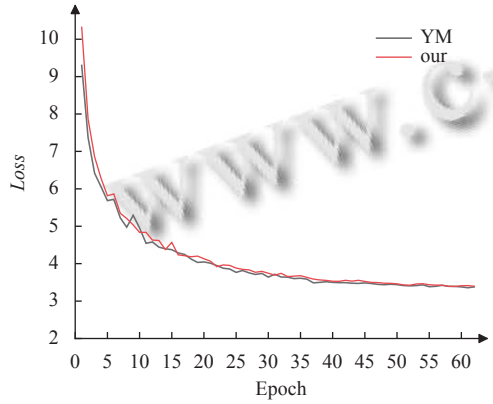


图 6 VOC 数据集的损失曲线图

从图 5 和图 6 中两个数据集训练的损失值曲线可知, 改进后的模型, 随着不断训练的过程, 逐渐收敛、趋于稳定.

3.4 实验结果对比

3.4.1 华为云数据集实验对比

为验证本文算法对轻量化目标检测的性能, 在华为云数据集上进行对比实验, YM 算法与改进算法的检测结果如表 3 所示. *F*1 表示准确率和召回率的调和平均值, *Recall* 表示召回率. 与 YM 算法相比, 改进算法的 *mAP* 提高了 2.33%, *F*1 提高了 4.62%, *Recall* 提高了 3.91%. 实验结果表明, 在华为云数据集中检测性能略有提升.

表 3 华为云数据集上的结果对比 (%)

方法	*mAP*	*F*1	*Recall*
YM	66.60	61.93	55.05
our	68.93	66.55	58.96

3.4.2 VOC07+12 数据集实验对比

在 VOC07+12 的公共数据集上进行消融实验. 本文使用 YOLOv4 作为基线模型; 首先进行主干网络的替换, 使其轻量化, 然后在轻量化模型 YOLOv4-MobileNetV3 的基础上继续改进, 主要改进了两个部分. 为了提高网络对小目标物体的检测能力, 融合空洞率大小为 2 的空洞卷积; 将 SPP 层采用串联的方式连接, 实验结果对比如表 4 所示. YM 表示 YOLOv4-MobileNetV3 算法, YM+P 表示在 YM 的基础上只改进 SPP, YM+K 表示只引入空洞卷积, our 表示本文算法. 实验结果表明, 虽然改进算法比重量级网络 YOLOv4 的 *mAP* 值降低了 1.58%, 但是在精度相差不大的情况下, 改进算法的参数量降低了 82%, 使得模型极大轻量化, *FPS* 也从 4 提升至 13, 与替换主干后的轻量化 YM模型相比, YM+P 算法的 *mAP* 比 YM 算法提高了 1.99%, YM+K 算法比 YM 算法提高了 1.41%, 可知, 改进的轻量化算法是有效的.

表 4 消融实验

方法	*FPS*	*mAP* (%)	Param ($\times 10^6$)
YOLOv4	4	88.90	64.10
YM	13	84.20	11.47
YM+P	14	86.19	11.47
YM+K	13	85.62	11.51
our	13	87.32	11.51

表 5 进一步将本文算法与其他文献的方法进行结果对比, 文献 [20] 所使用的方法是基于 YOLOv4-tiny 提出一种自适应非极大抑制的多尺度检测方法, 相比于原算法, *mAP* 提高了 2.84%, 文献 [21] 提出一种使用 K-means 聚类算法、Mish 激活函数对模型调整, *mAP* 提高了 2.81%, 而本文算法 *mAP* 提高了 3.12%.

表 5 与其他改进方法性能提升对比 (%)

方法	提升
文献[20]	2.84
文献[21]	2.81
our	3.12

3.4.3 不同模型的性能对比

为证明本文算法的有效, 选择与经典算法 SSD、Faster R-CNN 等相比较, 实验结果如表 6 所示, 与其他文献的算法相比, 本文算法在性能上略有提升. 虽然改进后的 *mAP* 略低于 YOLOv4, 但是参数量降低至原来的 18%, 而且与轻量化后的 YM 算法相比, 本文算法的 *mAP* 也有一定的提高, 符合本文轻量化的思想, 有一定的应用价值.

原算法与本文算法的检测结果如图 7 所示, 左边表示仅替换主干网络算法的检测结果, 右边表示本文算法的检测结果.

表 6 经典算法实验结果对比

方法	Backbone	Dataset	*mAP* (%)
SSD_300[2]	VGG	VOC07+12	72.4
YOLOv2_416[4]	Darknet-19	VOC07+12	76.8
Faster R-CNN[9]	VGG	VOC07+12	73.2
HDC+FL[14]	Darknet-53	VOC07+12	81.5
MLKP[22]	ResNet-101	VOC07+12	80.6
DetNAS[23]	ResNet-101	VOC07+12	80.1
Auto-FPN[24]	ResNet-50	VOC07+12	81.8
YOLOv4	CSP-Darknet-53	VOC07+12	88.9
YM	MobileNetV3	VOC07+12	84.2
our	MobileNetV3	VOC07+12	87.3

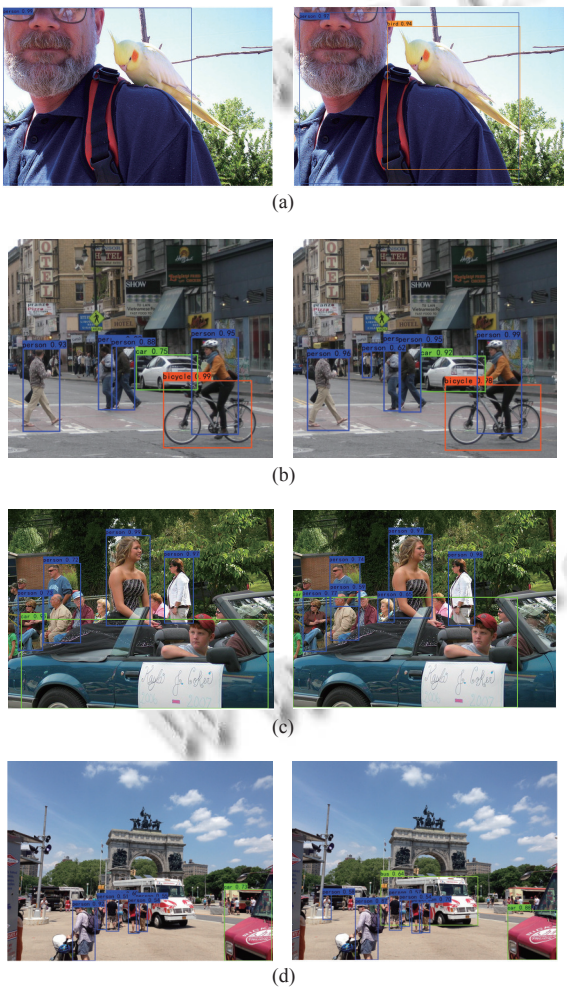


图 7 YM 与本文算法的检测结果图

由实验结果可知, 图 7(a) 中改进算法能检测到原算法未检测到的鹦鹉, 图 7(b) 中改进算法能检测到远处的目标, 图 7(c) 能检测到汽车后面只有一个头的老太太, 图 7(d) 能检测到一些远处的目标, 以及被遮挡一部分的目标, 检测性能有所提升, 证明了本文改进算法的有效性.

4 总结

本文提出了一种轻量化目标检测的方法, 能极大降低参数量, 有利于嵌入移动端设备, 检测精度也有一定的提升; 在主干网络 104×104 输出的特征层引入空洞率为 2 的空洞卷积, 并添加批归一化和 ReLU 激活函数, 与 52×52 特征层的特征进行融合, 细化提取到的物体特征; 改进 SPP 池化层的结构, 降低计算量, 以提高模型检测精度.

未来还需进一步探究, 在保证检测精度的条件下, 如何降低模型的参数量, 使模型进一步轻量化.

参考文献

1 赵书, 陈宁. 复杂路面小尺度行人检测综述. 计算机系统应用, 2022, 31(7): 1–11. [doi: 10.15888/j.cnki.csa.008545]

2 Liu W, Anguelov D, Erhan D, *et al*. SSD: Single shot MultiBox detector. Proceedings of the 14th European Conference on Computer Vision. Amsterdam: Springer, 2016. 21–37.

3 Redmon J, Divvala S, Girshick R, *et al*. You only look once: Unified, real-time object detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE, 2016. 779–788. [doi: 10.1109/CVPR.2016.91]

4 Redmon J, Farhadi A. YOLO9000: Better, faster, stronger. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu: IEEE, 2017. 6517–6525. [doi: 10.1109/CVPR.2017.690]

5 Redmon J, Farhadi A. YOLOv3: An Incremental Improvement. arXiv.1804.02767, 2018.

6 Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal speed and accuracy of object detection. arXiv:2004.10934, 2020.

7 Lin TY, Goyal P, Girshick R, *et al*. Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(2): 318–327. [doi: 10.1109/TPAMI.2018.2858826]

8 Girshick R. Fast R-CNN. Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV). Santiago: IEEE, 2015. 1440–1448. [doi: 10.1109/ICCV.2015.

169]
9 Ren SQ, He KM, Girshick R, *et al*. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137–1149. [doi: 10.1109/TPAMI.2016.2577031]
10 Wang CY, Liao HYM, Wu YH, *et al*. CSPNet: A new backbone that can enhance learning capability of CNN. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. Seattle: IEEE, 2020. 1571–1580.
11 He KM, Zhang XY, Ren SQ, *et al*. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904–1916. [doi: 10.1109/TPAMI.2015.2389824]
12 Liu S, Qi L, Qin HF, *et al*. Path aggregation network for instance segmentation. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 8759–8768. [doi: 10.1109/CVPR.2018.00913]
13 陈智超, 焦海宁, 杨杰, 等. 基于改进 MobileNet v2 的垃圾图像分类算法. 浙江大学学报 (工学版), 2021, 55(8): 1490–1499. [doi: 10.3785/j.issn.1008-973X.2021.08.010]
14 许腾, 唐贵进, 刘清萍, 等. 基于空洞卷积和 Focal Loss 的改进 YOLOv3 算法. 南京邮电大学学报 (自然科学版), 2020, 40(6): 100–108.
15 鞠默然, 罗江宁, 王仲博, 等. 融合注意力机制的多尺度目标检测算法. 光学学报, 2020, 40(13): 132–140.
16 Howard A, Sandler M, Chen B, *et al*. Searching for MobileNetV3. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul: IEEE, 2019. 1314–1324.
17 Howard AG, Zhu ML, Chen B, *et al*. MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861, 2017.
18 Sandler M, Howard A, Zhu ML, *et al*. MobileNetV2: Inverted residuals and linear bottlenecks. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 4510–4520. [doi: 10.1109/CVPR.2018.00474]
19 Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions. Proceedings of the 4th International Conference on Learning Representations. San Juan, 2016.
20 王长清, 贺坤宇, 蒋帅. 改进 YOLOv4-tiny 网络的狭小空间目标检测方法. 计算机工程与应用, 2022, 58(10): 240–248. [doi: 10.3778/j.issn.1002-8331.2112-0593]
21 张伟, 刘娜, 江洋, 等. 基于 YOLO 神经网络的垃圾检测与分类. 电子科技, 2022, 35(10): 45–50.
22 Wang H, Wang QL, Gao MQ, *et al*. Multi-scale location-aware kernel representation for object detection. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 1248–1257.
23 Chen YK, Yang T, Zhang XY, *et al*. DetNAS: Backbone search for object detection. Proceedings of the 33rd Conference on Neural Information Processing Systems. Vancouver: Curran Associates Inc., 2019. 596.
24 Xu A, Yao A, Li A, *et al*. Auto-FPN: Automatic network architecture adaptation for object detection beyond classification. Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV). Seoul: IEEE, 2019. 6648–6657. [doi: 10.1109/ICCV.2019.00675]

(校对责编: 孙君艳)