面向移动目标识别的轻量化网络模型

符惠桐，王鹏，李晓艳，吕志刚，邸若海
（西安工业大学电子信息工程学院，710021，西安）

摘要：针对卷积神经网络模型体积大、运算量高，在体积小、资源有限的嵌入式平台上运行效率低，而现有轻量化模型无法兼顾检测速度和检测精度的问题，提出了一种基于 Ghost 模块的 YOLO 目标识别算法 GS-YOLO。以 YOLOv4 模型为基础，基于 Ghost 模块重构目标识别网络，减少模型参数与卷积运算量，提升目标识别速率；通过融入多个空间金字塔池化模块优化目标识别精度；利用通道剪枝极限压缩方法剔除冗余参数，进一步减小模型体积与计算量；利用微调技术提升剪枝后模型的精度。实验结果表明：在自主构建的测试集和相同的测试环境下，与 YOLOv4 相比，GS-YOLO 将 YOLOv4 模型体积压缩 96%，浮点型计算量减少 91.2%，预测速度提升 2.9 倍，压缩后模型识别精度达到 87.63%，精度仅损失 2.43%。

关键词：目标识别；轻量化模型；Ghost 模块；通道剪枝；空间金字塔池化
中图分类号：TP391.9 **文献标志码**：A
DOI：10.7652/xjtuxb202107014 **文章编号**：0253-987X(2021)07-0124-08

OSID 码

Lightweight Network Model for Moving Object Recognition

FU Huitong, WANG Peng, LI Xiaoyan, LÜ Zhigang, DI Ruohai
(School of Electronic and Information Engineering, Xi'an Technological University, Xi'an 710021, China)

Abstract: The convolutional neural network model with large volume and computation amount cannot be run by the embedded platform with small volume and limited resources, and the existing lightweight models are unable to juggle detection speed and accuracy. An object recognition algorithm of YOLO based on Ghost module (GS-YOLO) was proposed in this study. Following YOLOv4 model, the object recognition network was reconstructed based on Ghost module to reduce model parameters and convolution operations and improve object recognition rate. Multiple spatial pyramid pooling modules were incorporated to optimize the identification model and improve the accuracy of model identification. The channel pruning limit compression method was adopted to eliminate the redundant parameters and further reduce the model volume and calculation. The accuracy of pruning model was improved by fine tuning technique. Experimental data show that for the self-built test set and the same test environment, compared with YOLOv4 object recognition algorithm, GS-YOLO algorithm reduces the volume of YOLOv4 model by 96%, reduces the amount of floating point calculation by 91.2%, and increases the prediction speed by 2.9 times. After compression, the model recognition accuracy achieves 87.63%, and the accuracy loss is only 2.43%.

Keywords: object recognition; lightweight model; Ghost module; channel pruning; spatial pyramid pooling

收稿日期：2021-03-08。 作者简介：符惠桐(1996—)，男，硕士生；王鹏(通信作者)，男，教授。 基金项目：国家自然科学基金资助项目(62031021)；西安市科技计划资助项目(2020KJRC0037)；西安工业大学校长基金面上培育项目(XGPY200217)。
网络出版时间：2021-04-08 网络出版地址：http://kns.cnki.net/kcms/detail/61.1069.T.20210407.1727.004.html

目标识别的主要研究内容是从海量图像中高效、准确地定位和识别预定义的特殊目标[1]。深度学习强大的表征能力和建模能力在很大程度上克服了复杂环境、光照变换、物体形变等对识别结果的影响[2-3]。在多种领域，传统的人力活动将被智能化、无人化的智能设备所取代[4]。此外，在实际生活中，可见光图像大多受到光线、遮挡或者背景复杂等因素的影响，目标识别精度降低[5-6]，而且人们也无法携带大型设备来支撑基于深度学习的目标检测算法，将轻量化的目标检测模型植入嵌入式设备才更接近于实际生活的使用要求[7]。目前，大多数的目标检测算法需要 GPU 的加速支持才能在短时间内训练完成深度学习算法，达到实时检测的效果[8]。

随着计算机硬件水平的发展，计算机的算力资源得到极大提升，基于深度学习的目标识别算法发展也日新月异。2017 年，Howard 等提出了轻量级骨干网络 MobileNets，引入深度可分离卷积模块，减少了大量的卷积运算与浮点运算[9]，但网络本身参数少，提取特征能力不足，识别精度较低。2019 年，Lee 等基于 VoVNetV2 骨干网络提出了一种轻量化的无锚点目标检测实时算法 CenterMask-Lite[10]，对骨干网络进行优化，缓解通道信息丢失问题，在 COCO 数据集[11]上均值平均精度（mAP）达到 40.7%，但需要类似于 RTX 2080Ti 显卡的算力支持。同年，Google 公司的 Tan 等提出了可扩展的目标识别模型 EfficientDet，基于神经网络架构搜索技术提出了多种规模模型以适应不同平台[12]，但存在小规模模型精确度不足、大规模模型训练困难的问题。2020 年，Han 等提出了新的轻量化网络 GhostNet，利用线性变化生成更多的特征图，以很小的代价从原始特征中挖掘更多特征信息，在效率以及准确率方面都优于最新的轻量化网络[13]。同年，Bochkovskiy 等汇集主流的优化技巧和更复杂的网络架构设计出 YOLOv4，可以在单显卡上进行快速、精准的训练与检测，是一个实用性极强的目标检测算法[14]，但模型参数量依然很多，模型体积很大。图 1 展示了目前主流算法在 COCO 数据集上检测的准确率与检测速度，算法运行环境均为 Nvidia RTX 2080Ti 显卡，蓝色区域表示检测速度高于 25 帧/s，可以看出，很难有算法可同时满足高检测速度和高检测精度的需求。

针对现有的轻量化模型无法兼顾检测速度和检测精度的问题，本文选取目前主流的目标识别算法 YOLOv4 进行模型轻量化，通过融入多个空间金字塔池化（SPP）模块优化模型性能，并且利用轻量化 Ghost 卷积模块结合通道剪枝极限压缩的方法，对模型进行高效的压缩与加速，获得精度高、体积小、效率高的目标识别模型，从而适合部署于资源有限的嵌入式平台上。

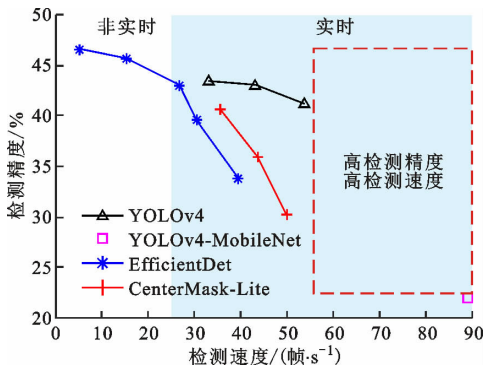


图 1　部分主流目标检测算法性能

Fig. 1　Performance of some mainstream target detection algorithms

1　YOLOv4 目标识别算法

YOLOv4[14]目标识别算法兼顾了检测效率和准确率，是一种实时高效的一阶段目标识别算法。YOLOv4 算法在原有的 YOLO 框架之上，对特征提取骨干网络、用于特征融合的颈部网络以及进行分类回归的预测头输出进行了优化，集成了近年来深度卷积神经网络中优秀的算法和模型。骨干网络在 YOLOv3 原有的 Darknet53 基础之上，结合交叉阶段部分连接（CSP）[15]设计出 CSPDarknet53 特征提取网络，新的骨干网络在减少计算复杂度的同时，保证了网络的识别效果；颈部网络在原有的特征金字塔[16]基础上，结合路径聚合网络[17]的思想，引入 SPP[18]，SPP 模块有效增大了网络的感受野，将局部特征与全局特征进行融合，提升了检测器性能，路径聚合的连接方式有利于特征信息更好地从下向上传递，有效改善了深层网络丢失浅层特征信息的问题；预测头结构没有变化，对损失函数及非极大值抑制算法（NMS）进行了优化。采用 CIOU-Loss 损失函数，综合考虑目标框重叠面积（即交并比 IoU）、中心点距离、长宽比的各项损失，使预测框的回归速度与精度均达到最优。非极大值抑制算法采用DIOU-NMS，综合考虑重合边界框的 IOU 和距离信息，有效提升了重叠目标的检测性能。YOLOv4 的算法框架如图 2 所示。图中：CBL 和 CBM 模块是在卷积层（Conv）后融入批处理归一化（BN）[19]操作以及

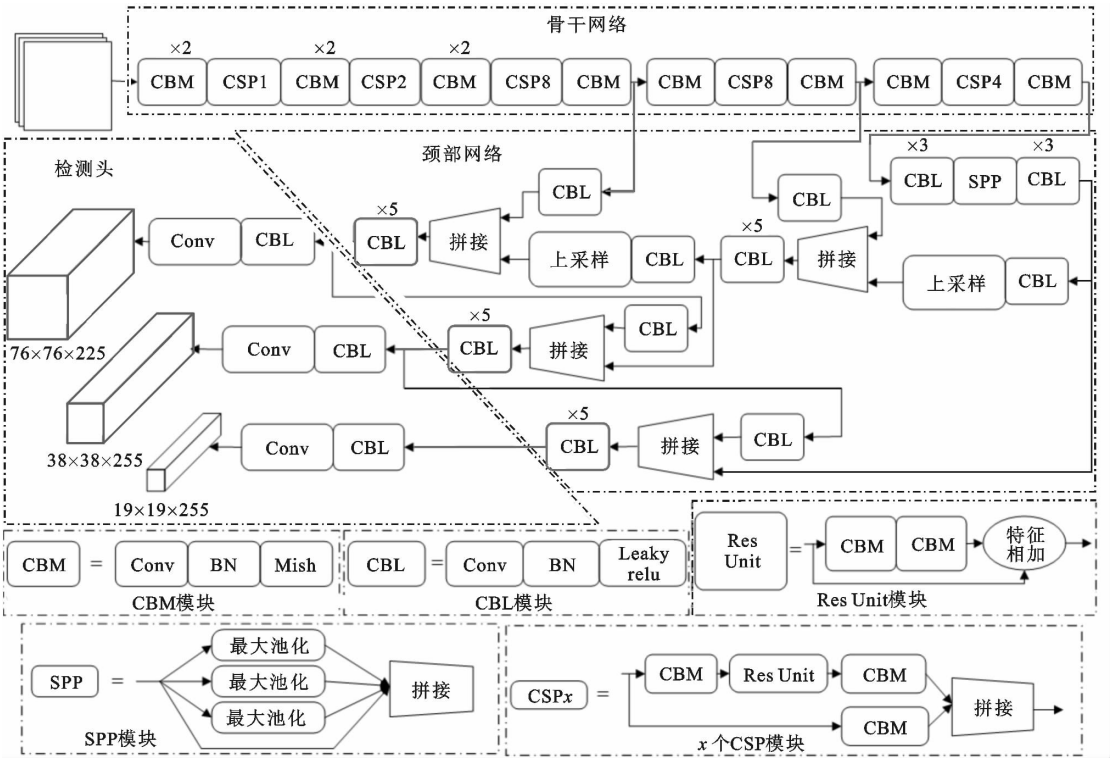


图 2 YOLOv4 算法框架

Fig. 2 YOLOv4 algorithm structure

采用 Leaky relu、Mish 激活函数的卷积模块；拼接操作表示对相同尺寸的特征图进行特征通道的拼接；特征相加表示对具有相同尺寸的特征图进行相加。

2 采用 Ghost 模块的 YOLO 目标识别算法

为压缩 YOLOv4 模型体积，本文提出了一种采用 Ghost 模块的 YOLO 目标识别算法 GS-YOLO。首先基于 Ghost 模块重构目标识别模型；其次融入多个空间金字塔池化模块，在不同尺度上进行特征信息的融合；然后采用通道剪枝极限压缩的方法压缩识别模型；最后利用微调的方法恢复识别精度，从而获得高精度、体积小、运行效率高的轻量化目标识别模型。

GS-YOLO 算法整体流程如图 3 所示。首先，建立初始目标检测网络模型 G-YOLO。采用 Ghost 卷积模块完成卷积操作，以简单的线性映射生成 Ghost 特征图，在保证通道数不变的情况下，减少网络参数和浮点型运算，缩减模型体积，提升计算速度。在模型中融入多个 SPP 模块，在不同的尺度将不同感受野尺寸的特征信息进行融合，丰富特征图信息，从而有效提升检测器的检测精度，完成 G-YOLO 模型的建立。然后，利用通道剪枝极限压缩方法剔除冗余参数，得到紧凑的 GS-YOLO 模型。基于通道稀疏化的方式训练初始模型 G-YOLO 有助于通道剪枝时区分重要信道与非重要信道。利用通道裁剪操作，去除小尺度因子的非重要通道，减小网络模型体积与运算量、防止网络模型过拟合。最后，对剪枝模型进行微调，在保证检测速度的同时，提升模型压缩后目标识别的检测精度。

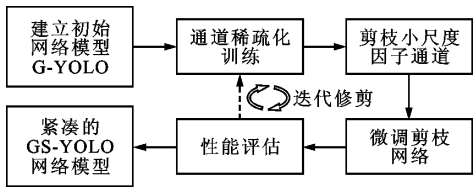


图 3 GS-YOLO 算法流程

Fig. 3 GS-YOLO algorithm flow chart

2.1 基于 Ghost 卷积模块重构识别网络

Han 等发现，特征中的冗余是卷积神经网络成功应用到图像领域的重要原因，因此提出 Ghost 轻量化卷积模块[13]。在进行卷积运算时：输入数据为 $X\in\mathbf{R}^{c\times h\times w}$，其中 c 为输入的通道数，h 和 w 为输入数据的长和宽；卷积核 $f\in\mathbf{R}^{c\times k\times k\times n}$，其中 k 为卷积核尺寸，n 为卷积核数量。通过标准卷积公式得到特征图，公式为

$$Y = X * f + b \tag{1}$$

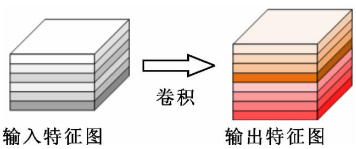
式中：$*$ 为卷积操作；b 为偏置项；$Y \in \mathbf{R}^{h' \times w' \times n}$ 为输出特征图，h' 与 w' 为输出特征的长和宽。标准的卷积操作中浮点型运算量为 $n \times h' \times w' \times c \times k \times k$，卷积操作如图 4a 所示，其中包含着大量相似的冗余特征。在 Ghost 模块中，采用更廉价的线性操作完成冗余特征的生成，减少大量的卷积运算。Ghost 模块首先利用标准卷积生成 m 层通道的本源特征 Y'，$Y' \in \mathbf{R}^{h' \times w' \times m}$，$m$ 为本源特征通道数且 $m \leqslant n$。然后，基于 Y' 利用简单的线性变化得到幻影特征

$$y_{ij} = \phi_{i,j}(y'_i), \quad \forall i = 1,2,\cdots,m, \quad \forall j = 1,2,\cdots,s \tag{2}$$

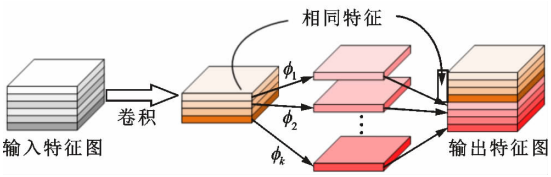
式中：y'_i 为 Y' 中第 i 个本源特征图；$\phi_{i,j}$ 为第 j 次线性运算，用于生成第 j 个幻影特征 y_{ij}，即 y'_i 具有一个或多个幻影特征 $\{y_{ij}\}_{j=1}^{s}$。当 $j=s$ 时，保留本源特征不进行变化，如图 4b 所示。最终，得到 ms 个特征图 $Y=\{y_{11},y_{12},\cdots,y_{ms}\}$ 作为 Ghost 卷积模块的输出，$ms=n$，保证最终特征图通道数不变。所以，浮点型运算量包含 1 个本源特征卷积过程与 $m(s-1)=\dfrac{n}{s(s-1)}$ 个线性操作两部分，线性操作部分采用的卷积核平均尺寸为 $d \times d$，故理论加速比率 R 为

$$R = \frac{nh'w'ckk}{ns^{-1}h'w'ckk + (s-1)ns^{-1}h'w'dd} = \frac{ckk}{s^{-1}ckk + (s-1)s^{-1}dd} = \frac{sc}{s+c-1} \approx s \tag{3}$$

若 $d \times d$ 与 $k \times k$ 相同，且 $s \ll c$，则理论上速度提升为原来的 s 倍。在整个 YOLOv4 的网络中，本文替换所有的传统卷积模块为 Ghost 卷积模块。



(a)标准卷积



(b)Ghost 卷积

图 4　标准卷积与 Ghost 卷积

Fig. 4　Standard convolution and Ghost convolution

2.2　YOLOv4 模型优化

为了更好地增大感受野，融合局部特征与全局特征，YOLOv3 借鉴 SPPNet 网络架构引入了 SPP 模块，以很少的计算量为代价，获得更高的识别准确度。SPP 模块如图 5 所示，图中⌈⌉表示向上取整。

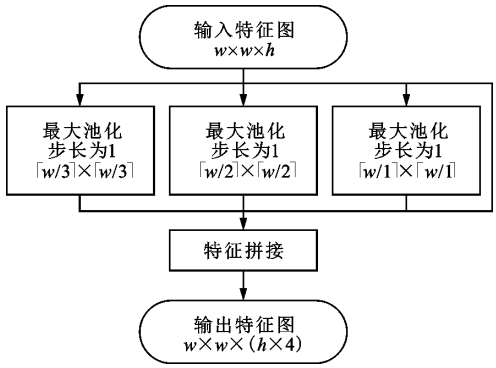


图 5　SPP 模块

Fig. 5　SPP module

输入的特征图通过 3 个不同尺度的内核做最大池化操作，再通过拼接操作将得到的特征图进行拼接。最终，输出特征图的特征通道数为输入特征图通道数的 4 倍。输出特征图尺寸为

$$f_{\text{size}} = \left\lfloor \frac{w + 2p - k}{s_t} \right\rfloor + 1 \tag{4}$$

式中：$p=(k-1)/2$；⌊⌋表示向下取整。由于步长 s_t 为 1，且进行了填充值为 p 的填充操作，由式(4)可知特征图输出尺寸不发生改变。

最大池化操作可以有效地保留特征，SPP 模块相对于固定内核尺寸的最大池化操作扩大了感受野，通过不同尺寸的内核得到了全局特征和局部特征，并进行了特征融合，丰富了特征信息，从而提升了检测性能。

本文综合考虑引入参数以及 SPP 模块的实际应用，在颈部网络末端融入多个 SPP 模块。考虑到参数影响，采用 YOLOv4 中各个内核尺寸，即 13×13、9×9、5×5。此外，将 SPP 模块后一个卷积层的卷积核尺寸由 3×3 修改为 1×1，减少模型参数量。SPP 模块引入位置如图 6 所示。引入多个 SPP 模块后的 YOLOv4 模型称为 YOLOv4-3SPP。

2.3　模型通道稀疏化训练

通道剪枝需要删除对应通道的所有输入与输出连接，Liu 等从通道剪枝的角度出发，提出了基于通道剪枝的网络瘦身方法[20]，通道稀疏化训练可以帮助通道剪枝操作区分重要通道与非重要通道[21]。图 7 是通道稀疏化训练及剪枝，第 i 个卷积层每一个通道分配一个尺度因子 γ，以 γ 的绝对值作为评价该通道重要性的依据。经过稀疏化训练后，尺度因子 γ 的分布向 0 靠近，通过设定阈值剪切掉这些

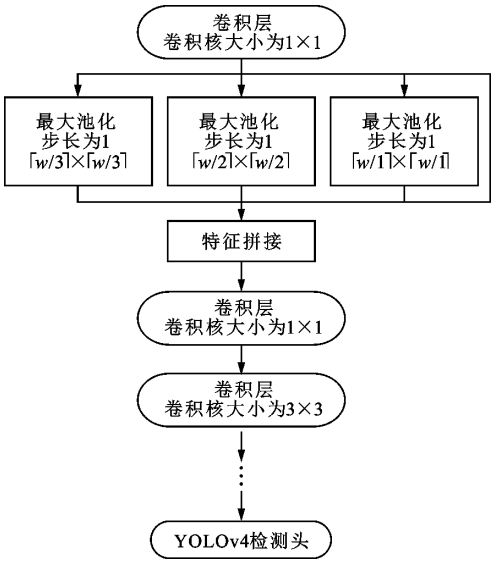


图 6 SPP 模块引入位置

Fig. 6 The location where SPP modules are introduced

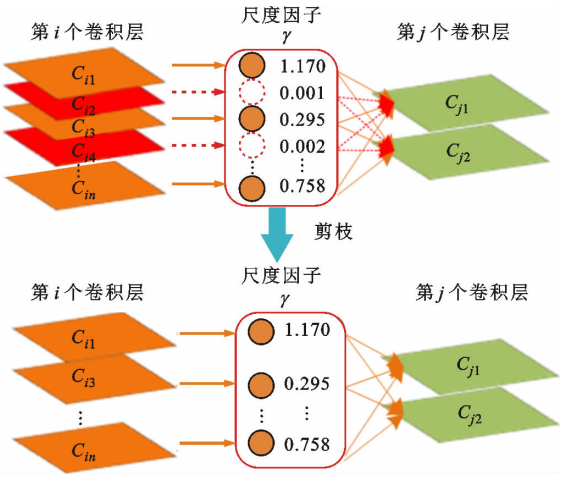


图 7 通道稀疏化训练及剪枝

Fig. 7 Channel sparse training and pruning

通道及连接，进而减少计算量和模型大小。

在 YOLOv4 中，除预测头结构外，每一个卷积层后均有 BN 层。若 y_{in} 和 y_{out} 表示 BN 层的输入和输出，则 BN 层的转换为

$$y_{\text{out}} = \gamma \frac{y_{\text{in}} - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{5}$$

式中：μ 和 σ^2 为小批次输入特征图的均值和方差；β 为偏置；ε 为极小因子，防止分母为 0。在训练时，通过 L_1 正则项对 γ 进行稀疏化训练，稀疏化训练的损失函数为

$$L = L_{\text{yolo}} + \alpha \sum_{\gamma \in \Gamma} f(\gamma) \tag{6}$$

式中：L_{yolo} 表示 YOLO 的损失函数；$\sum_{\gamma \in \Gamma} f(\gamma)$ 表示 L_1 范数，其中 $f(\gamma) = |\gamma|$；α 为惩罚项，用于平衡两个损失项。

2.4 目标检测模型剪枝与微调

完成模型的稀疏化训练后，通过 2.3 小节引入的尺度因子 γ 作为评价该通道重要性的依据。定义一个全局变量 $\hat{\gamma}$ 作为整个尺度因子的阈值，通道对应的尺度因子若低于 $\hat{\gamma}$，则该通道将被去除。如果希望减去 80%的通道，则 $\hat{\gamma}$ 将会低于 80%的 γ，此时整体网络将减少 80%的通道，以此达到加速的目的。此外，大比例的裁剪可能会折损模型的精确度，完成训练后需采用微调的方法补偿精确度的损失，甚至微调后的网络性能可能优于原始网络。

3 实验与分析

3.1 实验环境

实验的测试环境为 Ubuntu 16.04 操作系统。利用 Pytorch 和 Darknet 深度学习框架实现 GS-YOLO 目标识别算法。服务器硬件配置为 Nvidia GTX1060 显卡、Intel Core i7 处理器；性能验证嵌入式平台采用 Nvidia Jetson TX2 移动开发板。

3.2 实验数据集

根据实际应用的需求，本文着重对人、车辆、飞机等目标进行识别。因为研究背景特殊，目前没有公开数据集，故首先进行数据集收集制作。数据集图像主要来源于：内部资料、Imagenet 数据集、标准 COCO 数据集、网络资源下载、影视资源截图等。

通过裁剪、旋转、增加噪声等方法对搜集的数据集进行增强，最终获得30 704张数据样本，包含人、车辆、飞机的样本各20 429、9 877、3 972张。按照 YOLO 算法所需的格式进行人工数据标注，图像样本如图 8 所示。

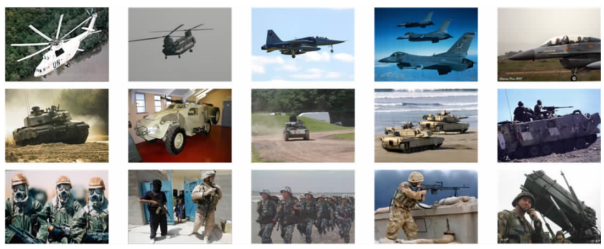


图 8 部分数据样本

Fig. 8 Some data samples

3.3 模型重构及优化实验

基于 YOLOv4 融入多个 SPP 模块得到 YOLOv4-3SPP 模型。在此基础上，基于 Ghost 模块重构得到 YOLOv4-3SPP-Ghost(即 G-YOLO)模型。基于 Ghost 模块重构 YOLOv4-Tiny 得到 YOLOv4-Tiny-

Ghost 模型。表 1 针对这些模型的浮点型运算量、模型体积、预测精度以及在 GTX 1060 和 TX2 平台上的预测时间进行了对比。本文中所有实验模型的图像输入尺寸为 608×608 像素。

从表 1 可以清晰地看出，YOLOv4-3SPP 模型虽然 SPP 模块数量增加，但相较于 YOLOv4 模型浮点型运算量、模型体积以及在相同平台上的预测时间均有所降低，精确度提高 1.16%。将传统卷积模块替换为轻量化的 Ghost 模块后，YOLOv4-3SPP-Ghost 模型的运算量减少至 YOLOv4 模型的 48.88%，模型体积压缩至 47.35%，速度在 GTX 1060 和 TX2 上分别提升 1.4 倍和 1.6 倍，精度提升 0.47%。相较于传统的 YOLOv4-Tiny 模型，YOLOv4-Tiny-Ghost 的效率在 GTX 1060 和 TX2 上分别提升 1.4 倍、1.6 倍，且在 TX2 开发板上达到了 19 帧/s 的检测速度，精度仅下降 1.43%。

表 1　改进 YOLOv4 模型对比

Table 1　Performance comparison of improved YOLOv4 models

模型	浮点型运算量	模型体积/MB	预测精度/%	预测时间/ms	
				GTX 1060	TX2
YOLOv4	128.46	245.8	90.06	95.15	555.56
YOLOv4-3SPP	121.26	227.4	**91.22**	92.08	538.50
YOLOv4-3SPP-Ghost	62.79	116.0	90.53	69.79	346.02
YOLOv4-Tiny	6.79	22.4	72.40	19.08	83.26
YOLOv4-Tiny-Ghost	**3.45**	**11.9**	70.97	**13.77**	**51.91**

注：表中加粗数据为最优值。

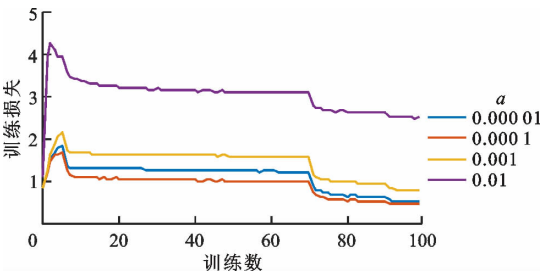
表 1 表明，本小节基于 YOLOv4 所改进的 YOLOv4-3SPP-Ghost 模型相较于传统的 YOLOv4 模型参数更少，识别精度更高。由此证明：在颈部不同尺度融入多个优化后的 SPP 模块可以有效提升目标检测模型精度，对模型的预测效率也有略微提升；采用 Ghost 模块进行模型轻量化操作后，模型的体积、运算量得到有效压缩，并且预测时间也有效降低。

3.4　模型稀疏化实验

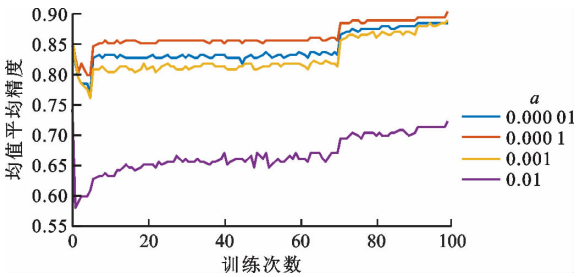
设置不同的惩罚项 α 对 YOLOv4-3SPP-Ghost 进行稀疏化训练，训练过程中的损失以及经过稀疏化训练后的均值平均精度如图 9 所示。

由图 9 可以看出，针对 YOLOv4-3SPP-Ghost 模型以及数据集，当 $\alpha=0.000\ 1$ 时，训练 100 次，平均检测精度以及训练损失分别达到最佳和最低。因此，本文稀疏化训练选取 $\alpha=0.000\ 1$。

当 $\alpha=0.000\ 1$ 时，100 次稀疏化训练的过程如图 10 所示，可以看出，不同通道的尺度因子均向 0 靠近，可有效地进行通道裁剪。



(a)训练损失



(b)均值平均精度

图 9　不同惩罚项的训练损失与均值平均精度变化曲线

Fig. 9　Variation curves of training loss and mAP with different penalty terms

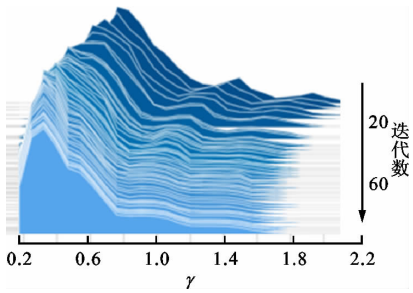


图 10　稀疏化过程中尺度因子分布情况

Fig. 10　Distribution of scaling factor in sparse process

3.5　模型裁剪对比

为了验证模型裁剪的有效性，本文对模型 YOLOv4-

3SPP-Ghost 进行裁剪，裁剪率分别为 0.5、0.8 以及 0.9，用 YOLOv4-3SPP-Ghost-0.5、YOLOv4-3SPP-Ghost-0.8、YOLOv4-3SPP-Ghost-0.9 表示。不同裁剪率模型的性能对比如表 2 所示。

表 2 不同裁剪率模型性能对比

Table 2 Performance comparison of different pruned models

模型	浮点型运算量	模型体积/MB	预测精度/%	预测时间/ms	
				GTX 1060	TX2
YOLOv4	128.46	245.8	90.06	95.15	555.56
YOLOv4-3SPP-Ghost-0.5	30.02	41.8	**89.93**	39.62	279.33
YOLOv4-3SPP-Ghost-0.8	10.66	9.6	87.63	32.59	204.08
YOLOv4-3SPP-Ghost-0.9	**5.76**	**3.5**	45.13	**25.91**	**178.57**

注：表中加粗数据为最优值。

由表 2 可以看出，随着裁剪比例的提升，模型体积、浮点型运算量及预测时间均在降低，在平台 GTX 1060 均达到实时检测。YOLOv4-3SPP-Ghost-0.8 模型的浮点型运算量压缩至 YOLOv4 模型的 8.2%，体积压缩至 4%，检测速度在 GTX 1060 和 TX2 平台上分别提高 2.9 倍和 2.7 倍。实验结果证明，模型通道裁剪在保证精确度的基础上，可有效地压缩体积和浮点型运算量，提升算法的检测速度，压缩后的模型体积与预测速度均优于传统 YOLOv4 模型。YOLOv4-3SPP-Ghost-0.8 模型部分识别结果如图 11 所示。

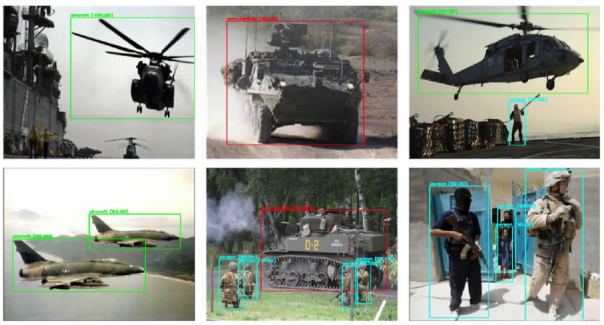


图 11 YOLOv4-3SPP-Ghost-0.8 模型部分识别结果

Fig. 11 Some object recognition results of YOLOv4-3SPP-Ghost-0.8

4 结 论

针对现有的轻量化模型无法兼顾检测速度和检测精度的问题，本文提出了一种基于 Ghost 模块的 YOLO 目标识别算法 GS-YOLO。该算法通过增加 SPP 模块优化网络模型，重新设计了 SPP 模块的引入位置，在减少模型浮点型计算量的同时，提升基础网络的检测精度与检测速度。利用高效轻量化模块替换结合通道剪枝极限压缩的方法，在保证精度的同时，有效压缩了模型体积和运算量，减少了模型占有的内存资源，提升了模型的运行效率。其中：采用轻量化 Ghost 卷积模块重构目标识别模型，减少卷积操作，提升算法的检测速率；采用通道剪枝极限压缩方法裁剪模型并对模型进行微调，在保证精确度的基础上，进一步减少参数量、运算量并压缩模型体积，提升检测速度。但是，本文只考虑了降低参数量，并未考虑不同硬件平台的特点，后期可以针对不同的平台设计不同的模型，对扩展模型的泛化能力进行进一步研究。

实验结果证明，本文提出的方法有效地对 YOLOv4 模型进行了压缩，相较于传统的 YOLOv4 模型有更小的体积和更快的检测速度，对算力低、内存少的嵌入式平台友好。此外，本文所提方案不仅可以对 YOLOv4 模型进行高效的压缩与加速，基于卷积层和 BN 层所设计的卷积神经网络模型都可参照本文方案进行卷积操作的替换以及通道裁剪，为深度学习算法部署在资源有限的嵌入式平台上提供理论支撑。

参考文献：

[1] 乔梦雨，王鹏，吴娇，等. 面向陆战场目标识别的轻量级卷积神经网络 [J]. 计算机科学，2020，47(5)：161-165.
QIAO Mengyu，WANG Peng，WU Jiao，et al. Lightweight convolutional neural networks for land battle target recognition [J]. Computer Science，2020，47(5)：161-165.

[2] 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 Yongqiang，RAO Yuan，DONG Shipeng，et al. Survey on deep learning object detection [J]. Journal of Image and Graphics，2020，25(4)：629-654.

[3] WANG Peng，SUN Mengyu，WANG Haiyan，et al. Convolution operators for visual tracking based on spa-

tial-temporal regularization [J]. Neural Computing and Applications, 2020, 32(10): 5339-5351.

[4] SHARKEY N. The ethical frontiers of robotics [J]. Science, 2008, 322(5909): 1800-1801.

[5] 刘俊，孟伟秀，余杰，等. 面向军事目标识别的 DRFCN 深度网络设计及实现 [J]. 光电工程，2019，46(4)：180307.
LIU Jun, MENG Weixiu, YU Jie, et al. Design and implementation of DRFCN in-depth network for military target identification [J]. Opto-Electronic Engineering, 2019, 46(4): 180307.

[6] WANG Peng, WU Jiao, WANG Haiyan, et al. Low-light-level image enhancement algorithm based on integrated networks [J]. Multimedia Systems, 2020(10): 1-11.

[7] LI Xiaoyan, LV Zhigang, WANG Peng, et al. Combination weighted clustering algorithms in cognitive radio networks [J]. Concurrency and Computation: Practice and Experience, 2020, 32(23): e5516.

[8] LEMLEY J, BAZRAFKAN S, CORCORAN P. Deep learning for consumer devices and services: pushing the limits for machine learning, artificial intelligence, and computer vision [J]. IEEE Consumer Electronics Magazine, 2017, 6(2): 48-56.

[9] HOWARD A G, ZHU M, CHEN B, et al. MobileNets: efficient convolutional neural networks for mobile vision applications [EB/OL]. (2017-04-17)[2021-03-04]. https://arxiv.org/abs/1704.04861.

[10] LEE Y, PARK J. CenterMask: real-time anchor-free instance segmentation [C]// Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ, USA: IEEE, 2020: 13903-13912.

[11] LIN T Y, MAIRE M, BELONGIE S, et al. Microsoft COCO: common objects in context [C]// Proceedings of the 2014 European Conference on Computer Vision (ECCV). Cham, Germany: Springer, 2014: 740-755.

[12] TAN Mingxing, PANG Ruoming, LE Q V. EfficientDet: scalable and efficient object detection [C]// Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ, USA: IEEE, 2020: 10781-10790.

[13] HAN Kai, WANG Yunhe, TIAN Qi, et al. GhostNet: more features from cheap operations [C]// Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ, USA: IEEE, 2020: 1577-1586.

[14] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. YOLOv4: optimal speed and accuracy of object detection [EB/OL]. [2021-03-04]. https://arxiv.org/abs/2004.10934.

[15] WANG C Y, LIAO H Y M, WU Y H, et al. CSPNet: a new backbone that can enhance learning capability of CNN [C]// Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). Piscataway, NJ, USA: IEEE, 2020: 390-391.

[16] LIN T Y, DOLLÁR P, GIRSHICK R, et al. Feature pyramid networks for object detection [C]// Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ, USA: IEEE, 2017: 936-944.

[17] LIU Shu, QI Lu, QIN Haifang, et al. Path aggregation network for instance segmentation [C]// Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Piscataway, NJ, USA: IEEE, 2018: 8759-8768.

[18] HE Kaiming, ZHANG Xiangyu, REN Shaoqing, 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.

[19] IOFFE S, SZEGEDY C. Batch normalization: accelerating deep network training by reducing internal covariate shift [C]// Proceedings of the 32nd International Conference on Machine Learning (ICML). Princeton, NJ, USA: IMLS, 2015: 448-456.

[20] LIU Zhuang, LI Jianguo, SHEN Zhiqiang, et al. Learning efficient convolutional networks through network slimming [C]// Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV). Piscataway, NJ, USA: IEEE, 2017: 2736-2744.

[21] ZHANG Pengyi, ZHONG Yunxin, LI Xiaoqiong. SlimYOLOv3: narrower, faster and better for real-time UAV applications [C]// Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). Piscataway, NJ, USA: IEEE, 2019: 37-45.

（编辑　陶晴）