DOI:10.19651/j.cnki.emt.2211459

基于 ARM 平台目标检测的轻量化方法

张 雷 童虎庆 谢锦昌 杨 昆
(沈阳航空航天大学电子信息工程学院 沈阳 110136)

摘 要: 为了解决基于深度学习的目标检测算法庞大的计算量和内存占用,导致在 ARM 平台的边端设备上部署难度大的问题。本文提出一种基于 ARM 平台目标检测的轻量化方法,首次将网络中的批标准化层缩放因子和卷积层卷积核参数同时添加约束,稀疏训练后将其作为通道重要性判断的两个准则,将不重要的通道双准则剪枝;针对剪枝效果较差的层结合 CBAM 注意力设计轻量化结构替换;再对结构替换后的模型重新训练得到最终模型。在单目标检测和多目标检测场景,分别对改进的 YOLOv5n 和 YOLOv5s 实验,结果表明该方法在 ARM 设备上均优于常规轻量化方法。在人物检测场景中,对 YOLOv5n 优化后的模型大小仅有 0.68 MB,在 ARM 设备上单核 CPU 部署时检测速度达到 45 fps,完全满足实时性要求,大幅度降低边端设备部署难度和硬件成本。

关键词: ARM 平台;目标检测;YOLOv5;稀疏训练;轻量化结构;算法部署
中图分类号: TP391.4 **文献标识码:** A **国家标准学科分类代码:** 510.8060

Lightweight method of object detection based on ARM platform

Zhang Lei Tong Huqing Xie Jinchang Yang Kun
(School of Electronic Information Engineering, Shenyang Aerospace University, Shenyang 110136, China)

Abstract: To tackle the difficulty in deploying on the side devices of the ARM platform triggered by the huge computation amount and memory occupation of deep learning-based object detection algorithms, this paper presents a lightweight method based on ARM platform object detection. Innovatively, this research adds constraints to the scaling factor of the batched normalized layer and the convolution kernel parameter of the convolution layer in the network, performs sparse training, and uses the scaling factor and the convolution kernel parameter as two criteria for judging the importance of channels and thus pruning the unimportant channels. Furthermore, CBAM attention is adopted to achieve lightweight structure replacement of layers with poor pruning effect. On this basis, the model processed with structure replacement is re-trained to eventually build the final model. Lastly, the optimized YOLOv5n and YOLOv5s are tested respectively in single-object detection and multi-object detection scenarios. The test results show that the method proposed in this research is superior to the conventional lightweight method on ARM devices. In the character detection scenario, the size of the optimized YOLOv5n model is just 0.68 MB, and the detection speed can reach 45 fps when the single-core CPU is deployed on ARM devices, which can well meet the real-time requirements and also greatly reduce the difficulty and hardware cost of the deployment on side devices.

Keywords: ARM platform;target detection;YOLOv5;sparse training;lightweight structure;algorithm deployment

0 引 言

近年来,随着计算机性能迅速发展,构建更深、更宽的深度卷积神经网络(convolutional neural networks, CNNs)逐渐成为可能,而更复杂的 CNNs 往往意味着可以有更好的信息提取能力[1]。目前基于 CNNs 的目标检测算法已经在自动驾驶[2]、智慧安防等场景中大规模应用。但是基于 CNNs 的目标算法在部署时需要大量的计算和内存资源,在硬件资源受限、实时性要求较高的场景中,目前的目标检测算法仍然不符合要求。基于 ARM 的通用处理器由于其低功耗、高性价比、易用性强等特点广泛应用在现实场景中,但受到计算资源和内存成本的影响,ARM 平台的目标检测算法部署效果不佳。因此如何将优秀的目标检测算法部署在基于 ARM 平台的边端设备中,是目前亟需解决的问题。

目前,基于卷积神经网络的目标检测算法主要分为两类[3]:一是两阶段目标检测算法,例如 R-CNN[4]、Fast

收稿日期:2022-09-20

R-CNN[5] 等;二是一阶段目标检测算法,例如 SSD[6]、YOLO[7] 等。前者精度高但计算量大,不适合部署在边端设备中,一阶段算法中,YOLOv5 由于具备检测精度高、计算资源要求相对较低等优点,是目前应用最广泛的目标检测算法,但仍然需要高性能 GPU 环境以满足实时性要求。在基于 YOLOv5 的轻量化研究中,文献[8-9]中作者将原算法中部分网络层使用轻量化结构替换并添加注意力机制,减少计算量的同时提高识别精度,将其应用在果树产量和水下目标测定场景;文献[10-11]中作者添加注意力机制强化模型抗干扰能力和使用网络剪枝技术减小参数量,将其应用在 PCB 检测和交通标志检测场景;文献[12]中作者将原算法中的主干网络替换成 MobileNetV3[13] 轻量型网络并添加注意力机制以减小参数量,将其应用在安全帽检测场景中。

以上研究对模型的轻量化主要使用两种方法:轻量化主干网络替换和网络结构剪枝。上述方法存在以下不足:直接将主干网络全部替换的操作降低了模型的泛化性能,破坏了原算法的通用性,并不适合直接应用在边端检测场景中;结构剪枝可以有效减少模型的计算量,但是只根据单一准则判断输出层的重要性,没有考虑影响卷积输出的其他因素,导致大幅度剪枝时精度骤降;两种方法融合时均采用先确定网络结构后剪枝的方案,此时剪枝前的网络结构成为模型精度的瓶颈,剪枝的力度大小决定了模型的最终性能,此方案仍有较大局限性。为了使目标检测算法应用在更多的边端场景中,本文结合 YOLOv5 目前轻量化研究的优缺点,提出一种实时目标检测算法的优化方法,本文贡献如下:

1)将剪枝作为模型参数搜索的一种方法,先双准则剪枝确定模型的网络层参数,再进行结构替换,一定程度上解决了极限剪枝精度骤降的问题,可后期调整结构增加模型的泛化性能。

2)针对 YOLOv5 模型结构,设计轻量化卷积层替换结构,并添加 CBAM 轻量注意力机制,在保证精度的同时,降低卷积层的参数量和计算时间。

3)将 BN 层缩放因子和卷积核权重分别添加约束,进行双准则剪枝。将双准则剪枝后的模型参数作为基础,将部分卷积层替换为轻量化卷积结构,重新训练产生新的网络模型,并将模型部署到 ARM 平台。

4)分别在人物检测数据集和 VOC 数据集上测试,验证本文方法的泛化性。

实验表明,经过本文方法优化后的 YOLOv5 模型,在人物检测场景中使用 ARM 平台单核 CPU 部署时,检测速度最高可达 45 fps,多目标检测场景中检测速度为 34 fps,完全满足实时性要求。

1　模型轻量化算法

1.1　目标检测算法介绍

YOLOv5 算法由 Ultralytics LLC 公司提出,截止到本研究开始时已经发展到第 6 个版本(v6.0)。v6.0 版本在此前基础上减小 YOLOv5s 的特征图宽度,提出了 YOLOv5n。在网络结构上,此前版本的 Focus 模块对输入图片进行切片操作,减少了模型参数量和计算量,但并行操作的增加和内存访问成本的增高,导致其不适合部署在 ARM 平台中。因此在 v6.0 版本中将 Focus 模块换成了卷积核大小为 6×6,步长为 2 的卷积层,该卷积层的输出和 Focus 模块输出大小一致,在没有丢失信息的前提下,减小了内存访问代价。较小的特征图宽度和网络的优化使 YOLOv5n 更适合基于 ARM 的环境下的部署。

1.2　双准则通道剪枝流程

通道剪枝可以去掉相对不重要的通道,为了尽可能压缩模型中的参数量和计算量,本文将批标准化层(batch normalization,BN)的缩放因子 γ 和卷积层卷积核权重 w 同时作为判断卷积核重要性准则。双准则通道剪枝流程如图 1 所示,首先对原模型中的 γ 和 w 稀疏化训练,然后依据双准则裁剪非重要通道,并对裁剪后的模型微调以恢复精度。

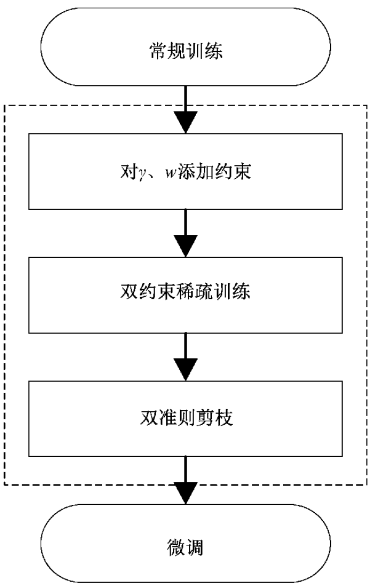


图 1　双准则通道剪枝流程

1.3　卷积核权重和 BN 层缩放因子稀疏化

在基于卷积神经网络设计的算法中,卷积层占用了大量的计算成本和内存空间。YOLOv5 网络中每个卷积层后均有 BN 层,BN 层的输入为卷积层的输出,BN 层的输出作为下一网络层的输入。由于最终的输出同时受到卷积层和 BN 层的影响,现有的单准则剪枝无法同时兼顾。

卷积层的计算过程为卷积核对输入矩阵滑动计算内积,得到输出矩阵,矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{W}$ 内积计算公式如式(1)所示。卷积核参数决定了卷积层的输出,只对卷积核权重稀疏化时,BN 层将影响最终的输出。若最终输出的值足够大,单准则剪枝将会导致其误裁剪。

$$\boldsymbol{A}\cdot\boldsymbol{W}=\sum_i\sum_j\boldsymbol{A}_{ij}\boldsymbol{W}_{ij} \tag{1}$$

BN 层是为了解决在训练过程中，中间层数据分布发生变化的情况，能够提高训练效率、防止梯度爆炸等[14]。BN 层的计算公式和流程如下：

$$\mu=\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{z}_{in} \tag{2}$$

$$\delta=\frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{z}_{in}-\mu)^2 \tag{3}$$

$$\hat{\boldsymbol{z}}=\frac{\boldsymbol{z}_{in}-\mu}{\sqrt{\delta+\varepsilon}} \tag{4}$$

$$\boldsymbol{z}_{out}=\gamma\hat{\boldsymbol{z}}+\beta \tag{5}$$

首先通过式(2)～(3)得到 BN 层样本的均值和方差，其中 $\boldsymbol{z}_{in}$ 为上一个卷积层的输出、m 表示最小批大小(mini-batch size)；再通过式(4)对 BN 层样本进行正则化操作，其中 ε 的作用是防止分母为 0；式(5)对正则化后的 $\hat{\boldsymbol{z}}$ 进行平移和缩放处理，其中 $\boldsymbol{\gamma}$ 和 $\boldsymbol{\beta}$ 分别是可学习参数，通过训练可以使网络学习到原始网络所需要的特征分布。BN 层的缩放因子 γ 可以作为评估通道重要性的准则之一[15]，单准则剪枝将会根据稀疏化后的 $\boldsymbol{\gamma}$ 判断通道的重要性，从而误裁剪输出较大的值。因此同时将卷积层权重 $\boldsymbol{w}$ 和 BN 层缩放因子 $\boldsymbol{\gamma}$ 稀疏化，融合两种剪枝方法可以尽可能确保 $\boldsymbol{z}_{out}$ 趋向于 0。

对于卷积核权重稀疏化而言：当卷积核 $\boldsymbol{W}$ 内参数趋近于 0 时，该卷积核对任意输入的输出 $\boldsymbol{A}\cdot\boldsymbol{W}$ 也趋向于0，L1 正则化可以作为数据无关卷积核选择的一个准则[16]。在执行卷积运算时对卷积核参数添加 L1 正则化项，训练时可以使卷积核参数稀疏化。如式(6)中，L_1 为原损失函数，添加 L1 正则化项，令 λ 为约束系数，w 为卷积核参数，那么在反向传播时对式(6)求导得到式(7)，训练时在原损失函数基础上加 $\lambda\times sign(\boldsymbol{w})$ 可以实现对卷积核参数 w 的稀疏化。

$$L=L_1+\lambda\sum|\boldsymbol{w}| \tag{6}$$

$$L'=L_1'+\lambda\sum sign(\boldsymbol{w}) \tag{7}$$

对于 BN 层缩放因子 γ 稀疏化而言：训练中通过同样可以对 $\boldsymbol{\gamma}$ 参数添加 L1 正则化项，达到约束 $\boldsymbol{\gamma}$ 参数的目的，最终训练时的损失函数如下：

$$L'=L_1'+\lambda\sum sign(\boldsymbol{\gamma}) \tag{8}$$

1.4　双准则通道剪枝

稀疏训练后将会得到稀疏化的卷积核权重 w 和缩放因子 $\boldsymbol{\gamma}$，对于双准则通道剪枝，通道的重要性评估步骤需要将两种准则融合。

卷积层稀疏化卷积核参数 $\boldsymbol{w}$ 是一个多维矩阵，此评判准则需要求出每一个 $\boldsymbol{w}$ 矩阵中所有元素绝对值的和(即 1 范数)，对 $\boldsymbol{w}$ 矩阵的 1 范数排列，设置一个合适的剪枝比例，得到一个对应通道选择的 1 范数区间。BN 层稀疏化后的缩放因子 $\boldsymbol{\gamma}$ 是一个一维向量，此评判准则只需要将所有的 BN 层 $\boldsymbol{\gamma}$ 排序，设置一个合适的剪枝比例，得到重要通道和不重要通道对应的 $\boldsymbol{\gamma}$ 参数区间。将两种通道重要性评判准则结合：只有当两个准则均认为某个通道重要时，该通道才可以保留，常规双准则剪枝示意图如图 2 所示。当某一层中所有通道都不符合时，分别按照两个准则对当前层通道重要性进行排序，在两个评估准则中各取一半较为重要的通道，合并为当前网络层。

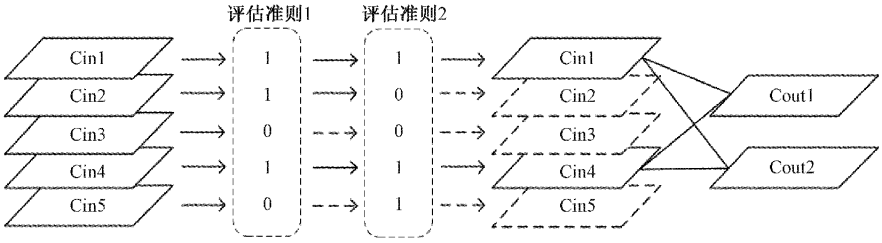


图 2　双通道剪枝示意图

1.5　轻量化结构替换

稀疏训练并不能将所有的卷积层卷积核权重 w 和 BN 层缩放因子 γ 都稀疏到理想的效果，导致剪枝到某一阈值时精度骤降的情况，与参数量和计算量的下降比例不成线性关系。针对此种情况，本文采用轻量化结构替换的方案，替换掉剪枝效果不佳的网络层，压缩网络的同时能够保证模型精度满足检测要求。

在 MobileNet[17]轻量级网络中，常规卷积被拆分为逐通道卷积和逐点卷积组合成深度可分离卷积。降低了网络计算量，但却丢失了大量的信息，精度降低明显。文献[18]中强调了内存访问成本和大量并行结构对 CPU 计算的影响，并提出了 4 个高效网络设计准则。本文将其作为参考并结合 YOLOv5 卷积层特点设计出一种融合 CBAM 注意力机制的轻量化卷积替换结构，结构如图 3 所示。

在 YOLOv5 网络中，卷积层的输入输出通道大部分为倍数关系，导致卷积运算时内存访问成本较高。使用图 3 轻量化网络结构主要优势有：将输入分为两个部分，尽可能将卷积层输入输出通道数保持一致，有效减少了内存访问成本；使用深度卷积和点卷积替换常规卷积，大幅度降低了原卷积运算的计算量；不使用分组卷积，使用 Concat 算子而不使用 Add 算子，将输出与原卷积保持一致的同时，减少了内存访问成本和计算量。

CBAM[19]注意力机制分别从通道和空间两个维度提取注意力，相较于单一维度注意力而言有更好的特征提取

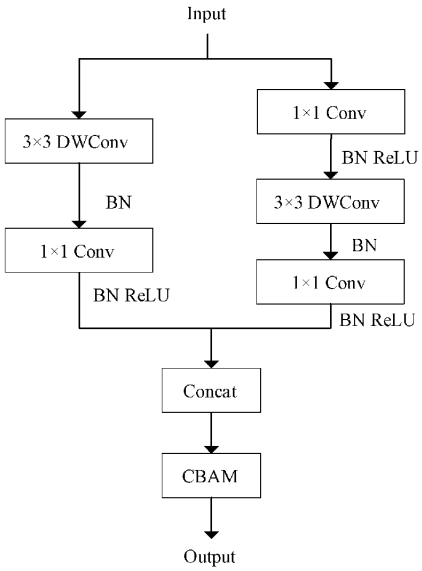


图 3　轻量化替换结构

性能。CBAM 模块结构如图 4 所示,其由两部分组成:通道注意力模块和空间注意力模块,通道注意力用于处理特征图各通道的分配关系,空间注意力可以更加关注特征图中起决定性作用的像素。该模块有通用性强、轻量化程度高等优点,虽然增加了少量计算量,但提高了网络特征提取能力。

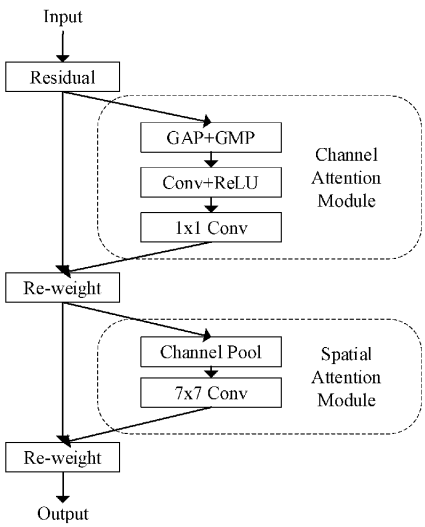


图 4　CBAM 注意力模块

2　实验及结果分析

2.1　实验环境及数据集介绍

本文实验训练平台操作系统为 Windows10,硬件配置为 Intel Core i9-10850K 处理器,GPU 为 NVIDIA Geforce RTX 3060,使用 Pytorch1.9.0 训练框架,CUDA 版本为 11.0。

实验中将算法分别应用在通用目标检测和人物安防检测两个场景中,分别验证了该算法在多目标检测和单目标检测场景下的实验效果。开源数据集 PASCAL VOC 中有 20 个类别,将其中 07+12 训练和测试数据切分,得到训练集 16 551 张图片和测试集 4 952 张图片,将此数据集作为通用目标检测场景中数据集。再将 PASCAL VOC 中人物目标分离,得到训练集 6 095 张和测试集 2 007 张图片,作为人物安防检测场景数据集。

2.2　实验指标介绍

目标检测领域中平均准确率(mean average precision,mAP)是算法精度的重要评测指标之一,本实验将 mAP 作为评判模型精度的指标,mAP 为所有类别的准确率均值,可以准确表达一个模型在当前应用场景的精度表现;浮点运算次数(floating point operations,FLOPs)表示模型的浮点运算总次数,常用来统计模型的计算量,参数量(Parameter)指的是模型总的参数量,本实验中使用 FLOPs 和 Parameter 衡量模型的复杂度。

2.3　实验记录及其分析

为了验证算法的可靠性,本文分别对 YOLOv5s 和 YOLOv5n 两种网络实验,对两种网络的优化方法完全一致。首先对原 YOLOv5 算法进行常规训练,对 YOLOv5n 和 YOLOv5s 两个网络在人物数据集 A 和通用数据集 B 训练,训练结果如表 1 所示。在人物检测数据集中,YOLOv5s 在计算量和参数量大幅度增加的情况下,mAP 仅提高了 0.8%,说明在 YOLOv5s 存在大量冗余信息;在通用数据集中,YOLOv5n 虽然计算量和参数量更少,但是相较于 YOLOv5s 精度下降明显,因此说明在此数据集中前者并不能学习到足够的信息。

表 1　常规训练结果

模型	计算量(GFLOPs)	参数量	mAP/%	模型大小/MB
YOLOv5s^{A}	10.21	7 022 326	89.6	13.6
YOLOv5n^{A}	2.70	1 765 270	88.8	3.62
YOLOv5s^{B}	10.31	7 073 569	83.5	13.7
YOLOv5n^{B}	2.76	1 790 977	77.1	3.67

基于表 1 的实验结果,在人物检测数据集上 YOLOv5n 更加精简,为了评判双准则剪枝的效果,本文对此数据集下的 YOLOv5n 进行了消融实验。实验中分别对不同的剪枝方法分别设置了 8 组剪枝阈值,对剪枝后的模型在不同计算量、参数量和模型大小下的 mAP 统计。图 5 为 3 种剪枝方法下 mAP 和模型大小之间的关系曲线,从数据上看,BN+Conv 双准则剪枝的效果在剪枝率较高时精度优势明显。当剪枝的模型大小为 1 069 KB 时,mAP 为 86.0%,模型大小降低 71% 的情况下,mAP 仅降低 2.8%,而使用 BN 和 CONV 单准则剪枝,当模型大小近似时,mAP 分别为 73.4%(1 072 KB)和 84.8%(1 072 KB)。并

且双准则剪枝方法在极限剪枝时仍然有明显优势，在模型大小只有 772 KB 时，模型的 mAP 仍然有 78.6%。

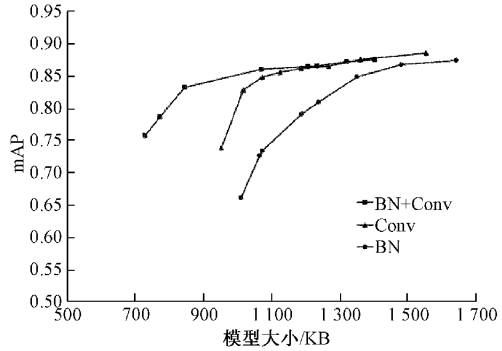


图 5　3 种剪枝方法 mAP 与模型大小关系曲线

模型 mAP 与计算量之间的关系曲线如图 6 所示，从计算量来看，在极限剪枝的情况下，相较于单准则通道剪枝，双准则剪枝在计算量下降更多的同时，模型精度更高。从消融实验整体来看，双准则剪枝算法相较于传统的单准则剪枝算法在各个维度均有更好的轻量化效果，在计算量和参数量相近时，双准则剪枝有更高的精度。

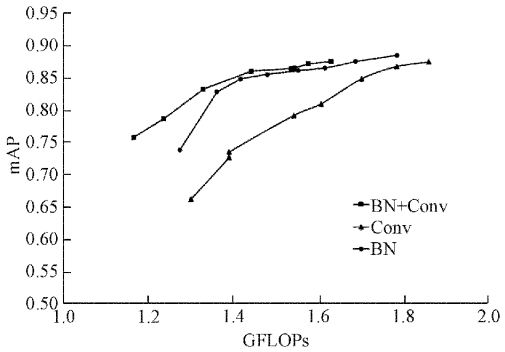


图 6　3 种剪枝方法 mAP 与计算量关系曲线

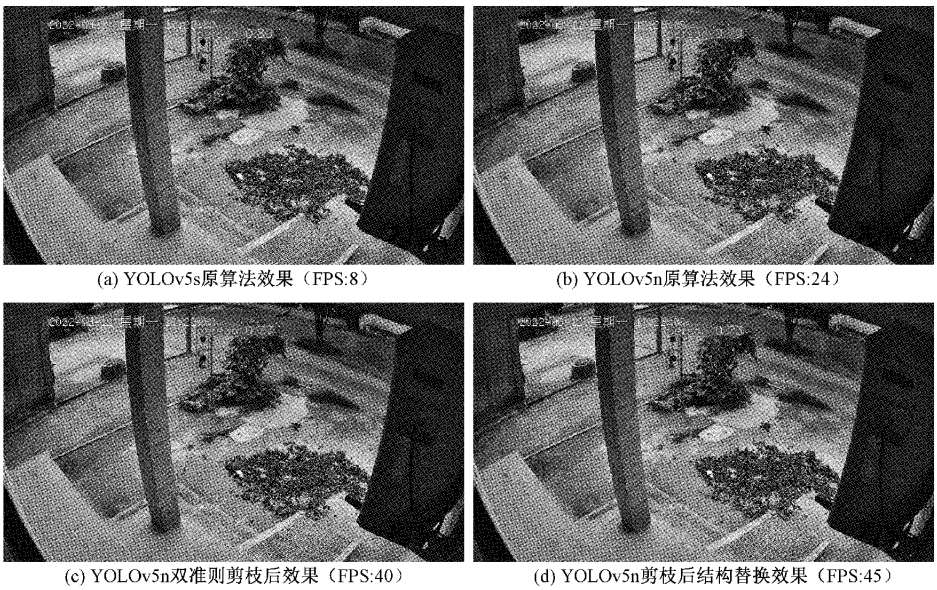
由消融实验中的曲线可以看出：在双准则剪枝后，当剪枝比例达到某一阈值，并继续加大剪枝力度时，会导致模型精度大幅度下降，但模型的计算量和参数量下降并不与精度成线性关系。实验中对比精度骤降前后的卷积层，将导致精度骤降的卷积层替换成轻量化结构，对精度骤降前的模型替换轻量化网络结构后，再次进行训练得到最终的模型。在对人物目标检测场景中，使用 YOLOv5n 进行上述操作，并将最终的模型在小米 11Pro 手机上部署，测试单核 CPU 实际推理速度，对模型前向推理 100 次取平均值得到最终的耗时，实验结果如表 2 所示，其中 YOLOv5n^0、YOLOv5n^1 和 YOLOv5n^2 分别代表原模型、剪枝后和剪枝后并轻量化结构替换的模型。

从表 2 数据来看，剪枝之后进行轻量化结构替换可以显著避免模型 mAP 精度骤降的情况，虽然精度有一定程度降低，但结合消融实验来看 mAP 和模型大小下降幅度明显优于直接大幅度剪枝的效果。最终模型大小降低了 81%，计算量减少了 63%，实际部署的推理时间大幅降低了 45%。

表 2　YOLOv5n 人物检测实验结果

模型	计算量 (GFLOPs)	mAP/%	模型大小/MB	耗时/ms
YOLOv5n^0	2.70	**88.8**	3.62	30.6
YOLOv5n^1	1.44	86.0	1.04	18.5
YOLOv5n^2	**0.99**	82.4	**0.68**	**16.7**

图 7 分别为人物安防检测场景下 ARM 平台实际的部署效果，4 个模型均正确检测出了摄像头中的任务目标。图 7(a)为 YOLOV5s 算法在此数据集下的实际表现，其精度较高，但是 FPS 只有 8，严重影响了实时性；图 7(b)为



(a) YOLOv5s原算法效果（FPS:8）　(b) YOLOv5n原算法效果（FPS:24）

(c) YOLOv5n双准则剪枝后效果（FPS:40）　(d) YOLOv5n剪枝后结构替换效果（FPS:45）

图 7　人物安防场景下各模型实际效果对比

YOLOv5n 的实际表现其在差不多的精度下 FPS 为 24,说明其确实更适合在 ARM 平台上部署;图 7(c)为使用本文方法双准则剪枝后精度虽略有降低,但是检测速度提升了 67%,保证了该算法的实时性;图 7(d)为剪枝并替换结构后的效果,相比原算法,检测精度有一定程度下降,但是模型大小只有 0.68 MB,FPS 达到了 45,提升了 87.5%;在摄像头视频流为 30 fps 的前提下,本文提出的算法均可以实现实时监测。

为了进一步验证本文提出的方法在更多数据集上的有效性,实验在 PASCAL VOC 数据上分别对 YOLOv5s 和 YOLOv5n 进行剪枝并进行结构性替换。此外将文献[20]中使用全 Ghost 卷积 YOLOv5 方法的 YOLOv5^{Ghost} 实现并加入对比,并参照文献[12]中使用 MobileNetV3 替换 YOLOv5s 主干的方法,设计了两种不同主干宽度的 YOLOv5^{V3L} 和 YOLOv5^{V3S} 应用在 PASCAL VOC 数据集中,最终的实验结果如表 3 所示。

表 3　YOLOv5 PASCAL VOC 实验结果

模型	计算量(GFLOPs)	mAP/%	模型大小/MB	耗时/ms
YOLOv5n^{0}	2.76	77.1	3.67	31.6
YOLOv5n^{1}	1.99	73.1	1.82	24.8
YOLOv5n^{2}	**1.42**	70.8	**1.30**	**21.9**
YOLOv5^{Ghost}	8.36	74.1	7.47	112.2
YOLOv5^{V3L}	10.32	75.8	10.36	107.2
YOLOv5^{V3S}	6.33	66.8	7.15	63.3
YOLOv5s^{0}	10.31	**83.5**	13.7	98.2
YOLOv5s^{1}	5.39	79.2	3.85	54.2
YOLOv5s^{2}	4.49	75.9	2.68	46.0

在 PASCAL VOC 数据集中,YOLOv5n 更小的网络宽度获得了快于 YOLOv5s 的速度。在 YOLOv5n 上使用本文方法优化模型后,模型降低了 65%的存储空间和 49%的计算量,并提升了 31%的推理速度。YOLOv5^{Ghost} 和 YOLOv5^{V3L} 虽然体积更小但是由于访存成本较高,ARM 平台中并不存在速度优势,精度也有下降。YOLOv5s 由于更宽的网络深度,获得了更高的精度,在使用本文优化方法后,裁剪了更多的冗余参数,最终在通用数据集上减少了 53%的推理时间,模型大小降低了 80%,并保持了较高精度。

3　结　　论

目前深度学习算法在低算力和低内存平台部署存在着较大难度,本文在首先对网络中 BN 层和卷积层权重添加约束,稀疏训练后进行双准则通道剪枝,然后对剪枝效果不好的网络层融合 CBAM 注意力设计轻量化结构替换。该方法在简单任务和复杂任务中相比原算法在内存占用和推理时间上都有明显优化效果,并优于现有轻量化算法。在人物检测场景中模型大小仅有 0.68 MB,FPS 提升了 87.5%,而精度并没有骤降,降低了部署平台的内存和计算资源的要求。

本文提出的算法不与知识蒸馏和量化算法冲突,在本文算法基础上可以进一步优化提升模型推理速度和精度。同时,剪枝后的模型对原网络进行了通道选择,可以用作为优化原算法的设计思路,为原网络的精简做出贡献。未来本文将结合上述优化方法,对目标检测算法进一步降低硬件部署成本,将深度学习算法应用在更多场景。

参考文献

[1] 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, 2016: 770-778.

[2] 王文胜,李继旺,吴波,等. 基于 YOLOv5 交通标志识别的智能车设计[J]. 国外电子测量技术, 2021, 40(10):158-164.

[3] 张慧, 王坤峰, 王飞跃. 深度学习在目标视觉检测中的应用进展与展望[J]. 自动化学报, 2017, 43(8): 1289-1305.

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

[5] GIRSHICK R. Fast R-CNN[C]. Proceedings of the IEEE International Conference on Computer Vision, 2015: 1440-1448.

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

[7] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. YOLOv4: Optimal speed and accuracy of object detection[J]. ArXiv Preprint,2020, ArXiv:2004.10934.

[8] 李志军,杨圣慧,史德帅,等. 基于轻量化改进 YOLOv5 的苹果树产量测定方法[J]. 智慧农业(中英文),2021,3(2):100-114.

[9] 林森,刘美怡,陶志勇. 采用注意力机制与改进 YOLOv5 的水下珍品检测[J]. 农业工程学报,2021, 37(18):307-314.

[10] 王恒涛,张上,张朝阳,等. 基于 YOLOv5 的轻量化 PCB 缺陷检测[J]. 无线电工程,2022,52(11):2094-2100.

[11] 张上,王恒涛,冉秀康. 基于 YOLOv5 的轻量化交通标志检测方法[J]. 电子测量技术,2022,45(8):129-135.

[12] 杨永波,李栋. 改进 YOLOv5 的轻量级安全帽佩戴检测算法[J]. 计算机工程与应用,2022,58(9):201-207.

[13] HOWARD A, SANDLER M, CHU G, et al.

Searching for MobilenetV3[C]. Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019: 1314-1324.

[14] IOFFE S, SZEGEDY C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]. JMLR, 2015.

[15] LIU Z, LI J, SHEN Z, et al. Learning efficient convolutional networks through network slimming[C]. Proceedings of the IEEE International Conference on Computer Vision, 2017: 2736-2744.

[16] LI H, KADAV A, DURDANOVIC I, et al. Pruning filters for efficient convnets[J]. ArXiv Preprint, 2016, ArXiv: 1608.08710.

[17] HOWARD A G, ZHU M, CHEN B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. ArXiv Preprint, 2017, ArXiv: 1704.04861.

[18] MA N, ZHANG X, ZHENG H T, et al. Shufflenet v2: Practical guidelines for efficient cnn architecture design[C]. Proceedings of the European Conference on Computer Vision(ECCV), 2018: 116-131.

[19] WOO S, PARK J, LEE J Y, et al. Cbam: Convolutional block attention module[C]. Proceedings of the European Conference on Computer Vision (ECCV), 2018: 3-19.

[20] DONG X, YAN S, DUAN C. A lightweight vehicles detection network model based on YOLOv5[J]. Engineering Applications of Artificial Intelligence, 2022, 113: 104914.

作者简介

张雷，博士，副教授，主要研究方向为图像处理、图像压缩。

E-mail: rd_zhangl@126.com

童虎庆(通信作者)，硕士研究生，主要研究方向为图像处理与传输技术。

E-mail: 1769620861@qq.com