

文章编号:1007-2780(2021)07-1006-12

基于 MPSOC 的航空图像目标检测系统设计

任 彬^{1,2}, 王宇庆^{1*}, 丛 振³, 聂海涛¹, 杨 航¹

(1. 中国科学院 长春光学精密机械与物理研究所, 吉林 长春 130033;

2. 中国科学院大学, 北京 100049;

3. 中国人民解放军联勤保障部队第 946 医院 医学工程科, 吉林 长春 130033)

摘要:近年来,航空光学成像技术快速发展,机载图像处理系统对于目标检测精度和检测速度的要求越来越高,传统的目标检测算法已经无法满足要求。与此同时,基于深度学习的目标检测算法凭借更优的性能表现得到了学术界的广泛关注。但这类算法往往参数较多,时间复杂度高且移动端移植困难。针对上述问题,本文提出了一种基于 Yolo V3 算法的 MPSOC 平台实现方案。利用改进的 k 均值聚类算法获取新的初始锚框,之后通过改变特征图的大小提高算法对小目标的检测精度,通过基于敏感度的剪枝方法压缩算法大小,最后利用 VISDRONE 数据集在 MPSOC 平台进行了验证。实验结果表明:改善的 Yolo 算法的 MAP 提高了 1.3%,误检率也得到了极大降低。算法经过压缩后,检测速度提高了 1 倍,体积仅为原来的 37%,基本满足了对航空图像目标检测的设计要求,同时为深度学习算法在 MPSOC 中实现提供了可行的解决方案。

关键词:航空图像;目标检测;模型压缩;MPSOC

中图分类号:TP391.4 文献标识码:A doi:10.37188/CJLCD.2020-0310

Design of aerial image target detection system based on MPSOC

REN Bin^{1,2}, WANG Yu-qing^{1*}, CONG Zhen³, NIE Hai-tao¹, YANG Hang¹

(1. Changchun Institute of Optics, Final Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China;

2. University of Chinese Academy of Sciences, Beijing 100049, China;

3. Department Medical Engineering, The 964th Hospital, Joint Service Support Unit, Chinese People's Liberation Army, Changchun 130033, China)

Abstract: In recent years, the traditional aerial image target detection algorithms have been unable to meet the requirements of detection accuracy and speed, while the rapid development of target detection algorithms based on deep learning provides a new idea for target detection. However, this kind of algorithm is often accompanied by large scale and highly dependent on GPU devices, which makes the migration of the mobile end of the algorithm difficult. Aiming at the above problems, this paper proposes a MPSOC platform implementation scheme based on Yolo V3 algorithm. Firstly, the anchor frame of the original network is re-selected by means of k-means clustering, the detection ac-

收稿日期:2020-11-22;修订日期:2020-12-16.

基金项目:国家自然科学基金青年基金(No.61401425)

Supported by Youth Program of National Natural Science Foundation of China(No.61401425)

* 通信联系人, E-mail:wyg7903@163.com

curacy of the algorithm is increased by adjusting the convolutional layer, and then the model scale is compressed by sensity-based pruning operation. Finally, VISDRONE data set is used to verify the Xilinx ZYNQ series MPSOC platform. The experimental results show that MAP of the improved Yolo algorithm increases by 1.3%, and the false detection rate is also greatly reduced. After the model is compressed, the detection speed is doubled and the volume becomes 37% of the original. It basically meets the design requirements of aerial image target detection, and provides a feasible solution for the implementation of deep learning algorithm in MPSOC.

Key words: aviation image; target detection; model compression; MPSOC

1 引言

目标检测作为机器学习的一个重要研究领域,经过了传统的目标检测算法和基于深度学习的目标检测算法两个阶段^[1]。2014年之前,传统的目标检测算法一直占据着主流地位。2001年,Paul Viola在CVPR会议中提出了Viola-Jones算法,通过Haar-like特征和Adaboost算法有机结合实现了人脸检测。4年之后,Dalal提出了HOG+SVM的方法,并应用于行人检测。2008年,Pedro Felzenszwalb在其基础上提出了DPM算法,通过多组件(Component)的策略解决了多视角的问题,通过基于图结构(Pictorial Structure)的部件模型策略解决了目标的形变问题。但这些传统的目标检测算法在进行区域选择时往往需要遍历整个图像,存在时间复杂度高的问题,同时手工设计的特征鲁棒性也较差。2014年之后,基于深度学习的目标检测算法快速崛起,形成了基于候选区域的目标检测算法和基于回归的目标检测算法两个流派。前者以R-CNN为代表,这类算法精度较高,但实时性较差,不利于工程项目的实际应用^[2];后者以Yolo为代表,这类算法虽然在精度方面略有逊色,但检测速度却远快于基于候选区域的目标检测算法。

另一方面,随着无人机在军事侦察和森林火灾预警方面的应用日益普遍^[3],针对无人机平台的航空图像目标检测技术引起越来越多国家的注意。国外很早便开展了针对无人机平台的检测跟踪系统的研究^[4]。1997年,Olson等人提出了基于模型的准实时性跟踪系统,该系统在目标发生一定形变或者被短暂遮挡时依旧能保持较好的准确度^[5]。2006年,美国先进技术研究局发布了COCOA监控系统,可以实现对行人、车辆等多种类型目标的自动检测和轨迹跟踪^[6]。2010年,

Ibrahim提出的MODAT系统弥补了COCOA系统在复杂背景下准确度不高的问题^[7]。国内相关方面的研究虽然起步较晚,但发展迅速。2011年,谭熊等人提出了一种计算量小、满足实时性的目标检测与跟踪算法^[8],2018年,汤轶等人提出了一种基于Kalman滤波和粒子群优化(PSO)算法的目标跟踪检测算法,将PSO算法代替穷举法,极大地降低了计算量^[9]。2020年,李航等人将深度可分离卷积应用于Yolo算法,通过减少参数量加快了网络传播速度,使算法具备嵌入式平台移植基础^[10]。

在实际无人机目标检测场景中检测速度往往是比检测精度更重要的评价指标,所以本文选择了速度较快的一阶段网络Yolo V3作为基础网络。针对航空图像目标检测的应用场景,本文设计了合适的锚框(Anchor),通过对网络的结构进行调整,使其平均精度均值(MAP)提升了1.3%。之后对各卷积层进行基于L1范数的敏感度分析,通过剪枝操作,使模型具备了多处理器片上系统(MPSOC)平台的移植基础。

本文所采用的MPSOC硬件平台为Xilinx推出的第二代多处理器片上系统(SOC)器件Zynq UltraScale+ MPSoC。作为异构SOC平台,该芯片中集成了多核应用处理器(四核ARM Cortex-A53应用处理器)、多核图形处理器(双核ARM Mali-400图形处理器)、多核实时处理器(双核ARM Cortex-R5实时处理器)、平台管理单元(电源管理、错误管理、配置管理以及安全管理)和可编程逻辑资源(高性能计算和丰富I/O扩展等)。与传统的SOC平台相比该平台拥有更多的资源和更高的安全性。

2 YOLO V3 检测原理

Yolo V3主干网络采用DarkNet53,只保留

了前 52 个卷积层,用于图片特征的选择与提取。网络的具体结构如图 1 所示。整个网络由 5 个 Yolo 模块组成,每个 Yolo 模块所含有的残差模块数目互不相同。以第二个 Yolo 模块为例,整个 Yolo 模块由 2 个残差模块组成,而每个残差模块又由 2 个卷积层组成,第一个卷积层所用的卷积核大小为 3×3 ,第二个卷积层所用的卷积核的大小为 1×1 。同时 Yolo V3 算法借鉴了 FPN(Feature pyramid networks)的思想,利用不同的特征层检测不同尺度的目标^[11]。整个算法会产生 3 个不同尺度特征图,分别对应图 1 中的 scale1、scale2 和 scale3。对于输入 608×608 的图像会得到 19×19 、 38×38 和 76×76 的 3 个特征图。

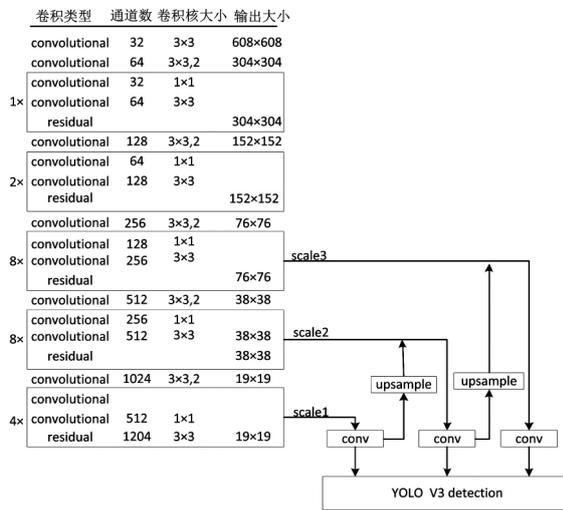


图 1 Yolo V3 网络结构图

Fig.1 Yolo V3 network structure diagram

特征图中每个单像素点(特征点)都对应着原图中的一个区域。如 76×76 的特征图,它由原图经 8 倍降采样后得到,每个点对应的原图区域大小为 8×8 。在网络训练的过程中,真实框(Ground truth)的中心点落在特征图中哪个特征点对应的区域,哪个特征点就负责这个真实框的回归。每个特征点对应 3 种不同的锚框,所以一个特征点

可同时回归 3 个不同的真实框。负责该真实框的特征点会选择与其拥有最大交并比的锚框对其位置进行回归。在原始 Yolo V3 网络中检测结果的通道数为 255。每个 255 维的通道由边框回归和分类情况两部分组成。对于每个点的每个预测框(bounding box)的回归情况需要用 $(x, y, w, h, confidence)$ 5 个基本参数描述,其中 x, y 为中心点坐标, w, h 为预测框的长宽, $confidence$ 为置信度。对于分类情况部分,类别数目等于分类部分通道的维数。如 COCO 数据集,需要检测的类别数为 80,那么就需要一个 80 维的通道描述其类别情况。所以最终确定的维数情况为: $3 \times (5 + 80) = 255$ 。其中 3 为每个单像素点拥有预测框的数量, 5 为边框回归情况, 80 为类别数。

3 模型改进

3.1 数据集的选择

一个好的训练数据集首先需要与应用场景一致。目前应用较多的 COCO、VOC 等数据集显然不具备目标尺寸偏小、多尺度以及成像背景复杂等航空图像的基本特点。目前可供选择的数据集主要有 DATO 和 VISDRONE。DATO 数据集的目标为高空无人机所拍摄,目标多为飞机、船舶、篮球场一类,且拥有的图片数量为 2 806 张。VISDRONE 数据集由天津大学标注,该数据集覆盖了中国从北到南 14 个城市的城市/郊区。整个数据集共有 12 类,分别是: people, pedestrian, bicycle, car, van, truck, tricycle, awning-tricycle, bus, motor, other, ignore regions, 图片数量为 8 629 张。往往数据集越大,所得到的模型泛化能力和精度越高。本文最终选择的数据集为 VISDRONE。VISDRONE 数据集主类为前 10 类,所以在进行模型训练与检测时,只采用了前 10 类。训练集、测试集、验证集中各个类别的分布情况如表 1 所示。

表 1 VISDRONE 数据分布情况

Tab.1 The distribution of the VISDRONE dataset

类别	训练集目标数量	验证集目标数量	测试集目标数量	合计
people	27 059	5 125	6 376	38 560
pedestrian	79 337	8 844	21 006	109 187

续 表

类别	训练集目标数量	验证集目标数量	测试集目标数量	合计
bicycle	10 480	1 287	1 302	13 069
car	144 867	14 064	28 074	187 005
van	24 956	1 975	5 771	32 702
truck	12 875	750	2 659	16 284
bicycle	4 812	1 045	530	6 387
awning-tricycle	3 246	532	599	4 377
bus	5 926	251	2 940	9 117
motor	29 641	4 886	5 845	40 372

3.2 锚框的选择

Yolo V3 原始模型在进行训练与验证任务时所选择的锚框都是根据 COCO 数据集而设定,并不适用于 VISDRONE 数据集中的目标,存在尺度偏大的问题,严重影响回归的准确度和效率,特别是对小目标的检测。此处采用改进的 k 均值聚类方法对 VISDRONE 数据集中的锚框重新聚类。由于锚框聚类的根本目的在于,使锚框与所有真实框最相似,即拥有最大的平均交并比(IOU)。所以在聚类时直接用交并比代替传统的欧氏距离作为距离度量指标。IOU 的具体定义如图 2 所示:两个矩形分别代表聚类中心边框和待分类边框,上面的黑色区域为两个边框的交集面积,下面黑色区域为两个边框的并集面积。

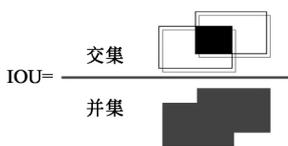


图 2 IOU 定义

Fig.2 Definition of IOU

虽然 IOU 可以较好地描述两个边框的距离,但我们通常把损失函数定义为:与模型的性能呈负相关,即损失函数值越小,模型性能更好。最终将 k 均值聚类的损失函数定义为:

$$\text{loss} = 1 - \text{IOU}_{\text{avg}}, \quad (1)$$

其中 loss 为定义的损失函数,1 为常数,IOU_{avg} 为所有类别交并比的平均,算法终止的条件为 loss

不再变化。由于 k 均值聚类需要事先给出聚类的类别数量 k,这里对 k 依次取值 1~19,观察 IOU_{avg} 的变化情况。

平均 IOU 随 k 的变化趋势如图 3 所示,可以发现随着 k 的增大,IOU_{avg} 也在增大且趋于平缓。普遍认为曲线由陡峭到平滑的拐点为 anchor box 的最优值^[12]。从图 3 可以看出 k 的最优值为 2,此时既可以加快损失函数的收敛,又可以消除候选框带来的误差。多次试验后发现得到的两个聚类中心框均为细长型,即高度远大于宽度。所以此处为了保持 Yolo V3 的原始结构的优势,只对最小的两个细长型 anchor 进行替换。替换后的 9 个 anchor box 值为:[10,17],[16,30],[33,23],[27,38],[62,45],[59,119],[116,90],[156,198],[373,326]。

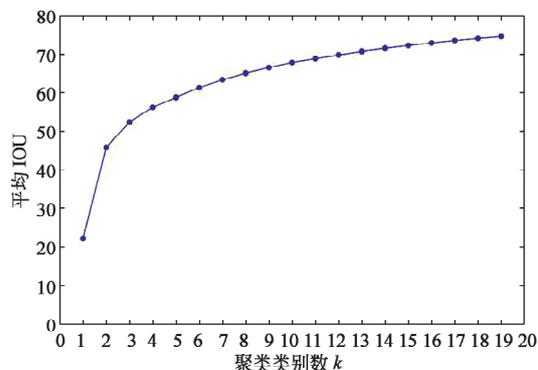
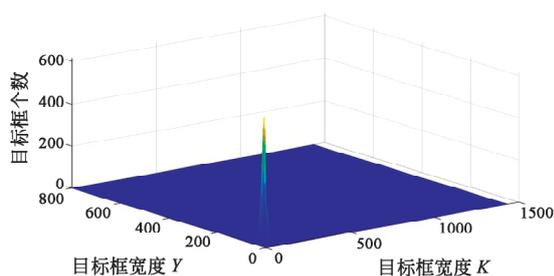


图 3 类别 k 与平均 IOU 关系曲线

Fig.3 Relationship curve between category variable k and variable average IOU

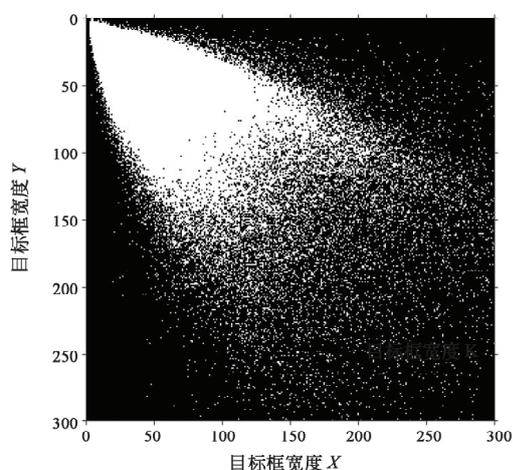
3.3 网络结构的调整

Yolo V3 网络利用 3 个不同尺度的特征图对不同大小的目标进行检测^[13]。当输入的图片为 608×608 时,最大的特征图也只有 76×76 。由于在该尺度特征图中每一个点都对应着原图一个 8×8 的区域。若待检测目标在原图中是一个 7×7 的区域,那么它将会变成特征图上的单像素点,目标检测几乎不可能。在更大的特征图中尚且如此,在更小的特征图中该区域将会变得更小,更加无法检测。



(a) 目标框分布三维图

(a) 3D distribution image of target box



(b) 目标框分布平面图

(b) Target box distribution plan

图 4 目标框的分布情况

Fig.4 Distribution of the ground truth

对 VISDRONE 数据集中真实框的分布进行统计,如图 4 所示:图 4(a) 图为目标框的三维分布情况,可以发现目标分布比较集中,且多为小目标。图 4(b) 为图 4(a) 的俯视图并二值化后的图像。定性分析并不利于问题的解决,所以将分布情况进行量化。将所有的真实框被分为 5 类,统

计目标框真实框的长和宽在不同取值范围的实际分布情况。具体如表 2 所示。

表 2 真实框分布情况

Tab.2 Ground truth distribution

Size	$x < 5,$ $y < 5$	$5 \leq x < 9,$ $5 \leq y < 9$	$9 \leq x < 17,$ $9 \leq y < 17$	$x \geq 17,$ $y \geq 17$
Num.	3 716	21 770	70 661	360 919

表 2 中第一行表示真实框的大小。 x 、 y 分别表示边框的长与宽。第二行表示在该长宽范围内边框的数量。可以发现 25 486 个真实框在原图上的长和宽均小于 8,约占目标总量的 5%,这也符合在实际航空图像中的目标分布特点。这些真实框在最大特征图中也仅仅是单像素点,特征无法得到很好的表达,最终造成目标无法检出。本文提出的解决办法为去除一个降采样过程,这样得到的最大特征图中,一个点将代表原图一个 4×4 的区域,这时只有 3 716 个实际目标框回归会比较难,占总目数比例不足 1%。极大地增加了模型对小目标的适应性。

去除一个降采样过程,有图 5 所示的 4 种方式:图 5(a) 所示方式是直接去除 darknet53 网络的最后一个 Yolo 模块,此时主干网络卷积层数目将变为 44,极大地减小了模型的参数量。但目前大量的研究表明,网络层数越多越有利于特征的表达和检测精度的提高^[14],所以并没有选择图 5(a) 方式。另一种方式为保留网络的整体结构不变,分别将第一个或第二个或第三个降采样过程中 3×3 卷积核的步长改变为 1。在越大的尺度上,小目标所占的像素越多,对其进行卷积特征提取工作越有意义。所以在小目标特征表达方面图 5(b) 性能优于图 5(c),图 5(c) 性能优于图 5(d)。对于图中的图 5(b),去除一个降采样过程后会在第一个卷积模块中集中大量卷积层,且这些操作是在 608×608 的图像中进行,会增加大量的计算。图 5(c) 与图 5(d) 结构的变化也会在对位置增加大量计算。在计算量方面图 5(b) > 图 5(c) > 图 5(d),为了最终在 MPSOC 平台中进行部署,需要在小目标特征表达与计算量之间权衡,本文最终选择了图 5(c) 结构。

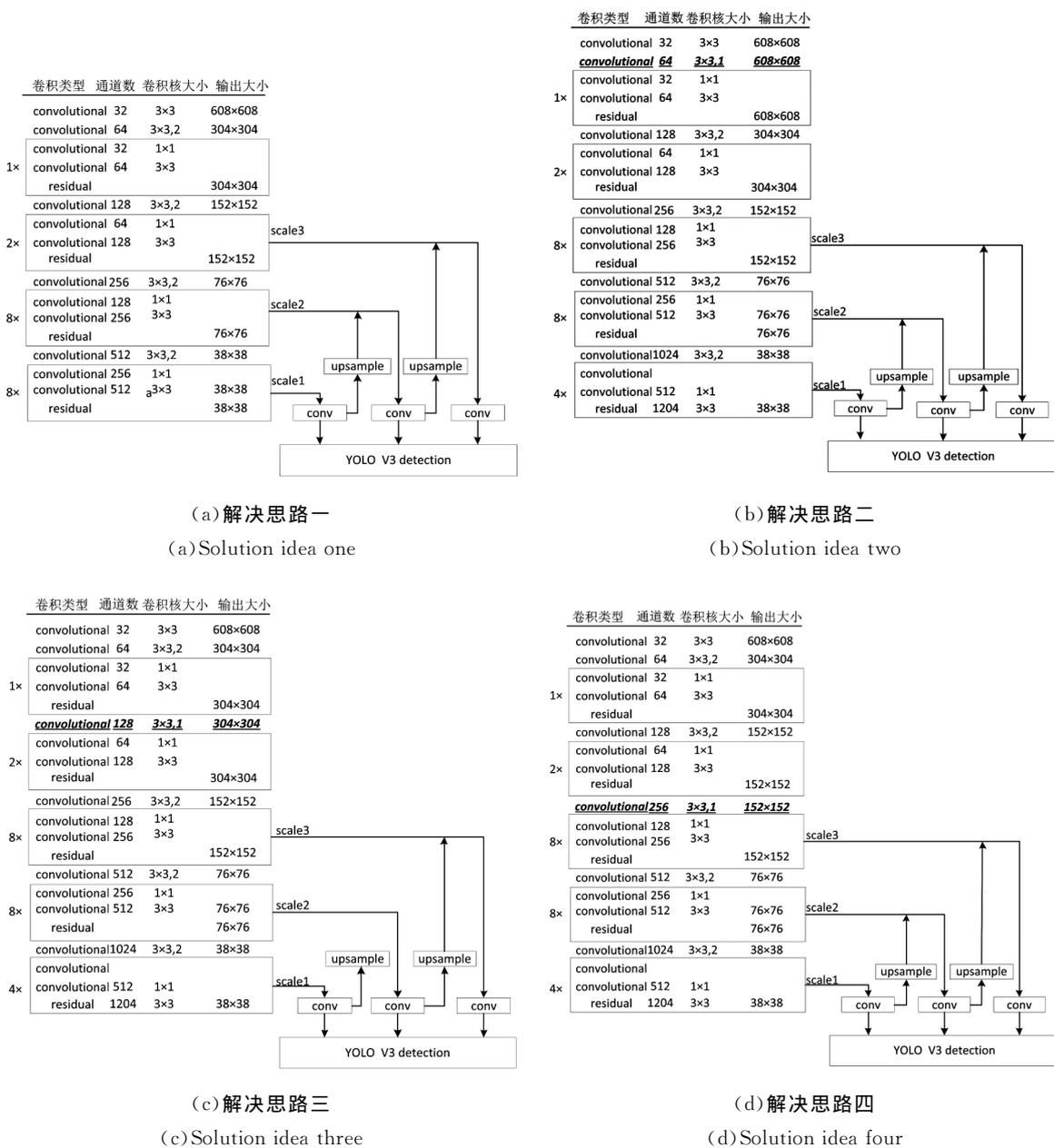


图 5 Yolo V3 结构变体

Fig.5 Changes in the Yolo V3 mode

3.4 模型压缩

目前主流进行模型压缩的方式有:量化、剪枝、蒸馏^[15]。本文采用基于 L1 范数的敏感度模型剪枝方法。这种方法以每一个卷积层为基本单元,依次对每一个卷积层裁剪一定比例卷积核,计算剪枝之后模型 MAP 的变化情况。MAP 下降越多代表卷积层越敏感。每一个卷积层拥有数量较多的卷积核,如何确定剪枝顺序也是一个难题。本文通过对每个卷积核计算其 L1

范数,按照 L1 范数从大到小排列,来确定裁剪顺序。L1 范数越小越先被裁减掉,这是由于 L1 范数较小的卷积核趋向于产生激活值较小的特征图^[16]。表 3 为对主干网络 darknet53 的敏感度分析结果。

通过大量实验,最终确定各个卷积层的裁减比例依次为 0.41, 0.35, 0.77, 0.10, 0.40, 0.67, 0.11, 0.91, 0.14, 0.14, 0.95, 0.16, 0.95, 0.15, 0.95, 0.07, 0.95, 0.07, 0.95, 0.10, 0.95, 0.13,

0.95, 0.08, 0.95, 0.06, 0.31, 0.95, 0.26, 0.95, 0.26, 0.95, 0.23, 0.95, 0.21, 0.95, 0.22, 0.95, 0.27, 0.95, 0.27, 0.95, 0.28, 0.76, 0.95, 0.49, 0.95, 0.59, 0.95, 0.49, 0.95, 0.55。为了恢复模

型的精度,需要对剪枝后的模型进行再训练,以恢复精度。本文没有采用传统的每剪枝一个卷积层就再训练一次的方法,而是一次性剪枝所有层,最后统一再训练,极大地降低了工作量。

表 3 各卷积层敏感分析结果

Tab.3 Sensitivity analysis results of each convolutional layer

卷积层名	裁剪率									
	0.05	0.15	0.25	0.35	0.45	0.55	0.65	0.75	0.85	0.95
Conv1,3×3	41.17	41.07	41.02	41.00	28.30	9.21	14.43	12.97	2.87	0.0
Conv2,3×3	40.06	38.43	37.63	37.11	32.30	28.85	19.33	10.43	4.24	0.0
Conv3,1×1	40.95	40.96	40.73	40.62	40.22	39.72	38.91	38.18	35.88	33.53
Conv4,3×3	38.21	35.79	33.87	33.33	20.87	15.55	3.73	4.68	0.91	0.0
Conv5,3×3	41.11	40.67	39.66	37.52	34.06	31.22	22.83	19.09	2.32	0.0
Conv6,1×1	40.86	41.05	40.79	40.18	39.83	39.22	38.41	35.64	33.67	32.97
Conv7,3×3	40.35	35.24	33.07	27.26	20.03	16.28	6.80	0.91	0.0	0.0
Conv8,1×1	41.06	40.87	40.53	40.17	39.81	39.44	39.58	38.94	38.31	35.49
Conv9,3×3	39.89	37.01	30.09	10.11	9.94	9.04	4.13	1.56	0.91	0.0
Conv10,3×3	40.5	36.78	34.12	28.22	23.46	16.00	4.27	0.91	0.0	0.0
Conv11,1×1	41.15	41.12	40.99	40.90	40.84	40.99	40.74	40.59	40.81	38.73
Conv12,3×3	40.35	37.51	33.44	30.19	21.28	14.47	6.68	1.82	0.0	0.0
Conv13,1×1	41.15	41.02	41.06	41.03	40.41	40.51	41.00	40.54	40.58	40.01
Conv14,3×3	40.01	40.15	37.17	33.98	27.79	13.02	6.65	0.0	0.0	0.0
Conv15,1×1	41.15	41.10	41.05	41.04	41.44	41.19	41.10	41.02	41.12	40.66
Conv16,3×3	38.82	31.88	22.87	15.70	6.07	2.21	0.0	0.91	0.0	0.0
Conv17,1×1	41.10	40.83	40.48	40.70	40.52	40.19	40.05	39.63	39.07	38.53
Conv18,3×3	37.89	33.28	26.92	14.90	7.83	1.36	0.23	0.0	0.0	0.0
Conv19,1×1	40.95	40.84	40.49	40.36	40.22	39.98	39.48	38.84	38.48	38.44
Conv20,3×3	40.47	34.46	26.48	18.80	4.32	0.91	0.0	0.0	0.0	0.0
Conv21,1×1	41.11	40.99	40.91	40.56	40.39	40.17	40.05	39.61	39.06	38.27
Conv22,3×3	40.35	36.21	27.19	18.38	9.66	7.09	0.95	0.0	0.0	0.0
Conv23,1×1	41.05	40.90	41.10	41.04	40.88	40.80	40.17	39.54	39.01	38.78
Conv24,3×3	39.57	31.82	17.40	12.36	1.93	0.30	0.0	0.0	0.0	0.0
Conv25,1×1	41.02	40.83	40.79	40.83	40.12	39.36	39.03	39.29	38.66	37.78
Conv26,3×3	38.42	32.41	24.10	13.05	4.54	2.65	0.52	0.45	0.0	0.0
Conv27,3×3	40.75	39.34	38.25	36.23	34.10	30.58	23.43	15.71	9.51	5.50
Conv28,1×1	41.13	41.05	40.92	40.93	40.84	40.86	40.64	40.99	40.17	39.84
Conv29,3×3	40.83	39.25	37.27	33.48	28.90	22.07	14.44	8.83	5.64	4.88

续 表

卷积层名	裁剪率									
	0.05	0.15	0.25	0.35	0.45	0.55	0.65	0.75	0.85	0.95
Conv30,1×1	41.13	41.07	40.96	40.90	40.88	40.53	40.68	40.62	40.36	40.01
Conv31,3×3	40.54	39.74	37.62	34.36	28.91	22.37	16.57	10.87	5.21	5.02
Conv32,1×1	41.12	41.06	41.11	40.90	40.98	40.61	40.18	40.23	39.15	38.52
Conv33,3×3	40.41	38.64	36.14	31.68	28.62	19.37	11.22	7.62	5.05	5.91
Conv34,1×1	41.10	40.90	40.60	40.77	40.73	40.60	40.04	39.54	39.43	38.94
Conv35,3×3	40.28	38.77	35.60	32.74	28.26	23.04	15.01	5.63	3.95	3.85
Conv36,1×1	41.13	41.09	40.86	41.03	40.99	40.94	40.84	40.81	40.93	41.00
Conv37,3×3	40.29	38.96	36.53	28.98	24.41	17.87	10.21	5.06	4.40	4.67
Conv38,1×1	41.05	41.00	40.99	40.81	40.69	40.62	40.40	40.04	39.70	38.66
Conv39,3×3	40.50	39.21	37.36	34.92	30.42	25.50	16.43	8.36	5.31	4.52
Conv40,1×1	41.10	41.11	41.10	40.81	40.85	40.71	40.61	40.69	40.88	40.87
Conv41,3×3	40.47	39.25	37.64	33.53	25.19	17.15	8.13	5.48	5.33	4.64
Conv42,1×1	41.11	41.13	40.97	40.94	40.99	40.91	40.89	40.88	40.90	40.68
Conv43,3×3	39.98	39.30	37.78	35.96	33.18	26.88	19.57	14.33	7.12	5.40
Conv44,3×3	40.89	40.71	40.57	40.25	39.60	39.22	38.25	37.37	34.81	27.73
Conv45,1×1	41.17	41.08	41.06	41.10	41.07	41.01	41.00	40.77	40.80	40.35
Conv46,3×1	40.91	40.63	40.23	39.02	38.00	35.97	33.03	28.75	27.16	23.24
Conv47,1×1	41.14	41.14	40.98	41.05	41.07	41.05	40.97	40.97	41.22	40.70
Conv48,3×3	40.91	40.72	40.36	39.73	38.70	37.65	35.62	33.00	29.21	25.51
Conv49,1×1	41.14	40.88	40.83	40.75	40.78	40.44	39.95	39.56	38.20	37.77
Conv50,3×3	40.82	40.53	39.86	39.05	37.84	36.39	34.56	31.43	28.49	25.40
Conv51,1×1	41.15	41.06	41.05	41.10	41.02	41.00	40.76	40.77	40.70	40.29
Conv52,3×3	40.96	40.75	40.48	39.51	38.97	37.09	35.85	31.73	27.76	24.51

4 实验结果

本文所有模型训练及模型压缩过程均在 NVIDIA 2080Ti 中进行,最终模型的 MPSOC 验证在百度 Edgeboard FZ3 开发板中进行。FZ3 主控芯片采用 Xilinx 的 MPSOC 平台 ZYNQ,具体型号为 XAZU3EG-1SFVC784I。

XAZU3EG 芯片提供的硬件资源具体如图 6 所示。可以看出 ZYNQ 芯片由两部分组成,分别为多核同构的 ARM(Advanced RISC Machine)部分(PS)和传统现场可编程门阵列(FPGA)部分

(PL)。这种异构结构将处理器的软件可编程性与 FPGA 的硬件可编程性有机结合在一块芯片上。利用片内 AXI 方式代替传统的 PCI-E 通信,可以带来更大的通信带宽和通信稳定性。具体的硬件资源为:运行速率高达 1.5 GHz 的四核 ARM Cortex-A53 平台,双核 Cortex-R5 实时处理器、Mali-400 MP2 图形处理单元及 16 nm Fin-FET+可编程逻辑。在算力方面,与模型训练平台 RTX 2080Ti 相比,ZYNQ 系列 MPSOC 远落后于其 1 755 MHz 的核心频率和 4 352 个流处理单元。但在功耗方面,MPSOC 却拥有着巨大的优势。正常工作时 MPSOC 功耗只有 7~10 W,

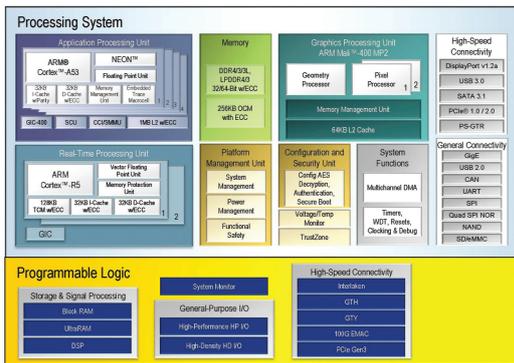


图 6 MPSOC 资源介绍^[17]

Fig.6 Introduction to MPSOC hardware resources

RTX 2080Ti 却达到了 260 W。在便携性和体积方面,ZYNQ 系列 MPSOC 拥有更大的优势。采用 MPSOC 异构结构同时简化了开发过程,PS 部分为主控部分,可以先行验证速算法的可行性,之后利用 FPGA 部分对算法进行硬件加速,提高算法性能。在开发工具方面,Xilinx 提供的 vivado 系列工具极大地简化了开发流程。本文同时对系统环境进行了搭建,具体工作包括利用 Petalinux 制作 Linux 系统,编译与移植 OpenCV 库及百度开发的深度学习接口工具 PaddleLite 等。MPSOC

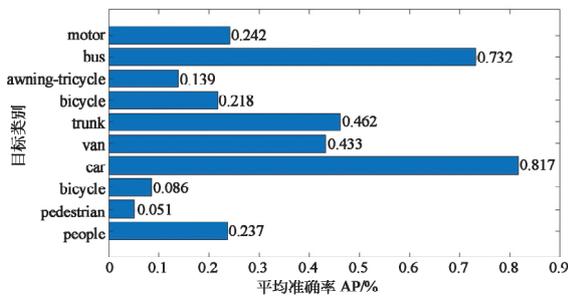


图 7 优化前模型 AP

Fig.7 AP index of the model before optimization

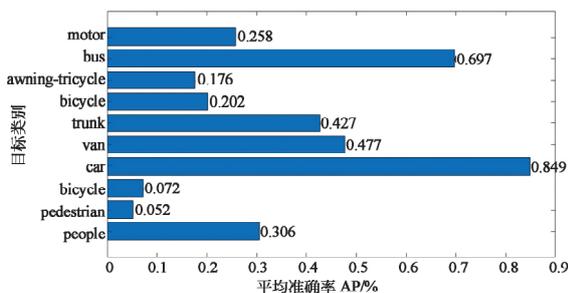


图 8 优化后模型 AP

Fig.8 AP index of the optimized model

平台外接的硬件设备包括 USB 摄像头与 HDMI 显示器,分别用于对图像进行采集与结果显示。

对优化前后模型的性能进行分析,评价指标为平均准确率(Average Precision, AP)和召回率。其中 AP 用于衡量模型对某一类的检测精度。对图 7 和图 8 比较可以发现,大多数类别的 AP 都有所上升。这是由于去除一个降采样过程后,不只是小目标的特征得到了更好的表达,识别难度不大的大目标特征也能得到更好的表达。

另一个评价指标为召回率。召回率用于描述阳性样本经过模型后的输出结果依旧为阳性的样本占有所有阳性的比例。每一类的召回率所取的值为 $P-R$ 曲线中最小非零精确度所对应的召回率。从图 9 可以发现优化后模型的各个类的召回率都有所上升。其中对于 car 这一类目标的召回率达到了 93% 以上。在图 10 中横坐标为模型回归所得到的预测框的数量和数据集中实际拥有边框的比值。以 people 为例,其验证集中共有目标 21 006 个,优化前模型检测出 people 类共有 8 053 个,优化后模型检出 4 946 个,所以其横坐标

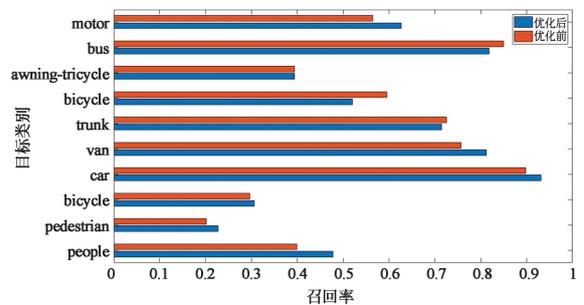


图 9 优化前后模型召回率变化

Fig.9 Change of model recall rate before and after optimization

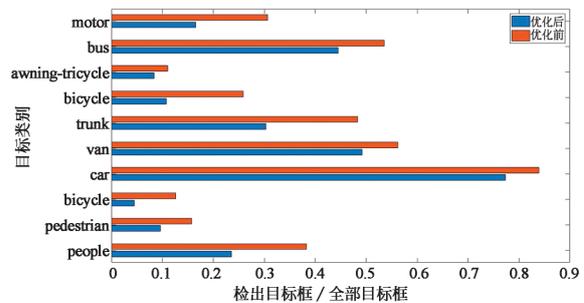


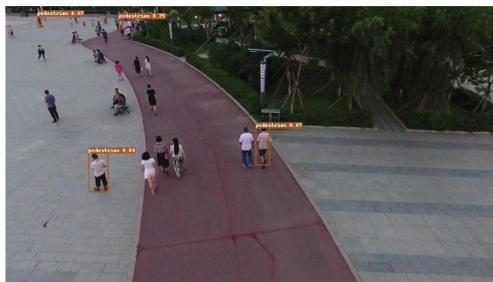
图 10 优化前后模型回归边框数量变化

Fig.10 Number of regression borders changed before and after optimization

分别为 0.38 和 0.23。结合图 9 可以发现,优化前的模型可以检测较多的 people 预测框,但是其召回率却不高,这意味着将很多不是 people 的物体检测为 people,误检率较高。而这种误检情况在优化前模型各个类别中普遍存在。与原始模型相比优化后的模型误检率降低,准确度得到了较大提高。

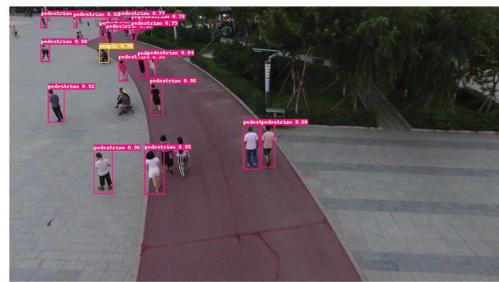
图 11 所示为改进前后模型对同一张图片的检测效果的比较。左侧图(a)、(c)为原始模型检测结果,右侧图(b)、(d)为优化后模型的检测结果。

可以发现无论是对于小目标还是大目标,优化后的网络检测精度都得到了明显提高。利用 VISDRONE 测试集对算法改进前后、剪枝前后的性能进行了统计比较。如表 4 所示,其中 MAP50 为 10 个类的 AP50 平均后的结果,可以得到优化后网络的 MAP50 提高了 1.3%,性能甚至优于最新的 YoloV4 算法。通过基于敏感度的剪枝后模型 MAP 虽然有所下降,但在算法规模方面剪枝后网络却达到了 91.9 M,是原模型 246 M 的 37%。检测速度达到 34.4 fps,是原模型的 2 倍。



(a)优化前模型检测结果一

(a) Detection result 1 of the model before optimization



(b)优化后模型检测结果一

(b) Detection result 1 of the optimized model



(c)优化前模型检测结果二

(c) Detection result 2 of the model before optimization



(d)优化后模型检测结果二

(d) Detection result 2 of the optimized model

图 11 检测结果对比图

Fig.11 Comparison of the test results

表 4 模型性能比较

Tab.4 Model performance comparison

	原始 Yolo V3	改进 Yolo V3	改进 Yolo V3+剪枝	Yolo V4
MAP 50/%	36.1	37.4	30.6	28.1
速度/fps(608×608)	45.4	17.2	34.4	30.4
体积/M	246.5	246.5	91.9	256.2

同时将剪枝后的模型在 MPSOC 平台中进行了实景验证,搭建的实验平台如图 12 所示。该算法对视频序列中出现的一些较小目标也取得了较



图 12 MPSOC 硬件平台

Fig.12 MPSOC hardware platform

好的效果,相比较原始算法,速度方面也有了较大提升。对于接入的 608×608 图像,检测速度可以达到 13 fps,基本满足航空图像目标检测的要求。与 GPU 平台相比,该平台可以在资源有限的情况下实现对航空图像中多类目标的检测,同时拥有更高的便携性和更低的功耗。

5 结 论

本文针对航空图像目标检测的特点,通过改变锚框和调整卷积层结构对 Yolo V3 算法进行优化,使算法的 MAP 提高了 1.3%。同时利用基于敏感度的剪枝算法对模型进行压缩,参数规模变为原来的 37%,检测速度提高了 1 倍。最后在 MPSOC 平台进行了实验验证,并取得了较好的检测效果。实验结果表明,该算法基本满足航空图像目标检测的需求,同时系统平台也为深度学习算法在无人机端的移植奠定了基础。

参 考 文 献:

- [1] 梁华,宋玉龙,钱锋,等.基于深度学习的航空对地小目标检测[J].液晶与显示,2018,33(9):793-800.
LIANG H, SONG Y L, QIAN F, *et al.* Detection of small target in aerial photography based on deep learning [J]. *Chinese Journal of Liquid Crystals and Displays*, 2018, 33(9): 793-800. (in Chinese)
- [2] 马永杰,宋晓凤.基于 YOLO 和嵌入式系统的车流量检测[J].液晶与显示,2019,34(6):613-618.
MA Y J, SONG X F. Vehicle flow detection based on YOLO and embedded system [J]. *Chinese Journal of Liquid Crystals and Displays*, 2019, 34(6): 613-618. (in Chinese)
- [3] 裴信彪,吴和龙,马萍,等.基于无人机遥感的不同施氮水稻光谱与植被指数分析[J].中国光学,2018,11(5):832-840.
PEI X B, WU H L, MA P, *et al.* Analysis of the spectrum and vegetation index of rice under different nitrogen levels based on unmanned aerial vehicle remote sensing [J]. *Chinese Optics*, 2018, 11(5): 832-840. (in Chinese)
- [4] 裴伟,许晏铭,朱永英,等.改进的 SSD 航拍目标检测方法[J].软件学报,2019,30(3):738-758.
PEI W, XU Y M, ZHU Y Y, *et al.* The target detection method of aerial photography images with improved SSD [J]. *Journal of Software*, 2019, 30(3): 738-758. (in Chinese)
- [5] BOYKOV Y, HUTTENLOCHER D P. A bayesian framework for model based tracking [R]. Place Ithaca, NY, United States: Cornell University, 1999.
- [6] ALI S, SHAH M. COCOA: tracking in aerial imagery [C]// *Proceedings of SPIE 6209, Airborne Intelligence, Surveillance, Reconnaissance (ISR) Systems and Applications III*. Orlando (Kissimmee), Florida, United States: SPIE, 2006: 62090D.
- [7] IBRAHIM A W N, CHING P W, SEET G L G, *et al.* Moving objects detection and tracking framework for UAV-based surveillance [C]// *Proceedings of 2010 Fourth Pacific-Rim Symposium on Image and Video Technology*. Singapore: IEEE, 2010.
- [8] 谭熊,余旭初,刘景正,等.基于无人机视频的运动目标快速跟踪[J].测绘通报,2011(9):32-34,41.
TAN X, YU X C, LIU J Z, *et al.* Object fast tracking based on unmanned aerial vehicle video [J]. *Bulletin of Surveying and Mapping*, 2011(9): 32-34, 41. (in Chinese)

- [9] 汤轶,周鹏程,肖璇,等.基于无人机平台的运动目标检测与跟踪算法研究[J].机器人技术与应用,2017(3):35-37.
TANG Y, ZHOU P C, XIAO X, *et al.* Research on moving target detection and tracking algorithm based on UAV platform [J]. *Robot Technique and Application*, 2017(3): 35-37. (in Chinese)
- [10] 李航,朱明.基于深度卷积神经网络的小目标检测算法[J].计算机工程与科学,2020,42(4):649-657.
LI H, ZHU M. A small object detection algorithm based on deep convolutional neural network [J]. *Computer Engineering & Science*, 2020, 42(4): 649-657. (in Chinese)
- [11] 范丽丽,赵宏伟,赵浩宇,等.基于深度卷积神经网络的目标检测研究综述[J].光学精密工程,2020,28(5):1152-1164.
FAN L L, ZHAO H W, ZHAO H Y, *et al.* Survey of target detection based on deep convolutional neural networks [J]. *Optics and Precision Engineering*, 2020, 28(5): 1152-1164. (in Chinese)
- [12] 陈科峻,张叶.基于YOLO-V3模型压缩的卫星图像船只实时检测[J].液晶与显示,2020,35(11):1168-1176.
CHEN K J, ZHANG Y. Real-time ship detection in satellite images based on YOLO-v3 model compression [J]. *Chinese Journal of Liquid Crystals and Displays*, 2020, 35(11): 1168-1176. (in Chinese)
- [13] 谢晓竹,薛帅.改进YOLOv3模型对航拍汽车的目标检测[J].装甲兵工程学院学报,2019,33(3):84-89.
XIE X Z, XUE S. Target detection of aerial photographic vehicles based on improved YOLOv3 model [J]. *Journal of Academy of Armored Force Engineering*, 2019, 33(3): 84-89. (in Chinese)
- [14] 李宇,刘雪莹,张洪群,等.基于卷积神经网络的光学遥感图像检索[J].光学精密工程,2018,26(1):200-207.
LI Y, LIU X Y, ZHANG H Q, *et al.* Optical remote sensing image retrieval based on convolutional neural networks [J]. *Optics and Precision Engineering*, 2018, 26(1): 200-207. (in Chinese)
- [15] HU H Y, PENG R, TAI Y W, *et al.* Network trimming: a data-driven neuron pruning approach towards efficient deep architectures [J]. *arXiv*:1607.03250, 2016.
- [16] LI H, KADAV A, DURDANOVIC I, *et al.* Pruning filters for efficient ConvNets [C]// *Proceedings of the International Conference on Learning Representations*. Toulon, France; ICLR, 2017.
- [17] Xilinx. Zynq UltraScale+ MPSoC product tables and product selection guide [R]. American; Xilinx, 2016.

作者简介:



任 彬(1996—),男,山东淄博人,硕士研究生,2018年于哈尔滨理工大学获得学士学位,主要从事计算机视觉,数字图像处理方面的研究。E-mail: Ren1393138227@163.com



王宇庆(1979—),男,吉林长春人,副研究员,2008年于中科院长春光学精密机械与物理研究所获得博士学位,主要研究方向:基于人工智能的快速目标检测方法、人工智能芯片的研究与应用、面阵三维成像、嵌入式数字图像处理、图像质量评价等。E-mail: wyq7903@163.com