DOI:10.19652/j.cnki.femt.2405924

基于 MPSoC 的轻量化汽车检测系统及硬件加速平台设计与优化*

王 伟[1] 王 坤[2] 许圳兴[2] 付相为[2]
(1 无锡学院江苏省集成电路可靠性技术与检测系统工程研究中心 无锡 214105;
2 南京信息工程大学电子与信息工程学院 南京 210044)

摘 要:针对车辆分类检测在精度和实时性方面存在的挑战,提出了一项改进方案,以优化 YOLOv5s 模型,旨在实现轻量化的汽车检测。通过在 MPSoC 硬件架构的现场可编程门阵列(FPGA)上设计系统,成功打造了一个具备高精度、快速检测和低能耗的解决方案。为了使得模型更适合嵌入式设备部署,采用了 MobileNetv3 Small 替代 YOLOv5s 的主干网络,并引入卷积块注意力模块(CBAM)注意力机制和 Inner-IoU Loss 优化方法,使模型在轻量化的同时提升了检测精度和速度。改进后的模型相较于原始 Yolov5s 模型,平均精度均值(mAP)提升了 14.8%,参数量减少了 49.7%,模型体积减小了 40.7%,计算量减少了 48.9%,在 NVIDIA 3060 上,改进后的检测速度提升了 48.8%,达到了 82 fps。此外,还利用 FPGA 对 YOLOv5s 进行了硬件加速。经过优化的系统达到了 45 fps 的检测帧率,并保持了较高的精度和速度,这一系统易于部署,适用于智能交通系统,满足其高效实时监测的需求。

关键词:车辆分类;YOLOv5s 轻量化;MobileNetv3 Small;FPGA;硬件部署
中图分类号: TN2 **文献标识码**:A **国家标准学科分类代码**: 520.60

Design and optimization of a lightweight automotive detection system and hardware acceleration platform based on MPSoC

Wang Wei[1] Wang Kun[2] Xu Zhenxing[2] Fu Xiangwei[2]
(1. Jiangsu Province Engineering Research Center of Integrated Circuit Reliability Technology and Testing System, Wuxi University, Wuxi 214105, China; 2. College of Electrical and Information Engineering, Nanjing University of Information Science & Technology, Nanjing 210044, China)

Abstract: In response to the challenges regarding accuracy and real-time performance in vehicle classification detection, this study proposes an improved lightweight model for vehicle detection based on YOLOv5s. The objective to achieve a solution that balances high detection accuracy, swift detection, and low power consumption through a system designed on an FPGA within an MPSoC hardware architecture. In order to make the model more suitable for embedded device deployment, This research replaces the backbone network of YOLOv5s with MobileNetv3 Small and incorporates CBAM attention mechanism and Inner-IoU Loss optimization. This modification aim to achieve lightweighting while enhancing detection accuracy and speed. Compared to the original YOLOv5s model, the enhanced model exhibits a 14.8% increase in mAP, a reduction of 49.7% in parameters, a 40.7% decrease in model volume, and a 48.9% decrease in computational load. On the NVIDIA 3060 platform, the improved detection speed has surged by 48.8%, reaching 82 fps. Additionally, hardware acceleration using FPGA has been implemented for YOLOv5s. The optimized system achieves a detection frame rate of 45 fps while maintaining high precision and speed. This system is easily deployable and suits the demands of intelligent transportation systems, fulfilling the need for efficient real-time monitoring.

Keywords: vehicle classification; YOLOv5s lightweight; MobileNetv3 Small; FPGA; hardware deployment

收稿日期:2024-01-09
* **基金项目**:南京信息工程大学滨江学院人才启动科研项目(2019r005,550219005)、企业横向(2021320205000041,2023320205000242,2023320205000242)项目资助

0 引 言

为缓解交通阻塞和减少交通事故发生率，在深度学习和计算机硬件发展的基础上，提出了智慧交通和无人驾驶等解决方案，其中实时和准确的车辆检测是构建智能交通的必不可少部分，许多场景不仅对实时性和准确性具备严格要求，还对计算资源和能源消耗进行限制。[1]。基于深度学习的车辆检测方法比传统方法拥有更高的准确性，成为当下主流方法，但其较高的计算量成为嵌入式设备部署的难题，实时性与低功耗往往不能同时兼得。而 FPGA 具有可重构设计、高并行性以及流水线处理，在处理速度和功耗方面具有较大优势，异构多核处理片上系统（multi-processor system on chip，MPSoC）包含处理器系统（processing system，PS）和可编程逻辑（programmable logic，PL）两部分，利用 PL 端进行深度学习的数据计算，PS 负责后处理，可以加速车辆目标检测的速度，达到实时性以及低功耗设计[2]。

在汽车检测领域，会使用激光雷达和毫米波雷达，激光雷达会容易受到天气影响，而毫米波雷达识别精度较不理想。2020 年，张三川等[3]采用 77 GHz 的毫米波雷达传感器设计了汽车前向防撞探测系统装置，在对两个动态目标进行同时探测，速度误差控制在了 5%以内，这种方法在一定程度上能够克服恶劣天气的影响，但在精确性和分辨率上会存在一些限制，图像识别在抗干扰和检测精度均具有优异的性能。2023 年，张壮壮[4]在现场可编程门阵列（FPGA）实现了卷积神经网络（convolutional neural networks，CNN）进行了对 MobileNet 的硬件加速提高计算效率，但对 FPGA 硬件资源消耗过高且灵活性差。为了兼顾模型的部署的灵活性同时能够降低功耗，本文使用了 Xilinx 深度学习处理器单元（deep learning processor unit，DPU）对卷积运算进行加速，DPU 是专用于卷积神经网络加速的可编程引擎，拥有专门的指令集，可以高效地为许多深度神经网络的卷积部分提供运算任务[5-6]。本文通过采用 MobileNetv3 算法对 YOLOv5s 进行轻量化优化，并部署在 MPSoC 嵌入式设备，利用 PL 进行并行计算，实现了高帧率低功耗的汽车分类识别，满足目标检测任务的实时性。

1 基于 YOLOv5s 算法改进

1.1 轻量化主干网络改进

YOLOv5s 模型结构包括 Input、Backbone、Neck 和 Head 4 部分。640×640 大小的图像由 Input 进行输入，并经过数据增强、自适应锚框以及自适应图像缩放等技术对图像进行预处理。Backbone 使用 CSPDarknet53 或 ResNet 骨干网络，用来提取图像特征的网络，将原始输入图像转化为多层特征图，提供后续的目标检测任务使用；Neck 部分采用了特征图金字塔网络（feature pyramid networks，FPN）和路径聚合网络（path aggregation network，PAN）组合而成的 PANet 网络，连接主干网络和头部网络的中间层，并融合不同尺度的特征图，用于处理不同尺度信息的部分；FPN 是通过在网络的不同层级上收集和整合特征图来构建的，目的是使网络能够对不同尺度的目标进行有效检测，解决目标检测任务中多尺度信息的问题，通过在不同层级上创建特征金字塔，从而使网络能够同时利用低层级（高分辨率但语义信息较少）和高层级（低分辨率但语义信息更丰富）的特征；PAN 是一种用于处理多尺度信息的网络结构，旨在解决深度神经网络中不同层级特征图的信息整合问题，该结构使用自上而下和自下而上的路径，从高层级到底层级和从底层级到高层级传递信息，这些路径可以通过上采样（upsampling）和池化等操作实现，而自上而下的路径用于传递高层级的语义信息，自下而上的路径用于传递底层级的细节信息，同时在不同层级上，通过融合相应的特征图，PAN 能够保留底层的高分辨率信息和高层的语义信息，该设计旨在提高网络对不同尺度物体的感知能力，通过整合多尺度信息，网络可以更好地适应不同大小的目标；Neck 部分的设计旨在提高模型对不同尺度目标的检测能力，使其更具适应性和鲁棒性，这有助于在处理复杂场景时提高模型的性能[7]。Head 部分涵盖了 3 个不同尺寸的检测分支，分别为 20×20、40×40 和 80×80，用于检测小、中、大目标，最终的最优目标框通过非极大值抑制（non maximum suppression，NMS）算法得到[8]。

本文旨在减轻 YOLOv5s 的计算负担并缩减其模型尺寸。鉴于 MobileNetv3 Small 架构在运行速度和计算复杂度方面的优势，本文通过将其应用于替换 YOLOv5s 的主干特征提取网络，成功降低了模型的参数数量，从而实现了模型的轻量化[9]。MobileNetv3 Small 的网络结构如图 1 所示，MobileNetV3 Small 基于倒置残差结构和线性瓶颈原理，结合了深度可分离卷积技术以大幅减少计算量和参数数量，该模型采用了 H-Swish 激活函数，这是一种高效的 Swish 函数近似，H-Swish 通过使用简单的 ReLU 操作提高了计算效率，同时保持了 Swish 的性能，进一步提升了计算效率，同时该架构通过网络结构搜索技术进行了优化，还融入了 SE（Squeeze-and-Excitation）模块，引入了注意力机制来强调重要特征[10]。MobileNetV3 Small 在保持较低延迟的同时，能够有效处理图像分类和目标检测等多种任务，适合于对实时处理和能效有严格要求的应用场景。

MobileNetv3 网络采用深度可分离卷积进行特征提取，这种卷积方式通过分离空间维度和通道维度的相关性，有效减少了卷积运算所需的参数数量，从而显著削减了计算负担和模型尺寸[11]。标准卷积包含卷积和特征融合两部分，卷积操作使用卷积核在输入特征图上进行卷积计算，提取出特征信息；特征融合是将不同卷积核提取出的特征信息进行融合，得到最终的特征表示。标准卷积核具有较高的准确性，但计算量和参数较大，并不利于嵌入

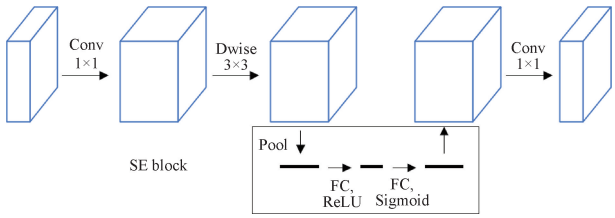

图 1　Bneck 网络结构

Fig. 1　Bneck network structure diagram

式系统的实时计算。假设使用卷积核尺寸为 $F_k \times F_k$，与长为 H，宽为 W 的特征图进行卷积，其中 M 为输入通道，输出通道数为 N，则标准卷积的计算量为[12]：

$$S = H \times W \times M \times N \times F_K \times F_K \tag{1}$$

深度可分离卷积技术将常规卷积分为深度卷积和 1×1 的点卷积两个步骤，这样做既减少了计算量又降低了参数数目，在这个过程中，深度卷积维持了标准卷积的特征提取能力，但不执行特征的融合操作，1×1 点卷积使用一个大小为 1×1 的卷积核对深度卷积的输出特征图进行卷积，从而得到最终的输出特征图[13]。深度卷积对输入特征图的每个通道应用一个卷积核，得到一个深度特征图，然后由 1×1 点卷积将不同深度特征图的每个位置上的像素进行线性组合，得到最终的输出特征图。1×1 点卷积实现了标准卷积中的特征融合功能，但计算量很小。深度可分离卷积的的计算量为[14]：

$$S = H \times W \times M \times F_K \times F_K + H \times W \times M \times N \tag{2}$$

由式(1)和(2)得到深度可分离卷积与标准卷积的计算量比值如下：

$$\frac{P}{S} = \frac{1}{N} + \frac{1}{F_K^2} \tag{3}$$

由式(3)可知，可以得知深度可分离卷积在降低网络计算量和参数数量方面发挥了作用，从而实现了模型的轻量化，并显著提高了检测速度[15]。深度卷积只在每一个输入通道上施加一个与深度相对应的卷积操作，这样就会造成深度卷积输出特征仅和相应输入相关联，降低了特征提取能力，造成检测精度的降低。

1.2　注意力机制改进

为了针对车辆检测出现重叠遮挡等情况，本文引入了轻量级的注意力机制卷积块注意力模块(CBAM)。CBAM 通过顺序集成通道和空间注意力两个子模块，优化了特征图的表征，通道注意力模块通过聚焦于不同通道的重要性来强调有意义的特征，而空间注意力模块则关注于特征图的哪些部分是最关键的，这种组合方式使 CBAM 能够在全局视野中综合考虑特征的重要性，从而提升网络在图像分类、目标检测等任务中的性能，CBAM 结构如图 2 所示[16]。CBAM 是一种高效的注意力机制，用来提高模型对关键特征的识别能力，其主要优势在于其轻量级且模块化的设计，使其能够轻松集成到各种现有的卷积神经网络结构中，无需大幅度修改网络架构或增加显著的计算负担[17]。综上得到改进后的 YOLOv5s 模型结构如图 3 所示。

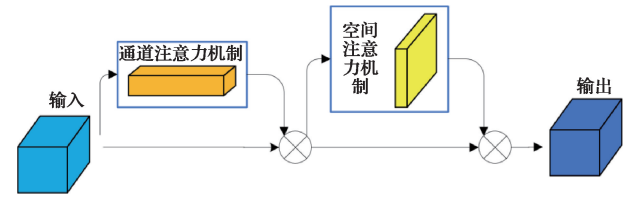

图 2　CBAM 网络结构

Fig. 2　CBAM network structure diagram

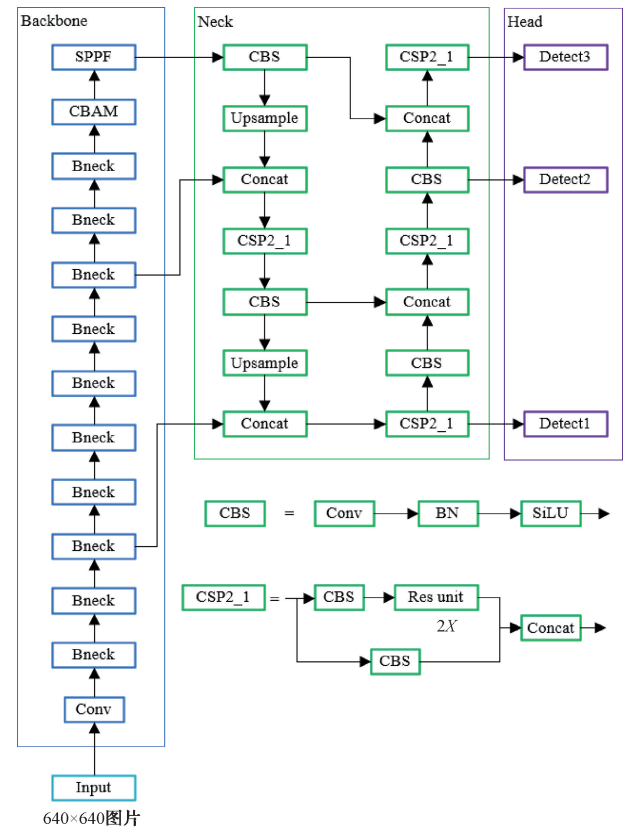

图 3　改进后 YOLOv5s 模型结构

Fig. 3　Improved YOLOv5s model structure

1.3　损失函数的改进

在 YOLOv5s 中采用的 CIoU Loss 是一种先进的损失函数，主要用于改进目标检测中的边界框回归，其优势在于能够更精确地反映目标框的匹配程度，它不仅考虑了边界框的重叠区域，还综合了中心点距离、长宽比例以及对角线长度等因素，这使得 CIoU Loss 在处理不同尺度的目标时更为鲁棒，特别是在小目标检测方面表现出更高的敏感性，从而提升检测精度，但是这种方法的缺点在于计算上相对更复杂，会带来一定的计算开销[18]。实验采用 Inner-IoU Loss 替代 CIoU Loss 作为损失函数，Inner-IoU

Loss 基于辅助边框计算 IoU 损失并针对不同的数据集与检测器，引入尺度因子 ratio 控制辅助边框的尺度大小用于计算损失[19]。如图 4 所示，GT 框和锚框分别使用 b^{gt} 和 b 表示，GT 框内部的中心点用 (x_c^{gt},y_c^{gt}) 表示，使用 (x_c,y_c) 表示内部锚框的中心点；GT 框的宽度和高度分别表示为 w^{gt} 和 h^{gt}，而锚框的宽度和高度分别表示为 w 和 h，尺度因子 $ratio$ 对应的的取范围为[0.5,1.5]。

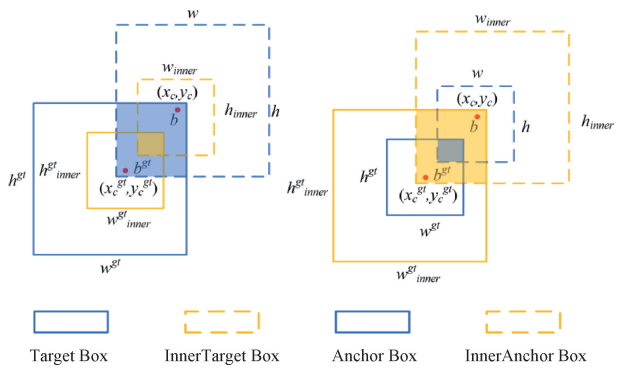

图 4 Inner-IoU Loss 示意图

Fig. 4 Schematic diagram of inner IoU Loss

Inner-IoU Loss 定义如下：

$$b_l^{gt}=x_c^{gt}-\frac{w^{gt}\cdot ratio}{2},b_r^{gt}=x_c^{gt}+\frac{w^{gt}\cdot ratio}{2} \tag{4}$$

$$b_t^{gt}=y_c^{gt}-\frac{h^{gt}\cdot ratio}{2},b_b^{gt}=y_c^{gt}+\frac{h^{gt}\cdot ratio}{2} \tag{5}$$

$$b_l=x_c-\frac{w\cdot ratio}{2},b_r=x_c+\frac{w\cdot ratio}{2} \tag{6}$$

$$b_t=y_c-\frac{h\cdot ratio}{2},b_b=y_c+\frac{h\cdot ratio}{2} \tag{7}$$

$$inter=(min(b_r^{gt},b_r)-max(b_l^{gt},b_l))\cdot(min(b_b^{gt},b_b)-max(b_t^{gt},b_t)) \tag{8}$$

$$union=(w^{gt}\cdot h^{gt})\cdot(ratio)^2+(w\cdot h)\cdot(ratio)^2-inter \tag{9}$$

$$IoU^{inner}=\frac{inter}{union} \tag{10}$$

Inner-IoU Loss 除了具备自身特点外还继承了 IoU Loss 的一些特征，Inner-IoU Loss 取值范围为[0,1]。Inner-IoU Loss 辅助边框与实际边框仅存在尺度上的差异，损失函数计算方式相同，Inner-IoU Deviation 曲线与 IoU Deviation 曲线相似[20]。与 IoU Loss 相比较，当 $ratio<1$，辅助边框尺寸小于实际边框，其回归的有效范围小于 IoU Loss，但其梯度绝对值大于 IoU Loss 所得的梯度，能够加速高 IoU 样本的收敛。当 $ratio>1$，较大尺度的辅助边框扩大了回归的有效范围，对于低 IoU 的回归有所增益。将 Inner-IoU Loss 应用在基于 IoU 的边框回归损失函数中，其 $L_{Inner-IoU}$、$L_{Inner-GIoU}$、$L_{Inner-DIoU}$、$L_{Inner-CIoU}$、$L_{Inner-EIoU}$、$L_{Inner-SIoU}$ 定义如下：

$$L_{Inner-IoU}=1-IoU^{inner} \tag{11}$$

$$L_{Inner-GIoU}=L_{GIoU}+IoU-IoU^{inner} \tag{12}$$

$$L_{Inner-DIoU}=L_{DIoU}+IoU-IoU^{inner} \tag{13}$$

$$L_{Inner-CIoU}=L_{CIoU}+IoU-IoU^{inner} \tag{14}$$

$$L_{Inner-EIoU}=L_{EIoU}+IoU-IoU^{inner} \tag{15}$$

$$L_{Inner-SIoU}=L_{SIoU}+IoU-IoU^{inner} \tag{16}$$

2 改进 YOLOv5s 算法数据分析

2.1 实验环境

本文数据集经网络搜集并进行归类手工标注，该数据集包含 7 种类别，分别是一类客车(tinycar)、二类客车(midcar)、三类客车(bigcar)、一类货车(smalltruck)、二类货车(bigtruck)、油罐车(oil truck)以及特殊车辆(specialcar)。实验按照 4∶1 的比例将数据集划分为训练集和测试集，本文使用训练环境如表 1 所示。

表 1 训练环境

Table 1 Training environment

名称	参数
CPU	Intel(R) Core i7-12700@2.10 GHz
GPU	NVIDIA GeForce RTX 3060
操作系统	Windows 10 22H2 专业版
深度学习环境	PyTorch+cuda11.7+cudnn8.5.0

实验参数配置如下：Batchsize 设置为 32，图片大小设置为 640×640，初始学习率为 0.01，动量设置为 0.94，衰减系数设置为 5×10^{-4}，IoU 阈值设置为 0.45。

2.2 改进前后模型对比

本文针对对原始和改进后的 YOLOv5s 模型进行了训练，并对比检测精度。如图 5 所示，经过改进的 YOLOv5s 模型在平均精度均值(mAP)mAP@0.5 指标上达到了 0.921%，相比原模型提高了 14.5%。此外，7 个汽车类别的识别准确性也均有所增强。其中一类客车提升 8.7%，二类客车提升 5.2%，三类客车提升 11.9%，一类货车提升 10.3%，二类货车提升 12.8%，油罐车提升 4.5%，特殊车辆提升 1.6%。

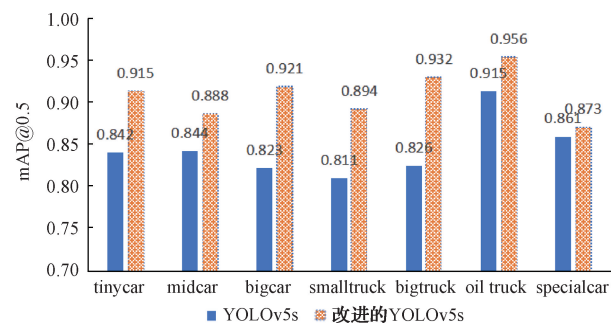

图 5 改进前后检测精度对比

Fig. 5 Comparison of detection accuracy before and after improvement

实验不仅对 YOLOv5s 模型进行了改进，还比较了改进前后模型在参数数量、尺寸、检测速度和计算需求方面的差异，结果如表 2 所示。根据表 2 数据得出改进版的 YOLOv5s 在整体性能上超越了原版。改进版模型的参数减少了 49.7%，体积缩小了 40.7%，计算需求降低了 48.9%。此外，检测速度提高了 48.8%，达到了 82 fps。

表 2　改进前后性能对比

Table 2　Performance comparison before and after improvement

模型	参数量	体积/MB	计算量/GFLOPs	检测帧率/fps
YOLOv5s	7 096 833	27	18.0	42
改进 YOLOv5s	3 569 842	16	9.2	82

2.3　不同模型对比

为了确证所提出模型的有效性，本文在统一的数据集和实验条件下，执行了与其他模型的比较性实验，结果如表 3 所示。实验针对 YOLOv3-tiny、YOLOv5s、YOLOv7-tiny、YOLOv8s、文献[21-22]进行了对比，其中改进的 YOLOv5s 模型的 mAP@0.5 比 YOLOv3-tiny、YOLOv5s、YOLOv7-tiny、文献[21-22]分别提升了 33.1%、14.8%、7.6%、11.8%和 2.0%；检测速度比 YOLOv5s、YOLOv7-tiny、YOLOv8s、文献[21-22]分别提升了 40、20、37、14 和 39 fps；模型体积比 YOLOv5s、YOLOv7-tiny、YOLOv8s、文献[21-22]分别减少了 40.7%、50.0%、69.2%、30.4%以及 51.5%。在对比实验中，YOLOv8s 模型展现出最高的检测精度，达到 94.3%。相比之下，虽然本文的改进 YOLOv5s 模型的检测精度略有不足，但其检测速度更快，达到 82 fps，并且模型大小减少了 36 MB。与此同时，YOLOv3-tiny 模型虽然体积最小，且检测速度最高，达到 96 fps，但其检测精度较低。在处理车辆检测问题时，重要的是在确保较高检测速度的同时提升检测精度。这些对比结果表明，改进的 YOLOv5s 模型在检测速度和精度方面实现了良好的均衡。

表 3　不同模型对比结果

Table 3　Comparison results of different models

模型	mAP@0.5	检测帧率/fps	体积/MB
YOLOv3-tiny	0.692	96	13
YOLOv5s	0.802	42	27
YOLOv7-tiny	0.856	62	32
YOLOv8s	0.943	45	52
文献[21]	0.824	68	23
文献[22]	0.903	43	33
改进 YOLOv5s	0.921	82	16

2.4　消融实验

为验证不同设计方案对模型的性能影响，在实验环境以及其他参数不变的情况下，以 YOLOv5s 为基准进行消融实验，实验结果如表 4 所示。实验表明，将 YOLOv5s 主干网络更改为 MobileNetv3 Small，检测精度有所下降，但检测速度提升到了 132 fps；随后引入 CBAM 注意力后，帧率稍微下降，但检测精度提升了 6.4%；而将 Inner-IoU Loss 替代 CIoU Loss 作为损失函数，检测精度提升到了 0.921，比未改进之前提升了 14.8%。将 YOLOv5s 主干网络更改为 MobileNetv3 Small，添加 CBAM 注意力和采用 Inner-IoU Loss 时，在模型大小和精度之间达到最佳的平衡，达到预期实验效果，为后续硬件部署做好了准备。

表 4　消融实验

Table 4　Ablation experiment

序号	MobileNetv3 Small	CBAM 注意力	Inner-IoU Loss	mAP@0.5	检测帧率/fps
1				0.802	42
2	√			0.796	132
3	√	√		0.853	89
4	√	√	√	0.921	82

3　基于 MPSoC 硬件加速平台设计

3.1　DPU 模型构建与硬件平台设计

通过使用 MobileNetv3 Small 将 YOLOv5s 主干网络轻量化，同时使用 CBAM 注意力机制和 Inner-IoU Loss 对精度进行提升，使模型大小和精度之间达到最佳的平衡，这种优化使得模型更适合各种边缘设备和嵌入式系统，扩展了模型应用的范围，降低功耗并延长设备的使用时间。模型轻量化可以显著减小模型的体积和计算需求，使其能够在这些资源受限的设备上高效运行。通过轻量化模型，可以降低计算负担，提高推理速度，从而满足实时性的要求。

Vitis-AI 是 AMD 公司推出的嵌入式 AI 开发工具，用于在 ZYNQ MPSoc 系列 FPGA 的硬件平台进行 AI 推断，工具包含已经优化的深度学习处理器单元 IP 核、深度神经网络模型库、运行库以及工具库构成，Vitis-AI 包含优化器、量化器、编译器以及性能可视化分析工具[23]。其开发框架如图 6 所示，前端使用 PyTorch 框架进行训练，将训练后的模型进行 Freeze 得到 32 bit 的浮点网络模型结构(YOLOv5s_car.pb)文件，然后经数据预处理脚本进行量化校准，去除模型冗余部分并转化成无符号 8 bit 的数据类型，该步骤得到能够在嵌入式平台进行部署的网

络模型模型(yolov5s_car_model.pb),随后经编译工具映射为DPU指令集,编译后的模型包含了深度学习模型在特定Xilinx硬件上的编译结果,包含了在硬件上执行模型所需的指令和操作,描述了模型的计算图、层之间的连接以及相应的计算任务;同时将模型的层类型、权重、激活值等信息转化为硬件可执行的形式,以便在硬件上进行推理,还包含优化器对模型进行的特定硬件优化的信息,使深度学习模型在硬件上能够高效运行。最后将工具库、DPU驱动库以及应用程序(App.elf)部署至MPSoc4EV平台,实现检测加速。在FPGA使用DPU进行推理可以大幅提高计算速度和降低功耗,同时能够保证检测精度。

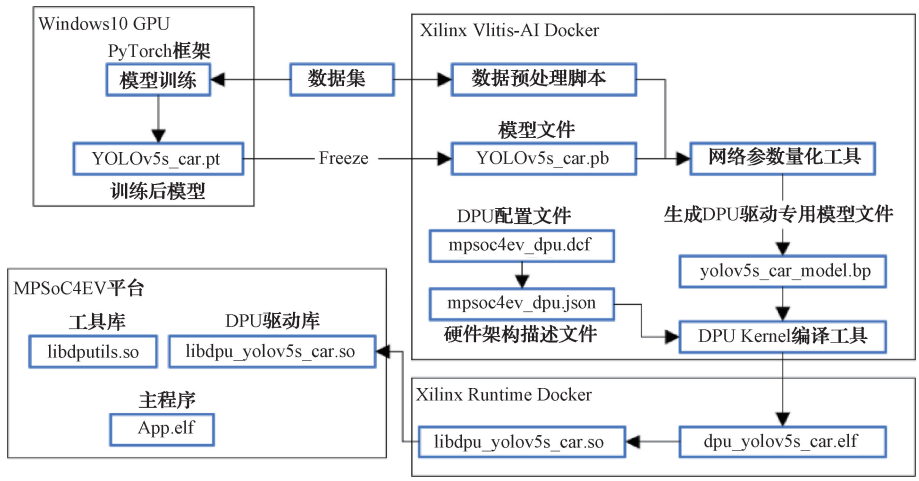

图6 Vitis-AI DPU开发框架图

Fig.6 Vitis AI DPU development framework diagram

整个工程采用Vivado 22.2软件环境进行搭建,DPU硬件连接设计如图7所示,DPU IP核作为模块集成到所用ZYNQ器件的PL中,并通过AXI总线连接到PS中,PS端通过AXI-Lite总线负责配置DPU,同时DPU通过PS提供的三路AXI HP高带宽接口,访问特定内存地址中输入的图像数据以及输出处理后的数据。本文实验还包含PS端NVME、网络和PL端HDMI接口配置,NVME负责存储图片等数据,网络负责主机与开发平台的数据交互,同时HDMI接口提供图形界面能够直观展示输出结果。整个系统使用PetaLinux 22.2进行系统构建和应用开发,ARM处理器为系统核心,负责完成应用层任务,对DPU进行功能配置,接收处理DPU的数据流,并进行后处理完成实时检测。DPU端进行深度学习的卷积加速运算[24]。

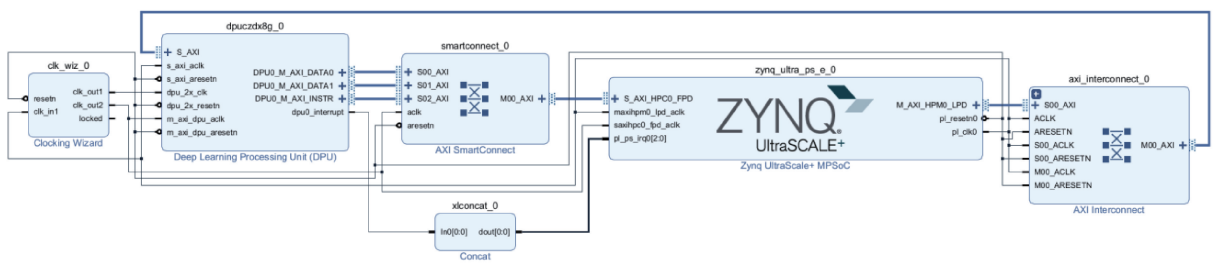

图7 DPU硬件连接设计

Fig.7 DPU hardware connection design

3.2 硬件加速平台软件系统构建

本文使用的AXU4EV-P硬件平台是基于Zynq UltraScale+MPSoc4EV架构的FPGA平台,集成了四核ARM Cortex-A53内核,提供高度集成的计算和通信功能,在设计中负责对图片进行缩放操作以及后处理;FPGA部分例化了DPU内核,并通过AXI AmartConnect总线连接至PS部分,进行读取和写入模型参数、输入数据和输出数据。其硬件加速平台如图8所示,开发板通过网线连接网络,进行程序和数据文件的拷贝,通过串口线接收用户指令以及完成程序部署,通过HDMI显示屏直观展示程序运行结果。系统构建先由VIvado22.2生成XSA硬件平台文件,通过PetaLinux 22.2创建Linux系统工程,经编译后会生成系统镜像文件。

将构建好的系统烧写进开发板中,待系统启动后,将改进后的YOLOv5s模型以及检测程序部署至硬件平台并将车辆数据集复制到固态存储盘中,最后执行检测程序,测得平均时间为22 ms,即FPGA加速系统的检测帧率为45.5 fps,平均每秒可以检测45张图片。为了进一

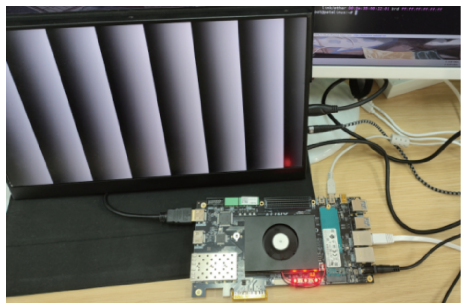

图 8　硬件环境搭建

Fig. 8　Hardware environment setup

步分析设计提出的加速系统的检测性能，在 Intel(R) Core i7-12700 以及 ARM-Cortex-A76 平台使用 OnnxRuntime 推理框架进行目标检测推理，对比三者检测精度以及检测速度分析设计的优点与不足。其加速性能对比如表 5 所示，ARM-Cortex-A76 平台推理时间远大于 Intel(R) Core i7-12700 平台，但是识别精度最高，ZYNQ-XCZU4EV-DPU 平台由于模型进行了量化，检测精度稍有下降，但仍然保持在很高的状态，其推理时间远远小于 Intel(R) Core i7-12700 和 ARM Cortex-A76，满足在工程应用的实时性。该系统在 Vivado22 软件平台综合结果如图 9 所示。图 9(a)为硬件逻辑资源消耗，该系统在 MPSoC4EV 芯片平台上所使用的资源主要是存储单元(block RAM，BRAM)，DPU 进行卷积运算需要使用 RAM 进行数据缓冲，因此占用了大量的 BRAM 资源。从图 9(b)可以看出，芯片的负载约为 6 W，当系统处于低功耗时，静态功耗只有 0.5 W，因此该系统不仅具备较强的实时性还能够保证较低的功耗，能够进行小型嵌入式产品设计。

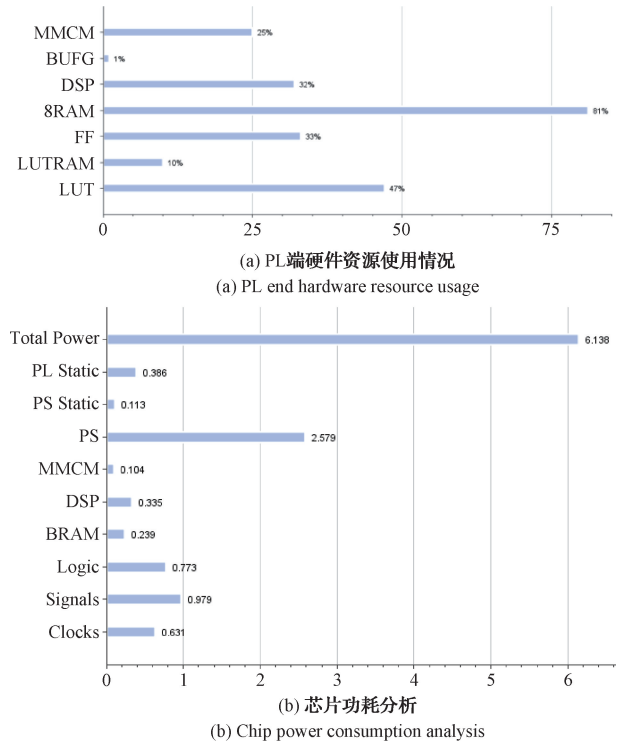

图 9　Vivado22 综合结果

Fig. 9　Comprehensive results of Vivado22

表 5　模型在不同平台测试结果

Table 5　Model test results on different platforms

处理平台	mAP	平均时间/ms
Intel(R) Core i7-12700	0.933	1 256
ARM Cortex-A76	0.922	2 365
ZYNQ-XCZU4EV-DPU	0.896	22

3.3　检测效果对比

实验选取一组图片，利用改进前后的模型在电脑端进行测试，同时也在硬件端进行同一组图片测试，检测结果如图 10 所示。由图 10 得知，改进后的 YOLOv5s 模型均检测精度具有明显的提升，并提升了特殊车辆的检测精

图 10　模型检测效果

Fig. 10　Model detection effect diagram

度。而在硬件部署的检测结果与使用电脑检测的精度稍微下降，与未改进的 YOLOv5s 精度具有小幅度提升。检验结果证实了改进的 YOLOv5s 模型的有效性，其在硬件部署不仅具有较高的精度还具有实时性，为车辆实时分类检测提供了方案。

4 结 论

本文针对车辆分类的检测任务在嵌入式设备无法做到实时性的问题，设计了一种基于 MPSoC 的检测系统。首先为了使得模型更适合嵌入式设备部署，针对 YOLOv5s 进行改进优化，包括将 YOLOv5s 主干网络更改为 MobileNetv3 Small，添加 CBAM 注意力并使用 Inner-IoU Loss 损失函数，模型体积减小 40.7%，检测速度达到了 82 fps，实验表明基于 YOLOv5s 改进后的模型，在实现轻量化和模型精度上均具有优异的表现，适合在嵌入式设备进行部署，能够用于汽车分类实时监测系统的生产需求。然后本文基于 MPSoC 架构的 FPGA 芯片构建了硬件加速平台，利用 PL 构建 DPU 处理平台，PS 负责后处理，该系统检测帧率达到了 45 fps，与 Intel(R) Core i7-12700 和 ARM-Cortex-A76 平台相比检测速度分别提升了 54 和 106 倍，同时芯片的负载约为 6 W。该系统在汽车分类的检测的精度、速度以及功耗方面具有很大改善，实现了速度和功耗的平衡，适合小型嵌入式设备部署与应用。

参 考 文 献

[1] 蒲玲玲，杨柳. 改进 YOLOv5 的多车辆目标实时检测及跟踪算法[J]. 科学技术与工程，2023，23(28)：12159-12167.
PU L L，YANG L. Improving YOLOv5′s multi vehicle target real time detection and tracking algorithm [J]. Science and Technology and Engineering，2023，23 (28)：12159-12167.

[2] 张利丰，田莹. 改进 YOLOv8 的多尺度轻量型车辆目标检测算法[J]. 计算机工程与应用，2024，60(3)：129-137.
ZHANG L F，TIAN Y. Improved multi-scale lightweight vehicle target detection algorithm for YOLOv8 [J]. Computer Engineering and Applications，2024，60 (3)：129-137.

[3] 张三川，叶建明，师艳娟. 基于毫米波雷达的汽车前防撞预警系统设计[J]. 郑州大学学报(工学版)，2020，41(6)：13-18.
ZHANG S CH，YE J M，SHI Y J. Design of a car front collision warning system based on millimeter wave radar [J]. Journal of Zhengzhou University (Engineering Edition)，2020，41 (6)：13-18.

[4] 张壮壮. 基于 FPGA 和 CNN 的车辆目标检测系统设计[D]. 南京：南京信息工程大学，2023.
ZHANG ZH ZH. Design of vehicle target detection system based on FPGA and CNN [D]. Nanjing：Nanjing University of Information Technology，2023.

[5] Aetina Corporation. Aetina collaborates with innodisk and NVIDIA to drive AI to the industrial edge[J]. M2 Presswire，2023，6(15)：569-573.

[6] TANG Z，LU J J，CHEN Z Y. Improved Pest-YOLO：Real-time pest detection based on efficient channel attention mechanism and transformer encoder [J]. Ecological Informatics，2023，78(17)：854-862.

[7] YASHAR A，ALIREZA K，HOSSEIN A. Hierarchical approach for pulmonary-nodule identification from CT images using YOLO model and a 3D neural network classifier. [J]. Radiological Physics and Technology，2023，12(1)：1229-1231，.

[8] 江兴旺，赵兴强. 改进 YOLOv7 的木材表面缺陷检测算法[J]. 计算机工程与应用，2023，56(15)：124-131.
JIANG X W，ZHAO X Q. Improved Wood surface defect detection algorithm for YOLOv7 [J]. Computer Engineering and Applications，2023，56(15)：124-131.

[9] 蒋博，万毅，谢显中. 改进 YOLOv5s 的轻量化钢材表面缺陷检测模型[J]. 计算机科学，2023，50(S2)：271-277.
JIANG B，WAN Y，XIE X ZH. Improvement of YOLOv5s lightweight steel surface defect detection model [J]. Computer Science，2023，50 (S2)：271-277.

[10] WANG B，ASAD R S. Solution for sports image classification using modified MobileNetV3 optimized by modified battle royal optimization algorithm [J]. Heliyon，2023，9(11)：236-240.

[11] HUANG C Y，LEI Z Y，LI L H. A method for detecting key points of transferring barrel valve by integrating keypoint R-CNN and MobileNetV3 [J]. Electronics，2023，12(20)：332-339.

[12] RAJENDRA P S B，SAI C B. Mobilenetv3：A deep learning technique for human face expressions identification[J]. International Journal of Information Technology，2023，15(6)：3229-3243.

[13] PAN K L，HU H Y，GU P. A more accurate YOLO for defect detection in weld X-ray images[J]. Sensors，2023，23(21)：263-272.

[14] ANDLER M，HOWARD A，ZHU M. MobileNetV2：Inverted residuals and linear bottlenecks [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，2018：4510-4520.

[15] MA B，LI K，XU J. Enhancing the security of image

steganography via multiple adversarial networks and channel attention modules [J]. Digital Signal Processing,2023,141(18):643-651.

[16] KLOMP S R,WIJNHOVEN R G J,DE WITH P H N. Performance-efficiency comparisons of channel attention modules for ResNets[J]. Neural Processing Letters,2023,55(5):6797-6813.

[17] WANG X L,KONG L K,ZHANG Z G. Keypoint regression strategy and angle loss based YOLO for object detection[J]. Scientific Reports,2023,13(1):20117-20117.

[18] ZHAO W Q,XU M F,CHENG X F,et al. An insulator in transmission lines recognition and fault detection model based on improved faster R-CNN[J]. IEEE Transactions on Instrumentation and Measurement,2021,70:1-8.

[19] DU X Q,CHENG H C,MA Z H. A detection method for ground-planted strawberry fruits under different occlusion levels[J]. Computers and Electronics in Agriculture,2023,214(17):236-248.

[20] 吴亚尉,明帮铭,何剑锋,等. 基于YOLO-GR算法的轻量化钢材表面缺陷检测[J]. 组合机床与自动化加工技术,2023,42(11):107-111,115.
WU Y W,MING B M,HE J F,et al. Surface defect detection of lightweight steel based on YOLO-GR algorithm [J]. Combination Machine Tool and Automation Processing Technology,2023,42(11):107-111,115.

[21] ZHANG X,ZHOU X,LIN M. Shuffenet: An extremely efficient convolutional neural network for mobile devices [C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,2018:6848-6856.

[22] 周耀威,孔令军,李慧刚. 基于通道注意力机制与金字塔池化的包裹破损检测算法[J]. 无线电工程,2023,53(11):2626-2634.
ZHOU Y W,KONG L J LI H G. Package damage detection algorithm based on channel attention mechanism and pyramid pooling [J]. Radio Engineering,2023,53(11):2626-2634.

[23] 李慧琳,柴志雷. 基于Vitis AI的可行驶区域检测定制计算系统设计[J]. 现代信息科技,2022,6(1):73-78.
LI H L,CHAI ZH L. Design of a customized computing system for travelable area detection based on Vitis AI [J]. Modern Information Technology,2022,6(1):73-78.

[24] 胡凯,刘彤,武亚恒,等. 基于Vitis-AI架构的语义分割ENET模型实现[J]. 电子与封装,2022,22(3):77-81.
HU K,LIU T,WU Y H,et al. Implementation of semantic segmentation ENET model based on Vitis AI architecture [J]. Electronics and Packaging,2022,22(3):77-81.

作者简介

王伟(通信作者),博士,高级工程师,硕士生导师,主要研究方向为半导体光电子技术。
E-mail:wangwei_gd@cwxu.edu.cn

王坤,硕士研究生,主要研究方向为嵌入式软硬件设计,图像深度学习。

许圳兴,硕士研究生,主要研究方向为医学图像处理、图像深度学习。

付相为,硕士研究生,主要研究方向为嵌入式软件设计,图像处理,深度学习。