

LIU Bing, ZHOU Hai, BIAN Chunjiang, CHENG Xiaolei, WANG Pengfei, ZHANG Biao. Lightweight Yolov5 Algorithm Target Detection System Based on Space-grade NPU (in Chinese). *Chinese Journal of Space Science*, 2025, 45(4): 1123–1133. CSTR:32142.14.cjss.2024-0103. DOI:10.11728/cjss2025.04.2024-0103

# 基于宇航级 NPU 的轻量化 Yolov5 算法的目标检测系统\*

刘冰<sup>1,2</sup> 周海<sup>1</sup> 卞春江<sup>1</sup>  
成晓蕾<sup>1,2</sup> 王鹏飞<sup>1</sup> 张彪<sup>1</sup>

1(中国科学院国家空间科学中心 北京 100190)

2(中国科学院大学 北京 100049)

**摘要** 由于空间探测任务中需处理的遥感图像数量增多, 对目标检测系统的鲁棒性和时效性要求越来越高, 将大量遥感数据传输给地面再进行处理已经无法满足空间探测任务的需求. 针对此问题, 本文开展了基于宇航级 NPU 的遥感图像在轨目标检测系统研究, 以 Yolov5s 网络为基础, 替换与 NPU 适配度低的部分并引入注意力机制提升精度, 解决深度学习算法模型复杂和计算量大导致其在星上难以部署的问题. 优化后的网络迭代训练后部署在开发板上, 经 CPU - NPU 并行协同处理, 并行执行图像处理三部分, 充分利用 Yulong810A 平台资源. 实验显示, 优化后的网络部署到星上平台后参数量减少 75%, mAP 值达 71.25%, 检测速度为 47.67 frame·s<sup>-1</sup>, 均超过原版 Yolov5s 网络, 成功构建了一个更轻量快速的目标检测系统.

**关键词** 空间探测, 遥感图像, 星上处理平台, 深度学习, 并行处理设计

中图分类号 V557

## Lightweight Yolov5 Algorithm Target Detection System Based on Space-grade NPU

LIU Bing<sup>1,2</sup> ZHOU Hai<sup>1</sup> BIAN Chunjiang<sup>1</sup>  
CHENG Xiaolei<sup>1,2</sup> WANG Pengfei<sup>1</sup> ZHANG Biao<sup>1</sup>

1(National Space Science Center, Chinese Academy of Sciences, Beijing 100190)

2(University of Chinese Academy of Sciences, Beijing 100049)

**Abstract** With the continuous progress and expansion of space exploration missions, the quantity of remote sensing images that need to be processed has been increased substantially. In such a context, the target detection systems are confronted with ever-higher demands in terms of robustness and timeliness.

\* 中国科学院青年创新促进会项目资助 (E0293401)

2024-08-20 收到原稿, 2024-12-25 收到修定稿

©The Author(s) 2025. This is an open access article under the CC-BY 4.0 License  
(<https://creativecommons.org/licenses/by/4.0/>)

The traditional approach of transmitting a large volume of remote sensing data back to the ground for processing has become infeasible due to various limitations such as communication bandwidth and time delay. To address this critical issue, this research focuses on the on-orbit target detection system of remote sensing images, which is based on the astronautics-grade Neural Network Processor (NPU). Specifically, the YOLOv5s network is taken as the foundation and optimized. The components with relatively low compatibility with the NPU are replaced, and an attention mechanism is incorporated. This not only overcomes the challenges that the complex network structure and excessive computational requirements of deep learning-based object detection algorithms pose for deployment on satellite processing platforms with limited resources, but also compensates for potential losses in network accuracy. The optimized network is trained iteratively on the GPU using the public dataset VOC. After the CPU-NPU parallel co-processing design, the three main aspects of image processing, namely image preprocessing, feature extraction, and target classification and localization, are executed in parallel. This approach maximizes the utilization of the limited computing and storage resources of the Yulong810A platform. Experimental results demonstrate that when the optimized network is deployed on the Yulong810A on-board processing platform, it achieves remarkable improvements. The number of parameters is significantly reduced by 75%, and compared with the original YOLOv5s network, the accuracy is enhanced. The mean Average Precision (mAP) value reaches 71.25%, and the target detection speed attains 47.67 frames per second (fps), which is more than twice the speed of the original YOLOv5s network. In summary, this research realizes a more lightweight and faster object detection system, which holds great potential for promoting the development and efficiency of space exploration missions.

**Key words** Space exploration, Remote sensing images, On-board processing platform, Deep learning, Parallel processing design

## 0 引言

随着航空航天技术与遥感工程研究的飞速发展以及对地探测、深空探测等任务的不断深入,空间科学和遥感技术已经成为推动人类探索宇宙的关键力量,同时也对图像处理技术提出了更高要求<sup>[1,2]</sup>.能够于短时间内准确检测、识别大量遥感图像中各类目标的位置和种类信息变得尤为重要.传统的基于深度学习的遥感图像检测方法是采集后的图像传回地面再进行识别,需要传递的遥感图像数据量过大导致目标检测系统的处理效率低下,无法在短时间内完成识别任务<sup>[3]</sup>.若在轨对遥感图像进行检测处理后再将结果返回,可以大大减少星地传输的数据量,加快数据传输的速率.目前星上在轨的一些目标检测算法为了适应计算和存储资源有限的星上环境,需对神经网络进行简化后再部署到嵌入式平台中,这会导致检测精度下降,检测速度并未有相应的大幅提升<sup>[4-6]</sup>.因此,发展高效精准的星上在轨图像检测技术对于执行空间探测任务有重要意义.

目前基于深度学习的目标检测算法主要分为两大类:一类是以 R-CNN<sup>[7,8]</sup> 和 R-FCN<sup>[9]</sup> 等为代表的两阶段目标检测;另一类是以 YOLO<sup>[10-12]</sup> 和 SSD<sup>[13]</sup> 等为代表的单阶段目标检测方法.其中 YOLO 系列算法较为轻便,适合部署于空间有限的星上平台完成空间探测任务.该算法改进了检测流程,网络直接输出边界框的位置和所属类别显著提升了检测速度.其中 YOLOv5s<sup>[14,15]</sup> 网络在 YOLO 系列中具有模型精度较高、处理速度快且权重文件小的特点.该网络经轻量化处理后适合部署在对资源和处理速度都有较高要求的星上平台上.该模型已广泛应用于中央处理器 (CPU)、图形处理器 (GPU)、神经处理单元 (NPU)、张量处理器 (TPU) 等不同嵌入式平台来完成图像处理任务. Chen 和 Lin<sup>[16]</sup> 在集成了 CPU 和 NPU 的嵌入式平台部署了优化后的 YOLOv3 网络,最终处理图像的平均精度值 (mean Average Precision, mAP) 达到 69.79%,检测速度为 28 frame·s<sup>-1</sup>. Boltov 等<sup>[17]</sup> 在集成了 GPU 和高级精简指令集机器 (Advanced RISC Machines, ARM) 的 Jetson Nano 嵌入式平台

上部署了 YOLOv5, 虽然精度较高但是检测速度只有  $7 \text{ frame}\cdot\text{s}^{-1}$ . Amin 等<sup>[18]</sup> 在 FPGA 平台上实现了高效的实时目标检测系统, 高准确率的同时检测速度达  $15 \text{ frame}\cdot\text{s}^{-1}$ . Kaymakci 等<sup>[19]</sup> 在集成了 GPU 和 TPU 的 Jetson Orin 平台上部署了 YOLO 系列网络, 其中部署后的 YOLOv5 网络准确率较低, 但是处理速度最快, 为  $35.2 \text{ frame}\cdot\text{s}^{-1}$ , 高于 YOLOv7 和 YOLOv8 的  $24.8 \text{ frame}\cdot\text{s}^{-1}$  和  $22.6 \text{ frame}\cdot\text{s}^{-1}$ . Xue 等<sup>[20]</sup> 在 GPU 上基于 YOLOv7-tiny 网络进行优化, 提高了无人机端针对小目标图像的检测识别能力. 由于在地面上的深度学习算法运行有 GPU 等处理器加持, 但其并不满足航天和空间探测任务的可靠性要求, 而在轨处理器往往抗辐射性能良好但计算处理性能较弱, 需要对目标检测算法的识别性能和鲁棒性进行针对性优化以适应苛刻的在轨环境. 因此, 关于基于星载平台的遥感图像在轨处理技术的探究一直备受关注. Wagstaff 等<sup>[21]</sup> 在 EO-1 卫星上使用基于机器学习的云过滤和异常检测技术, 这表明了深度学习在空间图像处理中的潜力. 其研究展示了利用机载机器学习技术提高空间图像处理的效率和准确性. Xu 等<sup>[22]</sup> 探讨了在 Hisea-1 卫星上对捕获的遥感图像基于恒虚警率并结合轻量级 YOLOv4 网络进行实时舰船检测, 展示了深度学习算法在处理遥感图像的潜力, 这对于空间探测和海洋监测等领域具有重要意义. 本文致力于利用深度学习技术, 特别是针对改进目标检测算法提高网络与硬件平台的适配程度, 最终提高星上图像处理任务的精度和效率.

在资源受限的星上环境部署网络模型时, 原 YOLOv5s 网络的参数冗余, 不仅增加了模型的大小, 还导致处理速度的降低. 此外, 原版网络在 NPU 上的硬件支持度不足也影响了最终图像处理的精度, 无法展现其原本的优势. 为解决这些问题, 本文提出一种轻量化的 YOLOv5-Ghost+CBAM 网络模型, 用 Ghost Bottleneck 模块替换了部分复杂的 BottleneckCSP 模块, 有效减少了模型的计算量和体积. 同时, 为减小其带来的精度损失, 在网络中引入卷积注意力模块 (Convolutional Block Attention Module, CBAM), 在较低开销的前提下, 显著增强了网络对关键特征的捕捉能力, 从而提升了整体的检测精度. 在下一步将网络部署到 NPU 的过程中, 本文采用 CPU 与 NPU 的协同并行处理策略, 充分利用星上平

台的计算资源, 进一步提升了图像处理速度. 这种创新的部署方式不仅满足了星上实时处理图像任务的严苛要求, 而且确保了在有效资源中进行图像处理的高效性和准确性.

## 1 YOLOv5 网络的轻量化设计

YOLOv5s 的网络结构如图 1 所示, 主要分为 4 大块, 由输入端、Backbone、Neck 和 Prediction 输出端组成<sup>[23]</sup>. Backbone 模块的主要功能是对输入的图像提取一些通用特征, 由 Focus 结构和 BottleneckCSP 等组成. Neck 模块在特征金字塔网络的基础上增加了自下而上的金字塔结构, 提高网络对不同尺度的目标检测能力, 最后输出分类结果和对象坐标.

针对内存、计算资源有限的星上图像处理平台, 直接使用原版 YOLOv5s 网络时, 网络模型和计算量过大且消耗的资源过多, 处理大量图像数据时表现较差. 针对此问题, 本文提出两点优化: 首先使用 Ghost Bottleneck<sup>[24]</sup> 轻量化模块替换主干网络中的 3 个 BottleneckCSP 模块, 该模块首先用逐点卷积代替了常规卷积, 并结合深度卷积模块, 大幅减少了计算量并加快网络的处理速度; 其次, 为能进一步提高准确率, 在改进后的主干网络中的 Ghost Bottleneck 模块后引入 CBAM<sup>[25]</sup>, 该模块是轻量化模块, 其开销较小, 可以忽略. 数据进入 CBAM 中, 依次经过通道注意力模块和空间注意力模块, 二者串联可以通过学习要强调或者抑制相应的信息来有效提高图像处理的准确率, 同时对网络模型大小的影响较小, 优化后的网络模型有效减小了模型的参数和计算量, 提高了识别的准确率, 经过量化后可以很好地满足星上平台的要求.

Ghost 模块的核心思想是为了消除冗余特征, 获得更轻的模型, 并弥补 YOLOv5 细节特征不完整的缺点. 图 2 所示为 Ghost 模块结构, 可以看出该模块分为 3 个步骤: 卷积、Ghost 生成和特征图拼接. 首先使用常规卷积得到特征映射, 然后对每个通道的固有特征图进行  $\Phi$  操作处理, 生成 Ghost 特征图, 其中  $\Phi$  操作是  $3\times 3$  和  $5\times 5$  等大小的内核卷积, 只需一个通道就能完成卷积运算, 有效降低了计算成本. 最后将第一步得到的特征图和第二步生成的 Ghost 特征图连接起来, 得到最终的输出.

本文选择的 NPU 平台可以十分高效地实现卷积

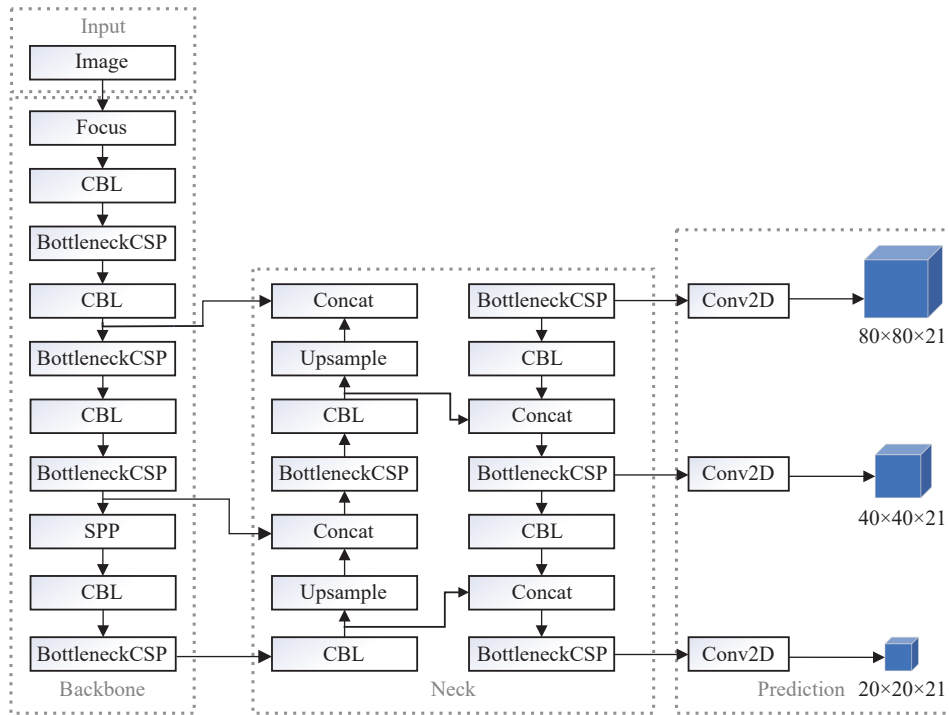


图 1 YOLOv5s 的网络结构

Fig. 1 Structure of YOLOv5s network

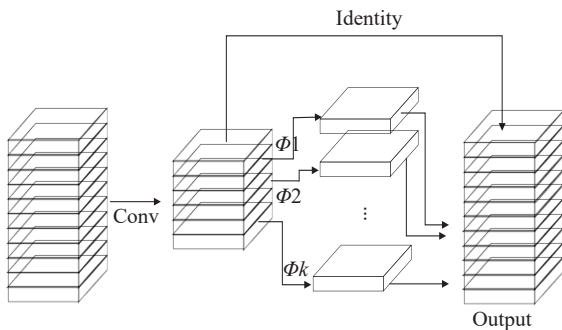


图 2 Ghost 模块结构

Fig. 2 Structure of the Ghost module

运算, 为进一步提高效率, 用点卷积代替 Ghost 模块中所有常规卷积. 因此选择深度卷积实现  $\Phi$  操作, 并且由于统一计算形式有利于在 NPU 上部署网络, 将每层用  $\Phi$  的大小设置统一以构建高效的神经网络. 普通卷积操作的计算量和参数量与 Ghost 模块的计算量和参数量之比如下:

$$r_s = \frac{nh_1w_1ckk}{\frac{n}{s}h_1w_1ckk + (s-1)\frac{n}{s}h_1w_1dd} = \frac{sc}{sc-1}. \quad (1)$$

其中,  $c$ ,  $n$ ,  $h_1$ ,  $w_1$  分别为输入的通道数、输出的通道数、输出特征图的高和宽,  $k$  为常规卷积核大小,  $d$  为

线性变换卷积核大小,  $s$  为变换次数. 很明显, 通过训练时寻找合适的  $s$  值, 可以使得 Ghost 模块的计算量和参数大小相比常规卷积模块大幅降低.

本文所用 Ghost Bottleneck 模块的内部结构由两个 Ghost 模块堆栈组成的. 如图 3 所示, 该模块根据步幅大小分为两种. 具体而言, 左侧 Ghost 模块作为扩展层, 其作用是增加通道数量, 以匹配快捷路径, 并添加批归一化 (Batch Normalization, BN) 和 ReLU 激活函数提升特征的表达能力. 右侧 Ghost 模块则在每层后省略了 ReLU 激活函数, 避免在深度卷积后的信息丢失. 该设计有利于深化图像特征和捕获, 提高特征提取的深度和精度. 该模块有利于对图像特征的细化和捕获. BottleneckCSP 模块是传统 YOLOv5s 中较为为基础的特征提取模块, 但其设计的局限性导致该模块在提取深度特征时效率较低且处理速度较慢. 因此这里利用 Ghost Bottleneck 模块替换掉繁琐的 BottleneckCSP 模块, 该模块可在同等卷积输入时减少模型的参数和计算量, 舍弃了传统的卷积层, 提高了权值的检测速度. 该模块在对精度影响较低的前提下大幅降低了网络的模型大小, 使其更适合在资源受限的星上平台部署.

图 4 所示为替换掉 YOLOv5s 主干网络中 BottleneckCSP 模块后形成的新的主干网络模型. YOLOv5s 网络中大部分的推理计算都是在 Backbone 中实现的, 替换 Neck 中的结构对整体模型的时长优化不明显且容易导致准确率降低, 基于此只代替了主干网络中 3 个 BottleneckCSP 模块, 对 Neck 中的 BottleneckCSP 结构不做替换.

这里选用的是 CBAM 轻量级注意力模块, 内部分为通道注意力模块 (Channel Attention Module) 和空间注意力模块 (Spatial Attention Module), 其模型结构简单且消耗的资源较少. 该模块具体结构如图 5 所示. 在通道注意力模块中, 输入特征图经过平均池化层和最大池化层得到两个特征, 再分别经过共享

多层感知机后生成两组特征映射, 经过 concat 操作叠加成一个特征图, 送入 ReLU 激活函数模块后得到输出. 将输入和输出相乘后送入空间注意力模块. 在空间注意力模块中, 输入量依次通过最大池化层、平均池化层和激活函数模块后得到输出. 将空间注意力模块的输入量与输出量相乘得到 CBAM 整体模块的输出.

为确保网络部署到嵌入式平台后满足精度的要求, 在改进后的 YOLOv5-Ghost 网络中引入注意力机制. 将 CBAM 模块加入 YOLOv5-Ghost 网络后得到 YOLOv5-CBAM+Ghost 网络模型. 最终网络模型如图 6 所示, 为了控制网络结构的大小, 将 3 个 CBAM 模块插入主干网络中构成最终的网络模型. 该网络可以在保持较小开销的前提下, 提高对正确特征的提取, 抑制对无关特征的提取, 从而增强网络的准确性.

实际部署到 NPU 端时, 由于 Focus 中的特征图拼接操作在嵌入式芯片中的实现过程过于繁琐, 需要先将数据搬移到内存中由 CPU 完成计算, 再将结果搬移回 NPU 进行后续计算, 因此在硬件上部署时使用 Focus 操作进行下采样反而会带来较高的性能损耗, 于是将该模块用简单的 Conv 卷积模块代替. 该卷积运算的卷积核大小为 6, 步长、padding 为 2. 输入尺寸为 (3, 640, 640) 的特征图后, 经步长为 2 的卷积进行下采样操作得到 (32, 32, 320) 的输出特征图. 替换后对模型精度影响很小并且可以有效提升硬件上的处理效率.

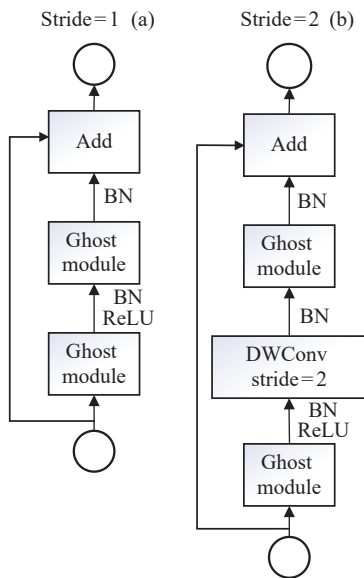


图 3 Ghost bottleneck 模块的结构模型

Fig. 3 Structure of Ghost bottleneck module

## 2 NPU 的并行协同处理加速设计

本文选择的硬件平台是搭载了 Arm CPU+NPU

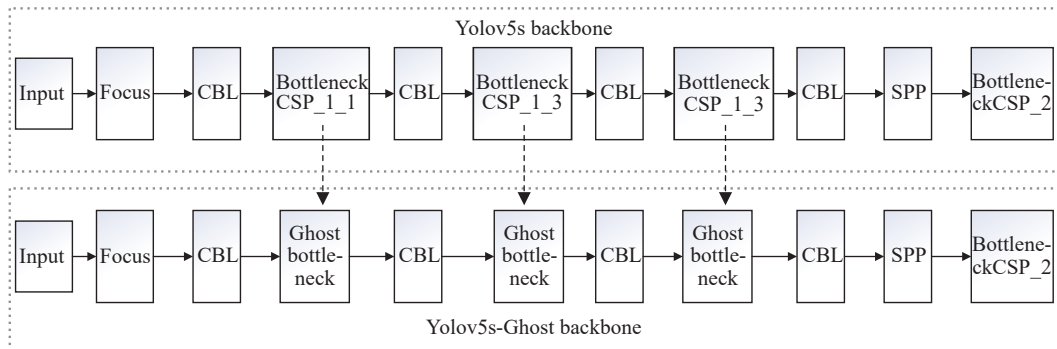


图 4 改进后的 YOLOv5s-Ghost 的主干网络模型

Fig. 4 Structure of the improved YOLOv5s-Ghost backbone

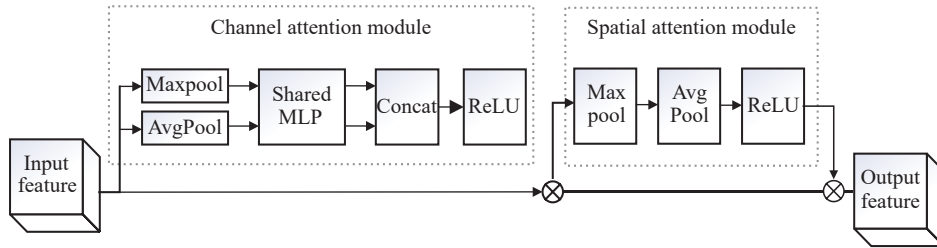


图5 CBAM 模块结构

Fig. 5 Structure of the CBAM module

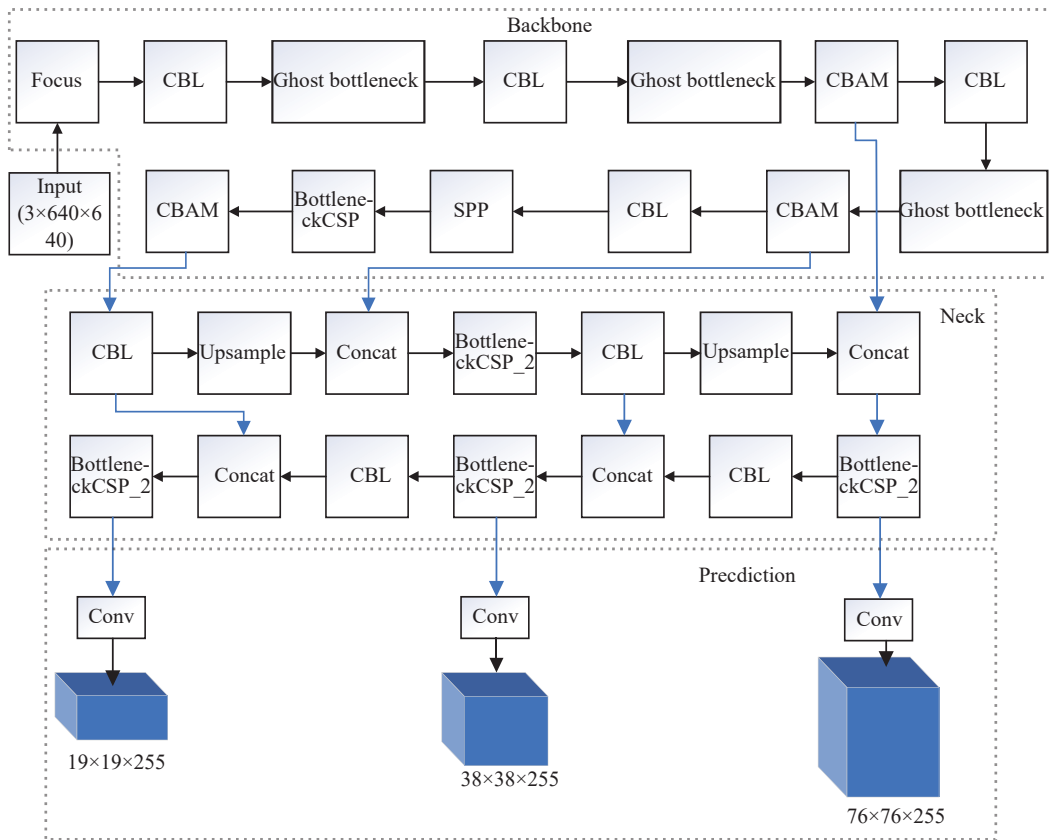


图6 Yolov5-CBAM+Ghost 网络模型的结构

Fig. 6 Structure of Yolov5-CBAM+Ghost network module

的 Yulong810 A 板卡. 该板卡所用 NPU 无法直接识别 Pytorch 框架训练出的模型, 需要将模型改为 ONNX 格式. 在转换过程中会变成统一的中间态 IR (Intermediate Representation, 中间表示) Graph. 为提高计算效率和硬件适配度, 需要对中间态 IR 形式的模型进行优化. 因此实施了包括图融合和统一缓冲区融合 (Unified Buffer, UB) 之类的算子融合. 以网络中常用的 Conv2D 算子和 Batch Norm 算子为例, 其计算公式如下:

$$Y = WX + B,$$

$$y_i = \gamma \frac{x_i - \mu}{\sqrt{\sigma_B^2 + \varepsilon}} + \beta. \quad (2)$$

将式 (2) 中的两个分式代入融合后可得

$$Y' = \left( \frac{\gamma W}{\sqrt{\sigma_B^2 + \varepsilon}} \right) x_i + \left[ \beta - \frac{\gamma(B - \mu)}{\sqrt{\sigma_B^2 + \varepsilon}} \right] = W'X + B',$$

$$W' = \left( \frac{\gamma W}{\sqrt{\sigma_B^2 + \varepsilon}} \right),$$

$$B' = \left[ \beta - \frac{\gamma(B - \mu)}{\sqrt{\sigma_B^2 + \varepsilon}} \right]. \quad (3)$$

原卷积核的权重和偏置分别是  $W$  和  $B$ , 化简后得到新的  $W'$  和  $B'$ , 两个算子融合形成一个新算子  $Y'$ . 二者可以进行融合的原因是在 NPU 上进行计算处理时其网络参数已经确定, 不用考虑后续权值更新, 可以直接输入计算参数得到结果, 如图 7 所示, 通过该融合减少了批归一化层参数的计算和存储, 提升计算效率.

UB 融合是为了解决 NPU 处理数据时算子间的数据重复搬移问题. 一般情况下, NPU 中一个算子得到的结果会存入缓冲区中, 然后再搬入内存中用于下一步处理. 下一个算子计算时需要再将结果从内存搬运到缓冲区中. 此时数据流向为缓冲区—内存—缓冲区, 存在数据移动的冗余. 通过 UB 融合实现缓冲区的数据复用, 减少部分数据的搬移时间, 提高 NPU 进行推理的效率.

训练好的网络通过添加后处理和批处理, 实现了对指定地址下图片进行批量处理并输出结果的功能. 开发板接收图像后, 需要拷贝到生成的 Graph 通道中依次进行网络的前处理、网络推理、后处理. 如果完全按串行思路设计, 则处理一张图像需要的时间是所有处理时间的累加, 具体处理流程如图 8 所示. 显然该思路不利于充分利用 NPU 平台上的有限资源.

针对这一问题, 提出了一种 CPU 与 NPU 多线程协同计算的策略, 以加速图像处理流程并提升平台整体的处理效率, 减少图像处理过程中一些不必要的数据传输和推理时间. 当开发板接收到原始图像数据后, 需要将数据复制传递给创建好的 Graph. 其中, 传递一张  $1920 \text{ pixel} \times 1080 \text{ pixel}$  的图像数据耗时约为  $18 \text{ ms}$ , 时间较长. 因此, 这里将数据拷贝和预处理与网络推理等同步执行. 首先把预处理中的图像裁剪和格式转换等步骤都在 NPU 上运行, 省去预处理和图

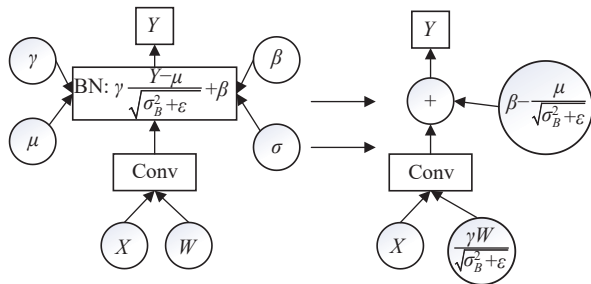


图 7 算子融合前后计算示例

Fig. 7 Schematic diagram of the calculation before and after operator fusion

像推理之间数据拷贝的时间. 通过添加环境变量并指定缩放后的图像尺寸等操作, 最终在 NPU 上集成了预处理和网络推理的功能. 为实现该并行设计, 创建了 2 个 Graph, 当 Graph2 接收第一张图像数据并开始进行网络的预处理时, Graph1 开始将新一轮的数据拷贝, 实现数据拷贝和网络计算的并行处理. 由于开发板搭载 4 个 CPU, 进一步创建了 Graph3, 将网络的后处理计算分配给额外的 CPU 核心, 同样当 Graph2 中执行到预处理部分时, Graph3 开始执行上一张图片的后处理任务. 最终流程如图 9 所示, 数据拷贝和网络后处理由 CPU 完成, 预处理和网络推理由 NPU 完成, 实现 CPU 和 NPU 的并行协同处理, 充分利用了平台的计算资源.

设计的最终处理流程如图 10 所示, CPU 负责数据的拷贝和网络的后处理, NPU 负责网络的预处理和网络推理, 省去了原本预处理和网络推理之间的数据拷贝的耗时, 虚化了输入数据的传输时间. 最终在该平台上实现了输入数据的拷贝、网络预处理和推

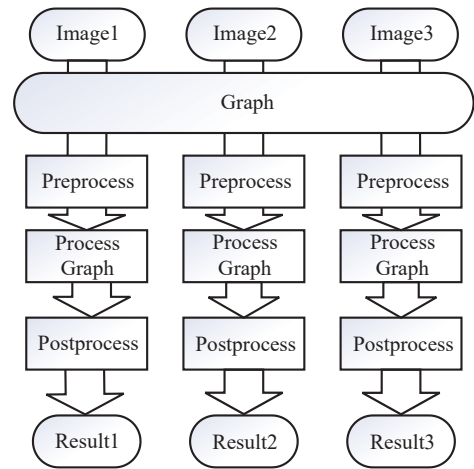


图 8 NPU 的图像处理流程

Fig. 8 Image processing flow diagram of the NPU

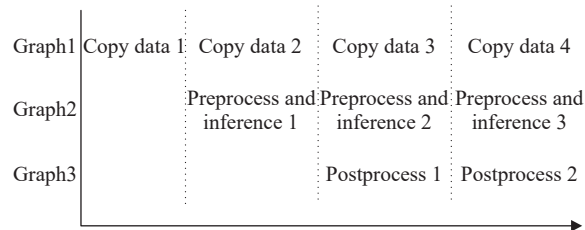


图 9 多线程并行图像处理流程

Fig. 9 Multi-threaded parallel image processing flowchart

理、网络后处理三者的并行处理设计,优化了星上图像处理平台的资源分配,有效减少了数据处理和计算中的不必要延迟,进一步缩短了处理周期,使得图像处理效率进一步提升.

### 3 结果与分析

实验选择 PASCAL VOC 数据集为基础并加入自建遥感图像数据来进行网络的训练和测试,图像大小大部分裁剪为统一的  $640 \times 640$ . 该数据集是目标检测常用的数据集,包含常见的 4 个大类和 20 个小类,数据集中目标样本丰富、背景特征不一,适用于不同环境下应用的深度学习目标检测算法,提供了丰富的视觉信息. 在数据集中加入 1174 张遥感图像,加入的遥感图像大小为  $12 \sim 70$  pixel,以  $20 \sim 50$  pixel 的中小尺寸目标为主,包含 7 个 aircraft, overpass, freeway 等不同类别,按 7:1:2 随机划分训练集、验证集、测试集,满足适用性的同时加强其在处理遥感图像方面的能力并针对性地调整网络权重参数. 为增强网络的泛化能力,采取数据增强技术,包括但不限于随机裁剪、翻转、颜色调整等. 该方法有助于提高网络的鲁棒性,筛选后去除部分标注不清或目标重叠的数据,最终训练过程中也相对稳定.

在传统的 PC 机上使用 Pytorch 框架进行模型的训练. 具体参数配置列于表 1. 对训练震荡幅度和精度影响较大的实验参数是训练周期 (Epoch) 和批次数 (Batch-size). 一个 Epoch 是通过网络对所有训练图像进行一次训练, Batch-size 代表每次迭代中训练图像的数量. 训练时为防止过拟合,设 Epoch 为 50, Batch-size 为 16. 设置其他不同的超参数并进行

比较,得到最优的训练结果.

评价模型精确度指标选取 mAP 值和 GIoU 损失函数. mAP 是由网络模型训练时得到的每一类检测目标的精确率 (Precision, 定义符号  $P$ ) 与召回率 (Recall, 定义符号  $R$ ) 积分求和后,除以检测目标总的类别数得到的,定义符号  $f_{\text{mAP}}$ .  $F_1$  是精确率和召回率的加权和平均值. 与  $F_1$  值相比, mAP 值能够反映对更多类目标的平均检测精度. 计算方法如下:

$$\begin{aligned} P &= \frac{N_{\text{TP}}}{N_{\text{TP}} + N_{\text{FP}}}, \\ R &= \frac{N_{\text{TP}}}{N_{\text{TP}} + N_{\text{FN}}}, \\ F_1 &= \frac{2N_{\text{TP}}}{2N_{\text{TP}} + N_{\text{FP}} + N_{\text{FN}}}, \\ f_{\text{mAP}} &= \sum_{n=1}^N \int_0^1 P_n dR_n. \end{aligned} \quad (4)$$

损失函数选取的是 GIoU (定义符号  $l_{\text{GIoU}}$ ) 而不是 IoU (定义符号  $l_{\text{IoU}}$ ), 与 IoU 相比, GIoU 综合考量了重叠区域和其他非重叠区域,可以更全面均衡地反映重合的程度,有

$$\begin{aligned} l_{\text{IoU}} &= \frac{|A \cap B|}{|A \cup B|}, \\ l_{\text{GIoU}} &= l_{\text{IoU}} - \frac{|C/(A \cap B)|}{|C|}. \end{aligned} \quad (5)$$

训练网络过程中得到的 GIoU 损失函数值和 mAP 值如图 11 所示. GIoU 值逐步减小. 在测试集上验证训练后的网络得到的各项统计值如表 2 所示. 将结果与改进前的模型进行比较,通过与改进前的模型进行对比,发现不仅在权重参数上实现了显著降低,而且在涵盖的 7 类目标检测任务测试集中,模型的 mAP 达到 78.47%, 与原 YOLOv5s 模型相比提升了约 2%.

非对称 8 bit 量化部署在 Yulong810 A 平台上,从测试集中随机选取图片进行图像处理测试,测试集

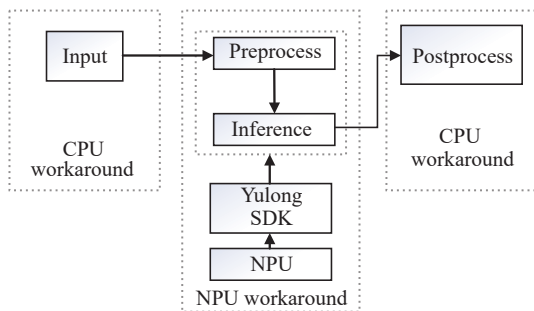


图 10 CPU-NPU 并行计算处理

Fig. 10 Image processing flow diagram of the CPU-NPU

表 1 网络训练用到的配置参数

Table 1 Configurations used for network training

| 配置   | 参数                                           |
|------|----------------------------------------------|
| CPU  | Intel Core i5-10400                          |
| GPU  | NVIDIA RTX 2060 SUPER                        |
| 内存   | 16 GByte                                     |
| 软件环境 | Python3.6, CUDA10.1, cuDNN7.6.5, OpenCV3.4.5 |

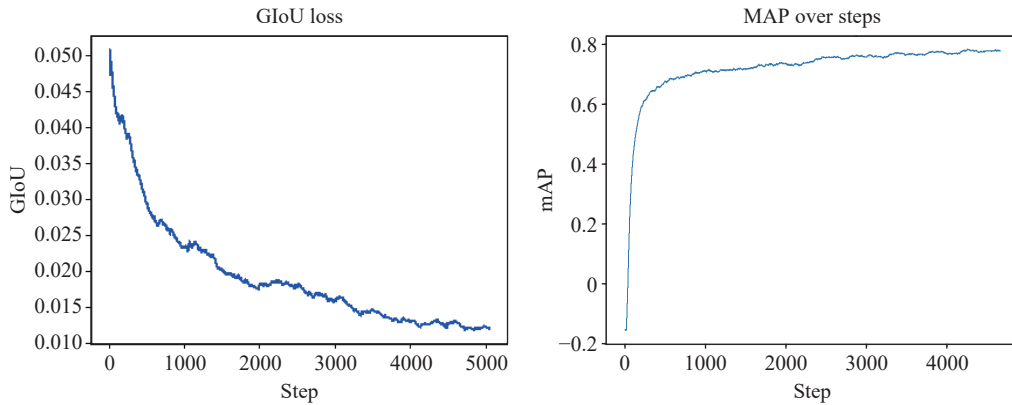


图 11 改进后网络的 GIoU 和 mAP 曲线变化  
Fig. 11 GIoU and mAP of the improved network

表 2 模型改进前后的性能结果

Table 2 Performance results before and after model improvements

| 模型        | 权重参数/MByte | mAP/(%) | 计算量/GFLOPs |
|-----------|------------|---------|------------|
| 原Yolov5s  | 15.2       | 76.29   | 16.0       |
| 本文Yolov5s | 6.71       | 78.47   | 10.5       |

图像大小为 640 pixel×640 pixel 和 1920 pixel×1080 pixel, 最终结果为所有结果的均值. 结果如表 3 所示, 其中采取单线程的方式运行原版 Yolov5s 网络的推理时间为 74.8 ms, 换算后推理速度为 13.2 frame·s<sup>-1</sup>, mAP 值为 69.97%, 而本文提出的改进后 Yolov5s 模型在 Yulong810 A 上的推理速度比原版提高了 2 倍以上. 量化后将浮点数转换为 8 位整数, 位数的减少使得数据精度的降低, 在进行激活函数和权重累加时会产生非线性误差, 导致量化后在 Yulong810 A 平台上进行目标检测的 mAP 值有所降低, 最终的 mAP 值为 71.25%, 虽然有一定的下降但也高于原版 Yolov5s 量化后的网络精度, 平台图像处理性能显著提升. 与其他一些嵌入式平台相比, 本文系统在 mAP 值和推理速度上均实现了不同程度的优化和提升. 其中 RK3399 pro 平台运行 Yolov5s 时由于其输入的图像数据为 416 pixel×416 pixel, 图像数据较小, 经统计验证后, 若输入为 640 pixel×640 pixel 其处理速度会减小到 40 frame·s<sup>-1</sup> 以下, 精度也低于本文方法. 文献 [25] 中采取 RK3399 pro 平台外接 NPU 运行优化后的 Yolov4 s 实现的模型与本文方法相比在参数量和推理速度方面有所欠缺.

## 4 结论

在当前空间科学和探测领域, 现有研究主要集中在遥感图像目标识别算法的开发上, 对于这些算法在遥感卫星平台上的部署应用技术缺乏系统研究. 考虑到深度学习算法在轨应用与地面应用在环境、算力等方面的差异性, 遥感卫星在轨运行的深度学习算法仍面临诸多挑战. 并且由于采集的遥感图像通常具有复杂的背景, 会干扰目标检测, 尤其是执行空间探测任务时, 获取的图像背景和目标颜色、形状相似, 因此其精度下降和识别难度增大. 针对上述问题, 本文从算法设计和硬件实现两个层面对遥感图像在轨目标识别技术进行了系统研究. 在 Yolov5s 算法基础上进行优化, 选择 Ghost Bottleneck 模块对 Yolov5s 主干网络进行替换, 并引入 CBAM 注意力机制加入主干网络中, 该模块可以使模型能够专注于目标区域, 减少背景干扰以提升检测准确率, 最终实现网络的轻量化处理, 并且在加入遥感图像数据集后的 PASCAL VOC 数据集上使得目标 mAP 达到 78.47%, 比原版 Yolov5s 提高约 2%. 在硬件实现方面, 在部署网络时进行并行处理设计, 首先将算法的前处理集成到 NPU 上省去了部分数据复制的时间, 随后将由 CPU 执行的后处理与 NPU 执行的网络推理并行处理, 充分利用平台有限的计算资源. 最终该算法整体处理速度从 13.2 frame·s<sup>-1</sup> 提升至 47.67 frame·s<sup>-1</sup>, 比运行原 Yolov5s 网络的速度提升 3 倍以上, 整体 mAP 由于速度的提升下降至 71.25%. 对比其他嵌入式平台, 本文设计的算法在识别速度与精度方面都有着显著优势, 通过对面向遥感载荷原始数据的目标识

表 3 不同模型在嵌入式平台的性能结果  
Table 3 Performance results of different models on embedded platforms

| 模型选择                      | 参数量/MByte | 处理平台的硬件选择      | 推理速度/(frame·s <sup>-1</sup> ) | mAP/(%) |
|---------------------------|-----------|----------------|-------------------------------|---------|
| 原Yolov5s <sup>[26]</sup>  | 3.8       | Yulong810 A    | 13.2                          | 69.97   |
| 优化Yolov3 <sup>[16]</sup>  | —         | CPU-NPU        | 28                            | 69.79   |
| 优化Yolov5s <sup>[27]</sup> | 2.2       | RK3399 pro     | 50                            | 66.23   |
| 优化Yolov4s <sup>[28]</sup> | 11.09     | RK3399 pro+NPU | 35.37                         | 84.79   |
| 本文Yolov5s                 | 2.9       | Yulong810 A    | 47.67                         | 71.25   |

别算法的优化以及星上平台的资源高效利用设计, 为空间科学和探测领域提供了更为高效、精准的处理思路和方法探索. 不仅为空间科学和探测领域提供了技术支持, 也对遥感卫星在轨应用做了进一步探索.

参考文献

[1] YU Shuangming, WU Wenbo, DOU Runjiang, *et al.* Design and implementation of remote sensing image processing system with intelligent vision processor[J]. *Spacecraft Recovery & Remote Sensing*, 2023, **44**(5): 54-64 (于双铭, 武文波, 窦润江, 等. 基于视觉处理器芯片的遥感图像智能处理系统设计与验证 [J]. *航天返回与遥感*, 2023, **44**(5): 54-64)

[2] DU Xin, ZHONG Ruofei, LI Qingyang, *et al.* Design and implementation of ground simulation system for remote sensing image on-board intelligent processing[J]. *Remote Sensing Technology and Application*, 2022, **37**(5): 1198-1208 (杜鑫, 钟若飞, 李清扬, 等. 遥感图像星上智能处理地面仿真模拟系统设计与实现 [J]. *遥感技术与应用*, 2022, **37**(5): 1198-1208)

[3] XU Siyuan, WU Weilin. Research on target detection algorithm based on multi-scale feature fusion in remote sensing images[J/OL]. *Computer Engineering and Applications*, 1-10[2024-07-24] (许思源, 吴伟林. 多尺度特征融合的遥感图像目标检测算法研究 [J/OL]. *计算机工程与应用*, 1-10[2024-07-24]. <http://kns.cnki.net/kcms/detail/11.2127.TP.20240625.1708.023.html>.)

[4] BHAGYA C, SHYNA A. An overview of deep learning based object detection techniques[C]//2019 1st International Conference on Innovations in Information and Communication Technology (ICHICT). Chennai: IEEE, 2019: 1-6. DOI: [10.1109/ICHICT1.2019.8741359](https://doi.org/10.1109/ICHICT1.2019.8741359)

[5] M R, L A R. Comprehensive investigation on real-time object detection in deep learning[C]//2023 IEEE Fifth International Conference on Advances in Electronics, Computers and Communications (ICAIECC). Bengaluru: IEEE, 2023: 1-5. DOI: [10.1109/ICAIECC59324.2023.10560332](https://doi.org/10.1109/ICAIECC59324.2023.10560332)

[6] KUMAR A S S, MAHESH K. An investigation of deep neural network based techniques for object detection and recognition task in computer vision[C]//2023 2nd International Conference on Edge Computing and Applications (ICECAA). Namakkal: IEEE, 2023: 385-390. DOI: [10.1109/ICECAA58104.2023.10212307](https://doi.org/10.1109/ICECAA58104.2023.10212307)

[7] KUMAR D, Zhang X L. Improving more instance segmentation and better object detection in remote sensing imagery based on cascade mask R-CNN[C]//2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS. Brussels: IEEE, 2021: 4672-4675. DOI: [10.1109/IGARSS47720.2021.9554512](https://doi.org/10.1109/IGARSS47720.2021.9554512)

[8] LE H, NGUYEN T, LE V, *et al.* Guided anchoring cascade R-CNN: an intensive improvement of R-CNN in Vietnamese document detection[C]//2021 8th NAFOSTED Conference on Information and Computer Science (NICS). Hanoi: IEEE, 2021: 188-193. DOI: [10.1109/NICS54270.2021.9701510](https://doi.org/10.1109/NICS54270.2021.9701510)

[9] ZHANG J L, CHEN S X, HOU Y W. Accurate Object detection with relation module on improved R-FCN[C]//2020 Chinese Automation Congress (CAC). Shanghai: IEEE, 2020: 7131-7135. DOI: [10.1109/CAC51589.2020.9326543](https://doi.org/10.1109/CAC51589.2020.9326543)

[10] PULIPALUPULA M, PATLOLA S, NAYAKI M, *et al.* Object detection using You Only Look Once (YOLO) algorithm in Convolution Neural Network (CNN)[C]//2023 IEEE 8th International Conference for Convergence in Technology (I2CT). Lonavla: IEEE, 2023: 1-4. DOI: [10.1109/I2CT57861.2023.10126213](https://doi.org/10.1109/I2CT57861.2023.10126213)

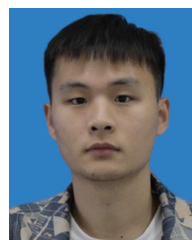
[11] LU Y H, ZHANG L W, XIE W. YOLO-compact: an efficient YOLO network for single category real-time object detection[C]//2020 Chinese Control and Decision Conference (CCDC). Hefei: IEEE, 2020: 1931-1936. DOI: [10.1109/CCDC49329.2020.9164580](https://doi.org/10.1109/CCDC49329.2020.9164580)

[12] SHAO Yanhua, ZHANG Duo, CHU Hongyu, *et al.* A review of YOLO object detection based on deep learning[J]. *Journal of Electronics & Information Technology*, 2022, **44**(10): 3697-3708 (邵延华, 张铎, 楚红雨, 等. 基于深度学习的YOLO目标检测综述 [J]. *电子与信息学报*, 2022, **44**(10): 3697-3708)

[13] HU Yan, YUAN Zihao, TU Xiaoguang, *et al.* Improved SSD object detection algorithm based on contrastive learning[J]. *Infrared Technology*, 2024, **46**(5): 548-555 (胡焱, 原子昊, 涂晓光, 等. 基于对比学习的改进SSD目标检测算法 [J]. *红外技术*, 2024, **46**(5): 548-555)

- [14] ZHU D R, DAI L F, DU P R. CCE-YOLOv5s: an improved YOLOv5 model for UAV small target detection [C]//2023 IEEE 5th International Conference on Civil Aviation Safety and Information Technology (ICCASIT). Dali: IEEE, 2023: 824-829. DOI: [10.1109/ICCASIT58768.2023.10351744](https://doi.org/10.1109/ICCASIT58768.2023.10351744)
- [15] BISWAS S, NANDY A, NASKAR A K, *et al.* Real time gesture recognition using improved YOLOv5 model[C]//2024 11th International Conference on Signal Processing and Integrated Networks (SPIN). Noida: IEEE, 2024: 328-333. DOI: [10.1109/SPIN60856.2024.10511787](https://doi.org/10.1109/SPIN60856.2024.10511787)
- [16] CHEN S B, LIN W. Embedded system real-time vehicle detection based on improved YOLO network[C]//2019 IEEE 3rd Advanced Information Management, Communications, Electronic and Automation Control Conference (IMCEC). Chongqing: IEEE, 2019: 1400-1403. DOI: [10.1109/IMCEC46724.2019.8984055](https://doi.org/10.1109/IMCEC46724.2019.8984055)
- [17] BOLTOV Y, SKARGA-BANDUROVA I, DERKACH M. A comparative analysis of deep learning-based object detectors for embedded systems[C]//2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS). Dortmund: IEEE, 2023: 1156-1160. DOI: [10.1109/IDAACS58523.2023.10348642](https://doi.org/10.1109/IDAACS58523.2023.10348642)
- [18] AI AMIN R, HASAN M, WIESE V, *et al.* FPGA-based real-time object detection and classification system using YOLO for edge computing[J]. *IEEE Access*, 2024, **12**: 73268-73278
- [19] KAYMAKCI Z E, AKARSU M, ÖZTÜRK C N. Multiple small-scale object detection in aerial vehicle images using standard or optimized YOLO detectors[C]//2023 International Conference on Innovations in Intelligent Systems and Applications (INISTA). Hammamet: IEEE, 2023: 1-5. DOI: [10.1109/INISTA59065.2023.10310562](https://doi.org/10.1109/INISTA59065.2023.10310562)
- [20] XUE Shan, AN Hongyu, LV Qiongying *et al.* Image object detection algorithm based on YOLOv7-tiny in complex background[J]. *Infrared and laser engineering*, 2024, **53**(01): 269-280 (薛珊, 安宏宇, 吕琼莹, 等. 复杂背景下基于 YOLOv7-tiny 的图像目标检测算法 [J]. 红外与激光工程, 2024, **53**(01): 269-280)
- [21] Wagstaff K, Shu H, Castano R. How to leverage existing spectral knowledge when clustering hyperspectral data[C]//AGU Fall Meeting, San Francisco: AGU, 2004
- [22] XU P, LI Q Y, ZHANG B, *et al.* On-board real-time ship detection in hisea-1 SAR images based on CFAR and lightweight deep learning[J]. *Remote Sensing*, 2021, **13**(10): 1995
- [23] ZHOU F B, ZHAO H L, NIE Z. Safety helmet detection based on YOLOv5[C]//2021 IEEE International Conference on Power Electronics, Computer Applications (ICPECA). Shenyang: IEEE, 2021: 6-11. DOI: [10.1109/ICPECA51329.2021.9362711](https://doi.org/10.1109/ICPECA51329.2021.9362711)
- [24] HAN K, WANG Y H, TIAN Q, *et al.* GhostNet: more features from cheap operations[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Seattle: IEEE, 2020: 1577-1586. DOI: [10.1109/CVPR42600.2020.00165](https://doi.org/10.1109/CVPR42600.2020.00165)
- [25] WOO S, PARK J, LEE J Y, *et al.* CBAM: Convolutional block attention module[C]//Proceedings of the 15th European Conference on Computer Vision (ECCV). Munich: Springer, 2018: 3-19
- [26] WENG Xiaoran. A Deep Neural Network End-To-End Deployment System for Edge NPU Platforms[D]. Chengdu: University of Electronic Science and Technology of China, 2023 (翁道然. 面向边缘端 NPU 平台的深度神经网络端到端部署系统 [D]. 成都: 电子科技大学, 2023)
- [27] ZHANG Lihong, CAI Jingju. Target detection system based on lightweight YOLOv5 algorithm[J]. *Computer Technology and Development*, 2022, **32**(11): 134-139 (张利红, 蔡敬菊. 基于轻量化 YOLOv5 算法的目标检测系统 [J]. 计算机技术与发展, 2022, **32**(11): 134-139)
- [28] WU Miao. Lightweight Design of Wearing Detection Algorithm for Hotel Cleaning Staff and Implementation of RK3399pro Platform[D]. Hangzhou: Zhejiang University, 2023 (吴淼. 酒店卫勤人员规范穿戴检测网络轻量化设计与 RK3399pro 平台实现 [D]. 杭州: 浙江大学, 2023)

## 作者简介



刘 冰 (第一作者) 男, 2000 年出生于河南省新乡市, 现为中国科学院国家空间科学中心学生, 研究方向为智能星上信息处理、星上目标检测系统研究等。

E-mail: [2492869534@qq.com](mailto:2492869534@qq.com)



周 海 (通信作者) 男, 1987 年出生于安徽省蚌埠市, 现为中国科学院国家空间科学中心智能室副主任, 博士生导师, 研究员, 研究方向为目标探测与识别、遥感图像处理等。

E-mail: [zhouhai@nssc.ac.cn](mailto:zhouhai@nssc.ac.cn)