文章编号:1671-4598(2025)01-0020-09 DOI:10.16526/j.cnki.11-4762/tp.2025.01.003 中图分类号:TP391.41;TP36;X799 文献标识码:A

# 一种改进 YOLOv5 的轻量化垃圾检测算法

万 涛,李 博,相雨涛
(中北大学 仪器科学与动态测试教育部重点实验室,太原 030051)

**摘要**:生活垃圾及其危害已引起人们的关注,而机器人与目标检测技术的发展为生活垃圾的自动化处理带来了可能性;针对目前生活垃圾检测算法在背景复杂、目标尺寸多样的情况下检测精度低,模型参数量大,深度学习检测算法综合性能不平衡以及在嵌入式设备难以部署等问题,提出了一种改进 YOLOv5 的轻量化垃圾检测算法;在 YOLOv5 模型中用 GSConv 模块代替传统卷积降低计算复杂度,引入了 CBAM 注意力机制,以提取和融合空间和通道信息,增强了网络对目标的表达能力,通过权重量化将模型进行压缩以减少模型大小加快推理速度;实验结果表明,相比于原始的 YOLOv5 算法,改进算法在模型的准确率和平均精确度分别提高了 3%和 2.3%,文件大小减小了 26.6%,综合性能超越了传统的深度学习目标检测算法,对嵌入式平台更加友好。

**关键词**:GSConv;目标检测;轻量化;嵌入式设备;权重量化

# An Improved Lightweight Garbage Detection Algorithm for YOLOv5

WAN Tao, LI Bo, XIANG Yutao
(Key Laboratory of Instrumental Science and Dynamic Testing, Ministry of Education, North University of China, Taiyuan 030051, China)

**Abstract**: Domestic waste and its hazards have attracted people's attention, the development of robots and objection detection technology has brought possibilities for automatic processing domestic waste. In light of the problems such as the low detection accuracy of current domestic waste detection algorithms in complex backgrounds and diverse target sizes, the large number of model parameters, the imbalance in the comprehensive performance of deep learning detection algorithms, and the challenges in deployment on embedded devices, an lightweight garbage detection algorithm based on improved YOLOv5 is proposed. In the YOLOv5 model, the GSConv module is replaced by the traditional convolution to reduce computational complexity. The CBAM attention mechanism is introduced to extract and fuse spatial and channel information, thereby strengthening the expressive capacity of the network on the target. The model is compressed via weight quantization to reduce the model size and accelerate the inference speed. Experimental results show that compared with the original YOLOv5 algorithm, the improved algorithm increases the accuracy and average precision of the model by 3% and 2.3%, respectively, reduces the file size by 26.6%, and its comprehensive performance is superior to that of traditional deep learning object detection algorithms, with a greater friendliness to embedded platforms.

**Keywords**: GSConv, object detection, lightweight, embedded devices, weight quantification

## 0 引言

随着经济的快速发展,我国城镇人口数量急剧增加,人们的生活水平也得到了提高,城市生活垃圾也日益增多,特别是在一些偏远城市,垃圾污染十分严重。巨量垃圾对生态环境造成了非常大的压力。如何实现垃圾的无害化、资源化处理是一个亟待解决的重要问题。对垃圾进行精确合理的检测及分类,并妥善处理垃圾,能够在有效地保护生态的同时,更好地利用可回收资源。垃圾检测及分类是处理垃圾危害的有效解决方案[1]。因此,对于垃圾的高效处理也是具有一定的现实和研究意义。

传统的垃圾检测及分类主要是通过人工的方式进行分拣和分类,存在着劳动强度大,工作效率低,工作环

**收稿日期**:2024-07-30; **修回日期**:2024-08-22。
**基金项目**:国家自然科学基金(61471325);国家自然科学基金青年科学基金(52006114)。
**作者简介**:万 涛(1998-),男,硕士研究生。
李 博(1972-),男,博士,副教授。
**引用格式**:万 涛,李 博,相雨涛. 一种改进 YOLOv5 的轻量化垃圾检测算法[J]. 计算机测量与控制,2025,33(1):20-28.

境差等缺点。正因为如今人工智能的快速发展，可以通过智能化、自动化的方式来代替人工方法来进行垃圾检测及分类任务，减少了人力物力消耗的同时，降低了经济成本也提高了工作效率。目标检测算法结合自动化设备代替人工的方案得到了更多的关注，也就是将垃圾检测应用到嵌入式设备上，让移动端设备来进行垃圾检测及分类，以此实现垃圾检测及分类自动化、智能化[2]。

目前，目标检测技术已被广泛应用于医疗、交通场景检测、智能视频监控等多个领域[3]。该技术主要通过卷积神经网络的卷积层来提取更深层次的目标特征，从而提升检测性能。现在主流的目标检测算法大致分为两类：第一类是双阶段目标检测算法，其中以区域来进行卷积的神经网络（CNN，convolutional neural network)[4]〗最具代表性。虽然其检测精度较高，但速度相对较慢，因为该算法是先生成候选框，再进行检测。第二类则是单阶段目标检测算法，其中只看一次网络（YOLO，you only look once)[5]〗和单镜头多框检测器（SSD，single shot multibox detector)[6]最具代表性，这类算法在速度上具有优势，但通常在精度上较于前一种算法要低一些，该算法将候选框生成和检测过程融合为一体，因此更适合应用于对实时性要求较高的目标检测任务。

在众多的目标检测算法中 YOLO 算法一直更新优化，相比于其他算法在综合性能上有许多优势。文献[5] 首次提出了 YOLO 目标检测框架，该框架将整张图片作为输入，并在输出层上直接预测目标框的位置和检测目标的类别，将目标检测问题转换为回归问题进行处理。之后陆续提出优化过后的 YOLO 算法，例如 YOLOv2[7]、YOLOv3[8]、YOLOv4[9]，对预测准确率、速度、识别对象三方面进行了改进，通过融合特征金字塔网络结构来实现多尺度检测。在 YOLO 目标检测算法系列中，YOLOv5[10]模型更小，训练和推理速度更快，受到广泛使用。

近年来，随着边缘计算和智能终端技术的快速发展，将人工智能模型部署到嵌入式设备上变为可能，受到了广泛关注。其中，类似于 Jetson Nano 这样的人工智能计算平台因其便携性和低功耗的特点，被广泛应用于安防、交通等多个领域。同时，通过使用 TensorRT 或神经处理单元（NPU)，能够使得嵌入式设备在保持检测精度的前提下，实现推理过程的加速[11]。为了提高检测精度，通常需要采用更为复杂的网络模型，这将导致参数数量的增加和计算复杂度的上升，从而需要更强大的图像处理器。然而，考虑到嵌入式设备往往内存有限且计算能力较弱，因此，需要开发更轻量化的目标检测算法，以适配嵌入式设备的资源要求以及计算能力和实时处理的要求。文献［12］提出了一种新型的轻量级卷积技术（GSConv，generalized-sparse convolution)，目的是提高实时目标检测架构的性能。通过引入 GSConv 技术，使轻量级卷积的表示能力尽可能接近标准卷积，降低计算成本，以减轻模型负担，同时保持准确性。还提供基于 GSConv 的设计建议，称为 slim-neck（SNs)，其中包括 GSConv 技术、高效跨阶段部分块（VoV-GSCSP)、改进轻量级检测器的自由技巧、激活函数和边界框回归损失函数的正确选择，这些策略可以在保证准确率的前提下提高实时检测器的计算成本效益。文献［13］基于 GSConv 技术改进 YOLOv8 算法，实现轻量化目标检测。文献［14］设计出一种通道注意力模块和空间注意力模块相结合的注意力机制，卷积块注意模块（CBAM，convolutional block attention module)。

此注意力机制模块是一个简单而有效的注意力模块，方便集成到任何卷积神经网络模型中，并且开销较小，能够与基础 CNN 实现端到端的联合训练。此外，还有 SE[15]和 CA[16]等注意力机制可以提高效率和性能，特别是对于轻型探测器。通过正确使用这些模块，可以获得更好的性价比。文献［17］设计出一种新的神经网络架构，通过使用深度可分离卷积大大降低了模型的计算量和复杂度。此模型 MobileNet 更适用于嵌入式，移动端设备，在降低模型复杂度的同时，也可保证了模型一定的精度，但深度可分离卷积会造成部分语义信息的丢失，达不到标准卷积的效果。文献［18］设计了 AquaVision 模型，通过采用 RetinaNet 作为对象检测模型，使用 ResNet-50 作为骨干网络。RetinaNet 通过 Focal Loss 函数提高预测精度。文献［19］针对嵌入式平台对 YOLOv5 算法进行来改进，提出了 Space Stem 代替原始的 Focus 模块，使用更小内核的 SPP 模块，使主干网络架构更轻量化，也对嵌入式平台更加友好，利于网络量化操作。文献［20］提出轻量级特征融合单镜头多盒检测器（LSSD，lightweight feature fusion single shot multibox detector）算法，此算法是将不同层的不同尺度的特征融合起来。通过使用下采样层来生成新的特征金字塔，将其结果送到指定的多盒检测器来预测最终检测结果。文献［21］基于 YOLOv7 算法融合 SimAM 注意力机制，改进了非极大值抑制算法，相比于原模型，垃圾检测的准确度和速度得到了提升，但这也提高了模型的复杂程度。

针对生活垃圾的检测，设计出更适合嵌入式设备的神经网络算法，实现在移动端设备上进行垃圾检测及分类。目前垃圾检测算法在背景复杂、目标尺寸多样的情况下检测精度低，并且模型参数量大，本文采用 YOLO 系列算法中的 YOLOv5s 作为算法模型，并基于 YOLOv5s 对模型进行优化，设计出一种更适配于嵌入

式设备的轻量化垃圾检测算法。

本文的主要工作如下：

1）采用 GSConv 技术，通过 GSConv 卷积核模块代替传统卷积核，在保证优秀性能的前提下进行特征聚合。以 GSConv 为基础，引入轻量级瓶颈层 GSbottleneck，使用一次性聚合方法来设计跨级部分网络（GSCSP）模块 VoV-GSCSP，将原有的 CBS 模块替换为 GSConv 模块，C3 模块替换为 VoV-GSCSP 模块，进一步降低模型参数量，以减轻模型负担，降低计算量，同时保持准确性。

2）采用 EIoU 损失来替代 CIoU 损失来提高边界框的可靠性，以及选择 SiLU 激活函数防止梯度消失，融合更多的输入信息，从而提升模型性能。

3）使用 CBAM 注意力机制模块，结合了通道注意力和空间注意力，实现对输入特征的双重精炼，提高模型的表征能力和准确性。

4）对训练好的模型，进行权重量化，将权重参数更改为 int16 类型，在模型推理的过程中，极大地减少了计算成本和计算资源的消耗。

## 1 YOLOv5 算法

YOLO 系列算法是结合识别和定位于一体一阶段检测算法。其是一种边框回归的算法，每一阶段网络模型共有 5 个不同版本，其检测性能和网络深度均由浅入深依次增加，这 5 个版本分别为 $n$、$s$、$m$、$l$、$x$，随着模型参数数量的增加，这些版本的检测精度也相应提高。然而，模型的训练和推理时间也随之延长。为了更好地满足嵌入式移动端设备的需求，本文选择检测精度与模型复杂程度较为平衡的网络模型 YOLOv5s，其网络结构如图 1 所示，并在此基础上进行改进优化。

算法主要包括四部分，分别为输入层、主干网络（Backbone）、颈部（Neck）和头部（Head）。网络输入部分需要对数据进行预处理，通过马赛克（Mosai）数据增强等方式对图像预处理。数据增强包括对数据集进行扩充，提高数据量，更加有助于模型的训练。对于存在大量冗余信息的数据以及有信息丢失的数据需要进行剔除、筛选和补充。除此之外，还需要对数据集进行相应的转换，使其满足模型输入层输入数据的要求。经过预处理后的图片数据进入主干网络，主干网络中的 Focus 结构将图像进行切片，以提高图像的处理速度。经过主干网络中的 C3 模块和 Conv 模块进一步提取更深层次的特征数据，有效提高了模型检测的能力。最后，特征图经过 SPPF 模块将多尺度特征信息进行融合。

在 Neck 层使用 FPN（feature pyramid network）＋PAN（path aggregation network）结构实现特征融合，加强了网络特征融合能力，将浅层空间信息和深层语义融合起来，提高了对小目标检测的能力，提高了算法性能。

在头部则是将颈部所输出的不同尺度的特征图生成边界框并预测目标的分类。不同尺度的特征图可以更好对应不同尺寸的目标。

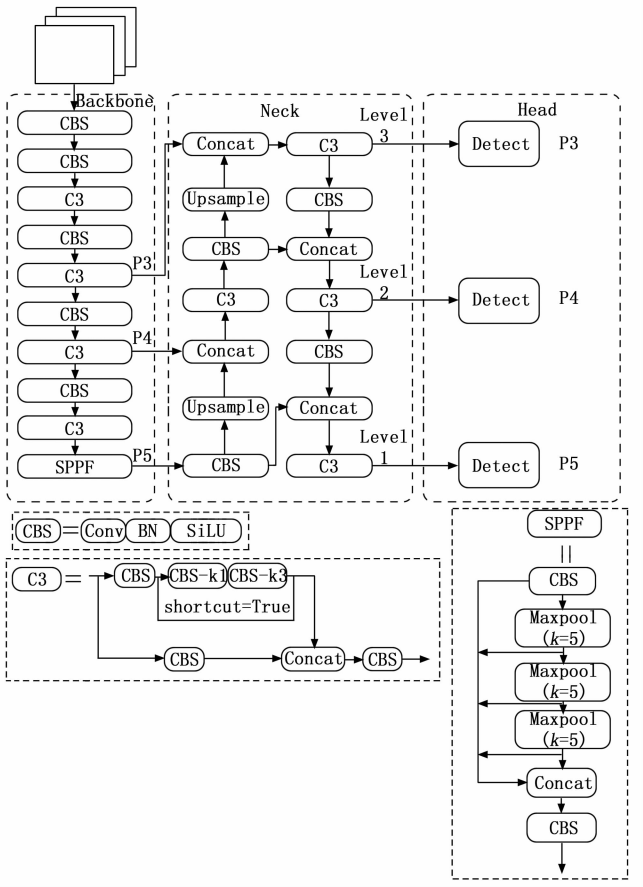



图 1 算法结构

## 2 算法改进

针对现有的 YOLO 算法模型较为复杂，模型网络深度较深，参数量较大并且都是 Float32 类型，计算所消耗的资源时间都十分巨大。YOLO 算法虽然比传统的卷积神经网络 CNN 速度更快，网络深度略浅但仍然是对于嵌入式设备不太友好，对移动端设备部署 AI 算法和边缘计算场景下运用深度神经网络有较大限制。由于垃圾体积小，背景复杂，目标尺寸多样化，检测算法的精度低。本文对算法的改进主要是注重于算法的轻量化，以更适配于嵌入式设备以及边缘计算场景。因此，最终的算法结果是在保持一定的精度下而更加轻量化，计算资源消耗更小，对嵌入式设备更加友好。对 YOLOv5s 算法进行具体的改进，首先是使用 GSConv 技术，通过 GSConv 卷积核模块代替传统卷积以减轻模型负担，降低计算量，同时保持准确性；再使用加入 CBAM 注意力机制模块，加强通道信息和空间信息的

提取和保留；最后通过 AI 开发板将模型进行权重量化，减少权重参数的位宽，来降低计算复杂度，从而压缩模型体积，加快推理速度。

## 2.1　GSConv 技术

相比于深度可分离卷积核（DSC，depth-wise separable convolution）可以降低计算量，提高检测速度，但由于特征的空间维度（宽度和高度）压缩和通道维度扩展都会造成部分语义信息的丢失，达不到标准卷积（SC，standard convolution）可以最大限度地保留了每个通道之间的隐藏连接。而 GSConv 以更低的时间复杂度尽可能地保留这些连接，如图 2 所示，通过使用 shuffle[22-23] 将通道密集卷积生成的特征渗透到 DSC 生成的特征的每一部分中。

shuffle 是一种均匀的混合，它允许来自 SC 的信息完全混合到 DSC 的输出中，通过在不同的通道上均匀地交换本地特征信息，而不需要其他操作。

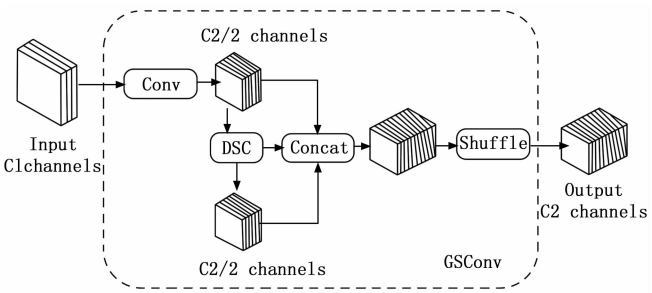
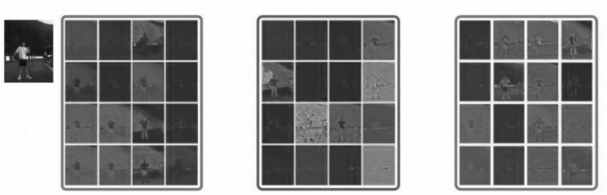



图 2　GSConv 模块结构

GSConv 是使用 SC、DSC 和混洗的混合卷积，为了使 DSC 的输出尽可能接近 SC。图 3 展示了 SC、DSC 和 GSConv 的可视化结果。GSConv 的特征图与 SC 的相似性明显高于 DSC 与 SC 的相似性。文献［12］在轻量级模型上，通过仅使用 GSConv 层替换 SC 层获得了显著的准确性提升；在其他模型上，当在骨干网络中使用 SC 并在颈部使用 GSConv 时，模型的准确性非常接近原始模型；如果再加入一些技巧，模型的准确性和速度将超过原始模型。可以基于 GSConv 的设计建议 slim-neck 来进行模型优化。



(a) features from SC　(b) features from DSC　(c) features from GSConv

图 3　卷积特征图

GSConv 的计算成本约为 SC 的 50%，但对模型学习能力的贡献与 SC 相当。基于 GSConv，引入 GS BottleNeck，其结构如图 4（a）所示。通过结合增强 CNN 学习能力的广义方法，并使用一次聚合策略设计了高效的跨阶段部分网络（CSP）模块 VoV-GSCSP，以降低计算复杂度和推理时间，同时保持准确性。图 4（b）分别展示了 VoV-GSCSP 的另一种设计结构，是简单、直接和更快的推理。

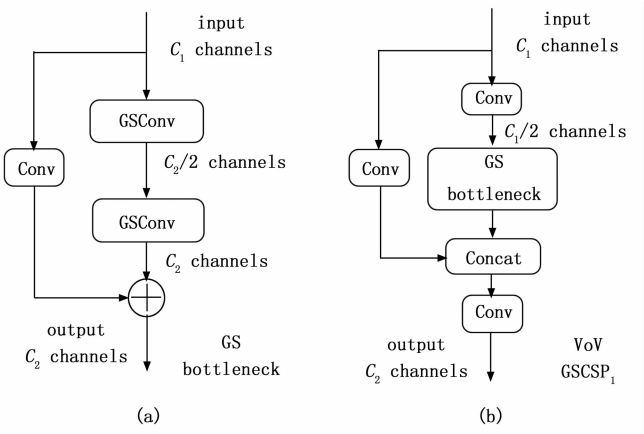



图 4　GS 瓶颈层和 VoV 跨级部分网络模块

实际上，由于硬件友好，更简单的结构被更多地使用，VoV-GSCSP1 表现出更高的性能价格比。通过 GSConv 引入了 Slim-Neck 方法，以减轻模型的复杂度同时可以保持精度。GSConv 更好地平衡了模型的准确性和速度。slim-neck 与 GSConv 结合的方法最小化了 DSC 缺陷对模型的负面影响，并有效利用了 DSC 的优势。

## 2.2　损失和激活函数

IoU 损失对于基于深度学习的检测器具有很大的价值。使预测边界框回归的定位更加准确。随着研究的不断深入，人们提出了更先进的 IoU 损失函数，如 GIoU[24]、DIoU[25]、CIoU[26] 和 EIoU[27]。目前使用最广泛的是 CIoU 损失，公式如（1）所示：

$$Loss_{\mathrm{CIoU}} = 1 - IoU + \frac{\rho^2(b,b^{gt})}{d^2} + \alpha v, IoU = \frac{A \cap B}{A \cup B}$$

$$\alpha = \frac{v}{(1-IoU)+v},\ v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}}\right) \qquad (1)$$

其中：参数“$A$”和“$B$”分别为边界框面积和预测边界框面积；参数“$C$”表示实际边界框和预测边界框的最小围框面积；参数“$d$”表示最小外接围框的对角线顶点的距离；参数 $\rho^2$（$b$，$b^{gt}$）表示实际边界框和预测边界框中心点的欧氏距离，$b$，$b^{gt}$ 分别为实际边界框和预测边界框的中心点；参数“$\alpha$”是权衡指标，参数“$v$”是评价实际边界框和预测边界框纵横比一致性的指标，$w$，$h$，$w^{gt}$ 和 $h^{gt}$ 分别为实际边界框和预测边界框的宽高。

然而，在现实世界的 CIoU 中有一个明显的问题：根据 CIoU 的定义，如果 $\{(w = kw^{gt}, h = kh^{gt}) \mid k$

$\in R+\}$，CIoU 将退化为 DIoU，即 CIoU 中添加的处罚项的相对比例不起作用。此外，由于 CIOU 损失函数只考虑了边界框的长宽比，而忽略了宽高与置信度之间的真正差别。因此，随着预测框与真实框的宽高比呈线性关系时，其宽度与高度很难同步增大或减小，从而导致回归优化停止。对于 EIoU 没有面临这个问题，因为它直接使用预测边界框的 $w$ 和 $h$ 作为惩罚项，而不是 $w$ 和 $h$ 的比值。EioU 的公式如（2）所示：

$$Loss_{\mathrm{EIoU}} = 1 - IoU + \frac{\rho^2(b,b^{gt})}{d^2} + \frac{\rho^2(w,w^{gt})}{C_w^2} + \frac{\rho^2(h,h^{gt})}{C_h^2} \tag{2}$$

其中：参数“$d$”表示最小外接围框的对角线顶点的距离；$C_w$、$C_h$ 是最小外接围框的宽和高。当物体宽度与高度相差较大时，EIoU 的边界框回归更为精确。

EIOU 是在 CIOU 的惩罚项的基础上将预测框和真实框的宽高比的影响因子分开，从而分别计算预测框和真实框的长和宽，并且加入 Focal Loss 优化了边界框回归任务中的样本不平衡问题，这样就避免了 CIOU 存在的问题。先前基于 IOU 的损失，例如 CIOU 和 GIOU，不能有效地测量目标框和锚框之间的差异，这导致边界框回归模型优化的收敛速度慢，定位不准确，因此，选择 EIOU 的效果更好。

任意一个算法模型，激活函数的质量对网络的性能是至关重要的。本文选择 SiLU[28] 激活函数，SiLU 激活函数能够有效防止梯度消失和对于输入为负值时无响应的问题，在接近零时具有更平滑的曲线，在整个定义域内都有导数，有利于优化，并且由于其使用了 sigmoid 函数，可以使网络的输出范围在 0 和 1 之间，这有利于融合更多的输入信息，从而提升模型性能。计算方法如式（3）所示：

$$SiLU = x \cdot Sigmoid(x) = \frac{xe^x}{e^x + 1} \tag{3}$$

其中：$x$ 为输入特征。

## 2.3 CBAM 注意力机制

卷积块注意模块（CBAM），这是一种简洁而有效的前馈卷积神经网络注意机制，由通道注意力模块（Channel Attention Module）和空间注意力模块（Spatial Attention Module）两部分组成，其整体结构如图 5 所示。对于给定的中间特征映射，该模块依次在通道和空间两个独立维度上推导出注意力映射，并将这些注意力映射与输入特征映射相乘，从而实现自适应特征的细化。值得注意的是，CBAM 作为一种轻量级的通用模块，能够在减少参数数量和计算成本的同时，无缝地集成到各种卷积神经网络架构中，体现出即插即用的特性，并能够与基础的 CNN 一起进行端到端的训练。引入了 CBAM 注意力机制，以提取和融合空间和通道信息，增强了网络对目标的表达能力。

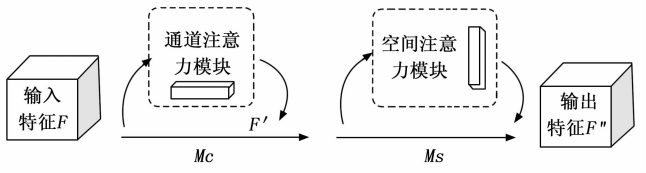



图 5 注意力机制模块

将输入特征图 $F$ 与通道注意力模块 $Mc$ 进行卷积，再将得到的输出与原特征图 $F$ 相乘得到中间特征图 $F'$，然后将 $F'$ 与空间注意力模块 $Ms$ 进行卷积，将得到的输出再与 $F'$ 相乘，得到最终结果 $F''$。具体过程可用以下公式表示：

$$F' = M_C(F) \otimes F \tag{4}$$

$$F'' = M_S(F') \otimes F' \tag{5}$$

CBAM 注意力机制包含通道注意力和空间注意力两个注意力模块，其结构如图 6 所示。

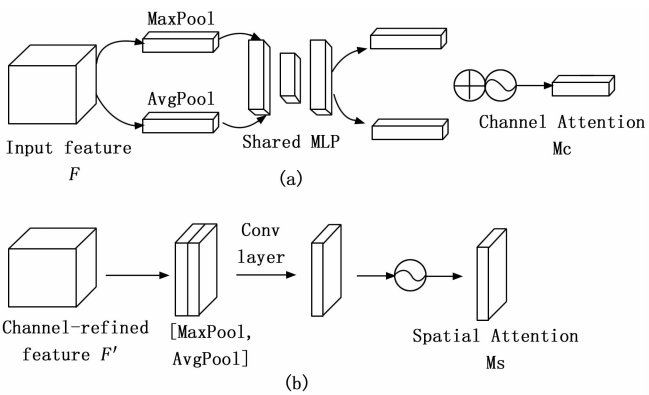



图 6 通道注意力和空间注意力图

其中通道注意力模块（Channel Attention Module）是为了提高卷积神经网络在通道维度上的特征提取能力。通过对特征图在通道维度上进行统计，模块能够计算每个通道的重要性权重，使模型更加关注更重要的通道，同时忽略不重要的通道信息。这种机制使模型在处理特征图时能够聚焦于重要的通道特征，忽略不重要的通道信息，从而提高计算效率。

通道注意力机制通过对特征图进行空间压缩，生成一个一维矢量。具体而言，它采用平均池化（Average Pooling）和最大池化（Max Pooling）在空间维度上对特征图进行压缩。平均池化和最大池化操作分别聚合特征图的空间信息，并将结果输入到一个共享网络中。然后，压缩输入特征图的空间维数，再通过逐元素求和的方式合并这些结果，从而生成通道注意力图。通道注意力机制的过程可以表示为以下公式：

$$M_C(F) = \sigma\{MLP[AvgPool(F)] + MLP[MaxPool(F)]\} = \sigma\{W_1[W_0(F^c_{avg})] + W_1[W_0(F^c_{max})]\} \tag{6}$$

其中：$\sigma$ 表示 *sigmoid* 函数，*MLP* 是多层感知机，$w_0$ 和 $w_1$ 是权重。

同样，空间注意力机制（Spatial Attention Module）是利用特征的空间关系生成空间注意图。与通道注意力机制不同，空间注意力机制侧重于关注信息在特征图中的“位置”，与通道注意力机制形成互补关系。在计算空间注意力时，通过在通道维度上执行平均池化和最大池化操作来提取特征信息，将其结果相连接，得到新的特征图。再用标准卷积对其进行连接、卷积，得到二维空间注意力图。空间注意力计算为：

$$M_C(F)=\sigma(f^{7\times7}([AvgPool(F);MaxPool(F)])=\sigma(f^{7\times7}([F^S_{avg};F^S_{max}])) \tag{7}$$

其中：$\sigma$ 表示 *sigmoid* 函数，$f^{7\times7}$ 表示为 7×7 的卷积核的卷积运算。

这两个模块可以以并行或顺序的方式放置。结果表明，顺序排列的结果比并行排列的结果好。对于排列的顺序，通道在前面略优于空间在前面。

CBAM 通过结合通道注意力和空间注意力，对输入特征进行了双重精炼。它能够根据任务需求和内容上下文，动态调整特征图中各个通道和空间位置的重要性。这种设计使模型能够更专注于重要的通道和空间信息，从而进一步提高模型的表征能力和决策的准确性。对于嵌入式设备，CBAM 模块设计简洁并且灵活，计算效率高，在保持较低的计算成本的前提下提高性能。

### 2.4　模型权重量化

由于嵌入式平台计算资源，存储空间有限，对于传统的深度学习算法来说，嵌入式平台无法满足如此庞大的计算量，也达不到如此高的算力，传统的卷积计算对嵌入式设备并不友好，即使可以进行推理，但所消耗的时间较长，无法满足实时处理需求。为解决此类问题，可以对模型进行压缩。模型的压缩方法有：网络剪枝、量化、低秩分解、知识蒸馏。

网络剪枝、知识蒸馏都是针对模型而言，设计出一种更加精炼的模型，在模型精度保持一定的前提下，降低模型整体的复杂度和大小，从而使得模型更加轻量化。低秩分解是运用矩阵的相关性原理，如果矩阵之间各行的相关性很强，那么就表示这个矩阵实际可以投影到更低维的线性子空间，也就是用几个向量就可以完全表达，那么它就是低秩的，这样也就降低了模型的复杂度。

所谓模型量化就是将浮点存储（运算）转换为整型存储（运算）的一种模型压缩技术。实际上就是原来表示一个权重需要使用 float32 表示，量化后只需要使用 int8 来表示，仅仅这一个操作，就可以获得接近 4 倍的网络加速。神经网络模型性能越高，引入的参数量和计算量就越大，通过模型量化，将模型一些特定算子的权重参数由 float32 变为 int8，或者 int16，有效减小了模型的大小和计算强度，有效地减少了模型的内存占用，以此更有利于模型在边缘终端设备上进行部署，也可以加快模型的推理速度。虽然模型量化减少了模型的大小以及计算量，也加快了推理速度，但也有一定的缺点。权重参数的类型发生了改变，使参数的位宽降低，导致了模型的精度降低，模型的准确度下降。这对于传统的卷积神经网络算法来说是不利的，但对于边缘嵌入式设备来说是在可接受的范围内。在损失一定的模型精度的情况下，减小了模型的大小和计算量，加快模型的推理速度更适合于嵌入式移动端设备的部署及使用。为了保证模型一定的精度，本文算法采用 int16 量化，虽然 int16 的压缩效果不如 int8 量化，但是还是保持了模型较高的准确度。

## 3　实验结果及分析

### 3.1　数据集

本文使用的数据集是通过网上收集所组成的数据集。该数据集包含来自与各种生活垃圾物品相关的 44 个不同类别，分别为：一次性快餐盒，书籍纸张，充电宝，剩饭剩菜，包，垃圾桶，塑料器皿，塑料玩具，塑料衣架，大骨头，干电池，快递纸袋，插头电线，旧衣服，易拉罐，枕头，果皮果肉，毛绒玩具，污损塑料，污损用纸，洗护用品，烟蒂，牙签，玻璃器皿，砧板，筷子，纸盒纸箱，花盆，茶叶渣，菜帮菜叶，蛋壳，调料瓶，软膏，过期药物，酒瓶，金属厨具，金属器皿，金属食品罐，锅，陶瓷器皿，鞋，食用油桶，饮料瓶，鱼骨。其中所有图像都有良好的标注信息以获得结果的准确性。并将数据集按照 8∶2 的比例划分为训练集和验证集。

### 3.2　实验平台搭建

实验分为两步，第一步是搭建模型，并进行训练调优，第二步是将训练好的模型进行模型量化并转换成 AI 开发板所支持的模型格式。本实验所使用的实验环境及计算机配置如下。

1）PC 平台：操作系统为 Window11 操作系统，CPU 为 Gen Intel（R）Core（TM）i7-12700H 2.30 GHz，GPU 为 NVIDIA GeForce RTX3060；显存 6 G，内存 16 G。编译环境为 python 3.8.19，pytorch 1.13.1，cuda 11.6。

2）嵌入式平台使用的是 RK3568，CPU 为四核 Cortex-A55，2.0 GHz，NPU 1.0Tops，GPU 为 Mali G52。

### 3.3　参数设置及评价指标

输入图片大小（Img-Size）设定为 640 * 640，批次数量（Batch-Size）设定为 32，轮次（Epochs）设定为 300 轮，初始学习率为 0.01，循环学习率为 0.2，权值

衰减为 0.000 5，网络训练使用 SGD 优化器。

本研究主要从精确度（P，precision）、召回率（R，recall）、平均精密度（mAP，mean average precision）、每秒帧数（FPS，frames per second）、参数量（Params）、浮点运算数（FLOPs/G）、文件权重大小等作为对模型评价指标。上述指标的公式为：

$$Precision = \frac{TP}{TP + FP} \tag{8}$$

$$recall = \frac{TP}{TP + FN} \tag{9}$$

$$mAP = \frac{1}{m}\sum_{i=0}^{m}\int_{0}^{1} P(R)\mathrm{d}R \tag{10}$$

其中：$TP$ 为样本标签为正，实际预测为正的数量；$FP$ 为样本标签为负，实际预测为正的数量；$FN$ 为样本标签为正，实际预测为负的数量。$mAP$ 是所有检测类别的平均精度的均值，其中 $AP$ 是平均精确度，是在不同召回率水平下精度的平均值。在 $PR$ 曲线图上表现为 $PR$ 曲线下面的面积。$AP$ 的值越大，则说明模型的平均准确率越高。

### 3.4 目标检测算法对比

为进一步验证本文算法的合理性和有效性，将本文选用的算法与经典网络模型如 Mask RCNN、EfficientDet-D2、SSD、YOLOv3 和 YOLOv4 进行横向比较，其实验结果如表 1 所示。

表 1 不同算法对比算法

| 算法 | $P$/% | $R$/% | $mAP$/% |
|---|---|---|---|
| SSD | 70 | 63.8 | 67.2 |
| YOLOv3 | 70.3 | 63.4 | 66.3 |
| YOLOv4 | 71 | 65.5 | 67.4 |
| YOLOv5s | 72.4 | 65 | 68.3 |
| EfficientDet | 66.5 | 54.3 | 58.4 |
| Mask RCNN | 69.3 | 60.2 | 63.6 |
| 本文算法 | 75.4 | 67.3 | 70.6 |

由表中结果可以看出，相较于经典的 SSD、YOLOv3、YOLOv4 和 EfficientDet 网络，YOLOv5s 在模型规模、查准率以及查全率等方面均表现出显著的优势，因此选择对 YOLOv5s 进行改进，基于 YOLOv5s 模型采用 GSConv 技术、Slim-Neck 优化策略进行改进，精确度、平均精密度提升 2%，虽然提升范围小，但模型的计算量、权重大小都较于原始 YOLOv5 模型有所减少，对嵌入式设备更加友好。

### 3.5 注意力机制对比

为了进一步检验 CBAM 模块在模型中的作用，本实验将原始模型与不同的注意力机制进行结合对比，实验结果如表 2 所示。相较于其他两种注意力机制 SE、CA，CBAM 在模型大小和计算量增长较少的情况下，模型的综合性能提升较高。

表 2 注意力机制对比

| 注意力机制 | $P$/% | $R$/% | $mAP$/% | 权重文件/MB |
|---|---|---|---|---|
| SE | 72.1 | 62.9 | 67.5 | 14.6 |
| CA | 72.8 | 62.6 | 67.6 | 14.6 |
| CBAM | 73.6 | 63.8 | 69.3 | 14.6 |

### 3.6 消融实验

为了更好地检验各个模块和各个模块间的结合对原有模型性能的提高效果。本节通过消融实验在原有模型的基础上，逐个增加改进方法来进行对比实验。实验结果如表 3 所示。其中 √ 表示模型有该模块，× 则表示没有该模块。通过表 3 的实验结果可以发现，每一步的 YOLOv5 优化操作都会对 YOLOv5 模型产生影响。其中，表 3 结果表明，将 Neck 层中的传统卷积模块替换为 GSConv 后，不仅有效减少了参数量（Params）和浮点运算次数（FLOPs），还在模型性能 mAP@0.5 上实现了一定程度的提升。虽然提升幅度有限，但验证了 Slim-Neck 设计范式在 YOLOv5 上的可行性。并且将 EIoU 损失来代替原始模型的 CIoU 损失来提高边界框的可靠性，损失的替换也对模型的平均精度有微小的提升。此外，算法模型还引入了 CBAM 注意力机制，分别在单独使用和与 Slim-Neck 结合使用两种场景下进行了验证。在单独使用注意力机制的情况下，和与 Slim-Neck 结合使用的情况下，由于前面比较了三种注意力机制的效果，其中 CBAM 效果最佳，因此选择将 CBAM 与 Slim-Neck 组合使用。结果表明，配合使用 Slim-Neck 的模型相比基线模型，性能有所提升。

表 3 消融实验

| Slim-Neck | CMAM | EIoU | FLOPs/G | $P$/% | $mAP$/% | Params/M |
|---|---|---|---|---|---|---|
| × | × | × | 15.8 | 72.4 | 68.3 | 7.01 |
| √ | × | × | 14.7 | 73.3 | 69.4 | 6.80 |
| × | × | √ | 15.8 | 72.7 | 68.9 | 7.01 |
| × | √ | √ | 15.8 | 73.6 | 69.3 | 7.04 |
| √ | √ | √ | 14.7 | 75.4 | 70.6 | 6.83 |

原始的 YOLOv5s 算法在处理不同大小，复杂背景，以及较小的图像时没有达到了良好的效果，并且模型文件也不适用于在移动端嵌入式设备中部署。针对上述情况，对改进后的 YOLOv5 算法的性能进行了研究。随机选取符合训练数据集类别的照片做测试。图 7 显示了算法的测试结果。

### 3.7 模型量化结果分析

在训练结束后，将性能最好的权重文件进行 int16 量化，理论上说 int16 类型的参数比 float32 类型的参数要减少两倍的体积，推理速度提升两倍左右。模型量化后的效果如表 4 所示，由于 YOLOv5s 算法模型中一部分的权重参数是 float16 类型，模型量化后模型的大小

图 7　算法检测图

降低 30%左右，推理速度加快，模型量化后更有利于部署于嵌入式设备以及移动端设备上，实现 AI 算法的落地应用。对一些边缘计算场景有重要的实际意义和应用价值。

表 4　量化前后对比

| 模型 | FPS | 权重大小/MB |
|---|---|---|
| 量化前 | 97 | 14.3 |
| 量化后 | 118 | 10.5 |

## 4　结束语

针对传统目标检测算法不适于部署在嵌入式设备上，且在垃圾检测背景复杂、目标尺寸多样的情况下检测精度低等问题，本文提出了一种改进 YOLOv5 的轻量化检测算法，通过引入 GSconv 卷积核模块代替传统卷积核以及轻量级瓶颈层 GSbottleneck，进行特征聚合，在保持准确性的情况下，降低模型参数量和计算量，以减轻模型负担。采用 EIoU 损失来提高边界框的可靠性，以及选择 SiLU 激活函数有效防止梯度消失，融合更多的输入信息，从而提升模型性能。使用 CBAM 注意力机制模块，结合了通道注意力和空间注意力，实现对输入特征的双重精炼，提高了模型的表征能力和准确性。最后对训练好的模型，进行权重量化，极大地减少了计算资源的消耗，并更利于嵌入式设备的部署。通过实验表明，改进后的算法与 YOLOv5 和其他主流算法相比具有更好综合性能以及减少了存储和计算资源，更利于在嵌入式设备上部署，具有较高的实际应用意义。

**参考文献：**


[1] 李金玉，陈晓雷，张爱华，等. 基于深度学习的垃圾分类方法综述 [J]. 计算机工程，2022，48 (2)：1－9.

[2] 赵和月. 基于深度学习的生活垃圾检测算法研究 [D]. 南京：南京邮电大学，2023.

[3] 罗会兰，陈鸿坤. 基于深度学习的目标检测研究综述 [J]. 电子学报，2020，48 (6)：1230－1239.

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

[5] Redmon J，Divvala S，Girshick R，et al. You only look once：Unified，real-time object detection [C] //Proceedings of the IEEE conference on computer vision and pattern recognition. 2016：779－788.

[6] LIU Wei，ANGUELOV D，ERHAN D，et al. SSD：Single shot MultiBox detector [C] //The 14th European Conference on Computer Vision，Amsterdam，The Netherlands，2016：21－37.

[7] REDMON J，FARHADI A. YOLO9000：Better，faster，stronger [C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition，2017：7263－7271.

[8] REDMON J，FARHADI A. YOLOv3：An incremental improvement [J]. Arxiv Preprint Arxiv：1804.02767，2018.

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

[10] Jaiswal S K，Agrawal R. A comprehensive review of YOLOv5：advances in real-time object detection [J]. International Journal of Innovative Research in Computer Science & Technology，2024，12 (3)：75－80.

[11] WANG X，YUE X，LI H，et al. A high-efficiency dirty-egg detection system based on YOLOv4 and TensorRT [C] // 2021 International Conference on Advanced Mechatronic Systems (ICAMechS). IEEE，2021：75－80.

[12] LI H，LI J，WEI H，et al. Slim-neck by GSConv：a lightweight-design for real-time detector architectures [J]. Journal of Real-Time Image Processing，2024，21 (3)：62.

[13] 刘子洋，徐慧英，朱信忠，等. Bi-YOLO：一种基于 YOLOv8 改进的轻量化目标检测算法 [J]. 计算机工程与科学，2024，46 (8)：1444－1454.

[14] 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.

[15] HU J，SHEN L，SUN G. Squeeze-and-excitation networks [C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018：7132

-7141.
[16] HOU Q, ZHOU D, FENG J. Coordinate attention for efficient mobile network design [C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021: 13713-13722.
[17] SANDLER M, HOWARD A, ZHU M, et al. Mobilenetv2: Inverted residuals and linear bottlenecks [C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 4510-4520.
[18] PANWAR H, GUPTA P K, SIDDIQUI M K, et al. AquaVision: Automating the detection of waste in water bodies using deep transfer learning [J]. Case Studies in Chemical and Environmental Engineering, 2020, 2: 100026.
[19] 刘乔寿, 赵志源, 王均成, 等. 高性能 YOLOv5: 面向嵌入式平台高性能目标检测算法研究 [J]. 电子与信息学报, 2023, 45 (6): 2205-2215.
[20] MA W, WANG X, YU J. A lightweight feature fusion single shot multibox detector for garbage detection [J]. IEEE Access, 2020, 8: 188577-188586.
[21] 陈 君, 赵小会, 王博士, 等. 基于 YOLOv7 的垃圾检测方法研究 [J]. 计算机测量与控制, 2024, 32 (12): 1-8.
[22] ZHANG X, ZHOU X, LIN M, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices [C] //Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 6848-6856.
[23] 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.
[24] REZATOFIGHI H, TSOI N, GWAK J Y, et al. Generalized intersection over union: A metric and a loss for bounding box regression [C] //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 658-666.
[25] ZHENG Z, WANG P, LIU W, et al. Distance-IoU loss: Faster and better learning for bounding box regression [C] //Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34 (7): 12993-13000.
[26] ZHENG Z, WANG P, REN D, et al. Enhancing geometric factors in model learning and inference for object detection and instance segmentation [J]. IEEE Transactions on Cybernetics, 2021, 52 (8): 8574-8586.
[27] ZHANG Y F, REN W, ZHANG Z, et al. Focal and efficient IOU loss for accurate bounding box regression [J]. Neurocomputing, 2022, 506: 146-157.
[28] ELFWING S, UCHIBE E, DOYA K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning [J]. Neural Networks, 2018, 107: 3-11.

(上接第 8 页)

[40] 张 欢, 周 严, 李嘉豪. 基于振动的滚珠丝杠预紧力失效预测 [J]. 组合机床与自动化加工技术, 2023 (6): 145-148.
[41] 徐卫晓, 谭继文, 战 红. 改进 DST 在数控机床滚珠丝杠副智能故障诊断中的应用 [J]. 制造技术与机床, 2014 (9): 4.
[42] 唐 旭, 谭继文, 徐卫晓, 等. 基于卷积神经网络的数控机床滚珠丝杠副故障诊断研究 [J]. 煤矿机械, 2019, 40 (1): 3.
[43] XIE Y, LIU C, HUANG L, et al. Ball screw fault diagnosis based on wavelet convolution transfer learning [J]. Sensors, 2022, 22 (16): 6270.
[44] YIN C, WANG Y, HE Y, et al. Early fault diagnosis of ball screws based on 1-D convolution neural network and orthogonal design [J]. Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability, 2021, 235 (5): 783-797.
[45] RIAZ N, SHAH S I A, REHMAN F, et al. An intelligent hybrid scheme for identification of faults in industrial ball screw linear motion systems [J]. IEEE Access, 2021, 9: 35136-35150.
[46] ZHANG L, GAO H. A deep learning-based multi-sensor data fusion method for degradation monitoring of ball screws [C] //2016 Prognostics and System Health Management Conference (PHM-Chengdu), 2017.
[47] AZAMFAR M, LI X, LEE J. Intelligent ball screw fault diagnosis using a deep domain adaptation methodology [J]. Mechanism and Machine Theory, 2020, 151: 103932.
[48] PANDHARE V, LI X, MILLER M, et al. Intelligent diagnostics for ball screw fault through indirect sensing using deep domain adaptation [J]. IEEE Transactions on Instrumentation and Measurement, 2020, 70: 1-11.
[49] AN Z, JIANG X, CAO J, et al. Self-learning transferable neural network for intelligent fault diagnosis of rotating machinery with unlabeled and imbalanced data [J]. Knowledge-Based Systems, 2021, 230: 107374.
[50] WEISS K, KHOSHGOFTAAR T M, WANG D. A survey of transfer learning [J]. Journal of Big Data, 2016, 3 (1): 1-40.
[51] ZHUANG F, QI Z, DUAN K, et al. A comprehensive survey on transfer learning [J]. Proceedings of the IEEE, 2020, 109 (1): 43-76.