



# 基于改进 YOLOv5s 的轻量化布匹瑕疵检测算法

邹宏睿<sup>1</sup>, 任佳<sup>1,2</sup>, 潘海鹏<sup>1,2</sup>, 周传辉<sup>1</sup>

(1. 浙江理工大学信息科学与工程学院, 杭州 310018; 2. 浙江理工大学常山研究院有限公司, 衢州 324299)

**摘要:** 针对纺织生产中布匹瑕疵检测高精度、实时性的需求, 提出了一种基于改进 YOLOv5s 的轻量化布匹瑕疵检测算法 (GhostNet-CBAM-Partial convolution-YOLOv5s, GCP-YOLOv5s)。该算法首先引入 GhostNet 中的 GhostConv 模块, 对原主干网络进行优化重构, 大幅减少网络参数; 其次, 在主干特征提取网络中加入 CBAM (Convolutional block attention module) 注意力机制, 增加网络的特征提取能力; 最后, 设计了基于 Partial convolution 的改进 C3 模块 (C3-Partial convolution, C3-P), 在降低模型参数数量的同时提高特征融合能力。在自建布匹瑕疵数据集上进行了对比测试, 结果表明: 与基准模型 YOLOv5s 相比, GCP-YOLOv5s 的参数数量降低了 41.6%, 计算量降低了 43.1%, 检测速度提高了 12 FPS, 检测精度提升了 1.7%。GCP-YOLOv5s 算法在保证模型轻量化的同时具有较高的检测精度, 可以满足布匹瑕疵检测的高精度和实时性要求。

**关键词:** 布匹瑕疵检测; YOLOv5s; GhostNet; 注意力机制; 高精度; 实时性

**中图分类号:** TP183

**文献标志码:** A

**文章编号:** 1673-3851 (2024) 05-0389-10

**引文格式:** 邹宏睿, 任佳, 潘海鹏, 等. 基于改进 YOLOv5s 的轻量化布匹瑕疵检测算法[J]. 浙江理工大学学报(自然科学), 2024, 51(3): 389-398.

**Reference Format:** ZOU Hongrui, REN Jia, PAN Haipeng, et al. A lightweight fabric defect detection algorithm based on improved YOLOv5s[J]. Journal of Zhejiang Sci-Tech University, 2024, 51(3): 389-398.

## A lightweight fabric defect detection algorithm based on improved YOLOv5s

ZOU Hongrui<sup>1</sup>, REN Jia<sup>1,2</sup>, PAN Haipeng<sup>1,2</sup>, ZHOU Chuanhui<sup>1</sup>

(1. School of Information Science and Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China; 2. Zhejiang Sci-Tech University Changshan Research Institute Co., Ltd., Quzhou 324299, China)

**Abstract:** A lightweight fabric defect detection algorithm, GhostNet-CBAM-Partial convolution-YOLOv5s (GCP-YOLOv5s) based on improved YOLOv5s was proposed to meet the high-precision and real-time requirements of fabric defect detection in textile production. Firstly, this algorithm introduced the GhostConv module in GhostNet to optimize and reconstruct the original backbone network, significantly reducing network parameters. Secondly, CBAM (convolutional block attention module) attention mechanism was incorporated into the backbone network to increase its feature extraction capability. Finally, an improved C3 module C3-P (C3-Partial convolution) module based on Partial convolution was designed to reduce the number of model parameters while improving the feature fusion capability. Comparative tests were conducted on a dataset of self-built fabric defects on site, and the results showed that compared with the benchmark model YOLOv5s, the parameters and computational

收稿日期: 2023-12-08 网络出版日期: 2024-03-13

基金项目: 浙江省“尖兵”研发攻关计划项目(2023C01002); 浙江省教育厅一般科研项目(Y202147717); 浙江理工大学研究生优秀学位论文培育基金(LW-YP2022080)

作者简介: 邹宏睿(1999-), 男, 广东惠州人, 硕士研究生, 主要从事故障诊断方面的研究。

通信作者: 任佳, E-mail: jren@zstu.edu.cn

complexity of GCP-YOLOv5s were reduced by 41.6% and 43.1%, respectively, while the detection speed and detection accuracy were increased by 12 FPS and 1.7%, respectively. The GCP-YOLOv5s algorithm has high detection accuracy while ensuring model lightweighting, meeting the requirements of high precision and real-time performance in fabric defect detection at the same time.

**Key words:** fabric defect detection; YOLOv5s; GhostNet; attention mechanism; high-precision; real-time

## 0 引言

布匹瑕疵检测在纺织生产的质量控制中起着至关重要的作用。布匹瑕疵检测主要包括规格尺寸、疵点、色差等内容,通过瑕疵检测可以及时发现并解决生产过程中的质量问题。目前,大部分纺织企业采用人工检测的方式进行布匹瑕疵检测,但该方法成本高、准确率低并且效率低下。部分纺织企业采用传统机器视觉的方式进行布匹瑕疵检测,通过阈值分割、形态学等图像处理方法对布匹图像进行瑕疵检测,但是这些方法对图像的质量有较高要求,在布匹瑕疵与图像背景对比度低等情况下检测效果差,并且算法的参数设置依赖专家经验,难以迁移近年来,随着深度学习算法的发展,布匹瑕疵检测逐渐向自动化、智能化发展。深度学习算法在计算复杂度、检测精度等方面较传统机器视觉方法有着显著优势,可以实现布匹瑕疵的实时检测。

基于深度学习的布匹瑕疵检测算法,根据训练数据是否带有标签信息可以分为有监督学习<sup>[1]</sup>、半监督学习<sup>[2]</sup>、无监督学习<sup>[3]</sup>三类。布匹瑕疵检测的目标主要是找出布匹中瑕疵的位置,并将其正确分类,因此可以将其视为目标检测任务。目标检测任务相关算法可以按照候选区域的处理方式分为单阶段和双阶段两大类,其中:单阶段算法的代表有 SSD (Single shot multibox detector)<sup>[4]</sup> 和 YOLO (You only look once)<sup>[5]</sup>,双阶段算法的典型代表为 Faster R-CNN (Faster region-cnn)<sup>[6]</sup>。YOLO 系列算法作为单阶段检测的代表算法,将特征提取、目标分类和位置回归任务进行合并,具有参数量小、检测精度高、检测速度快的优点。

布匹瑕疵检测任务是特殊的目标检测任务,具有数据集样本不均衡、瑕疵目标占比小、瑕疵种类多样以及精度、实时性要求等特点,往往需要结合其任务特点来设计算法。针对布匹瑕疵数据集中类别数量不均衡的问题,蔡兆信等<sup>[7]</sup>改进了 RPN (Region proposal network) 网络中 anchors (锚框) 的大小和数量,提高了瑕疵检测的准确率;该方法的不足之处在于锚框数量难以确定,并且在锚框数量过多时检

测的速度较慢。针对网络参数量大的问题,Zhou 等<sup>[8]</sup>使用了 CARAFE 轻量化上采样技术,减少了特征融合网络的参数。针对瑕疵样本不足的问题,Liu 等<sup>[9]</sup>使用 GAN (Generative adversarial network) 网络来生成瑕疵图像,进行训练样本扩充;这种方法可以有效增加样本图像数量,但是也带来了训练时间过长的问题。针对小瑕疵难以检测的问题,李辉等<sup>[10]</sup>提出一种双路高分辨率转换网络算法,利用双路高分辨率特征提取方法来消除背景的噪声干扰,提高了小瑕疵的检测准确率,以满足布匹瑕疵检测中高准确率的要求;但是该算法检测速度慢,难以满足实时性要求,不能在嵌入式设备上部署。

尽管基于深度学习的布匹瑕疵检测方法已经取得了大量的成果,但这些方法普遍存在计算复杂度高、耗时长等问题,在实际应用时存在一定的局限性<sup>[11]</sup>,如布匹瑕疵检测不及时、布匹瑕疵定位不精准等。难以满足布匹瑕疵检测的高精度和实时性要求,也不适合在嵌入式和移动设备中进行部署。随着 MobileNet<sup>[12-13]</sup>、ShuffleNet<sup>[14-15]</sup>、GhostNet<sup>[16-17]</sup> 等轻量化网络的提出,兼顾检测精度和实时性的深度学习算法成为了新的研究方向。

为了实现布匹瑕疵的高精度和实时检测,本文提出了一种基于 YOLOv5s 的轻量化布匹瑕疵检测算法 (GhostNet-CBAM-Partial convolution-YOLOv5s, GCP-YOLOv5s)。该算法首先引入 GhostNet 来实现主干特征提取网络的轻量化;然后添加 CBAM (Convolutional block attention module) 注意力机制来增强主干网络的特征提取能力;最后设计了一种基于 Partial convolution 的改进 C3 模块 C3-P (C3-Partial convolution),在降低模型参数量的同时提高特征融合能力。本文在课题组采集的布匹瑕疵数据集上进行了实验,以验证该算法的有效性。

## 1 YOLOv5 算法介绍

YOLOv5 是 YOLO 系列发展至今的第 5 代检测算法,它在更加轻量化的同时实现了更高的检测精度。根据网络的深度和宽度,YOLOv5 有 4 个主要版本,分别为 YOLOv5s、YOLOv5m、YOLOv5l 和 YOLOv5x;这



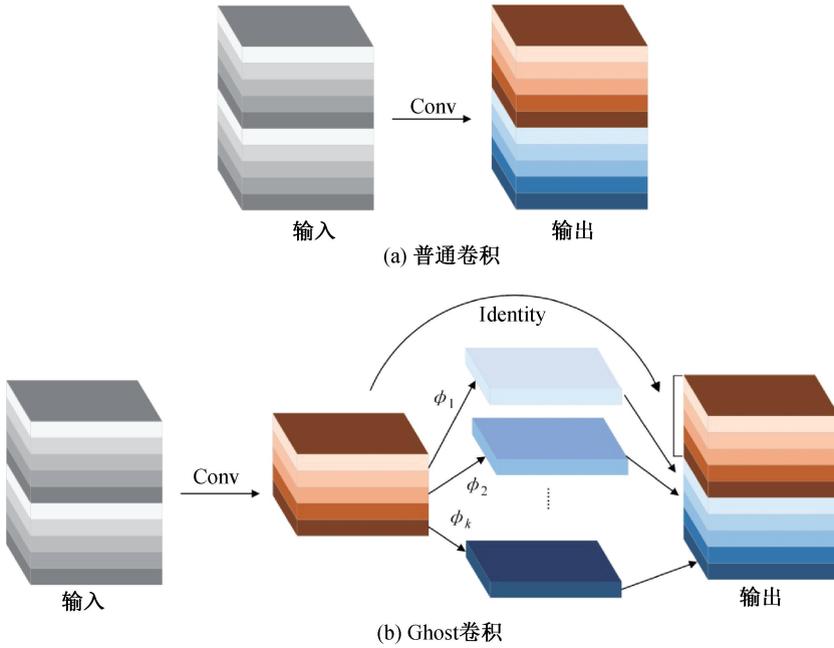


图3 普通卷积和 Ghost 卷积操作示意图

具体计算过程描述如下:假设输入特征图的尺寸为  $h \times w \times c$ , 使用卷积核为  $k \times k$  的卷积块, 产生尺寸为  $h' \times w' \times n$  的输出特征图, 其中:  $h, w, c$  分别是输入特征图的高、宽和通道数,  $k$  为卷积核尺寸,  $h', w', n$  分别是输出特征图的高、宽和通道数。

使用普通卷积进行一次特征提取所需要的运算量  $p_{\text{normal}}$  如下:

$$p_{\text{normal}} = n \times h' \times w' \times c \times k \times k \quad (1)$$

Ghost 卷积的计算过程如下: 先使用普通卷积生成特征图, 其通道数  $n = c/2$ ; 再对特征图中的通道进行简单的线性运算增加通道数:

$$y_{ij} = \phi_j(y'_i), \forall i = 1, \dots, m, j = 1, \dots, s \quad (2)$$

其中:  $s$  代表线性运算的次数,  $y'_i$  代表第  $i$  个一般特

征图,  $\phi_j$  代表对一般特征图进行第  $j$  个线性变换得到的简单特征图。对  $m$  个特征图进行  $s$  次线性运算可以获得的相似特征图数量为  $m \times s$  个, 为了使输出通道和原输出特征通道相同, 取前  $n$  个相似特征图作为最后的输出结果。

使用 Ghost 卷积进行一次特征提取所需要的运算量  $p_{\text{Ghost}}$  如下:

$$p_{\text{Ghost}} = \frac{n}{s} \times h' \times w' \times k \times k \times c + (s-1) \times \frac{n}{s} \times h' \times w' \times k \times k \quad (3)$$

普通卷积和 Ghost 卷积的参数数量之比和计算量之比见式(4)~(5):

$$r_p = \frac{n \cdot h' \times w' \times c \times k \times k}{\frac{n}{s} \times h' \times w' \times k \times k \times c + (s-1) \times \frac{n}{s} \times h' \times w' \times k \times k} = \frac{c \times k \times k}{\frac{1}{s} \times c \times k \times k + \frac{s-1}{s} \times d \times d} \approx \frac{s \times c}{s+c-1} \approx s \quad (4)$$

$$r_c = \frac{n \times c \times k \times k}{\frac{n}{s} \times c \times k \times k + (s-1) \times \frac{n}{s} \times d \times d} \approx \frac{s \times c}{s+c-1} \approx s \quad (5)$$

其中:  $r_p$  为普通卷积和 Ghost 卷积的参数数量之比,  $r_c$  为普通卷积和 Ghost 卷积的计算量之比。

由于 Ghost 卷积利用了特征图冗余信息, 所以其参数数量和计算量相比普通卷积都有着大幅度下降, 更适合部署在嵌入式设备之中。在 YOLOv5s

的主干特征提取网络中, CBS 模块由普通卷积、批正则化、Silu 激活函数组合而成。将其中的普通卷积模块替换为 Ghost 卷积, 得到由 Ghost 卷积、批正则化、Silu 激活函数组成 GhostCBS 模块。为了避免主干网络特征提取能力的过度下降, 本文保留主干网络中的第一个 CBS 模块, 将其他 CBS 模块替换为 GhostCBS 模块。

由于主干特征提取网络中的 C3 模块包含多个卷积操作, 参数数量和计算量较大, 本文考虑对其进行

轻量化改进。为了在不影响主干网络的特征提取能力,将 C3 模块内 Bottleneck 结构中的卷积操作替换为 Ghost 卷积, C3 模块内其余卷积操作保持不变。将改进后的 C3 模块命名为 C3Ghost 模块。

## 2.2 CBAM 注意力机制模块

CBAM 注意力机制模块包含通道注意力和空间注意力:通道注意力用于增强网络对图像特征的

提取能力,空间注意力用于增强网络对目标的聚焦能力,二者结合可有效淡化无用信息,增强模型鲁棒性和检测准确率。本文采用串联形式的 CBAM 模块,其结构如图 4 所示,其中:图 4(a)是 CBAM 注意力机制的整体结构图,图 4(b)是图 4(a)中通道注意力机制的结构图,图 4(c)是图 4(a)中空间注意力机制的结构图。

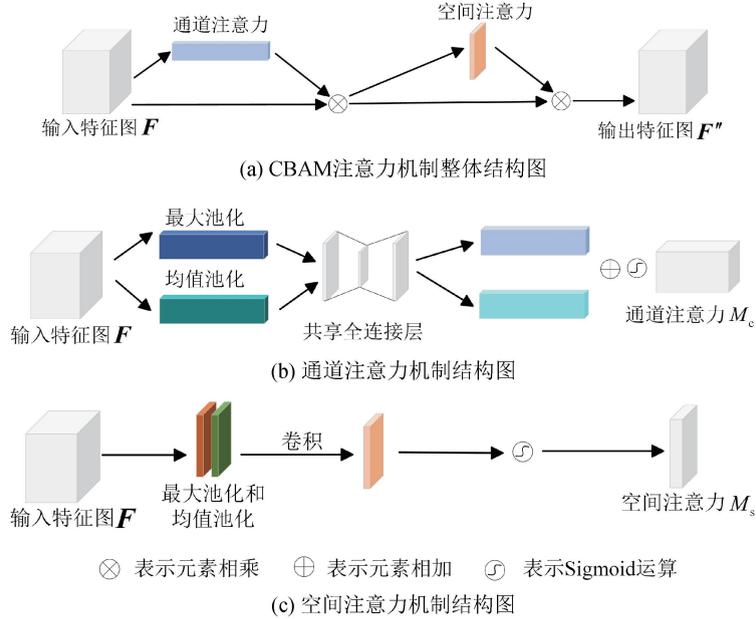


图 4 CBAM 注意力机制结构图

通道注意力模块关注特征图中的重要信息,计算流程如下:

$$\begin{aligned} M_c(\mathbf{F}) &= \sigma(\text{MLP}(\text{AvgPool}(\mathbf{F})) + \text{MLP}(\text{MaxPool}(\mathbf{F}))) \\ &= \sigma(W_1(W_0(\mathbf{F}_{\text{avg}}^c)) + W_1(W_0(\mathbf{F}_{\text{max}}^c))) \end{aligned} \quad (6)$$

其中: $\mathbf{F}$  为原始特征图; $M_c(\mathbf{F})$  为输出的通道注意力值; $\sigma$  为 Sigmoid 函数; $W_0$  和  $W_1$  为权重参数; $\mathbf{F}_{\text{avg}}^c$  为平均池化特征; $\mathbf{F}_{\text{max}}^c$  为最大池化特征。

将通道注意力模块的输出值输入到空间注意力模块之中,空间注意力模块的计算流程如下:

$$\begin{aligned} M_s(\mathbf{F}) &= \sigma(f^{7 \times 7}([\text{AvgPool}(\mathbf{F}); \text{MaxPool}(\mathbf{F})])) \\ &= \sigma(f^{7 \times 7}([\mathbf{F}_{\text{avg}}^s; \mathbf{F}_{\text{max}}^s])) \end{aligned} \quad (7)$$

其中: $M_s(\mathbf{F})$  为输出的空间注意力值; $f^{7 \times 7}$  是卷积核尺寸为  $7 \times 7$  的卷积运算; $\mathbf{F}_{\text{avg}}^s$  为平均池化特征; $\mathbf{F}_{\text{max}}^s$  为最大池化特征。

考虑到注意力机制模块利用了空间和通道维度的信息,在只增加了少量计算量的情况下,可以有效地提高网络的特征提取能力,本文选择将 CBAM 注意力机制模块放在主干特征提取网络的 SPPF 模块之后,利用注意力机制来增强主干特征提取网络的特征提取能力。

## 2.3 基于 Partial Convolution 的改进 Bottleneck 结构

Ghost 模块考虑到特征之间的相关性和冗余性,舍弃了部分常规卷积操作,使用简单操作来获得冗余特征图。受 Ghost 模块的启发,本文引入一种更轻量化的卷积模块 Partial convolution(PCConv)。对于连续或者规则的内容访问,PCConv 将第一个或者最后一个连续的通道视为整个特征图的代表,并只对少数几个通道应用滤波器,其余的通道则保持不变。一般情况而言,输入和输出拥有相同数量的通道。PCConv 只需要在输入通道的一部分上应用常规卷积进行空间特征提取,并保持其余通道不变,即可生成与输入拥有相同通道的输出。因此,PCConv 可以减少冗余计算以及内存访问,它拥有比常规卷积更低的浮点运算数(floating-point operations, FLOPs)以及比深度可分离卷积更高的每秒浮点运算数(floating-point operations per second, FLOPs)。PCConv 和普通卷积、深度可分离卷积的操作示意图如图 5 所示。

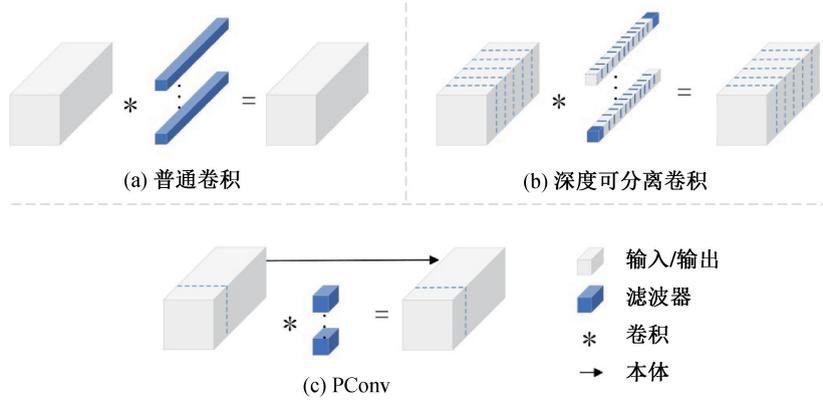


图5 普通卷积、深度可分离卷积和 PConv 操作示意图

在原 YOLOv5s 模型中,C3 模块由三个卷积模块和 Bottleneck 结构组成,其主要功能是增加网络的深度和感受野,通过 Bottleneck 引入残差结构,降低模型的优化难度。Bottleneck 结构的核心思想是使用多个小卷积来替代大的卷积,从而达到降低参数数量的目的。在原始的 C3 模块中,Bottleneck 由  $1 \times 1$  卷积和  $3 \times 3$  卷积组成,通过  $1 \times 1$  卷积来降低通道数,再使用  $3 \times 3$  卷积进行下采样。与此同时,使用 Shortcut 操作将输入和输出连接起来。

PConv 和 GhostConv 相比,参数量、计算消耗更小,但其特征提取能力因此也有所下降。在主干网络中,考虑到模型需要更充分地进行提取特征,故

选择了参数量稍大的 GhostConv。主干网络中使用的 C3Ghost 模块结构示意图如图 6 所示。在特征融合阶段,为了进一步加快模型的推理速度,在保证特征融合能力的情况下利用冗余信息减少计算量,本文选择了更轻量化的 PConv,将 C3 模块内 Bottleneck 结构中的卷积操作替换为 PConv,C3 模块内其余卷积操作保持不变。将改进后的 C3 模块命名为 C3-P,其结构示意图如图 6 所示。由于 Bottleneck 结构是由多个相同模块串行连接而成,PConv 中未参与卷积的通道会在其他的相同模块中参与卷积,从而缓解了单独使用 PConv 所带来的特征提取不充分的现象。

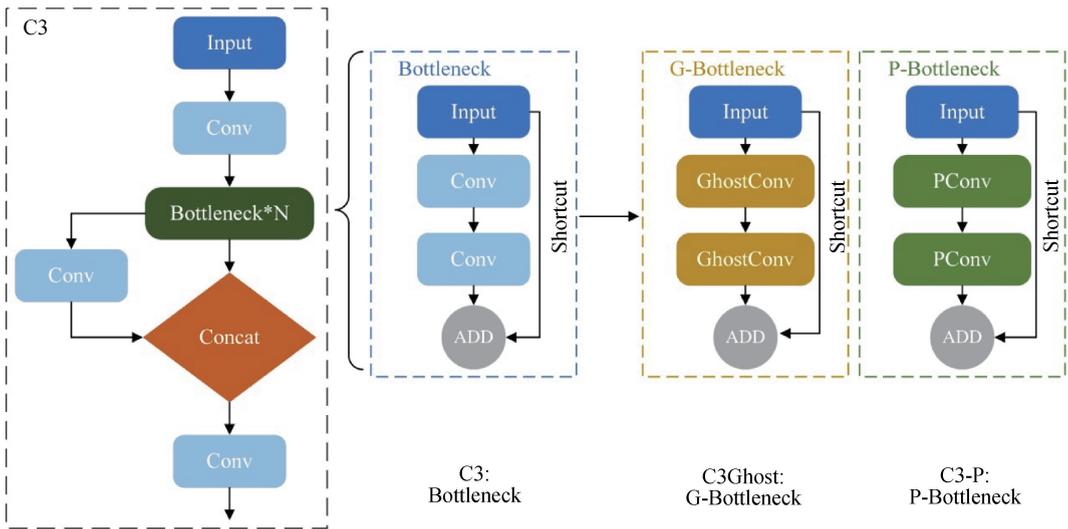


图6 C3 模块、C3Ghost 模块、C3-P 模块结构示意图

### 3 实验结果及分析

本文的实验环境配置如下:操作系统 Windows11(64 位),CPU 11th Gen Intel(R)Core (TM) i5-11400 H,显卡为 RTX3060,运行内存 16 GiB,软件运行环境为 Pycharm。

模型参数设置如下:初始学习率为 0.01,动量

设置为 0.937,使用随机梯度下降法(SGD)进行优化,batch\_size 设置为 16,epoch 设置为 300。

#### 3.1 数据集介绍

本文数据集采集自某纺织生产企业,数据集图像由面阵 CCD 相机拍摄而成。为了实现布匹瑕疵的实时监测,将相机设置于出布口,采集图像为  $3072 \times 96$  像素。由于现场实际采集到的图像为

3072×96 像素的极端尺寸,直接处理会丢失经向瑕疵特征信息,因此将采集到的瑕疵图像切割成 96×96 像素的图像后再用于训练。图像在切割之后使用 LabelImg 标注并制作数据集,数据集中含有毛纬、带纬、白档子、曲纤、双经和断经共 6 种类型的瑕疵图以及正常图像样本。考虑到布匹由经线和纬线两组线组纺织而成,为了进行机器故障的定位,将瑕疵样本按照瑕疵产生的原因分别标注为“jing”和

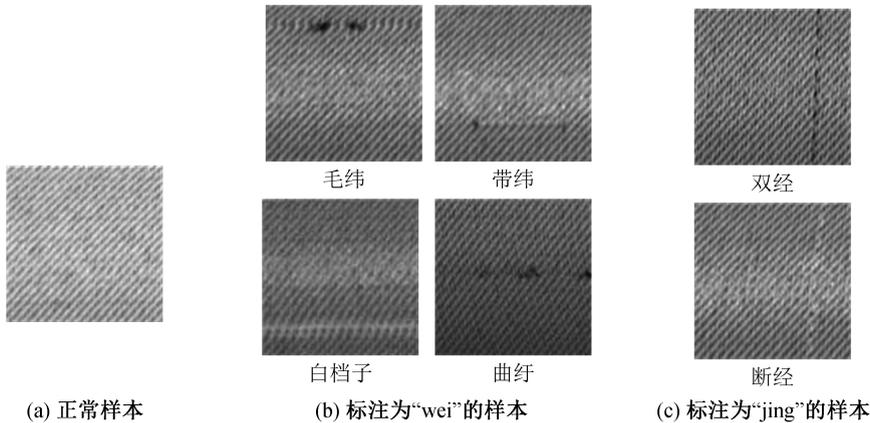


图 7 瑕疵样本及标注类别示例图像

### 3.2 评价指标

本文选用的算法评价指标有参数量、浮点计算量(FLOPs)、查准率(Precision)、查全率(Recall)、平均精度(Average precision, AP)、平均精度均值(Mean average precision, mAP)、每秒传输帧数(Frames per second, FPS)、检出率和误检率,其公式分别如式(8)—(12)所示:

$$P_{\text{mis}} = 1 - P_p = 1 - \frac{T_p}{T_p + F_p} \quad (8)$$

其中:  $P_{\text{mis}}$  为误检率,表示模型检测为正样本但是检测错误的目标数占正样本的比例;  $P_p$  表示查准率,即所有预测为正样本中,实际有多少样本是正样本的比例;  $T_p$  为模型预测正确的正确样本数;  $F_p$  为模型预测错误的正确样本数。

$$P_{\text{true}} = P_r = \frac{T_p}{T_p + F_n} \quad (9)$$

其中:  $P_{\text{true}}$  为检出率,表示模型正确检测出的目标数占总目标数的比例;  $P_r$  表示查全率,即所标注为正样本中被正确检测为正样本的比例;  $F_n$  为模型预测错误的错误样本数。

$$S_{\text{AP}} = \int_0^1 P(R) dR \quad (10)$$

其中:  $S_{\text{AP}}$  为召回率和准确率曲线下的面积。

$$P_{\text{mA}} = \sum_{i=1}^N \frac{S_{\text{AP}i}}{N} \quad (11)$$

“wei”两类,瑕疵样本以及标注类别示例图像如图 7 所示。在经过旋转、噪声、调节亮度等数据增强方法扩充样本数量后,数据集中共含有图像样本 1356 张,其中训练集 1220 张,测试集 136 张。在线检测时,把待检测的采样图像按顺序切分成 32 张 96×96 像素的图像,如果模型的检测速度大于 32 FPS,即可满足布匹瑕疵的实时检测需求。

其中:  $P_{\text{mA}}$  为所有类别的  $S_{\text{AP}}$  均值,  $N$  为检测的总次数。

$$F_{\text{PS}} = \frac{1}{t_{\text{inf}} + t_{\text{NMS}}} \quad (12)$$

其中:  $F_{\text{PS}}$  表示每秒检测帧数,  $t_{\text{inf}}$  表示模型的推理时间,  $t_{\text{NMS}}$  表示模型进行非极大值运算的时间。

### 3.3 实验结果及分析

#### 3.3.1 不同输入尺寸的对比实验

由于数据集中的瑕疵在整幅图像中的占比小,卷积过程中容易造成瑕疵特征信息丢失或失真,因此考虑对图像进行等比例放大再进行输入。本文使用双线性插值的方法进行图像等比放大,再放大图像时,目标像素的值是通过源图像中 4 个最近邻像素点的值进行加权和计算而来。为了探究不同的图像输入尺寸对瑕疵检测造成影响,本文设计了不同输入图像尺寸的检测效果对照实验,结果如表 1 所示,选取每秒检测帧数  $F_{\text{PS}}$ 、 $P_{\text{mis}}$ 、 $P_{\text{true}}$ 、 $P_{\text{mA}}@0.5$ 、 $P_{\text{mA}}@0.50-0.95$  作为评价指标。

根据表 1 实验结果可以看出,当在输入图像尺寸在 640 像素点之内时,检测效果随输入尺寸变大而提高。输入尺寸大于 640 像素点时,检测效果的提升不明显,但是模型的预测时间增加,会导致每秒检测帧数的下降。为了平衡精度与计算成本,本文选择将数据集中的图像等比例放大至 640×640 像素作为模型的输入。

表1 不同尺寸输入图像的实验中本文算法的评价指标

输入图像尺寸/像素	$F_{ps}/\text{Hz}$	$P_{mis}/\%$	$P_{true}/\%$	$P_{mA}@0.5/\%$	$P_{mA}@0.50-0.95/\%$
96×96	133	8.7	89.7	93.8	50.8
256×256	126	4.5	94.0	96.7	60.6
416×416	125	3.6	95.6	96.6	61.9
640×640	116	3.9	95.7	96.6	62.8
800×800	112	3.1	96.1	96.8	62.1

### 3.3.2 CBAM 注意力机制加入位置的对比实验

CBAM 注意力机制模块作为一种即插即用的模块,可以添加在 YOLOv5s 网络中的主干特征提取网络、特征融合网络以及检测头之中。因此,为了

选择 CBAM 注意力机制模块在 YOLOv5s 网络中合适的加入位置,分别选取  $P_{mis}$ 、 $P_{true}$ 、 $P_{mA}@0.5$ 、 $P_{mA}@0.50-0.95$  作为评价指标进行了对照实验,其结果如表 2 所示。

表2 CBAM 注意力机制不同加入位置的实验中本文算法的评价指标

位置	$P_{mis}/\%$	$P_{true}/\%$	$P_{mA}@0.5/\%$	$P_{mA}@0.50-0.95/\%$
未加入	5.7	93.6	<b>96.8</b>	62.1
主干特征提取网络	<b>5.3</b>	94.4	<b>96.8</b>	<b>63.6</b>
特征融合网络	5.5	95.1	95.9	61.7
检测头	5.6	<b>95.2</b>	95.7	62.6

注:加粗字体代表当前评价指标的最好结果。

由表 2 可知,CBAM 注意力机制添加在主干特征提取网络中时,误检率降低、检出率提高,同时检测精度维持不变。当 CBAM 注意力机制添加在特征融合网络以及检测头中时,误检率降低、检出率提高,但是检测精度有所下降。因此,本文选择将 CBAM 注意力机制添加在主干特征提取网络之中。

考虑到轻量化主干网络降低了参数量,可能会弱化特征提取能力,本文选择将 CBAM 注意力机制模块添加到主干特征提取网络的末端,以补偿模型

轻量化可能带来的特征提取能力损失。

### 3.3.3 不同注意力机制的对比实验

为了衡量 CBAM 注意力机制对于主干网络特征提取能力的提升效果,分别选取参数量、 $P_{mis}$ 、 $P_{true}$ 、 $P_{mA}@0.5$ 、 $P_{mA}@0.50-0.95$  作为评价指标,并将 SE 注意力机制(Squeeze-and-excitation attention)、CA 注意力机制(Coordinate attention)、CBAM 注意力机制分别与 GhostConv-YOLOv5s 相结合,检测结果如表 3 所示。

表3 不同注意力机制与轻量化主干网络结合的实验中本文算法的评价指标

检测框架	参数量/MiB	$P_{mis}/\%$	$P_{true}/\%$	$P_{mA}@0.5/\%$	$P_{mA}@0.50-0.95/\%$
GhostConv	5.09	5.7	93.6	<b>96.8</b>	62.1
GhostConv+SE	5.09	5.3	93.7	<b>96.8</b>	63.5
GhostConv+CA	5.10	5.4	94.0	96.7	60.1
GhostConv+CBAM	5.12	5.3	<b>94.4</b>	<b>96.8</b>	<b>63.6</b>

注:加粗字体代表当前评价指标的最好结果。

由表 3 可知,CBAM 和 SE 注意力机制虽然少量增加了模型的参数量,但是提升了主干网络的特征提取能力,查准率、查全率、 $P_{mA}@0.50-0.95$  得到提升,本文选择在主干网络中使用表现更佳的 CBAM 注意力机制。

### 3.3.4 横向对比实验和消融试验

本文在测试样本中进行消融实验和对比实验,进一步测试所提方案的性能,实验结果分别如表 4—表 5。

表 4 给出了在相同测试集下,本文算法同 7 种主流目标检测算法的对比测试结果。通过分析表 4 可知,本文提出的 GCP-YOLOv5s 模型在保证检测

精度的同时,降低了模型的参数量和浮点计算量,提高了每秒检测帧数。

表 5 展示了本文算法的消融实验对比结果。对比 YOLOv5s 和 G-YOLOv5s 可知:在更换主干网络之后,模型的参数量下降,同时  $P_{mA}$  也有所下降。对比 G-YOLOv5s 和 GC-YOLOv5s 可知:在使用了注意力机制后,模型的参数量略微上升, $P_{mA}@0.50-0.95$  提升了 1.5%,在少量增加参数的情况下,使主干网络的特征提取能力得到提升。对比 GC-YOLOv5s 和 GCP-YOLOv5s 可知:在使用 PConv 改进了 C3 模块之后,模型的参数量下降了

19.9%,  $P_{mA}@0.5$  提高了 0.3%,  $P_{mA}@0.50-0.95$  提高了 0.1%, Neck 部分在更少的参数下得到了更强的特征融合能力。对比 YOLOv5s 和 GCP-YOLOv5s 的参数量、FLOPs、 $F_{PS}$ 、 $P_{mA}@0.5$  和  $P_{mA}@0.50-0.95$  可知:本文算法相比基准模型参数

量下降了 41.6%, 计算量下降了 43.1%, 检测速度提高了 12 FPS,  $P_{mA}@0.5$  结果相同,  $P_{mA}@0.50-0.95$  提高了 1.7%。由此可见,本文改进算法在更少参数量、计算量的情况下,实现了更高的检测精度和更快的检测速度。

表 4 自建数据集实验中不同模型的评价指标

算法	参数量/MiB	$P_{mis}/\%$	$P_{true}/\%$	FLOPs/G	$F_{PS}/Hz$	$P_{mA}@0.5/\%$
Faster-RCNN	135.0	12.0	80.6	890.0	33	93.6
SSD	23.0	11.3	85.0	61.5	67	90.2
YOLOv4-tiny	5.7	8.3	90.3	6.9	106	92.3
YOLOv5m	20.8	5.1	<b>96.3</b>	48.2	70	<b>97.6</b>
YOLOv5s	7.0	4.9	95.0	16.0	104	97.1
YOLOv7-tiny	6.0	6.4	89.6	<b>6.0</b>	81	95.2
YOLOXs	9.0	4.2	96.0	26.8	73	97.3
YOLOv8s	11.1	4.8	95.7	28.4	104	96.2
GCP-YOLOv5s(本文)	<b>4.1</b>	<b>3.9</b>	96.2	9.1	<b>116</b>	97.1

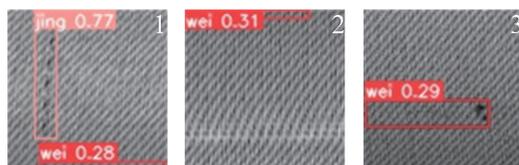
注:加粗字体代表当前评价指标的最好结果。

表 5 消融实验评价指标

算法	轻量化主干	CBAM	C3-P	参数量/MiB	FLOPs/G	$F_{PS}/Hz$	$P_{mA}@0.5/\%$	$P_{mA}@0.50-0.95/\%$
YOLOv5s (baseline)	✘	✘	✘	7.03	16.0	104	97.1	62.6
G-YOLOv5s	✓	✘	✘	5.09	10.7	117	96.8	62.1
GC-YOLOv5s	✓	✓	✘	5.12	10.8	115	96.8	63.6
GCP-YOLOv5s	✓	✓	✓	4.10	9.1	116	97.1	63.7

注:✘表示该实验不包含此模块;✓表示该实验包含此模块。

YOLOv5s(基准模型)和 GCP-YOLOv5s 对实验数据集的检测结果的示例图像如图 8 所示,其评价指标见表 4。对比图 8(a)和图 8(b)的图像 1 可知,基准模型有一处误检,而本文方法未出现误检;对比图 8(a)和图 8(b)的图像 2 可知,基准模型有一处漏检,而本文方法未出现漏检。对比图 8(a)和图 8(b)的图像 3 可知,本文方法的定位比基准模型更准确。从图 8 和表 4 可知,本文提出的改进算法在降低了误检率的同时提高了检测率。



(a) 基准模型的布匹瑕疵检测效果



(b) GCP-YOLOv5s 的布匹瑕疵检测效果

图 8 基准模型和 GCP-YOLOv5s 检测结果示例图像

## 4 结论

本文基于基准模型 YOLOv5s,通过引入 Ghost 卷积重构主干特征提取网络,添加 CBAM 注意力机制增强主干网络特征提取能力,使用 PConv 将 C3 模块重构为 C3-P 模块,设计了一种基于 YOLOv5s 的改进轻量化布匹瑕疵检测算法 GCP-YOLOv5s。在自建数据集的实验结果显示,相较于基准模型 YOLOv5s,本文所提 GCP-YOLOv5s 算法参数量下降了 41.6%,计算量下降了 43.1%,检测速度提高了 12FPS,检测精度提高了 1.7%。相比于其他主流目标检测算法,GCP-YOLOv5s 在检测速度更快、模型参数量更小的同时保持了较高的检测精度。本文所提算法可以满足实际工业现场的应用需求,具有较好的推广性。

## 参考文献:

- [1] 时造雄,茅正冲. 基于改进 Yolov5 的花色布匹瑕疵检测方法[J]. 计算机测量与控制, 2023, 31(4): 56-62.
- [2] 许玉格,钟铭,吴宗泽,等. 基于深度学习的纹理布匹

- 瑕疵检测方法[J]. 自动化学报, 2023, 49(4): 857-871.
- [3] 俞文静, 刘航, 李梓瑞, 等. 基于图像增强和CNN的布匹瑕疵检测算法[J]. 计算机技术与发展, 2021, 31(5): 90-95.
- [4] Jia D Y, Zhou J L, Zhang C W. Detection of cervical cells based on improved SSD network[J]. *Multimedia Tools and Applications*, 2022, 81(10): 13371-13387.
- [5] Diwan T, Anirudh G, Tembhurne J V. Object detection using YOLO: Challenges, architectural successors, datasets and applications [J]. *Multimedia Tools and Applications*, 2023, 82(6): 9243-9275.
- [6] Sharma V K, Mir R N. Saliency guided faster-RCNN (SGFr-RCNN) model for object detection and recognition [J]. *Journal of King Saud University-Computer and Information Sciences*, 2022, 34(5): 1687-1699.
- [7] 蔡兆信, 李瑞新, 戴逸丹, 等. 基于Faster RCNN的布匹瑕疵识别系统[J]. 计算机系统应用, 2021, 30(2): 83-88.
- [8] Zhou S, Zhao J, Shi Y S, et al. Research on improving YOLOv5s algorithm for fabric defect detection [J]. *International Journal of Clothing Science and Technology*, 2023, 35(1): 88-106.
- [9] Liu J H, Wang C Y, Su H, et al. Multistage GAN for fabric defect detection[J]. *IEEE Transactions on Image Processing*, 2019, 29: 3388-3400.
- [10] 李辉, 吕祥聪, 申贝贝, 等. 双路高分辨率转换网络的花色布匹瑕疵检测[J]. 计算机工程与设计, 2023, 44(9): 2731-2739.
- [11] 倪欣, 任佳. 基于高分辨率网络和自注意力机制的歌声分离算法[J]. 浙江理工大学学报(自然科学版), 2022, 47(3): 405-412.
- [12] 黄裕凯, 王青旺, 沈韬, 等. 基于MobileNet的多尺度感受野特征融合算法[J]. 激光与光电子学进展, 2023, 60(4): 280-288.
- [13] 任宇杰, 杨剑, 刘方涛, 等. 基于SSD和MobileNet网络的目标检测方法的研究[J]. 计算机科学与探索, 2019, 13(11): 1881-1893.
- [14] Chen Z C, Yang J, Chen L F, et al. Garbage classification system based on improved ShuffleNet v2 [J]. *Resources, Conservation and Recycling*, 2022, 178: 106090.
- [15] 胡斌汉, 李曙. 基于改进 ShuffleNetV2 的轻量级花色布匹瑕疵检测[J]. 计算机系统应用, 2023, 32(4): 161-169.
- [16] Cao M L, Fu H, Zhu J Y, et al. Lightweight tea bud recognition network integrating GhostNet and YOLOv5 [J]. *Mathematical Biosciences and Engineering: MBE*, 2022, 19(12): 12897-12914.
- [17] Li S J, Sultonov F, Tursunboev J, et al. Ghostformer: A GhostNet-based two-stage transformer for small object detection[J]. *Sensors*, 2022, 22(18): 6939.
- [18] 陈金广, 李雪, 邵景峰, 等. 改进 YOLOv5 网络的轻量级服装目标检测方法[J]. 纺织学报, 2022, 43(10): 155-160.
- [19] 郭波, 吕文涛, 余序宜, 等. 基于改进 YOLOv5 模型的织物疵点检测算法[J]. 浙江理工大学学报(自然科学版), 2022, 47(5): 755-763.
- [20] 雷源毅, 朱文球, 廖欢. 复杂场景下的改进 YOLOv8n 安全帽佩戴检测算法[J]. 软件工程, 2023, 26(12): 46-51.
- [21] 王翔, 任佳. 基于多注意力机制的神经网络故障诊断算法[J]. 浙江理工大学学报(自然科学版), 2020, 43(2): 224-231.

(责任编辑:康 锋)