计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2023,32(8):151−161 [doi: 10.15888/j.cnki.csa.009185]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

改进 YOLOv5 的瓷砖表面缺陷检测①

余松森, 张明威, 杨　欢

(华南师范大学 软件学院, 佛山 528225)
通信作者: 杨　欢, E-mail: YOH16897188@qq.com

摘　要: 现有瓷砖表面缺陷检测存在识别微小目标缺陷能力不足、检测速度有待提升的问题, 为此本文提出了基于改进 YOLOv5 的瓷砖表面缺陷检测方法. 首先, 由于瓷砖表面缺陷尺寸偏小的特性, 对比分析 YOLOv5s 的 3 个目标检测头分支的检测能力, 发现删除大目标检测头, 只保留中目标检测头和小目标检测头的模型检测效果最佳. 其次, 为了进一步实现模型轻量化, 使用 ghost convolution 和 C3Ghost 模块替换 YOLOv5s 在 Backbone 网络中的普通卷积和 C3 模块, 减少模型参数量和计算量. 最后, 在 YOLOv5s 的 Backbone 和 Neck 网络末端添加 coordinate attention 注意力机制模块, 解决原模型无注意力偏好的问题. 该方法在天池瓷砖瑕疵检测数据集上进行实验, 实验结果表明: 改进后的检测模型的平均精度均值达 66%, 相比于原 YOLOv5s 模型提升了 1.8%; 且模型大小只有 10.14 MB, 参数量相比于原模型减少了 48.7%, 计算量减少了 38.7%.

关键词: 机器视觉; 深度学习; 目标检测; YOLOv5 算法; 注意力机制; 瓷砖表面缺陷

引用格式: 余松森,张明威,杨欢.改进 YOLOv5 的瓷砖表面缺陷检测.计算机系统应用,2023,32(8):151–161. http://www.c-s-a.org.cn/1003-3254/9185.html

Detection of Ceramic Tile Surface Defects Based on Improved YOLOv5

YU Song-Sen, ZHANG Ming-Wei, YANG Huan

(School of Software, South China Normal University, Foshan 528225, China)

Abstract: The existing detection method of ceramic tile surface defects has the problem of insufficient ability to identify small target defects, and the detection speed needs to be improved. Therefore, this study proposes a ceramic tile surface defect detection method based on improved YOLOv5. Firstly, due to the small size of ceramic tile surface defects, the detection abilities of three target detection head branches of YOLOv5s are compared and analyzed. It is found that the effectiveness of the model that removes the large target detection head and retains only the medium and small target detection heads is optimal. Secondly, to further realize the lightweight of the model, the study applies ghost convolution and C3Ghost modules to replace the ordinary convolution and C3 modules of YOLOv5s in the Backbone network, thus reducing the number of model parameters and the calculation amount. Finally, the coordinate attention mechanism module is added at the end of the Backbone and Neck networks of YOLOv5s to solve the problem of no attention preference in the original model. The proposed method is tested on the Tianchi ceramic tile defect detection dataset. The results show that the mean precision of the improved detection model averages 66%, which is 1.8% higher than the original YOLOv5s model. Besides, the size of the model is only 10.14 MB, and the number of parameters and the calculation amount is reduced by 48.7% and 38.7% respectively compared with the original model.

Key words: machine vision; deep learning; object detection; YOLOv5 algorithm; attention mechanism; ceramic tile surface defects

① 基金项目: 广东省基础与应用基础研究基金 (2020B1515120089)
收稿时间: 2023-01-19; 修改时间: 2023-02-23; 采用时间: 2023-03-03; csa 在线出版时间: 2023-05-22
CNKI 网络首发时间: 2023-05-24

目前在瓷砖表面缺陷检测环节中, 主要通过质检人员利用人眼主观地去判断瓷砖表面是否存在瑕疵. 质检人员长时间在强光高噪音的环境下工作, 存在检测效率低、人力成本高、检测质量不稳定等问题. 相较于人工检测, 使用机器来实现瓷砖表面缺陷的自动检测也存在很多问题, 主要包括: 1) 瓷砖表面缺陷包含极小缺陷, 可能只有几个像素点, 机器检测出极小目标的能力有限; 2) 瓷砖缺陷包括: 斑点、白点、磕碰、落脏等, 种类繁多; 3) 有些瓷砖表面缺陷特征与瓷砖本身花纹有相似之处. 因此, 瓷砖表面缺陷检测一直是困扰瓷砖行业的痛点, 也是瓷砖行业发展的瓶颈.

很多学者开始使用机器视觉的相关技术去解决瓷砖表面缺陷检测的问题, Sameer Ahamad 等人[1] 对图像进行形态学操作, 并采用模糊规则对瓷砖的缺陷进行分类. Samarawickrama 等人[2] 使用 Matlab 软件对瓷砖表面的角缺陷、边缺陷以及划痕进行检测, 通过计算图像的白色像素和整副图像的白色像素的数值判定区域是否存在缺陷. Alamsyah 等人[3] 使用数字图像处理的方法去检测瓷砖表面缺陷, 首先使用中位滤波器对图像数据进行预处理, 然后基于纹理采用灰度共生矩阵 (GLCM) 方法提取纹理数据, 最后采用最临近节点算法 (KNN) 方法对图像数据进行分类. Birlutiu 等人[4] 使用深度学习的方法对瓷盘进行检测, 网络仅包含 784 个神经元, 准确率却达到 89%. 相比于传统的目标检测方法, 基于深度学习的目标检测方法具有更高的目标检测准确率, 拥有更强的鲁棒性和泛化能力.

近年来, 基于深度学习的目标检测算法在工业领域得到了广泛应用, 它可以迅速准确地定位和分类目标, 主要分为两大类: 第 1 类是两阶段目标检测算法, 该类算法在模型训练时分为 2 个步骤, 第 1 步是训练区域建议网络 (region proposal network), 第 2 步是训练目标区域检测的网络. 在推理过程中, 算法生成一系列作为样本的候选框, 再通过卷积神经网络进行样本分类, 代表模型有 R-CNN[5]、Fast R-CNN[6]、Faster R-CNN[7]、Cascade R-CNN[8]. 另一类目标检测算法是以 YOLO (you only look once)[9] 为代表的一阶段目标检测算法, 该类算法将物体检测任务当成回归问题来处理, 它并不需要区域建议网络, 直接产生物体的类别概率和位置坐标值. 经过单次检测即可直接得到最终的检测结果, 因此它的检测速度非常快, 代表模型还有 SSD (single shot multi-box detector)[10]、YOLOv3[11]、YOLOv4[12] 和 RetinaNet[13].

瓷砖生产过程中对表面缺陷检测的实时性要求非常高, 并且瓷砖表面缺陷只占瓷砖总面积的很小部分, 因此检测模型需要具备检测微小目标缺陷的能力. 目前有学者尝试把 YOLOv5 检测模型应用到该工业场景中, Lu 等人[14] 提出了一种基于 YOLOv5 的瓷砖表面缺陷检测方法, 采用优化瓶颈层和主干网络并添加 SE[15] 注意力机制模块的方法去改进 YOLOv5 模型. 但是该研究所使用的数据集是未公开的, 其改进后的模型参数量和计算量仍然较大. 此外, 该研究没有分析瓷砖表面缺陷偏小的特性并对检测模型进行改进, 且目前有较多比 SE 模块更先进的注意力机制.

针对上述问题, 本文提出一种基于改进 YOLOv5 的瓷砖表面缺陷检测方法, 主要贡献如下: 1) 根据瓷砖表面缺陷的特征, 删除冗余目标检测头分支, 对检测模型结构进行优化. 2) 在网络中使用基于 ghost module[16] 的轻量化模块, 提高模型的检测速度. 3) 添加 CA (coordinate attention)[17] 注意力机制模块, 增强模型提取目标特征的能力. 本文的总体执行流程如图 1 所示, 我们用公开的瓷砖表面缺陷数据集来训练改进后的 YOLOv5s 模型, 得到训练好的检测模型, 把待检测的瓷砖图像输入到训练好的模型进行检测, 最终得到瓷砖图像的检测结果.

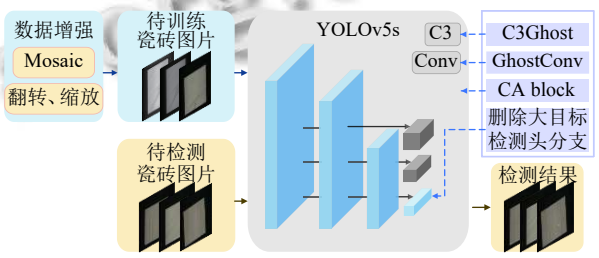


图 1 总体流程图

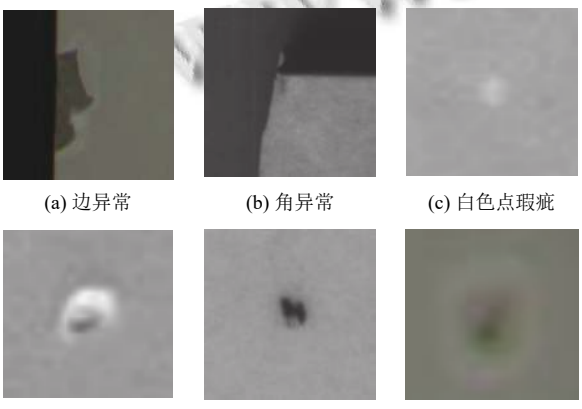
1 瓷砖表面缺陷数据集

1.1 瓷砖表面缺陷类型

本文使用的数据集是天池瓷砖瑕疵检测数据集, 共有 5 388 张图像. 如图 2 所示, 主要分为 6 种缺陷, 分别是边异常、角异常、白色点瑕疵、浅色块瑕疵、深色点块瑕疵和光圈瑕疵. 本文从中随机选取 4 168 张作为训练集, 610 张作为验证集, 610 张作为测试集.

1.2 瓷砖表面缺陷数据集分析

对瓷砖瑕疵检测数据集 (https://tianchi.aliyun.com/dataset/110088) 进行数据分析, 图 3 统计了数据集中各种缺陷类型的样本数量, 发现深色点块瑕疵出现的频率最高, 超过总样本数量的一半. 图 4 为瓷砖表面缺陷的位置分布图, 发现目标缺陷分布较为均匀, 其中边异常和角异常大概率发生在制品的边缘处, 所以四角的缺陷数量会更多. 图 5(a) 为瓷砖表面缺陷的高度尺寸分布直方图, 图 5(b) 为宽度尺寸分布直方图, 发现绝大部分瓷砖表面缺陷高度尺寸要小于原始图像高度尺寸的 0.5%, 宽度尺寸小于原始图像宽度尺寸 2%, 瓷砖表面缺陷只占原始图像中很小一部分面积, 且目标缺陷的长宽尺寸较小, 因此小目标检测是瓷砖表面缺陷检测的重点.



(a) 边异常 (b) 角异常 (c) 白色点瑕疵

(d) 浅色块瑕疵 (e) 深色点块瑕疵 (f) 光圈瑕疵

图 2 瓷砖表面缺陷类型

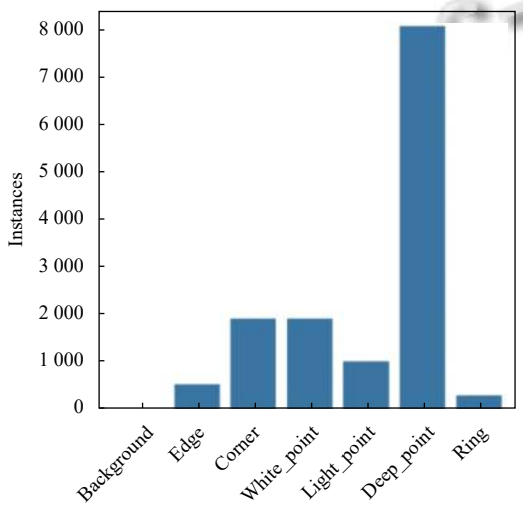


图 3 缺陷数量统计图

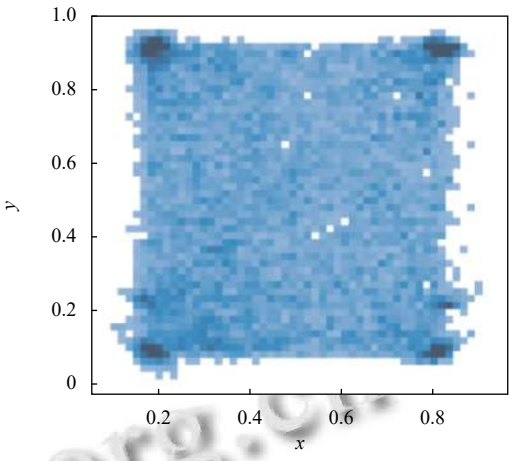
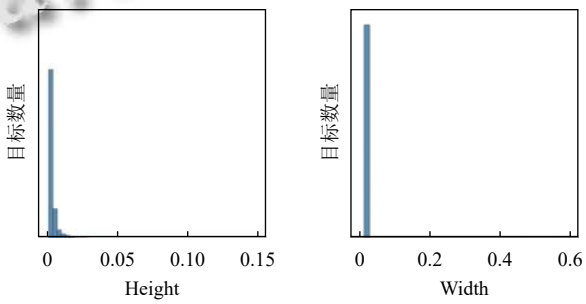


图 4 缺陷位置分布图



(a) 高度尺寸分布直方图 (b) 宽度尺寸分布直方图

图 5 缺陷尺寸分布直方图

2 YOLOv5 检测算法及改进

2.1 YOLOv5 算法

YOLOv5s 的整体网络结构如图 6 所示, 可以分为 Backbone、Neck 和 Prediction 这 3 个部分. 其中 Backbone 网络主要由 Focus、CBS、C3 和 SPPF 模块组成, 主要功能是提取输入图像的信息. 在 Backbone 网络最后加入 SPPF (spatial pyramid pooling-fast) 模块, 能显著增加了感受野, 分离出最重要的上下文特征, 对比普通的 SPP[18] 方法, SPPF 以更少的计算量和更快的速度就能产生与 SPP 一样的效果. Neck 网络借鉴了 FPN+PANet[19] 网络结构, 它的主要功能是实现 Backbone 网络中不同层级之间的信息交互. 在 Prediction 网络, YOLOv5s 有 3 个目标检测头分支, 分别对小目标、中目标和大目标进行目标检测, 本文输入网络的图像尺寸为 2816×2816×3, 在 Prediction 网络会使用 352×352、176×176 和 88×88 的网格分别检测小目标、中目标和大目标. YOLOv5使用 CIoU loss[20] 作为边界框坐标回归的损失函数, 解决了 IoU 和 GIoU[21] 存在的收敛速度慢和回归不精确等问题.

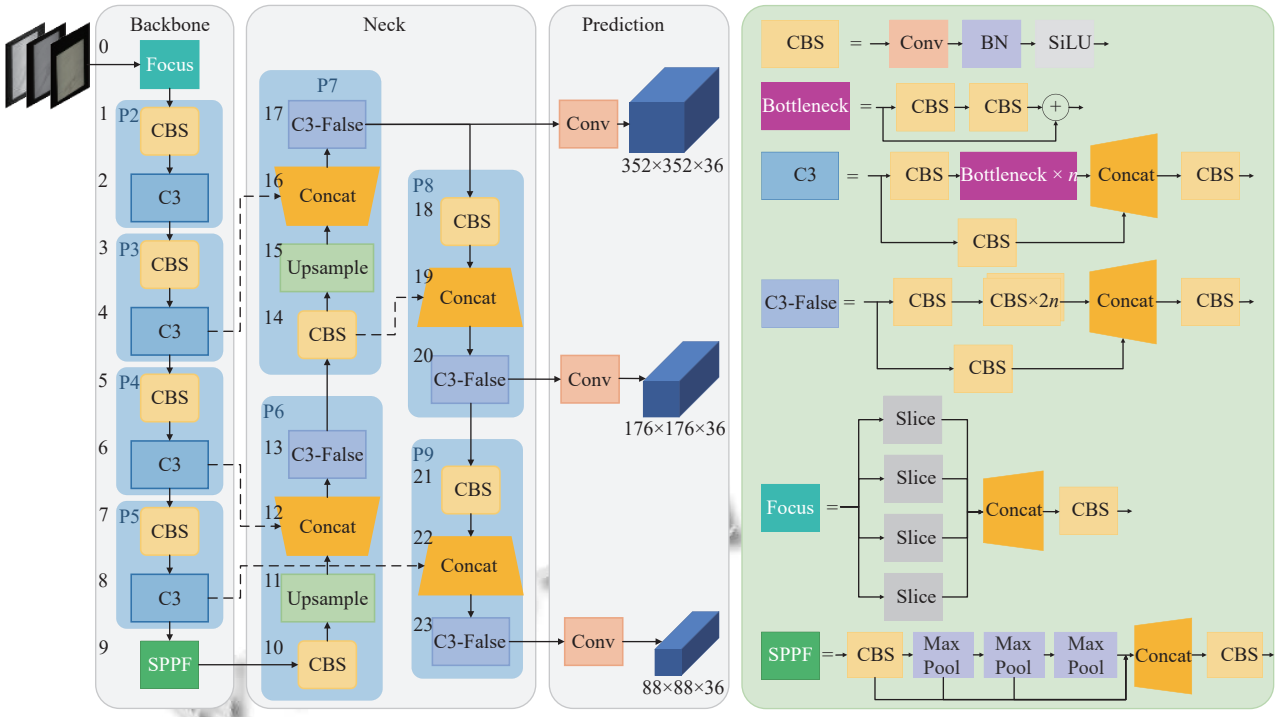


图 6 YOLOv5s 的网络结构

2.2 YOLOv5 算法改进

为了解决瓷砖表面缺陷检测存在的小目标检测困难、实时性要求高的问题, 本文提出的适用于瓷砖表面缺陷检测的改进型 YOLOv5s 模型. 如图 7 所示, 改进点主要分为 3 个部分: 1) 模型结构改进: 删除原模型中专门用来检测大目标的冗余检测头分支; 2) 模型轻量化改进: 使用轻量化模块 ghost convolution 和 C3Ghost 替换原模型 Backbone 网络中的 C3 和普通卷积模块; 3) 添加注意力机制: 为了让模型更关注重要的区域, 在 Backbone 和 Neck 网络末端添加 CA (coordinate attention) block.

2.2.1 模型结构改进

YOLOv5s 在 Prediction 网络有 3 个目标检测头分支, 它们的感受野各不相同. 其中大目标检测头所使用网格大小为 88×88, 每个网格的长宽尺寸为输入图像的长宽尺寸的 1.14%. 但是数据集中绝大多数缺陷的高度尺寸要小于原始图像高度尺寸的 0.5%, 因此大目标检测头分支检测瓷砖表面缺陷的能力会较弱.

图 8 为 YOLOv5s 在不同阶段生成的特征图, 可以发现在 stage 17 和 stage 20 生成的特征图中目标缺陷的特征信息最清晰, 但是从 stage 23 生成的特征图可以发现, 缺陷的特征信息开始变得模糊, 部分特征信息已经消失.

图 9 为 YOLOv5s 的结构简图, 由于瓷砖表面缺陷的尺寸很小, P9 部分输出的特征图是专门用来检测大尺寸目标, 因此该部分的作用有限, 改进后的 YOLOv5s 模型删除了原模型 P9 部分. 图 10 为删除大目标检测头后的模型, 模型命名为 YOLOv5s-de, 该模型可以减少原始模型的参数量和计算量, 提升模型的检测速度.

为了对比拥有不同数量目标检测头分支的 YOLOv5s 模型, 本文在 Prediction 网络添加了一个专门针对微小目标的检测头, 如图 11 所示, 在 Neck 网络中采用 FPN+PANet[19] 的网络结构, 添加了 A1 和 A2 部分, A1 与 P2 输出的特征信息拼接后直接对微小目标进行检测, 本文把添加微小目标检测头的模型命名为 YOLOv5s-ad.

图 8(e) 为 A1 部分输出的特征图, 可以发现, 在 YOLOv5s 模型的基础上, 所添加的微小目标缺陷检测头并不能有效提取目标缺陷的特征信息, 检测效果较差. 主要原因是所使用的模型是 YOLOv5s, 属于 YOLOv5 系列中较为轻量的模型, 输入的图像信息只经过 Backbone 网络中 P2 部分处理后, 就与 Neck 网络 A1 部分输入的信息进行拼接, 且 P2 部分中的 C3 模块只有一层 bottleneck, 提取特征信息的能力较弱, 因此所添加的微小目标检测头的检测能力有限.

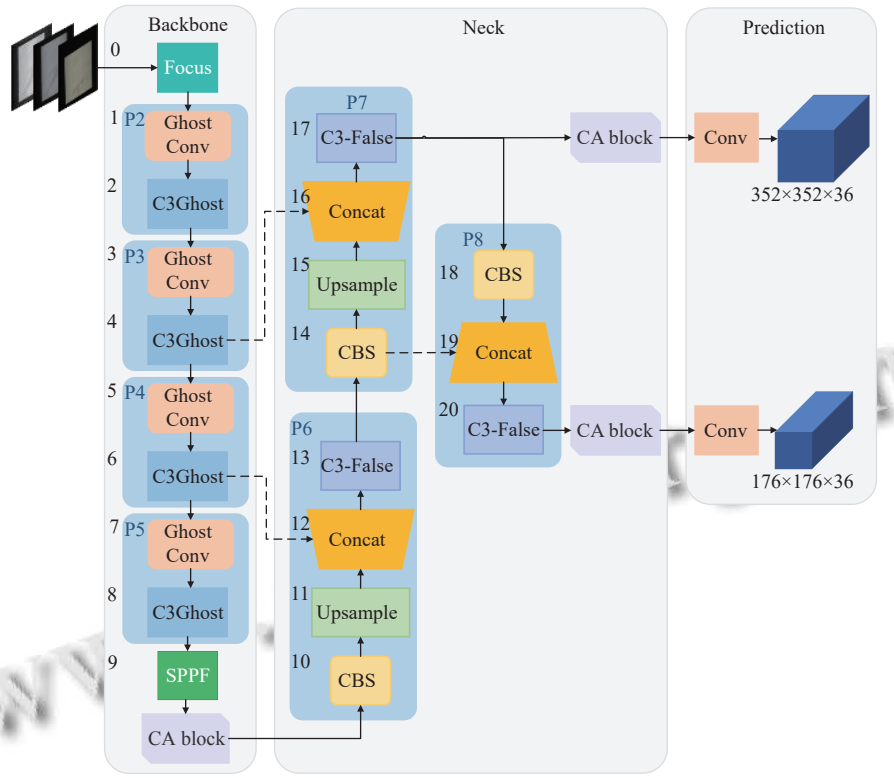
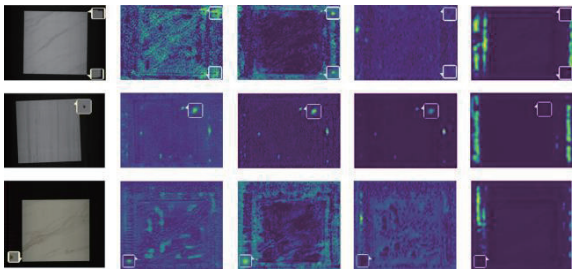


图 7 改进后的 YOLOv5s 网络结构



(a) 原始图 (b) Stage 17 (c) Stage 20 (d) Stage 23 (e) A1

图 8 YOLOv5s 各阶段生成的特征图

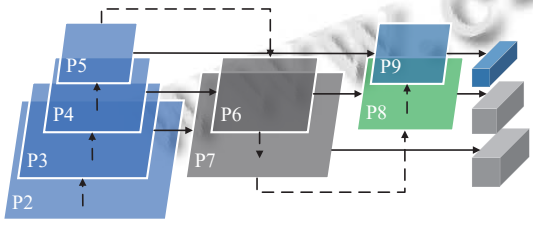


图 9 原模型结构简图

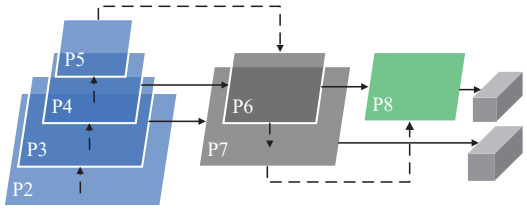


图 10 YOLOv5s-de 结构简图

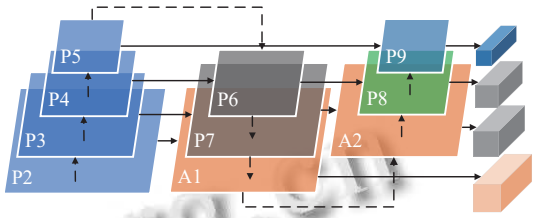


图 11 YOLOv5s-ad 结构简图

2.2.2 模型轻量化改进

本文对 YOLOv5s 模型进行轻量化改进思路是使用基于 ghost module[16] 的 ghost convolution 和 C3Ghost 去分别替换原模型 Backbone 网络中的普通卷积和 C3 模块.

假设输入特征是$X \in \mathbb{R}^{c\times h\times w}$, 卷积核为$f \in \mathbb{R}^{c\times k\times k\times n}$, 输出特征是 $Y \in \mathbb{R}^{h'\times w'\times n}$, 其中$c$、$h$、$w$分别表示输入数据的通道数、高度和宽度, $k\times k$表示f的内核大小, h'、w'、n分别表示输出特征图的高度、宽度和通道数, 则普通卷积所需要的 FLOPs 数量为$n\cdot h'\cdot w'\cdot c\cdot k\cdot k$. 但是普通卷积输出特征里面有大量冗余特征, 没有必要消耗大量的计算量和参数去生成这些冗余信息.

为解决上述问题, ghost module 将卷积操作分为 2 个部分: 第 1 部分涉及普通卷积, 但是输出只有m层

原始特征图. 第 2 部分应用一系列简单的线性操作来生成更多的特征映射, 根据式 (1) 对每个原始特征进行廉价的线性映射, 生成s个 ghost 特征映射图.

$$y_{ij}=\Phi_{i,j}(y'_i),\ \forall i=1,\cdots,m,\ \ j=1,\cdots,s \tag{1}$$

其中, y'_i表示第i个原始特征图, 它经过线性变换$\Phi_{i,j}$后得到第j个 ghost 特征映射图y_{ij}. Ghost module 相比于普通卷积的理论加速比为:

$$\begin{aligned} r_s&=\frac{n\cdot h'\cdot w'\cdot c\cdot k\cdot k}{\frac{n}{s}\cdot h'\cdot w'\cdot c\cdot k\cdot k+(s-1)\cdot\frac{n}{s}\cdot h'\cdot w'\cdot d\cdot d}\\ &=\frac{c\cdot k\cdot k}{\frac{1}{s}\cdot c\cdot k\cdot k+\frac{s-1}{s}\cdot d\cdot d}\approx\frac{s\cdot c}{s+c-1}\approx s \end{aligned} \tag{2}$$

其中, $d\times d$表示每个线性运算的平均内核大小, $d\times d$大小与$k\times k$相近并且$s\ll c$; 相似地, 压缩比可以计算为:

$$r_c=\frac{n\cdot c\cdot k\cdot k}{\frac{n}{s}\cdot c\cdot k\cdot k+(s-1)\cdot\frac{n}{s}\cdot d\cdot d}\approx\frac{s\cdot c}{s+c-1}\approx s \tag{3}$$

与普通的卷积神经网络相比, ghost module 总体所需的参数量和计算量都更少. 另外, ghost module 尝试利用简单的线性操作获得更多的相似特征图, 这些 ghost 特征映射可以充分揭示内在特征的信息, 从而增强模型的特征提取能力. 虽然生成的 ghost 特征图来源于原始特征图, 但他们之间存在着显著差异, 这意味着生成的特征图足够灵活, 能够满足特定任务的需求.

如图 12 所示, ghost convolution 的第 1 个卷积是 CBS 模块, 但输出特征图的通道数只有普通 CBS 模块输出通道数的一半. 第 2 个卷积是深度可分离卷积[22], 使用 DWConv 表示, 它的卷积核大小为 5×5. 相较于普通卷积, ghost convolution 减少了非关键特征的学习成本, 通过组合少量卷积核与更廉价的线性操作去代替常规卷积方式, 能有效降低模型对计算资源需求.

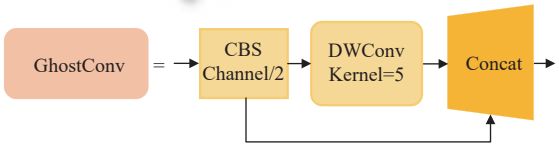


图 12 Ghost convolution 模块结构

C3Ghost 是在 C3 模块中使用 ghost bottleneck 作为瓶颈层的网络结构, 其中, ghost bottleneck 把 ghost module 引用到瓶颈层模块中. 如图 13 所示, ghost bottleneck 有 stride=1 和 stride=2 两种情况, 第 1 个 ghost module 是一个增加通道数量的扩展层, 将输出通道数与输入通道数之间的比率称为扩展比率; 第 2 个 ghost module 减少了与 shortcut 路径相匹配通道的数量, 并使用 shortcut 方法融合这 2 个 ghost module 的输入和输出. 在实际应用中, ghost module 的卷积主要是使用逐点卷积来完成, 可以有效提高模型的效率.

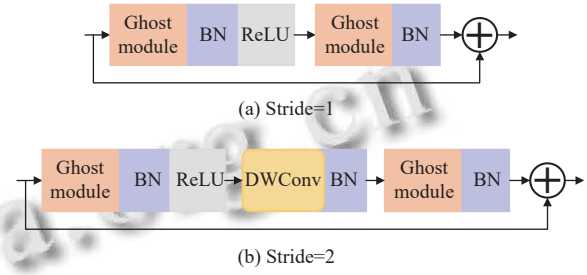


图 13 Ghost bottleneck 模块结构

2.2.3 添加注意力机制

注意力模块可以使用全局信息来有选择地强调信息特征, 关注信息量丰富的通道特征, 并抑制不重要的通道特征. 本文在改进模型的基础上对比了两种不同的注意力机制模块, 分别是 SE block[15] 和 CA block[17].

CA block 网络结构如图 14 所示, 相比于 SE block, 它除了考虑通道间的关系外, 还考虑了特征空间的位置信息, 在通道注意力中结合水平方向信息和垂直方向信息, 这能帮助模型更加精确地定位和识别感兴趣的目标.

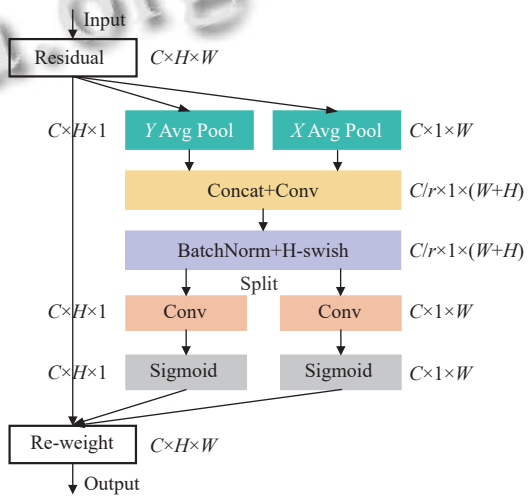


图 14 Coordinate attention block 结构

首先对输入的特征信息x_c分别使用尺寸为 (H, 1) 和 (1, W) 的池化核进行编码, 相比于 SE block 的二维

全局池化, CA block 能够捕获具有精确位置信息的空间长程依赖. 通道数为 c 且在高度 h 处的输出表示为:

$$z_c^h(h)=\frac{1}{W}\sum_{0\leqslant i<W}x_c(h,i) \tag{4}$$

相似地, 通道数为 c 且在宽度 w 处的输出表示为:

$$z_c^w(w)=\frac{1}{H}\sum_{0\leqslant j<H}x_c(j,w) \tag{5}$$

然后对式 (4) 和式 (5) 输出的特征图进行拼接操作, 如式 (6) 所示, 送入到共享的 1×1 卷积变换函数F_1, 对特征信息进行批量归一化并使用非线性函数激活, 得到特征图$f\in\mathbb{R}^{C/r\times(H+W)}$, 其中, r 表示通道下采样比率.

$$f=\delta\left(F_1\left(\left[z^h,z^w\right]\right)\right) \tag{6}$$

接下来将特征图f沿着空间维度切分成两个独立的张量$f^h\in\mathbb{R}^{C/r\times H}$和$f^w\in\mathbb{R}^{C/r\times W}$, 如式 (7) 和式 (8) 所示, 利用两个 1×1 卷积变换F_h和F_w分别对特征图f^h和f^w进行转换, 转换后特征图的通道数和输入相同. 然后使用 Sigmoid 激活函数进行计算, 得到水平方向和垂直方向的注意力权重.

$$g^h=\sigma\left(F_h\left(f^h\right)\right) \tag{7}$$

$$g^w=\sigma\left(F_w\left(f^w\right)\right) \tag{8}$$

最后用注意力权重乘以原始输入的特征图, 得到注意力模块最终输出的特征信息:

$$y_c(i,j)=x_c(i,j)\times g_c^h(i)\times g_c^w(j) \tag{9}$$

3 实验结果分析

3.1 实验环境及评价指标

本文训练和测试所使用的 GPU 为 Nvidia RTX 3090, 操作系统为 Ubuntu, 版本为 20.04, 采用 PyTorch 深度学习框架, 版本号为 1.9.0, CUDA 版本为 11.2, cuDNN 版本 8.0.5. 训练的初始学习率为 0.01, 最大迭代次数为 400, batch size 设置为 4.

本文从平均精度均值 (mean average precision, *mAP*)、参数量、计算量、模型权重 4 个评价指标去衡量模型的性能. 如式 (10)–式 (12) 所示, 平均精度均值 (*mAP*) 是模型检测精度的评价指标, 它与精确率 (*precision*) 和召回率 (*recall*) 相关, *AP*(*c*) 表示类别 *c* 的平均精确率, *N*(*classes*) 表示多目标分类任务中类别的个数, 某个类别的平均精确率 (*AP*) 是指 P-R 曲线和坐标轴所围区域的面积. 进行多标签图像分类任务时, 把每一个类别的 *AP* 值相加后除以类别数量, 最终得到所有类别的平均精度均值.

$$precision=\frac{TP}{TP+FP} \tag{10}$$

$$recall=\frac{TP}{TP+FN} \tag{11}$$

$$mAP=\frac{\sum_{c\in classes}AP(c)}{N(classes)} \tag{12}$$

3.2 实验结果对比分析

3.2.1 YOLOv5 不同复杂度模型性能对比

YOLOv5 应用了类似 EfficientNet[23] 中的 channel 和 layer 控制因子, 提供了 5 种不同复杂度的版本. 由于瓷砖表面缺陷检测对实时性要求很高, 检测模型需要具备更快的检测速度, 因此本文对复杂度较小的前 3 种模型作对比, 分别是 YOLOv5n、YOLOv5s 和 YOLOv5m. 实验结果如表 1 所示, YOLOv5s 能够保证在较高平均精度均值的前提下, 所需要耗费的计算量较少, 检测时间较短, 在检测速度和检测精度两个方面更为均衡, 因此后面的模型均是在 YOLOv5s 的基础上进行改进.

表 1 YOLOv5 不同复杂度模型性能对比

Model	$mAP_{0.5}$ (%)	Params (M)	GFLOPs	Weights (MB)	Speed (ms)
YOLOv5n	60.9	1.8	81.4	6.79	12.4
YOLOv5s	64.2	7.0	307.8	16.86	25.0
YOLOv5m	66.3	20.9	931.5	43.37	61.8

3.2.2 模型结构改进分析

由于瓷砖表面缺陷尺寸偏小的特性, 本文提出了在 YOLOv5s 基础上删除大目标检测头的 YOLOv5s-de 模型. 为了确定最佳的检测头分支数量, 本文还构建了添加微小目标检测头的 YOLOv5s-ad 模型.

实验结果如表 2 所示, 可以发现 YOLOv5s-de 模型在平均精度均值、模型参数量、计算量、模型权重以及检测速度方面均是最优, 相比于 YOLOv5s 模型, $mAP_{0.5}$ 提升 0.6%, 参数量减少 25.7%, 计算量减少 9%. 而 YOLOv5s-ad 模型检测效果很差. 因此, 通过对比拥有不同数量的目标检测头的模型, 发现在 YOLOv5s 模型的基础上只保留小目标检测头和中目标检测头是模型结构改进的最佳策略.

表 2 拥有不同数量目标检测头的模型实验结果对比

Model	$mAP_{0.5}$ (%)	Params (M)	GFLOPs	Weights (MB)	Speed (ms)
YOLOv5s	64.2	7.0	307.8	16.86	25.0
YOLOv5s-de	**64.8**	**5.2**	**280.1**	**13.20**	**23.1**
YOLOv5s-ad	60.1	7.7	450.1	28.10	36.7

3.2.3 消融实验

以 YOLOv5s-de 为基准模型进行消融实验, 实验主要包括以下 2 个部分: 第 1 部分是对比 3 种不同的轻量化模块, 包括深度可分离卷积, ghost convolution 以及 C3Ghost. 第 2 部分对比两种不同的注意力机制模块, 包括 SE block 和 CA block.

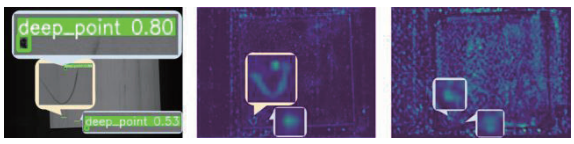
从表 3 可以发现使用 ghost convolution 和 C3Ghost 模块可以显著减少原模型的参数量和计算量, 使用后模型的平均精度均值、参数量和计算量均要优于基准模型, 其中 $mAP_{0.5}$ 提升了 0.1%, 参数量和计算量均减少了 36%. 在主干网络使用 ghost module 替代卷积层, 使用廉价的线性操作即可生成大量的特征映射, 帮助模型以更少的参数实现了更高的识别性能.

表 3 消融实验

DWConv	GhostConv	C3Ghost	SE block	CA block	$mAP_{0.5}$ (%)	Params (M)	GFLOPs
√	—	—	—	—	63.9	3.7	208.1
—	√	—	—	—	64.6	4.4	245.0
—	—	√	—	—	63.4	4.1	214.2
√	—	√	—	—	63.7	2.5	142.2
—	√	√	—	—	64.9	3.3	179.2
√	—	√	√	—	64.0	2.7	147.0
√	—	√	—	√	64.3	2.8	151.8
—	√	√	√	—	65.4	3.5	183.9
—	√	√	—	√	66.0	3.6	188.7

使用 ghost convolution 和 C3Ghost 两种轻量化模块, 再选择添加 CA block 后的模型平均精度均值最高, 达到 66%, 相比于原 YOLOv5s 模型, $mAP_{0.5}$ 提升了 1.8%, 参数量减少了 48.7%, 计算量减少了 38.7%, 在参数量和计算量少于原模型的基础上, 实现了更高的平均精度均值, 更快的检测速度.

对比图 15 和图 16 的检测结果可以发现, 添加 CA block 的模型对目标缺陷的预测置信度要高于添加 SE block 的模型. 对比图 15(b) 和图 16(b) 在 stage 17 输出的特征信息可以发现, 对于一些干扰信息, 如图 15(a) 中出现的用黑色油性笔所做的标记, 添加 CA block 的检测模型对这些干扰信息的抑制效果更好. 对比图 15(c) 和图 16(c) 在 stage 20 输出的特征信息可以发现, 添加 CA block 的检测模型提取目标缺陷特征信息的能力要明显强于添加 SE block 的模型.



(a) 检测结果 (b) Stage 17 (c) Stage 20

图 15 添加 SE block 后模型的检测结果及生成的特征图

(a) 检测结果 (b) Stage 17 (c) Stage 20

图 16 添加 CA block 后模型的检测结果及生成的特征图

3.2.4 不同模型性能对比

本文对比不同的目标检测模型, 包括 Faster R-CNN[7]、Cascade R-CNN[8]、RetinaNet[13] 和 VarifocalNet[24], 它们所使用的主干网络结构均为 ResNet-101[25]. 对比的模型还包括 YOLOv3-tiny 和 YOLOX-s 模型, 它们分别是 YOLOv3[11] 和 YOLOX[26]模型的轻量化版本, 输入的图像尺寸大小均为 2816×2816×3.

从表 4 可以看出 YOLOv5s 模型对比其他目标检测模型, 平均精度均值更高, 并且模型的参数量和计算量更少. 在所有对比的模型中, 改进后的模型 YOLOv5s-Ours 在平均精度均值、参数量、计算量和模型权重方面均为最优. YOLOv5s-Ours 在模型参数量和计算量最少的基础上, 实现了最高的检测精度.

3.2.5 不同类型缺陷识别性能对比

瓷砖表面缺陷检测的重点和难点是检测出微小目

标缺陷, 表 5 对比了不同模型检测各类型缺陷的平均精确率, 可以发现对于出现频率极高的小目标缺陷如深色点块瑕疵、浅色点瑕疵以及白色点瑕疵, YOLOv5s-Ours 的平均精确率远远高于原 YOLOv5s 模型, 对小目标缺陷拥有更出色的检测能力.

表 4 不同目标检测模型性能对比

Model	$mAP_{0.5}$ (%)	Params (M)	GFLOPs	Weights (MB)
Faster R-CNN	31.0	60.15	2095.7	460.44
Cascade R-CNN	34.7	87.94	2123.5	672.49
RetinaNet	48.1	55.22	2191.2	422.87
VarifocalNet	43.3	53.54	1394.6	410.16
YOLOX-s	50.1	8.94	515.9	68.52
YOLOv3-tiny	63.6	8.68	250.9	21.10
YOLOv5s	64.2	7.04	307.8	16.86
YOLOv5s-Ours	**66.0**	**3.61**	**188.7**	**10.19**

对比图 17 中 YOLOv5s 和 YOLOv5s-Ours 的混淆矩阵可以发现, 改进后的模型错误分类更少, 对所有缺陷类型都实现了更精准的分类, 尤其是对小目标缺陷的分类能力要强于 YOLOv5s 模型.

3.2.6 不同模型实测性能对比

针对测试数据, 图 18 展示了不同目标检测模型的检测结果, 从实验结果可以看出 YOLOv5s-Ours 对目标缺陷的预测置信度要高于原 YOLOv5s 模型, 并且 YOLOv5s-Ours 对比其他检测模型, 能够检测出更多的目标缺陷, 改进后的模型对小目标缺陷检测效果优异, 适用于检测瓷砖表面缺陷这种类型的微小目标. 并且改进后模型参数量和计算量最少, 能够很好地满足瓷砖表面缺陷检测对实时性的要求.

表 5 各种缺陷类型的平均精确率

Category	Number	$AP_{0.5}$ (%)	
		YOLOv5s	YOLOv5s-Ours
Deep point	802	0.574	**0.633**
Light point	102	0.312	**0.424**
White point	271	0.418	**0.465**
Corner	241	0.869	**0.870**
Edge	54	**0.885**	0.830
Ring	48	**0.792**	0.739

4 结论与展望

本文提出了一种基于改进的 YOLOv5 的瓷砖表面缺陷检测方法, 通过删除冗余检测头分支以及使用 ghost convolution 和 C3Ghost 来实现模型的轻量化, 并且添加 CA 注意力机制模块, 提升模型关注重要区域信息的能力. 改进后的模型在平均精度均值、参数量、计算量以及模型权重方面均优于 YOLOv5s 和其他对比的目标检测模型, 对小目标缺陷拥有出色的检测效果. 在后续的工作中, 会将改进后的模型部署在嵌入式设备上, 应用到实际瓷砖表面缺陷检测环节中, 实现应用落地.

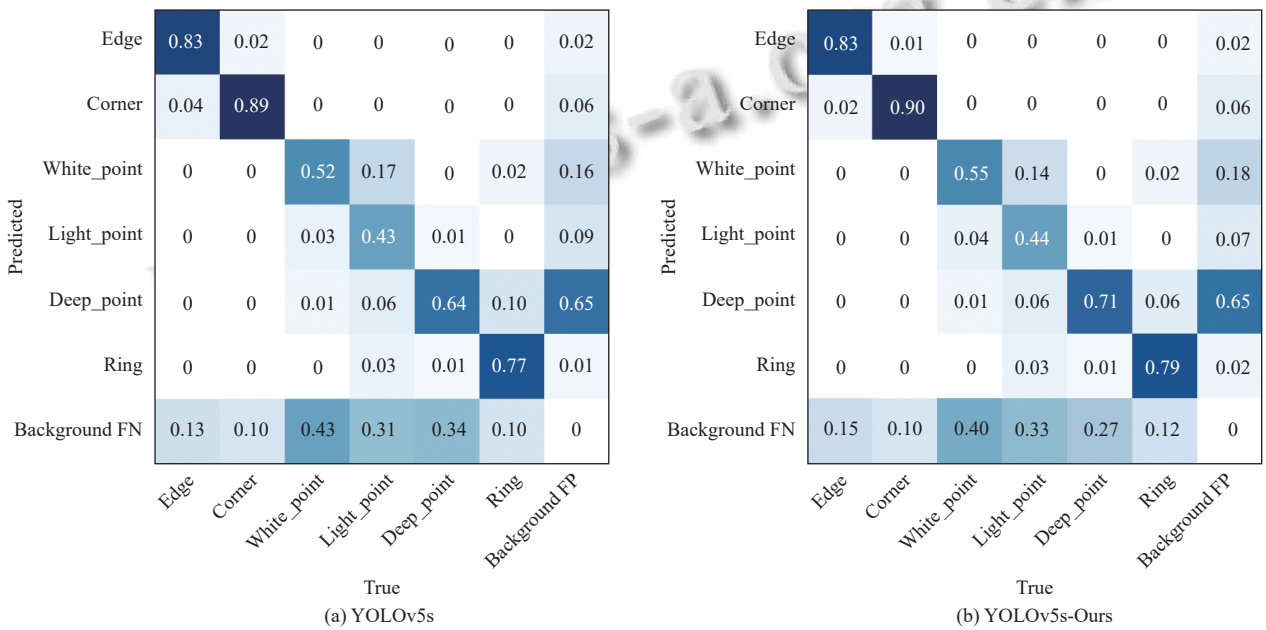
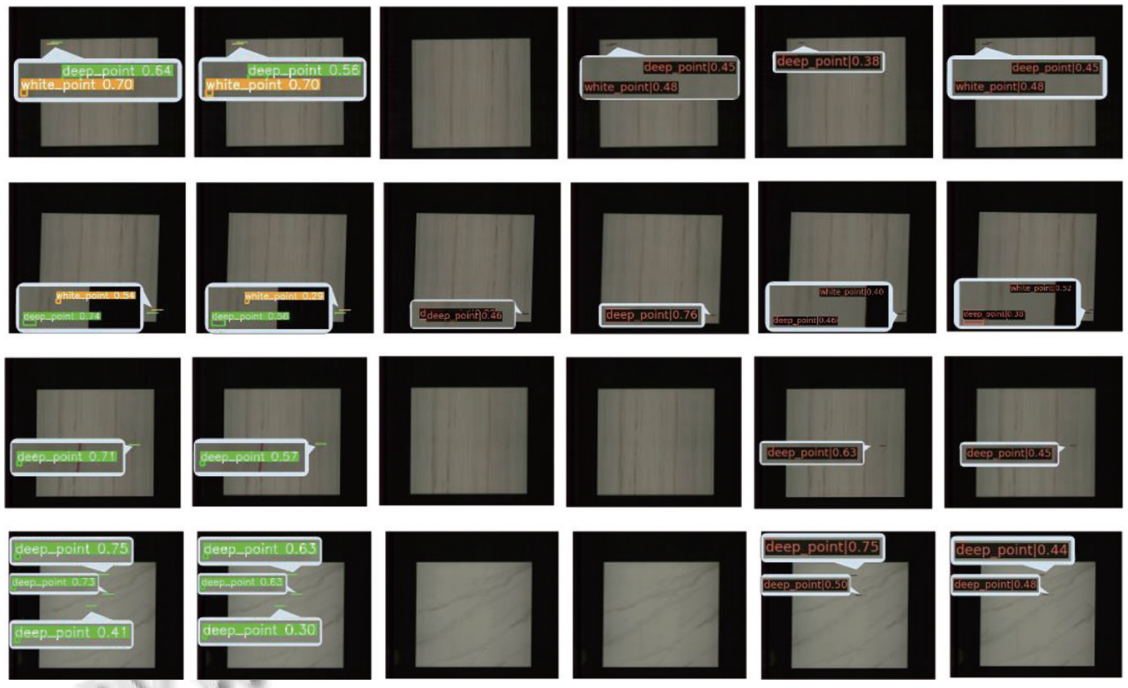


图 17 混淆矩阵



(a)YOLOv5s-Ours (b) YOLOv5s (c) Faster R-CNN (d) Cascade R-CNN (e) RetinaNet (f) VarifocalNet

图 18 检测结果

参考文献

1 Sameer Ahamad N, Bhaskara Rao J. Analysis and detection of surface defects in ceramic tile using image processing techniques. Microelectronics, Electromagnetics and Telecommunications. New Delhi: Springer, 2016. 575–582.

2 Samarawickrama YC, Wickramasinghe CD. Matlab based automated surface defect detection system for ceremic tiles using image processing. Proceedings of the 6th National Conference on Technology and Management (NCTM). Malabe: IEEE, 2017. 34–39.

3 Alamsyah R, Wiranata AD, Rafie. Defect detection of ceramic tiles using median filtering, morphological techniques, gray level co-occurrence matrix and K-nearest neighbor method. Scientific Research Journal, 2019, 7(4): 41–46.

4 Birlutiu A, Burlacu A, Kadar M, *et al*. Defect detection in porcelain industry based on deep learning techniques. Proceedings of the 19th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC). Timisoara: IEEE, 2017. 263–270.

5 Girshick R, Donahue J, Darrell T, *et al*. Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. Columbus: IEEE, 2014. 580–587.

6 Girshick R. Fast R-CNN. Proceedings of the 2015 IEEE International Conference on Computer Vision. Santiago: IEEE, 2015. 1440–1448.

7 Ren SQ, He KM, Girshick R, *et al*. Faster R-CNN: Towards real-time object detection with region proposal networks. Proceedings of the 28th International Conference on Neural Information Processing Systems. Montreal: MIT Press, 2015. 91–99.

8 Cai ZW, Vasconcelos N. Cascade R-CNN: Delving into high quality object detection. Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 6154–6162.

9 Redmon J, Divvala S, Girshick R, *et al*. You only look once: Unified, real-time object detection. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 779–788.

10 Liu W, Anguelov D, Erhan D, *et al*. SSD: Single shot multibox detector. Proceedings of the 14th European Conference on Computer Vision. Amsterdam: Springer, 2016. 21–37.

11 Redmon J, Farhadi A. YOLOv3: An incremental improvement. arXiv:1804.02767, 2018.

12 Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal speed and accuracy of object detection. arXiv:2004.10934, 2020.

13 Lin TY, Goyal P, Girshick R, *et al*. Focal loss for dense object detection. Proceedings of the 2017 IEEE International Conference on Computer Vision. Venice: IEEE, 2017. 2980–2988.

14 Lu QH, Lin JM, Luo LF, *et al*. A supervised approach for automated surface defect detection in ceramic tile quality control. Advanced Engineering Informatics, 2022, 53: 101692. [doi: 10.1016/j.aei.2022.101692]

15 Hu J, Shen L, Sun G. Squeeze-and-excitation networks. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 7132–7141.

16 Han K, Wang YH, Tian Q, *et al*. GhostNet: More features from cheap operations. Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle: IEEE, 2020. 1577–1586.

17 Hou QB, Zhou DQ, Feng JS. Coordinate attention for efficient mobile network design. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 13708–13717.

18 He KM, Zhang XY, Ren SQ, *et al*. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904–1916. [doi: 10.1109/TPAMI.2015.2389824]

19 Liu S, Qi L, Qin HF, *et al*. Path aggregation network for instance segmentation. Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, 2018. 8759–8768.

20 Zheng ZH, Wang P, Liu W, *et al*. Distance-IoU loss: Faster and better learning for bounding box regression. Proceedings of the 34th AAAI Conference on Artificial Intelligence, AAAI 2020, the 32nd Innovative Applications of Artificial Intelligence Conference, IAAI 2020, the 10th AAAI Symposium on Educational Advances in Artificial Intelligence. New York: AAAI Press, 2020: 12993–13000.

21 Rezatofighi H, Tsoi N, Gwak JY, *et al*. Generalized intersection over union: A metric and a loss for bounding box regression. Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach: IEEE, 2019. 658–666.

22 Chollet F. Xception: Deep learning with depthwise separable convolutions. Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition. Honolulu: IEEE, 2017. 1800–1807.

23 Tan MX, Le QV. EfficientNet: Rethinking model scaling for convolutional neural networks. Proceedings of the 36th International Conference on Machine Learning. Long Beach: PMLR, 2019. 6105–6114.

24 Zhang HY, Wang Y, Dayoub F, *et al*. VarifocalNet: An IoU-aware dense object detector. Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville: IEEE, 2021. 8510–8519.

25 He KM, Zhang XY, Ren SQ, *et al*. Deep residual learning for image recognition. Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 770–778.

26 Ge Z, Liu ST, Wang F, *et al*. YOLOX: Exceeding YOLO series in 2021. arXiv:2107.08430, 2021.

(校对责编: 牛欣悦)