

基于改进 ShuffleNetV2 的轻量级花色布匹瑕疵检测^①



胡斌汉¹, 李 曙^{1,2}

¹(吉首大学 通信与电子工程学院, 吉首 416000)

²(电子科技大学 信息与通信工程学院, 成都 611731)

通信作者: 李 曙, E-mail: shuli@jsu.edu.cn

摘 要: 布匹瑕疵检测是纺织业质量管理的重要环节. 在嵌入式设备上实现准确、快速的布匹瑕疵检测能有效降低成本, 因而价值巨大. 考虑到实际生产中花色布匹瑕疵具有背景复杂、数量差异大、极端长宽比和小瑕疵占比高等结构特性, 提出一种基于轻量级模型的花色布匹瑕疵检测方法并将其部署在嵌入式设备 Raspberry Pi 4B 上. 首先在一阶段目标检测网络 YOLO 的基础上用轻量级特征提取网络 ShuffleNetV2 提取花色布匹瑕疵的特征, 以减少网络结构复杂度及参数量, 提升检测速度; 其次是检测头的解耦合, 将分类与定位任务分离, 以提升模型收敛速度; 此外引入 CIoU 作为瑕疵位置回归损失函数, 提高瑕疵定位准确性. 实验结果表明, 本文算法在 Raspberry Pi 4B 上可达 8.6 FPS 的检测速度, 可满足纺织工业应用需求.

关键词: 布匹瑕疵检测; 轻量级模型; Raspberry Pi 4B; YOLO; ShuffleNetV2

引用格式: 胡斌汉, 李曙. 基于改进 ShuffleNetV2 的轻量级花色布匹瑕疵检测. 计算机系统应用, 2023, 32(4): 161-169. <http://www.c-s-a.org.cn/1003-3254/9051.html>

Lightweight Colored Fabric Defect Detection Based on Improved ShuffleNetV2

HU Bin-Han¹, LI Shu^{1,2}

¹(School of Communication and Electronic Engineering, Jishou University, Jishou 416000, China)

²(School of Information and Communication Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract: Fabric defect detection is an important link in the quality management of the textile industry. Accurate and fast fabric defect detection on embedded devices can effectively reduce the detection cost, thus being of great value. Considering the structural characteristics of colored fabric defects in actual production, such as a complex background, large differences in the quantity of defects, an extremely high aspect ratio, and a high proportion of small defects, a colored fabric defect detection method based on a lightweight model is proposed and implemented on an embedded circuit board Raspberry Pi 4B. The lightweight feature extraction network ShuffleNetV2 is first used to extract the features of colored fabric defects on the basis of the one-stage target detection network, you only look once (YOLO), so as to reduce the complexity of the network structure and the number of parameters and improve the detection speed. Then, the detection head is decoupled to separate the classification and localization tasks so that the convergence speed of the model can be improved. In addition, the complete intersection over union (CIoU) is introduced as the loss function of defect location regression to improve the accuracy of defect location. The experimental results show that the proposed algorithm can achieve a detection speed of 8.6 FPS on Raspberry Pi 4B, which can meet the needs of the textile industry.

Key words: fabric defect detection; lightweight model; Raspberry Pi 4B; you only look once (YOLO); ShuffleNetV2

① 基金项目: 国家自然科学基金 (42164006); 中国博士后基金面上项目 (2021M700682); 湖南省自然科学基金 (2022JJ30474); 湖南省教育厅优秀青年项目 (21B0507); 国家级大学生创新创业训练计划 (201910531053); 吉首大学校级科研项目 (Jdy21061)

收稿时间: 2022-09-20; 修改时间: 2022-10-19; 采用时间: 2022-11-04; csa 在线出版时间: 2023-01-06

CNKI 网络首发时间: 2023-01-06

布匹瑕疵检测是纺织业在布匹质量管理的重要一环,布匹表面的瑕疵严重影响着其美观和质量,导致布匹产品滞销和经济损失^[1]。目前,大多数纺织业仍采用人工检测的方法,该方法存在成本高、效率低、会导致误检和漏检等问题^[2]。部分纺织业使用传统机器视觉方法等自动化方法实现瑕疵检测,但传统机器视觉方法存在检测瑕疵类型单一、计算成本高、复杂花色背景检测效果差等缺陷,不能满足检测要求。压缩生产成本、提高生产效率是当前纺织业日益迫切的需求。研究实现智能化、轻量化的花色布匹瑕疵检测方法具有重要的实际意义。

传统机器视觉方法根据布匹图像纹理特征分为基于统计学方法、基于频谱方法、基于模型方法。统计学方法量化分析布匹图像数据,提取纹理信息进行统计处理,常用方法有:直方图统计法^[3]、灰度共生矩阵法^[4]和数学形态学^[5]等。该类方法适用于无复杂背景图案、分辨率较小的布匹图像,存在检测布匹类型单一、瑕疵定位不准确等缺陷。频谱法针对布匹图像频谱特性进行分析,实现瑕疵分类,方法有傅里叶变换^[6]、小波变换^[7]和 Gabor 滤波^[8]等。频谱法对规律性纹理的布匹图像检测效果较好,但难以在空间上对瑕疵位置进行定位。模型法将正常布匹图像作为分布模型,将检测图像输入模型进行匹配判别瑕疵位置,方法有自回归模型^[9]和马尔科夫随机场模型^[10]。该类方法对表面纹理单一的图像检测效果较好,对于纹理复杂、不清晰图像的检测存在局限性。综上所述,传统方法对花色布匹瑕疵检测有一定借鉴作用,但存在计算剪度高、收敛速度慢、参数调节困难及检测精度低等缺点。同时,传统方法通常依赖人工经验调参实现,生产过程中难以实现实时检测,不能满足纺织业日益增长的工业化生产需求。

随着深度学习技术的发展,布匹瑕疵检测朝着自动化、智能化的方向发展。深度学习算法对复杂花色背景下的布匹瑕疵有优异的检测性能,可实现传统方法难以达到的检测效果。基于深度学习的目标检测算法按候选区域的处理方式不同分为两大类:第1类是基于 region-convolutional neural networks (R-CNN) 系列算法^[11-13]的两阶段目标检测网络。如基于 Faster R-CNN^[14]的布匹瑕疵检测系统针对布匹数据集中类别数量不均衡、瑕疵目标小的特点,对 region proposal network (RPN) 网络进行改进,能准确标识瑕疵点的位置。

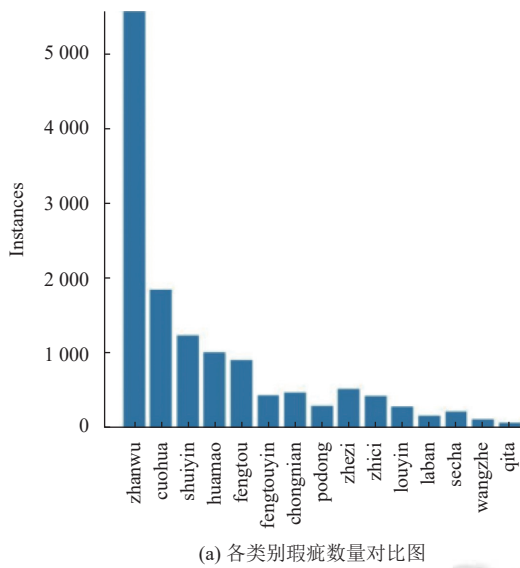
两阶段检测网络能够有效对高分辨率、小瑕疵的布匹图像进行检测,但在检测速度上不理想。第2类是基于 you only look once (YOLO) 系列算法^[15-18]的单阶段端到端目标检测网络。如 Jing 等^[19]提出一种基于改进 YOLOv3 的实时性能高的织物瑕疵检测方法,将低层特征与高层信息相结合应用于不同特征层进行布匹瑕疵检测,该方法主要适用于色调单一的坯布与有规律性背景的格子布。Zheng 等^[20]提出了一种基于 squeeze-and-excitation (SE) 模块的 SE-YOLOv5 高效布匹瑕疵检测算法,可以满足布匹瑕疵检测的需要,但该算法难以在嵌入式设备上实现检测。

尽管基于深度学习的方法在布匹瑕疵检测中取得了较好的检测效果,但此类方法计算剪剪度高、剪剪量大,使其难以在嵌入式设备上实现剪剪精度、剪剪实时性等检测要求。近年来,随着移动及嵌入式设备对深度学习算法的需求增大,基于轻量级网络结构的目标检测方法的研究增多,各种轻量级检测网络被提出,如 MobileNet^[21,22]、ShuffleNet^[23,24]及 GhostNet 网络^[25]等。此类轻量级检测网络可在嵌入式设备上实现智能化花色布匹瑕疵检测,以此为基础,本文提出了一种基于改进 ShuffleNetV2 的轻量级花色布匹瑕疵检测方法,旨在有效降低纺织业检测成本,在嵌入式设备上实现准确、快速的花色布匹瑕疵检测。

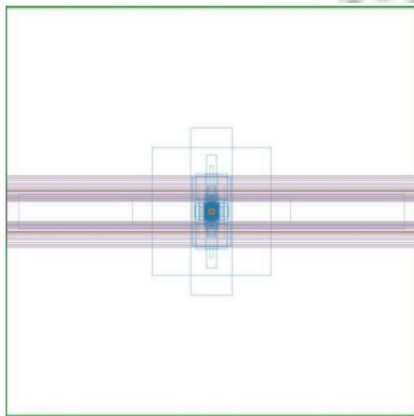
1 花色布匹瑕疵数据集构建与数据增强处理

花色布匹瑕疵数据集来源于天池工业智创新大赛^[26],包含 4354 张瑕疵图像,16457 个瑕疵标注,标注为每个瑕疵的类别及边界框。瑕疵类别为实际生产中常见的 15 类瑕疵,如缝头、错花和沾污等。布匹图像的像素大小为 4096×1800。本文随机抽取 90% 作为训练集,10% 为测试集,其中训练集中 20% 为验证集。各别瑕疵数量对比图与标注框中心化显示图如图 1。

从图 1 中可得出以下信息:(1) 不同类别瑕疵之间数量差异大;(2) 瑕疵尺寸差异大,小瑕疵剪剪比高;(3) 长宽比大。这些都是目标检测领域中的重点、难点所在。为降低不同类别瑕疵之间数量差异大对检测效果的影响,本文在训练过程中,采取随机翻转、变形缩放等方式对数据集进行增强处理。实验中为减少模型计算量及加快收敛速度,将图像统一缩小为 416×416 像素。花色布匹瑕疵数据集中部分瑕疵样本如图 2。



(a) 各类别瑕疵数量对比图



(b) 标注框中心化显示图

图1 各类别瑕疵数量对比图与标注框中心化显示图

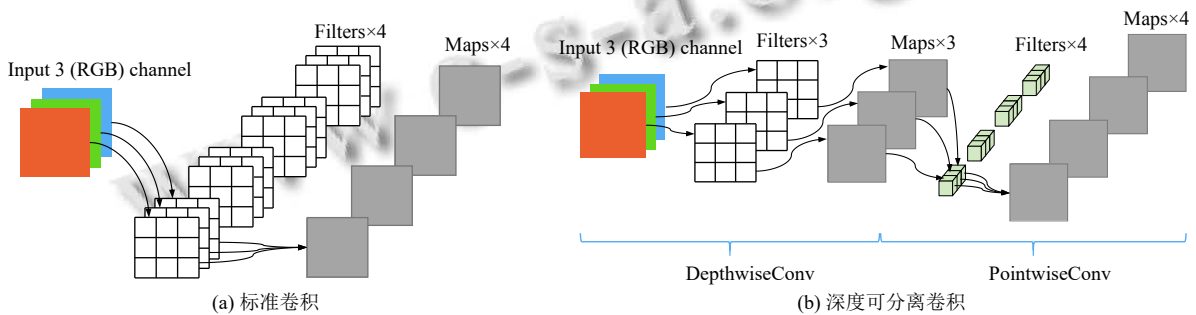


图3 标准卷积与深度可分离卷积

从图3(a)可以看出,对于宽、高和通道数分别为 W 、 H 和 C 的输入图像,得到大小为 $w \times h \times c$ 的特征图 (feature map), 标准卷积参数量 (parameters, P) 为:

$$P_{std} = D_k \times D_k \times C \times c \quad (1)$$

其中, $D_k \times D_k$ 为卷积核的大小, C 为输入特征图通道数,

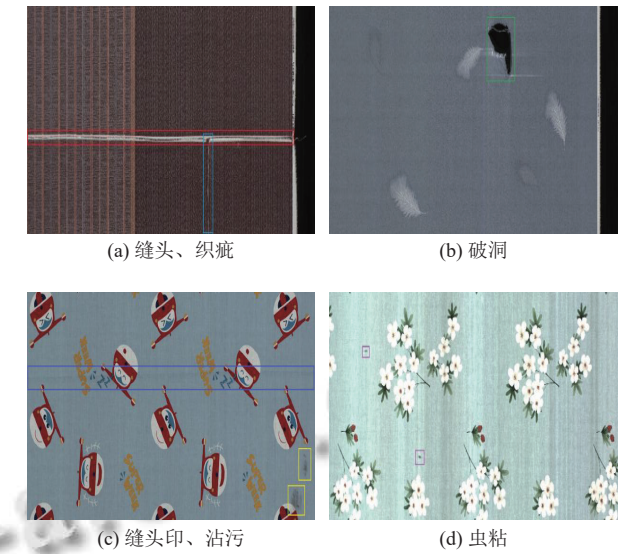


图2 花色布匹瑕疵数据集部分瑕疵样本

2 花色布匹瑕疵检测网络结构设计

2.1 深度可分离卷积

传统卷积神经网络内存占用大、运算量大导致在嵌入式设备上运行困难. 深度可分离卷积 (depthwise separable convolution, DepthSepConv) 可大大减少卷积运算复杂度与参数量. 该卷积由深度卷积 (3×3 DepthwiseConv) 和点卷积 (1×1 PointwiseConv) 组成, 其首次在 MobileNetV1^[21] 中被提出, 并逐渐成为在边缘设备上运行轻量化网络的可靠组件. 标准卷积 (standard convolution) 与深度可分离卷积如图3所示.

c 为输出特征图通道数, 也为卷积核的数量. 标准卷积的计算量 (FLOPs, F) 为:

$$F_{std} = D_k \times D_k \times C \times c \times W \times H \quad (2)$$

深度可分离卷积中每个输入特征图通道只与一个卷积核进行运算, 得到中间特征图后与点卷积进行加

权混合生成新的特征图. 深度可分离卷积的参数量为:

$$P_{ds} = D_k \times D_k \times C + C \times c \quad (3)$$

深度可分离卷积的计算量为:

$$F_{ds} = D_k \times D_k \times C \times W \times H + C \times c \times W \times H \quad (4)$$

可得深度可分离卷积与标准卷积的参数量和计算量之比为:

$$\frac{F_{ds}}{F_{std}} = \frac{P_{ds}}{P_{std}} = \frac{1}{c} + \frac{1}{D_k^2} \quad (5)$$

从深度可分离卷积与标准卷积的参数量和计算量之比可以看出, 在获得与标准卷积计算相同效果时, 无论是参数量还是计算量上比标准卷积少许多倍, 极大减少了模型参数量与计算量. ShuffleNetV2 中特征提取基本组件 basic block 使用深度可分离卷积. 在 Raspberry Pi 4B 等嵌入式开发板上实现花色布匹瑕疵检测算法的难点之一是模型结构的优化, 而深度可分离卷积则更加适合在嵌入式设备上部署运行.

2.2 ShuffleNetV2 网络模型

为降低网络模型计算量, 提高推理速度和检测精度, ShuffleNetV2 以 ShuffleNetV1 为基础, 其保留了 ShuffleNetV1 中通道混洗和深度可分离卷积等优秀组件, 减少了模型参数量与计算量. 同时 ShuffleNetV2 进一步分析了影响模型推理速度的原因, 充分考虑内存访问成本 (memory access cost, MAC) 与网络并行度 (degree of parallelism, DP) 这两个重要指标, 提出了更为高效的基本单元 (basic block), 如图 4 所示.

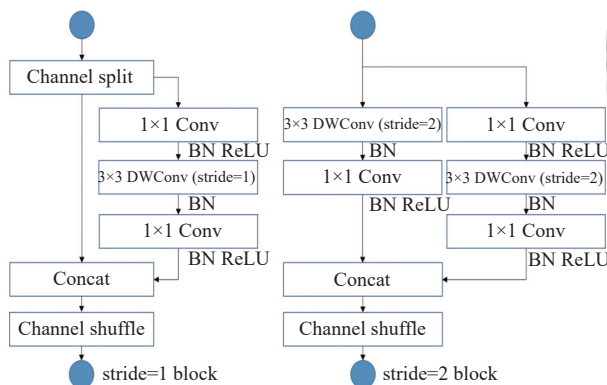


图 4 基本单元

ShuffleNetV2 网络结构如图 5 所示.

从图 5 中可以看出, ShuffleNetV2 网络结构中 Stage2、Stage3、Stage4 和 Conv5 的输出通道数 (output channels) 依次增大, 分为 4 个版本, 分别是 0.5×、1×、

1.5×和 2×. 随着版本增大, 基本单元不变, 组件 Stage 结构的输出通道数依次增加. 网络深度的增加使得模型的特征提取能力逐渐增强, 检测精度提高, 但随之而来的是模型的浮点运算量 (FLOPs) 和模型大小的增大. 由于 Raspberry Pi 4B 嵌入式设备算力条件有限, 为保证花色布匹瑕疵检测有更好的检测速度, 本文选择计算量及参数量较低的 0.5×版本作为本文算法的主干网络 (Backbone).

2.3 基于改进 ShuffleNetV2 的花色布匹瑕疵检测网络

ShuffleNetV2 0.5×的网络结构如图 5 所示. 输入图像分辨率大小为 3×224×224, 先经过 24 个大小为 3×3、步数 (stride) 为 2 的标准卷积进行特征提取, 得到 24×112×112 的特征图 (feature map). 然后使用 MaxPool 做最大池化处理达到下采样的目的, 得到 24×56×56 的特征图后连接 3 个由 ShuffleNetV2 网络基本组件 stride=1 block 和 stride=2 block 构成的 Stage 结构, 提取目标图像不同层级的语义信息. Stage2、Stage3 和 Stage4 结构输出特征图的通道数大小分别为 48、96 和 192, 它们中的 block2 和 block1 的重复次数分别为 1:3、1:7 和 1:3, 之后由 1024 个 1×1 的卷积扩充特征提取通道, 进一步提升模型对目标的特征提取. 最后由全局池化层充分融合空间信息, 防止模型过拟合, 使用全连接层 (fully connected layers, FC) 进行目标分类.

本文在以 ShuffleNetV2 0.5×作为花色布匹瑕疵特征提取网络的基础上, 有效融合浅层与深层特征信息, 达到可在嵌入式设备上实现的花色布匹瑕疵检测模型, 所提出的基于改进 ShuffleNetV2 的花色布匹瑕疵检测算法整体网络结构如图 6 所示.

本文检测算法整体网络结构为端到端一阶网络 YOLO 架构, 分为输入端 (input)、主干网络 (backbone)、颈部网络 (neck) 和检测输出 (prediction). 输入端为 416×416 像素大小、3 通道 (RGB) 的花色布匹图像. 主干网络又称为特征提取网络, 本文由 ShuffleNetV2 网络的前大部分结构构成, 得益于其优异的功能组件, 使在嵌入式设备上实现花色布匹瑕疵检测成为可能. 颈部网络又称为特征融合网络, 本文将深层 Stage4 后的特征图用于小布匹瑕疵的检测. 将深层 Stage4 后的特征图经上采样后与 Stage3 拼接, 将中层特征与深层特征融合用于中等大小布匹瑕疵的检测. 同样也将 Stage3 结构提取的特征信息与 Stage2 的特征信息进行融合, 这样可以更好融合浅层的表征位置信息与深层语义信息从而提升模型的检测能力.

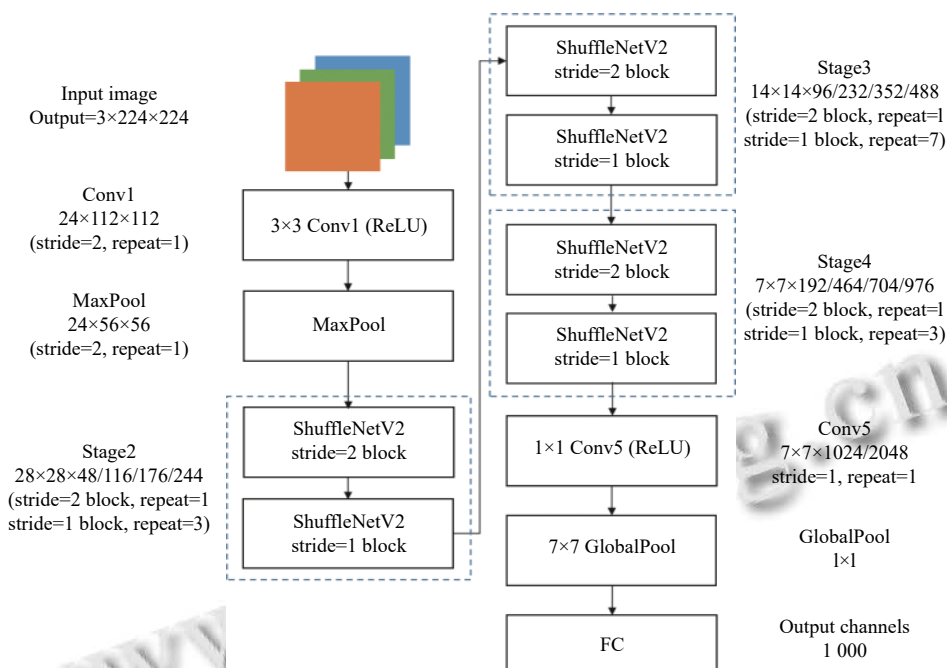


图5 ShuffleNetV2网络结构图

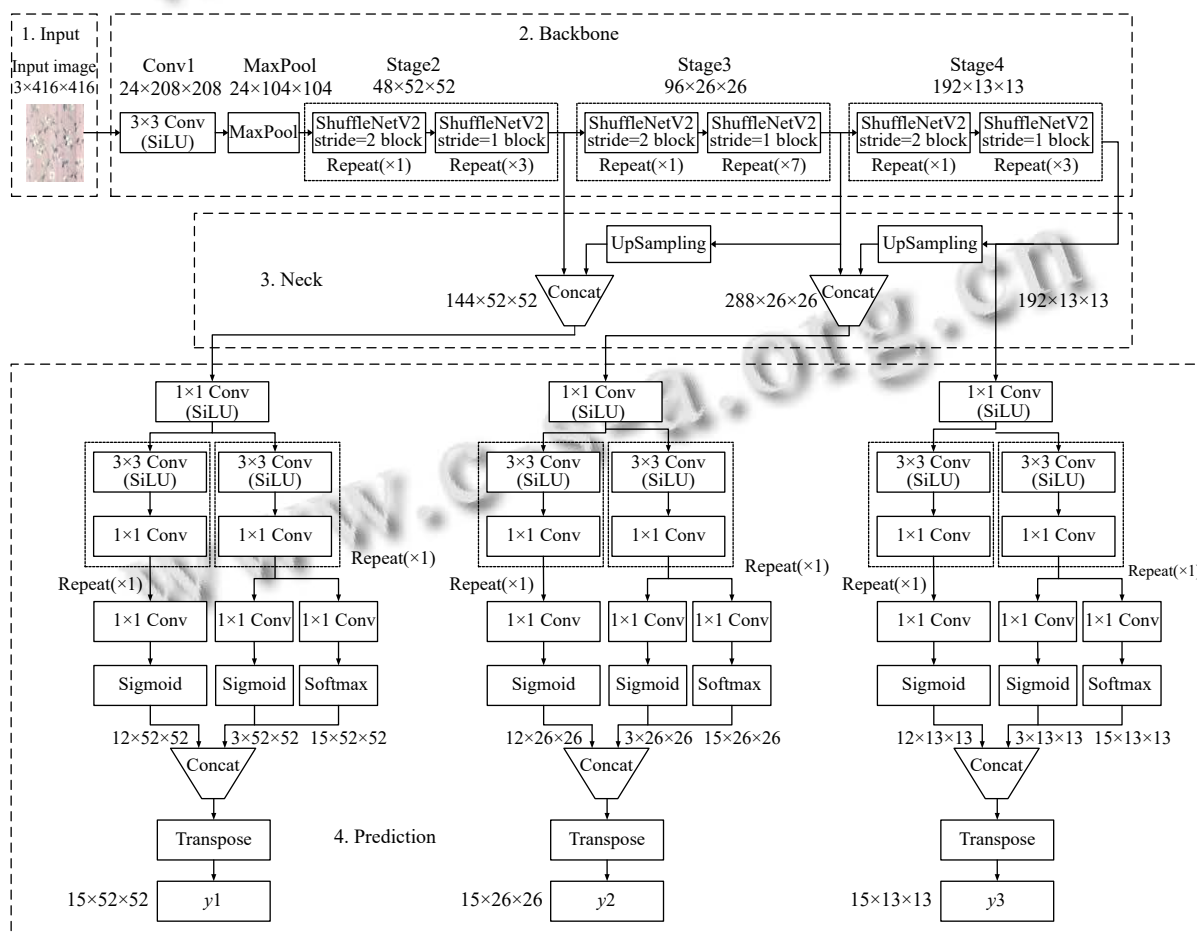


图6 本文检测算法整体网络结构图

对于检测输出 prediction, 也称为 YOLO 检测头 (YOLO Head), 用于将获得的 3 种尺寸大小的布匹瑕疵特征进行处理, 输出预测. 在目标检测的分类和定位任务中, 会存在 spatial misalignment 的问题^[27], 分类与定位任务的聚焦点不一样. 分类任务关注所提取的特征与所有检测类别的哪一类更为相近, 而定位任务更加关注与目标真实框 (ground truth box) 的位置坐标信息, 进而修正边界框参数. 本文将目标检测算法 YOLOv3-v4^[17,18] 使用同一个特征图进行分类和定位的 coupled head 方法改进为解耦头 (decoupled head)^[28] 的结构设计, coupled head 与 decoupled head 结构如图 7 所示.

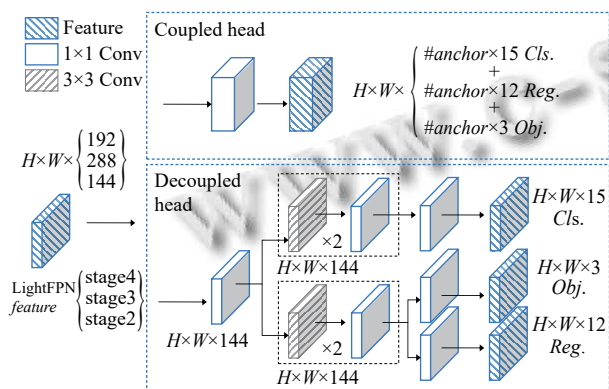


图 7 Coupled head 与 decoupled head 结构图

使用解耦头结构将特征图解耦成分别针对分类和定位任务后将获得 3 个预测结果, 分别是瑕疵框的前景背景信息判断 (Out_Obj), 大小为 $(3 \times w \times h)$, 输出通道数 3 为针对每个特征点产生的 3 个预测框是否有布匹瑕疵的概率信息; 瑕疵框的位置坐标 (Out_Reg), 大小为 $(12 \times w \times h)$, 输出通道数 12 为 3 个预测框的中心点对该特征点的坐标偏移情况及宽高对于该特征点的对数指数的参数信息; 瑕疵框的类别 (Out_Cls), 大小为 $(15 \times w \times h)$, 输出通道数 15 表示每一个特征点对应 15 类布匹瑕疵的概率信息.

2.4 损失函数

本文的分类概率损失 $Loss_{Cls}$ 和置信度损失 $Loss_{Obj}$ 所用的损失函数分别为交叉熵损失函数 (crossentropy loss)、二元交叉熵损失函数 (BCE with logits loss). IoU (intersection over union) 是一种在目标检测中检测相应目标准确度的一个标准, 计算预测框与真实框的交集和并集的值. IoU 和损失函数的计算公式为:

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (6)$$

$$Loss_{IoU} = 1 - IoU \quad (7)$$

其中, A 代表预测框, B 代表真实框. 在 IoU 的基础上, 瑕疵框的位置回归损失 $Loss_{Reg}$ 采用 $CIoU$ (complete IoU loss)^[29] 损失函数计算边框损失. $CIoU$ 的计算公式为:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(p, p^{gt})}{c^2} + \alpha V \quad (8)$$

$$\alpha = \begin{cases} 0, & \text{if } IoU < 0.5 \\ \frac{V}{(1 - IoU) + V}, & \text{if } IoU > 0.5 \end{cases} \quad (9)$$

$$V = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w}{h} \right)^2 \quad (10)$$

其中, ρ 为欧式距离, p 为预测框的中心点, c 为预测框与真实瑕疵框的最小外接矩形对角线距离, p^{gt} 为真实瑕疵框的中心点, w 为预测框的宽, w^{gt} 为真实瑕疵框的宽, h 为预测框的高, h^{gt} 为真实瑕疵框的高, α 为权重系数, V 为宽高比度量函数. $CIoU$ 重点将预测框与真实框重叠面积、中心点距离和长宽比等影响瑕疵框回归的因素全部考虑在内, 既提升了花色布匹瑕疵的检测精度, 也提高了瑕疵框回归的稳定性^[30].

3 实验与分析

3.1 实验环境与训练过程

本文的实验平台是 Ubuntu 18.04 LTS 64 位操作系统, CPU 为 Intel i9-10850K, 16 GB 内存, 显卡为 NVIDIA GeForce RTX 3070, 显卡显存为 8 GB, 包管理环境为 Anaconda. 使用 Python 3.8、PyTorch 1.9.0、CUDA 11.1 搭建网络模型.

嵌入式平台 Raspberry Pi 4B 搭载 1.5 GHz 的 64 位四核处理器 (Broadcom BCM2711, Cortex-A72 (ARM v8) 64-bit), 8 GB LPDDR4 内存, 配置系统为 Raspbian 64 位操作系统.

在模型训练过程中, 先将图像尺寸经过缩小调整为 416×416 大小, 再将图像做归一化处理输入网络进行训练. 网络参数训练优化采用随机梯度下降 (stochastic gradient descent, SGD) 优化器进行参数调整, 其中动量参数 (momentum) 设为 0.95, 权重衰减 (weight_decay) 为 0.0005. 训练迭代 300 次 (epoch), 初始学习率为 0.001, 训练批次 (batch_size) 大小为 100.

3.2 评价指标

本文的模型性能评价指标为在 Raspberry Pi 4B 上

用对测试集的检测速度、检测精度、模型大小来衡量模型性能. 检测速度为在 Raspberry Pi 4B 上每秒检测的布匹图像帧数 (FPS). 检测精度包括精确率 (precision, P)、召回率 (recall, R)、平均精度 (average precision, AP) 和平均精度均值 (mean average precision, mAP) 等. 计算公式为:

$$P = TP / (TP + FP) \quad (11)$$

$$R = TP / (TP + FN) \quad (12)$$

$$AP = \int_0^1 P(R) dR \quad (13)$$

$$mAP = \frac{1}{M} \sum_{m=1}^M AP^{(m)} \quad (14)$$

其中, TP 为判别正确的正样本数量, FP 为判别错误的正样本数量, FN 为判别错误的负样本数量. 其中精确率表示被认为是正样本中实际是正样本的目标个数占

被认为是正样本的比例. 召回率是指被认为是正样本中实际是正样本的目标个数占测试集中是正样本的比例. AP 表示在单个瑕疵类别下的平均精度, mAP 表示所有瑕疵类别的 AP 均值, 本文以 mAP 为最终评判标准.

3.3 结果与分析

为直观评估本文将检测头解耦合的影响, 验证改进模块对网络整体性能的效果. 本文将 coupled head 与 decoupled head 结构分别进行实验, 在本文数据集上进行训练损失与检测效果的可视化, 结果如图 8 所示. 从图 8(a) 训练损失曲线可以看出, decoupled head 结构的平滑收敛速度明显优于 coupled head 结构. 图 8(b) 为每 10 个 epoch 评估一次本文验证集上的 mAP . 显然, 从图中可以看出本文算法使用 decoupled head 结构的收敛速度比 coupled head 快得多, 并最终获得了更好的检测结果.

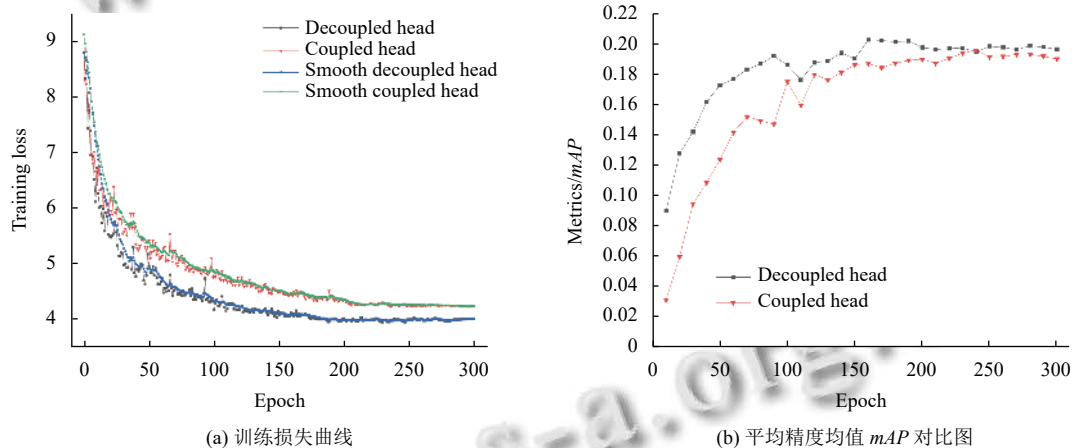


图 8 Coupled head 与 decoupled head 结构训练损失曲线与 mAP 对比图

为验证本文算法在 Raspberry Pi 4B 上检测精度与检测速度的有效性和先进性, 采用当前目标检测算法中适合在嵌入式设备上部署的 YOLOv5-lite-s^[31] 与

NanoDet-m-416^[32] 等轻量级算法进行实验对比. 使用本文花色布匹数据集的测试集对各算法进行对比测试, 本文算法与另两种检测算法的检测结果如表 1 所示.

表 1 不同算法的花色布匹瑕疵检测结果

Algorithm	Input size	mAP (%)	Params (M)	FLOPs (G)	FPS (img/s)	Weight size (M)
YOLOv5-lite-s	416×416	21.9	1.49	3.7	3.2	3.4
NanoDet-m-416	416×416	14.4	0.89	1.2	3.8	3.8
本文	416×416	20.1	0.27	0.3	8.6	1.3

从表 1 可以看出, 在花色布匹瑕疵检测中, 与算法 YOLOv5-lite-s 和 NanoDet-m-416 算法相比, 本文算法在参数量 (Params) 和浮点运算量上 (FLOPs) 上有明显

优势, 本文方法的参数量仅为 0.27 MB, 远小于另两种检测算法的 1.49 MB 和 0.89 MB, FLOPs 的大小也仅为 0.3 G. 同 YOLOv5-lite-s 检测算法相比, 在平均精度

均值 mAP 低 1.8% 的情况下, 本文算法在 Raspberry Pi 4B 上的检测速度达到 8.6 FPS, 在检测速度上约为另两种算法的 2.5 倍. 此外在模型大小上仅为另外两种算法的

1/3. 通过对比数据可知, 本文算法在保证一定精确度时大大提升了花色布匹瑕疵的检测速度, 更加适合于嵌入式设备上的检测部署. 本文算法检测结果如图 9 所示.

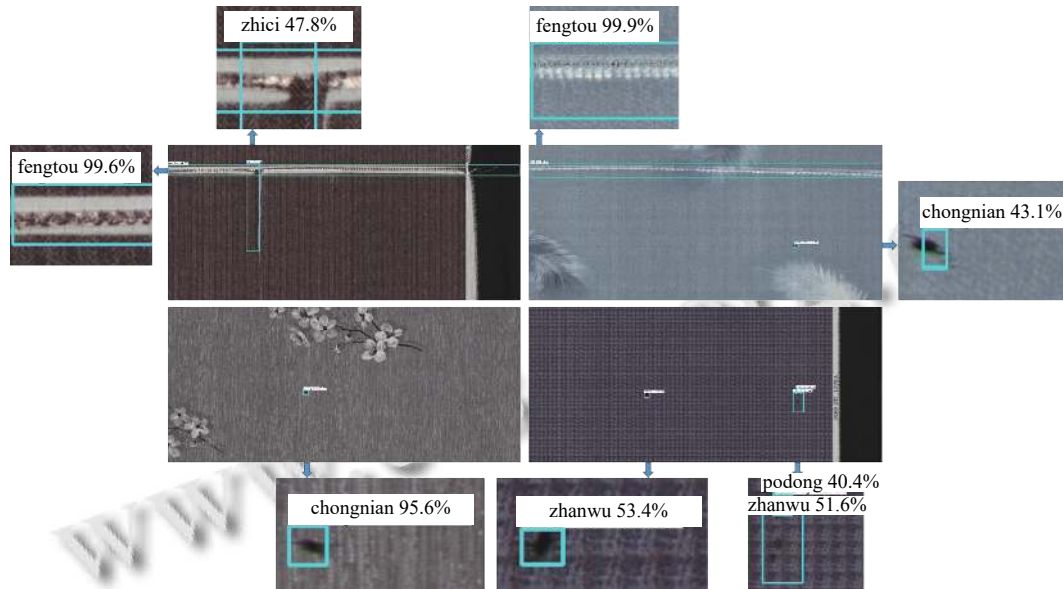


图9 花色布匹瑕疵检测结果图

从检测结果图可以看出, 本文算法不仅能对有较大长宽比的“缝头”类型瑕疵进行精准检测, 其类别置信度可达 99% 以上, 对于像素占比少的“虫粘”“沾污”和“破洞”等小瑕疵类型也有较好的检出效果, 在嵌入式设备上实现了准确、快速的检测.

4 结论

针对目前纺织业花色布匹瑕疵检测成本高、效率低和漏检率高等问题, 提出了一种基于改进 ShuffleNetV2 的轻量级花色布匹瑕疵检测方法并实现在嵌入式设备 Raspberry Pi 4B 上的部署检测. 本文方法在改进 ShuffleNetV2 网络基础上充分融合浅层位置信息与深层高级语义信息; 在检测输出部分将检测头进行解耦计算, 加快模型训练和收敛速度; 为提升瑕疵预测框的定位精度与加快收敛, 预测框采用 CIoU 损失函数计算边框损失. 实验结果表明, 本文方法在检测速度与模型参数量上有显著优势, 在极大减少企业花色布匹瑕疵检测成本的同时能满足瑕疵检测的要求.

参考文献

1 Ngan HYT, Pang GKH, Yung NHC. Automated fabric defect

detection—A review. *Image and Vision Computing*, 2011, 29(7): 442–458. [doi: 10.1016/j.imavis.2011.02.002]

2 孟志青, 邱健数. 基于级联卷积神经网络的复杂花色布匹瑕疵检测算法. *模式识别与人工智能*, 2020, 33(12): 1135–1144. [doi: 10.16451/j.cnki.issn1003-6059.202012008]

3 李春雷, 高广帅, 刘洲峰, 等. 应用方向梯度直方图和低秩分解的织物疵点检测算法. *纺织学报*, 2017, 38(3): 149–154. [doi: 10.13475/j.fzxb.20160304106]

4 孙国栋, 林松, 艾成汉, 等. 基于灰度共生矩阵与反向投影的织物疵点检测. *计算机测量与控制*, 2016, 24(7): 65–67. [doi: 10.16526/j.cnki.11-4762/tp.2016.07.018]

5 Mak KL, Peng P, Yiu KFC. Fabric defect detection using morphological filters. *Image and Vision Computing*, 2009, 27(10): 1585–1592. [doi: 10.1016/j.imavis.2009.03.007]

6 陈剑威, 冯晓霞, 柳成林. 基于频率域滤波器的织物疵点检测. *闽南师范大学学报(自然科学版)*, 2020, 33(1): 18–23. [doi: 10.16007/j.cnki.issn2095-7122.2020.01.004]

7 Hu GH, Wang QH, Zhang GH. Unsupervised defect detection in textiles based on Fourier analysis and wavelet shrinkage. *Applied Optics*, 2015, 54(10): 2963–2980. [doi: 10.1364/AO.54.002963]

8 赵宏威, 王亦红. 基于改进 Gabor 优化选择的布匹瑕疵检测方法. *计算机工程与应用*, 2019, 55(24): 202–207. [doi: 10.3778/j.issn.1002-8331.1901-0403]

- 9 Bu HG, Huang XB, Wang J, *et al.* Detection of fabric defects by auto-regressive spectral analysis and support vector data description. *Textile Research Journal*, 2010, 80(7): 579–589. [doi: [10.1177/0040517509340599](https://doi.org/10.1177/0040517509340599)]
- 10 Zhang Y, Jiang GM, Yao JZ, *et al.* Intelligent segmentation of jacquard warp-knitted fabric using a multiresolution Markov random field with adaptive weighting in the wavelet domain. *Textile Research Journal*, 2014, 84(1): 28–39. [doi: [10.1177/0040517513485629](https://doi.org/10.1177/0040517513485629)]
- 11 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.
- 12 Girshick R. Fast R-CNN. *Proceedings of 2015 IEEE International Conference on Computer Vision*. Santiago: IEEE, 2015. 1440–1448.
- 13 Ren SQ, He KM, Girshick R, *et al.* Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39(6): 1137–1149. [doi: [10.1109/TPAMI.2016.2577031](https://doi.org/10.1109/TPAMI.2016.2577031)]
- 14 蔡兆信, 李瑞新, 戴逸丹, 等. 基于 Faster RCNN 的布匹瑕疵识别系统. *计算机系统应用*, 2021, 30(2): 83–88. [doi: [10.15888/j.cnki.csa.007818](https://doi.org/10.15888/j.cnki.csa.007818)]
- 15 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.
- 16 Redmon J, Farhadi A. YOLO9000: Better, faster, stronger. *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu: IEEE, 2017. 6517–6525.
- 17 Redmon J, Farhadi A. YOLOv3: An incremental improvement. *arXiv:1804.02767*, 2018.
- 18 Bochkovskiy A, Wang CY, Liao HYM. YOLOv4: Optimal speed and accuracy of object detection. *arXiv:2004.10934*, 2020.
- 19 Jing JF, Zhuo D, Zhang HH, *et al.* Fabric defect detection using the improved YOLOv3 model. *Journal of Engineered Fibers and Fabrics*, 2020, 15: 1558925020908268.
- 20 Zheng LM, Wang XJ, Wang Q, *et al.* A fabric defect detection method based on improved YOLOv5. *Proceedings of the 2021 7th International Conference on Computer and Communications (ICCC)*. Chengdu: IEEE, 2021. 620–624.
- 21 Howard AG, Zhu ML, Chen B, *et al.* MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861*, 2017.
- 22 Howard A, Sandler M, Chu B, *et al.* Searching for MobileNetV3. *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*. Seoul: IEEE, 2019. 1314–1324.
- 23 Zhang XY, Zhou XY, Lin MX, *et al.* ShuffleNet: An extremely efficient convolutional neural network for mobile devices. *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City: IEEE, 2018. 6848–6856.
- 24 Ma NN, Zhang XY, Zheng HT, *et al.* ShuffleNet V2: Practical guidelines for efficient CNN architecture design. *Proceedings of the 15th European Conference on Computer Vision (ECCV)*. Munich: Springer, 2018. 122–138.
- 25 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.
- 26 Tianchi. Smart diagnosis of cloth flaw dataset. <https://tianchi.aliyun.com/dataset/dataDetail?dataId=79336>. (2020-10-21).
- 27 Song GL, Liu Y, Wang XG. Revisiting the sibling head in object detector. *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Seattle: IEEE, 2020. 11560–11569.
- 28 Ge Z, Liu ST, Wang F, *et al.* YOLOX: Exceeding YOLO series in 2021. *arXiv:2107.08430*, 2021.
- 29 Zheng ZH, Wang P, Ren DW, *et al.* Enhancing geometric factors in model learning and inference for object detection and instance segmentation. *IEEE Transactions on Cybernetics*, 2022, 52(8): 8574–8586. [doi: [10.1109/TCYB.2021.3095305](https://doi.org/10.1109/TCYB.2021.3095305)]
- 30 王宸, 唐禹, 张秀峰, 等. 基于改进 EfficientNet 的锻件磁粉探伤智能检测方法研究. *仪器仪表学报*, 2021, 42(9): 89–96. [doi: [10.19650/j.cnki.cjsi.J2107796](https://doi.org/10.19650/j.cnki.cjsi.J2107796)]
- 31 Chen X, Gong Z. YOLOv5-lite: Lighter, faster and easier to deploy. <https://pythonawesome.com/yolov5-lite-lighter-faster-and-easier-to-deploy/>. (2021-09-05).
- 32 RangiLyu. NanoDet-Plus: Super fast and high accuracy lightweight anchor-free object detection model. <https://github.com/RangiLyu/nanodet>.

(校对责编: 牛欣悦)