

# 基于特征残差的色织物瑕疵检测<sup>①</sup>



包晓安, 林德守, 张 娜

(浙江理工大学 信息学院, 杭州 310018)

通讯作者: 张 娜, E-mail: zhangna@zstu.edu.cn

**摘 要:** 为解决自动织物瑕疵检测算法中, 未知花色织物瑕疵检测困难的问题, 提出了一种基于特征残差的色织物瑕疵检测方法. 首先使用瑕疵织物图像与模板织物图像的瑕疵残差和正常无标注织物图像进行融合, 生成新花色瑕疵织物样本; 然后改进特征提取网络采用共享权值方法, 对瑕疵织物和模板织物提取特征后计算得到特征残差; 最后使用 ROIAlign 方法将全局上下文信息缩放到和感兴趣区域统一大小后进行特征融合, 对融合特征进行瑕疵分类和位置回归. 实验针对不包含未知花色和包含未知花色的不同测试集分别进行算法测试实验, 结果表明改进后的算法能够较好地消除织物花色对检测结果的影响, 在不包含未知花色的测试集中精度得到了较大的提升, 在包含未知花色的测试集中, 瑕疵检测效果依旧保持不错的精度, 相较于改进前的通用算法, 最终 score 分别提升了 15.4% 和 16.2%.

**关键词:** 织物瑕疵; 瑕疵检测; 特征残差; 深度学习; 目标检测

引用格式: 包晓安, 林德守, 张娜. 基于特征残差的色织物瑕疵检测. 计算机系统应用, 2021, 30(10):224-231. <http://www.c-s-a.org.cn/1003-3254/8159.html>

## Fabric Defect Detection Based on Feature Residual

BAO Xiao-An, LIN De-Shou, ZHANG Na

(School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou 310018, China)

**Abstract:** To solve the difficulty in detecting the defects in fabrics of unknown styles in the automatic fabric defect detection algorithm, this study proposes a fabric defect detection method based on feature residuals. First, the defect residuals of the defective and template fabric images are fused with that of the normal unlabeled fabric image to generate a new defective fabric sample. Then, the improved feature extraction network uses the shared weight method to extract features from the defective and template fabrics and calculate the feature residuals. Finally, the ROIAlign method is used to mix the global context information and the region of interest for feature fusion. The fused features are subject to defect classification and location return. Experiments are separately conducted on two test sets containing and not containing fabrics of unknown styles. The results show that the improved algorithm can better eliminate the influence of fabric styles on the detection results. The accuracy is greatly improved in the test set that does not include unknown styles. In the test set containing unknown styles, defect detection maintains high accuracy. Compared with that of the general algorithm before the improvement, the final scores have increased by 15.4% and 16.2%, respectively.

**Key words:** fabric defect; defect detection; feature residual; deep learning; object detection

① 基金项目: 国家自然科学基金 (6207050141); 浙江省自然科学基金 (LQ20F050010); 浙江省重点研发计划 (2020C03094)

Foundation item: National Natural Science Foundation of China (6207050141); Natural Science Foundation of Zhejiang Province (LQ20F050010); Key R & D Program of Zhejiang Province (2020C03094)

收稿时间: 2020-12-30; 修改时间: 2021-01-29, 2021-02-08, 2021-02-23; 采用时间: 2021-03-03

在纺织品的质量控制阶段, 织物疵点检测是其中的一个重要环节<sup>[1]</sup>. 由于织物疵点种类多, 形态变化大, 观察识别难等, 导致织物疵点智能检测是困扰行业多年的技术瓶颈. 目前大部分的纺织企业仍采用人工方法检测织物瑕疵<sup>[2]</sup>, 它易受主观因素影响, 缺乏一致性; 且检测人员在强光下长时间工作对视力影响较大; 还存在人工成本较高、漏检率和误检率较高、效率不高等问题.

随着计算机视觉技术的发展提高, 自动瑕疵检测技术逐渐成为新的研究热点. 目前, 国内外对织物疵点检测算法的研究主要是针对素色织物进行的<sup>[3]</sup>. 相较于素色织物, 色织物由于颜色等特征的加入, 导致检测难度较大, 所以目前对色织物疵点检测方法的研究较少, 且检测效果都不理想<sup>[3]</sup>. 其中, 纪旋等人<sup>[4]</sup>提出结合模板校正与低秩分解的纺织品瑕疵检测方法, 解决周期性织物存在拉伸变形问题; 兰瑜洁等人<sup>[5]</sup>提出基于频域滤波和区域生长法的瑕疵区域自适应分割方法, 提高瑕疵的检测精度; 张波等人<sup>[6]</sup>提出基于结构-纹理模型与自适应数学形态学的织物瑕疵加成算法, 解决织物纹理多样性和瑕疵形状尺寸不确定性造成的检测效果差的问题.

上述方法主要使用传统图像算法, 存在检测算法设计只能针对特定场景, 泛化能力弱等问题. 随着深度学习算法的不断改进, 在泛化能力、精度等方面都超越了传统图像算法. Wang 等人<sup>[7]</sup>提出了基于卷积神经网络的深度学习检测算法, 在更少先验知识的情况下, 达到快速、精确的对织物瑕疵进行检测; Zhao 等人<sup>[8]</sup>提出了基于正样本训练的缺陷检测框架, 结合 GAN 和自动编码进行缺陷图像重建, 解决缺陷样本或阴性样本少的情况下检测的问题; Tabernik 等人<sup>[9]</sup>提出了基于语义分割的织物瑕疵区域的定位方法, 可以在少量样本的训练数据中, 达到较好的检测效果等.

但是, 基于卷积神经网络的深度学习解决方案, 还存在着两个问题: (1) 实际场景中, 存在部分织物花色未在训练数据集中出现, 此类织物检测效果会有较大程度退化; (2) 织物花色种类繁多, 大量无标注正常织物样本没有得到很好地利用. 基于此, 本文通过使用特征残差改进 Cascade RCNN 算法<sup>[10]</sup>, 提高对未知花色织物的检测能力; 使用瑕疵残差增强方案, 有效利用无标注正常织物样本, 提高算法泛化能力.

## 1 相关工作

### 1.1 Cascade RCNN 算法概述

Cascade RCNN 算法是基于 Faster RCNN 算法<sup>[11]</sup>的两阶段目标检测算法. Faster RCNN 算法首先在第一阶段, 通过特征提取网络对图像提取特征, 使用不同比例和大小的预定义框在特征图中进行滑动窗口操作, 生成若干 anchors, 再使用区域推荐网络, 将这些 anchors 分为前景和背景, 对前景进行回归, 得到边界框, 作为推荐区域; 第二阶段将第一阶段生成的推荐区域进行第二次分类, 细分瑕疵类别, 并进行第二次回归, 得到更加精确的边界框, 作为算法的检测结果. 由于在第一阶段使用区域推荐网络进行了前景和背景的分类, 双阶段的检测算法很好地处理了正负样本比例失衡问题, 算法的检测精度较高. Cascade RCNN 算法在第二阶段, 级联 3 个大小递增的 IOU 头部, 每个头部对上一步得到的结果再次进行回归和分类, 使得到的边界框更加精确.

### 1.2 边框回归

如图 1 所示, 候选框  $P$  为预定义的 anchor, 目标框  $G$  为真实标注框, 边框回归的最终目的即将模型输出的预测框  $G'$  更加接近真实目标框  $G$ . 当候选框  $P$  与标注框  $G$  的 IOU 较大时, 从候选框到预测框  $G'$  的变换  $d^*$  可以认为近似的线性变换. 定义目标框的中心  $(G_x, G_y)$ , 宽高  $(G_w, G_h)$ , 候选框中心  $(P_x, P_y)$ , 宽高  $(P_w, P_h)$ , 建立回归模型:

$$\begin{cases} G'_x = P_x + P_w d_x(P) \\ G'_y = P_y + P_h d_y(P) \\ G'_w = P_w \exp(d_w(P)) \\ G'_h = P_h \exp(d_h(P)) \\ d_*(P) = w_*^T \phi(P) \end{cases} \quad (1)$$

其中,  $\phi(P)$  为候选框特征信息,  $w_*$  为要学习的参数.

### 1.3 损失函数

Cascade RCNN 训练损失函数由目标框和预测框的边框误差和分类的置信度误差加权求和得到. 损失函数如式 (2) 所示.

$$L(y_i, x_i, G'_i, G_i) = \sum_{i=1}^N ((L_{cls}(y_i, h(x_i)) + \alpha L_{loc}(G'_i, G_i))) \quad (2)$$

其中,  $N$  为预测框的数量,  $y_i$  为目标框类别,  $x_i$  为输入特征向量,  $h(x_i)$  为 Softmax 分类函数,  $G'_i$  为预测框的位置信息,  $G_i$  为目标框位置信息. 边框损失采用 SmoothL1

作为损失函数, 优化函数如式 (3) 所示.

$$L_{loc} = \sum_{i=1}^N SmoothL1(G'_i, G_i)$$

$$SmoothL1(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & otherwise \end{cases} \quad (3)$$

置信度损失  $L_{cls}$  是预测类别置信度与目标类别的误差, 损失函数采用交叉熵损失函数, 计算公式如式 (4) 所示.

$$\begin{cases} L_{cls}(y_i, h(x_i)) = y_i \ln(h(x_i)) + (1 - y_i) \ln(1 - h(x_i)) \\ h(x_i) = \frac{e^j}{\sum_j e^j} \end{cases} \quad (4)$$

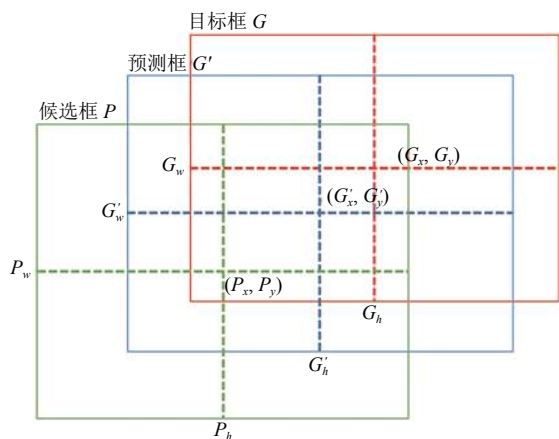


图1 边框回归

## 2 算法改进

### 2.1 算法整体结构

FR-Cascade RCNN 算法对 Cascade RCNN 算法进行改进, 算法整体结构为, 首先使用共享权值的特征提

取网络对模板和瑕疵图像分别提取特征, 将两个特征图相减得到特征残差图, 并增加一条支路, 对瑕疵特征图使用 ROIAlign<sup>[12]</sup> 将特征图缩放到  $7 \times 7$  大小作为全局特征信息; 通过使用 FPN<sup>[13]</sup> 网络, 增加特征的多尺度信息, 将 FPN 网络的输出结果使用 RPN 区域推荐网络生成 ROI 区域, 此时将  $7 \times 7$  大小的全局特征信息叠加到 ROI 区域, 通过 Cascade 的级联方式, 对 ROI 区域进行分类和回归, 得到检测结果. 算法框架如图 2 所示.

### 2.2 数据增强方法

常规数据增强操作只能通过对原有瑕疵样本进行变化得到新的训练样本, 不能将瑕疵特征扩展到其他样本数据中, 无法有效利用无标注的正常样本, 存在较大的局限性. 在实际应用中, 织物花色的种类较多, 存在部分花色未在训练样本中出现, 而训练样本中存在的瑕疵可能只在一些花色中存在, 导致使用原始数据或常规数据增强得到的数据训练得到的模型泛化性能必然较差. 针对此问题, 提出一种残差增强方法, 解决瑕疵特征不能扩展到其他训练样本中的问题. 该方法通过使用瑕疵样本和模板样本逐像素相减求取瑕疵残差图, 将残差图的瑕疵区域与其他花色的正常无标注图像进行融合, 得到新的训练样本. 此数据增强方式在不破坏新花色背景特征的同时将不同瑕疵与不同花色织物相互组合, 有效的利用了正常无标注织物样本, 生成特征丰富的训练数据. 使用瑕疵残差增强后,  $mAP$  指标得到了较高的提升. 增强计算方法如式 (5) 所示.

$$D_r = D_r + (F_r - T_r) \quad (5)$$

式中,  $D$  为待增强织物图像,  $F$  为瑕疵的织物图像,  $T$  为  $F$  的模板图像, 下标  $r$  为瑕疵区域编号. 图像增强效果图如图 3, 图 4 所示.

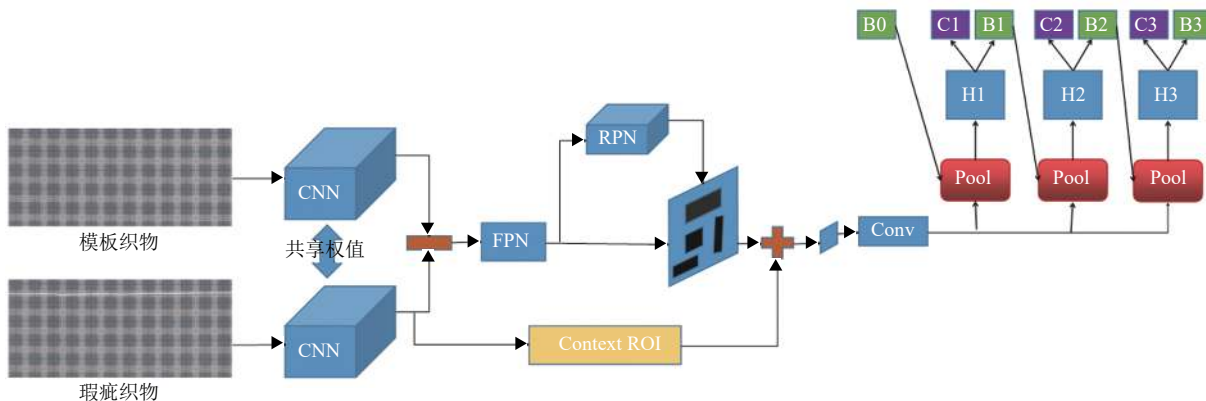


图2 算法整体框架

数据增强流程首先从数据集中随机选择一张织物图像,如果织物含有瑕疵,直接将图像输入到算法模型进行训练;如果织物不包含瑕疵,随机抽取瑕疵图像,提取瑕疵区域和该织物进行瑕疵残差增强,将生成图像输入算法模型进行训练,算法中数据增强流程如图5所示。

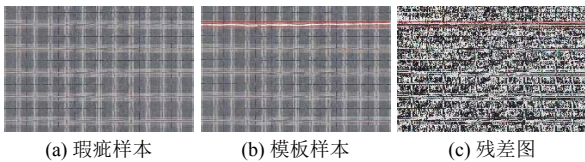


图3 瑕疵样本和模板样本逐像素相减得到残差图

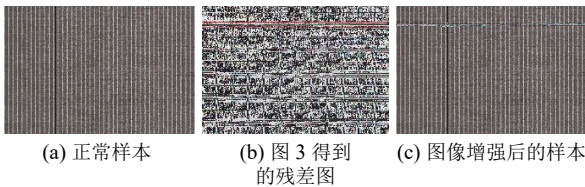


图4 正常样本与残差图相加得到目标样本

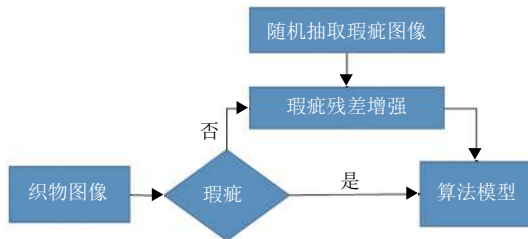


图5 残差增强流程

### 2.3 特征残差

针对实际场景中,存在未经训练的织物花色,此类样本中存在的瑕疵通过常规的检测模型较难被检出,提出了FR-Cascade RCNN算法,解决此问题.FR-Cascade RCNN模型通过改进Cascade RCNN算法,在特征提取层加入特征残差结构,消除样本背景的干扰,提高模型对复杂或未训练花色样本瑕疵的检出能力.FR-Cascade RCNN中特征提取网络特征输出函数如式(6)所示。

$$O_i = D_i - T_i \tag{6}$$

其中,  $O$  为输出特征向量,  $D$  为瑕疵特征向量,  $T$  为模板特征向量,  $i$  为特征提取网络中的第  $i$  阶段.FR-Cascade RCNN中特征提取网络结构如图6所示。

图6中,特征提取网络采用ResNext50<sup>[14]</sup>+DCN<sup>[15]</sup>模型结构,模型结构如表1所示。

织物瑕疵图像和模板图像经过共享权值的特征提

取网络,提取特征后,取特征差值,再通过FPN,增加瑕疵特征的多尺度信息,再通过RPN区域推荐网络,得到ROI区域,最后通过Cascade head,对ROI区域进行分类和回归,得到最终瑕疵的区域和类别。

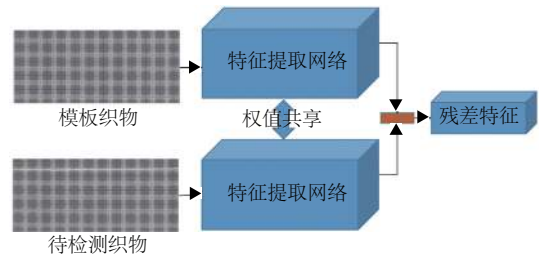


图6 特征残差计算方法

表1 特征提取网络模型结构

Stage	Output	DCN	ResNext50				
Conv1	2048×900	×	7×7, 64, stride 2 3×3 max pool, stride 2				
Layer1	1024×450	×	<table border="1" style="display: inline-table;"> <tr> <td>1×1, 128</td> <td rowspan="3">×3</td> </tr> <tr> <td>3×3, 128, C = 32</td> </tr> <tr> <td>1×1, 256</td> </tr> </table>	1×1, 128	×3	3×3, 128, C = 32	1×1, 256
1×1, 128	×3						
3×3, 128, C = 32							
1×1, 256							
Layer2	512×225	√	<table border="1" style="display: inline-table;"> <tr> <td>1×1, 256</td> <td rowspan="3">×4</td> </tr> <tr> <td>3×3, 256, C = 32</td> </tr> <tr> <td>1×1, 512</td> </tr> </table>	1×1, 256	×4	3×3, 256, C = 32	1×1, 512
1×1, 256	×4						
3×3, 256, C = 32							
1×1, 512							
Layer3	256×112	√	<table border="1" style="display: inline-table;"> <tr> <td>1×1, 512</td> <td rowspan="3">×6</td> </tr> <tr> <td>3×3, 512, C = 32</td> </tr> <tr> <td>1×1, 1024</td> </tr> </table>	1×1, 512	×6	3×3, 512, C = 32	1×1, 1024
1×1, 512	×6						
3×3, 512, C = 32							
1×1, 1024							
Layer4	128×56	√	<table border="1" style="display: inline-table;"> <tr> <td>1×1, 1024</td> <td rowspan="3">×3</td> </tr> <tr> <td>3×3, 1024, C = 32</td> </tr> <tr> <td>1×1, 2048</td> </tr> </table>	1×1, 1024	×3	3×3, 1024, C = 32	1×1, 2048
1×1, 1024	×3						
3×3, 1024, C = 32							
1×1, 2048							

在实际场景中需要对未知花色的图像进行检测,容易想到消除背景干扰对检测结果会有较大帮助.可以通过采取瑕疵图像和模板图像做差后进行检测,但是由于瑕疵图像和模板图像并不是完全对齐,可能存在几个像素至几十个像素之间的偏移,所以此种方法并不能提高模型的检测能力,反而会降低模型的检测精度.由于卷积操作本身具备平滑局部特征的能力,所以通过使用共享权值的特征提取网络提取特征后,对特征差值进行检测,在较大程度上消除了图像背景的干扰,解决了此问题,较大地提高了对未知花色的瑕疵检出能力。

### 2.4 全局上下文信息

由于上一步的特征残差网络对背景特征进行了极大的过滤,在RPN阶段,可以更好地提取得到前景框,但是输出的候选框失去了目标周围的上下文信息,而在检测网络中,目标周围特征对于目标的定位也有一定帮助,所以网络需要一个全局特征信息去辅助局部

瑕疵区域的分类和定位. 把全局特征当作一个 ROI 叠加到每一个候选框中去可以为局部瑕疵提供全局特征, 网络根据上下文周围信息进行更加精确的定位. 在模型中添加全局上下文信息流程图如图 7 所示.

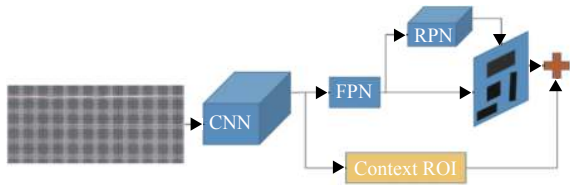


图 7 全局上下文信息添加过程

由于全局特征图的维度和候选框特征图的维度不相同, 需要将全局特征图的维度使用 ROIAlign 的方法降低到和候选框特征图相同的维度.

### 3 实验环境和实验数据

#### 3.1 数据集介绍

数据集由瑕疵样本和无瑕疵样本组成, 其中瑕疵样本图片有 4202 张, 无瑕疵样本图片有 894 张, 共

68 种花色, 分辨率为 4096×1800. 训练集由 3523 张瑕疵样本和 572 张无瑕疵样本组成, 样本总计 4095 张, 共 63 种花色; 测试集由 679 张瑕疵样本和 321 张无瑕疵样本组成, 样本总计 1000 张, 共 68 种花色, 其中, 未在训练集中出现的花色有 5 种, 共 279 张. 其中包含污渍、错花、水印、花毛、缝头、缝头印、虫粘、破洞、褶子、织疵、漏印、蜡斑、色差、网折和其他等 15 类瑕疵, 共约 17500 个瑕疵区域. 数据如表 2 所示.

瑕疵类别部分样本中瑕疵区域如图 8 所示.

将测试集分为测试集 A 和测试集 B, 其中测试集 A 中不包含未知花色样本, 共 400 张图像, 测试集 B 中包含所有图像, 共 679 张图像 (测试集 B 由测试集 A 和未知花色图像组成). 实验数据部分织物如图 9 所示.

表 2 数据集样本数量

数据集	瑕疵样本	正常样本	花色
训练集	3523	572	63
测试集	679	321	68
合计	4202	894	68

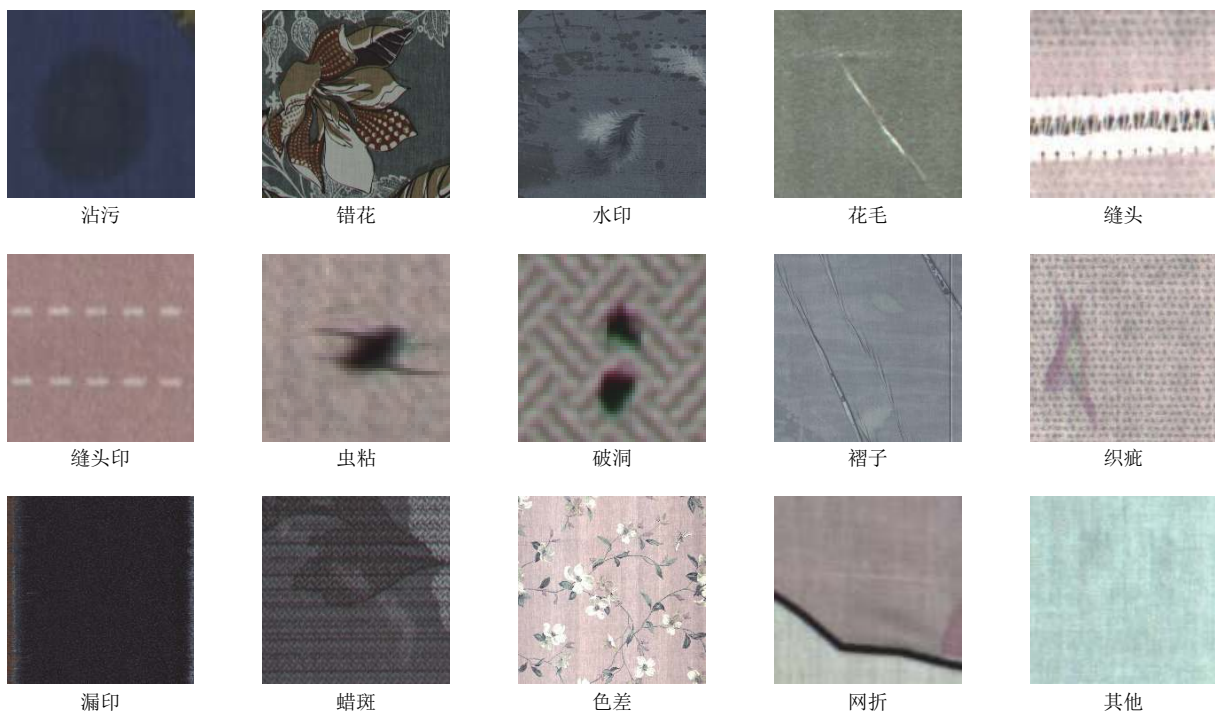


图 8 瑕疵类别

#### 3.2 实验设置与评价指标

实验基于 Ubuntu 16.0.4 操作系统, 使用 GPU 版 PyTorch 深度学习框架, GPU 型号为 Tesla T4, 显存为

15 GB. 由于数据集规模不大, 模型采用经过 COCO 数据集预训练过的参数进行初始化, 做迁移学习. 由于图像较大, batchsize 设置为 1. 训练时增加了多尺度训练,

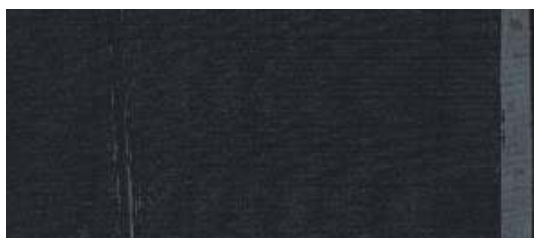
增加模型对不同尺度瑕疵的适应能力, 训练设置长边尺寸为 4096, 短边尺寸为 1024–1440. 由于瑕疵长宽比变化范围较大, 采用 K-means 方法聚类得到 anchor 比例为 [0.04, 0.15, 0.5, 1.0, 2.0, 5.0, 25.0]. 学习率初始设置为 0.001 25, 模型训练初始损失值为 0.9514, 经过 12 轮训练, 损失值收敛到 0.1941, 此时降低学习率到 0.000 125, 训练到 15 轮, 最终损失值收敛到 0.1423. 损失值和  $mAP$  收敛曲线如图 10, 图 11 所示.



(a) 花色 1



(b) 花色 2



(c) 花色 3



(d) 花色 4

图 9 部分织物数据

部分瑕疵检测实验效果图如图 12 所示.

实际环境中, 算法模型除了要有较高的瑕疵检测能力, 织物是否含有瑕疵的准确率也是一个重要指标, 故评价指标设置为  $score=0.8 \times mAP+0.2 \times Acc$ , 使用  $mAP$ <sup>[16]</sup> 衡量算法的瑕疵检测能力,  $Acc$  衡量织物是否

含有瑕疵的准确率.  $mAP$  和  $Acc$  权值系数可根据算法着重点进行调整.

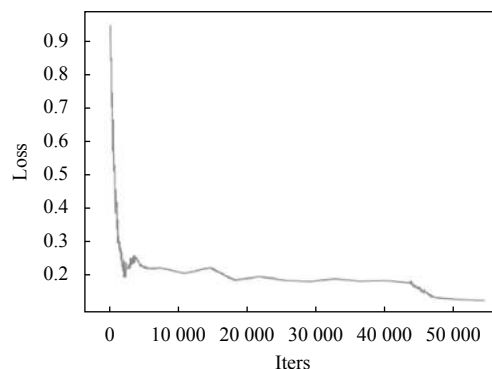
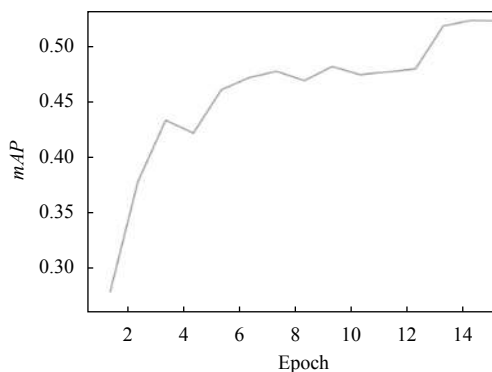


图 10 损失值收敛图

图 11  $mAP$  收敛图

### 3.3 实验结果

实验结果如表 3 所示, RA 为瑕疵残差增强 (Residual Augment), Cascade+RA 指在 Cascade RCNN 基础上, 加入 RA 模块; FR 为特征残差 (Feature Residual), FR-Cascade 指在 Cascade RCNN 基础上增加了 FR 模块; FR+RA 指在 FR-Cascade 基础上加入 RA 模块; GC 为全局上下文信息 (Global Context), FR+RA+GC 指在 Cascade RCNN 模块上同时加入这 3 个模块; A- $Acc$  为测试集 A 的  $Acc$  指标, B- $Acc$  为测试集 B 的  $Acc$  指标, 其他同理. 最终各个瑕疵在测试集 B 中的检测结果如表 4 所示.

改进后的模型实验结果如表 3 所示, 其中瑕疵残差增强相比于增强前,  $Acc$  和  $mAP$  指标, 都有较大的提高, 在测试集 A 和测试集 B 中分别提高了 5.6% 和 2.4%; 改进的 FR-Cascade RCNN 算法, 在  $Acc$  和  $mAP$  指标上都得到了较大的提高,  $Acc$  在测试集 A 和测试集

B中分别提高了19.2%和20%,  $mAP$ 分别提高了11.1%和11.8%。在FR-Cascade RCNN基础上增加RA和GC,得到的最终结果,在测试集A和测试集B中,  $Acc$ 分别提高了19.7%和20.5%,  $mAP$ 分别提高了14.3%和15.2%。实验结果表明,改进后的FR-Cascade RCNN算法,不仅对已知花色有较强的检测能力,在未知花色中,依然保持了较高的精度。

模型对不同瑕疵的检测效果如表4所示,改进后的模型虽然在检测精度上得到了较高的提升,但是部分类别由于存在大量小目标、瑕疵不明显、极端长宽比、长尾分布等情况,检测精度还存在一些不足,这些将是算法后序的优化方向。



(a) 油污



(b) 褶子



(c) 缝头



(d) 破洞

图12 瑕疵检测效果图

## 4 结束语

基于织物含有模板的前提下,通过特征残差改进的FR-Cascade RCNN算法,降低模型对织物背景花色的敏感度,不仅提高了对已知花色织物瑕疵的检测能力,同时,较大程度上解决了对未知花色织物检测困难的问题。在此基础上,通过对ROI区域添加全局上下信息,提高瑕疵的定位能力,对最终的检测结果也有一定的提升。同时提出了瑕疵残差数据增强方案,有效利用正常无标注样本,使模型的泛化能力得到一定的提升。但是,算法对小瑕疵、不明显瑕疵等的检测上依然存在一些不足,这将是算法后序的优化方向。

表3 各算法实验结果

算法	A-Acc	A-mAP	A-score	B-Acc	B-mAP	B-score
SSD512	0.725	0.355	0.429	0.712	0.339	0.414
RetinaNet	0.730	0.371	0.443	0.719	0.356	0.429
Faster RCNN	0.733	0.376	0.447	0.720	0.362	0.434
Cascade RCNN	0.736	0.382	0.453	0.722	0.369	0.440
Cascade+RA	0.792	0.406	0.483	0.788	0.397	0.475
FR-Cascade	0.928	0.493	0.580	0.922	0.487	0.574
FR+RA	0.926	0.511	0.594	0.919	0.508	0.590
FR+RA+GC	0.933	0.525	0.607	0.927	0.521	0.602

表4 最终实验结果

类别	目标数量	AP
油污	1515	0.355
错花	1148	0.291
水印	974	0.279
花毛	508	0.412
缝头	133	0.648
缝头印	64	0.307
虫粘	86	0.986
破洞	71	0.715
褶子	146	0.737
织疵	59	0.628
漏印	102	0.521
蜡斑	47	0.403
色差	84	0.221
网折	36	0.959
其他	52	0.357
$mAP$		0.521

## 参考文献

- 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]
- Kumar A. Computer vision-based fabric defect detection: A

- survey. *IEEE Transactions on Industrial Electronics*, 2008, 55(1): 348–363. [doi: [10.1109/TIE.1930.896476](https://doi.org/10.1109/TIE.1930.896476)]
- 3 周文明, 周建, 潘如如. 应用上下文视觉显著性的色织物疵点检测. *纺织学报*, 2020, 41(8): 39–44.
  - 4 纪旋, 梁久祯, 侯振杰, 等. 基于模板校正与低秩分解的纺织品瑕疵检测方法. *模式识别与人工智能*, 2019, 32(3): 268–277.
  - 5 兰瑜洁, 钟舜聪. 基于区域生长法的自适应图像分割的网眼织物瑕疵检测. *机电工程*, 2015, 32(11): 1513–1518.
  - 6 张波, 汤春明. 基于相对总变差模型与自适应形态学的织物瑕疵检测. *纺织学报*, 2017, 38(5): 145–149, 162.
  - 7 Wang T, Chen Y, Qiao MN, *et al.* A fast and robust convolutional neural network-based defect detection model in product quality control. *The International Journal of Advanced Manufacturing Technology*, 2018, 94(9): 3465–3471.
  - 8 Zhao ZX, Li B, Dong R, *et al.* A surface defect detection method based on positive samples. *Proceedings of the 15th Pacific Rim International Conference on Artificial Intelligence*. Cham: Springer, 2018. 473–481.
  - 9 Tabernik D, Šela S, Skvarč J, *et al.* Segmentation-based deep-learning approach for surface-defect detection. *Journal of Intelligent Manufacturing*, 2020, 31(3): 759–776. [doi: [10.1007/s10845-019-01476-x](https://doi.org/10.1007/s10845-019-01476-x)]
  - 10 Cai ZW, Vasconcelos N. Cascade R-CNN: Delving into high quality object detection. *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City: IEEE, 2018. 6154–6162.
  - 11 Ren SQ, He KM, Girshick R, *et al.* Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*. Montreal: NIPS, 2015. 91–99.
  - 12 He KM, Gkioxari G, Dollár P, *et al.* Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*. Venice: IEEE, 2017. 2961–2969.
  - 13 Lin TY, Dollár P, Girshick R, *et al.* Feature pyramid networks for object detection. *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu: IEEE, 2017. 2117–2125.
  - 14 Xie SN, Girshick R, Dollár P, *et al.* Aggregated residual transformations for deep neural networks. *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu: IEEE, 2017. 1492–1500.
  - 15 Dai JF, Qi HZ, Xiong YW, *et al.* Deformable convolutional networks. *Proceedings of the 2017 IEEE International Conference on Computer Vision*. Venice: IEEE, 2017. 764–773.
  - 16 Everingham M, Van Gool L, Williams CKI, *et al.* The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision*, 2010, 88(2): 303–338. [doi: [10.1007/s11263-009-0275-4](https://doi.org/10.1007/s11263-009-0275-4)]