

基于 YOLOv3-spp 的缺陷检测优化模型^①



曾 凯, 李 响, 贾建梅, 文继锋, 王 翔

(南京南瑞继保电气有限公司, 南京 211102)

通信作者: 曾 凯, E-mail: zengkai2@nrec.com

摘 要: 目前基于传统的机器视觉分析方法筛选后的 PCB 焊接缺陷图像还需要进行人工的复检流程, 工作量大导致视觉疲劳后容易出错. 为了改善这种现状, 本文设计应用 YOLOv3-spp 的目标检测算法来构建焊接缺陷检测模型. 为提升检测速度, 采用模型剪枝、模型蒸馏、模型量化等技术对检测模型进行压缩优化, 采用深度学习加速组件 OpenVINO 来加载压缩优化后的检测模型, 实现对 PCB 焊接缺陷图像的复检. 基于该优化算法设计了一种基于深度学习技术的 PCB 焊接缺陷检测识别系统. 它能快速、准确地识别焊接缺陷并定位缺陷位置, 解决了人工目检带来的效率低下、漏检误检率高等问题.

关键词: 深度学习; 缺陷检测; 目标检测算法; 模型压缩

引用格式: 曾凯, 李响, 贾建梅, 文继锋, 王翔. 基于 YOLOv3-spp 的缺陷检测优化模型. 计算机系统应用, 2022, 31(2): 213-219. <http://www.c-s-a.org.cn/1003-3254/8318.html>

Optimal Model for Defect Detection Based on YOLOv3-spp

ZENG Kai, LI Xiang, JIA Jian-Mei, WEN Ji-Feng, WANG Xiang

(NR Electric Co. Ltd., Nanjing 211102, China)

Abstract: At present, PCB welding defect images screened with traditional machine vision analysis methods still need manual reinspection, which is easy to make mistakes after visual fatigue due to heavy workload. In view of this, the study designs and applies the YOLOv3-spp object detection algorithm to build a welding defect detection model. For a higher detection speed, model pruning, model distillation, model quantization and other technologies are used to compress and optimize the detection model. OpenVINO, a deep learning acceleration component, is employed to load the compressed and optimized detection model for the reinspection of PCB welding defect images. With the help of this optimization algorithm, this study designs a PCB welding defect detection and identification system based on deep learning technology. It can quickly and accurately identify welding defects and locate the defects, addressing the low efficiency and high rates of missed detection and false detection caused by manual visual inspection.

Key words: deep learning; defect detection; object detection algorithm; model compression

随着印刷电路板行业的迅速发展, PCB 板的质量检测是各企业非常重视的环节. 目前, 主流的检测方法是采用人工目检的方式或者采用基于传统机器视觉的 PCB 缺陷检测 (automatic optical inspection, AOI) 方式. 人工目检的缺点是显而易见的, 不仅检测准确率不高, 还耗费大量的人力物力, 不是一种很好的应用方

案. AOI 的检测方式检测速度快、无损伤, 相比于人工目检, 其检测范围更广、检测精度更高. 目前 AOI 的检测方法大都依赖于传统的机器视觉技术, 王玉萍等^[1]对参考图像使用边缘检测算子提取特征, 与待测缺陷图像匹配得出缺陷区域. 陈寿宏等^[2]先利用方向梯度直方图 (HOG) 对待检测的焊点图像提取形状纹理等

^① 基金项目: 南瑞集团有限公司科技项目 (JS2001712)

收稿时间: 2021-04-11; 修改时间: 2021-05-11, 2021-05-25; 采用时间: 2021-06-02; csa 在线出版时间: 2022-01-17

特征,接着使用支持向量机(SVM)并选择合适的核函数进行初步分类,对误检焊点采用HOG+SVM多分类算法进行二次分类。

受限于传统机器视觉算法的局限性,AOI的检测方式还是需要通过复杂的参数调整与不断的人工修正来达到较好的检测效果,而且为了防止漏掉焊接异常的PCB,往往会将检测参数阈值设置较高值,这样会产生大量的焊接异常误报信息,此时需要经验丰富的生产人员去判断焊接异常信息是否确实为误报信息,会消耗大量的人力,且检测效率不高。

随着人工智能技术日益发展,以深度学习为代表的机器学习技术在图像识别领域得到广泛关注^[3]。将基于深度学习的图像分析技术应用在PCB缺陷检测领域具有非常广阔的前景。王永利等^[4]先采用差分算法提取图像中的缺陷区域,再设计一个结构简单的卷积神经网络用于缺陷特征的提取及分类,实现了PCB上缺陷的检测,但是该方法中缺陷区域检测任务及分类任务是割裂开的,且特征提取网络简单,对于复杂场景下的特征提取能力不足。孙灿等^[5]同样是采用差分算法来识别缺陷区域,其采用了结构相对更复杂的Inception-ResNet-v2分类网络来提取更多高层次的特征,但在部分图像中存在缺陷区域较小且形状相近的情况,实际应用识别准确率还有待提升。耿振宇^[6]提出了应用Faster R-CNN算法来对PCB表面进行缺陷检测,以解决传统网络对小目标的识别效果较差的问题,该算法可以同时进行分类与回归的任务,但是网络冗余度较高,推理效率方面不太理想,对硬件资源要求较高,部署应用的条件较为苛刻。

为了提升深度学习模型检测识别的准确度,如今主流的检测算法往往会采用复杂且深层次的网络结构,伴随而来的是庞大的计算量及硬件资源的占用,工业部署应用的条件比较苛刻。因此,为了更好的满足工业部署应用的要求,一方面需要采用推理效率高、检测效果好的深度学习检测算法,另一方面还需要对检测模型进行压缩优化。

深度学习模型的压缩优化技术^[7,8]是指利用神经网络参数的冗余性和网络结构的冗余性来精简模型,在不影响任务完成度的情况下,得到参数量更少、结构更精简的模型。

综合上述现状,本文以深度学习目标检测技术为核心思想,应用工业界综合表现优秀的YOLOv3-spp目标检测算法来构建焊接缺陷检测模型,并且为了提

升检测效率、降低部署成本进行了模型的裁剪优化。基于该优化算法设计了PCB焊接缺陷检测方案,开发了PCB-Checker焊接缺陷检测软件。

1 基于YOLOv3-spp优化模型缺陷检测技术

基于深度学习目标检测技术的智能图像分析功能是本软件的实现关键技术之一。缺陷检测模型的好坏直接影响到PCB板上焊接缺陷的识别准确度及识别速度,采用优秀的目标检测算法、融合以各种模型压缩优化和加速手段,可以极大的提升模型的健壮度、泛化能力及推理效率。

本文基于以下步骤实现缺陷检测模型的训练及部署,整体流程图如图1所示。

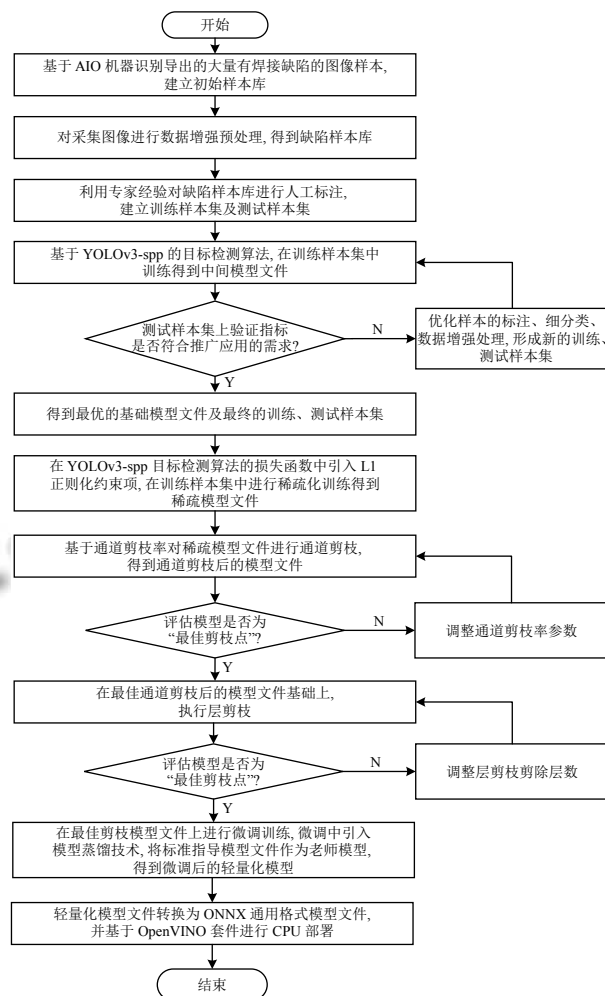


图1 基于深度学习的缺陷检测模型的训练及部署流程图

1.1 训练、测试样本集的建立

基于 AOI 机器识别导出的大量有焊接缺陷的图像

样本,建立初始样本库.对存在有焊接缺陷的图像样本进行人工分类.本文中,将焊接缺陷共分为以下13类:芯片虚焊、电阻电容虚焊、少锡、桥接、偏移、管脚偏移、缺件、反贴、电阻立碑、芯片立碑、电容立碑、侧立和缺锡.统计样本库中各分类的样本数量,梳理出样本库中的正负样本、难易样本.针对少分类、难识别的图像样本,通过复制图像后的随机数据增强方法来扩充样本得到分类相对均衡的训练样本库.

采用的数据增强方法包括光学变换、几何变换、增加噪声、数据源扩充等4大类方法.其中,光学变换包括亮度、对比度等调节方法;几何变换包括扩展、裁剪和镜像等操作方法;增加噪声表示在原始图像上增加一定的扰动,比如高斯噪声;数据源扩充表示将检测的目标背景进行替换.

利用专家经验对训练样本库中焊接缺陷的关键区域使用开源工具进行人工标注,标注后使用脚本转换形成YOLOv3-spp算法支持的标注文件.训练样本库按4:1的比例随机建立训练样本集和测试样本集.

1.2 标准化训练及模型评估

目标检测领域的深度学习方法^[9,10]主要分为两类:two-stage的目标检测算法;one-stage的目标检测算法.前者是先由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;后者则不用产生候选框,直接将目标边框定位的问题转化为回归问题处理.相对来说two-stage在检测准确率和定位精度上占优,one-stage在算法速度上占优.YOLO系列算法^[11]在实际应用中是one-stage目标检测方法中性能及精度最好的方法之一.YOLOv3^[12,13]是YOLO系列中成熟且优秀的算法,它整合了大量的计算机视觉前沿技术,能显著改善目标检测的性能.YOLOv3-spp算法在标准的YOLOv3算法基础上增加了SPP模块,该模块借鉴了空间金字塔的思想,通过SPP模块实现了局部特征和全局特征的融合,丰富了特征图的表达能力.尤其是对于YOLOv3这种复杂的多目标检测,YOLOv3-spp算法相比YOLOv3标准算法在精度等指标上都有几个百分点的提升.

本文采用Ultralytics官方提供的PyTorch框架下的YOLOv3-spp开源算法,在训练样本集中训练得到中间模型文件.

在测试样本集中,对中间模型文件进行模型评估,评估指标综合考虑深度学习模型的mAP(多分类平均

精度)、Precision(准确率)、Recall(召回率)等主流指标.如果评估指标未达到技术规范或推广应用的要求,针对指标效果不佳的分类采取各类优化方法,比如细分类、重新标注、数据增强等方式来重新建立训练样本集和测试样本集.通过反复迭代训练、评估的过程,得到最优的基础模型文件及训练样本库.

本文设置标准训练的epoch为250轮,训练结果如表1所示.本文表格中, P 表示准确率、 R 表示召回率、mAP@0.5表示在交并比为0.5时的多分类平均精度、Params表示参数量、Size of weights表示模型文件的大小、Infer_time表示单张图像在硬件配置为Tesla T4的GPU服务器上的推理耗时.

表1 标准训练模型的参数

参数	P	R	mAP@0.5	Params	Size of weights	Infer_time
值	0.976	0.949	0.944	62.61 M	250.77 MB	21.3 ms

1.3 稀疏化训练及模型评估

稀疏化训练就是精度和稀疏度的博弈过程,即需要保证稀疏后的模型在保持高精度的同时实现高稀疏度.

本文采用L1正则化施加到BN层的缩放因子 γ 上的方法来进行稀疏训练^[14].L1正则化可以对模型参数进行限制,产生稀疏化的模型,同时增加模型的泛化能力.在YOLO算法的损失函数中叠加了针对缩放因子 γ 的L1正则表达式,因此在训练过程中很多通道的缩放因子 γ 正则项都会逐渐趋于0.

设置epoch为350轮,在训练集上进行稀疏化训练.训练过程中BN层权值的分布情况如图2所示.图2(a)为正常训练时的BN权值变化图,总体上是在1附近的类正态分布;图2(b)为稀疏训练BN权值变化图,训练过程中大部分值逐渐被压到接近0,BN层对应通道输出值也近似于常量,因此非常适合于后续的剪枝过程.

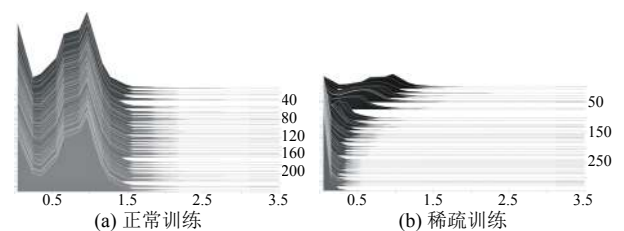


图2 正常训练和稀疏训练过程中的BN权值变化

稀疏化模型的评估结果如表2所示。

可以观察到,在稀疏化训练后,模型的 mAP@0.5 指标下降了3个点左右,模型参数量、推理耗时等指标基本不变。

表2 稀疏化训练模型的参数

参数	P	R	mAP@0.5	Params	Size of weights	Infer_time
值	0.969	0.919	0.911	62.61 M	250.99 MB	21.3 ms

1.4 对稀疏化模型执行通道剪枝

模型剪枝^[15,16]包括通道剪枝和层剪枝两种类型,两者都是利用BN层的 γ 系数评价对应通道的重要性来达到剪枝的目的。因此,稀疏训练是否充分将直接影响剪枝的效果。

通道剪枝的大致原理是,统计所有参与剪枝层的BN参数 γ 权重值并进行排序,依据剪枝率确定参数阈值,将所有小于阈值的通道全部减掉,如果有依赖则将依赖的对应部分也剪掉。如果一层中所有的通道都需要被移除,为了保证网络结构不变,需保留剪枝参数所设置的最小通道值。

本文设置初始通道剪枝率为0.5,对剪枝后的模型在测试样本集中进行评估,与剪枝前的模型比较 P 、mAP@0.5等综合指标。剪枝率先以0.1的级差进行尝试性剪枝后评估,到了临界值时再以0.01的级差进行微调,迭代剪枝并评估找到最佳的通道剪枝率参数。即以牺牲最小的准确率指标为代价,剪除更多的通道来提升模型性能,得到最佳的通道剪枝率参数及最佳的通道剪枝模型文件。

本文中,不同的通道剪枝率下检测模型的参数如表3所示。

可以观察到,稀疏化模型在剪枝率为0.66时,通道剪枝过程达到“最佳剪枝点”,mAP@0.5指标仅下降0.01,参数量及模型文件大小均减少70%左右,单张图片推理耗时下降为原来的60%左右。

表3 不同通道剪枝率下模型的参数

Pruning rate	mAP@0.5	Params (M)	Size of weights (MB)	Infer_time (ms)
base(0.00)	0.931	62.64	250.99	21.2
0.70	0.099	15.14	60.70	11.8
0.68	0.865	16.63	66.66	12.2
0.66	0.921	18.24	73.10	12.5
0.65	0.927	19.00	76.14	12.7
0.60	0.924	21.55	86.34	13.3
0.50	0.927	25.16	100.79	14.3

1.5 对通道剪枝后模型执行层剪枝

通道剪枝能大大减小模型的参数和计算量,降低模型对资源的占用。层剪枝可以进一步减小计算量,提高模型的推理速度。

YOLOv3层剪枝是对每一个残差单元中shortcut捷径连接的前一个CBL(Conv+BN+Leaky_ReLU)的BN层进行评价,对各层的 γ 均值进行排序,取最小的进行层剪枝。为保证YOLOv3结构完整,每剪一个shortcut捷径连接结构,会同时剪掉一个shortcut和它前面的两个CBL卷积层结构。

本文以全局剪枝率为0.66时生成的通道剪枝模型文件为基础执行层剪枝。分别尝试不同层剪枝shortcut个数时的模型指标,如表4所示。

表4 通道剪枝率0.66时,不同剪层个数下模型的参数

Shortcut 剪除数	P	R	mAP@0.5	Params (M)	Size of weights (MB)	Infer_time (ms)
12	0.231	0.004	0.004	13.95	55.88	9.5
8	0.585	0.180	0.179	15.66	62.76	10.9
7	0.753	0.316	0.314	16.34	65.49	11.0
6	0.829	0.486	0.483	16.55	66.29	11.4
5	0.893	0.639	0.633	17.22	69.01	11.8
4	0.952	0.749	0.743	17.33	69.42	12.0

可以观察到,随着shortcut剪除的个数增加, P 、 R 和mAP@0.5都会不同程度的下降,当剪除shortcut个数为8时, R 和mAP@0.5出现了大幅度的下滑,且差距比较大,如果再提升剪枝层数,效果非常差。为了保持最大化的剪枝效果,采用剪除7个shortcut的方式,即为最佳的层剪枝点。虽然各项指标都下降比较明显,但是可以在后续的微调训练中恢复。经过层剪枝后的模型,参数量及模型文件大小相比标准模型减少74%左右,单张图片推理耗时下降为标准模型的52%左右。

1.6 引入蒸馏技术的微调训练

微调训练的目的在于恢复剪枝后的模型精度指标,保证剪枝前后的模型评估指标总体上相差不大。

模型蒸馏^[17,18]是指将拥有复杂网络结构的老师模型中学习到的知识迁移到相对简单网络结构的学习模型中,在蒸馏的过程中,小模型学习到了大模型的泛化能力,保留了接近于大模型的性能。

对最佳的剪枝模型文件进行微调训练,并引入模型蒸馏技术,将YOLOv3-spp标准化训练模型作为老师模型,将最佳的剪枝模型作为学生模型,通过老师模型指导学生模型进行微调训练的方式,得到最终的轻量化

模型文件. 经过微调训练后, 剪枝后的模型文件参数量保持不变, 但是 $mAP@0.5$ 等指标都有了一定程度的恢复.

本文以全局剪枝率为 0.66、层剪枝为 7 个 shortcut 后生成的模型文件为基础执行微调训练, 设置 epochs 为 80 轮, 训练得到轻量化模型. 各阶段的模型文件在测试样本集上的评估指标, 如表 5 所示.

表 5 各阶段模型指标比较表

Stage	P	R	mAP@0.5	Params (M)	Size of weights (MB)	Infer_time (ms)
标准	0.976	0.949	0.944	62.61	250.77	21.3
稀疏	0.969	0.919	0.911	62.61	250.99	21.3
剪枝	0.958	0.929	0.920	16.34	65.49	10.8
微调						

从表 5 中对比数据可以看出, 稀疏模型相较于标准模型在 R、 $mAP@0.5$ 下降 3 个多点, 但是经过剪枝及微调处理后指标会有所提升. 轻量化模型在 P、R 及 $mAP@0.5$ 等指标上比标准模型仅下降 2 个点左右, 但是参数量及模型大小减少 74% 左右, 推理时间下降为标准模型的 50% 左右, 取得了较好的模型压缩效果.

图 3 展示了剪枝微调后模型在测试样本集上的焊接缺陷推理情况.

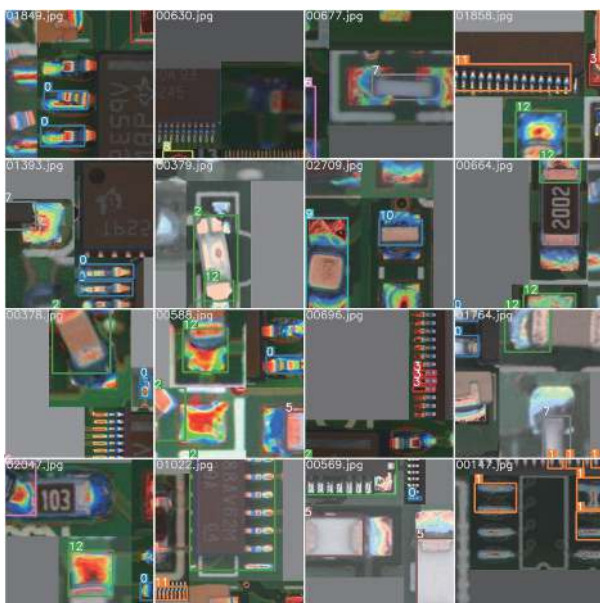


图 3 推理模型在测试集上的推理示例

1.7 模型转换及模型量化

经过微调训练后的轻量化模型文件的格式是 PyTorch, 在部署应用中, 先用脚本将其转换为通用的 onnx 格式

模型文件.onnx (开放神经网络交换) 是一种针对深度学习所设计的开放式的通用文件格式, 用于存储训练好的模型, 它使得不同的人工智能框架 (如 PyTorch、TensorFlow、Mxnet 等) 可以采用相同格式存储模型数据并交互, 支持模型在不同框架之间进行转移.

采用 Intel 推出的深度学习加速组件 OpenVINO 的 Model Optimizer 模块对模型进行量化优化处理, 可以根据需要将模型中参数精度由 32 位浮点型量化成 16 位浮点型, 模型大小可以缩小一半, 同时保证有限的精度损失, 以进一步提升模型在检测应用时的推理速度. 本文为了保证检测的准确率, 未进行模型量化处理.

1.8 检测模型的部署

本文调用 OpenVINO 组件提供的 Inference Engine C++推理接口函数, 加载检测模型文件, 以待检测的初筛局部图像为输入, 经过检测模型的前向推理运算, 得到缺陷分类及缺陷位置的信息, 完成 AI 推理的应用及展示. 一般而言, 剪枝微调后的模型在 OpenVINO 加速框架下, 使用 intel CPU、GPU 等硬件平台进行推理还可以获得 30% 以上的速度提升.

2 缺陷检测软件设计与实现

2.1 软件模块和场景设计

图 4 展示了 PCB-Checker 软件的模块划分及功能分类, 支持缺陷信息展示、缺陷图浏览、缺陷图智能分析、缺陷图定位和缺陷过滤等功能.

PCB-Checker 软件典型的操作流程如图 5 所示: ① 从 AOI 检测仪器上导出初筛的缺陷检测图像目录, 利用工程管理模块新建工程; ② 点击触发扫描流程, 分析并展示待检目录下异常 PCB 的板卡型号、异常元件编号、异常引脚编号等信息, 并开启基于深度学习的分析子线程; ③ 在所设置的间隔时间内 (默认 10 s), 对一部分图像运行推理输出结果, 并刷新表格化界面的内容; ④ 持续对未进行过智能分析的图像进行推理输出, 直至所有图像都完成分析, 结束分析子线程; ⑤ 导入初筛缺陷图像所属 PCB 的整板图像文件及元件位置坐标文件, 标记若干组元件名称及元件像素坐标, 完成整板图的导入; ⑥ 在表格化的缺陷信息展示界面中, 对智能分析后确实存在缺陷的条目进行校核. 可以利用缺陷类型过滤功能进行过滤展示, 结合缺陷局部图在 PCB 整板图中的定位展示功能, 完成 PCB 焊接缺陷的定位识别.

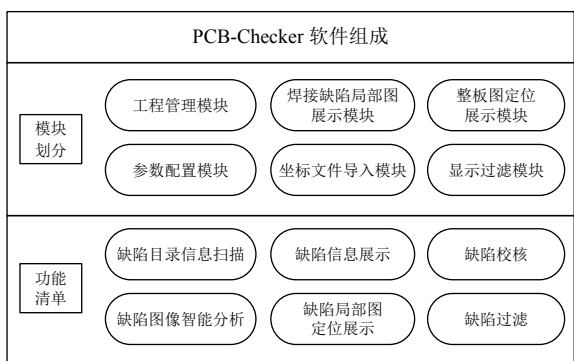


图4 PCB-Checker 软件模型划分及功能列表

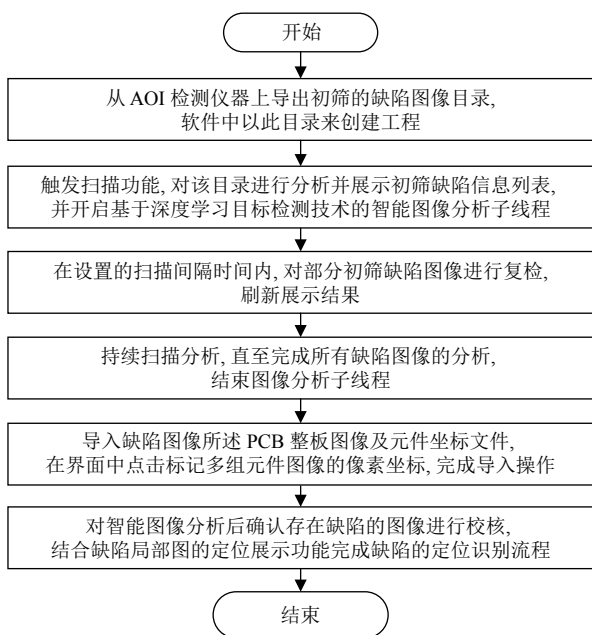


图5 PCB-Checker 软件典型操作流程

2.2 基于一元线性回归算法的缺陷位置定位

由于 PCB 板上元件的集成度非常高, 元件尺寸很小, AOI 检测形成的缺陷成像一般是缺陷区域的特写, 因此需要将缺陷局部图定位至 PCB 整板图中的具体位置并进行可视化的展示, 以提升生产效率. 本系统基于一元线性回归算法实现了局部缺陷图的定位显示功能.

PCB 板在设计完毕后支持导出一张整板图像文件及一个包含各元件坐标分布的文本文件. 在 PCB-Checker 软件中导入该整板原图或者是经过裁切、翻转、缩放等处理后的整板图, 并同时导入元件坐标文件, 以 PCB 板类型和正反面标识为索引, 存储每一对导入的信息到配置文件中.

基于导入的整板图及元件坐标文件, 提供可视化界面允许在 PCB 整板图中鼠标左键单击标记若干枚元件的位置坐标, 以获取多组包含元件名称及坐标的信息, 如图 6 所示. 若干组必须是两组及以上的信息, 一般来说记录的信息越多, 拟合出来的函数系数能更具有泛化能力, 但是过多也存在过拟合的情况, 建议是标记 4 组或者 5 组信息即可.

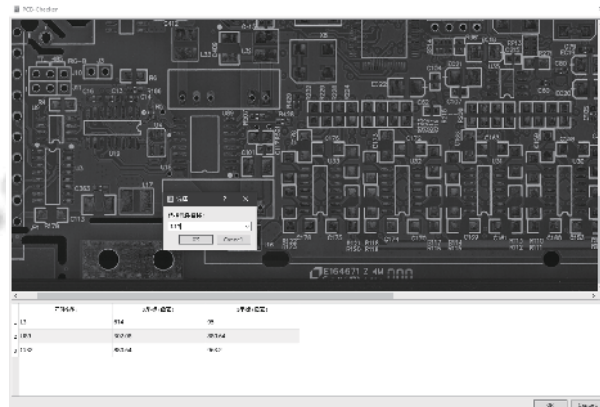


图6 PCB 整板图中元件坐标标记界面

采用一元线性回归算法, 推算出 PCB 整板图像中元件的像素坐标与坐标文件中记录的坐标之间的关系表达式为:

$$\begin{cases} y_p = k_1 \times y_f + b_1 \\ x_p = k_2 \times x_f + b_2 \end{cases} \quad (1)$$

其中, y_p 表示图像中某元件的像素 y 坐标值; y_f 表示坐标文件中某元件的像素 y 坐标值; x_p 表示图像中某元件的像素 x 坐标值; x_f 表示坐标文件中某元件的像素 x 坐标值; k_1 、 k_2 、 b_1 、 b_2 为需要进行计算的一元线性回归方程的参数值.

从标记元件的名称及坐标信息中, 通过若干组包含元件名称及像素坐标的散点信息, 分别拟合出 y 坐标、 x 坐标的线性关系, 计算得到的 k_1 、 k_2 、 b_1 、 b_2 参数值.

从 PCB 焊接缺陷局部图像所对应的缺陷描述文件中, 获取该局部图像所对应的元件名称. 再从 PCB 元件坐标文本文件中查找到该名称所对应的坐标位置, 即 y_f 、 x_f 的值. 基于推导出的坐标关系表达式及元件名称、坐标位置, 计算出该元件在 PCB 整板图像中的像素坐标位置, 即 y_p 、 x_p 的值.

2.3 软件主界面

PCB-Checker 提供了界面并列展示 PCB 焊接异常

局部图及其在 PCB 整板图中的定位位置,达到可以在 PCB 整板图上快速定位异常局部图位置的目的.软件采用组件式架构设计^[19,20],其主界面如图 7 所示.



图 7 PCB-Checker 软件主界面

3 结论与展望

本文介绍了一种基于深度学习技术的 PCB 焊接缺陷检测识别系统.系统以 YOLOv3-spp 的目标检测算法为基础,设计采用模型剪枝、模型蒸馏及模型量化等手段来压缩模型,以牺牲有限准确率为代价大幅提升了 AI 缺陷检测的速度.另外,本系统还基于一元线性回归算法实现了焊接缺陷局部图的定位展示功能.PCB-Checker 系统界面简洁、运行稳定,在南瑞继保电气有限公司智能制造中心多条电力 PCB 生产线进行了实际应用,结果表明该软件提升了生产效率和质量管理能力.

参考文献

- 王玉萍,郭峰林.基于机器视觉的 PCB 板缺陷检测系统关键技术研究.科技通报,2017,33(1): 101-105.
- 陈寿宏,赵爽,马峻,等.基于多特征的 SVM 多分类 PCB 焊点缺陷检测方法.激光杂志,2019,40(6): 21-26.
- 卢宏涛,张秦川.深度卷积神经网络在计算机视觉中的应用研究综述.数据采集与处理,2016,31(1): 1-17.
- 王永利,曹江涛,姬晓飞.基于卷积神经网络的 PCB 缺陷检测与识别算法.电子测量与仪器学报,2019,33(8): 78-84.
- 孙灿,邓小颖,李扬,等.基于改进 Inception-ResNet-v2 的 PCB 缺陷检测.信息技术,2020,44(9): 33-36.
- 耿振宇.基于 Faster R-CNN 的 PCB 表面缺陷检测.科学技术创新,2021,(11): 86-87. [doi: 10.3969/j.issn.1673-1328.2021.11.036]
- 雷杰,高鑫,宋杰,等.深度网络模型压缩综述.软件学报,2018,29(2): 251-266. [doi: 10.13328/j.cnki.jos.005428]
- 葛道辉,李洪升,张亮,等.轻量级神经网络架构综述.软件学报,2020,31(9): 2627-2653. [doi: 10.13328/j.cnki.jos.005942]
- 张慧,王坤峰,王飞跃.深度学习在目标视觉检测中的应用进展与展望.自动化学报,2017,43(8): 1289-1305.
- 姚群力,胡显,雷宏.深度卷积神经网络在目标检测中的研究进展.计算机工程与应用,2018,54(17): 1-9. [doi: 10.3778/j.issn.1002-8331.1806-0377]
- Redmon J, Divvala S, Girshick R, *et al.* You only look once: Unified, real-time object detection. Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: IEEE, 2016. 779-788.
- Redmon J, Farhadi A. YOLOv3: An incremental improvement. arXiv: 1804.02767, 2018.
- 张明伟,蔡坚勇,李科,等.基于 DE-YOLO 的室内人员检测方法.计算机系统应用,2020,29(1): 203-208. [doi: 10.15888/j.cnki.csa.007240]
- Liu Z, Li JG, Shen ZQ, *et al.* Learning efficient convolutional networks through network slimming. Proceedings of the 2017 IEEE International Conference on Computer Vision. Venice: IEEE, 2017. 2755-2763.
- He YH, Lin J, Liu ZJ, *et al.* AMC: AutoML for model compression and acceleration on mobile devices. Proceedings of the 15th European Conference on Computer Vision. Munich: Springer, 2018. 815-832.
- Lin SH, Ji RR, Li YC, *et al.* Accelerating convolutional networks via global & dynamic filter pruning. Proceedings of the 27th International Joint Conference on Artificial Intelligence. Stockholm: IJCAI, 2018. 2425-2432.
- Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. Computer Science, 2015, 14(7): 38-39.
- Lopes RG, Fenu S, Starmer T. Data-free knowledge distillation for deep neural networks. arXiv: 1710.07535, 2017.
- 陈宏君,刘克金,张磊,等.跨平台人机交互软件组件集成框架.计算机系统应用,2015,24(11): 52-57. [doi: 10.3969/j.issn.1003-3254.2015.11.008]
- 贺敏,陈宏君.组件开发架构及在继电保护配置软件中的应用.软件工程,2016,19(1): 35-38. [doi: 10.3969/j.issn.1008-0775.2016.01.013]