

基于云-边缘协同计算的表面缺陷检测系统研究

梁程,薛建彬

(南京航空航天大学机电学院,江苏南京210016)

摘要:基于现有表面缺陷检测系统所存在的实时检测难、硬件要求高等问题,提出一种基于云计算与边缘协同计算的表面缺陷检测系统。将轻量化改进后的YOLOv4缺陷检测算法模型部署到边缘端嵌入式设备中,在边缘端完成对表面缺陷的检测,并在边缘端和云端设备部署KubeEdge框架进行通信和管理。通过案例验证该系统不仅能够满足检测实时性的要求,还能够提取缺陷检测关键信息,同时便于部署在价格低廉的嵌入式设备。

关键词:云计算;边缘计算;表面缺陷检测;深度学习算法

中图分类号:TP183;TP391.41 **文献标志码:**A **文章编号:**1001-2257(2022)02-0065-06

Research on a Surface Defect Detection System Based on the Cloud-edge Computing

LIANG Cheng, XUE Jianbin

(College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: Existing surface defect detection system has a series of problems such as the difficulty of real-time detection and the high hardware requirements. To solve these problems, a surface defect detection system is proposed using the cloud-edge computing. The system deploys the lightweight YOLOv4 defect detection algorithm model to the edge-end embedded device, completes the detection of surface defects at the edge-end, and deploys the KubeEdge framework on the edge and cloud devices for communication and management. Through case verification, the proposed system can not only meet the requirements of real-time detection, but also extract key information about defect detection, and is easy to deploy in low-cost embedded devices.

Key words: cloud computing; edge computing; surface defect detection; deep learning algorithm

0 引言

表面缺陷一般是指产品表面局部物理或化学性质不均匀的区域,如工件的划痕、裂纹、毛边和污点等^[1]。表面缺陷的存在不仅影响产品美观,还可能影响产品使用性能,甚至存在安全隐患。因此,在工业生产过程中应及时检测出产品所存在的表面缺陷并分析出缺陷产生的原因。

传统的表面缺陷检测采用人工目检对工件进行查验,人工检测存在着检测精度不足和检测效率

低下等一系列问题。随着工业自动化技术的发展,机器视觉设备逐渐代替人类进行缺陷检测工作,相比于人工目检,基于机器视觉的表面缺陷检测具有可靠性高、检测准确率高、检测速度快以及综合成本低等一系列优点,但也存在难以识别微小瑕疵等问题。近些年,随着人工智能的发展,深度学习模型也在表面缺陷检测领域取得成功,常用的深度学习目标检测模型已能够满足工业生产中缺陷检测的精度要求,但是多数模型检测速率达不到实时检测要求。

收稿日期:2021-08-16

作者简介:梁程(1997-),男,江苏扬州人,硕士研究生,研究方向为物联网、缺陷检测;薛建彬(1970-),男,江苏南京人,副教授,研究方向为机器人应用、智能化制造。

近些年,工业物联网迅速发展,物联网技术通过传感器进行数据采集并传输到云端进行数据分析和逻辑控制,能够很好地存储分析生产过程中的关键数据。但随着工业物联网的不断发展,传感器所产生的数据量也海量爆炸,传统云计算模式很难满足实时检测的要求。基于上述问题,以边缘计算模型为核心的面向网络边缘设备所产生数据计算的边缘大数据处理应运而生,其与现有以云计算模型为核心的集中式大数据处理相结合,二者相辅相成,很好地解决了工业物联网所存在的问题。Wang等^[2]提出了一种基于云计算与边缘计算协同的智能表面检测系统,将表面检测计算任务部署到边缘端,避免了数据泄露的风险,同时保证了计算的实时性。尹子会等^[3]提出了基于云计算与边缘计算协同的变电站设备典型视觉缺陷检测系统,与传统模式直接上传到云端计算相比,传输量减少 90% 以上,同时检测速率也有较大提高。上述研究将云计算与边缘计算融合的模式和检测行业相结合,取得不错的成绩,但所采用的深度学习模型都需要采用多块 GPU 进行训练推理,对硬件成本要求极高。

本文提出一种基于边缘计算的表面缺陷检测系统,通过基于深度学习框架的边缘端嵌入式设备对生产产品进行实时的表面缺陷检测;同时对边缘端和云端进行调度和管理,搭建边缘端和云端的数据通道将检测关键信息上传至云端。

1 基于云-边缘协同计算的表面缺陷检测方案

传统的表面缺陷检测系统方案有 2 种,分别是本地计算和云计算。本地计算从图像采集到图像缺陷检测,再到检测结果分析全部在本地设备实现,因此无需进行网络通信,不存在检测延时的问题。但是由于检测结果的数据存储和计算都在本地进行,数据存储存在安全和隐私风险,同时由于深度学习模型对本地设备硬件要求极高,检测成本也会急剧上升。

使用云计算方案,传感器等设备会将所有数据上传至云端,随后云端进行数据处理、模型训练和数据计算等一系列工作,之后再检测结果返回至终端。云计算解决了本地计算数据存储安全的问题,能够更好地进行大数据分析。但是在推理计算过程中,云计算数据需要在通过网络进行通信,造成极大的响应时延。为了解决上述方案所存在的

问题,本文提出了基于云-边缘协同计算的表面缺陷检测系统方案。该系统由终端设备、边缘端设备和云端设备 3 部分组成,检测系统总体方案如图 1 所示。

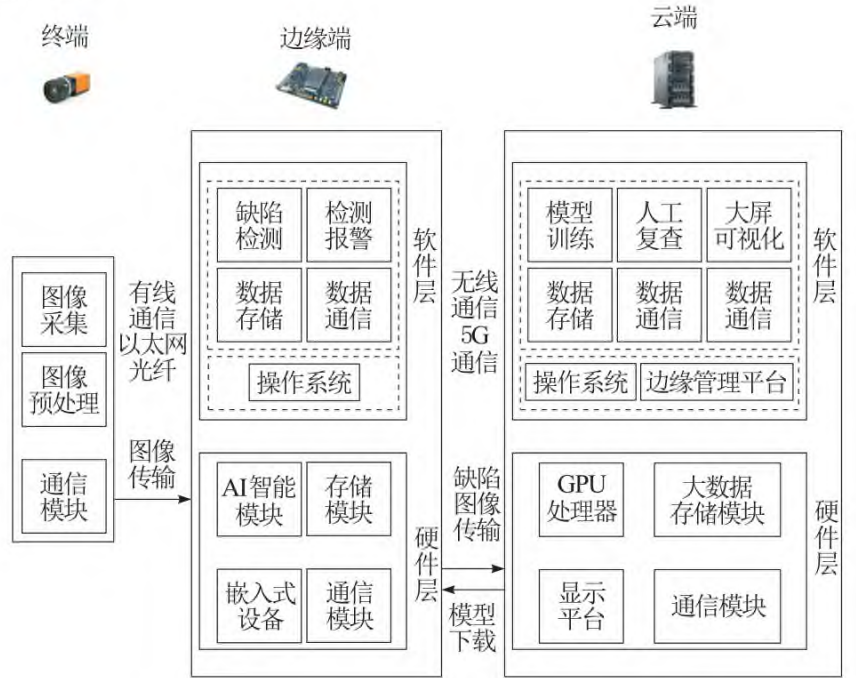


图 1 基于云-边缘协同计算的检测系统总体方案

a.终端。终端设备为实时采集产品图像的工业相机,其不具备计算推理能力,主要功能为图像采集和图像预处理。终端设备与边缘端设备通过以太网光纤进行有线通信,将所采集到的图像传输至边缘端设备。

b.边缘端。边缘端设备选择搭载 AI 芯片的嵌入式设备,部署轻量的深度学习模型,能够满足检测实时性的要求,对检测到的表面缺陷进行及时报警。边缘端设备配备数据存储模块和数据通信模块,接收到终端所采集的图像;部署边缘端软件平台对应用进行管理和调度,同时完成与云端设备的通信。

c.云端。由于边缘端为嵌入式设备,受硬件的性能与成本限制,仅能进行轻量化模型的计算,而难以完成模型的训练,因此云端需要搭载高性能 GPU 处理器以完成模型的训练任务。同时,云端设备搭载大数据存储和数据通信模块,接收边缘端设备传输的图像,部署边缘管理平台。

本文的研究重点在于边缘端与云端,即云端边缘端协同功能、通信功能和边缘端表面缺陷检测功能。

2 基于云-边缘协同计算的表面缺陷检测系统功能实现

2.1 云-边缘协同计算平台架构

云-边缘协同计算的表面缺陷检测系统需要完

成云端与边缘端之间的调度和管理,同时还需提高边缘端设备的可靠性、安全性和资源使用率。近年来,虚拟化技术 Docker 容器和 Kubernetes 容器编排引擎广泛应用于云计算与边缘计算领域。Docker 容器通过 Linux 系统中的 CGroup 和 NameSpace 系统隔离技术将应用解耦为多个独立的功能模块,使得应用部署的磁盘空间占用率更小,启动速率更快;将上层应用与底层的基础架构分离,能够保证在不同的操作系统上运行,其架构如图 2 所示。Kubernetes 容器编排引擎用于对容器的部署、管理、扩容以及运维,通过资源操作入口 APIServer 组件里的应用程序编程接口 Restful API(application programming interface)完成各模块的管理、通信以及控制,通过 Kubelet 组件完成对节点的管理和控制并与控制平面 APIServer 组件进行数据交互,Kubernetes 架构如图 3 所示。

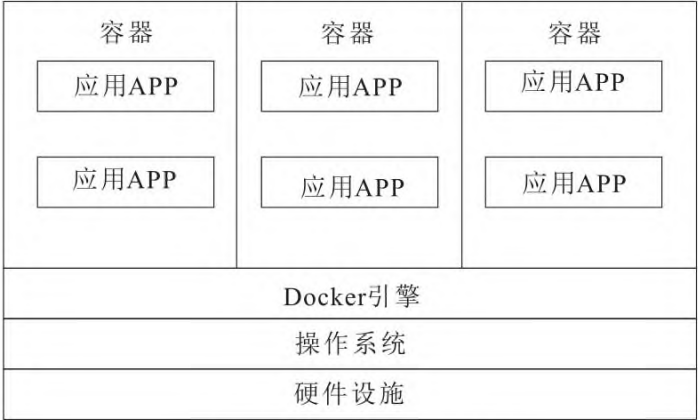


图 2 Docker 容器框架

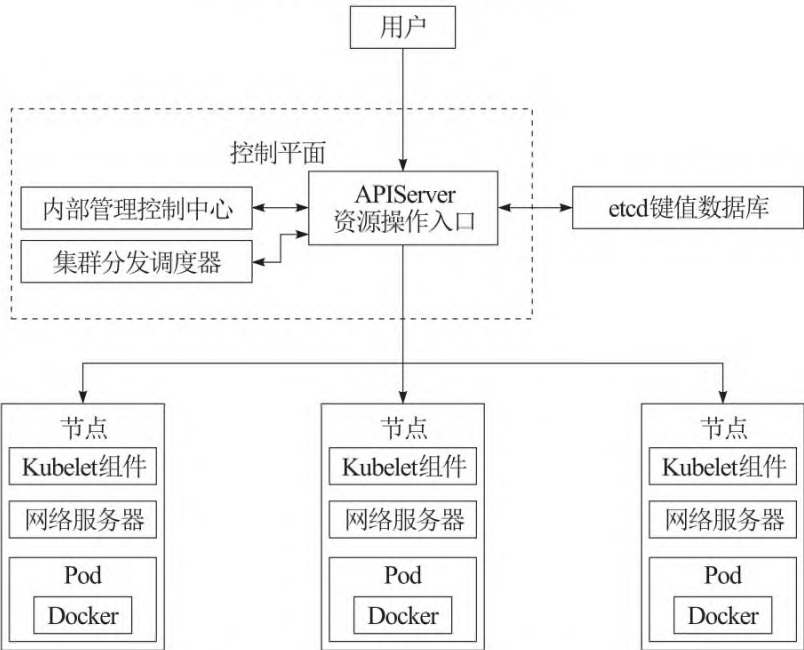


图 3 Kubernetes 框架

云-边缘协同计算的表面缺陷检测系统采用基于 Kubernetes 技术的 KubeEdge 平台架构。如图 4 所示,KubeEdge 平台架构分为云端、边缘端和设备端 3 部分,通过将 Kubernetes 的功能扩展部署到边缘端完成云端与边缘端设备之间的协同、调度和管理工作。

a.云端组件。云端设备部署 Kubernetes 控制

平面,包含云端通信中心 CloudHub 组件、边缘管理控制器和设备管理控制器。其中,CloudHub 实质为套接字服务器,负责完成与边缘端设备 EdgeHub 的数据通信;边缘管理控制器负责管理边缘端设备,同时负责边缘元数据在边缘端和云端之间的同步;设备管理控制器负责管理设备,同时负责设备元数据的同步工作。

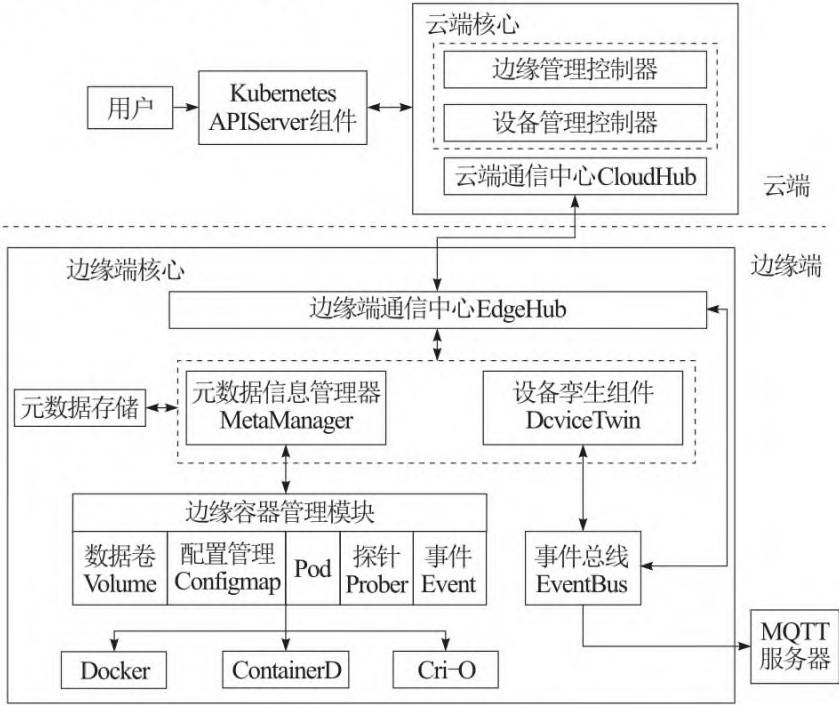


图 4 KubeEdge 架构

b.边缘端组件。边缘端设备主要包含边缘端通信中心 EdgeHub 组件、Edged 边缘节点管理容器等组件。EdgeHub 组件与 CloudHub 相对应,是一个套接字客户端,用于负责接收云端的资源,同时将设备信息和缺陷图像等数据传输至云端设备分析存储;元数据管理器 MetaManager 组件用于通信中心和容器组件之心的信息处理,用于管理元数据;设备孪生 DeviceTwin 组件通过事件总线 and 边缘端通信中心同步设备端所采集到的数据,同时管理和处理设备端的元数据;边缘容器管理模块相当于轻量化的 Kubelet 组件,主要用于在边缘端部署容器,管理容器的生命周期。

2.2 边缘-云数据通信

云-边缘融合计算的表面缺陷检测系统中边缘端与云端的数据通信主要包括图像的传输、模型应用的下发,同时还包含云端和边缘端的参数信息。如图 5 所示,边缘端和云端创建 HTTP 协议栈,在传输层之上提供 TCP 接口;同时,在传输层上分别创建服务器代理和客户端代理,监听和转发相应的请求,实现边缘端和云端之间的定向双工通信。通信的具体步骤为:

a.客户端代理与服务器代理之间通过 3 次握手建立 TCP 连接。

- b.客户端代理通过 TCP 接口监听 HTTP 请求,接收到请求后将该请求转发给相应的(云端或边缘端)服务器。
- c.服务器代理接收到请求后根据请求进行相应操作,并发送答复给相应的客户端代理。
- d.如果客户端再次监听到 HTTP 请求,将从步骤 b 开始执行。

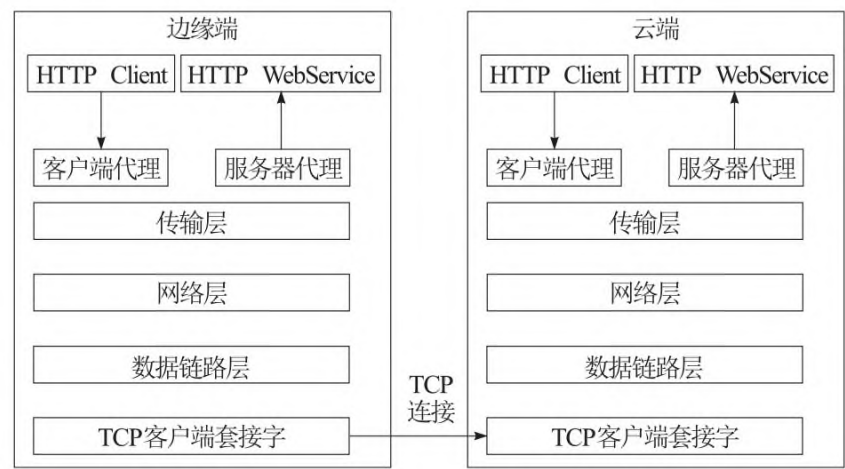


图 5 边缘端云端通信流程

2.3 基于轻量化 YOLOv4 算法的表面缺陷检测

常见的缺陷检测算法可以划分为以 YOLO^[4]为代表的一阶段(one stage)网络算法和以 Faster R-CNN^[5]为代表的二阶段(two stage)网络算法。二阶段网络在检测中首先要生成候选框(region proposal)判断区域内是前景还是背景,随后再进行目标的分类,此类算法检测精度较高但检测速度缓慢,往往难以满足实时性的要求;而一阶段网路则是将目标检测问题转化为回归问题,直接进行预测缺陷的位置并进行分类,此类算法不仅有着较好的检测准确率,同时检测速度较快能够很好地满足实

际应用场景中对实时性的要求。YOLOv4 算法模型由输入端、主干特征提取网络 CSPDarknet53^[6]、加强特征提取网络 SPP(spatial pyramid pooling)模块^[7]、路径聚合网络 PANet(path aggregation network)模块^[8]和 YOLO Head 回归预测网络构成,由于集成了许多算法技巧,YOLOv4 检测精度相比前代 YOLOv3 算法有了卓越的提升。但由于 YOLOv4 算法的主干特征提取网络 CSPDarknet 结构复杂,网络的训练与推算对硬件要求高,难以部署到计算能力有限的边缘嵌入式设备。本文从轻量化角度考虑,将 YOLOv4 算法的主干特征提取网络 CSPDarknet 改为 MobileNet 网络,改进后主干特征网络的结构如图 6 所示。MobileNet 网络是 Google 针对嵌入式设备所提出的一种轻量化的神经网络,无需价格昂贵的设备即可完成深度学习的训练与计算。

在表面缺陷检测网络训练的过程中,轻量化 YOLOv4 模型首先将采集到的缺陷图像调整大小为 416×416,在输入端对原始图像进行数据增强并传入 MobileNet 网络进行特征提取。通过 SPP 网络增加特征网络的感受野,通过 PANet 网络进行上下采样和不同尺度的特征融合,最终将图像划分为 13×13、26×26、52×52 的特征网格,每个尺寸的网格都会生成 9 个先验框。随后,YOLO Head 回归预测网络对不同尺寸特征网格进行位置、类别和置信度预测,并通过与真实值进行对比通过损失函数不断调整,整个检测步骤如图 7 所示。

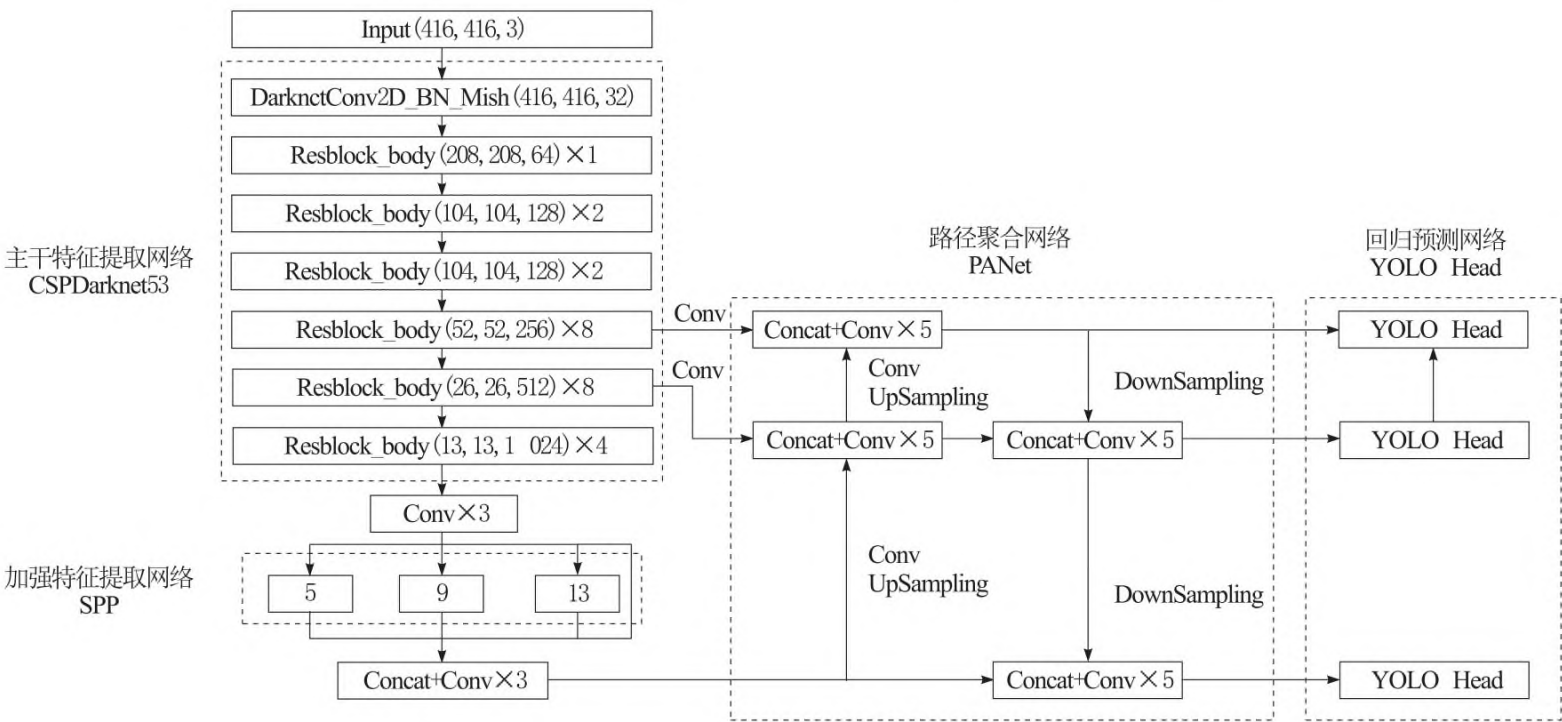


图 6 轻量化 YOLOv4 的网络架构

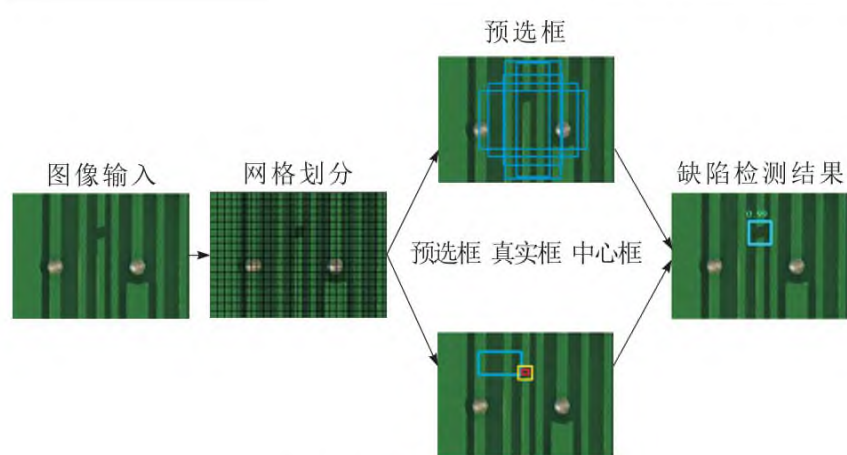


图7 缺陷检测过程

改进后的 YOLOv4 的损失函数分为 3 个部分,分别为位置损失、目标分类损失和置信度损失。位置损失函数 YOLOv4 采用 CIOU^[9] 作为损失函数,相比之前的损失函数 IOU, CIOU 考虑到预测框和目标框的重叠面积、中心距离和长宽比,同时还将预测框和目标框的长宽比考虑入内, CIOU 损失函数公式为:

$$IOU = \frac{|A \cap B|}{|A \cup B|} \quad (1)$$

$$\text{loss}_{\text{location}} = \text{CIOU} = IOU - \frac{\rho^2(b, b^{\text{gt}})}{c^2} - \alpha v \quad (2)$$

b, b^{gt} 分别为预测框和目标框的中心点; ρ^2 为 2 个中心点之间的欧氏距离; c 为能够包含预测框和目标框的最小闭包区域的对角线距离。其中:

$$\alpha = \frac{v}{1 - IOU + v} \quad (3)$$

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

w, w^{gt} 分别为预测框和目标框的宽度; h, h^{gt} 分别为预测框和目标框的高度。

置信度损失函数采用二元交叉熵损失函数,分为有目标置信度损失和无目标置信度损失 2 部分,即

$$\begin{aligned} \text{loss}_{\text{confidence}} = & \lambda_{\text{coord}} \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{\text{obj}} \left[\hat{C}_i \log(C_i) + \right. \\ & \left. (1 - \hat{C}_i) \log(1 - C_i) \right] - \lambda_{\text{noobj}} \sum_{i=0}^{K \times K} \sum_{j=0}^M I_{ij}^{\text{noobj}} \left[\hat{C}_i \log(C_i) + \right. \\ & \left. (1 - \hat{C}_i) \log(1 - C_i) \right] \end{aligned} \quad (5)$$

$K \times K$ 为像特征提取后划分的网格尺寸; M 为每个网格所生成的先验框个数; C_i 为置信度得分; \hat{C}_i 为预测框和目标框的 IOU; λ 为损失函数加权参数; λ_{coord} 为最高权重; λ_{noobj} 为未探测到目标时所取的最低权重。目标分类损失函数对每一个类别计算二元交叉熵损失函数并求和,即

$$\text{loss}_{\text{class}} = \sum_{i=0}^{K \times K} \sum_{c \in \text{classes}} \left[\hat{p}_i \log(p_i(c)) + \right.$$

$$\left. (1 - \hat{p}_i(c)) \log(1 - p_i(c)) \right] \quad (6)$$

c 为所识别的类别; $p_i(c)$ 为预测类别的概率; $\hat{p}_i(c)$ 为真实概率。最终的缺陷检测损失函数为

$$\text{loss} = \text{loss}_{\text{location}} + \text{loss}_{\text{confidence}} + \text{loss}_{\text{class}} \quad (7)$$

缺陷检测模型训练的过程中,通过不断减小损失函数的大小,即可使得目标预测框不断逼近真实框,提高检测类别和置信度的准确率。

3 实际案例与结果分析

印刷电路板(printed circuit board, PCB)是重要的电子部件,是电子元器件电气连接的载体。随着电子产品行业的快速发展,PCB 电路板也不断向高性能、高集成以及微元化方向演变,对 PCB 电路板进行表面缺陷检测也面临着巨大的挑战^[10]。本文选取 PCB 电路板裸板的表面缺陷作为实际案例,检测系统的可行性。选用北京大学智能机器人开放实验室发布的 PCB 缺陷数据集,该数据集包含漏孔、鼠咬、断路、短路、毛刺和余铜 6 种典型缺陷^[11]。

3.1 硬件部署

基于边缘计算的表面缺陷检测系统需要边缘端设备具有一定的深度学习计算能力,可以完成轻量化深度学习模型的实时性检测,同时能够与云端进行通信。基于需求,硬件选用为:NVIDIA Jetson Nano 开发板,CPU 为 128-core Maxwell, GPU 为四核 ARM A57 @ 1.43 GHz,操作系统为 Ubuntu18.04,环境为 Python3.6.13,部署 Tensorflow-gpu1.13.2 和 Keras2.1.5 深度学习框架,配备 CUDA10.0 并行计算架构和 cuDNN7.4.1.5 版深度学习加速库,同时扩展 SIM7600G-H 通信模块,从而实现边缘端设备与云端的数据通信。

云端设备相比边缘端设备有更高的算力,能够更快地完成深度学习模型算法的训练。本文选取的云端设备 CPU 为 Intel(R) Core(TM) i7-9750H 2.6 GHz 处理器,内存 16 GB, GPU 为 NVIDIA GeForce GTX 1660 Ti 6 GB GDDR6,显存为 6 GB。操作系统与边缘端设备同为 Ubuntu18.04,实验运行的环境为 Python3.6.13,深度学习的框架是 Tensorflow-gpu1.13.2 和 Keras2.1.5,配备 CUDA10.0 并行计算架构和 cuDNN7.4.1.5 版深度学习加速库。

3.2 模型训练

设置轻量化 YOLOv4 的训练批次大小为 16,

动量参数为 0.9,学习率为 0.001。在训练的过程中,首先冻结主干网络层进行训练,进行 50 轮次训练,对分类细节进行识别学习,提高检测精度;随后设置学习率为 0.000 1,解冻主干网络层再训练 100 轮次,对网络进行微调处理。

3.3 结果与分析

如表 1 所示,基于云边缘协同计算的表面缺陷检测系统数据响应时间仅为 32 ms,与本地计算方案响应时间基本持平,远低于云计算模式的响应时间。

表 1 数据传输响应时间

方案	响应时间/ms
云-边缘协同计算	32
本地计算	30
云计算	102

为评估表面缺陷算法的效果,本文选取轻量化 YOLOv4 算法、YOLOv4 算法、YOLOv3 算法和 Faster R-CNN 算法 4 种网络模型进行训练检测;同时,以模型的检测精度、检测速度以及模型大小作为评判标准。最终结果如表 2 所示,轻量化 YOLOv4 算法的 mAP 值分别为 96.41%,略低于改进前的 YOLOv4 模型,优于其他模型;轻量化 YOLOv4 推理速度达到 4.7 帧/s,远快于其他 3 种模型;同时,轻量化 YOLOv4 的模型大小仅为 53 MB,远小于其他模型。结果说明,轻量化 YOLOv4 有较好的检测精度,能够达到实时检测能够更好地在边缘端嵌入式设备上运行。

表 2 各模型检测结果比较

名称	mAP/%	检测速率/(帧·s ⁻¹)	模型大小/MB
轻量化 YOLOv4	96.41	4.7	53
YOLOv4	98.20	0.9	255
YOLOv3	92.77	0.8	239
Faster R-CNN	79.59	0.1	484

4 结束语

本文针对现有表面缺陷检测系统数据利用率低、检测实时性低和检测模型大等问题,提出了一种基于边缘计算的表面缺陷检测系统,在云端设备通过改进后的 YOLOv4 算法模型进行训练后,将其部署至边缘端的嵌入式设备,并搭建云端边缘端之间协同。实验表明,提出的云-边缘协同计算的表面缺陷检测系统具有检测实时性好、检测精度较高和

模型易部署等优势。后续会继续开发云端功能,能够更好地实现云端信息的数据交汇,分析出生产中所存在的问题。

参考文献:

[1] 陶显,侯伟,徐德.基于深度学习的表面缺陷检测方法综述[J].自动化学报,2021,47(5):1017-1034.

[2] WANG Y B,LIU M G,ZHENG P,et al.A smart surface inspection system using faster R-CNN in cloud-edge computing environment[J].Advanced engineering informatics,2020,43:101037.1-101037.9.

[3] 尹子会,孟荣,范晓丹,李冰,赵振兵.融合边缘计算和改进 Faster R-CNN 的变电站设备典型视觉缺陷检测系统[J].中国科技论文,2021,16(3):343-348.

[4] REDMON J,DIVVALA S,GIRSHICK R,et al.You only look once: unified,real-time object detection [C]//IEEE Conference on Computer Vision and Pattern Recognition,2016:779-788.

[5] REN S Q,HE K M,GIRSHICK R,et al.Faster R-CNN: towards real-time object detection with region proposal networks[J].IEEE Transactions on pattern analysis and machine intelligence,2017,39(6):1137-1149.

[6] WANG C Y,LIAO H Y M,WU Y H,et al.CSPNet: a new backbone that can enhance learning capability of CNN [C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). New York:IEEE,2020:1571-1580.

[7] HE K M,ZHANG X Y,REN S Q,et al.Spatial pyramid pooling in deep convolutional networks for visual recognition[J].IEEE Transactions on pattern analysis and machine intelligence,2015,37(9):1904-1916.

[8] LIU S,QI L,QIN H F,et al.Path aggregation network for instance segmentation [C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),2018:8759-8768.

[9] ZHENG Z H,WANG P,LIU W,et al.Distance-IoU Loss: faster and better learning for bounding box regression[C]// AAAI Conference on Artificial Intelligence,2020:12993-13000.

[10] SHI W S,CAO J,ZHANG Q,et al.Edge computing: vision and challenges[J].Internet of things journal, IEEE,2016,3(5):637-646.

[11] DING R W,DAI L H,LI G P,et al.TDD-Net: a tiny defect detection network for printed circuit boards [J].CAAI Transactions on intelligence technology,2019,4(2):110-116.