

云边端架构的工业产品质量检测系统设计

潘 桥, 郑明杰

(河海大学 信息科学与工程学院, 江苏 常州 213200)

摘要: 随着对产品质量检测效率与质量要求的提高与云原生相关技术的发展, 现有的本地单体应用模式难以实现整体上云或是利用云原生技术提供的资源, 因此提出了基于云边端架构的工业产品质量检测系统; 结合行业现状与工业互联网技术特点, 构建了由端侧实施层、边侧联动层、云侧平台层三层, 跨域互联系统、平行控域系统、综合测调系统三系统构成的云边端协同计算架构, 对相应层次与系统的组织结构、功能模块划分进行详细说明分析; 基于所提架构进行系统硬件与软件的设计与部署, 提出分布式异构系统资源调度模型; 最终通过运行实例与架构评估对提出的云边端系统架构设计与功能进行有效性验证, 所提系统满足了工业产品质量检测的应用。

关键词: 云边端; 工业互联网; 质量检测; 系统架构; 资源调度模型

Design of Industrial Product Quality Inspection System Based on Cloud-Edge-End Architecture

PAN Qiao, ZHENG Mingjie

(College of Information Science and Engineering, Hohai University, Changzhou 213200, China)

Abstract: With the increasing demands for efficiency and quality in product quality inspecting and the development of cloud-native technologies, it is difficult for an existing local monolithic application model to achieve overall cloud or utilize resources provided by cloud-native technology. Therefore, this paper proposes an industrial product quality inspection system based on a cloud-edge-end architecture. By considering the current state of the industry and the characteristics of industrial internet technologies, the system comprises three layers: the end deployment layer, the edge collaboration layer, and the cloud platform layer. Additionally, it incorporates three systems: the cross-domain interconnection system, the parallel control domain system, and the comprehensive measurement and adjustment system, with a detailed analysis and explanation of the organizational structure and functional module division for each respective layer and system. Based on the proposed architecture, the hardware and software design and deployment of the system were carried out. Besides, a distributed heterogeneous system resource scheduling model was proposed. Finally, the proposed cloud-edge-end system verifies the effectiveness of architecture and functions through operational examples and architecture analysis, which meets the application requirements for industrial product quality inspection.

Keywords: cloud-edge-end; industrial Internet; quality inspection; system architecture; resource scheduling model

0 引言

在工业产品制造的过程中, 由于技术限制, 缺陷难以避免, 这会导致其出现外观缺损、性质异常、功能指标无法达到预设要求等情况, 甚至完全损坏。因此, 在产品制造的后道工序中^[1], 质量检测是非常必要的, 通过它完成对缺陷产品的剔除和产品质量信息的反馈^[2], 及时发现各工序质量隐患, 避免大批量残次品的出现,

提高企业生产效益。

现如今, 部分制造商仍采用传统 PLC 上位机的模板匹配^[3]技术, 或是单独购置成品装置进行产品缺陷检测。在工业 4.0 时代, 随着对产品质量要求的不断提高以及产量的不断增加, 传统方案所带来的效率不高、结果欠佳、数据闭塞等不良影响, 已较大地阻碍相关产业发展。此外, 在企业内部存在着不同的应用系统, 而这些系统由于开发的时间不同, 采用的开发工具不同, 单

收稿日期:2024-07-19; 修回日期:2024-08-30。

基金项目:河海大学国家级大学生创新创业训练计划资助项目(202310294089Z)。

作者简介:潘 桥(2002-),男,大学本科。

引用格式:潘 桥,郑明杰. 云边端架构的工业产品质量检测系统设计[J]. 计算机测量与控制, 2025, 33(9):63-73.

一业务请求很难有效地调用所有的应用系统,出现了“信息孤岛^[4]”的情况,给系统整体的演化与迭代升级带来挑战,势必不利于信息的统一管理与大数据分析应用,进而影响企业生产效益。

随着云原生云计算相关技术的发展,现有的本地单体应用模式难以实现整体上云或是采用去中心化的分布式处理架构,导致系统难以利用云原生技术提供的资源。而针对部分本地对云依赖的业务,企业只能将这些业务单独挂载到云平台进行孤立处理,导致业务处理流程的严重割裂。为了解决上述问题,文献[5]提出了一种适用于智慧园区未来各类智慧化需求的云边端融合算网调度体系,研究以融合的云边端算网架构在现有的算力与网络基础上,用较小的升级改造成本提升园区智能化水平,该算网架构有效解决了园区云测、边缘侧、终端侧的存量算力资源分散问题并实现协同统一管理,但对于物联网端侧、边侧设备的系统接入方面有所不足。文献[6]提出了一种基于微服务的云边端协同的视觉感知 AI 中台,结合业务特点灵活协调好网络、算力资源的关系,实现端侧图片视频流的上传交付与云侧的识别处理,在单向数据流的应用场景有着较好的表现,但该系统缺少对于边缘设备的控制方法,支持的工作环境较为有限。文献[7]提出了基于云边端的感传算协同网络架构,促进多维系统、协作传输、异构资源、智能计算的深度融合,设计拓扑优化和路由策略继承算法,有效提高了网络传输效率和系统可靠性,但该套架构与算法在实践过程中可能缺乏适配度。上述研究对云边端协同计算架构^[8]的设计与优化作出了重要贡献,为本文提供了有益借鉴,然而通过进一步的检索,在工业互联网与传统制造业的融合发展方面,该架构的相关研究与应用仍然较少。

针对上述分析,本文设计了一种基于云边端架构的工业产品质量检测系统以解决上述现有技术中存在的缺陷与不足。通过打通产线全流程数据链路,实现产品全程、全量在线质检,从事后抽检转变为事前预测与事中全检,构建人机业务闭环的智能制造系统,实现数据的高效处理和分析,提高生产过程的效率和质量。

本系统的主要创新点在于:

1) 改进的“云—边—端”架构。根据现代工业环境实际情况,设计了一套完整的云边端协同计算架构,其具有纵向协同、横向扩展的特点。层与层之间、模块与模块之间、服务与服务之间松散解耦合,可以根据实际情况灵活地增添、替换、修改其中若干个组成部分,且各部分广泛兼容现有协议,便于构件与模块的复用,具有较强的适应性、可修改性、可维护性、可移植性。

2) 完备可行的系统层次与模块划分方案。系统具有物理连接与逻辑连接分离、相对独立物理设备与高度

融合软硬件系统共同组合成层的特性,并通过合理地划分与设置各个层次中模块、功能和数据流的分布位置、组织关系、逻辑顺序及交互作用,以实现系统运转的性能、效率的最大化,对资源的统一管理^[9]和动态调度。

3) 对应的软硬件实物系统。设计并部署了基于云边端架构的工业产品质量检测云上智能体,具有架构设计中完整的各组成部分。其理论研究与研发应用成果着眼于“云—边—端”架构系统设计部署,可推广应用至其它工业产品进行各类检测。

1 云边端协同计算架构

如图 1 所示,本文提出的系统架构基于传统“云—边—端”模型。当前对于云边端模型在系统中的应用已有一定的研究,文献[10]提出了基于“云边端”的工业互联网通用系统架构,将云平台端、设备端、边缘段两两相连。文献[11]提出了由端层、边缘层、管道层、云层构成的云管边端技术架构,通过边缘侧与中心云的协同实现系统的高效运作。文献[12]提出了数据经由云侧、端侧、边缘侧,最终回到云侧的单向流转“云边端”架构。文献[13]提出了一种由应用层、软件定义层、边缘层、节点层四层组成的车联网架构。

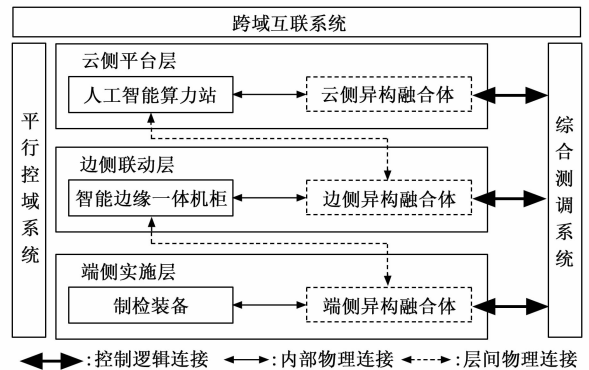


图 1 系统架构设计图

相较于现有的各类“云—边—端”架构,本文所提出的系统架构采用“2 元+3 层+3 系统”的组织模式,不同实体之间的连接分为物理连接与逻辑连接。具体细节为:

1) “2 元”: 层内由相对独立的物理设备与高度融合的软硬件系统共同组合而成。各层内部的模块连接为内部物理连接。

2) “3 层”: 核心软硬件系统自底向上由端侧实施层、边侧联动层、云侧平台层组成。各层之间通过综合测调系统建立控制逻辑连接,而对于各层相对独立的物理设备,则通过下层异构融合体建立物理连接,来间接地实现先前提到的逻辑链接。对于实现核心功能的各层软硬件系统而言,这些物理设备与物理连接是透明的,它们不需要对它们进行直接控制,而是通过驱动与

应用程序接口与相邻层异构融合体进行交互。

3) “3 系统”: 通过跨域互联系统以实现系统的横向扩展, 通过平行控域系统与综合测调系统以实现系统的纵向协同。

1.1 端侧实施层

如图 2 所示, 端侧实施层是物理域与数字域的数据出入口, 由质检装备和端侧异构融合体组成, 主要实现物理设备的接入与控制。

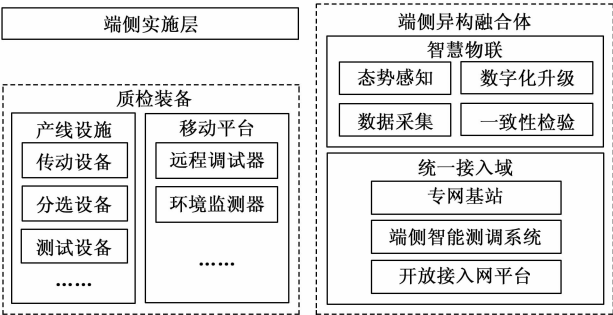


图 2 端侧实施层组织结构图

质检装备由产线设施与移动平台构成。产线设施为大型固定装备, 包含传动设备、分选设备、测试设备; 移动平台为小型可动装置, 包含远程调试器, 环境监测器。

端侧异构融合体由智慧物联与统一接入域构成。智慧物联管理制检装备的运转方式。态势感知通过传感器收集产线环境信息。数字化升级和数据采集通过在产线设施上安装额外的信息采集模块, 与运动平台相互合作, 获取质检所需的各类信息。一致性检验则是在端侧保证采集数据的合法性, 进行初步处理后通过统一接入域上传。统一接入域面向万亿级物端设备千差万别的接入需求, 基于通信、感知、计算深度融合, 提供全域覆盖、无缝智能、质量可控的边缘网络部署接入。

1.2 边侧联动层

如图 3 所示, 边侧联动层是自主可控的网络平台, 支撑系统广域互联与传输, 由智能边缘一体机柜和边侧异构融合体组成。

智能边缘一体机柜由一系列边缘设备与最小系统拓扑连接模块构成, 通过对松散分布的边缘设备进行局部统一接入管理, 支持根据部署的设备类型自动选择驱动程序, 并有序分配线路资源, 向上提供一致的业务出口。边侧异构融合体由统一部署域与智能核心网组成。统一部署域通过与端侧统一接入域、云侧对接, 提供不同类型数据的交换服务。

1.3 云侧平台层

如图 4 所示, 云侧平台层是系统进行计算和存储的数据处理中枢, 为各类云平台应用提供数据洪流下云端算力保障和服务能力, 由人工智能算力站和云侧异构融合

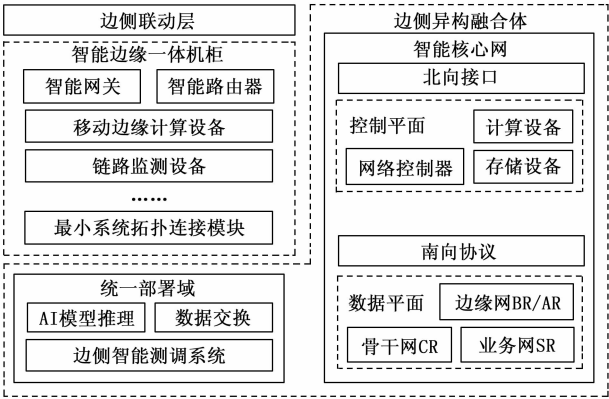


图 3 边侧联动层组织结构图

合体组成。

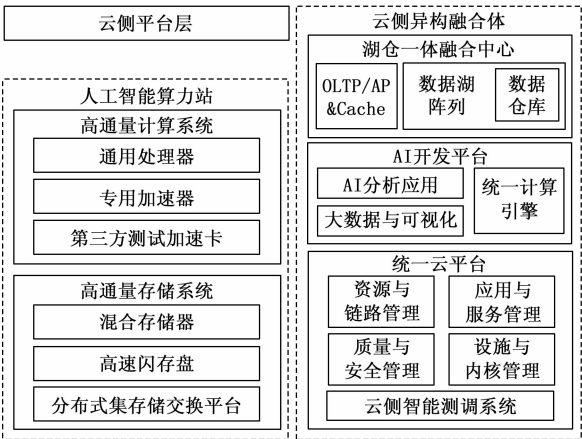


图 4 云侧平台层组织结构图

人工智能算力站由高通量计算系统与高通量存储系统构成。云侧异构融合体由湖仓一体融合中心、AI 开发平台与统一云平台构成。

1.4 综合测调系统

为应对端侧与边侧的连接洪流、边侧智能核心网的数据洪流、边侧与云侧的服务处理洪流, 综合测调系统以三层子系统为接入点, 在对全链路核心部件的状态、性能、安全性、服务质量实时测量的基础上, 对全系统所有可控部分进行算法、算力、算据的全局调度与分配, 满足系统全工作负载情况下的实时保障需求, 是实现高 QoS 的核心保障。

1.5 平行控域系统

作为“云一边一端”架构系统逻辑层分布式隔离与共享机制, 平行控域系统通过将云边端资源进行虚拟切片为各类应用提供分布式可信执行环境, 保障系统中各单元安全共享算力、网络、存储资源。

1.6 跨域互联系统

基于联邦云技术平台, 跨域互联系统通过整合不同区域算力站资源向终端用户提供统一的云服务, 实现云

间互操作与算力横向扩展。其兼具公有云的广泛资源供给与弹性扩展能力,以及私有云的高品质服务与高安全特性。

2 系统硬件设计

本文提出的基于云边端架构的工业产品质量检测系统以集成电路芯片为检测对象,利用相机进行表面图像采集与缺陷检测。通过对芯片动力学特性进行研究来精细调节运动平台硬件的各项参数,以便推广至其它各类工业产品进行检测。总体上系统硬件部分由端侧制检装备、边侧智能边缘一体机柜、云侧人工智能算力站组成,分别对应系统架构的三层独立物理设备。

2.1 端侧制检装备

以三轴运动检测平台为产线设施中的传动、分选、测试设备,以智能声感单元、环境态势感知器为移动平台中的远程调试器、环境监测器。

如图 5 所示,三轴运动检测平台利用 4 个 42 步进电机驱动,其中 X 轴与 Y 轴采用电机驱动同步带的运动方式,以加快运动速度与快速启停等精细控制操作,在保证最大运行速度 200 mm/s 的同时,将运动精度控制在 0.01 mm 以内。Z 轴则采用电机驱动丝杆的方式,并使用 X 轴相同的多轮滑块,通过其与 Z 轴型材的耦合保证纵向支撑与运动的稳定性。三轴电机旁均使用物理限位检测器的设计。此外还加入了 E 轴电机,为凸轮驱动下压机构留有扩展空间。Y 轴采用移动载物台设计,并可改装接入传送带,以拟实际产线的工作环境。在机器底部装有 ESP32-S 型 SoC 及配套主板,主板在保证基本功能的前提下,实现了高可扩展性、可维修性。功能上,主板为电机、外置模块提供稳定的电

源,并接有载物台三轴加速度传感器、温度传感器,在端侧即实现环境参数的采集与处理。控制方式上,正常运行状态通过 ESP32-S 接入专网基站,由端侧智能测调系统实现控制,远程控制失效时则可通过本机或通过线缆连接至上位机进行调试。

如图 6 所示,环境态势感知器以 STM32F103C8T6 最小系统板为核心,将各个子模块部署至指定位置并连接至上位机。HX711 是一款专为高精度电子秤设计的 24 位 AD 转换器,将其部署在载物平台下方以实现芯片重量检测。将 BH1750 光线传感器部署在载物平台旁,以实现环境光线强度监测。通过 DHT-11 传感器实现温湿度检测,通过基于 LM393 的火焰传感器实现火光检测,通过 ESP8266 WiFi 无线传感模块实现与专网基站的连接,通过 MQ-2 传感器实现烟雾检测,通过 RC522 RFID 射频识别器、AS608 光学指纹识别模块实现用户身份识别。环境态势感知器根据采集到的相关信息进行响应:

1) 需要通过 NFC 与光电指纹进行身份识别后,才可进行本地配置的修改与保存。

2) 部署在运动平台上的环形 LED 光源将根据光线传感器检测到的光线强度自动调整补充照明亮度,0 lx 时 LED 设为 100% 亮度,500 lx 时设为 0% 亮度,0~500 lx 范围内则按照预设函数进行无级调节,最终使检测到的光线强度控制在 500 ± 50 lx。

3) 相机图像采集框内均识别完成后,将通知三轴运动检测平台以控制电机运动。

4) 压力传感器实时检测每批次产品重量并上传,系统根据重量变化情况与采集到的图像判断待检物

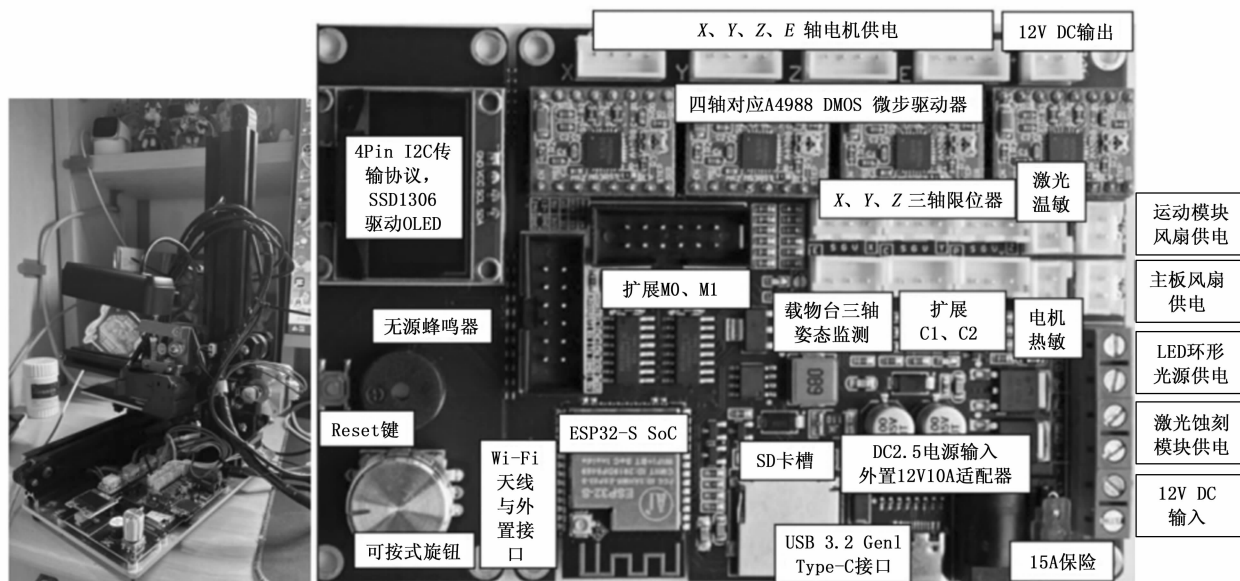


图 5 三轴运动检测平台实物图

类别。

5) 温湿度传感器、烟雾传感器、火焰传感器实时采集监测相关参数并上传。



图6 智能声感单元、环境态势感知器实物图

智能声感单元以 LU-ASR01 控制器为核心,代替大部分物理按键,供操作人员无接触调用端侧质检装备功能,并实时监听环境音,以监控装置运行状态。三轴运动检测平台将电机运动情况与待执行操作定期上传至智能声感模块,例如在已知 Z 轴电机运动的情况下,智能声感单元检测到啸叫声,其会记录相关音频与所执行的操作,写入日志、通知用户并上传至统一云平台。

2.2 边侧智能边缘一体机柜

以 ARMv8 架构 Snapdragon SM8550 SoC 的移动设备为智能网关,运行 Linux 系统,支持虚拟化且可编程。以 TL-XDR5450 与 TL-WDR7660 组网为智能路由器,构成智能网络的物理基础,承担着端侧实施层、边侧联动层、云侧平台层三层之间的远程网络连接。智能网关与智能路由器对不同来源与目的地的信息进行分组转发与路由选择。

以 x86_64 架构 Intel Core i5-1245U CPU 的工作站为移动边缘计算设备,包含 16 GB LPDDR5 5 200 MT/s 的物理内存与 256 GB SSD,以 $<10\text{ W}$ 的整机功耗实现较好的性能表现,用于协助处理低算力、高时效要求的计算任务,例如对芯片表面封装图像进行分析、处理、训练产生 YOLOv7^[14]模型后,边缘计算设备在实际检测过程中通过模型推理,即时生成检测结果,并将原始数据与检测结果汇聚融合后提交以进行进一步的模型训练^[15]。此外在工作站中设置虚拟链路监测设备,对链路进行实时监控,以便于对工作负载与流量进行限制。上述设备通过拓扑模块进行供电与有线通信连接,接入边侧智能测调系统。

2.3 云侧人工智能算力站

选用不同型号 CPU、GPU、不同容量物理内存、多容量 SSD 与 HDD 混合的多台工作站部署高通量计算与存储系统集群,以验证系统可行性、实际应用与推广价值,充分利用本文所提“云一边一端”架构系统相较于传统工程控制系统在数据处理、异构兼容、可扩展性

方面的优势。

工作站 1 配置信息:

- 1) 架构平台: x86_64 Windows 11;
- 2) CPU: Intel Core i7-10870H;
- 3) GPU: Nvidia GeForce RTX 2060 6 GB;
- 4) 内存: 32 GB;
- 5) 外存: 3.5 TB SSD (512 GB+1 TB+2 TB)。

工作站 2 配置信息:

- 1) 架构平台: x86_64 Windows 11;
- 2) CPU: AMD Ryzen 9 7950X3D;
- 3) GPU: Nvidia GeForce RTX 4080S 16 GB;
- 4) 内存: 64 GB;
- 5) 外存: 4 TB HDD, 2 TB SSD (1 TB×2 RAID 0)。

通过提供高并发数据处理能力和强实时的信息数据通路^[16],以满足新一代信息基础设施的建设需求。整体层次化设计,核心软件和硬件设备模块化,便于灵活替换与测试集成。系统支持多加速卡的流式协同加速,充分利用 PCIe 带宽和多卡并行计算能力。除支持视频处理加速卡、深度学习加速卡以外,还接入第三方测试设备以完成专业特化功能。

3 系统软件设计

3.1 智能核心网与测调系统

智能核心网为大规模的虚拟专网,是云边端互联的平台基础。相较于传统核心交换网络,智能核心网的角色从传输通道演变为多维资源协同平台,功能从单一静态演变为多元可定制,传输能力从尽力而为演变为应用感知的确定性智能传输。其技术架构如图 7 所示,总体上基于 SDN 架构构建网络,通过应用 APN 识别技术实现应用级运营,借助 AI 技术实现流量智能调度。基于网络切片技术保障 QoS 能力,通过新一代广域承载解决方案 SRv6 技术实现大规模动态组网与调度,借助 IPv6 特性实现网络内生安全,采用随流检测技术实现服务级别可视化与故障定位。

测调系统以一台边缘计算设备安装 Ubuntu 20.04,通过 Floodlight、Mininet、Open vSwitch 搭建部署智能核心网,实现计算通信多维资源测调,支持云边协同的边缘计算及智能决策。通过调用边侧智能边缘一体机柜的移动边缘计算设备,满足低时延业务需求;对于部分既有业务直接本地处理,缓解核心网数据传输压力,提高网络智能化水平。

3.2 湖仓一体融合中心与数据可视化

数据仓库穿插于数据湖阵列之间,“数据湖+数据仓库”的湖仓一体混合架构满足了结构化、半结构化、非结构化数据高效处理需求,解决了传统数据仓库在海

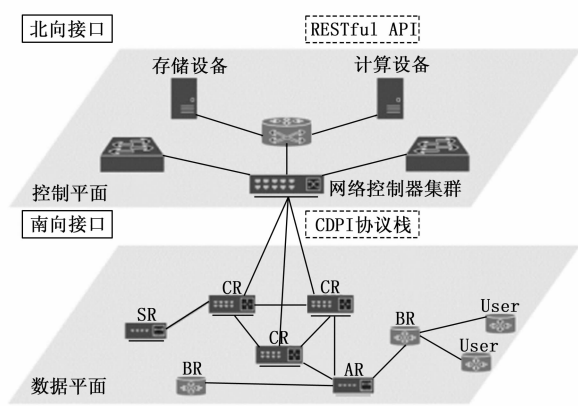


图 7 智能核心网组织架构图

量数据下加载慢、数据查询效率低、难以融合多种异构数据源进行分析的问题。将 OLTP、OLAP、Redis 联合起来，OLTP 用于处理企业级的常规业务操作，强调数据的精确、事务的原子性和并发性；OLAP 用于处理企业级的决策分析、战略分析以及业务分析等方面，采用多维数据分析技术和聚合算法，将大量数据划分成各种不同的角度，方便分析数据；Redis 是一种开源的高性能内存数据库，具备快速的读写能力和丰富的数据结构支持，其数据存储在内存中，采用高效的数据结构和算法，使得读写操作的速度非常快，适用于高并发场景。

数据可视化基于标准化组态设计，搭建工业大数据分析基础数据资源池，借助云侧平台层的湖仓一体融合中心模块提供的全时间全流程信息，适时对边侧统一部署域的产线数据进行大数据分析，在数字化能力基础上，面向不同角色与主题进一步开发有针对性的数据分析应用，形成边侧部署的数据产品，充分挖掘产线数据的价值，实现“智慧工厂/智慧车间”的高级智能化功能。此外，按“对象分解结构/分析主题”进行分类汇总，搭建产线运行状态可视化中央监控大屏。如图 8 所示，前端通过 HTML5+CSS+JavaScript 框架组合实现，交互逻辑采用 jQuery 框架，数据呈现采用 Echarts 组件包，后端采用 NumPy 进行数据处理。根据用户个性化需求进行灵活、实时、联动展示，涵盖事件分析、综合管理、晶圆制造、封装检测、流量监控等功能，最终实现“透明工厂/透明车间”的基础数字化功能。

3.3 综合控制面板

控制面板以开源的 PySide6 开发工具、QFluentWidgets 组件包、Javascript 设计前端程序，通过 OpenCV、PyTorch 等多项开源组件包设计基于自主改进的 YOLOv7 深度学习检测算法，利用 Nginx、PHP-fpm、MySQL 搭建管理数据库服务器。

前端软件均采用模块化设计，如图 9 所示，统一云



图 8 数据可视化应用界面

平台依托控制面板 Dashboard，实现资源与链路管理、应用与服务管理、质量与安全管理、设施与内核管理的相应功能；如图 10 所示，AI 开发平台依托检测应用 Detection，实现深度学习图像检测的相关功能。后端软件则面向专用硬件设计高效、稳定、开放、可扩展的功能服务，充分驱动硬件的基础上，实现指定的功能，并提供可调用接口以供上层应用控制与读取信息。

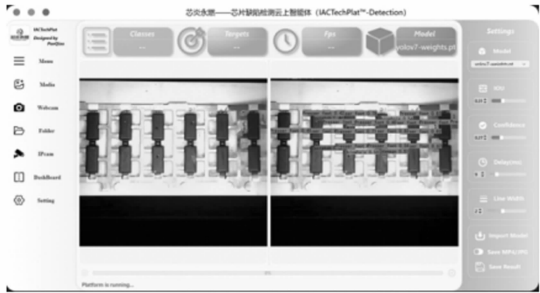


图 9 AI 开发平台应用界面



图 10 统一云平台应用界面

总体上重点考虑终端能力的上移和中心能力的下沉。基于 Kubernetes 与 Docker 设计容器化的微服务构件，利用 Serverless 技术，通过 FaaS 与 KVM 部署多项应用服务。

1) 中央大屏通过数据可视化展现，集中显示系统的几项关键核心信息，包括负载状况、算力利用情

况等。

2) 实施层管理显示厂区平面分布图, 可根据要素切换不同视图, 包括温度、湿度、光照、响度等, 并展示相应数据的分析汇总信息。

3) 联动层管理可对系统中在线的所有智能边缘一体机柜进行管理 & 配置维护, 数据库与服务器的配置维护

4) 云平台管理通过云桌面应用访问统一云平台的各项服务, 通过资源管理器访问湖仓一体融合中心。

5) 远程协作通过加密指令传输, 以进行便捷的合作办公。

6) AI 建模分析中, 通过对湖仓一体融合中心的多维数据进行挖掘、上卷下钻、查询分析, 对信息进行可视化展示。

7) 链路 & 调度直观呈现智能核心网与边缘网络各段的链路状态, 并可在调度栏中进行命令操作。

8) 资源监控对电源、环境、冷却水等关键基础资源进行监控管理。

9) 安全子界面可设置各类生物识别安全密钥。在更多和下载界面, 可对安装的模块进行添加删除。在通知与设置子界面, 可对系统进行各类个性化操作与设置。

4 分布式异构系统资源调度模型

在传统的单体调度模型中, 所有节点的资源及用户任务均由中央服务器统一管理和调度。然而, 这种集中式的架构模式不可避免地使中央服务器成为系统的单点瓶颈, 进而直接限制了调度的规模和服务类型的支持能力。单体调度架构之所以难以支持多样化的任务类型, 源于不同服务对调度框架和计算资源的需求存在显著差异。例如, 系统最初可能仅支持批处理任务, 这类任务处理静态数据; 随着系统的发展, 实时数据处理的流式任务逐渐被引入, 流式任务处理的是实时数据。随着任务类型的增加, 单体调度架构的复杂性也随之上升, 最终导致扩展性问题, 进而制约了系统的整体性能。

为了解决上述问题, 提高调度效率并支持多样化任务类型, 适配云边端架构特性, 提出了一种分布式异构系统资源调度模型。如图 11 所示, 这种模型通过将资源管理与任务调度分离, 使得不同层次的调度器各司其职, 从而更有效地支持复杂的任务需求。I 级调度器根据数据流向与质量属性要求决定资源请求的处理层; II 级调度器部署在人工智能算力站分布式集群系统中, 负责高通量计算存储资源的统筹与分配; III 级调度器则由各个应用程序框架完成, 其任务是根据 II 级调度器分配的资源, 结合任务需求, 执行资源与任务的匹配。

该调度模型基本原理如下所述:

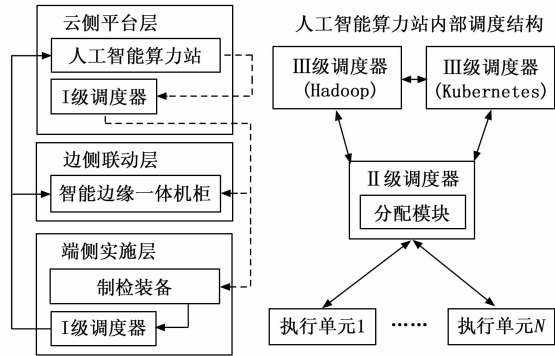


图 11 分布式异构系统资源调度模型架构图

1) I 级调度器根据资源请求的自然流向进行初次分配。端侧设备的资源请求由 I 级调度器决定由边侧或云侧处理, 而当该请求仅能由某一层处理时, I 级调度器直接交由指定层处理。云侧设备的请求由 I 级调度器决定由云侧或边侧处理, 而当该请求目标非云侧时, I 级调度器直接交由边侧处理。

2) 此后对于目标为边侧与端侧的资源请求, 采用抢占式优先级调度算法。

3) 对于目标为云侧的资源请求, 框架 (Hadoop 与 Kubernetes) 向 II 级调度器注册, 执行单元节点定期向 II 级调度器上报本节点空闲资源。

4) II 级调度器收集节点的空闲资源信息, 并以资源发放的方式将空闲资源发送给注册的框架;

5) 框架对应 III 级调度器接收到 II 级调度器的资源发放后, 根据框架提供的调度算法进行任务调度与匹配。匹配成功后, 将匹配结果下发 II 级调度器, 并由分配模块转发给相应执行单元节点。

I 级调度器仅决定资源请求的处理层, II 级调度器提供空闲资源情况并执行资源分配, III 级调度器则决定资源接收量与各任务的资源匹配, 从而实现了任务调度与资源管理的分离。

4.1 服务质量自学习算法

I 级调度器采用服务质量自学习算法进行资源请求的分配, 以满足不同场景个性化业务需求, 在时延、速率、可靠性^[17]三项技术指标间达成平衡。设 U 表示一种特定服务场景的资源需求度, 其满足下式的函数关系:

$$U_n = S(\tau_n, R_n, \alpha_n) \tag{1}$$

式中, τ_n 表示目标延时, 单位为 ms; R_n 表示目标传输速率, 单位为 Mbps; α_n 表示目标可靠性加权量化值, 公式为:

$$\alpha_n = K_n L^{(1-x)^{-1}} x, x \in (0, 1] \tag{2}$$

式中, K_n 为定向加权因子, 在运行中不断根据服务质量进行反馈调节; L 为固定加权因子, 由指定场景决定数

值; x 为目标可靠性, 为百分比形式。

在实际运行中, 由于某一场景的目标时延、速率、可靠性衡量值均为区间形式, 故最终资源需求度 U_n 也为区间形式。I 级调度器将计算得到的预期资源需求度 U_n 与不同调度方式下的理想 $U_m, \tau_m, R_m, \alpha_m$ 区间进行重合度比较, 通过硬判决决定最佳匹配调度。在实际执行过程中, 某一场景下的预期资源需求度及其自变量会根据执行后的反馈情况进行范围调整, 以不断接近理想调度方式。

下文以实例进一步说明。如图 12 所示, 初始状态下根据 x, y, z 三轴划分的 5、4、5 段形成 100 个资源需求区块, 以坐标方式唯一确定。以 3 种典型场景为例, 对于端侧发出的即时计算操作指令, 其需要 <1 Mbps 速率、 <1 ms 时延、100% 可靠性的目标资源, 初始区块坐标为 (1, 5, 5)。由于资源请求目标未指定, I 级调度器执行服务质量自学习算法, 首先通过判决划分给覆盖范围为 <1 Mbps 速率、 <10 ms 时延、 $>99\%$ 可靠性的执行单位, 单次执行后评价反馈为优, 故在一定时间范围内, 该场景业务均由该执行单位执行。累计执行多次后评价反馈仍然为优, 则该场景资源需求根据执行单位的数值放宽至 <1 Mbps 速率、 <2 ms 时延、 $>99.5\%$ 可靠性, 并再次执行服务质量自学习算法, 依据区间重合度比较判决划分执行单位。数次执行后 3 种典型场景下资源需求情况如图 13 所示, 资源敏感型业务所呈现的区块大小会小于初始分配区块, 其对目标资源的预期值会更加准确; 而资源不敏感业务的区块大小会大于初始分配区块, 其对目标资源的预期值较为宽泛, 需对该业务场景进一步细分以实现更合理的调度。

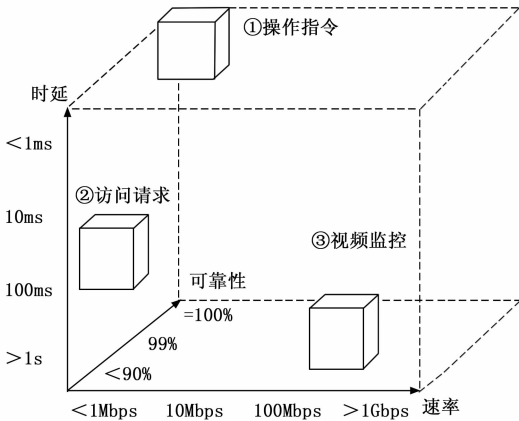


图 12 3 种典型场景下初次资源需求区块划分图

在服务质量自学习算法的实际执行过程中, 由于 I 级调度器需要以极小时延执行判定, 同时避免资源需求数值抖动, 覆盖范围符合式 (3) 的执行结果将被丢弃且不会触发自适应调整, 以减小不必要的计算量。

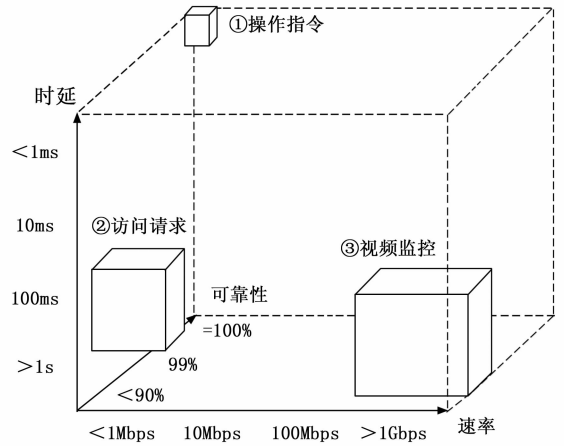


图 13 数次执行后资源需求情况图

$$U_n(t_{a+1}) \in \begin{cases} [0.99U_n(t_a), 1.01U_n(t_a)] \\ [0, 0.95U_n(t_a)] \\ [1.05U_n(t_a), +\infty] \end{cases} \quad (3)$$

具体执行过程中式 (3) 的权值将根据业务类型、资源分布情况等因素进行调整。

4.2 主导资源指向性算法

II 级调度器采用一种针对多资源环境的分配策略——主导资源指向性算法。通过计算每个任务在不同种类资源上的占比, 来确定其资源分配。对于每个任务, 占比最大的资源被称为主导资源, 对应的占比称为主导份额。不同用户的主导资源各异, 例如, 对于计算密集型的任务, CPU 为其主导资源; 而对于 I/O 密集型任务, 带宽为其主导资源。主导资源指向性算法通过在任务的主导资源上应用最大—最小公平原则, 来保证资源分配的公平性, 寻求优先最大化系统中最小的主导份额, 随后依次优化次小的主导份额, 以此类推。

定义如下变量:

- 1) $\mathbf{R} = [r_1, r_2, \dots, r_n]$: 表示系统中所有资源的总容量, 包含 n 种资源。
- 2) $\mathbf{C} = [c_1, c_2, \dots, c_n]$: 表示已被使用的资源, 初始值为 0, 即所有资源都是未使用的。
- 3) $s_i, i = 1, 2, \dots, m$: 表示用户 i 的主导份额, 初始值为 0。
- 4) $\mathbf{W}_i = [w_{i,1}, w_{i,2}, \dots, w_{i,n}]$: 表示每种资源的权重, 用于调整主导份额的计算, 使得不同资源对主导份额的贡献可以根据其权重进行缩放。
- 5) $\mathbf{U}_i = [u_{i,1}, u_{i,2}, \dots, u_{i,n}]$: 表示分配给用户 i 的资源向量, 初始值为 0。
- 6) 在每一轮资源分配中, 首先选择主导份额 s_i 最小的用户 i , $\mathbf{D}_i$ 表示用户 i 下一个任务的资源需求向量。

判断是否满足资源约束条件 $\mathbf{C} + \mathbf{D}_i \leq \mathbf{R}$:

如果满足, 即所需资源加上当前已分配的资源没有

超过系统资源总量, 则更新已使用的资源向量 C :

$$C = C + D_i$$

(4)

表示这些资源已经被使用。更新用户 i 的分配向量 U_i :

$$U_i = U_i + D_i$$

(5)

表示这些资源被分配给了用户 i 。重新计算用户 i 的主导份额 s_i :

$$s_i = \max_{j=1}^n (u_{i,j} / w_{i,j})$$

(6)

新的主导份额为用户 i 的资源占比中的最大值。
如果不满足资源约束条件, 即集群中的资源已不足以满足该用户的需求, 则算法终止。上述算法使用存储每个用户的主导份额的二叉堆来实现, 对于 n 个用户, 每个调度决策的时间复杂度为 $O(\log n)$ 。

5 系统运行实例与分析

为进一步验证本文提出的“云一边一端”架构系统及其配套软硬件的可行性, 按照前文提出的软硬件设计与资源调度模型进行实物部署, 并对系统架构进行评估。

5.1 系统运行实例

1) 端侧实施层控制质检装备工作方式。待检产品为集成电路芯片, 传动设备驱动级别设定为适合运送芯片的参数, 测试设备设定为检测芯片的表面封装情况, 分选设备按照检测出的不同种类, 包括划痕、凹陷、缺损、引脚缺损, 要求远程调试器与环境监测器按照各自设定的任务自动完成相应操作, 无需人工干预。

2) 端侧实施层通过智慧物联采集质检装备产的数据, 通过 I 级调度器将采集的数据发送给边侧联动层或云侧平台层。每批芯片经过测试设备都需要存储表面封装图像, 记录其批次、种类、重量信息, 环境检测器实时记录环境参数, I 级调度器判决数据交由智能边缘一体机柜处理。

3) 边侧联动层对采集的数据进行汇总、处理、分析。智能边缘一体机柜针对芯片表面封装图像利用统一部署域 YOLOv7 模型进行推理操作, 检测其表面存在的缺陷种类、个数、置信度等信息, 整理出各缺陷的图像, 进一步优化模型中缺陷程度的划分。对 1 000 张缺陷图像测试集进行检测, 识别正确率 96.5%, 在 1 280 × 720 分辨率下实现平均 40 FPS 的实时检测速度。

4) 边侧联动层将处理后的数据通过智能核心网发送给云侧平台层湖仓一体融合中心, 并将处理后的数据交由 AI 开发平台进行建模分析。借助人工智能算力站与 II、III 级调度器, 湖仓一体融合中心对接收的一切数据进行存储、挖掘, AI 分析应用针对湖仓一体融合中心得出的分析结果, 总结制造过程中存在的问题与薄弱环节, 并对质量检测过程进行准确率分析与评价, 进而

作出决策支持; 根据检测结果中集中出现同类缺陷的情况, 推断出某些批次产品可能存在通病, 进而对相应批次产品生产时的制造流程进行回溯追踪, 及时发现各工序质量隐患。

5) 云侧平台层将处理后的数据传给大数据与可视化模块以进一步对信息作针对性处理与展示。将某几个批次芯片的产量、良品率、缺陷分布情况等数据进行模型化可视化展示; 对时间上广泛分布的多批次数据进行比较与拟合, 以得到相关数据的历史趋势信息。

6) 用户通过向统一云平台管理传递远程控制信息, 以直接或间接的方式管理端侧实施层、边侧联动层、云侧平台层的各项配置, 并获取所需的数据与信息。用户通过 Https 和 VPN 连接该平台, 要求修改端侧实施层中测试设备对芯片进行缺陷检测的参数阈值, 修改边侧联动层中边缘计算设备的最高承载量, 修改云侧平台层中链路连接的安全策略等。用户通过该平台查看当前在线的终端设备数量、工作状态等信息, 查询日志了解端侧实施层中设备发生故障的情况。

7) 云侧平台层结合自身决策分析信息、远程控制信息、超级管理员设置信息对云侧平台层、边侧联动层、端侧实施层进行配置参数的更改, 以此构成反馈流程。云侧平台层在决策分析信息中对云侧应用框架编排、边侧智能核心网组织结构、端侧质检装备参数配置等信息进行调整。

5.2 分布式异构系统资源调度模型分析

通过测试仿真环境下的任务调度与执行效率情况, 截取了实际运行中 5 个不同时间段内链路网络中连续出现的 1 000 条信息, 在平台中测试本文所提任务调度模型、主导资源指向性算法、服务质量自学习算法与几种常见算法组合的性能并进行比较, 具体结果如表 1 所示。

表 1 几种算法运行结果比较

I 级调度算法	II 级调度算法	服务质量 达标比/%	调度性能 占用/%
服务质量自学习算法	主导资源指向性算法	97.7	0.5
服务质量自学习算法	主导资源公平算法	96.2	0.47
无	最大最小公平算法	90.1	0.39
无	基于槽的公平调度	87.1	0.37

在拟真场景下, 本文所提的分布式异构系统资源调度模型能够在调度性能占用小幅增加的情况下, 有效提高服务质量达标比, 从而可以提供更加高效的任务调度, 满足了实际环境中云边端资源分配的需要。

5.3 系统架构评估分析

系统架构的基本需求是在满足功能属性的前提下, 关注软件系统质量属性。软件系统质量属性 (Quality Attribute) 是一个系统的可测量或者可测试的属性, 用

表 2 运行期质量属性场景描述与评估

场景元素	质量属性及相应场景					
	可用性	性能	可靠性	安全性	可伸缩性	互操作性
刺激源	软件系统	1)系统触发 2)用户请求访问	软件系统	1)正确识别身份 2)未经授权的访问	用户请求访问	1)软件系统对外接口 2)对内硬件接口
刺激	1)用户无法登录 2)端侧实施层设备离线 3)计算存储设备故障	1)定期/偶然事件到达 2)突发/随机事件到达	1)应用运行错误 2)系统运行错误	试图查看/改变/删除数据,访问系统服务	1)请求数量增加,达到当前态最高负载 2)超载后请求数量回落至正常负载以下	1)向跨域互联系统下其它孤立系统请求数据 2)控制请求与数据访问
响应	1)自行检测并尝试隔离错误,保存日志通知维管人员 2)屏蔽故障源,指定时间内不可访问 3)提供有限容量和功能	1)调用一般处理设备承接任务,平衡性能与功率消耗 2)优先调用高性能处理设备承接任务	1)自行尝试重启应用服务 2)自动启用备用平行系统,尝试恢复原系统功能	1)根据身份记录权限访问/修改数据或服务 2)以不可读的格式存储数据,审核跟踪非正常请求并限制可用性	1)通过水平扩展方式,自动启用休眠状态人工智能算力站资源 2)根据调度算法指数级降低启用机数量	1)本系统接受其它孤立系统通过跨域互联接口返回的相应数据,拒绝或超时则发出通知 2)请求接收,交互成功
响应度量	1)故障自检通知时限<10 分钟 2)故障不影响正常运行设备 3)可用性>90%	1)允许最大 15%性能损失以最小化能耗,请求响应时延<100 ms 2)允许最大 50%基础频率提升以提高瞬时响应	1)故障应用重启后正常运行概率>90% 2)MTBF>1 000 h, 故障系统重启后正常运行概率>99%	1)正确提供对应身份服务权限概率>90% 2)24 h 内恢复校正数据,确定访问/修改数据的来源可能性>80%	1)最高可增加常规负载下 500%的计算存储资源 2)回落过程能够及时有效应对突发请求	1)未拒绝或超时情况下信息正确交换率 99% 2)信息正确交换率 95%

来描述系统满足利益相关者（Stakeholders）需求的程度^[18]。为了精确、定量地表达系统的质量属性，我们灵活运用质量属性场景描述，以对系统软件架构进行评估。质量属性场景是一种面向特定质量属性的需求。它由六部分组成：刺激源、刺激、环境、制品、响应、响应度量。

如表 2 所示为主要面向用户的运行期质量属性场景描述与评估，其主要包含可用性、性能、可靠性、安全性、可伸缩性、互操作性六项。可用性是系统能够正常运行的时间比例；性能是系统的响应能力；可靠性是软件系统在意外或错误使用的情况下维持功能特性的基本能力；安全性是指系统在向合法用户提供服务的同时能够阻止非授权用户使用的企图或拒绝服务的能力^[19]；互操作性是不同系统、设备或应用之间能够有效地交换信息并进行操作的能力。经测试分析，本文所提系统在一些运行期的常见场景中能较好地符合上述需求。

6 结束语

本文针对传统本地单体应用模式产品缺陷检测方案存在的效率不高、结果欠佳、数据闭塞等问题入手进行需求和现状分析，针对工业互联网和产业特点提出了基

于云边端架构的工业产品质量检测系统，详细转述了系统架构设计、硬件与软件设计方案并对核心部分加以实现，最后通过质量属性场景描述对多项关键质量属性进行验证分析。该系统已在针对集成电路芯片封装缺陷检测中得到应用，可显著提高芯片缺陷检测效率与质量控制能力^[20]，通过高度模块化设计用户友好型软件界面与数据可视化呈现，有效增加了其易用性与实用性。未来的研究工作中，还将持续完善该套系统，如提供面向业务数据托管的专有云数据镜像服务，在边侧搭建工业大数据分析基础数据资源池等，进一步提高检测过程的效率和质量。同时，也将积极在其它行业推广相关研究成果，推进各行业数字化转型进程，推动我国工业物联网与人工智能交叉发展，促进形成有力支撑先进制造业发展的工业互联网生态体系。

参考文献：

[1] 谢志强, 刘冬梅. 考虑同层后道工序的柔性设备网络综合调度算法 [J]. 电子与信息学报, 2024, 46 (7): 2961 - 2969.

[2] GEWOHN M, USLANDER T, BEYERER J, et al. Digital real-time feedback of quality-related information to

inspection and installation areas of vehicle assembly [C] // 11th CIRP Conference on Intelligent Computation in Manufacturing Engineering, Procedia CIRP, 2018; 458-463.

[3] GUPTA A, SINTORN I. Efficient high-resolution template matching with vector quantized nearest neighbour fields [J]. Pattern Recognition, 2023, 151. [2023-06-26]. <https://doi.org/10.48550/arXiv.2306.15010>.

[4] YAO L. Analysis of information island in civil aviation industry under epidemic situation [C] // Proceedings of 2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology, IEEE, 2024; 57-60.

[5] 宋 今, 张河翔. 云边端融合的算网架构在智慧园区中的应用研究 [J]. 信息通信技术与政策, 2024, 50 (2): 55-62.

[6] 孟 维, 陈 赟. 基于云边端协同机器视觉算法自训练系统的研究 [J]. 江苏通信, 2024, 40 (2): 87-93.

[7] 李竞博, 马 礼, 李 阳, 等. 感传算协同工业互联网优化设计 [J]. 通信学报, 2023, 44 (6): 12-22.

[8] 胡楚丽, 程 萍, 程一凡, 等. 城市网格化综合管理服务的云边端协同计算框架 [J]. 测绘与空间地理信息, 2023, 46 (9): 1-5.

[9] 周 旭, 李 琢. 面向算力网络的云边端协同调度技术 [J]. 中兴通讯技术, 2023, 29 (4): 32-37.

[10] 王 丹, 杨子君. 基于“云边端”架构的直角机器人控制系统设计 [J]. 仪表技术与传感器, 2024 (2): 73-78.

[11] 刘 永. 基于云管边端架构的高速 AI 稽查系统设计与实现 [J]. 电脑与信息技术, 2021, 29 (1): 58-60.

[12] 张扬帆, 王 珂, 李 阳, 等. 基于云-边-端协同的区域级风电场大数据中心数据管理框架及优化运行方法 [J]. 高电压技术, 2024, 50 (11): 5151-5163.

[13] 柳义筠, 陈善平. 软件定义车联网云边端协同一体化架构的研究 [J]. 计算机时代, 2023 (4): 23-28.

[14] WANG C, BOCHKOVSKIY A, LIAO H. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors [J/OL]. arXiv: [2024-07-03]. <https://doi.org/10.48550/arXiv.2207.02696>.

[15] 李建鑫, 陈思光. 面向联邦学习标签翻转攻击的客户端选择防御方法 [J/OL]. 物联网学报, [2024-07-19]. <https://link.cnki.net/urlid/10.1491.TP.20240719.1121.002>.

[16] 冯岳璐, 王 浩, 杨小渝, 等. 高通量计算筛选的材料设计接口应用软件 [J]. 计算机系统应用, 2020, 29 (11): 104-113.

[17] 李映辰, 何宇斌, 何锡祺, 等. 基于云边协同的充电设施协调控制系统设计 [J]. 计算机测量与控制, 2024, 32 (6): 111-117.

[18] 陈泽众, 邓玉欣. 关于安全案例论证构建的综述 [J]. 软件学报, 2024, 35 (9): 4013-4037.

[19] 李世新, 汪永飞. 一种无线电密钥分发方案设计与实现 [J]. 中国信息化, 2024 (6): 43-46.

[20] 孟珞珈, 郭元兴, 盛 强, 等. 模块类网络安全产品的智能测试系统设计 [J]. 计算机测量与控制, 2024, 32 (1): 37-44.

[21] 冯岳璐, 王 浩, 杨小渝, 等. 高通量计算筛选的材料设计接口应用软件 [J]. 计算机系统应用, 2020, 29 (11): 104-113.

[22] 李映辰, 何宇斌, 何锡祺, 等. 基于云边协同的充电设施协调控制系统设计 [J]. 计算机测量与控制, 2024, 32 (6): 111-117.

[23] 陈泽众, 邓玉欣. 关于安全案例论证构建的综述 [J]. 软件学报, 2024, 35 (9): 4013-4037.

[24] 李世新, 汪永飞. 一种无线电密钥分发方案设计与实现 [J]. 中国信息化, 2024 (6): 43-46.

[25] 孟珞珈, 郭元兴, 盛 强, 等. 模块类网络安全产品的智能测试系统设计 [J]. 计算机测量与控制, 2024, 32 (1): 37-44.

[26] 李志明, 杨啸龙, 李天昊, 等. 基于 YOLOv8 的煤矿用钢丝绳损伤检测算法 [J]. 电子测量技术, 2024, 47 (9): 120-128.

[27] 梁 滨, 宋雪峰, 高 嘉, 等. 基于 YOLO 算法框架的曳引钢丝绳缺陷检测方法 [J]. 无损探伤, 2023, 47 (6): 1-4.

[28] 刘晓磊, 吴国群, 阚 哲. 基于深度学习的煤矿钢丝绳缺损检测方法研究 [J]. 煤炭工程, 2023, 55 (11): 148-153.

[29] 高 嘉, 刘 涛, 王显峰, 等. TWRD-Net: 一种用于曳引钢丝绳缺陷的实时检测网络算法 [J]. 仪器仪表学报, 2023, 44 (6): 223-235.

[30] WANG C, YEH I, LIAO H. Yolov9: Learning what you want to learn using programmable gradient information [J]. ArXiv, 2024, ArXiv: 2402.13616.

[31] YANG S, CAO Z, LIU N, et al. Maritime electro-optical image object matching based on improved YOLOv9 [J]. Electronics, 2024, 13 (14): 2774.

[32] LI X, WANG W, HU X, et al. Selective kernel networks [J]. IEEE, 2020, 1903.06586v2: 1-12.

(上接第 62 页)

[10] 温冯睿, 关海鸥, 马晓丹, 等. 近红外光谱图像处理的霉变稻谷检测方法 [J]. 光谱学与光谱分析, 2022, 42 (2): 428-433.

[11] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: unified real-time object detection [J]. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 779-788.

[12] 苏立鹏, 娄益凡, 杨吴奔, 等. 基于改进 Faster-RCNN 的起重机钢丝绳表面缺陷识别方法 [EB/OL]. 机电工程, 1-10. <http://kns.cnki.net/kcms/detail/33.1088.TH.20250305.1016.002.html>.

[13] 韩慧健, 邢怀宇, 张云峰, 等. 基于 Transformer 多元注意力的钢材表面缺陷视觉检测 [EB/OL]. 郑州大学学报 (工学版), 1-8. <https://doi.org/10.13705/j.issn.1671-6833.2025.05.009>.

[14] 黄 毅, 郭金梦, 任子誉, 等. 基于改进元学习的小样本钢丝绳表面损伤检测 [J]. 计算机工程与应用, 2025, 61 (13): 256-269.

[15] 方旭东, 于 正, 杨发展, 等. 基于注意力机制的轻量级矿井钢丝绳断丝检测算法研究 [J]. 中国煤炭, 2024,