

# Research Proposal

## 学位论文研究内容

### 研究目标

近年来，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。但随着计算机与互联网技术不断发展，传感器所产生的海量数据也得到了爆炸性的增长，传统云计算模式很难满足实时监测的要求，需要云端与边缘端协同工作，才能够为用户提供方便快捷经济的计算服务。同时深度学习不断进步，对于模型的性能要求更好，训练一个庞大模型需要的资源和算力在普通服务器上很难满足要求，在云端训练又很难灵敏响应变化。因此本论文的研究目标就是以工业缺陷检测为场景，从提出一种能够部署在边缘节点的轻量级算法，设计一种基于云计算和边缘计算协同的工业产品表面缺陷检测系统，使其能够在工业生产中提供敏捷反应、实时响应的模型训练与推理服务。

### 研究内容

在明确了本学位论文的研究目标之后，得出目前需要研究的主要研究内容如下：

1. 传统的表面缺陷检测系统方法有两种，分别是本地计算和云计算。本地计算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。使用云计算方案，传感器等设备会将所有数据上传至云端，随后在云端进行数据处理、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。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、边缘设备和云端设备三部分组成。本文主要研究边缘端与云端，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. 常见的缺陷检测算法可分为以YOLO为代表的的一阶段网络算法与以Faster R-CNN为代表的二阶段网络算法，一阶段网络有个较好的检测准确率，检测速度较快能够很好地满足工业领域实时性的要求。但是由于传统的YOLO系列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。一般地，大模型往往是单个复杂网络或者是若干网络的集合，拥有良好的性能和泛化能力，而小模型因为网络规模较小，表达能力有限。因此，可以利用大模型学习到的知识去指导小模型训练，使得小模型具有与大模型相当的性能，但是参数数量大幅降低，从而实现模型压缩与加速本文主要从蒸馏学习入手，改进其主干网络，使得其无需依靠昂贵的高性能设备就能完成深度学习的训练与计算。

### 拟解决的关键性问题

以上为本论文的研究目标以及研究的主要内容，下面将解释本论文拟解决的关键性问题

1. 传统网络结构复杂，训练消耗资源大，网络轻量化已经是计算机视觉领域一大研究热点，如何将深度学习应用于普通设备，大规模应用于工业现场，同样是工业领域面临的问题。已有的基于轻量化网络方法，尚处于研究阶段，人存在许多训练精度、稳定性、泛化性等问题。本文将从蒸馏学习入手，研究模型网络轻量化的问题，是其在轻量的前提下，依然能够保持较好的性能。
2. 在当前的工业缺陷检测场景中，缺陷样本会在使用过程中源源不断地产生，但是现有的方法一般是提前将网络训练好并直接部署在工业现场，而不考虑后续对网络的优化。本论文通过云边协同的方案，利用这些在检测过程中得到的新样本对网络进行优化，能够使网络的性能得到动态的更新，持续提高网络的准确率。
3. 随着目前深度学习网络的层数逐渐加深、参数逐渐增多，网络的训练将会耗费越来越多的算力，对计算设备的性能要求也会越来越高。但是由于成本的限制，一般情况下，厂家不会在工业现场部署大量性能

较强的计算设备。所以现有的深度学习网络可能无法在现场直接运行。本论文研究的方案将网络的训练和推理过程分离，降低了对现场设备性能的要求。

## 学位论文研究依据

### 选题依据与研究意义

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。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。

### 国内外研究现状

#### 云边协同

云计算是一种计算范式,它可以根据用户的需求随时随地为最终用户提供无限的计算资源,用户只需为使用的服务付费。云中可以提供各种类型的服务,如资源池、弹性和灵活性、可扩展性(水平和垂直)、性能高可用性、托管服务等[4]。正是因为具有强大的服务能力,云计算成为了所有业务之首,为全世界提供就业机会,近十年来被学术界和工业界广泛研究。然而,根据数据机构IDC的预测,2020年底将有超过500亿的终端与设备联网[5],从而产生海量的异构数据。此时,传统的云计算已经不能满足一些对实时性比较敏感的应用,并且将全部数据都上传到云数据中心也会给网络带宽带来很大的压力。因此,以解决数据传输延迟、降低网络带宽为目标的边缘计算正迅速兴起。边缘计算是指在网络边缘执行计算的一种新型计算模式。边缘计算中的下行数据表示云服务,上行数据表示万物互联服务,边缘计算的边缘是指从数据源到云计算中心的路径之间的任意计算和网络资源[5]。边缘计算架构中,用户数据不再需要全部上传到云数据中心,而是通过部署在网络边缘的边缘节点快速处理部分数据,从而大大减轻了网络带宽的压力,大幅降低了网络边缘端智能设备的能耗。为此,针对边缘计算的探索性研究已经广泛展开。随着其市场规模的逐渐扩大,边缘计算成为了与云计算同台竞技的解决方案。为了更好地结合云计算与边缘计算的优势,云边协同作为一种新型计算模式成为了新的研究趋势。

云计算在近些年已经获得了巨大的发展和应用,同时也已有部分边缘计算产品被逐步推出,但云边协同的发展仍处于探索阶段。随着数据密集型应用与计算密集型应用的增加,需要利用云计算强大的计算能力以及通信资源与边缘计算短时传输的响应特性来实现并完成相应的应用请求。通过两者协同工作、各展所长,将边缘计算和云计算协作的价值最大化[6],从而有效地提高应用程序的性能。目前,针对云边协同的研究大多数集中在物联网、

工业互联网、智能交通、安全监控等诸多领域的应用场景上,主要目的是减少时延、降低能耗以及提高用户体验质量等。Ren等[7]提出的云边缘协作方法能够有效地提高延迟性能。Ding 等[8]提出了一种云边缘协作框架,通过浅层卷积神经网络模型提供持续时间长、响应速度快的认知服务,给用户带来了良好的体验。Zhang等[9]在工业互联网中提出了一个 Cloud-Edge 协作的工业设备管理服务系统,在一定程度上提高了工业现场系统的响应速度,减轻了数据传输带来的网络带宽负载压力,推动了工业物联网向智能化发展。

## 缺陷检测

目标检测任务要求不仅判断输入数据包含物体种类,还要定位物体位置并用矩形框框出。现如今目标检测算法可分成两种:(1)传统目标检测算法,(2)基于深度卷积神经网络的算法。传统目标检测法由区域选择、特征信息提取[10]以及分类器构成。区域选择较多使用基于滑动窗口 [11] 的方法,特征信息主要涉及颜色、边缘、尺度不变特征(SIFT) [12] 以及方向梯度直方图(HOG)特征 [13],分类器则有支持向量机 [14,15] 和AdaBoost[16]。传统算法在区域选择以及特征提取上时间复杂度较高,最终分类准确率也较低。随着深度卷积神经网络不断开拓创新,将其融合到目标检测算法成为研究的热点。现如今,基于深度卷积神经网络目标检测算法[17]可分成两种:一种是基于候选区域的Two-Stage 检测算法,另一种为基于回归的 One-Stage 检测算法。

### 基于候选区域的 Two-Stage 检测算法

Grishick R 等人[18]于 2014 年提出 R-CNN,第一次将深度卷积网络融入到目标检测领域。该算法框架首先采用选择搜索算法(Selective Search,SS)[19]生成 2000 个候选框,并将不同尺寸候选框缩放至  $227 \times 227$ ;然后算法使用 AlexNet 卷积网络对候选框进行特征信息提取;最后使用 SVM 算法确定目标的类别,使用回归器对候选框进行位置矫正。He 等人[20] 提出 SSP-Net 目标检测算法,通过特征金字塔将由 SS 算法得到的候选框统一尺度,避免 R-CNN 中候选框卷积计算量。2015 年 Grishick R[21]基于 SSP-Net 思想改进 R-CNN 提出 Fast-RCNN,该算法创新提出 ROI Pooling 层,避免了对候选框缩放操作,有效提高网络计算速度。算法还替换特征提取网络为 VGG16,SVM 分类层替换为 Softmax 函数,采用多任务学习即同时实现目标类别预测与位置回归计算,有效降低模型参数量与计算量,在 Pascal VOC 2007 取得 70%的 mAP (mean Average Precision),但由于 Fast R-CNN 仍采用 SS 方法选择候选框,导致算法无法实现低延时检测。Ren S等人[22]提出了 Faster-RCNN 目标检测算法,该算法提出候选框生成网络(RPN),降低候选区域选择带来的复杂计算量。Faster-RCNN 将候选框生成、特征信息提取、目标分类和位置回归融合进单个算法框架,实现目标检测算法端到端训练,网络检测精度以及速度得到较高提升,在 VOC2007 测试集 mAP 达到 73.2%。;2016 年,Dai J 等人[23]提出了 R-FCN 目标检测算法,该算法采用残差网络模块作为特征提取网络,使用卷积神经网络代替 ROI Pooling 层之后的全连接层,有效减少模型参数量。R-FCN 还加入位置感受得分图解决卷积平移不变性,将目标位置信息添加进 ROI Pooling 层中,在 PascalVOC2007 测试集检测精度提升到 80.5%。但是该模型计算复杂,检测实时性较低。

### 基于回归的 One-Stage 检测算法

2016 年,Redmon J 等人[24]首次将目标检测看待为一种回归问题,提出单阶段目标检测算法 YOLO V1,输入图像通过单次卷积就能得到图像中物体分类与所在位置结果。YOLO V1 相较于两阶段目标检测器在目标检测速度实现大幅提升,但算法对于小目标、密集分布物体预测精度较低,在 VOC2007 上 mAP 仅为 66.4%。2017 年,Redmon J 等人[25]在 YOLO V1 基础上提出 YOLO 9000,该网络对卷积层输出都采用批量归一化处理,同时采用了锚框机制并设计 Darknet-19 作为 YOLO 9000 的特征提取网络,有效提升了目标检测算法准确度,在 VOC2007 数据集上的 mAP 达到 78.6%;2018 年,Redmon J 等人[26]基于 YOLO 9000 提出 YOLO V3,它使用了特征提取效果更好的 Darknet-53,并且还采用多尺度分类预测方法以及锚框机制,使得网络在保证检测速度同时,预测精度也有效提高。2020 年,Bochkovskiy A 等人[27]提出了 YOLO V4 算法,该算法是在原有 YOLO 系列目标检测算法基础上,采用了近些年 CNN 领域中最优秀的优化策略,从数据预处理、主干网络设计、网络训练方法、激活函数、损失函数等各个方面着手对算法进行优化处理,在 COCO 数据集上,可达 43.5%AP,速度达到 65FPS。Liu W 等人[28]提出 SSD 目标检测框架,该方法融合 YOLO 以及 Faster-RCNN 中锚框机制,采用 VGG16部分卷积

层用于特征提取,并新增 6 个卷积层获得更多特征信息,使用多尺度检测实现对不同大小物体精确检测,在 VOC2007 上测试得到 75.1%的 mAP,检测速度达到 58FPS。

## 发展态势

## 选题在理论研究/实际应用的意义和价值

## 主要参考文献

- [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] KUMAR M, SHARMA S C, GOEL A, et al. A comprehensive survey for scheduling techniques in cloud computing[J]. Journal of Network and Computer Applications, 2019, 143:1-33. [5] SHI W S, SUN H, CAO J, et al. Edge computing: a new computing model for the Internet era [J]. Journal of Computer Research and Development, 2017, 54(5):907-924. [6] BOUSSELHAM M, BENAMAR N, ADDAIM A. A new Security Mechanism for Vehicular Cloud Computing Using Fog Computing System[C]//2019 International Conference on Wireless Technologies, Embedded and Intelligent Systems (WITS). IEEE, 2019:1-4 [7] REN J, HE Y, YU G, et al. Joint communication and computation resource allocation for cloud-edge collaborative system[C] // 2019 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2019:1-6. [8] DING C, ZHOU A, LIU Y, et al. A Cloud-Edge Collaboration Framework for Cognitive Service[J/OL]. IEEE Transactions on Cloud Computing, 2020. <https://ieeexplore.ieee.org/abstract/document/8895891>. [9] ZHANG H, CHEN S, ZOU P, et al. Research and Application of Industrial Equipment Management Service System Based on Cloud-Edge Collaboration[C] // 2019 Chinese Automation Congress (CAC). IEEE, 2019:5451-5456. [10] Nixon M S. Feature Extraction and Image Processing[M]. Publishing House of Electronics Industry, 2013. [11] Xu Y, D Xu, Lin S, et al. Sliding Window and Regression Based Cup Detection In Digital Fundus Images for Glaucoma Diagnosis[J]. Springer-Verlag, 2011. [12] Lowe DG. Distinctive Image Features from Scale-Invariant Keypoints[J]. International Journal of Computer Vision, 2004, 60(2):91-110. [13] 刘方园, 王水花, 张煜东. 方向梯度直方图综述[J]. 计算机工程与应用, 2017, 53(19). [14] Lin, Zhu, Yang, et al. Large-scale image classification: Fast feature extraction and SVM training. IEEE Computer Society, 2011. [15] 乔凤娟, 郭红利, 李伟等. 基于SVM的深度学习分类研究综述[J]. 齐鲁工业大学学报, 2018, 32(5):39-44. [16] 张溪樾. 基于 Adaboost 的行人检测综述[J]. 电子制作, 2019, (1):59-61. [17] 黄健, 张钢. 深度卷积神经网络的目标检测算法综述[J]. 计算机工程与应用, 2020, 56(17):12-23. DOI:10.3778/j.issn.1002-8331.2005-0021. [18] Girshick R, Donahue J, Darrell T, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[J]. IEEE Computer Society, 2013. [19] Uijlings, J.R.R., Van De Sande, et al. Selective search for object recognition[J]. International Journal of Computer Vision, 2013, 104(2):154-171. [20] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 37(9):1904-16. [21] Girshick R. Fast R-CNN[J]. Computer Science, 2015. [22] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6):1137-1149. [23] Dai J, Li Y, He K, et al. R-FCN: Object Detection via Region-based Fully Convolutional Networks[J]. Advances in Neural Information Processing Systems, 2016. [24] Redmon J, Divvala S, Girshick R, et al. You Only Look Once: Unified, Real-Time Object Detection[C]// Computer Vision & Pattern Recognition. IEEE, 2016. [25] Redmon J, Farhadi A. YOLO9000: Better, Faster, Stronger[J]. IEEE Conference on Computer Vision & Pattern Recognition, 2017:6517-6525. [26] Redmon J, Farhadi A. YOLOv3: An

Incremental Improvement[J]. Computer Science,2018,4(1):1-6. [27]Bochkovskiy A, Wang C Y, Liao H. YOLOv4: Optimal Speed and Accuracy of Object Detection[J]. 2020. [28]Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[J]. Springer Cham, 2016.

已有工作积累和研究成果

## 学位论文研究计划及预期目标

### 主要理论

完成本次毕业设计预计要掌握的理论有：云计算相关理论，边缘计算相关理论，docker、kubernetes技术，计算机网络，深度学习、蒸馏学习、YOLO系列算法相关知识

### 研究方法

1. 首先需要熟悉并掌握docker、k8s技术，并在此基础上学习KubeEdge系统的实现
2. 在本地实现KubeEdge的部署，能够完成基本流程的实现
3. 借助KubeEdge平台的各API，实现接入设备的管理，模型在云中心节点的训练，在边缘节点的推理部署
4. 对网络模型进行优化，使其实时性等关键指标能够获得提升以满足工业需求
5. 实现云中心能够依靠端节点采集的数据，实现模型的更新与下发至边缘节点

### 技术路线

1. 以KubeEdge为总体实现的系统，使其能完成边缘端与云端协同工作
2. 以YOLO框架为基础，选取其中适合的版本，实现缺陷检测算法
3. 通过蒸馏学习，对模型各项参数进行优化，实现网络的轻量化优化

### 实施方案

### 计划可行性

### 条件落实情况

联合培养单位提供一台超高性能的服务器与若干台性能强劲的服务器。性能超高的服务器可以作为整套云边系统中的云中心节点，其他若干台服务器可以作为边缘节点，设备运行情况与网络状况良好，方便搭建实验系统。

### 可能存在的问题及解决办法

可能存在问题：云中心服务器在极少数情况下可能出现无法连接的情况 解决办法：借助KubeEdge与消息队列的方式，使得在无法互相通信的情况下依然可以自主运行，并将消息暂存至队列，待网络恢复后能够及时进行处理

### 预期创新点及成果形式