年 级 2021 学 号 211250243 学生姓名 聂尔聪 指导教师 邬向前 职 称 教授

第二导师 葛季栋 职 称 副教授 提交日期 2025年 5月 7日 南京大学本科生毕业论文(设计、作品)中文摘要

题目:基于无监督深度学习的工业缺陷检测系统的设计和实现 院系:软件学院 专业:软件工程 本科生姓名:聂尔聪

指导教师(姓名、职称):邬向前 教授 葛季栋 副教授

摘要:

产品质量是工业制造的基础，随着工业制造业智能化的发展，对于产品质量检测高效性和准确性的要求越来越高，而缺陷检测正是工业制造业产品检测中不可或缺且至关重要的一环，是保证金属、芯片、纺织物等工业制品质量的重要方法。众所周知，传统的人工检测存在效率低、成本高、稳定性差等缺点，必然会被更加高效、经济的检测方法所替代。近年来，随着计算机视觉、工业成像、深度学习等技术的兴起，基于视觉的工业缺陷检测取得了重大突破，也为检测方法的更新提供了可能——建立神经网络，对样本进行训练，提取特征，实现缺陷检测自动化。最初的有监督缺陷检测需要大量的有缺陷标签训练样本，这些有缺陷的标签数据不仅难以获得、成本高，且泛化能力非常弱，已不能满足当前主流的工业生产需求，于是无监督深度学习缺陷检测方法应运而生。无监督学习仅依据正常样本进行，能够通过特征挖掘实现对缺陷的识别，无疑更适合复杂的工业场景，成为研究趋势。

然而，这些无监督系统的样本管理不足、检测流程或参数设置复杂、辅助信息有限、多尺度缺陷检测能力不足、界面布局混乱和功能模块割裂等问题，无法满足工业场景快速部署的需求。因此，本研究拟基于开源项目SimpleNet和AnomalyGPT，针对工业检测需求，开发无监督系统，旨在实现集样本管理、模型训练、缺陷检测模块于一体的基于无监督学习的缺陷监测系统，以减少标注数据的依赖，提高检测的灵活性，同时可通过模块化设计优化和增强功能，使系统用。

的易用性与适应性。具体研究内容如下:

2.基于简单的模型训练模块为了解决在自监督缺陷检测中，标注数据的大量使用、训练参数设置复杂、系统模型管理困难等问题，构建了由基于SimpleNet的无监督缺陷检测、用户指导的参数映射和模型状态管理等部分组成的模型训练模块，其中，用户指导的参数映射将复杂的模型参数映射关系替换为“精确程度-速度-缺陷大小”等选项。

关键词:计算机视觉;无监督深度学习;工业缺陷检测;自动化检测;缺陷检测

系统 南京大学本科生毕业论文(设计、作品)英文摘要

on Unsupervised Deep Learning

DEPARTMENT: Software Institute

SPECIALIZATION: Software Engineering

UNDERGRADUATE: Nie Ercong

ABSTRACT:

KEYWORDS: Computer Vision; Unsupervised Deep Learning; Industrial Defect De-

tection; Automated Detection; Defect Detection System

目 录

目 录 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . V

插图目录 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . IX

表格目录 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . XI

第一章 引言 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

1.1 研究背景及意义 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

1.2 国内外研究现状 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

1.2.1 工业缺陷检测技术 . . . . . . . . . . . . . . . . . . . . . . . . 2

1.2.2 工业缺陷检测系统 . . . . . . . . . . . . . . . . . . . . . . . . 3

1.3 研究内容与主要工作 . . . . . . . . . . . . . . . . . . . . . . . . . . 4

1.4 论文组织结构 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

1.5 本章小结 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

第二章 基本概念和相关工作 . . . . . . . . . . . . . . . . . . . . . . . . 7

2.1 基本概念 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

2.1.1 缺陷与异常的概念 . . . . . . . . . . . . . . . . . . . . . . . . 7

2.2 SimpleNet . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.2.1 SimpleNet概述 . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.2.2 SimpleNet分析 . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.3 AnomalyGPT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 9

2.3.1 AnomalyGPT概述 . . . . . . . . . . . . . . . . . . . . . . . . 9

2.3.2 AnomalyGPT分析 . . . . . . . . . . . . . . . . . . . . . . . . 10

2.4 技术栈 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

2.4.1 开发框架 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

2.4.2 数据库 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

2.4.3 部署工具 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

2.4.4 其它 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11

2.5 本章小结 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

第三章 系统设计 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

3.1 总体描述 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

3.1.1 系统目标 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

3.1.2 用户特征 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13

3.2 需求分析 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

3.2.1 功能性需求 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

3.2.2 用例描述 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

3.2.3 非功能性需求 . . . . . . . . . . . . . . . . . . . . . . . . . . 21

3.3 系统架构设计 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

3.3.1 总体设计 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

3.3.2 "4+1"视图 . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

3.3.3 模块划分 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

3.4 数据库设计 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

3.4.1 概述 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

3.4.2 E-R图设计 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

3.4.3 核心表结构 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

3.4.4 数据关系与完整性 . . . . . . . . . . . . . . . . . . . . . . . . 31

3.5 本章小结 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

第四章 系统实现 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

4.1 开发环境 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

4.2 核心模块实现 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

4.2.1 项目管理模块 . . . . . . . . . . . . . . . . . . . . . . . . . . 33

4.2.2 样本管理模块 . . . . . . . . . . . . . . . . . . . . . . . . . . 35

4.2.3 模型训练模块 . . . . . . . . . . . . . . . . . . . . . . . . . . 38

4.2.4 缺陷检测模块 . . . . . . . . . . . . . . . . . . . . . . . . . . 40

4.3 接口与其他 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

4.3.1 API接口 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

4.3.2 其他 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

4.4 部署 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50

4.4.1 服务器端部署 . . . . . . . . . . . . . . . . . . . . . . . . . . 50

4.4.2 客户端部署 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51

4.5 本章小结 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51

第五章 实验 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53

5.1 数据集 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53

5.2 评估指标 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54

5.3 实验设计 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54

5.4 结果分析 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54

5.4.1 SimpleNet异常检测 . . . . . . . . . . . . . . . . . . . . . . . 54

5.4.2 AnomalyGPT辅助分析 . . . . . . . . . . . . . . . . . . . . . 55

5.5 本章小结 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55

第六章 总结与展望 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

6.1 工作总结 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

6.2 未来展望 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57

参考文献 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59

致 谢 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61

插图目录

1-1 HALCON系统进行缺陷检测的界面 . . . . . . . . . . . . . . . . . . 4

2-1 SimpleNet概述。在训练阶段,样本被送入预训练的特征提取器以

获取局部特征。然后,利用特征适配器将预训练的特征适配到目标域,并通过添加高斯噪声来合成异常特征。最后分别用作正样

本和负样本来训练鉴别器。另外,推断时会移除异常特征生成器。 8

2-2 在 MVTec AD 基准测试中推理速度 (FPS) 与 I-AUROC 的对比。SimpleNet在准确性和效率上都远远超过了以前的所有方法。 . . . 9

2-3 AnomalyGPT与现有IAD方法以及现有LVLMs的比较现有

的IA D只能给出异常分数,需要手动设置阈值,而图9异常检测模型LVLMs不能检出图像异常。 AnomalyGPT不仅能够给出图像的异常信。

息，也可以用来指示不正常的存在和位置。 ............10。。

3-1 系统用例图 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

3-2 系统总体架构设计图 . . . . . . . . . . . . . . . . . . . . . . . . . . 22

3-3 "4+1"视图 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

3-4 逻辑视图 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

3-5 物理视图 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24

3-6 过程视图 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

3-7 开发视图 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

3-8 模块划分图 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

3-9 E-R图 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

3-10 样本组表 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

3-11 样本表 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

3-12 模型表 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

3-13 检测结果表 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

3-14 模型训练样本表 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

3-15 模型推理样本表 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 31

4-1 开始界面 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34

4-2 项目结构图 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

4-3 样本界面 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

4-4 样本组列表 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 36

4-5 裁剪画面 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 38

4-6 模型界面 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

4-7 参数设置 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

4-8 训练进度 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40

4-9 检测界面 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 41

4-10 检测过程 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42

4-11 AI对话 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 42

4-12 可视化分析标签页,又为检测报告界面,包含缺陷位置分布图 . . 43

4-13 区域特征分布图 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 44

4-14 统计数据标签页。包含缺陷类型分布图 . . . . . . . . . . . . . . . . 45

4-15 详细信息标签页 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

4-16 加载进度 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48

4-17 加载动画 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49

4-18 流程引导界面 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50

5-1 MVTec 异常检测数据集的五个纹理和十个对象类别的示例图像。

对于每个类别,都显示了一个无异常和异常的示例。上面一行显

示了输入图像,下面一行是特写视图。 . . . . . . . . . . . . . . . . 53

表格目录

3-1 缺陷检测项目管理用例 . . . . . . . . . . . . . . . . . . . . . . . . . 16

3-2 训练样本管理用例 . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

3-3 无监督学习模型训练用例 . . . . . . . . . . . . . . . . . . . . . . . . 18

3-4 无监督缺陷检测用例 . . . . . . . . . . . . . . . . . . . . . . . . . . 19

3-5 AI辅助检测分析用例 . . . . . . . . . . . . . . . . . . . . . . . . . . 20

3-6 缺陷检测结果报告用例 . . . . . . . . . . . . . . . . . . . . . . . . . 21

4-1 API接口说明 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 47

4-2 模型相关 API接口说明 . . . . . . . . . . . . . . . . . . . . . . . . . 47

4-3 样本组相关 API接口说明 . . . . . . . . . . . . . . . . . . . . . . . . 48

5-1 不同精度级别下异常检测的性能对比。其中,ACC\_LEVEL为所

选精度级别,I-AUROC 为图像级 AUROC,P-AUROC 为像素级

AUROC,TIME为单张样本检测耗时。 . . . . . . . . . . . . . . . . 55

5-2 不同速度级别下异常检测的性能对比。其中,SPEED\_LEVEL为

所选速度级别,I-AUROC为图像级 AUROC,P-AUROC为像素级

AUROC,TIME为单张样本检测耗时。 . . . . . . . . . . . . . . . . 55

5-3 不同缺陷类别下 AI 辅助判定缺陷存在性与定位缺陷的性能。其

中,DEFECT\_CLASS为样本缺陷类别,EXISTANCE\_ACC为异常

存在性准确率,POSITION\_ACC为缺陷位置准确率。 . . . . . . . 55

第一章 引言

1.1 研究背景及意义

适用于大规模工业生产的要求[3]。

但是基于规则的传统图像处理算法往往只适用于部分场景并且效果较为稳定，而对于一些纹理复杂、缺陷种类多样的背景则难以实现良好的适应性，需要频繁调参；随着深度学习的发展，有监督学习逐渐以其强大的理解与提取产品缺陷特征的的能力在检测准确率和检测时间上实现了双突破，但是其存在两个问题，一是在实际工业生产制造中，缺陷样本往往数量少并且多样化，同时收集大量带有标注的异常样本成本较高；二是实际生产制造中，生产过程的复杂性导致缺陷模式的多样化以及缺陷类型的不可预测性，而现有的基于有监督的缺陷检测方法对于未培训过的新型缺陷类型缺乏泛化能力，难以适应实际生产制造中快速更迭的生产工艺。而近些年发展起来的基于无监督学习的缺陷检测方法，其可以自动学习未知特征和模式，仅依据正常样本即可进行有效训练，逐渐成为了另一研究热点。经过对现有无监督检测方法的观察，发现其存在三个问题，一是计算量极大，例如某些基于GANs的方法，其检测精度较高但是对于检测的计算开销极高，难以满足实际工业生产制造中实时检测的需求；二是所需参数配置过于专业，导致实际工程应用门槛较高，难以满足非专业用户的需求；三是检测结果信息过少，检测方法对于工业生产制造中的缺陷检测的辅助作用不明显。除此之外，现有方法大多只是给出了异常的评分，而不能直接判定异常与否，需要人为事先设置阈值，不能很好地适应动态的生产情况。

1.2 国内外研究现状

1.2.1 工业缺陷检测技术

基于随机森林(Random Forest)等机器学习算法对缺陷类型进行分类，取得了较好的缺陷识别和分类的效果。但此类方法一般需要人为设定环境，只能用于简单场景和规则缺陷，无法应用于实际复杂的工业生产环境。

针对深度学习中缺陷样本过少的问题，无监督学习的方法被广泛研究。无监督缺陷检测方法是利用正常样本训练，并学习其分布模式，对缺陷样本进行检测。其不依赖于标注，不受缺陷类型的限制。根据数据使用的维度不同，无监督方法可分为基于图像相似度的无监督方法和基于特征相似度的无监督方法。基于图像相似度的有自编码器(AE)、变分自编码器(VAE)和生成对抗网络(GAN)等。

包括深度一类分类、流模型和教师-学生架构等[1]。

1.2.2 工业缺陷检测系统

图 1-1 HALCON系统进行缺陷检测的界面 工业落地。

1.3 研究内容与主要工作

本研究基于开源的无监督深度学习检测方法SimpleNet和缺陷检测大型视觉语言模型AnomalyGPT，针对其存在在系统侵入性以及用户可用性和易用性方面的不足设计实现无监督学习的缺陷检测系统。本文

的主要工作分为以下六个部分: 1. 基于 SimpleNet构建无监督缺陷检测框架,实现模型训练与检测核心功能。

2. 基于 AnomalyGPT实现视觉语言模型辅助判定与分析功能。

3. 实现裁剪、缩小域、数据增强等训练样本预处理操作。 4. 实现样本组和模型的管理,包括模型参数的动态配置。

5. 实现缺陷检测报告功能,完善检测流程。

6. 在实现上述功能的基础上,根据人机交互启发式原则,设计一个用户友好的系统。

1.4 论文组织结构

本文围绕无监督学习的缺陷检测系统的设计与实现展开,全文共分为六章, 具体组织结构如下:

第一章:引言。对文章研究背景、研究意义、研究现状、研究内容、论文结构进行介绍。第二章:基本概念与相关工作。对无监督异常检测的基本概念进行介绍,对系统开发相关工作与成果进行介绍,对技术栈进行介绍。第三章:系统设计。对研究项目进行总体描述,进行需求分析,并进行系统架构设计与数据库设计。第四章:系统实现。对开发环境进行介绍,对系统核心模块进行介绍,对系统接口与其他实现进行介绍,对部署方式进行介绍。第五章:实验。对数据集、实验设计、实验结果进行分析。第六章:总结与展望。对课题研究进行总结,对课题局限性进行分析,对后续工作进行展望。

1.5 本章小结

第二章 基本概念和相关工作

2.1 基本概念

2.1.1 缺陷与异常的概念

缺陷为产品表面存在裂纹、划痕、凹陷等常见的物理损伤；异常为图像不同于正常模式出现颜色、纹理或形状异常变化[6]。而在工业缺陷检测中，缺陷被看作是异常的，因此文章中“缺陷”是指在实际生产过程中，因生产工艺、生产设备、现场环境等因素的影响，使产品呈现与正常状态不同的不常见的表面异常变化[4]，例如瓶底表面裂纹、划痕，木板表面磨损，皮革表面污渍，螺母表面锈迹等。

2.2 SimpleNet

2.2.1 SimpleNet概述

2.2.2 SimpleNet分析

SimpleNet能够检测出从微弱变化到结构缺陷等不同阶段的异常，且适用于缺陷样本量少的工业场景。如图2-2，SimpleNet在MVTec AD数据集上的异常检测中AUROC高达99.6%，并且精准性和效率都远远高于之前的方法，满足了工业质检的实时性和质量要求。综上所述，SimpleNet拥有着优秀的性能以及推理速度，并且易于训练和应用，因此本文采用SimpleNet构建基于工业场景的无监督缺陷检测系统。

2.3 AnomalyGPT

2.3.1 AnomalyGPT概述

AnomalyGPT[10]是中国科学院提出的基于大型视觉语言模型（LVLMs）的IAD方法。如图2-3所示，以上现有的IAD方法以及LVLM都不能很好地解决工业异常检测问题。而AnomalyGPT使用合成的异常视觉文本数据微调LVLM，将IAD知识融入到模型中，增强了模型对图像内在细节的理解，是第一个将LVLM应用到工业异常检测领域的方法，其贡献如下：（1）可以不需要阈值手动调整，支持异常的检测定位，同时支持多轮对话。（2）设计了一个轻量级的视觉文本特征匹配解码器，解决了大语言模型无法解释细粒度图像细节的问题，同时突破了输出形式的限制，以图像热图、文本和图像组合三种形式提供更多元的信息。再使用提示嵌入（prompt embedding）进行微调，并利用缺陷检测数据与LVLM预训练数据进行训练，这使得LVLM原有的上下文理解和对话管理能力得以保留，多轮对话成为可能。最后，模型具有极强的可迁移性，通过上下文少样本学习，可快速迁移到新的数据集，表现出了非常高的性能。

图 2-3 AnomalyGPT、现有 IAD方法和现有 LVLMs之间的比较。现有的 IAD方法只能提供异常分数,需要手动设置阈值,而现有的 LVLMs无法检测图像中的异常。AnomalyGPT不仅能提供图像信息,还能指示异常的存在和位置。

2.3.2 AnomalyGPT分析

大多数的IAD方法，包括SimpleNet，仅提供测试样本的异常分数，需要将正常和异常样本的阈值手动设置，降低了缺陷检测系统的实际应用体验。因此，本文采用AnomalyGPT直接判断异常是否存在，适用于实际生产环境。除了异常位置和是否存在这两个必须参数之外，AnomalyGPT可提供图像相关的信息，并可以与用户交互式问答，增强检测的解释性和实用性。除此之外，AnomalyGPT还可以利用已有的少量正常样本进行学习，从而快速适应未见对象。在MVTec AD数据集上，AnomalyGPT仅使用了一个正常样本，就可以达到86.1%的准确率，94.1%的图像级AUC，95.3%的像素级AUC。这些特征表明AnomalyGPT具有良好的泛化能力，适合于缺陷样本少、缺陷种类多样的实际工业场景。

2.4 技术栈

2.4.1 开发框架

Qt是1991年由Qt Creator开发的用于开发跨平台的C++图形用户界面应用程序（GUI）的跨平台工具包，可以用于Windows、Linux、和Mac OS等桌面应用的开发，也可以用于iOS、Android等嵌入式开发平台。 Qt是一个面向对象的开发工具包，可以在创建新的对象时进行扩展，并且可用于组件编程。提供丰富的API以与不同的外部程序进行通信，并且有一个活跃的社区，提供各种书籍和教程，能够帮助开发者快速入门。

Qt Designer是qt官方开发的一个用于图形化设计GUI界面的工具，通过拖拽可视化小控件，用户可以轻松创建复杂的用户界面，而不需要编写代码。在提高UI界面开发效率的同时，也让视图和逻辑实现了解耦。图界面设计完成后，只需处理业务逻辑代码。

2.4.2 数据库

MySQL是最流行的开源关系型数据库管理系统，它采用结构化查询语言SQL进行数据管理，支持ACID特性，适用于Windows、Linux、MacOS等多种操作系统。 MySQL具有体积小、速度快、成本低、可靠性高的特点，被广泛应用于各种网站和应用程序中。

2.4.3 部署工具

PyInstall是一个跨平台的 Python应用程序打包工具,它能自动检测 Python

脚本依赖的模块和库,包括资源文件,将它们打包成一个独立的可执行文件。

2.4.4 其它

Sobel算子是经典的边缘检测算子，主要用于图像处理中寻找图像的边缘点。其采用两个3\*3的卷积核来计算图像每个像素点在水平和竖直方向的灰度变化，变化较大的位置即为边缘点。 Sobey算子可以提取产品的表面边缘点、划痕等线形特征以及纹理变化情况。

DBSCAN是一种密度聚类方法，该算法能够将给定的数据点划分为不同的簇，并且能够发现噪声点。它将簇定义为密度相连的最大值集合，簇生成的两个关键参数是邻域半径Em和最小样本数量MinPts，其中簇的核心点是ε邻域内MinPts个样本点，边界点是不属于核心点且在某个核心点的邻域内的样本点，噪声点既不是核心点也不是边界点。

2.5 本章小结

本章介绍了工业缺陷检测的基本概念以及检测技术的发展情况。首先对缺陷与缺陷检测进行了概念区分，接着介绍了SimpleNet方法和AnomalyGPT模型的工作原理和优点，最后介绍本系统的技术栈，包括开发框架、数据库、部署工具以及系统所依赖的库和算法。

第三章 系统设计

3.1 总体描述

3.1.1 系统目标

本系统是工业质检的辅助工具，帮助用户快速找到工业产品表面缺陷，提升检测效率。本系统是跨平台的桌面程序，前端采用QT框架及PySide6库开发，后端采用FastAPI框架，MySQL数据库作为后库。本系统主要采用样本管理模块、模型训练模块、缺陷检测和报告模块。本应用可实现样本编辑管理、模型可视化训练、缺陷实时检测、AI智能判别以及自动生成缺陷报告功能。帮助工业质检用户方便、快捷、直观地检测各种工业产品表面缺陷，并根据用户需求平衡检测的精准度与检测速度，满足各种尺寸的缺陷检测。

该系统的算法核心基于S im leNe和AnomalyGPT，可提供仅基于正常样本进行的无监督学习、缺陷检测以及智能化的结果解释。同时，系统应提供一个用户能进行参数配置的友好的界面，在专业用户和业余用户两个层面上都能够具备人机交互启发式原则，如提供系统状态的可见性、一致性、标准化等。另外，系统应能支持多线程，同时满足一定的数据处理效率。

3.1.2 用户特征

本系统面向工业质检人员和从事无监督学习缺陷检测的科研人员，使用者希望通过系统更加高效地检测工业产品表面缺陷，提升工业质检效率，降低人工成本。其中，鉴于工业质检人员的工作特点，部分质检员可能具有多年视觉检测的工作经验，但对于深度学习模型原理和参数配置缺乏专业知识，同时，传统工业检测习惯根深蒂固，部分质检员对于计算机辅助检测系统的使用存在一定门槛；而对于科研人员，他们可能需要更加灵活地配置系统参数，或需要获得更加细化的检测结果。因此，本系统需要尽可能地简化操作，提供参数设置可视化界面，同时支持从易用性到专业性的不同系统功能，既能够满足工业质检员快速上手的需求，又可以方便科研人员对算法进行细致的调参。

3.2 需求分析

3.2.1 功能性需求

3)模型训练功能。不必知道算法内部具体实现机制,用户可以根据可视化效果对模型进行细致调节,将嵌入维数、层数、补丁大小等系统参数映射为“精确度-速度-缺陷尺寸”等选项;同时,训练过程也可可视化,随时显示训练时长、训练效果等。

6)检测结果报告功能。可查看检测结果缺陷数据统计，检测结果报告，检测数据的位置分布、聚类分析等。可生成自动检测报告，提供丰富的可视化图表，并可以导出为PDF，方便分享和存档，报告内容不仅有技术数据还有可对工艺改进的建议。

系统用例图如图 3-1所示。 图 3-1 系统用例图

3.2.2 用例描述

1. 用户在缺陷检测系统进行项目管理的用例描述如下表 3-1所示。 表 3-1 缺陷检测项目管理用例 名称 内容描述

ID 01 用例名称 项目管理 参与者 用户 描述 允许用户创建新项目、打开已有项目、管理项目元数据和配置信息

触发条件 用户启动系统或选择项目管理功能 前置条件 系统正常运行

后置条件 创建项目或项目信息被用户修改时,系统自动保存 优先级 低 正常流程 1. 用户启动系统,显示项目管理界面

2. 用户选择"新建项目"或"打开项目"或"最近项目"

3. 新建项目时,用户输入项目名称和存储位置

4. 打开项目或最近项目时,用户选择本地存储的项目文件

5. 系统加载项目,初始化配置信息

扩展流程 3a. 如果项目名称已存在或非法,提示用户选择其他名称 4a. 如果项目文件损坏,显示错误信息

4b. 用户可通过拖放方式打开项目文件 4c. 如果最近项目不存在,自动从历史列表中删除 特殊需求 无

2. 用户在模型训练前导入样本并进行管理的用例描述如下表 3-2所示。 表 3-2 训练样本管理用例 名称 内容描述

ID 02 用例名称 样本管理 参与者 用户 描述 允许用户创建样本组、导入样本图像、进行预处理和数据增强,以及

上传至服务器 触发条件 用户进入样本管理流程 前置条件 已创建或打开项目 后置条件 样本上传至服务器 优先级 高

正常流程 1. 用户创建新样本组或导入已有样本组

2. 用户导入图像文件或文件夹

3. 系统显示导入的样本列表

4. 用户可选择图像进行裁剪、缩小域等预处理

5. 用户可对样本进行数据增强或添加伪缺陷

6. 用户将样本上传至服务器

扩展流程 2a. 如果导入非图像文件,系统过滤不支持的文件 4a. 用户可批量应用部分预处理操作

6a. 大批量图像处理时显示进度条 6b. 上传失败时显示错误信息 特殊需求 无

3. 用户在训练基于无监督学习的缺陷检测模型的用例描述如下表 3-3所示。 表 3-3 无监督学习模型训练用例

名称 内容描述 ID 03 用例名称 模型训练 参与者 用户

描述 允许用户配置和训练无监督缺陷检测模型,提供训练可视化和参数管 理 触发条件 用户进入模型训练流程

前置条件 已上传训练样本至服务器 后置条件 模型训练完成 优先级 高 正常流程 1. 用户创建模型组或导入已有模型组

2. 用户选择要训练的样本组

3. 用户通过直观界面配置模型参数

4. 用户启动训练过程

5. 系统显示训练进度和实时性能指标

6. 训练完成后,系统修改模型状态并通知用户

扩展流程 3a. 用户可选择自定义参数配置 4a. 训练过程中用户可手动终止训练 特殊需求 无

4. 用户使用训练好的模型进行缺陷检测的用例描述如下表 3-4所示。 表 3-4 无监督缺陷检测用例 名称 内容描述

ID 04 用例名称 缺陷检测 参与者 用户 描述 允许用户导入检测样本,使用训练好的模型进行缺陷检测,并查看实

后置条件 无 优先级 高 正常流程 1. 用户创建检测样本组并导入待检测图像

2. 用户选择处于空闲状态的模型

3. 用户启动检测过程

4. 系统执行检测并展示实时结果

5. 检测完成后,模型状态还原为空闲

扩展流程 1a. 支持批量导入检测样本 3a. 用户可调整阈值优化结果显示 5a. 用户可切换原图和热力图视图

名称 内容描述 ID 05 用例名称 AI智能分析 参与者 用户 描述 允许用户通过自然语言交互获取缺陷分析结果

触发条件 用户在缺陷检测前后选择 AI辅助分析 前置条件 已导入检测样本 后置条件 无 优先级 中

正常流程 1. 用户选择需要分析的检测样本

2. 系统加载大模型分析功能

3. 用户通过自然语言界面提问

4. 大模型分析图像并提供缺陷判断、位置描述和图像相关信息等

5. 用户可进行多轮对话深入了解分析结果扩展流程 3a. 用户可同时选择多张图像进行对比分析

5a. 用户可保存分析会话记录 特殊需求 无 6. 用户在缺陷检测后生成检测结果报告的用例描述如下表 3-6所示。

表 3-6 缺陷检测结果报告用例 名称 内容描述 ID 06 用例名称 检测结果报告 参与者 用户

描述 允许用户查看缺陷统计分析,生成和导出报告 触发条件 用户选择结果分析与报告

前置条件 已完成缺陷检测并有结果可供分析 后置条件 无 优先级 中 正常流程 1. 用户选择检测样本组

2. 用户配置报告参数,如聚类参数和块大小

3. 系统进行缺陷统计与聚类分析,生成缺陷分布热图和统计图表

4. 用户查看分析结果

5. 用户选择生成报告

6. 系统生成 PDF格式报告

扩展流程 无 特殊需求 无

3.2.3 非功能性需求

4)可维护性:系统应能方便地进行维护和更新,如添加新功能、修复已知问题、优化性能等。

6)可移植性:客户端应支持跨平台部署,如Windows、Linux、macOS等。

3.3 系统架构设计

3.3.1 总体设计

图 3-2 系统总体架构设计图

3.3.2 "4+1"视图

Krutzen于1995年提出了软件体系结构的“4+1”视图模型，如图3-3所示。观点模型基于5个不同的视图：逻辑视图、过程视图、物理视图、开发视图和场景视图。每个视图为系统中的一个特定群体描述，专注于系统的某个侧面，包含。

5个视图描述了软件系统结构的全部内容[11]。 图 3-3 "4+1"视图

图 3-4 逻辑视图

图 3-5 物理视图

图 3-6 过程视图

图 3-7 开发视图

3-1为系统场景视图(用例图)。

3.3.3 模块划分

图 3-8 模块划分图

3.4 数据库设计

3.4.1 概述

基于系统功能要求并结合数据特征,本文设计了由样本表、样本组表、模型表、检测结果表及用于跟踪模型训练样本和推理过程的模型训练样本关联表和模型推理样本关联表等六个表组成的数据库。数据库设计遵从原则,综合考虑数据库的实用性、有效性,以及查询的便利性和快捷性。

3.4.2 E-R图设计

图 3-9 E-R图

3.4.3 核心表结构

1. 样本组表 (sample\_group)如图 3-10所示。 图 3-10 样本组表

1 sample\_group(

2 id INT PRIMARY KEY AUTO\_INCREMENT, // 样本组ID,主键

3 name VARCHAR(100) NOT NULL // 样本组名称

4 ) 2. 样本表 (sample)如图 3-11所示。 图 3-11 样本表

1 sample(

2 id INT PRIMARY KEY AUTO\_INCREMENT, // 样本ID,主键

3 name VARCHAR(100) NOT NULL, // 样本名称

4 group\_id INT NOT NULL, // 所属样本组ID,外键

7 ) 3. 模型表 (model)如图 3-12所示。 图 3-12 模型表

1 model(

2 id INT PRIMARY KEY AUTO\_INCREMENT, // 模型ID,主键

3 name VARCHAR(100) NOT NULL, // 模型名称

4 path VARCHAR(255) NOT NULL, // 模型文件路径

5 input\_h INT NOT NULL, // 输入高度

6 input\_w INT NOT NULL, // 输入宽度

7 end\_acc FLOAT NOT NULL, // 结束精度

8 layers INT NOT NULL, // 使用的层数

9 patchsize INT NOT NULL, // 补丁大小

10 embed\_dimension INT NOT NULL, // 嵌入维度

11 status ENUM('NEW','TRAINING','READY','INFERRING'), // 模型状态

12 ) 4. 检测结果表 (detection\_result)如图 3-13所示。 图 3-13 检测结果表

1 detection\_result(

2 id INT PRIMARY KEY AUTO\_INCREMENT, // 结果ID,主键

3 name VARCHAR(100) NOT NULL, // 结果名称

4 sample\_id INT NOT NULL, // 检测的样本ID,外键

5 model\_id INT NOT NULL, // 检测的模型ID,外键

6 score FLOAT NOT NULL, // 异常分数

7 path VARCHAR(255) NOT NULL, // 检测结果文件路径

8 FOREIGN KEY (sample\_id) REFERENCES sample(id),

9 FOREIGN KEY (model\_id) REFERENCES model(id) 10 )

5. 模型训练样本表 (model\_trained\_sample)如图 3-14所示。 图 3-14 模型训练样本表

1 model\_training\_sample(

2 id INT PRIMARY KEY AUTO\_INCREMENT, // 关联ID,主键

3 model\_id INT NOT NULL, // 模型ID,外键

6 FOREIGN KEY (sample\_id) REFERENCES sample(sample\_id),

7 UNIQUE KEY (model\_id, sample\_id) //

确保一个样本在一个模型中只被训练一次 8 )

6. 模型推理样本表 (model\_inferred\_sample)如图 3-15所示。 图 3-15 模型推理样本表

1 model\_inference\_sample(

2 id INT PRIMARY KEY AUTO\_INCREMENT, // 关联ID,主键

3 model\_id INT NOT NULL, // 模型ID,外键

4 sample\_id INT NOT NULL, // 样本ID,外键

5 score FLOAT NOT NULL, // 异常分数

6 FOREIGN KEY (model\_id) REFERENCES model(model\_id),

7 FOREIGN KEY (sample\_id) REFERENCES sample(sample\_id),

8 UNIQUE KEY (model\_id, sample\_id) //

确保一个样本在一个模型中只被记录一次 9 )

3.4.4 数据关系与完整性

本关联表均引用 sample\_id和 model\_id,关联到模型表和样本表

通过这种简洁而功能完备的数据库结构,系统能够有效支持样本管理、模型训练和缺陷检测结果存储,为用户提供完整的数据支持。

3.5 本章小结

本章对缺陷检出系统进行了详细的论述,从实际工业质检的工作场景出发,对系统的整体分析设计做了详细的阐述,确认系统作为工业质检的辅助工具的目标,再通过用例对系统的功能需求和非功能需求做了详细分析,然后通过“4+1”视图模型从多个视角对系统的组成结构进行描述,将系统划分为四个子模块十一个功能模块,最后设计数据库结构作为系统的数据支撑。

第四章 系统实现

4.1 开发环境

客户端基于python3.11+PySide6.0的Windows1164位操作系统,部署在Ubuntu22.04的服务器上,采用Git进行版本控制。

4.2 核心模块实现

4.2.1 项目管理模块

(1)项目信息模块

(2)历史项目模块 图 4-1 开始界面

(3)计时器模块

计时器是项目中辅助的组件,用于追踪项目的执行时间,帮助用户记录和评估工作时间。该模块由项目中的FloatingTimer类实现,覆盖整个应用程序生命周期。

图 4-2 项目结构图

4.2.2 样本管理模块

样本管理模块是缺陷检测系统的基础组件,负责样本的组织、存储和预处理,为后续的模型训练和缺陷检测提供高质量的数据支持。

(1)样本组管理模块

样本组管理模块作为样本管理的基础模块,用于对样本集合以及样本组的状态进行组织管理控制。图4-3样本组管理控制图4-3用户创建、导入、删除样本组以及上传样本组到服务器的管理控制界面.导入、删除样本组管理控制界面如图4-4所示，可以查询出目前系统所创建的样本组的名称以及样本组的状态。新建或导入样本组后，会将样本组的入元数据记录下来，在下次打开项目工程时加载。样本组所在位置由样本列表栏进行显示，而且样本列表支持从目录中拖动批量导入样本，支持添加单个或多个样本。双击导入样本后的列表项即可切换到对应样本的查看编辑界面，点击选择按钮可以对样本进行多选或删除。当用户新建、切换或删除样本组后，系统会刷新样本列表栏并与视图刷新，修改样本组的状态。样本组的状态控制着样本列表操作按钮的显示与否，只有存在样本组时才具有导入、删除样本的操作权限。样本编辑完成后，用户可将样本组上传到服务器中，以备后续使用。系统会跟进上传进度，以进行动态反馈。此外，用户在开始模型训练之前，系统会自动判断样本组是否上传到服务器，以避免用户忘记上传。只有上传完毕后，才能进行下一步操作。

图 4-3 样本界面 图 4-4 样本组列表 (2)样本处理模块

样本处理模块是样本管理的主要部分,它负责单个或者多个样本的预处理工作,如裁剪、去除背景和数据增强。裁剪和去除背景的目的相同,都是去除不必要图像部分,重点表现被检目标,以减少背景影响,提高检出率;数据增强可以增加样本的多样性,提高模型的泛化能力。

背景排除缩小域功能基于智能分割算法，保留图像主要部分，即非背景部分，保证后续的数据增强、模型训练等操作在图像的正确区域进行，适用于背景相对明显的样本。相较于手动裁剪，其可自动化处理样本批量，效率更高。使用OpenCV库来分别进行图像阈值、轮廓、滤波操作，基于形态学操作来进行轮廓筛选，去除噪点及细小碎片。具体操作上，将图像灰度化，二值化处理以分离前景、背景，然后进行轮廓检测、筛选，保留主要目标区域，最后自动运算目标区域最小化外接矩形，完成图像裁剪；同时，为保证目标完整，防止过裁剪，会自动增加裁剪框一定边距。

数据增强可对图片进行多种变化,可以增加训练集样本数量,提高模型的泛化性能,适合于样本数量较少的情况;数据增强可以批量化操作,使用OpenCV库对选取的样本进行几何和外表的变换,包括旋转、翻转、亮度、颜色等,支持选择特定方式,也可以选择随机变换,变换后的图片将自动生成,刷新后显示在样本列表框中;由于无监督学习模型的训练集要求是阳性样本,数据增强只会少量修改样本。

图 4-5 裁剪画面

4.2.3 模型训练模块

模型训练模块是应用基于无监督深度学习的缺陷检测算法的核心组件,负责管理和训练缺陷检测模型。 (1)模型管理模块

图 4-6 模型界面 图 4-7 参数设置 (2)训练模块

训练模块实现模型训练的过程,具体包括训练、训练可视化、模型状态控制等部分,训练前自动判断样本组和模型组是否为空,训练的模型是否是未训练的模型。开始训练时,进入等待状态并显示加载界面,直到得到第一轮的训练结果。此时开始显示训练的实时进度,训练曲线如图4-8所示。图中展示了动态概率训练曲线,训练的实时轮次、训练用时、训练准确性等。训练曲线可视化程序在Qt图表框架基础上编写,定时从后台提取数据并更新数据,使用QChart和QLineSeries动态在坐标轴上绘制,也可进行全局和放大图的可视化交换,观察整体和细节的变化。此外,训练过程中如果需要立即使用模型,可以按下停止训练键结束训练。在模型训练结束达到结束精度或者用户提前中止模型训练以后，训练结束，自动恢复空闲状态。同样的，在用户开始缺陷检测流程前会自动检测模型是否训练结束并给出提醒，在训练结束后才能继续下一步。

图 4-8 训练进度

4.2.4 缺陷检测模块

(1)检测管理模块

图 4-9 检测界面 (2)检测模块

检测结果图由原图和异常检测热图共同组成,对比鲜明,能够直观地显示出异常的位置、大小等。热图以颜色作为异常概率,颜色越深表示异常的概率越大,反之亦然。当点选中央显示栏后,可以将检测结果图切换到原图。同时,在检测结果图下方显示了当前检测的具体信息,包括样本名称、检测时间、得分、检测阈值判定状态等。

图 4-10 检测过程 (3)AI辅助分析模块

图 4-11 AI对话 (4)检测报告模块

检测报告模块作为缺陷检测模块的结果综述，对检测的结果进行统计分析，并形成可视化结果报告。检测结束后，可以调整缺陷阈值、聚类参数及块大小，然后生成结果报告，其内容包括检测样本组基本信息、缺陷位置分析、区域特征统计以及缺陷类型分析。如图4-12所示，在解析完成后，可以点击报告生成界面，直接展示可视化结果、统计数据以及详细信息，也可以点击下载按钮，将报告以PDF的格式存储在本地。

图 4-12 可视化分析标签页,又为检测报告界面,包含缺陷位置分布图

使用OpenCV库实现灰度图提取缺陷位置和缺陷特征的工作，将检测出来的检测结果热图首先进行灰度化处理和二值化处理，利用OpenCV提取灰度化处理后的检测结果热图中的高热区域(缺陷位置)。然后分别处理每个缺陷区域的轮廓特征，提取缺陷区域的包围框、中心点和轮廓面积作为几何特征，用于统计缺陷位置；接着提取有效区域的灰度均值、方差和最大值，采用Sobel算子计算其梯度均值，用于缺陷纹理分析，至此，得到了缺陷位置和缺陷纹理特征特征向量。

然后对选取的原图进行区域特征统计。同理利用OpenCV库将原图统计为灰度图，并通过热图得出原图中异常区域。计算Sobelel边缘，得到图像梯度幅度，提高结构特征。将图片根据指定块大小划分为均衡的网格，以块为统计对象同时进行统计。以像素的百分之三十异常为基准判断网格是否异常，与此同时计算每个网格的灰度均值、灰度方差和边缘密度，结合统计结果，同时结合正常区域和异常区域得到区域亮度分布直方图、区域纹理复杂度分布直方图和区域边缘密度分布直方图，如图4-13所示。

图 4-13 区域特征分布图

亮度分布直方图可用于识别过亮、过暗、局部高反差等亮度异常造成的缺陷;纹理复杂度分布直方图可用于识别纹理断裂、纹理杂乱、纹理缺失等纹理异常造成的缺陷;边缘密度分布直方图可用于识别裂纹、划痕、轮廓缺失等边缘异常造成的缺陷。绿色和红色分别代表正常和异常的分布情况。

然后对检测样本的缺陷类型进行统计与分析；结合上一小节中所提取的位置信息和纹理特征信息，以缺陷的亮度阈值以及归一化面积确定缺陷的等级，从轻微污染到严重损坏，缺陷程度逐级增加，制作饼图，如图4-14所示。针对每一个缺陷等级又划分了具体八种缺陷形态，分别为划痕、裂缝、缺口等。

图 4-14 统计数据标签页。包含缺陷类型分布图

部分: 1. 基本信息,描述了检测样本组的基本信息。 2. 缺陷位置分析,展示了缺陷的空间分布特点以及聚类结果。

3. 区域特征统计,对比了正常区域与异常区域在亮度、纹理复杂度和边缘密度三个维度上的差异。

4. 缺陷类型统计,分析了所有缺陷区域,统计了每类缺陷的细分缺陷数量以及样本数量。

5.结论分析,根据统计结果给出可能存在的缺陷原因推断,并提出一些改进建议。详细信息页面以树状结构呈现，从基本信息到聚类分析到缺陷类型统计，每个层级都可以展开查看细节数据，如图4-15所示。

图 4-15 详细信息标签页

4.3 接口与其他

4.3.1 API接口

缺陷检测系统通过 HTTP接口与远程服务器通信,实现模型训练、推理和数 据管理等功能,具体 API如下:

(1) AI相关 API如表 4-1所示。 表 4-1 API接口说明 API方法 功能描述 主要参数 返回结果

anomaly\_gpt\_infer 使用大语言模型分析 缺陷图像 img\_list, question, nor-

mal\_img\_list, history 图像分析结果 文本列表 (2)模型相关 API如表 4-2所示。

表 4-2 模型相关 API接口说明 API方法 功能描述 主要参数 返回结果

add\_model 创建新模型 name, input\_h, input\_w,

end\_acc, layers, patchsize, embed\_dimension 模型 ID

finish\_model 手动终止训练 model\_id 操作结果

(3)样本组相关 API如表 4-3所示。 表 4-3 样本组相关 API接口说明

API方法 功能描述 主要参数 返回结果

据 get\_sample\_list 获取组内样本列表 group\_id 样本名称列表

add\_group 创建新样本组 group\_name 组 ID

get\_group\_id 根据名称获取组 ID group\_name 组 ID

4.3.2 其他

(1)异步处理进度显示与加载动画

图 4-16 加载进度 LoadingAnimation类基于 GIF动图实现,支持文本提示和动画显示,详见图

大模型推理这种不能判断剩余时间的操作,否则会无限提示,应该将这样的操作放到一个异步线程中执行,在主线程更新加载动画,给用户视觉反馈,避免用户产生卡顿的感觉。 4-17。

(2)事件过滤器处理点击事件 图 4-17 加载动画

系统采用QT的事件过滤器(Event Filter)组件对点击事件进行过滤，例如AI分析模块中的交互式图像查看，当点击当前图像时，便可无缝切换原图和结果图，便于图像的直观对比。事件过滤器可以对特定组件的事件流进行拦截，实现组件事件处理的自定义，这种非侵入式的设计，将处理逻辑和ui组件剥离开来，有助于增加新的组件，扩展性增强。

(3)点击空白处完成编辑

(4)流程引导

流程引导是系统的辅助功能之一，介绍了系统的各个功能模块和操作流程，以帮助非专业用户了解缺陷检测的工作流程，达到快速上手的目的。在用户创建项目后，打开项目就可以进入流程引导界面，如图4-18所示。引导界面采用分流式的引导设计，展示了系统的两种主要使用方向：采集样本，样本预处理，创建新模型并训练，训练完成后进行检测。重新使用训练好的模型进行检测。使用AI辅助分析，然后生成报告。

(5)人机交互启发式原则在系统中的体现 图 4-18 流程引导界面

系统多处体现人机交互启发式原则：帮助与文档：用户打开一个项目后，会看到流程引导。依赖识别而不是记忆：每个按钮都关联了一个图标，且图标直观可识别。一致性与标准化：采用统一的UI设计，确保不同界面在布局和样式风格上保持一致。反馈系统状态：系统状态在加载进度和加载动画中予以反馈。与真实世界吻合：系统所使用的缺陷检测流程模拟现实世界中从样本采集、模型训练、缺陷检测、AI辅助分析、报告生成流程中采样，如同人眼在观看样本、寻找缺陷，高等级质检人员辅助分析及人手书写报告。为防止错误发生，在检测时将界面上的组件禁用以限制用户行为。帮助用户识别、诊断和恢复错误：系统将错误以弹窗的形式予以反馈，如在新建项目时，为避免出现项目名称非法时予以提示。最后，系统使用的灵活性和高效性体现在支持拖拽文件打开项目、新建样本组或模型后无需确认就可直接按回车键确认，以及页面切换时可以自动提醒是否上传样本组或训练模型。

4.4 部署

4.4.1 服务器端部署

4.4.2 客户端部署

Pyinstaller打包为可运行可执行的Windows1164位系统可运行程序，该程序包括源程序、ui程序，以及Py-Side6等库，可部署在Windows1164位操作系统上。需要注意的是，在使用Pyinstaller进行打包时，需显式引入opencv库，引入ui文件夹。另外，对可运行可执行的exe文件进行运行时，界面图片无法正常显示，造成该问题出现的原因是exe运行目录和ui文件夹的相对位置发生变化，本系统对该问题解决方案是在运行程序时设置工作目录为应用程序目录。

4.5 本章小结

本章对缺陷检测系统的实现过程进行了详细介绍,从开发环境到十一个子模块的实现以及最后部署到落地的过程都做了系统的阐述。另外,本章还介绍了系统里一些处理,如异步处理、事件过滤器、截剪、流程引导等,遵循了人机交互领域的八大启发式原则,使得系统成为一个真正友好、高效、易用的系统。

第五章 实验

5.1 数据集

图5-1MVTec异常检测数据集部分五类纹理和十个人物类别的图像。每个类别中各包含一个无异常的示例和异常的示例。顶部一行为输入图像，底部一行为特写视图。

5.2 评估指标

5.3 实验设计

5.4 结果分析

5.4.1 SimpleNet异常检测

ACC\_LEVEL I-AUROC P-AUROC TIME(ms) LOW 90.35% 98.38% 16.55

MEDIUM 98.40% 98.83% 21.91 HIGH 99.48% 98.07% 102.77

SPEED\_LEVEL I-AUROC P-AUROC TIME(ms)

HIGH 96.85% 98.55% 21.79 MEDIUM 98.40% 98.83% 21.91

LOW 99.16% 98.78% 23.12

5.4.2 AnomalyGPT辅助分析

如表5-3所示，在bottle测试集三类样本中，AnomalyGPT辅助分析异常是否存在准确率都达到了100%，效果优秀。但在缺陷位置的准确率上表现一般，大缺陷位置处准确率只有70%，说明AnomalyGPT在辅助分析时，容易被大范围的异常区域所干扰，不能准确定位缺陷位置。在实验过程中发现，出现错误最多的场景是当检测到瓶底处四周存在异常时，AnomalyGPT会认为缺陷位于瓶底中心处，有可能是AnomalyGPT需要改进的地方。

DEFECT\_CLASS EXISTANCE\_ACC POSITION\_ACC GOOD 100% -

SMALL\_DEFECT 100% 86.3% BIG\_DEFECT 100% 70%

5.5 本章小结

本章对缺陷检测系统进行了实验。首先，介绍了用于实验的MVTec AD数据集和用于评估的指标。然后，设计了实验，对SimpleNet异常检测模型在不同精度和速度选取下的结果，AnomalyGPT辅助分析模型在异常是否存在和定位缺陷位置上的结果进行了验证。最后，对实验结果进行了分析，证明了精度和速度选取的合理性。

第六章 总结与展望

6.1 工作总结

本文针对工业缺陷检测中标注数据有限、缺陷种类繁多、现有系统操作复杂、配置要求高等问题，以无监督深度学习算法为基础，以源代码级别的无监督深度学习算法SimpleNet为核心，结合AnomalyGPT大语言视觉模型的辅助分析能力，在样本管理阶段采用了动态样本组管理，引入数据增强等预处理算法，解决了适合工业检测的样本质量不高、样本多样性低等问题；在模型训练阶段，为了降低非专业使用者的操作门槛，支持手动或自动参数配置，设计了参数映射功能，将深度学习中复杂的参数逻辑以“精－快－缺陷大小”的选项进行呈现；在缺陷检测阶段，为了支持从缺陷定位到工艺优化的闭环缺陷检测，结合SimpleNet的异常检测能力以及AnomalyGPT的自然语言交互能力，对检测结果做了可视化呈现，以结果热图、检测结果AI辅助、缺陷检测报告等形式提供多方位服务。

实验证明，基于该系统实现的检测结果是较为理想的，且在数据集MVTec AD上，采用默认参数设置情况下，SimpleNet检测胶囊类样本的平均I-AUROC达到98.4%，每张图像平均检测时长在22ms左右，AnomalyGPT对异常存在性的判定准确率达到100%；且系统基于Qt实现跨平台部署，所有功能实现模块化，采用异步处理模式保证了系统的响应稳定性，同时基于人机交互启发式设计原则对系统交互界面设计进行了优化，最后实现了一款具有高效性、易用性以及可解释性的工业质检系统，为智能化工业制造中的故障检测提供了高效且实用的解决方案。

6.2 未来展望

然而，缺陷检测系统在检测精度和实用性方面表现卓越的同时，也存在着诸多不足之处仍亟待解决，未来检测系统在功能完善、系统框架改造、检测算法优化、系统架构升级以及工业落地等方面具有广阔的研究空间和学习探索价值，包括但不限于：一是可探索多算法融合框架，实现不同场景下用户对工业产品检测算法的动态选择；二是可考虑结合有监督和无监督深度学习的优势，以少量标注数据提升缺陷检测系统的性能；三时可尝试系统输入的扩展，支持多模态数据的融合和分析，如红外、超声波等传感器数据，并结合生产需求增设的实时检测模式；四是可考虑应用分布式数据库优化和云端协同计算，提升海量数据的处理能力，同时可构建权限管理和数据共享机制，提升企业级应用协作能力。最后,与工厂合作,对细化的制造领域进行专项优化,导出相关领域的数值对大模型进行小范围的微调,完善AI辅助分析和缺陷检测报告,优化系统对生产工艺优化的支撑,使得达到真正意义上的智能化制造。

参考文献

[1] 罗东亮,蔡雨萱,杨子豪,等.工业缺陷检测深度学习方法综述[J].中国科学:信息科学, 2022, 52(06): 1002-1039.

[2] 韩慧.基于深度学习的工业缺陷检测方法研究[D].重庆邮电大学, 2019.

[3] 赵君爱.基于图像处理的工件表面缺陷检测理论与方法研究[D].东南大学,2016.

[4] 张涛,刘玉婷,杨亚宁,等.基于机器视觉的表面缺陷检测研究综述[J].科学技术与工程, 2020, 20(35): 14366-14376.

[5] 汤勃,孔建益,伍世虔.机器视觉表面缺陷检测综述[J].中国图象图形学报,2017, 22(12): 1640-1663.

[6] 王文鹏,秦寅畅,师文轩.工业缺陷检测无监督深度学习方法综述[J].计算机应用, 1-16.

[7] 伍麟,郝鸿宇,宋友.基于计算机视觉的工业金属表面缺陷检测综述[J/OL].自动化学报, 2024, 50(07): 1261-1283. DOI: 10.16383/j.aas.c230039.

[8] 魏利胜,李明.基于深度学习的工业产品表面缺陷检测综述[J].绥化学院学报, 2024, 44(06): 151-156.

[9] LIU Z, ZHOU Y, XU Y, et al. SimpleNet: A Simple Network for Image AnomalyDetection and Localization[EB/OL]. 2023. https://arxiv.org/abs/2303.15140.arXiv: 2303.15140 [cs.CV].

[10] GU Z, ZHU B, ZHU G, et al. AnomalyGPT: Detecting Industrial Anomalies

Using Large Vision-Language Models[EB/OL]. 2023. https://arxiv.org/abs/2308.15366. arXiv: 2308.15366 [cs.CV].

[11] 陆远蓉.基于 UML的"4+1"视图软件体系结构描述研究[J].现代计算机(专业版), 2011(04): 27-30.

致 谢

感谢葛季栋教授、邬向前教授以及马丁老师对毕业设计和论文的指导。感谢张研钦师兄、史云浩师兄在日常生活和毕业设计的帮助。

光阴似箭,日月如梭。四年时间,转瞬即逝。回首往昔,感慨万千。

儿时岁悠悠,人生不胜愁。不求人长乐,但愿心长留。