

山东大学软件学院 2023 年度项目实训	文档编号	0008	版本	Ver 1.5
	项目名称	工业品缺陷检测系统		
	项目来源	项目实训		

工业品缺陷检测系统 项目结题书

(内部资料 请勿外传)

编 写:	李岩霖	日 期:	2023/7/13
检 查:	李岩霖	日 期:	2023/7/13
审 核:	李岩霖	日 期:	2023/7/13
批 准:	李岩霖	日 期:	2023/7/14

山东大学软件学院

版权所有 不得复制

文档变更记录

序号	变更（+/-）说明	作者	版本号	日期	批准
1	(+)增加引言部分	李岩霖	Ver 1.0	2023/7/12	√
2	(+)增加整体概述部分	李岩霖	Ver 1.1	2023/7/12	√
3	(+)增加运行环境部分	李岩霖	Ver 1.2	2023/7/13	√
4	(+)修改整体概述部分	李岩霖	Ver 1.3	2023/7/13	√
5	(+)增加项目成果展示部分	李岩霖	Ver 1.4	2023/7/14	√
6	(+)增加项目开发流程部分	李岩霖	Ver 1.4	2023/7/14	√
7	(+)增加项目总结部分	李岩霖	Ver 1.5	2023/7/14	√

目 录

1. 引言	4
1.1 编写目的和范围	4
1.2 术语表	5
1.3 参考资料	5
1.4 人员分工	5
2. 整体概述	6
2.1 产品背景	6
2.2 系统目标	7
2.3 用户分类	7
2.4 项目任务	7
2.5 产品详情	8
3. 运行环境	10
3.1 硬件环境	10
3.2 部署环境	10
4. 项目开发流程	11
4.1 开发方式	11
4.2 开发历程	12
4.2.1 划分训练集和测试集	12
4.2.2 数据增强	13
4.2.3 数据清洗	16
4.2.4 深度学习框架选择	16
4.2.5 模型训练	18
4.2.6 模型量化	19
4.2.7 后端搭建	20
4.2.8 Web 前端搭建	23
4.2.9 微信小程序前端搭建	27
4.2.10 软件测试	29
4.3 软件项目管理	35
4.3.1 Github	35
5. 项目成果展示	36
5.1 Web 系统	36
5.2 微信小程序系统	38
6. 项目总结	39
6.1 感悟	39
6.2 项目展望	40
6.2.1 二分类，三分类，七分类模型的选择与思考	40
6.2.2 生产线部署设想	41

1. 引言

1.1 编写目的和范围

制造业的全面智能化发展对工业产品的质量检测提出了新的要求。表面缺陷检测是工业产品质量检测的关键部分。工业品检测是在制造过程中使用各种技术和方法来验证产品质量的过程。它的背景可以追溯到工业革命时期，当时随着生产规模的扩大，制造商们开始面临更多的质量问题和挑战。在现代制造业中，工业品检测发挥着至关重要的作用，对于确保产品质量、提高客户满意度以及保护品牌声誉都起着重要作用。

本项目针对若干张组装后的工业品图片，通过 AI 模型，识别工业品是否有缺陷（缺少螺丝），基于训练好的深度学习模型，建立 web 服务。提供上传图片，同时识别图片中物品是否有缺陷的功能。

本项目计划书编写的目的是说明系统的设计考虑，包括程序描述、输入/输出、算法和流程逻辑等，为软件编程和系统维护提供便利和基础。

本项目计划书主要面向本项目开发项目组成员，让项目组成员充分了解到本系统开发项目的需求、功能模块、业务逻辑等，从而完整、有效地开发以及实现系统全部的功能。

本项目计划书的预期读者为：

(1)项目经理：项目经理可以根据该文档了解预期产品的功能，并据此进行系统设计、项目管理，其中包括对系统进行配置管理和数据库文件更新管理，确保迭代版本的兼容性。

(2)系统分析师：系统分析师对系统背景进行调查，对需求进行分析，给出详细的需求文档。

(3)文档研读分析师：文档研读分析师，通过阅读大量的有关工业品检测的论文，搜索有关工业品检测的常用方法，为算法工程师提供思路，辅助算法工程师设计算法，训练模型。

(4)前端开发工程师：前端开发工程师按照系统设计师的设计开发文档进行系统前端的实现，并编写用户使用手册。

(5)后端开发工程师：后端开发工程师按照系统设计师的设计开发文档进行系统的实现，并编写用户使用手册。

(6)算法工程师：算法工程师根据文档研读分析师提供的思路，进行算法的设计和模型

的训练，与文档研读分析师一起进行模型推理加速工作。

(7)测试工程师：测试工程师根据开发工程师的用户使用手册以及本说明编写测试用例，并对软件产品进行功能性测试和非功能性测试。

(8)系统用户：系统用户可以根据本使用说明书探索并使用本系统的功能。

1.2 术语表

序号	术语或缩略语	说明性定义
1	PM	Project Manager，项目经理
2	SA	System Analyst，系统分析师
3	DRA	Document Research Analyst，文档研读分析师
4	FPD	Front-end Program Designer，前端开发工程师
5	BPD	Back-end Program Designer，后端开发工程师
6	AE	Algorithm Engineer，算法工程师
7	TE	Testing Engineer，测试工程师

1.3 参考资料

资料名称	作者	文件编号、版本
《2023 年-2020 级项目实训实施计划 - 学生版本》	戴鸿君	V3
《2023-暑期项目实训申报通知-Intel-2》	郑艳飞，张建宇	2023 版

1.4 人员分工

职位	人员	学号
PM	李岩霖	202000300051
SA	李岩霖	202000300051

DRA	玄小龙	202000300198
	周星驰	202000300036
FPD	王秀宇	202000300246
	张新钧	202000300019
	王修智	202000180102
	方正	202000300131
BPD	李岩霖	202000300051
AE	李岩霖	202000300051
TE	周星驰	202000300036
	玄小龙	202000300198

2. 整体概述

2.1 产品背景

随着工业生产规模的不断扩大和技术进步，工业品的质量要求越来越高。然而，在工业生产过程中，由于人为操作、设备故障或原材料问题等原因，工业品往往会出现一些缺陷。这些缺陷包括但不限于裂痕、变形、颜色不符合标准、表面不光滑等。如果这些工业品没有及时发现并处理，可能会导致产品质量下降、客户投诉增加甚至造成安全隐患。传统的缺陷检测方法通常依赖于人工目视检查，这种方法费时费力且容易出错，也无法满足高效率、高精度的缺陷检测需求。因此，利用计算机视觉和图像处理技术来开展工业品缺陷检测具有重要意义。通过建立智能检测系统，可以实现对工业品进行自动化、快速、准确的检测，大大提高了生产效率和产品质量。基于深度学习和机器学习算法的工业品缺陷检测系统在近年来取得了显著的进展。这些算法可以通过训练模型来学习工业品的正常状态和各种缺陷类型，从而能够自动地对工业品进行分类和检测。通过结合高分辨率图像采集设备、先进的图像处理算法和高性能计算平台，可以实现对工业品缺陷的高效、准确检测，大大提高了生产

线的质量控制水平。因此，开展工业品缺陷检测项目具有重要的现实意义和市场需求。通过引入先进的技术手段，可以有效解决工业品缺陷问题，提高产品质量和竞争力，满足市场对高质量工业品的需求。综合使用计算机技术、网络技术，使用 Vue，WTML，JavaScript，Flask，Tensorflow 等技术栈构建一个基于 Web 的 AI 检测服务。

2.2 系统目标

工业品检测系统是针对具体的一个工业品组装生产线设计的，主要实现以下目标：

- (1) 操作简单方便、界面简洁美观。
- (2) 网站整体结构和操作流程合理流畅，实现人性化设计。
- (3) 可以在平衡精度和速度的前提下，尽可能识别正确大部分图片。
- (4) 在识别出合格与不合格情况下，给出不合格的原因（缺少一螺丝，缺少对角两螺丝，缺少临边两螺丝，缺少三螺丝，缺少四螺丝），以及由于拍摄角度而造成的零件不完整（缺角）
- (5) 系统最大限度地实现易操作性、易维护性。
- (6) 系统运行稳定、安全可靠。

项目对时间进度的要求苛刻，建议在系统架构设计时尽量熟悉系统开发过程的目的和特点，使得架构具有一定的适应性和灵活性，可以先完成核心架构和基本功能，满足进度要求，后期再进行必要的改进和完善。

2.3 用户分类

本系统的使用用户如下：工业品缺陷检测的使用者

2.4 项目任务

需求分析阶段：根据系统应具备的功能，结合工业品检测的流程，进行需求分析。

概要设计阶段：将需求转化为计算机模型，并且对要实现的功能进行分配。

详细设计阶段：将提供各模块、任务、功能点的详细规划。

代码实现阶段：编码实现系统，完成程序的编写。

测试阶段：通过测试软件，进行客户端与本地文件之间数据的传送的测试；客户端联网服务的测试。

2.5 产品详情

产品名称：

工业品缺陷检测系统

应交付文档：

《工业品缺陷检测系统项目计划书》

《工业品缺陷检测系统概要设计方案》

《工业品缺陷检测系统需求分析说明书》

《工业品缺陷检测系统技术实现方案》

《工业品缺陷检测系统测试文档》

《工业品缺陷检测系统用户使用说明书》

《工业品缺陷检测系统项目结题书》

应交付代码：

Web 前端代码；微信小程序前端代码；后端代码以及模型算法

实际交付物：

相关文档 7 份：

《工业品缺陷检测系统项目计划书》

《工业品缺陷检测系统概要设计方案》

《工业品缺陷检测系统需求分析说明书》

《工业品缺陷检测系统技术实现方案》

《工业品缺陷检测系统测试文档》

《工业品缺陷检测系统用户使用说明书》

《工业品缺陷检测系统项目结题书》

周汇报 PPT4 份：

《20230620 工业品检测项目设想》

《20230627 工业品检测项目进度汇报》

《20230704 工业品检测项目进度汇报》

《20230711 工业品检测项目进度汇报》

工作日志 144 份：

《李岩霖_20230619》

《李岩霖_20230620》

《李岩霖_20230621》

《李岩霖_20230625》

《李岩霖_20230626》

《李岩霖_20230627》

《李岩霖_20230628》

《李岩霖_20230629》

《李岩霖_20230630》

《李岩霖_20230703》

《李岩霖_20230704》

《李岩霖_20230705》

《李岩霖_20230706》

《李岩霖_20230707》

《李岩霖_20230710》

《李岩霖_20230711》

《李岩霖_20230712》

《李岩霖_20230713》

产品演示视频 2 份：

《wechatproject_demo》微信小程序演示视频

《webproject_demo》Web 端演示视频

代码包 3 份

《vue-font》Web 前端代码

《WechatProject》微信小程序前端代码

《defec_product_backEnd》flask 后端以及模型训练代码

3. 运行环境

3.1 硬件环境

PC 机

操作系统：Windows7 及其以上版本

浏览器：兼容大部分主流浏览器（IE 浏览器除外）

手机微信

后端模型的部署所需要的环境和第三方库版本见 requirements.txt 文件

3.2 部署环境

flask 后端的部署对三方库的版本有限制，库的不同版本可能会导致出现一些不可预知的问题，因此建议用户按照下面给出的 requirements.txt 文件安装指定的第三方库版本。

requirements.txt 内容如下：

```
Flask==2.3.2
Flask_Cors==4.0.0
Flask_Mail==0.9.1
google_pasta==0.2.0
imageio==2.31.1
imgaug==0.4.0
keras==2.11.0
keras_flops==0.1.2
neural_compressor==2.2
numpy==1.24.4
opencv_contrib_python==4.5.5.64
opencv_python==4.8.0.74
pywinauto==0.6.8
selenium==4.10.0
tensorflow==2.11.1
tensorflow_intel==2.11.1
```

```
torch==2.0.1
```

```
torchvision==0.15.2
```

安装第三方库时，只需要在命令行执行如下命令：

```
pip install -r requirements.txt
```

4. 项目开发流程

4.1 开发方式

结合项目的时间节点，以及企业导师的建议，对于本项目，我们采用敏捷开发的方式，将模型算法部分空出来，后续迭代。

敏捷开发是一种迭代、增量式的软件开发方法，旨在提高开发团队的灵活性、适应性和协作能力。它强调通过快速反馈和紧密合作来满足客户需求，以便在不断变化的环境中及时交付高质量的软件。

敏捷开发的核心原则包括：

个体和互动优先于流程和工具：重视团队成员之间的沟通 and 协作，通过面对面的交流来解决问题。

可工作的软件优先于详尽的文档：注重开发可用的软件产品，并通过持续集成和测试来保证其质量。

客户合作优先于合同谈判：与客户紧密合作，及时获取反馈并进行调整。

响应变化优先于遵循计划：能够灵活应对需求变更和市场变化，持续地改进和调整软件。

敏捷开发使用迭代和增量的方式进行开发，每个迭代通常称为一个“冲刺”（Sprint），其长度通常为几周到一个月。每个冲刺开始前，团队确定要完成的任务，并在冲刺期间进行开发、测试和验收。每个冲刺结束时，团队会展示和交付可工作的软件功能，以便客户和利益相关者提供反馈，并根据需要进行调整。

敏捷开发还强调自组织团队和持续改进。团队成员在项目中具有更多的决策权和责任，并通过定期回顾会议（Retrospective）来识别问题和改进团队工作方式。

总之，敏捷开发是一种注重灵活性、高质量和客户合作的软件开发方法，能够适应快速变化的需求和市场环境。

本次项目的周具体工作和工作完成时间节点如下：

第一周后端算法工程师搭建后端，完成接口；前端工程师搭建微信小程序前端界面和接

口；给出微信小程序的模拟 demo，后端传递给前端的检测结果、置信度、推理时间都是一个固定的值。第二周后端算法工程师给出一个训练过的模型，即使模型精度和时间未达到要求，加入后端流程中；前端工程师给出 Web 前端的模拟 demo，这时两个项目 demo 中后端传递给前端的检测结果、置信度、推理时间都是模型推理得到的结果，即便精度和时间未达到要求。第三周后端算法工程师进行模型迭代和模型量化，前端工程师对界面进行优化，第四周进行软件测试和项目收尾。

4.2 开发历程

结合项目的时间节点，以及企业导师的建议，对于本项目，我们采用敏捷开发的方式，将模型算法部分空出来，后续迭代。

在四周的开发过程中，针对工业品缺陷检测这一核心任务，我们不断的提出新思路，尝试可能的新方案，增加项目的新功能，优化用户的使用体验。

4.2.1 划分训练集和测试集

为了保证数据增强之后每个类别中的数据都同时出现在训练集和测试集中，所以在数据增强之前，在每一类别中，将原始数据以某一比例（本项目训练集和测试集的比例为 8:2）划分成为训练集和测试集，然后再进行数据增强。数据增强前，每一类别下的数据数量如下：

分类		数量
anomaly	缺角零件	14
good	合格零件	16
one	不合格（缺一角螺丝零件）	13
diagonal_two	不合格（缺对角两螺丝零件）	20
neighbor_two	不合格（缺临边两螺丝零件）	18
three	不合格（缺三螺丝零件）	22
four	不合格（缺四螺丝零件）	14
总计		117

4.2.2 数据增强

数据增强是指通过对原始数据进行一系列变换和扩充，生成新的训练样本，以增加训练数据的多样性和数量。这种技术常用于机器学习和深度学习任务中，特别是在数据量有限的情况下。

数据增强的好处如下：

提高模型泛化能力：通过数据增强，可以生成更多样化的训练样本，使得模型能够学习到更广泛的特征和模式，从而提高其泛化能力，减少过拟合的风险。

缓解数据不平衡问题：在某些任务中，不同类别的样本数量可能存在不平衡。通过数据增强，可以生成更多少数类别的样本，从而平衡数据集，提高模型对少数类别的识别能力。

增加数据多样性：数据增强可以通过各种变换操作（如旋转、缩放、平移、翻转等）引入不同的视角和变化，使得模型能够更好地适应真实世界中的各种场景和变化。

提升模型鲁棒性：通过数据增强，可以模拟现实中的噪声、干扰和变形等情况，使得模型在面对这些干扰时能够更好地保持稳定性和准确性。

减少过拟合风险：数据增强可以有效扩充训练数据集，减少模型对于训练数据的过度依赖，从而降低过拟合的风险，提高模型的泛化能力。

总之，数据增强通过生成更多样化、更丰富的训练样本，可以提高模型的泛化能力、鲁棒性和准确性，同时缓解数据不平衡问题，是一种常用的提升机器学习和深度学习模型性能的技术手段。

而我们这一任务，原始的数据集中只有 117 张图片，数据集的数量过少，如果不经数据增强，直接输入训练的话，模型很容易过拟合，因此，我们考虑进行数据增强。

Python 中有很多有关数据增强的第三方库，以下是一些常用的数据增强的第三方库：

imgaug 是一个功能强大的图像增强库，支持多种图像增强技术，包括旋转、缩放、翻转、裁剪、亮度调整、对比度调整等。**albumentations** 是一个快速且灵活的图像增强库，支持多种图像增强技术，包括随机裁剪、旋转、缩放、翻转、模糊、颜色调整等。**Augmentor** 是一个用于图像增强的 Python 库，可以生成大量的增强图像样本。它支持多种增强操作，如旋转、缩放、翻转、平移、变形等。**Keras** 是一个流行的深度学习库，其中的 **ImageDataGenerator** 类提供了一系列图像增强方法，如旋转、缩放、翻转、剪切等。**Torchvision.transforms**：**Torchvision** 是 **PyTorch** 的计算机视觉库，其中的 **transforms** 模块提供了多种图像增强方法，如随机裁剪、旋转、缩放、翻转、亮度调整等。

这些库都提供了丰富的图像增强方法，可以帮助提升数据集的多样性和模型的鲁棒性。

其中，相对常用的是 `torchvision` 库和 `imgaug` 库，但考虑到本项目是基于 `Tensorflow2.0` 中的 `keras` 库进行搭建和训练模型，为了保证项目部署的一致性，所以本项目选用 `imgaug` 库进行数据增强。

`imgaug` 是一个用于图像增强的 `Python` 库，旨在帮助机器学习和计算机视觉任务中的数据预处理。它提供了丰富的图像增强技术和灵活的配置选项，使用户能够自定义和组合各种增强操作。

以下是 `imgaug` 的一些主要特点和功能：

多样的增强方法：`imgaug` 支持多种常见的图像增强方法，包括旋转、缩放、翻转、裁剪、亮度调整、对比度调整等。此外，它还提供了更高级的增强技术，如仿射变换、透视变换、弹性变形等，可以模拟真实世界中的图像变化。

灵活的参数设置：用户可以通过设置参数来控制增强效果。`imgaug` 提供了丰富的参数选项，如旋转角度、缩放比例、翻转概率、亮度范围等，使用户能够根据需求进行精细的调整。

增强序列：`imgaug` 允许用户通过链式调用来定义增强序列。用户可以按照自己的需求将多个增强操作组合成一个序列，并按顺序应用到图像上。这种方式可以方便地实现复杂的增强流程。

批量处理：`imgaug` 支持批量处理图像数据，可以高效地处理大规模的数据集。用户可以一次性对多个图像进行增强操作，提高处理速度和效率。

跨平台支持：`imgaug` 可以在多个主流深度学习框架中使用，如 `TensorFlow`、`Keras`、`PyTorch` 等。它提供了与这些框架兼容的接口，方便用户在不同环境下进行图像增强操作。

文档和示例代码：`imgaug` 提供了详细的文档和示例代码，帮助用户快速上手和理解库的使用。文档中包含了丰富的示例和说明，涵盖了各种常见的图像增强场景和用法。

在数据增强方法的选择中，我们最开始尝试了水平翻转，垂直翻转，旋转 90° ，旋转 270° ，在 -180° 到 180° 之间旋转随机角度，加入散粒噪声，加入高斯噪声，加入椒盐噪声，对比度变暗，对比度变亮，反转，伽马对比度，对数对比度，直方图均衡化等单一数据增强手段。还有一些组合数据增强手段，水平翻转后的图片加入散粒噪声，水平翻转后的图片加入高斯噪声，水平翻转后的图片加入椒盐噪声，水平翻转后的图片对比度变亮，水平翻转后的图片对比度变暗，水平翻转后的图片反转，水平翻转后的图片加伽马对比度，水平翻转后的图片加对数对比度，水平翻转后的图片做直方图均衡化；垂直翻转后的图片加入散粒

噪声，垂直翻转后的图片加入高斯噪声，垂直翻转后的图片加入椒盐噪声，垂直翻转后的图片对比度变亮，垂直翻转后的图片对比度变暗，垂直翻转后的图片反转，垂直翻转后的图片伽马对比度，垂直翻转后的图片加对数对比度，垂直翻转后的图片做直方图均衡化；旋转 90° 后的图片加入散粒噪声，旋转 90° 后的图片加入高斯噪声，旋转 90° 后的图片加入椒盐噪声，旋转 90° 后的图片对比度变亮，旋转 90° 后的图片对比度变暗，旋转 90° 后的图片反转，旋转 90° 后的图片伽马对比度，旋转 90° 后的图片加对数对比度，旋转 90° 后的图片做直方图均衡化；旋转 270° 后的图片加入散粒噪声，旋转 270° 后的图片加入高斯噪声，旋转 270° 后的图片加入椒盐噪声，旋转 270° 后的图片对比度变亮，旋转 270° 后的图片对比度变暗，旋转 270° 后的图片反转，旋转 270° 后的图片伽马对比度，旋转 270° 后的图片加对数对比度，旋转 270° 后的图片做直方图均衡化；在-180° 到 180° 之间旋转随机角度后的图片加入散粒噪声，在-180° 到 180° 之间旋转随机角度后的图片加入高斯噪声，在-180° 到 180° 之间旋转随机角度后的图片加入椒盐噪声，在-180° 到 180° 之间旋转随机角度后的图片对比度变亮，在-180° 到 180° 之间旋转随机角度后的图片对比度变暗，在-180° 到 180° 之间旋转随机角度后的图片反转，在-180° 到 180° 之间旋转随机角度后的图片伽马对比度，在-180° 到 180° 之间旋转随机角度后的图片加对数对比度，在-180° 到 180° 之间旋转随机角度后的图片做直方图均衡化。

但是发现模型的精度只有 72%，后来我们和企业导师探讨，导师给出建议：可能是数据增强中损失了很多信息，比如反转这种在自然场景中不会出现的情况，以及直方图均衡化后螺丝的孔眼的颜色已经黑到人眼无法识别，这些都是不好的数据增强，不好的数据增强反倒破坏了数据的特征。

因此，我们重新对数据进行增强，只保留了水平翻转，垂直翻转，旋转 90°，旋转 270°，在-180° 到 180° 之间旋转随机角度，加入散粒噪声，加入椒盐噪声，对比度变暗，对比度变亮这 9 种增强方式，加上原图，将数据集扩充到原来的 10 倍。由于原始图片的分辨率为 512×512，而使用迁移学习的 VGG19，ResNet 系列模型的输入尺寸为 224×224，因此，需要对图片进行 `resize`，这一步骤在数据增强过程中完成。

数据增强后的每一类别的数据数量如下：

分类		数量
anomaly	缺角零件	140
good	合格零件	16

one	不合格（缺一角螺丝零件）	130
diagonal_two	不合格（缺对角两螺丝零件）	200
neighbor_two	不合格（缺临边两螺丝零件）	180
three	不合格（缺三螺丝零件）	220
four	不合格（缺四螺丝零件）	140
总计		1170

4.2.3 数据清洗

因此，我们重新对数据进行增强，只保留了水平翻转，垂直翻转，旋转 90° ，旋转 270° ，在 -180° 到 180° 之间旋转随机角度，加入散粒噪声，加入椒盐噪声，对比度变暗，对比度变亮这 9 种增强方式，加上原图，将数据集扩充到原来的 10 倍。

数据增强后，由于 `resize` 操作，图片分辨率变小，图像中的细节和信息可能会被压缩或丢失，这是因为调整图像大小通常涉及到对图像进行插值或采样操作，以适应新的尺寸。在这个过程中，一些细节可能会被模糊化或丢失，从而导致图像特征的损失。

因此在数据增强之后，我对训练数据和测试数据进行了人为的清洗，数据清洗是指对原始数据进行处理和转换，以去除错误、不完整、重复或不相关的数据，从而使数据更加准确、一致和可用于分析和建模。具体操作为：有些原始图片，其中，零件所占的图片大小的比例比较小，只有图像中的一小部分，对于这样的图片，我们可以手动的进行一次裁剪，尽可能使得零件部分占据较大的图片比例。

4.2.4 深度学习框架选择

TensorFlow 2 和 Keras 是两个与深度学习相关的 Python 库。

TensorFlow 2 是 Google 开发的开源机器学习框架，它提供了一种灵活且高效的方式来构建和训练各种机器学习模型。相对于早期版本的 TensorFlow，TensorFlow 2 更加易用且用户友好。TensorFlow 2 采用了 Eager Execution 模式，使得代码编写更加直观，并且提供了大量高级 API，使开发者能够快速构建深度学习模型。此外，TensorFlow 2 还增加了许多新功能，如动态图计算、模型序列化和可部署性的改进等。它支持多种硬件设备（如 CPU、GPU 和 TPU）以及分布式计算，可以处理从小规模实验到大规模生产环境的需求。

Keras 是一个高级神经网络 API，最初是作为独立库开发的，后来被整合到 TensorFlow 中。它专注于简化深度学习模型的构建过程，提供了一组简洁而直观的接口，使得用户可以轻松地定义、训练和评估神经网络模型。Keras 具有可扩展性强、模块化设计和易用性等优点，适用于各种任务，包括图像分类、文本处理和序列生成等。在 TensorFlow 2 中，Keras 被作为其官方的高级 API，成为了构建深度学习模型的主要工具。

总结而言，TensorFlow 2 是一个功能强大的机器学习框架，而 Keras 则提供了简洁易用的接口来构建深度学习模型。由于它们的紧密结合，用户在使用 TensorFlow 2 时可以自由选择使用原生 TensorFlow API 或者更简单的 Keras API 来构建自己的模型。

TensorFlow 在工业界受欢迎的原因有以下几点：

广泛的支持和社区: TensorFlow 是由 Google 开发并维护的，得到了大型公司和机构的广泛支持。它拥有庞大而活跃的开发社区，提供了丰富的文档、教程、示例代码和预训练模型等资源，使得使用者可以轻松获取帮助和分享经验。

灵活性和可扩展性: TensorFlow 提供了强大的灵活性和可扩展性，适用于各种规模的任务和需求。它支持多种硬件设备（如 CPU、GPU 和 TPU），以及分布式计算，能够处理大规模数据和复杂模型的训练和推理。此外，TensorFlow 支持多种编程语言接口，包括 Python、C++ 和 Java，使得开发者可以根据自己的偏好选择合适的编程语言。

生态系统和部署能力: TensorFlow 提供了丰富的工具和库，构建了一个完整的深度学习生态系统。例如，TensorFlow Hub 提供了大量的预训练模型和模型组件，方便开发者快速搭建模型；TensorBoard 可以可视化模型训练过程和性能指标；TensorFlow Serving 和 TensorFlow Lite 等工具支持模型的部署和移植到生产环境。这些工具和库提供了综合而完善的解决方案，满足了工业界对于深度学习模型开发和部署的需求。

产业应用和成功案例: TensorFlow 在工业界有许多成功的应用案例，被广泛应用于各个领域，如图像识别、自然语言处理、语音识别、推荐系统等。许多大型公司和组织都在使用 TensorFlow 进行创新研究和产品开发，使得 TensorFlow 成为业界的首选之一。

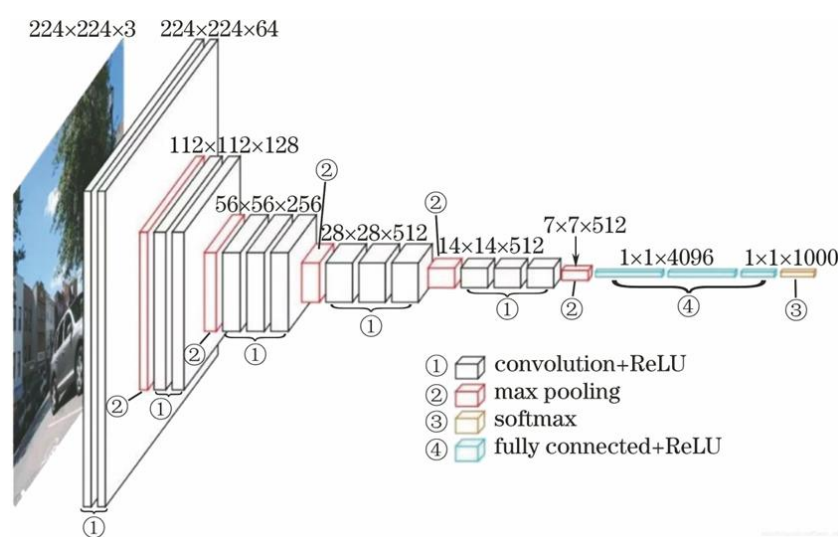
尽管 TensorFlow 在工业界拥有广泛的应用和支持，但也要根据具体需求选择合适的工具。近年来，PyTorch 也获得了很大的关注和增长，成为另一个备受青睐的深度学习框架，特别在学术界和研究领域有较高的使用率。

针对我们这一项目，我们最后决定选用 Tensorflow 中的 keras 来搭建训练模型。

4.2.5 模型训练

因此在数据增强之后，我对训练数据和测试数据进行了人为的清洗，数据清洗是指对原始数据进行处理和转换，以去除错误、不完整、重复或不相关的数据，从而使数据更加准确、一致和可用于分析和建模。具体操作为：有些原始图片，其中，零件所占的图片大小的比例比较小，只有图像中的一小部分，对于这样的图片，我们可以手动的进行一次裁剪，尽可能使得零件部分占据较大的图片比例。

调用 `keras.applications` 中的基于 ImageNet 的预训练模型，只需要模型的特征提取部分，重新搭建分类器，冻结特征提取层的参数，对分类器参数进行训练。



VGG19 的模型结构如图所示，我们的模型将最后 1000 的分类器去掉，更换成我们搭建的分类器，分类数量为 7 类。

在模型的训练过程中，我们尝试了多种预训练的模型，得到的训练结果如下所示：

模型名称 / 模型信息	精度	平均推理时间	参数量
VGG19	90.972%	0.4s	46M
ResNet152_inception	92.361%	3.1s	94M
ResNet152	87.459%	2.7s	161M
ResNet101	88.175%	1.9s	145M
ResNet50	90.732%	1.2s	126M

最后在精度和推理时间的平衡中，我们选择了 VGG19 模型，可以看到，相比于精度最

高的 ResNet152_inception 模型，精度只下降了 2%，但时间却减少到 ResNet152_inception 模型所需时间的 13%，出于精度和推理时间平衡的考量，我们选择了 VGG19 模型。

4.2.6 模型量化

Intel Neural Compressor 是由英特尔开发的一种神经网络压缩工具。它旨在通过减少神经网络模型的大小和复杂度，提高神经网络的推理性能和效率。

Intel Neural Compressor 使用了一系列压缩技术，包括权重量化、剪枝和量化。通过权重量化，它可以将神经网络模型中的浮点权重转换为低精度的整数权重，从而减少存储空间和内存访问的需求。剪枝技术可以去除冗余的神经元和连接，进一步减小模型的大小。量化技术可以将高精度的浮点数转换为低精度的定点数，降低模型的计算复杂度。

通过使用 Intel Neural Compressor，开发人员可以在不显著降低模型性能的情况下，大幅减小神经网络模型的大小和复杂度。这不仅有助于提高模型的推理速度和效率，还可以降低模型在边缘设备上的存储和计算要求。

在 Intel Neural Compressor 中，提供了如下图所示的 API：

Python-based APIs			
Quantization	Advanced Mixed Precision	Pruning(Sparsity)	Distillation
Orchestration	Benchmarking	Distributed Compression	Model Export

最初的想法是使用蒸馏（Distillation）的方法在不显著降低模型性能的前提下，减小神经网络模型的大小和复杂度，和企业导师商议后，最后我们选择量化（Quantization）的方式。

量化（Quantization）是指，将高精度的浮点数转换为低精度的定点数。通常情况下，浮点数需要更多的比特位来表示，而使用定点数可以大幅减小模型的大小。相比于蒸馏（Distillation）技术，量化在神经网络模型中具有以下几个优点：

模型大小减小更明显：量化可以将浮点权重转换为低精度的定点数，从而大幅减小模型的大小。相比之下，蒸馏技术只是通过训练一个较小的模型来近似大模型，减小的模型大小可能没有量化那么显著。

推理速度更快：低精度的定点数计算速度更快，可以加快神经网络模型的推理速度。而蒸馏技术并没有直接减小模型的计算复杂度，因此在推理速度上可能没有量化那么明显的优势。

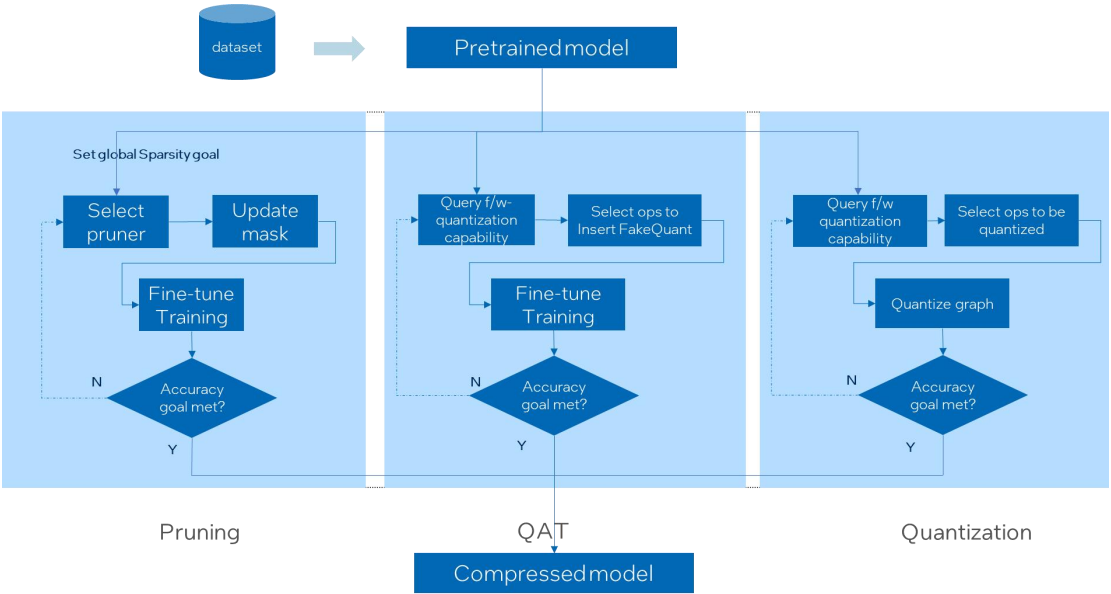
压缩率更高：量化可以将模型的权重量化为低精度的整数或定点数，从而大幅减小模型

的大小。相比之下，蒸馏技术通常只能通过训练一个较小的模型来近似大模型，压缩率可能没有量化那么高。

硬件支持更好：低精度的定点数计算更容易在硬件上实现加速。许多硬件加速器（如 FPGA、ASIC 等）可以更高效地执行低精度的计算操作，从而提高神经网络模型的推理性能。而蒸馏技术并没有直接改变模型的计算精度，对硬件加速的支持可能没有量化那么好。

因此工业界大多选择量化的方式进行神经网络的压缩，为了贴合真实工业界，我们选择量化的方式进行压缩。

量化的工作流程图如下所示：



量化后模型，推理速度有了一定的提升。

4.2.7 后端搭建

Flask 是一个轻量级的 Python Web 框架，它被设计成简单易用、灵活且可扩展的工具。它基于 Werkzeug 和 Jinja2 库构建，提供了构建 Web 应用程序所需的基本功能。

以下是一些 Flask 的主要特点：

简单易用：Flask 的设计理念是保持简单性和最小化的核心功能。它提供了一个简洁的 API，使得开发者能够快速上手并构建出高效的 Web 应用。

轻量级：Flask 没有过多的依赖项，因此它非常轻量级。这使得应用程序的启动时间快，并且占用的系统资源较少。

灵活性：Flask 允许开发者根据自己的需求进行定制。它没有强制性的项目结构，开发

者可以根据自己的喜好组织代码。

可扩展性：Flask 提供了丰富的扩展机制，开发者可以通过使用众多的第三方扩展来增加功能，如数据库集成、身份验证、缓存等。

RESTful 支持：Flask 对构建 RESTful API 提供了良好的支持。它提供了路由、请求处理、响应生成等功能，使开发者能够轻松地构建出符合 RESTful 设计原则的 API。

模板引擎：Flask 使用 Jinja2 作为默认模板引擎，它提供了强大的模板功能，使开发者能够方便地构建动态的 Web 页面。

测试支持：Flask 内置了测试客户端，可以帮助开发者编写单元测试和集成测试，保证应用程序的质量。

总之，Flask 是一个简单、灵活且功能强大的 Python Web 框架，适用于构建各种规模的 Web 应用程序。无论是快速原型开发还是构建大型生产应用，Flask 都能提供良好的支持和可扩展性。

基于上述原因，本项目选择使用 flask 作为后端开发框架，进行开发。

后端的主要接口有：

单张图片上传接口，接收邮箱地址接口，批量上传图片接口。

其中，还有一个调用加载训练好的模型进行推理的 `predict` 函数，接受图片路径，返回检测结果、推理时间、以及置信度。

单张图片上传接口，接口路由和方法为 `@app.route('/upload', methods=['POST'])`，接受前端用户上传的图片，然后保存在服务器，调用 `predict` 函数，传入图片路径，获得模型推理得到的检测结果、推理时间和置信度，以 json 格式返回给前端。

批量上传图片接口，接口路由和方法为 `@app.route('/upload_zip', methods=['POST'])`，接受前端用户上传的 zip 压缩包，然后保存在服务器，解压后，调用 `predict` 函数，传入图片路径，获得模型推理得到的检测结果、推理时间和置信度，将结果写入 `results.txt` 文件中。

接收邮箱地址接口，接口路由和方法为 `@app.route('/email', methods=['POST'])`，接受前端用户输入的邮箱地址，然后将 `results.txt` 作为邮件的附件发送给用户输入的邮箱。

在接口开发完成后，与前端通信连接之前，我们使用 Postman 对后端接口进行测试。

Postman 是一款广泛使用的 API 开发和测试工具，它提供了一个用户友好的界面，使得开发者能够轻松地创建、测试和调试 HTTP 请求。Postman 可以在多个平台上运行，包括 Windows、Mac 和 Linux。

以下是一些 Postman 的主要特点：

请求构建：Postman 允许开发者通过简单的界面构建各种类型的 HTTP 请求，包括 GET、POST、PUT、DELETE 等。开发者可以设置请求头、请求体、参数和身份验证等信息，以及选择合适的 HTTP 方法。

请求测试：Postman 提供了强大的测试功能，可以对请求的响应进行断言和验证。开发者可以编写测试脚本来检查响应的状态码、内容、结构等，并生成详细的测试报告。

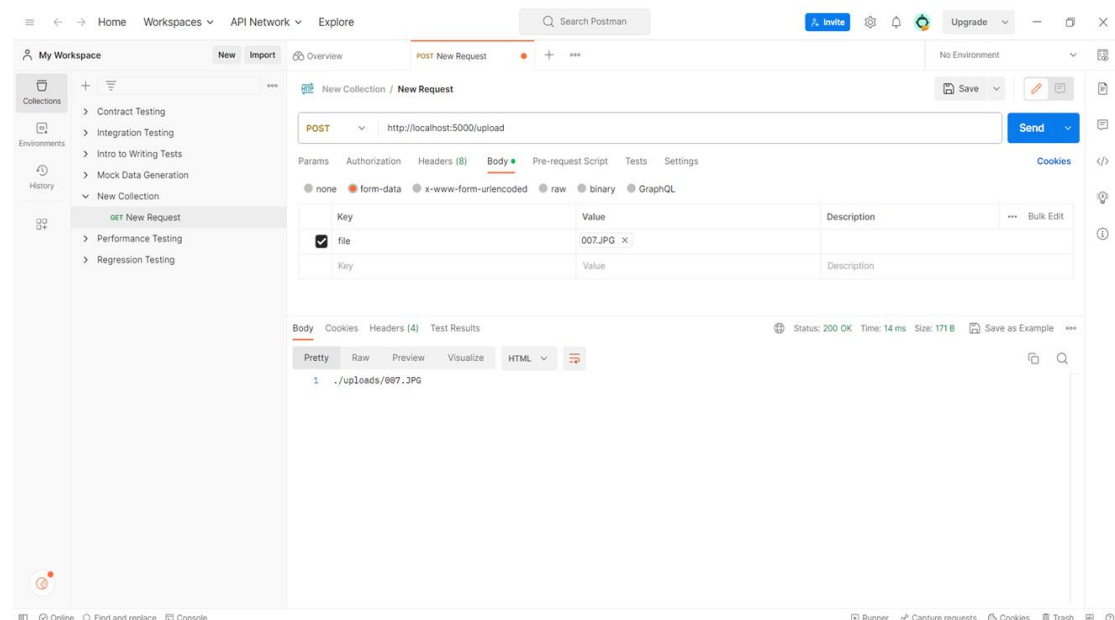
环境管理：Postman 支持环境变量的管理，开发者可以方便地切换不同的环境，如开发环境、测试环境和生产环境。这样可以减少手动修改请求的麻烦，并且提高了测试的灵活性。

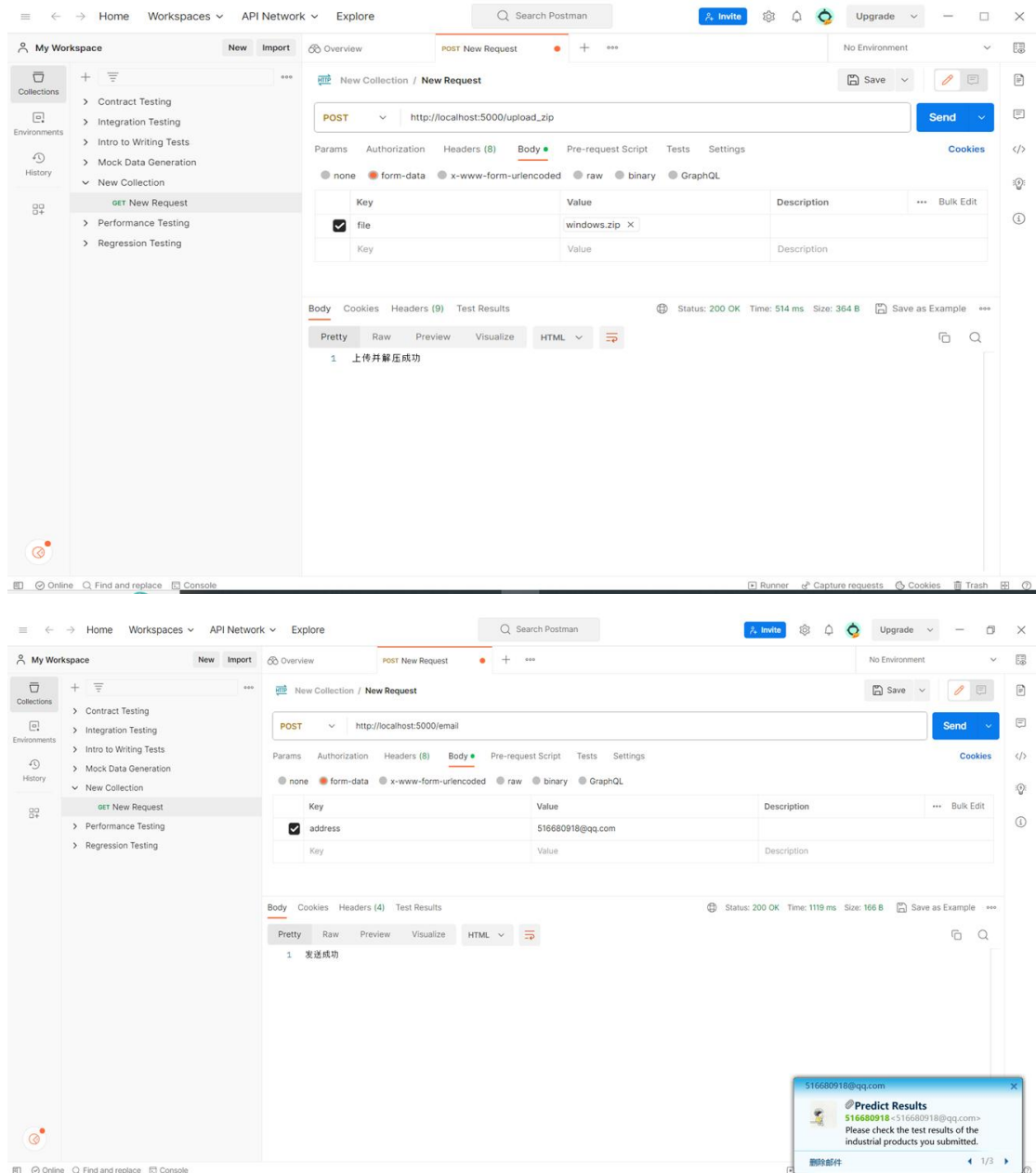
自动化集成：Postman 可以与其他工具和服务进行自动化集成，如持续集成/持续交付（CI/CD）流程中的测试自动化。它提供了 API 和命令行工具，使得开发者能够将 Postman 集成到自动化流程中，并进行自动化测试和监控。

文档生成：Postman 可以根据 API 请求和响应的信息自动生成文档，包括接口说明、参数列表、示例代码等。这样可以方便地与团队成员共享 API 文档，并提供给其他开发者参考和使用。

总之，Postman 是一款功能强大的 API 开发和测试工具，它提供了简单易用的界面和丰富的功能，使得开发者能够更高效地创建、测试和调试 HTTP 请求。无论是开发 API 还是进行 API 测试，Postman 都是一个非常有用的工具。

下面是我们使用 Postman 进行接口测试的结果：





4.2.8 Web 前端搭建

Vue.js（通常称为 Vue）是一种流行的 JavaScript 前端框架，用于构建用户界面。它专注于视图层，并通过简单、灵活的语法和组件化的思维方式，使开发者能够轻松地构建交互性强、可复用的 Web 应用。

以下是 Vue 的主要特点：

响应式：Vue 采用了响应式的数据绑定机制，即当数据发生变化时，相关的视图会自动更新。这使得开发者能够以声明式的方式编写代码，而无需手动操作 DOM。

组件化：Vue 鼓励开发者将 UI 拆分为独立的组件，每个组件具有自己的状态和功能。

组件可以嵌套使用，形成一个层次结构，从而使代码更加模块化、可维护和可复用。

虚拟 DOM：Vue 使用虚拟 DOM 来提高渲染性能。它在内存中创建一个轻量级的虚拟 DOM 树，与实际的 DOM 进行比较，并最小化实际 DOM 的更新操作，从而提高页面的响应速度。

模板语法：Vue 使用基于 HTML 的模板语法，允许开发者将 JavaScript 表达式直接嵌入到模板中。这样可以方便地处理动态数据和逻辑，同时提供了一些指令和过滤器来处理常见的 DOM 操作。

插件生态系统：Vue 拥有一个丰富的插件生态系统，开发者可以根据自己的需求选择合适的插件来扩展 Vue 的功能。这些插件可以用于路由管理、状态管理、表单验证等方面。

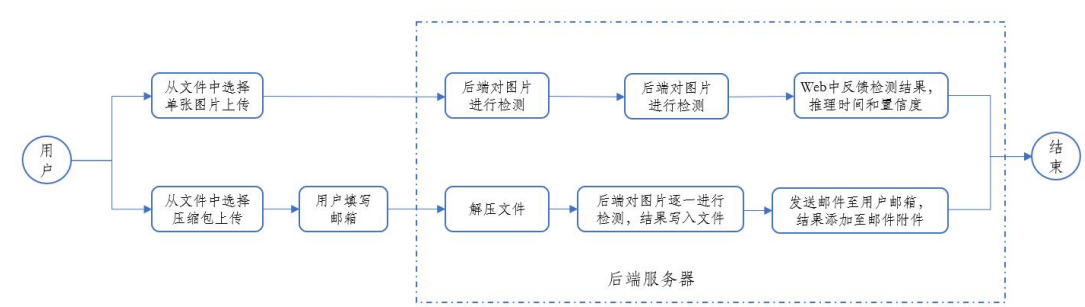
社区支持：Vue 拥有一个活跃的社区，提供了大量的文档、教程和示例代码。开发者可以从中获取帮助、分享经验，并与其他开发者进行交流。

总之，Vue 是一款简单、灵活且高效的 JavaScript 前端框架，它通过响应式数据绑定、组件化和虚拟 DOM 等特性，使开发者能够更轻松地构建交互性强、可复用的 Web 应用。无论是小型项目还是大型企业级应用，Vue 都是一个值得考虑的选择。

基于上述原因，本项目选择使用 Vue 来进行开发。

Web 端需要提供如下功能：单张图片上传，后端提供检测服务，前端显示检测结果，推理时间以及置信度；多张图片以压缩包形式上传，然后提示用户输入邮箱，后端将检测结果，推理时间以及置信度以邮件的形式发送至邮箱。

Web 端服务流程图如下：



具体而言，Web 端实现如下：

Vue-font 文件夹下是前端所有内容。

`node_modules` 在 Vue 项目中，`node_modules` 文件夹是用来存放所需的第三方库和依赖模块的目录。当你使用 `npm`（Node 包管理器）或者 `yarn` 安装依赖时，这些依赖项将会自动下载并保存在 `node_modules` 文件夹中。`node_modules` 是一个由 NPM 生成和维护的文件夹，并且它位于项目根目录下。该文件夹保存了每个依赖包的代码、资源文件和其他相关文件。当你在项目中引入某个依赖时，Vue 会从 `node_modules` 文件夹中查找相应的包，并加载它们的模块。尽管 `node_modules` 文件夹可能会包含大量文件，但你无需手动处理它们。构建工具（如 `webpack`）会自动处理这些依赖，并确保只打包所需的部分。这样可以减小最终生成的应用程序的大小。

`public` 是创建 vite 项目自带的标识，在 Vue 项目中，`public` 文件夹是存放静态资源的地方。当你将资源放置在 `public` 文件夹中时，它们会被直接复制到构建输出的根目录。这使得你可以轻松地引用这些资源，而无需进行特殊的配置。

`src` 里面是前端界面的各种内容。其中 `assets` 存放写好的静态方法 `global.css` 和 `vue3` 自带的图标。`components` 存放 `HelloWorld.vue` 为前端的主界面，`Header.vue` 为前端的头部框架。`Layout` 存放 `Layout.vue` 是前端具体架构布局。`Router` 中的 `index.js` 是前端用来配置开发环境和生产环境的配置参数，控制前端的界面跳转。`views` 中的 `mul.vue` 和 `single.vue` 是多图片和单图片两个功能各自的界面。`main.js` 设置入口文件并对前端使用的 `elementplus` 等进行引入。`style.css` 是对于一些组件、字体等用 `css` 的方法进行样式修改。

index.html

Vue 项目中的 `index.html` 是项目的主入口文件，它是一个 HTML 文件，作为整个 Vue 应用的起点。在这个文件中，你会找到以下重要内容：

`<!DOCTYPE html>`: 这是文档类型声明，告诉浏览器使用哪种 HTML 版本解析。

`<html>`: HTML 页面的根元素。

`<head>`: 包含了一些页面的元数据和引用的外部资源，例如 `css` 样式和 `JavaScript` 脚本等。

`<title>`: 指定网页的标题，在浏览器标签栏和搜索结果中展示。

`<body>`: 页面的主体部分，包含实际的内容。

`<div id="app">`: Vue 应用的根元素，Vue 实例将挂载到这个 DOM 元素上。

`<script>`: 引入各种 `JavaScript` 文件，包括 Vue 的核心库和其他自定义的脚本。

`<style>`: 定义页面的样式，可以是内联样式或者引入外部的 `CSS` 文件。

此外，Vue Cli 生成的默认 `index.html` 还会包含一些脚本和样式相关的标签。例如：

`<link rel="icon">`: 指定网站图标的链接。

`<meta name="viewport">`: 设置视口（viewport）属性，控制移动设备上显示的页面宽度和缩放比例。

`<script src="..."></script>`: 引入打包后的 JavaScript 文件。

在 `index.html` 中，你可以设置全局的样式和引入其他依赖的库，同时也是 Vue 应用的起始点。在 Vue 应用中，`index.html` 一般只有一个实例绑定的根元素，所有的组件都会在 Vue 组件中进行处理，并渲染到这个根元素中。

package.json: 在该 VUE 项目中，`package.json` 是一个重要的配置文件，用于管理项目依赖和脚本命令。它位于项目的根目录，并且包含以下主要内容：

name: 项目的名称 `"vue-font"`。

version: 项目的版本号 `"0.0.0"`。

"type": `"module"`。

scripts: 定义了一些可执行的脚本命令，比如运行开发服务器、打包项目等。`"dev": "vite", "build": "vite build", "preview": "vite preview"`

另外，`package.json` 还包含其它属性，用于配置项目的依赖管理和构建工具。例如：

dependencies: 项目的生产环境依赖，即项目在运行时所需的依赖。

devDependencies: 项目的开发环境依赖，即项目在开发过程中所需的依赖，如测试工具、构建工具等。

package-lock.json: `package-lock.json` 是 Vue 项目中的一个文件，记录了当前安装的依赖包的确切版本号和依赖关系树。它是由 `npm install` 命令自动生成并在每次安装或更新依赖时更新。作用为锁定依赖版本、加快依赖安装速度、保证构建一致性；当其他开发者从仓库拉取代码时，只需运行 `npm install` 即可根据 `package-lock.json` 安装相应的依赖。此时，`package-lock.json` 将会被读取，并且确保所有开发者使用的依赖版本与之前的一致。

vite.config.js: `vite.config.js` 是用于配置 Vite 项目的配置文件。插件部分：导入了 `'vite-plugin-vue'` 插件，并在 `plugins` 数组中进行了注册。服务器部分：在 `server` 对象中添加了 `proxy` 属性，该属性用于配置代理服务器。其中，`/api` 路径是要被代理的请求路径，`target` 属性指定了目标路径为 `"http://127.0.0.1:5000"`，`changeOrigin` 属性表示是否改变请求头中的 `host` 字段，`rewrite` 属性用于重写请求路径。

4.2.9 微信小程序前端搭建

微信开发者工具是一款由微信官方提供的集成开发环境（IDE），用于开发和调试微信小程序、小游戏和公众号。它提供了丰富的功能和工具，使开发者能够快速构建、预览和调试自己的应用。

微信开发者工具的主要特点包括：

代码编辑器：内置了代码编辑器，支持 JavaScript、CSS 和 WXML 的编写，并提供了语法高亮、自动补全等功能，方便开发者编写代码。

调试工具：提供了调试功能，可以在开发者工具中模拟真机环境进行实时调试和查看日志信息，帮助开发者快速定位问题并进行修复。

页面预览：可以在开发者工具中实时预览小程序的界面效果，包括页面布局、样式和交互等。开发者可以随时查看修改后的效果，提高开发效率。

功能扩展：开发者工具支持插件扩展，可以通过安装插件来增加额外的功能和工具，如代码检查、性能分析等。

小程序发布：提供了小程序的打包和上传功能，开发者可以将开发完成的小程序包上传到微信平台进行审核和发布。

关于微信小程序开发语言，WXML 是微信小程序的模板语言，用于描述小程序的界面结构。以下是 WXML 的一些特点：

类似 HTML：WXML 的语法与 HTML 类似，使用标签和属性来描述页面的结构和样式。

数据绑定：WXML 支持数据绑定，可以将 JavaScript 中的数据动态渲染到页面上，实现页面内容的动态更新。

事件处理：WXML 支持事件绑定，开发者可以在页面上绑定各种事件，如点击、滑动等，然后在对应的 JavaScript 代码中处理事件逻辑。

条件渲染和循环渲染：WXML 支持条件渲染和循环渲染，可以根据不同的条件或数据进行页面元素的显示和隐藏，以及列表数据的展示。

组件化：WXML 支持组件化开发，可以将页面拆分为多个组件，每个组件具有自己的模板、样式和逻辑，提高代码的复用性和可维护性。

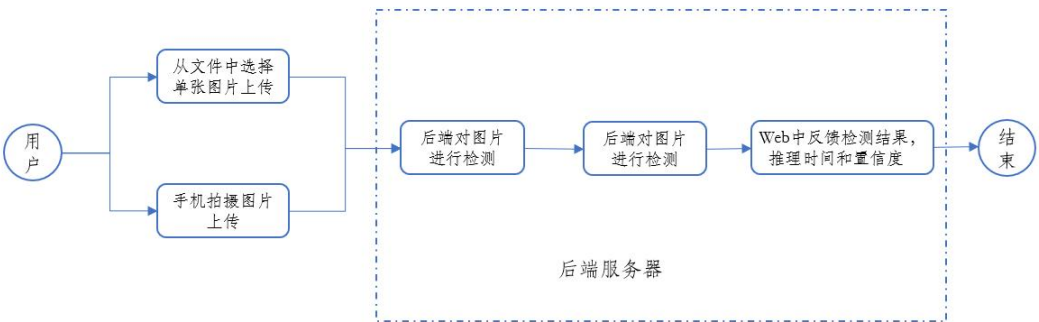
总之，微信开发者工具是一款用于开发和调试微信小程序的集成开发环境，提供了丰富的功能和工具。而 WXML 作为微信小程序的模板语言，用于描述小程序的界面结构，并支持数据绑定、事件处理、条件渲染和组件化等特性。这些工具和语言使开发者能够更高效地

开发出功能丰富、用户友好的微信小程序应用。

基于上述原因，本项目选择使用微信开发者工具作为 IDE，进行开发微信小程序。

微信小程序端需要提供：单张图片上传，后端提供检测服务，前端显示检测结果，推理时间以及置信度；单张图片手机拍照上传，后端提供检测服务，前端显示检测结果，推理时间以及置信度。

微信小程序端服务流程图如下：



具体而言，Web 端实现如下：

WechatProject 文件夹下是微信小程序前端所有内容。

Pages 文件夹包括前端所有页面内容，其下属文件夹是各个界面的设计。包括视图层（WXML 文件和 WXSS 文件）和逻辑层（JavaScript）。

Home 文件夹是小程序首页界面，home.wxml 和 home.wxss 是对首页界面的渲染文件，home.js 包含了此界面的逻辑设计，包括各种点击事件，响应事件，与后端的交互等。home.json 是对整个界面的全局设计。

Images 文件夹包括了整个小程序前端界面所使用到的各种图片背景等

Index 界面是小程序主要的运行界面，同样是分成四个文件，index.wxml 和 index.wxss 对界面进行渲染，index.js 完成了界面的逻辑设计，index.json 是对整个界面的全局设计

Utils 文件夹主要包含 util.js 文件，用来存放全局的一些 js 文件，公共用到的一些事件处理代码文件可以放到该文件夹下，用于全局调用。

App.js 文件是小程序入口文件，用于定义全局数据和函数的调用，可以指定微信小程序的生命周期函数。

App.json 文件对小程序进行配置，小程序的全局配置，小程序的所有页面路径、界面表

现、网络超时时间、底部 tab 等，可以在这个文件中配置小程序是由哪些页面组成，配置小程序的窗口及背景色，配置导航条样式，配置默认标题。

App.wxss 文件是全局的样式文件

Project.config.json 文件用来保存开发工具配置项

Sitemap.json 文件是网络地图，可以对小程序进行 seo 优化，让搜索排名靠前。

4.2.10 软件测试

测试主要分为单元测试，集成测试，功能测试，性能测试。

在编写完系统程序后，需要对诚信进行调试。而断点是调试的核心，它是 .NET 的一个指令，能够使代码运行到指定的行，然后停下来等待用户检查应用程序当前的状态。断点模式可以看作是一种超时，所有元素（如函数、变量和对象）都保留在内存中，但他们的移动和活动被挂起了。在中断模式下可以检查它们的位置和状态，以查看是否存在冲突或 bug。可以在中断模式下对程序进行调整，如果没有这个功能，调试大的程序几乎是不可能的。

首先是添加断点。设置一个断点，当遇到该断点所在的代码时就终端执行。在断点窗口可以查看文件中的断点信息。使用断点窗口可以查看文件中的断点信息。通过选择断点窗口中对应的复选框，能够启用或禁用所有断点。在该窗口中，可以禁用断点、删除断点、编辑断点的属性。该窗口中还显示了条件和命中次数两个可用属性，它们是非常有用的。右击“断点”，在弹出的快捷菜单中选择相应的条件，可对其进行编辑。

为提供更大的灵活性，通过 Visual Studio 调试器能够设置属性以修改断点的行为。通过属性菜单可对它们进行设置。右击断点轮廓左边缘，弹出断点属性菜单。在属性菜单中，前两个命令可删除或者禁用所选中的断点。当选择“禁用断点”命令时，该命令会发生切换，断点图标将显示为一个空心圆。

选择“位置”命令，将会弹出“文件断点”对话框，其中包括清晰地断点说明信息。选择“条件”命令，将会弹出“断点条件”对话框。在此对话框中的文本框内可以输入任何有效的表达式。当程序运行到断点时，将计算该表达式的值。依据所选中的单选按钮和条件表达式所计算的值，所运行应用程序或者停止，或者继续。两个单选按钮如下。为 True(T)：如果输入条件计算所得是一个 Boolean 值 True，那么应用程序将会暂停。如所输入条件发生变化，那么应用程序将会暂停。第一次运行这段正在调试的代码时，断点并不会使程序暂停，因为没有内容可比较。第二次运行这段代码时，会初始化表达式，同时将对二者进行比较。

选择“命中次数”命令，将会弹出“断点命中次数”对话框。在默认条件下，每次命中断点，调试器就中断执行。通过设置命中次数，可以让调试器每命中两次断点执行一次，也可以每 10 次、每 512 次或任意选择的次数中断一次。

选择“筛选器”命令，将会弹出“断点筛选器”对话框。在此对话框中可指定机器、进程、线程或者它们的任意组合设置断点，前提是该断点必须有效。

选择“命中条件”命令，将会弹出“命中断点时”对话框。在此对话框中指定在命中断点时应执行的操作。默认情况下，调试器中断执行，但是可以选择打印一条消息或者运行一个 Visual Studio 宏。如果选择打印消息而不是中断，则断点的效果与 Trace 语句非常类似。这种使用断点的方法称为跟踪点。当“打印消息”复选框被选中时，窗口左边红色圆形的断点图标会变成菱形。

软件单元测试 `UnitTest`，通过编写测试类，使用上下文断言，进行单元测试。`unittest` 是一个常用的测试框架。它是 Python 标准库中的一部分，为编写和运行测试提供了丰富的功能。`unittest` 提供了一个测试运行器和一组用于编写测试的类和方法。使用 `unittest` 可以轻松地创建测试用例、组织测试套件并运行测试。

在所有的类和方法均通过了单元测试之后，测试流程来到了集成测试。集成测试是要测试若干代码单元整合行为的结果，与单元测试那种最小代码单元的针对性测试不同，集成测试的写法不是简单的一加一，而是从高层层面模拟一种行为的开始，比如路由跳转，测试的结果，比如当前的路径，在一个黑盒中代码的运行、流转与交互会牵扯到多个代码单元，称之为“集成”。

Flask 对于执行集成测试有着很好的支持。最核心的接口是 `app.test_client()` 方法，它返回的是一个 `FlaskClient` 类型的对象。如果你比较熟悉 Flask 的话，应当知道，绝大部分框架相关的代码都需要在 `app_context` 的上下文中才能正常执行。因此，一个基本的测试样板代码是这样的：

```
app_ctx = current_app.app_context()
app_ctx.push()
client = current_app.test_client(use_cookies=True)
resp = client.get(url_for('/home.index'))
# TODO: 验证结果
app_ctx.pop()
```

以上代码对首页执行了 HTTP GET 请求，并且隐含了视图响应和模板渲染的完整调用

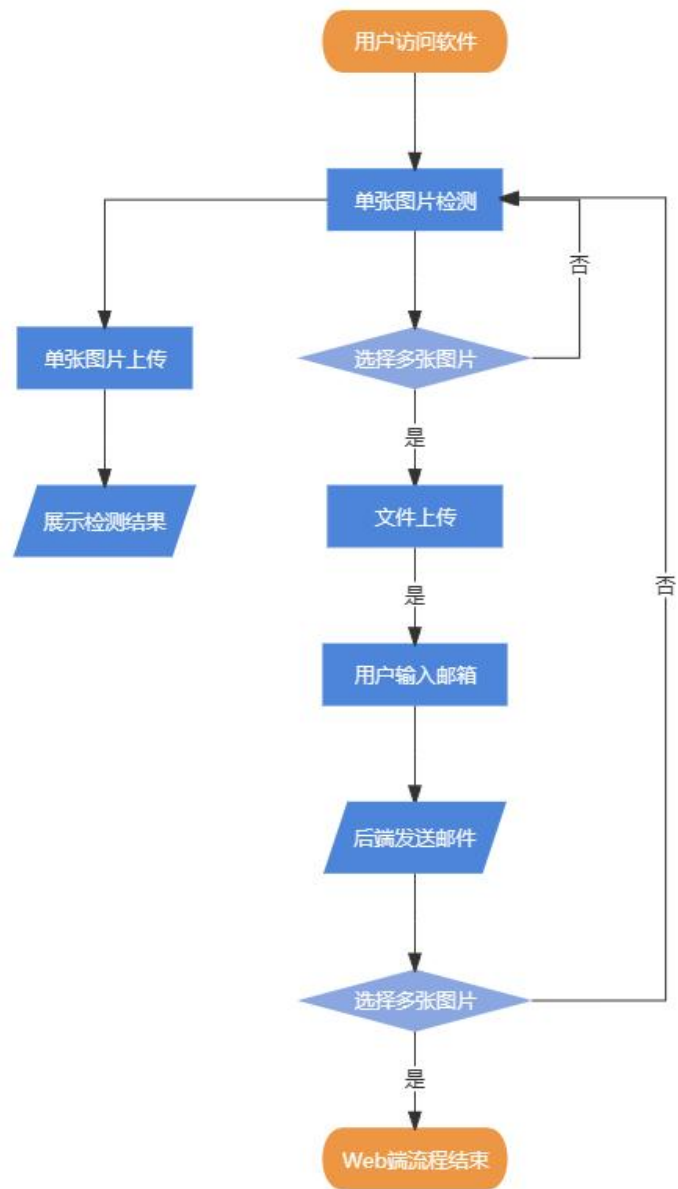
链，可以视为一次完整的服务调用过程。对于 POST 请求而言，则需要增加一个提交表单参数(相同的代码已省略):

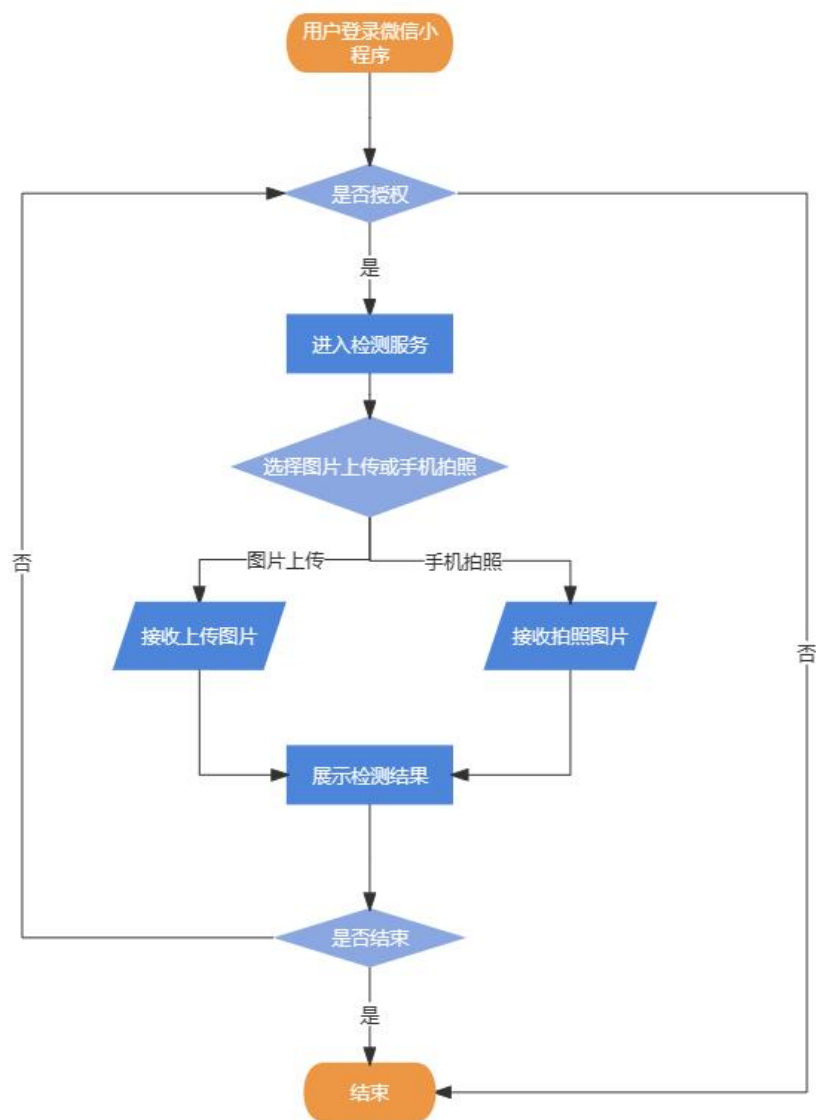
```
resp = client.post('/test_form', data={  
    'field': 'value'  
})
```

后端使用 Postman 对接口进行测试，验证测试结果均为正确后，根据接口文档与前端接口相连接。

根据需求文档中的功能需求分析，逐一对前台功能、后台功能进行测试。

功能总体设计图如下：





在开发中，性能测试是设计初期最容易忽略的问题，系统中出现的大量的字符串拼接、网络调用和频繁的数据库访问都会对系统性能产生影响。但是对于本项目，无需数据库的连接，也少有字符串的拼接操作，因此性能的最主要问题在模型的推理速度上。

性能测试主要包括以下几个方面：

- (1) 计算性能：简而言之就是执行一段代码所需要的时间
- (2) 内存消耗：程序运行所占用的内存大小
- (3) 启动时间：从启动程序到程序正常运行的时间
- (4) 用户察觉性能：不是程序实际运行有多快，而是用户感觉程序运行有多快

Python 为我们提供了 cProfile，一个内置的性能分析工具，可用于分析代码的函数调用和运行时间等；也可以采用 line_profiler，line_profiler 是一个第三方库，可以逐行分析代码的执行时间和内存使用情况。

本系统主要的耗时的函数是模型推理函数，因此只需要对模型推理函数进行性能分析。
代码如下：

```
import cProfile
import tensorflow as tf
import cv2
import numpy as np

classes_name_list=['图片缺角（建议更换图片角度）','合格','不合格（缺一角螺丝）','不合格（缺临边两螺丝）','不合格（缺对角两螺丝）','不合格（缺三角螺丝）','不合格（缺四角螺丝）']
model =
tf.keras.models.load_model(r'C:\Users\LiYanLin\Desktop\defect_product_backEnd\model\VGG19_ft.h5')
def predict(path):
    import time
    img = cv2.imread(path)
    img = cv2.resize(img, (224, 224))
    img_tensor = tf.convert_to_tensor(img, dtype=tf.float16)
    img_tensor /= 255.0
    test_img = np.expand_dims(img_tensor, 0)
    begin_time = time.time()
    out = model.predict(test_img)
    end_time = time.time()
    t = end_time - begin_time
    time_result = str(round(t, 6)) + 's'
    con = str(round(out[0][out.argmax()], 5))
    if round(out[0][out.argmax()], 5) > 0.975:
        return classes_name_list[out.argmax()], time_result, con
    else:
        return '该图片中无零件', time_result, con

# 运行性能分析器
profiler = cProfile.Profile()
profiler.enable()

# 执行要测试的函数
predict(r'C:\Users\LiYanLin\Desktop\defect_product_backEnd\dataset\anomaly\009.JPG")
```

```
profiler.disable()
profiler.print_stats()
```

软件通过了上述测试，下面使用 `selenium` 进行自动化流程测试。

`Selenium` 是一个用于自动化 Web 应用程序测试的开源工具集。它提供了一组 API 和库，可用于控制浏览器并模拟用户在 Web 应用程序中的交互操作。

`Selenium` 最初是为浏览器自动化测试而创建的，但它也可以用于各种其他任务，例如数据挖掘和 Web 内容提取。它支持多种编程语言，包括 Java、Python、C#等，使开发人员可以使用自己熟悉的语言来编写测试脚本。

`Selenium` 可以与各种主流浏览器（如 Chrome、Firefox、Safari 等）进行集成，并提供了一系列 API 和方法来模拟用户在浏览器中的操作，如点击、输入文本、提交表单等。它还可以用于处理 JavaScript 弹窗、处理 Cookie、管理浏览器会话等。

`Selenium` 提供了灵活且强大的功能，使开发人员能够自动化测试 Web 应用程序的各个方面，包括功能测试、界面测试、性能测试等。它可以模拟真实用户的行为，并通过验证结果来确保应用程序的正确性和稳定性。

总之，`Selenium` 是一个流行的用于自动化 Web 应用程序测试的工具，它简化了测试过程，并提供了丰富的功能和灵活性，使开发人员能够更高效地进行 Web 应用程序的测试和验证。

经过测试，软件流程正常。

4.3 软件项目管理

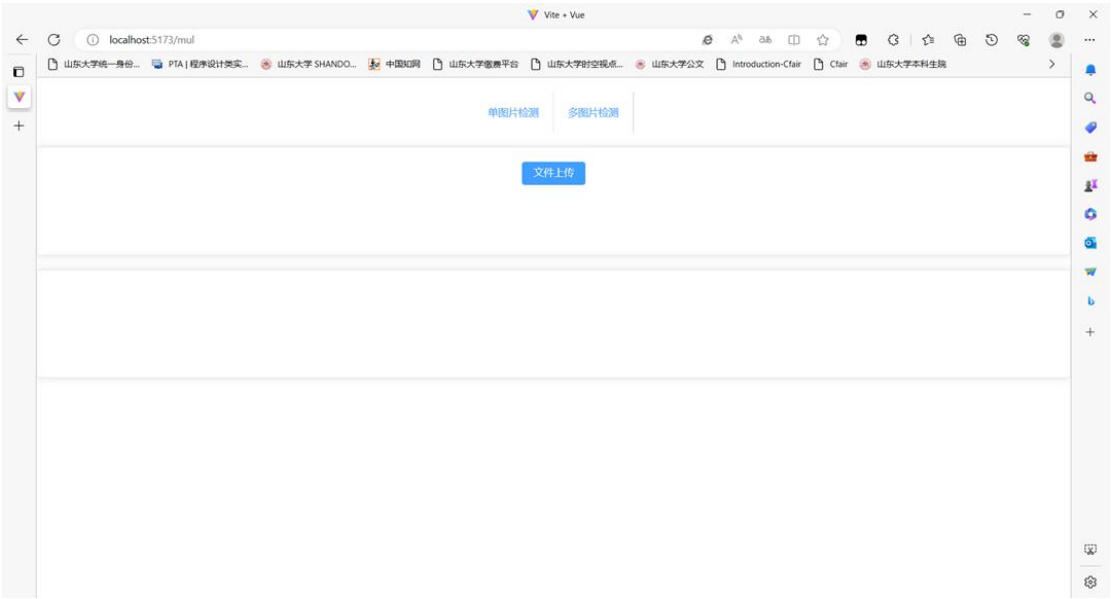
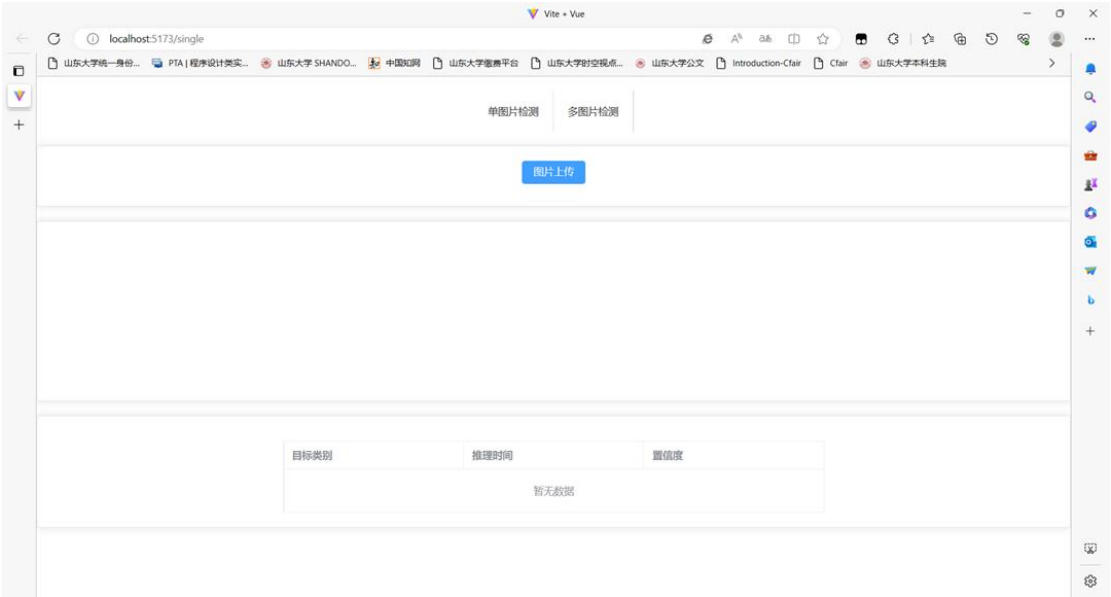
4.3.1 Github

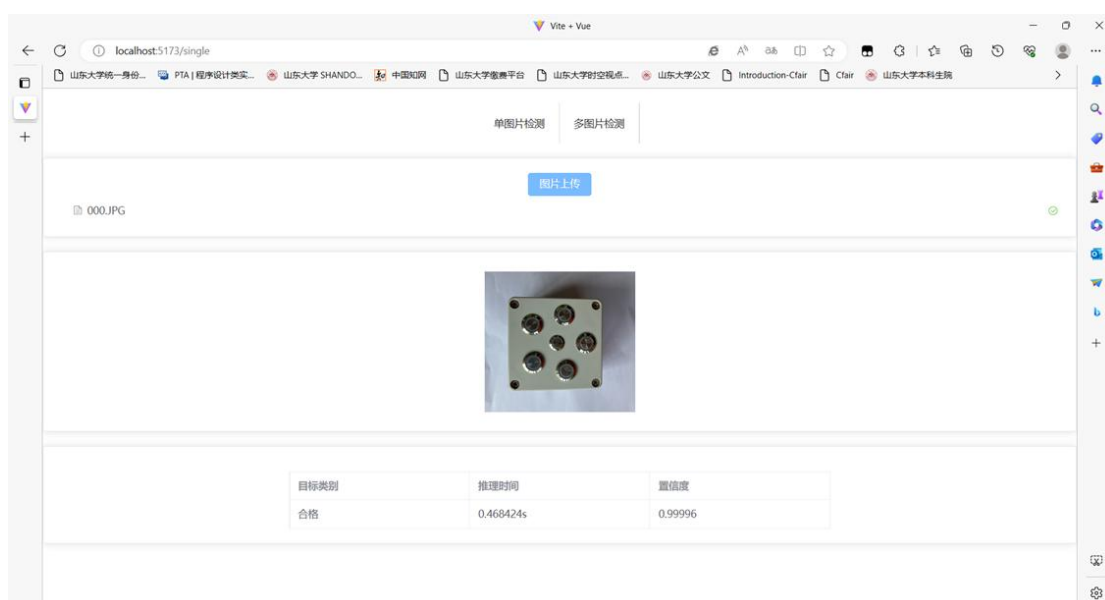
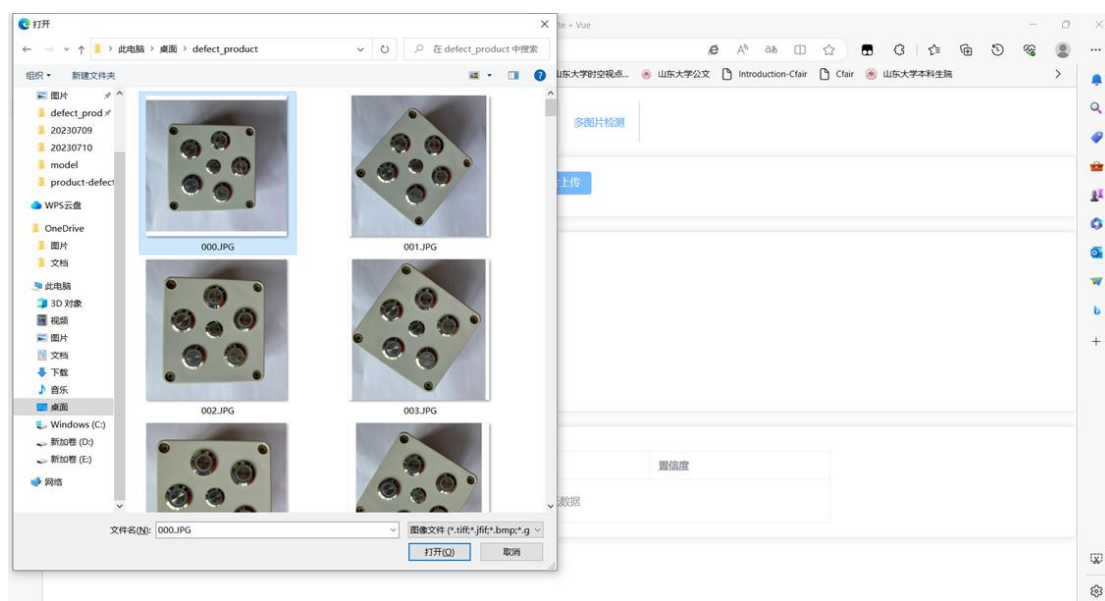
本项目产生的所有有用的过程代码，文档均已经上传 Github，网址为 https://github.com/liyanlin06/defect_product

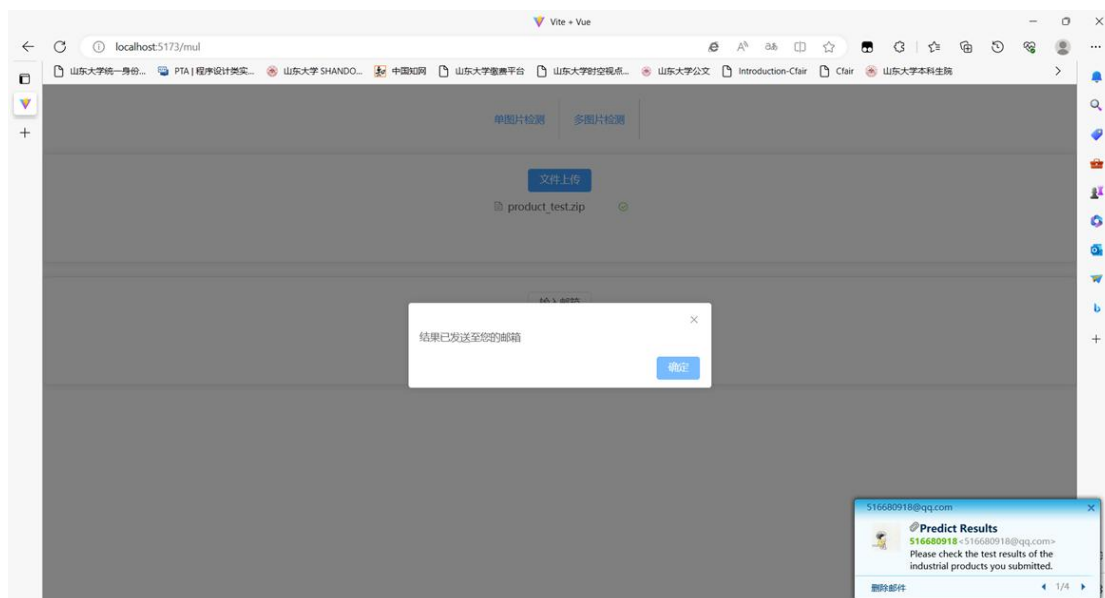
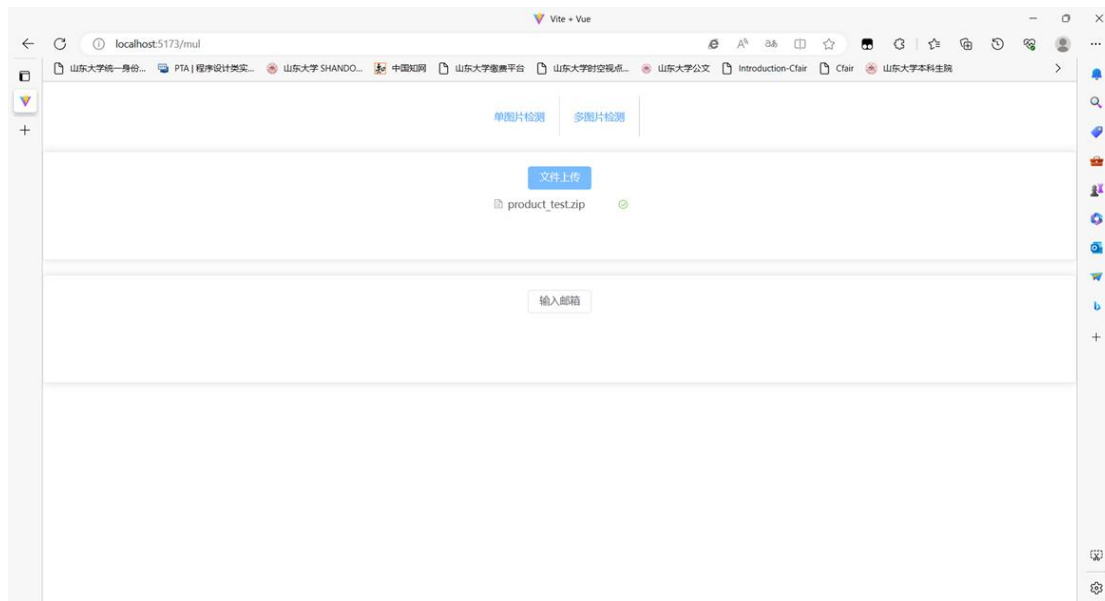
5. 项目成果展示

5.1 Web 系统

Web 前端系统截图如下：

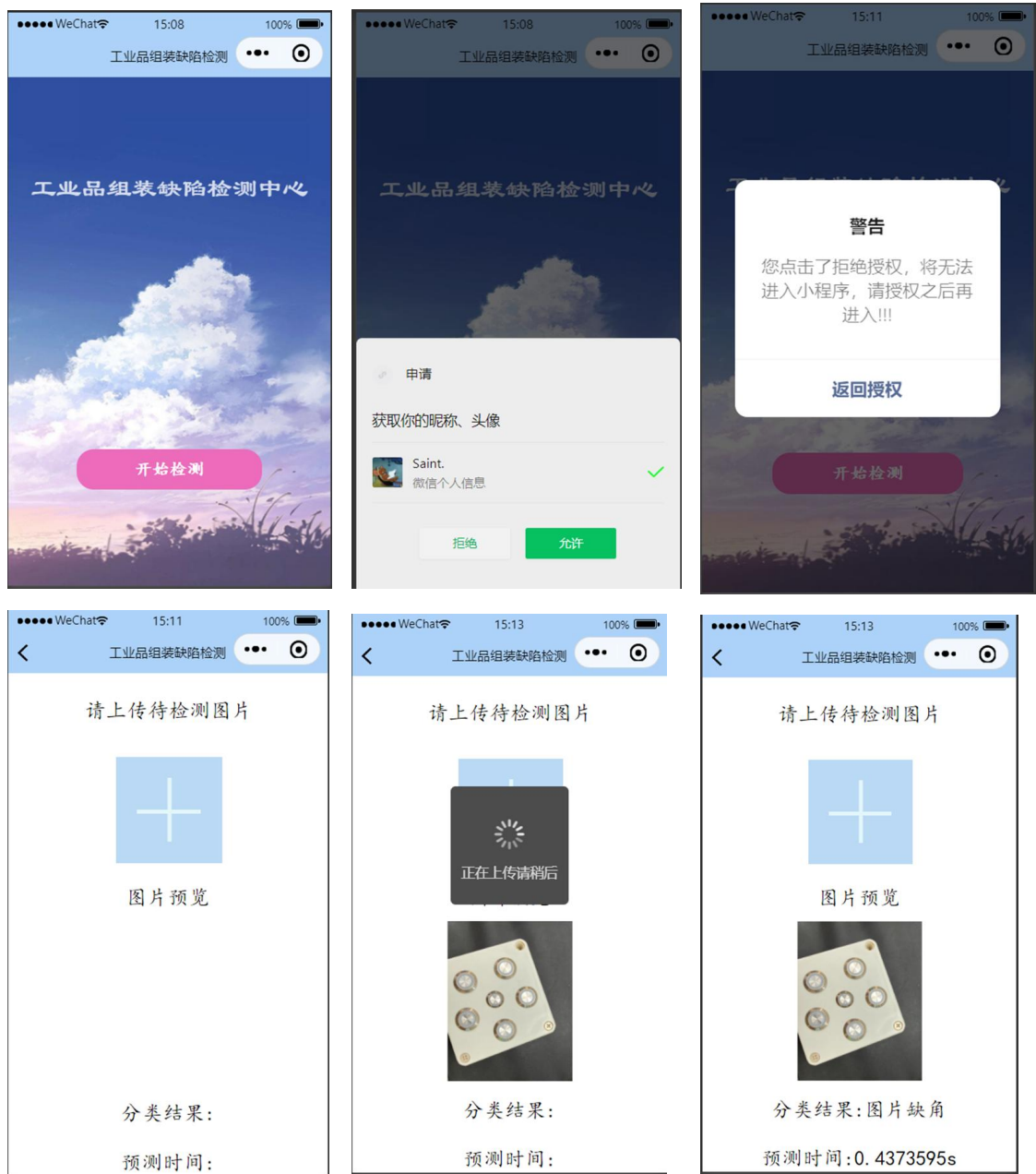






5.2 微信小程序系统

微信小程序系统截图如下：



6. 项目总结

6.1 感悟

在此次项目实训中，我们学到了许多宝贵的经验和知识。通过与团队成员的合作和指导教师的指导，我深入了解了项目开发的流程和技术细节。在这个过程中，我学到了许多关于软件工程和人工智能的实践知识和技能，也锻炼了自己的团队合作和解决问题的能力。

首先，我学习了如何进行项目规划和需求分析。在项目开始之前，我们首先明确了项目的目标和范围，并与团队成员一起制定了详细的项目计划。我们通过与项目相关方的沟通，

了解了他们的需求和期望，并将其转化为具体的功能和特性。这个过程让我意识到，项目的成功与否很大程度上取决于对需求的准确理解和合理规划。

其次，我学习了如何进行团队协作和沟通。在项目实训中，我们组成了一个团队，每个人负责不同的任务和角色。我们通过定期的会议和讨论，共同解决问题和制定决策。这个过程中，我学会了倾听他人的意见，尊重不同的观点，并在团队中建立了良好的合作关系。团队协作不仅提高了项目的效率，也丰富了我的人际交往和团队管理能力。

在技术方面，我学到了许多软件工程和人工智能相关的知识和技能。我们使用了各种开发工具和技术，如 Python、Java、深度学习算法等。通过实际操作和实践，我熟悉了软件开发的流程和方法，学会了使用版本控制工具和测试工具，提高了代码的质量和可维护性。在人工智能方面，我学习了机器学习和深度学习的基本原理和算法，并实践了一些相关的项目。这些技术的学习和应用让我更加深入地理解了人工智能的应用领域和发展趋势。

最后，我认识到项目实训是一个不断学习和成长的过程。在实践中，我们会遇到各种问题和挑战，需要不断调整和改进。通过解决这些问题，我学会了如何快速学习和适应新的技术和环境。同时，我也认识到项目实训是一个团队合作的过程，每个人的贡献都是不可或缺的。只有通过团队的努力和协作，我们才能取得最好的结果。

总的来说，这次项目实训让我收获颇丰。我不仅学到了许多专业知识和技能，也提高了自己的团队合作和问题解决能力。这次实训为我今后的学习和工作打下了坚实的基础，我会继续努力学习和提升自己，成为一名优秀的软件工程和人工智能专业人才。

6.2 项目展望

6.2.1 二分类，三分类，七分类模型的选择与思考

在整个项目实训过程中，关于数据集的分类一直是我們探讨和思考的重点。项目计划书中明确写出，“学生使用图像分类的 AI 算法，识别合格和不合格物品”，但是在数据集中有因为拍摄角度问题而缺角的图片，这些图片不应该算入合格，也不应该算入不合格——因为没有看到零件全貌；同样的，在不合格的零件中，可以具体细化成 5 类：缺一角螺丝，缺对角两螺丝，缺临边两螺丝，缺三螺丝，缺四螺丝。

因此，我们提出了三种分类方式：二分类，三分类，七分类。

二分类是指，整个数据集划分为两类：合格零件和不合格零件。缺角图片作为缺失数据

在数据清洗过程中清洗掉。但是这种模型无法处理缺角图片的情况，需要在模型之前加入传统图像处理方法，比如 Canny 边缘子检测来检查图像中的传统的图像因此我们提出三分类模型来处理缺角情况。

三分类是指，整个数据集划分为两类：合格零件，不合格零件，缺角零件。将缺角数据作为一个类别参与训练，就可以更好的处理缺角情况。但是出于工程角度，仅仅给出合格、不合格、是不具备工程指导意义的。因此，我们提出七分类模型。

七分类是指，整个数据集划分为 7 类：合格零件，缺角零件，缺一角螺丝零件，缺对角两螺丝零件，缺临边两螺丝零件，缺三螺丝零件，缺四螺丝零件。

针对不同的数据分类情况，我们使用基于 ImageNet 预训练的 VGG19 模型进行训练，得到的精度结果如下：

模型名称 / 模型信息	精度	分类情况	存在的问题
二分类 VGG19	97.893%	合格、不合格	无法处理缺角情况
三分类 VGG19	96.112%	合格、不合格、缺角	不合格情况没有细化
七分类 VGG19	90.972%	合格、缺角、缺 1、对角缺 2、临边缺 2、缺 3、缺 4	精度变低

针对七分类模型的精度问题，我们发现七分类 VGG19 模型分类出错的情况集中在缺一角螺丝和缺三角螺丝的情况，这两类之间的特征差异确实相对较小，综合考量，我们还是选用七分类模型，更具有工程意义。

6.2.2 生产线部署设想

在进度汇报中，企业导师提出了一个问题：我们的模型如果输入一个与零件无关的图片，我们的模型应当作何输出？

当时我们没有考虑到这个问题，在后续的迭代过程中，我们根据导师的建议：在推理出结果之后，对置信度进行一个限制。这里的置信度是指，模型 softmax 之后输出七个类别的概率，其中的最大值作为该次检测的置信度。我们设置一个阈值，置信度大于阈值我们认为这是一次有效的检测；如果置信度小于阈值，我们认为这次输入检测的图片中没有零件。

而这样的处理的问题是阈值的选值，异常的情况是无限的，我们没法实现预知所有的情

况，因此，阈值的选值也没有办法适应所有异常情况。

但是如果在真实情况中，在工业品组装的生产线上，固定摄像头在传送带的顶端，固定时间对生产线上的零件拍照，其实完全没有零件的图片就是传送带的照片，完全可以对模型进行拓展，将传送带图片作为一个类别参与训练，即可以解决大部分的异常情况。

