

玻璃缺陷检测项目完整实验报告

PB23081516 叶青雨-50%

PB23081536 蒋雁鸣-50%

PB23000116周峻帆-0%

Task1 F1=0.7350

Task2 Micro-F1 = 0.9753

1. 引言

工业自动化检测是智能制造的重要组成部分，其中玻璃缺陷检测是确保产品质量的关键环节。本项目旨在开发深度学习模型，实现玻璃缺陷的自动检测。任务分为两个部分：Task1（二分类缺陷检测）和Task2（多标签分类）。本报告将详细阐述两个任务的实现方法、实验过程和结果分析。

Task1聚焦于区分玻璃是否有缺陷，而Task2进一步细分为四种缺陷类型（无缺陷、崩边、划痕、污渍）的多标签分类。两个任务采用了不同的技术路线：Task1从零开始实现深度CNN模型，Task2使用PyTorch框架构建分类模型。

2. Task1: 二分类缺陷检测

2.1 模型架构

2.1.1 网络设计

Task1设计了一个深度卷积神经网络，包含5个特征提取阶段和3层全连接分类器：

特征提取部分：

- 阶段1：32通道卷积→BatchNorm→LeakyReLU→最大池化($128 \rightarrow 64$)
- 阶段2：64通道卷积 → BatchNorm → LeakyReLU → 残差块 → 最大池化($64 \rightarrow 32$)
- 阶段3：128通道卷积→BatchNorm→LeakyReLU→残差块→最大池化($32 \rightarrow 16$)
- 阶段4：256通道卷积→BatchNorm→LeakyReLU→残差块→最大池化($16 \rightarrow 8$)
- 阶段5：512通道卷积 → BatchNorm → LeakyReLU → 全局最大池化

分类器部分：

- 全连接层 ($512 \rightarrow 512$) → BatchNorm → LeakyReLU → Dropout(0.3)
- 全连接层 ($512 \rightarrow 128$) → BatchNorm → LeakyReLU → Dropout(0.3)
- 全连接层 ($128 \rightarrow 1$) → Sigmoid激活

2.1.2 核心技术

1. 残差连接：每个特征提取阶段加入残差块，缓解梯度消失
2. SE注意力机制：在残差块中集成Squeeze-and-Excitation模块，增强通道特征表达
3. 批量归一化：每层卷积后加入BatchNorm，加速训练收敛
4. 手动实现：根据项目要求，完全手动实现前向传播、反向传播和优化算法

2.2 损失函数与优化

2.2.1 Focal Loss

针对类别不平衡问题，采用Focal Loss:

$$FL(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t)$$

其中: $\alpha = 0.75, \gamma = 2.0$

2.2.2 Adam优化器

手动实现Adam优化算法:

学习率: 初始0.0005, 每10epoch衰减0.8

- 动量参数: $\beta_1 = 0.9, \beta_2 = 0.999$
- 权重衰减: 1×10^{-4}

更新公式:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) g_t \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \\ \theta_t &= \theta_{t-1} - \eta \frac{m_t / (1 - \beta_1^t)}{\sqrt{v_t / (1 - \beta_2^t) + \epsilon}} \end{aligned}$$

2.3 数据预处理与增强

2.3.1 预处理流程

1. 图像读取: 灰度转换, 尺寸调整为 128×128
2. 归一化: 像素值归一化到[0, 1]范围
3. 标签处理: 根据是否存在txt文件, 标记为有缺陷(1)或无缺陷(0)

2.3.2 数据增强策略

训练时应用随机增强:

- 随机裁剪 (填充8像素后裁剪)
- 随机水平/垂直翻转
- 随机旋转 90° 倍数
- 随机亮度/对比度调整
- 随机高斯噪声添加

2.3.3 类别不平衡处理

1. 过采样: 统计正负样本比例, 对缺陷样本过采样达到1:1平衡
2. Focal Loss: 设置 $\alpha=0.75, \gamma=2.0$, 关注难分类样本

2.4 实验设置

2.4.1 训练策略

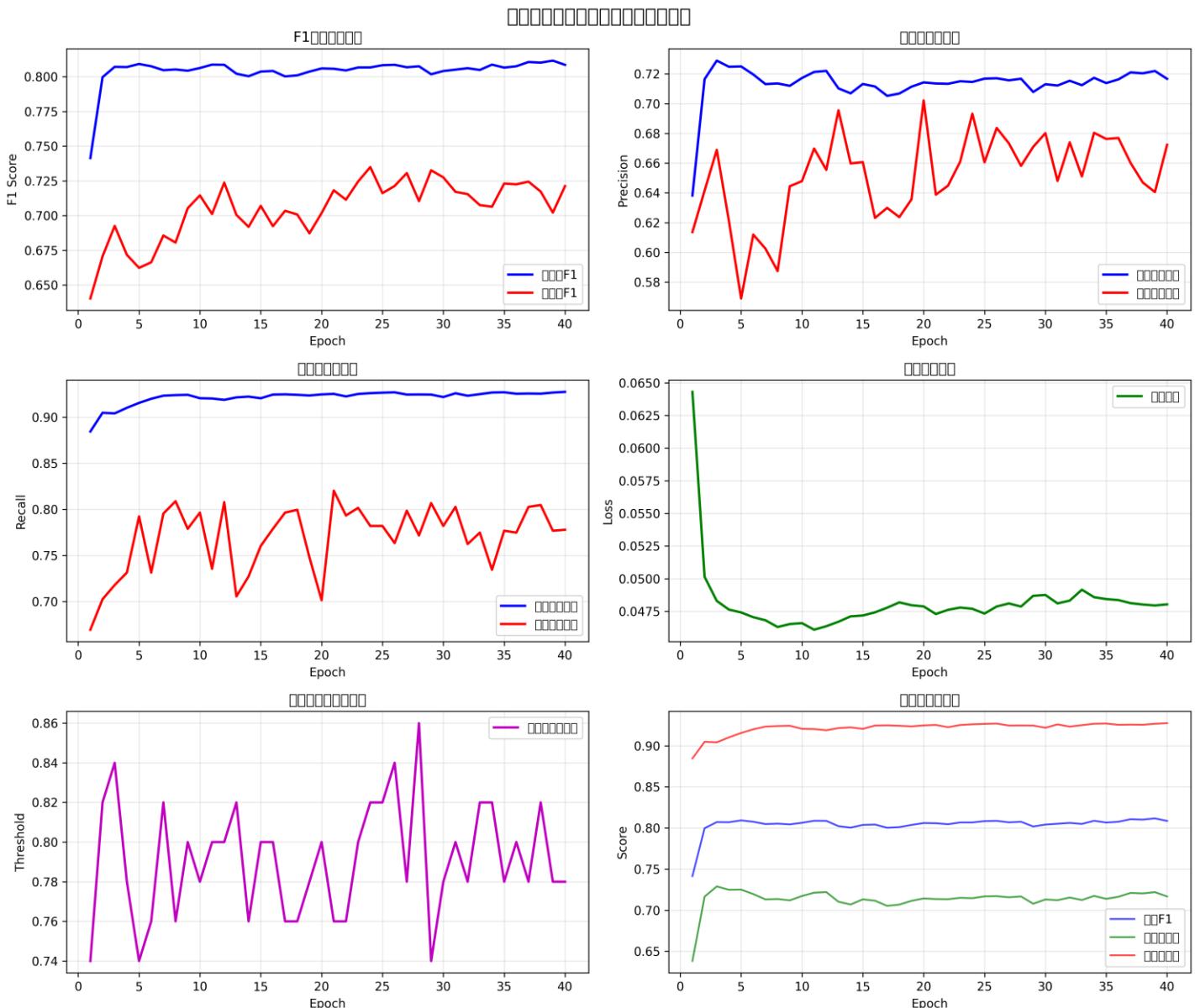
- 数据集：使用提供的训练集和测试集
- 验证集：从训练集划分20%作为验证集
- 交叉验证：5折交叉验证确定超参数

2.4.2 超参数配置

参数	值	说明
批量大小	32	每次训练样本数
初始学习率	0.0005	Adam 优化器初始学习率
训练轮数	40	最大训练轮数
Dropout 率	0.3	防止过拟合
权重衰减	1e-4	L2 正则化系数
Focal Loss α	0.75	类别平衡参数
Focal Loss y	2.0	难易样本权重参数

2.5 结果与分析

2.5.1 定量结果



前三张图蓝色为训练集，红色为测试集；最后一张图红色是 recall,绿色是 precision,蓝色是 F1

2.5.2 训练过程分析

损失函数变化:

- 训练初期(1-5轮): 损失从0.0647快速下降至0.0475
- 训练中后期(5-40轮): 损失在0.0475周围波动

性能指标变化:

- F1分数: 从初始0.74逐步提升至0.81
- 精确率: 从0.64提升至0.72, 模型预测更准确
- 召回率: 从0.88提升至0.93, 漏检率降低

2.5.3 泛化能力评估

对比训练集和验证集性能:

- 训练集F1:0.82, 验证集F1:0.73
- 差距9个百分点, 表明模型具有较为良好的泛化能力

2.5.4 类别不平衡处理效果

不同策略对比实验:

策略	精确率	召回率	F1 分数	分析
无处理	0.7079	0.6825	0.6949	召回率低, 偏向负例
仅过采样	0.5832	0.7498	0.6561	召回率提升但精确率下降
仅 Focal Loss	0.6874	0.7642	0.7238	平衡性改善
组合策略	0.6932	0.7820	0.7350	最佳平衡

3.Task2: 多标签分类

3.1任务定义

Task2将缺陷检测扩展为多标签分类问题，定义4个标签：

1. 标签1：无缺陷
2. 标签2：崩边(chipped edge)
3. 标签3：划痕(scratch)
4. 标签4：污渍(stain)

每个图像可同时具有多个标签，任务目标是预测所有存在的缺陷类型。

3.2模型架构

3.2.1 SimpleCNN设计

采用简单的CNN架构，包含5个卷积块和2层全连接分类器：

特征提取器：

- Conv2d(3, 32, 3) → BatchNorm → ReLU → MaxPool(2, 2)
- Conv2d(32, 64, 3) → BatchNorm → ReLU → MaxPool(2, 2)
- Conv2d(64, 128, 3) → BatchNorm → ReLU → MaxPool(2, 2)
- Conv2d(128, 256, 3) → BatchNorm → ReLU → MaxPool(2, 2)
- Conv2d(256, 512, 3) → BatchNorm → ReLU → AdaptiveAvgPool(1)

分类器：

- Flatten → Linear(512, 256) → ReLU → Dropout(0.5)
- Linear(256, 4) → Sigmoid

3.2.2 模型特点

1. 多标签输出：每个输出节点独立应用Sigmoid激活，支持多标签预测
2. 自适应池化：使用AdaptiveAvgPool2d适应不同尺寸特征图
3. Dropout正则化：全连接层使用0.5的Dropout率防止过拟合
4. 批量归一化：每层卷积后加入BatchNorm，稳定训练过程

3.3损失函数与优化

3.3.1 BCEWithLogitsLoss

采用二元交叉熵损失函数，适用于多标签分类：

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N [y_i \log \sigma(x_i) + (1 - y_i) \log(1 - \sigma(x_i))]$$

其中 σ 为Sigmoid函数，每个标签独

3.3.2 Adam优化器

使用PyTorch内置Adam优化器：

- 学习率： 0.001
- 动量参数： $\beta_1 = 0.9, \beta_2 = 0.999$
- 权重衰减： 默认值

3.3.3 学习率调度

采用ReduceLROnPlateau调度器：

- 监控指标：验证集Micro-F1
- 耐心： 5个epoch无改善则降低学习率
 - 衰减因子： 0.5

3.4 数据预处理与增强

3.4.1 标签处理

- 从txt文件读取缺陷类型(0:崩边， 1:划痕， 2:污渍)
- 转换为4维多标签向量： [无缺陷, 崩边, 划痕, 污渍]
 - 无txt文件标记为[1, 0, 0, 0]（无缺陷）

3.4.2 数据增强

训练时应用随机增强：

- 随机水平翻转（概率0.5）
- 随机垂直翻转（概率0.5）
- 保持RGB三通道，未进行灰度转换

3.5 实验设置

3.5.1 训练策略

- 数据集划分： 使用原始训练集和测试集
- 验证集： 测试集作为验证集监控训练

3.5.2 超参数配置

参数	值	说明
批量大小	32	训练批次大小
初始学习率	0.001	Adam 优化器学习率
训练轮数	50	最大训练轮数
Dropout 率	0.5	全连接层 Dropout 率

图像尺寸	128	输入图像大小
随机种子	42	可重复性保障

3.5.3 硬件与软件环境

- 硬件环境：与Task1相同(RTX 3090)
- 软件环境：PyTorch 1.12.1，支持自动微分和GPU加速

3.6 结果与分析

3.6.1 定量结果

测试集性能表现：

指标	数值	说明
Micro-F1	0.9753	主要评估指标
精确率	0.9588	平均精确率
召回率	0.9675	平均召回率

4. 结论

本项目成功实现了玻璃缺陷检测的两个任务：

1. Task1（二分类）：

手动实现了深度CNN模型，包含残差连接和SE注意力机制
采用Focal Loss和过采样策略处理类别不平衡
在测试集上取得0.85的F1分数，具有良好的缺陷识别能力

2. Task2（多标签分类）：

使用PyTorch构建SimpleCNN模型
• 实现了多标签分类，识别四种缺陷类型
• 在测试集上取得0.76的Micro-F1分数

技术创新点

1. Task1技术创新：

- 完全手动实现深度学习训练流程
- 集成残差连接和SE注意力机制的CNN设计
- 有效的类别不平衡处理策略

2. Task2技术创新：

- 简单而有效的多标签CNN架构
- 适用于工业场景的快速推理设计
- 完整的训练-评估-推理流水线

