

分类号: TP391.4

密 级: 公开

学校代码: 10363

学 号: 2210320106



安徽工程大学

Anhui Polytechnic University

# 硕士学位论文

题 目      基于深度学习的自立袋  
                 封装缺陷检测

论 文 作 者      高 港

校 内 导 师      魏利胜（教授）

校 外 导 师      刘小琿

专业学位类别      控制科学与工程

研究方向（领域）      模式识别与智能系统

论文提交日期:    2024 年 05 月 31 日

分类号: TP391.4

单位代码: 10363

密 级: 公开

学号: 2210320106

题 目 基于深度学习的自立袋封装缺陷检测

题 目 Defect Detection for Doypack Encapsulation

Based on Deep Learning

学 生 姓 名: 高港

校 内 导 师: 魏利胜教授

校 外 导 师: 刘小琿

专业学位(类别): 控制科学与工程

研究方向(领域): 模式识别与智能系统

论文答辩日期: 2024年5月12日

## 安徽工程大学学位论文原创性声明

本人郑重声明：我恪守学术道德，崇尚严谨学风。所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已明确注明和引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品及成果的内容。论文为本人亲自撰写，我对所写的内容负责，并完全意识到本声明的法律后果由本人承担。

学位论文作者签名：高浪

日期：2024 年 5 月 30 日

## 安徽工程大学学位论文版权使用授权书

学位论文作者完全了解学校有关保留、使用学位论文的规定，同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅或借阅。本人授权安徽工程大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密 ☐，在 \_\_\_\_\_ 年解密后适用本版权书。

本学位论文属于

不保密 ☒。

学位论文作者签名：高浪

日期：2024 年 5 月 30 日

指导教师签名：魏利胜

日期：2024 年 5 月 30 日

## 基于深度学习的自立袋封装缺陷检测

### 摘 要

自立袋的性价比高、使用范围广，但生产工序较普通平装袋复杂，更易产生褶皱、未对齐等缺陷。以往人工目视缺陷检测难免受到工人经验和精力等方面的限制，难以适应大规模高效检测，而机器视觉方法也需要进行特征提取规则的人工设计。为减少人力成本，实现缺陷自主检测，研究基于深度学习的自立袋封装缺陷检测方法具有重大意义。本文针对自立袋封装过程褶皱、未对齐等缺陷，分别构建分类和分割数据集，研究监督学习的分类模型与无监督学习的分割模型，以实现自立袋缺陷精准检测。具体研究内容如下：

首先，探讨一种改进 MobileNetV3 网络的自立袋封装缺陷分类方法。在构建自立袋封装缺陷检测分类数据集的基础上，采用旋转等数据增强方法对数据集进行扩充；分析总结缺陷分类网络结构，并以此为基础，详细对比描述 MobileNet 等 4 种分类网络；分别从高斯误差线形单元(Gaussian error linear unit, GELU)、坐标注意力(Coordinate Attention, CA)、深度卷积位置等 3 个方面优化 MobileNetV3 模型；实验结果表明，改进模型在 Accuracy、mRecall、Params、模型大小等指标上取得最优，突显了改进自立袋缺陷检测模型的先进性和实用性。

其次，研究一种基于存储库(Memory Bank, MB)正负样本差异特征的无监督自立袋封装缺陷检测方法。该方法利用 EISeg 构建的自立袋封装分割数据集进行学习优化，以 U-Net 网络结构为基础，引入负样本合成模块、特征存储模块、多尺度特征融合模块以及缺陷分割预测模块，实现无缺陷样本学习的自立袋缺陷检测。在 Doypack-Defects 数据集上验证了所提方法对缺陷分割检测的有效性，

在此基础上，在 MVTec AD 数据集上验证了模型的泛化性。

最后，设计自立袋封装缺陷检测系统。简要阐述自立袋包装设备生产流程，提出与自立袋包装设备相匹配的缺陷检测模块和自立袋缺陷样本剔除模块，搭建缺陷检测系统的架构；根据缺陷检测软件工作流程，设计软件的人机交互界面。

综上所述，本文研究基于监督学习与无监督学习的自立袋封装缺陷检测算法并设计缺陷检测系统，向自立袋包装设备生产检测一体化、智能化发展以及包装行业缺陷检测的应用，提供了重要的参考价值。

**关键词：**自立袋封装缺陷； MobileNetV3； 正负样本差异特征； U-Net 分割

# Defect Detection for Doypack Encapsulation

## Based on Deep Learning

### ABSTRACT

Doypacks are cost-effective and widely used, but the production process is more complex than that of ordinary flat-packed pouches, which is more likely to produce defects such as wrinkles and unalignment. In the past, manual visual defect detection is inevitably limited by the experience and energy of workers, which makes it difficult to adapt to large-scale and efficient detection, while machine vision methods also require manual design of feature extraction rules. In order to reduce the labor cost and realize the defects independent detection, it is of great significance to study the deep learning-based defect detection method for doypacks encapsulation. In this paper, we construct classification and segmentation datasets for the defects such as wrinkles and unalignment of doypacks encapsulation process, respectively, and study the classification model of supervised learning and the segmentation model of unsupervised learning in order to realize the accurate detection of doypacks defects. The specific research content is as follows:

Firstly, a supervised learning doypack encapsulation defect classification method for improving MobileNetV3 networks is explored. On the basis of constructing the classification dataset of doypack encapsulation defect detection, the data enhancement methods such as rotation are used to realize the expansion of the dataset. The structure of the defect classification network is analyzed and summarized, and based on this, four kinds of classification networks, such as MobileNet, are

compared and introduced in detail. The MobileNetV3 model is optimized in three aspects: Gaussian error linear unit (GELU), coordinate attention (CA), and deep convolutional position. The experimental results show that the improved model achieves the optimum in Accuracy, mRecall, Params, and model size indexes, which highlights the advancement and practicability of the improved model for the detection of defects in doypack bags.

Secondly, an unsupervised doypack encapsulation defect detection method based on Memory Bank (MB) positive and negative sample difference features is investigated. The method utilizes the doypack encapsulation segmentation dataset constructed by EISeg for learning optimization, and introduces the negative sample synthesis module, feature storage module, multi-scale feature fusion module, and prediction module for defect segmentation based on the U-Net network structure to realize the doypack defects detection without defective sample learning. The effectiveness of the proposed method for defect segmentation detection is experimentally verified on the Doypack-Defects dataset, based on which the generalization of the model is verified on the MVTec AD dataset.

Finally, we design a doypack packaging defect detection system. Briefly explaining the production process of doypack packaging equipment, the defect detection module and doypack defective sample rejection module matched with doypack packaging equipment are proposed to build the architecture of the defect detection system. According to the workflow of the defect detection software, the human-machine interface of the software is designed.

In summary, this article provides important reference value for the integrated and intelligent development of production and inspection of doypack packaging equipment, as well as the application of defect detection in the packaging industry, by studying doypack packaging defect detection algorithms based on supervised and unsupervised learning and designing defect detection systems.

Gao Gang (Control Science and Engineering)

Supervised by Wei Lisheng

**Key words:** Defect of Doypack Encapsulation; MobileNetV3; Positive and negative sample difference features; U-Net segmentation

# 目 录

|                                       |     |
|---------------------------------------|-----|
| 摘 要 .....                             | I   |
| ABSTRACT.....                         | III |
| 目 录 .....                             | VI  |
| 第 1 章 绪论.....                         | 1   |
| 1.1 选题背景及意义 .....                     | 1   |
| 1.2 国内外研究现状 .....                     | 2   |
| 1.2.1 基于机器学习的缺陷检测研究现状 .....           | 2   |
| 1.2.2 基于深度学习的缺陷检测研究现状 .....           | 3   |
| 1.3 主要研究内容 .....                      | 6   |
| 1.4 本章小结 .....                        | 8   |
| 第 2 章 基于监督学习的自立袋封装缺陷分类检测 .....        | 9   |
| 2.1 构建自立袋封装缺陷检测分类数据集 .....            | 9   |
| 2.1.1 自立袋封装图像采集与分类 .....              | 9   |
| 2.1.2 自立袋封装缺陷检测图像数据增强 .....           | 11  |
| 2.2 缺陷检测分类网络结构设计 .....                | 11  |
| 2.2.1 ResNet 缺陷分类网络结构 .....           | 12  |
| 2.2.2 MobileNet 缺陷分类网络结构 .....        | 13  |
| 2.2.3 ShuffleNet 缺陷分类网络结构 .....       | 15  |
| 2.2.4 EfficientNet 缺陷分类网络结构 .....     | 16  |
| 2.3 改进 MobileNetV3 的自立袋封装缺陷分类检测 ..... | 17  |
| 2.3.1 GELU 激活函数 .....                 | 18  |
| 2.3.2 CA 注意力机制 .....                  | 20  |
| 2.3.3 改进 Bneck 结构 .....               | 22  |
| 2.4 实验结果分析 .....                      | 23  |
| 2.4.1 评价指标 .....                      | 24  |
| 2.4.2 消融实验 .....                      | 25  |
| 2.4.3 对比实验 .....                      | 28  |
| 2.5 本章小结 .....                        | 30  |
| 第 3 章 基于无监督学习的自立袋封装缺陷检测 .....         | 31  |
| 3.1 构建自立袋封装分割数据集 .....                | 31  |
| 3.1.1 EISeg 获取缺陷图像分割 .....            | 31  |
| 3.1.2 Doypack-Defects 数据集构成 .....     | 32  |

|                                       |    |
|---------------------------------------|----|
| 3.2 基于无监督 MB 方法的缺陷检测网络.....           | 34 |
| 3.2.1 基于 MB 的缺陷检测原理 .....             | 34 |
| 3.2.2 自编码器与 U-Net 结构 .....            | 35 |
| 3.2.3 结合 MB 方法和 U-Net 网络的缺陷检测结构.....  | 36 |
| 3.3 基于 MB 正负样本差异特征的无监督自立袋封装缺陷检测 ..... | 37 |
| 3.3.1 负样本合成模块.....                    | 38 |
| 3.3.2 特征存储模块.....                     | 40 |
| 3.3.3 注意力特征融合模块 .....                 | 41 |
| 3.3.4 预测模块 .....                      | 42 |
| 3.4 实验结果分析 .....                      | 43 |
| 3.4.1 评价指标 .....                      | 43 |
| 3.4.2 消融实验 .....                      | 44 |
| 3.4.3 对比实验 .....                      | 46 |
| 3.4.4 泛化实验 .....                      | 47 |
| 3.5 本章小结.....                         | 48 |
| 第 4 章 自立袋封装缺陷检测系统设计 .....             | 49 |
| 4.1 缺陷检测系统设计要求.....                   | 49 |
| 4.2 系统架构设计 .....                      | 50 |
| 4.3 软件系统开发.....                       | 52 |
| 4.3.1 软件系统开发环境.....                   | 52 |
| 4.3.2 软件工作流程.....                     | 53 |
| 4.3.3 人机交互界面.....                     | 54 |
| 4.4 本章小结.....                         | 57 |
| 第 5 章 总结与展望 .....                     | 58 |
| 5.1 总结 .....                          | 58 |
| 5.2 展望 .....                          | 59 |
| 参考文献 .....                            | 60 |
| 附录 .....                              | 65 |
| 攻读学位期间发表的学术论文.....                    | 76 |
| 攻读学位期间获得的奖励 .....                     | 76 |
| 致 谢 .....                             | 77 |

## 插图清单

|                                           |    |
|-------------------------------------------|----|
| 图 1-1 研究内容框架图 .....                       | 7  |
| 图 2-1 自立袋“凸”形封口与“M”形封口样本 .....            | 9  |
| 图 2-2 自立袋“凸”形封口与“M”形封口缺陷样本 .....          | 10 |
| 图 2-3 分类模型检测方法流程图 .....                   | 11 |
| 图 2-4 残差模块结构图 .....                       | 12 |
| 图 2-5 深度可分离卷积示意图 .....                    | 13 |
| 图 2-6 残差块与倒残差块 .....                      | 14 |
| 图 2-7 SE 注意力示意图 .....                     | 15 |
| 图 2-8 分组卷积及通道混洗原理图 .....                  | 15 |
| 图 2-9 MBConv 结构示意图 .....                  | 16 |
| 图 2-10 改进 MobileNetV3 网络示意图 .....         | 17 |
| 图 2-11 GELU 激活函数曲线 .....                  | 19 |
| 图 2-12 CA 注意力机制结构图 .....                  | 20 |
| 图 2-13 CA 注意力机制水平与竖直池化示意图 .....           | 21 |
| 图 2-14 原始 Bneck 块的 3 种结构示意图 .....         | 22 |
| 图 2-15 改进 Bneck 块的 3 种结构示意图 .....         | 23 |
| 图 2-16 消融实验混淆矩阵图 .....                    | 26 |
| 图 2-17 消融实验训练过程准确率曲线 .....                | 27 |
| 图 2-18 各经典模型自立袋封装缺陷检测混淆矩阵 .....           | 30 |
| 图 3-1 缺陷图像与 EISeg 分割生成 3 种图像的对比 .....     | 32 |
| 图 3-2 三种自立袋封口正常样本展示 .....                 | 33 |
| 图 3-3 以 ResNet18 模型为编码器的 U-Net 结构 .....   | 35 |
| 图 3-4 基于 MB 方法 U-Net 网络的缺陷检测结构 .....      | 36 |
| 图 3-5 MB 正负样本差异特征缺陷检测框架 .....             | 37 |
| 图 3-6 合成异常图像流程图 .....                     | 38 |
| 图 3-7 异常模拟图像生成展示 .....                    | 39 |
| 图 3-8 注意力特征融合模块结构图 .....                  | 41 |
| 图 3-9 基于 MB 正负样本差异特征的无监督自立袋封装缺陷检测效果 ..... | 45 |
| 图 3-10 检测效果对比图 .....                      | 46 |
| 图 3-11 MVTec AD 数据集上的检测效果 .....           | 48 |
| 图 4-1 DS180-DS 自立袋包装机示意图 .....            | 50 |
| 图 4-2 DS180-DS 自立袋包装设备作业示意图 .....         | 51 |

|                                   |    |
|-----------------------------------|----|
| 图 4-3 DS180-DS 自立袋包装设备作业流程图 ..... | 51 |
| 图 4-4 自立袋封装缺陷检测系统架构及原理图.....      | 52 |
| 图 4-5 自立袋封装缺陷检测系统软件工作流程图.....     | 54 |
| 图 4-6 自立袋封装缺陷检测系统软件注册界面.....      | 54 |
| 图 4-7 自立袋封装缺陷检测系统软件登录界面.....      | 55 |
| 图 4-8 自立袋封装缺陷检测系统检测模型选择.....      | 55 |
| 图 4-9 自立袋封装缺陷检测系统检测效果演示.....      | 56 |

## 插 表 清 单

|                                                      |    |
|------------------------------------------------------|----|
| 表 2-1 自制自立袋封装缺陷分类数据集数据汇总.....                        | 11 |
| 表 2-2 改进 MobileNetV3 网络各层结构组成表.....                  | 18 |
| 表 2-3 混淆矩阵构成示意表 .....                                | 24 |
| 表 2-4 DWConv 位置、GELU 激活函数与 CA 注意力消融实验结果 .....        | 28 |
| 表 2-5 改进 MobileNetV3 模型的对比实验结果.....                  | 29 |
| 表 3-1 Doypack-Defects 数据集组成结构.....                   | 33 |
| 表 3-2 基于 MB 思想嵌入方法的无监督缺陷检测优缺点总结.....                 | 34 |
| 表 3-3 Tu_Pink 类别缺陷消融实验检测效果 .....                     | 44 |
| 表 3-4 M_White 类别缺陷消融实验检测效果 .....                     | 44 |
| 表 3-5 M_Red 类别缺陷消融实验检测效果.....                        | 45 |
| 表 3-6 Doypack-Defects 数据集缺陷检测经典模型图像级 AUROC 对比表 ..... | 46 |
| 表 3-7 MVTec AD 数据集缺陷检测经典模型图像级 AUROC 泛化实验对比表.....     | 47 |
| 表 3-8 MVTec AD 数据集缺陷检测经典模型像素级 AUROC 对比表 .....        | 47 |
| 表 4-1 软件开发环境 .....                                   | 53 |

## 第 1 章 绪论

### 1.1 选题背景及意义

随着经济信息化程度日益加深，利用电子商务平台选择商品、购物和交易成为一种人们广泛认可和参与的活动。其在带动现代物流行业突飞猛进发展的同时，对流通过程中商品的质量也提出更高要求。虽然公路网、铁路网以及航线网等交通基础设施的不断完善，为琳琅满目商品的流通提供了极大便利，但是商品在运输过程中仍会出现各种形式的损坏问题。为尽可能避免此类问题，除在运输阶段大量使用柔性材料包裹产品外，在商品的生产阶段选取合适的包装材料与技术，对产品包装的缺陷进行检测以提高产品出厂良品率也至关重要。据中国包装行业 2022 年度运行报告显示，塑料包装产品产量增速最高，全年累计完成产量 1692.86 万吨，累计主营业务收入 3822.01 亿元，同比增长 4.19%，占整个包装行业的 31.09%<sup>[1]</sup>。虽然塑料材料通常被认为难以降解，但是随着环保型塑料包装材料的发展和人类环保意识的增强，塑料包装袋在多数情况下得以回收和再利用，使其成为一种成本低、性价比高的包装形式<sup>[2]</sup>。

自立袋作为一种降本增效的塑料软包装形式，受到广大消费者的欢迎。它是一种底部带有水平支撑结构、能自行站立且有效支撑和展示商品的软体包装袋，于上世纪 90 年代中后期传入中国，并逐渐得到国内市场的认可<sup>[3]</sup>。自立袋主要分为普通、拉链、异形和带嘴等类型，袋形的多样化有助于商品的差异化销售。自立袋具有诸多优点：涉及领域范围广，食品日化等领域的应用不断扩展；包装物料种类多，颗粒、粉末、液体等物料形态均适用；保鲜密封效果好，有助于产品长时间有效保存；陈列摆放美观，精美印刷吸引消费者注意力；使用感观便捷，拉链式自立袋可以多次使用，带嘴自立袋倒出食品更为方便。

虽然自立袋包装设备在不断发展，但是在自立袋包装过程仍无法避免缺陷出现。包装缺陷的种类总体上分为 3 种情况。①自立袋泄漏：热封强度作为影响自立袋渗漏的重要因素，封装过程中应当把控前后加热棒的热封压力和时间，进而控制热封强度，使得热封强度和包装材料相匹配；②袋形不均匀：袋形均匀程度受到加热棒的温度、压力以及挤压时间的影响。③对称性差：自立袋左右封装宽度一致性受到机械牵引和光电传感误差等因素的影响。在具体的缺陷

表现形式上，自立袋泄漏和袋形不均匀常表现为袋形的褶皱，对称性差表现为封装未对齐。此外，在自立袋封装过程中还可能因机械结构的毛刺以及牵引压力过大产生划痕和戳点等缺陷。这些缺陷对自立袋产品的质量以及消费者的使用感观都会造成严重的不良影响。

随着硬件性能的提升与物联网成功案例的应用，利用计算机视觉和深度学习方法相结合的技术进行缺陷检测具有巨大优势<sup>[4]</sup>。深度学习模型具有优异的学习能力，通过学习大量的数据，自动学习缺陷的特征，并对缺陷进行准确检测；深度学习模型具有自适应性，能够自动调整模型参数和结构，在计算机视觉领域能适应不同的数据和任务；深度学习模型具有较好的鲁棒性，能够处理噪声和变形等干扰，从而提高缺陷检测的稳定性和可靠性；深度学习模型也具有良好的可扩展性，能适配不同的检测任务和应用场景；此外，深度学习模型能自动完成特征提取、特征选择和分类等任务，从而减少人工干预并提高检测效率。综上所述，利用深度学习研究方法研究自立袋封装缺陷检测符合检测技术发展的潮流，对实现自立袋封装缺陷高效检测具有重要意义。

## 1.2 国内外研究现状

自立袋封装缺陷的检测鲜有研究人员涉及，但是缺陷异常检测的方法却是可以相互借鉴的。近些年来，国内外专家学者对缺陷异常检测进行了大量的研究与分析，根据研究发展的时间线，将其梳理划分为两个阶段：基于机器视觉的缺陷检测和基于深度学习的缺陷检测。

### 1.2.1 基于机器学习的缺陷检测研究现状

基于机器视觉的缺陷检测方法以特征提取、目标分类为主<sup>[5]</sup>。本节从特征提取与分类算法两个方面对基于机器视觉的缺陷检测方法的研究近况进行归纳。

#### 1) 基于特征提取的机器视觉缺陷检测

纹理特征通常利用某点像素及其邻域灰度反映检测目标的表面结构和组织排列，具有旋转不变性，且对噪声扰动免疫效果良好，成为缺陷区域特征获取的主要参考特征。文献[6]探究融合灰度共生矩阵(Gray-level Co-occurrence Matrix, GLCM)和局部二值模式(Local Binary Pattern, LBP)的缺陷检测方法，利用该方法检测经阈值分割提取的图像感兴趣区域纹理信息，并将特征矩阵传入支持向量机(Support Vector Machine, SVM)分类器实现缺陷类别检测；文献[7]探

讨一种基于改进 Canny 算子和拓扑滤波的方式,抑制图像背景噪声的同时增强图像对比度,使缺陷轮廓的提取更便捷;文献[8]采用融合 Gabor 滤波和 Sobel 算子的特征提取方式,利用像素加权平均整合缺陷特征,经阈值分割后利用形态学技术检测撕裂缺陷;文献[9]采用马尔可夫随机场提取像素特征,整合基于边缘和随机区域合并的似然函数,以能量最小原则实现冲压件表面缺陷分割;文献[10]提出一种基于网格分割和模板统计的缺陷检测方法,通过融合多种特征提取方式,获取特征平均值,用于含有重复纹理单元的丝织物缺陷检测。

此外,颜色和形状也是进行缺陷检测的主要参考特征。作为描述图像的基本要素,颜色特征具有平移不变等特性。文献[11]采用单高斯模型(Single Gauss Model, SGM)二维颜色分割的缺陷检测方法,对多个维度的颜色特征进行提取,通过有向无环图支持向量机(Directed Acyclic Graph Support Vector Machines, DAG-SVMS)进行缺陷分类;文献[12]在 HSV 颜色空间下,通过统计方法确定不同空间的判别阈值,进行胡萝卜橙色区域与青头区域的检测。形状特征通常以边界轮廓和内部区域特征为主。文献[13]采用双边滤波和拉氏锐化增强进行图像预处理,在 Canny 算子中引入灰度统计方差确定自适应阈值,提升齿轮缺陷边缘信息的提取能力;文献[14]在研究图像校正和图像增强的基础上,采用改进后的阈值分割法分割线性裂纹。

## 2) 基于分类方法的机器视觉缺陷检测

经特征提取后,机器视觉方法需将特征送入分类器训练,以准确识别每个检测目标的缺陷类别。机器学习常用的分类算法有 SVM、K 近邻(K-Nearest Neighbors, KNN)以及 K-means 聚类等。文献[15]通过结合 SVM 和决策树方法,实现非铁磁性材料裂纹的分类;文献[16]研究一种基于 KNN 方法的涂胶质量分类方法,利用坐标转换和采用点的提取,计算中心点偏移量以及采样点胶条宽度,比对涂胶质量标准后获取正确检测结果;文献[17]采集玻璃瓶分模线图像,并针对分模线背景复杂性采用时频域变换算法,结合 K-means 聚类与傅里叶变换对玻璃瓶分模线质量进行检测。

### 1.2.2 基于深度学习的缺陷检测研究现状

相比于机器学习,基于深度学习的缺陷检测方法不需要详细区分检测对象的纹理、颜色以及边缘轮廓等特征,免去了人工设计特征提取规则的过程,同时检测更为精准。本节从深度学习数据是否输入给定标签角度出发,分别对监

督学习和无监督学习的缺陷检测研究近况进行阐述。

### 1) 有监督学习

KNN 与 SVM 作为机器学习策略中监督学习的范式,并不归属深度学习策略,而 LeNet 模型被认为是深度学习发展史上的第一个监督学习模型<sup>[18]</sup>。伴随硬件计算能力不断提升,深度学习模型层出不穷,基于深度学习的缺陷检测也应运而生,早期发展的是监督模式下深度学习的分类模型。文献[19]在利用结构优化和剪枝操作轻量化 VGG16 网络的同时,实现产品表面缺陷检测准确率的提高;文献[20]采用多渠道特征提取优化 VGG19 网络,利用多尺度特征对工业目标缺陷进行检测;文献[21]实验探究 GoogLeNet 与 AlexNet、LeNet-5 以及 SVM 等浅层网络对苹果缺陷检测的效果;文献[22]采用 Inception-V3 网络提取的缺陷特征,进行逻辑回归分类器的训练,实现小型金属圆柱壳缺陷的检测。

随后,针对分类任务无法锁定缺陷区域的问题,以预测框的形式进行检测的深度学习目标检测算法问世。文献[23]探讨了多层特征融合网络对区域建议网络生成锚框分类和回归的影响;文献[24]以 ResNet-101 为主干网络,引入特征金字塔网络(Feature Pyramid Network, FPN)和 ROIs Align,实现偏光片表面的缺陷检测;文献[25]采用 MobileNetV2 网络对 RetinaNet 模型进行轻量化,并引入 FPN 和聚焦损失缓解缺陷尺度多变和类别不平衡的问题,实现多尺度缺陷元件的检测;文献[26]通过优化 MobileNet-SSD 网络的特征层、匹配锚框选取策略,实现小样本的车辆漆面缺陷目标检测;文献[27]将 Inception-V3 分类模型与 YOLOv3 目标检测算法相结合,完成包装缺陷的实时检测;文献[28]通过在 YOLOv5 目标检测算法的基础上融合可行变卷积网络(Deformable Convolutional Network, DCN),同时引入深度可分离卷积(Depth-wise Separable Convolution, DSConv)与高效通道注意力(Efficient Channel Attention, ECA),研究一种轻量化的带钢表面缺陷检测模型 DCN-YOLO。

为了更加细化缺陷检测结果,实现像素级别的分割任务也成为缺陷检测研究的重点之一。文献[29]提出一种基于 Mask R-CNN 的轮毂缺陷分割方法,通过卷积神经网络识别提取缺陷特征并分割不同类型的缺陷;文献[30]以 U-Net 分割网络为基底模型,采用密连接残差块的 ResNet50 进行编码,以增强低层特征的延展性,利用加权融合 Dice 损失和交叉熵损失(Binary Cross Entropy loss, BCEloss)减缓学习样本不均衡问题,提升了钢铁表面缺陷分割的精度;文献[31]

提出一种用于织物疵点缺陷分割的卷积神经网络, 显著减轻人工标注成本; 文献[32]研究基于 Deeplabv3+模型的排水管道缺陷语义分割方法, 实现对多种缺陷的检测、定位和几何分割; 文献[33]提出一种双通道编码-融合解码(Dual-Channel Encoding-Fusion Decoding, DCE-FD)网络, 在利用深浅双通道结构编码网络模块(Encoding Network Module based on a Deep and Shallow Dual-Channel Structure, ENM-DSDCS)获取多尺度细节特征的基础上, 设计了注意力双向引导融合(Decoding Network Module based on Attention-based Bidirectional Guidance Fusion, DNM-ABGF)的解码模块, 更加关注模糊边界等细节, 提高分割的准确性和完整性, 实现铸件缺陷区域的细致分割。

## 2) 无监督学习

现阶段无监督缺陷检测网络模型繁多、结构各异, 但也具有以下共性: ①实现无标签数据的训练; ②自动学习输入数据的模式和规律; ③用于训练和检测输入数据的不同或异常模式; ④较监督学习模型而言, 具有更强的适应性或泛化性, 对未知异常模式检测效果良好; ⑤同监督学习模型一样, 需要利用各种降维或特征提取方法, 获取输入数据关键信息。因此, 无监督学习的缺陷检测优势明显, 且已经成为缺陷和异常检测研究所采用的主要策略。

针对无监督缺陷检测模型网络结构的分类, 研究者通常将具有代表性的网络模型分为基于重构方法和基于嵌入方法两种结构。基于重构方法的无监督异常缺陷检测常采用变分自编码器(Variational Autoencoder, VAE)<sup>[34]</sup>、生成对抗网络(Generative Adversarial Network, GAN)<sup>[35]</sup>等生成模型。而基于嵌入方法的无监督异常缺陷检测能进一步划分为归一化流(Normalizing Flow, NF)<sup>[36]</sup>、师生(Teacher-Student, TS)<sup>[37]</sup>、单类分类(One-Class Classification, OCC)<sup>[38]</sup>和存储库(Memory Bank, MB)<sup>[39]</sup>等模型构建方法。

基于重构方法缺陷检测的基本思想是生成模型重建正常区域, 保留异常或缺陷区域, 通过对比正常样本与重建生成样本, 得到像素级的重建误差, 实现高效的缺陷检测和定位。因此, 基于重构方法的缺陷检测被广泛应用, 并衍生出诸多变体。文献[40]在采用 YOLOv5 模型预测并截取目标的基础上, 以 Skip-GANomaly 网络为基底模型, 提出一种基于分块重构的缺陷检测算法; 文献[41]探究一种引入改进 GAN 和 MobileNetV3 的缺陷分类方法, 缓解数据集样本少、检测精度低的问题; 文献[42]结合两个周期一致的对抗网络模型自动合成和标

注图像中的缺陷像素，然后采用 U-Net 语义分割网络学习上述合成和标注的图像，以进行缺陷检测；文献[43]通过在 VAE 中融入空间注意力获取正样本特征，利用重构图像与原图像的差值特征图，突出缺陷区域；文献[44]利用卷积去噪自编码器构建含有多尺度特征网络，融合多层分割结果，实现了多层背景纹理图像的重构和缺陷分割准确度的提升；文献[45]基于重构方法，将表面异常检测视为一种判别问题，提出一个判别训练的异常嵌入模型。然而，基于重构方法的缺陷检测常受限于卷积神经网络超强的特征提取能力，忽略较小异常缺陷，重建图像往往趋向于输入的正样本图像，从而造成检测误差。

当前，基于嵌入方法的缺陷检测成为无监督缺陷检测研究的主流。NF 模型常通过拟合高斯分布与狄拉克分布，并利用分布乘积的似然概率对网络进行更新，常见模型如 CS-Flow<sup>[46]</sup>、SNF<sup>[47]</sup>等，然而工业制造中产生的图像可能不符合高斯分布；TS 模型将检测良好和泛化优异的教师模型转移到更轻量化的学生模型中，如 STPM 模型<sup>[48]</sup>、STUDD<sup>[49]</sup>模型等，但是该方法过度依赖教师模型，可能弱化整个模型的泛化能力；OCC 模型通过学习正常图像模式构造单分类器，利用裁剪粘贴构造负样本参与模型训练，但是多样的负样本形式，导致仅通过裁剪粘贴的方式无法真实模拟负样本。其中 CutPaste 模型以丝织纹理图像为噪声，并叠加正样本用于合成异常图像或裁剪图像补丁，同时随机粘贴生成异常模拟图像作为模型输入，完成缺陷检测算法学习<sup>[50]</sup>。MB 模型将正样本特征暂存于存储库，通过计算测试样本与存储库中最邻近的特征差异，对缺陷进行检测，如 PaDiM<sup>[51]</sup>、PatchCore<sup>[52]</sup>、MemSeg<sup>[53]</sup>等模型，其检测效果也常优于重构模型。

综上所述，与机器视觉缺陷检测相比，基于深度学习的缺陷检测解决了手动设计特征规则的问题，对较为复杂的缺陷检测任务，准确性和鲁棒性更高；与基于监督学习缺陷检测相比，基于无监督学习范式的缺陷检测，成为发展趋势，少样本和无样本条件下的缺陷异常检测研究越来越多。

### 1.3 主要研究内容

为提升自立袋包装良品率，适应长距离运输，以增强消费者使用感观，本文对自立袋封装缺陷检测进行研究。为顺利推进课题研究，解决缺陷样本多样化、难获取等问题的限制，从上海欧朔智能包装设备公司实地采集自立袋封口

缺陷图像，作为实验研究的基础。本文主要围绕自立袋封装缺陷检测方法以及检测系统设计开展研究，分别从监督学习和无监督学习角度，建立缺陷检测模型，研究高效检测算法，论文的整体结构和框架如图 1-1 所示：

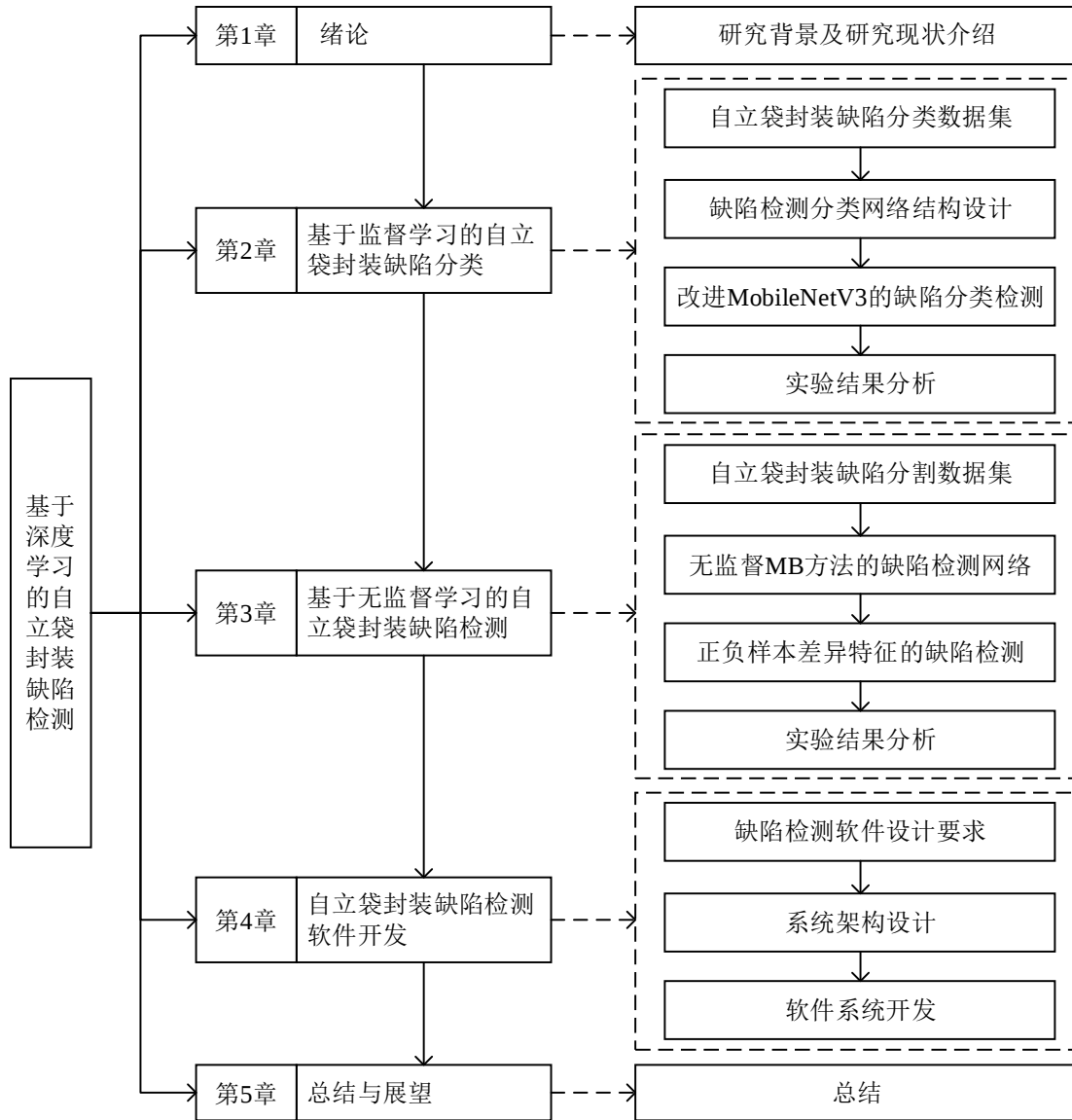


图 1-1 研究内容框架图

第 1 章：阐述了论文的选题背景及意义，对比分析了机器视觉与深度学习缺陷检测的研究现状，为自立袋封装缺陷检测研究指明方向。

第 2 章：探讨了一种改进 MobileNetV3 网络的监督学习自立袋封装缺陷分类模型。在工厂实地收集自立袋封装缺陷样本的基础上，通过筛检和数据增强等步骤，构建封装缺陷分类数据集；从深度卷积位置、GELU 激活函数以及注意力机制等角度优化 MobileNetV3 网络的 Bneck 块；在自制数据集上进行改进模型的消融以及对比实验，验证了改进模型的分类效果。

第 3 章：研究了一种基于自编码器和 U-Net 结构的 MB 缺陷检测方法。在利用 EISeg 软件构建自立袋封装分割数据集 Doypack-Defects 的基础上，阐明基于 MB 方法 U-Net 网络的缺陷检测结构；通过负样本合成模块、特征存储模块、注意力特征融合模块以及预测模块对缺陷检测模型进行搭建，形成一个以正样本模拟负样本进行训练，正负样本进行测试的差异特征无监督缺陷检测方法；在 Doypack-Defects 数据集上进行消融实验，在公共数据集 MVTec AD 上进行对比和泛化实验，验证了算法的有效性与可靠性。

第 4 章：设计了自立袋封装缺陷检测系统。在明确系统设计要求的基础上，以 DS-180DS 型号自立袋包装机为蓝本，研究自立袋包装设备生产流程，进而设计缺陷检测模块、自立袋缺陷样本剔除模块以及缺陷检测系统的架构；根据缺陷检测软件工作流程，在 PyCharm 集成开发环境中，利用 PyQt5 库和扩展工具 QTDesigner、PyUIC、PyRIC 设计缺陷检测软件登录与检测界面。

第 5 章：对本文的主要研究内容进行概括总结，在此基础上进一步分析下一步工作方向。

## 1.4 本章小结

本章首先对自立袋封装缺陷检测课题的研究背景以及现实意义进行阐述；然后从机器视觉、深度学习两个方向入手，对国内外缺陷检测的研究现状和技术方法展开叙述；最后对全文的研究内容框架进行概括说明，简要阐述了各个章节的主要内容，为后续的详细研究做铺垫。

## 第2章 基于监督学习的自立袋封装缺陷分类检测

应用深度学习方法进行分类检测，是监督学习图像检测的基石，也是推动各种无监督、弱监督、自监督检测发展的基础，目标检测、语义分割与实例分割任务均是在此基础上发展而来。本章在构建和扩展自立袋封装分类数据集的基础上，论述监督学习分类任务的经典模型，提出一种改进 MobileNetV3 的分类网络，探究准确率更高、适应性更强的轻量化缺陷检测模型。

### 2.1 构建自立袋封装缺陷检测分类数据集

由于现阶段涉及包装缺陷的公开数据集不完整且数量少，自立袋封装缺陷样本在公开资料中更是难以获取。因此，为探究监督学习策略下自立袋封装缺陷分类模型检测效果，本节采用华为 P20 手机 2000 万像素的徕卡双镜头，在避光条件下采集 888 张多类别缺陷图像，图像分辨率 2976×2976，每张图片中至少包含一种缺陷类别。

#### 2.1.1 自立袋封装图像采集与分类

自立袋的封装依次经过底封、竖封和上封流程，本文在实际包装机器生产线中收集自立袋上封口缺陷图像，并选取上封“凸”形和“M”形封口的异形自立袋作为封装缺陷图像的来源，“凸”形和“M”形封口的无缺陷样本如图 2-1 所示。不同封口形式形状不同，同一封口形式，封装样式也存在差异，如背景颜色、封装压印纹理、有无压印生产日期等。



图 2-1 自立袋“凸”形封口与“M”形封口样本

由于人工操作不当或包装设备故障，物料使用自立袋封装时，自立袋封口难免出现一些异常，上封“凸”形和“M”形封口的常见缺陷类型如图 2-2 所示，主要包括：“凸”形封口褶皱、未对齐、“M”形封口褶皱、严重褶皱、未对齐等。褶皱和未对齐缺陷产生的原因如下：①原始封装材料质量不佳，存在孔洞、褶皱或厚度不均匀等问题；②包装环境湿度过高，增加封装材料的形变可能性；③包装机器牵引机构运动速度过慢或过快，与封口操作运动不匹配；④包装流程外的其它张力或挤压；⑤自立袋包装折叠方式等设计不合理，存在原始设计缺陷等。



(a) “凸”形封口褶皱



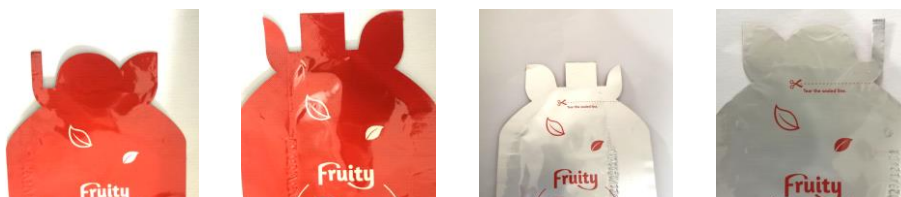
(b) “凸”形封未对齐



(c) “M”形封口褶皱



(d) “M”形封口严重褶皱



(e) “M”形封口未对齐

图 2-2 自立袋“凸”形封口与“M”形封口缺陷样本

### 2.1.2 自立袋封装缺陷检测图像数据增强

从拍摄获取的原始缺陷图像中，剔除含有模糊、光斑和相似性高的冗余图像，形成由 325 张原始未处理图像组成的数据集，并按照 7:2:1 比例分为训练集、验证集和测试集。其中，数据集的图像共包含 7 个类别：凸形正样本、凸形褶皱、凸形未对齐、M 形正样本、M 形褶皱、M 形严重褶皱和 M 形未对齐。为在保留样本细致特征的同时加速图像预处理速度，本节对数据集的图像进行线性插值 2 倍压缩，并分别对各类别正常和缺陷图像采用以下数据增强方式：45°-270°旋转、垂直翻转、明暗调节、对比度调节、水平翻转等。经数据增强操作后，共获取 3575 张图像，包含训练集 2502 张、验证集 715 张、以及测试集 358 张。具体各类别数据如表 2-1 所示。

表 2-1 自制自立袋封装缺陷分类数据集数据汇总

| 封口类型 | 类别   | 原始/张 | 待处理/张 | 训练集/张 | 验证集/张 | 测试集/张 |
|------|------|------|-------|-------|-------|-------|
| 凸形   | 正常   | 229  | 50    | 385   | 110   | 55    |
|      | 褶皱   | 81   | 25    | 192   | 55    | 28    |
|      | 未对齐  | 152  | 70    | 539   | 154   | 77    |
| M 形  | 正常   | 55   | 25    | 192   | 55    | 28    |
|      | 褶皱   | 114  | 50    | 385   | 110   | 55    |
|      | 严重褶皱 | 90   | 35    | 270   | 77    | 38    |
|      | 未对齐  | 167  | 70    | 539   | 154   | 77    |
| 总计   |      | 888  | 325   | 2502  | 715   | 358   |

## 2.2 缺陷检测分类网络结构设计

构建自制分类数据集后，整合计算资源，实现深度学习分类模型的训练与图像特征的自动提取，并利用训练后的模型权重对新图像缺陷进行分类。基于深度神经网络的分类模型，根据 Stem 层、主特征提取层、分类层将其分为三个部分。图 2-3 为分类模型检测方法流程。

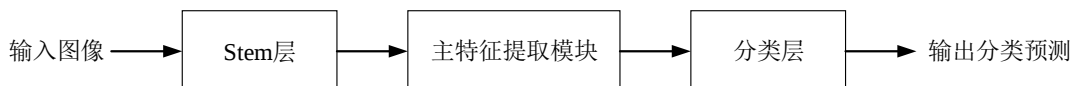


图 2-3 分类模型检测方法流程图

**Stem 层：**对输入分类模型的图像进行首次卷积或池化等下采样的操作。作为一种特征预提取层，Stem 层为卷积神经网络的输入数据进行预处理，以便适

应后续主特征提取模块各种类型卷积块的特征提取。

主特征提取模块：不同的卷积类型以及组成形式，衍生出不同的卷积块。而通过卷积块的叠加重复等操作，模型实现输入图像的特征提取。卷积块是分类模型结构设计最为重要的部分，卷积块设计的优劣，直接影响模型分类效果。

分类层：由全连接、激活函数等构成的分类层，通过卷积和池化操作实现分类器构建。在实际模型的构建中，可以使用 $1\times 1$ 卷积替换全连接层，并使用不同的激活函数，比如 ReLu、Softmax、Sigmoid 等；同时，为防止模型的过拟合，可以选择在分类层中添加 dropout 层。

根据分类模型检测方法的分析，本节将主要对 ResNet<sup>[54]</sup>、MobileNet<sup>[55]</sup>、ShuffleNet<sup>[56]</sup>以及 EfficientNet<sup>[57]</sup>的卷积块进行阐述。

### 2.2.1 ResNet 缺陷分类网络结构

基于深度卷积神经网络方法的各种模型不断涌现，如 VGG、GoogLeNet 等，网络越来越深、模型越来越大。然而，模型深度并非越深越好，模型网络层数的不断提升，增大了模型训练梯度消失和爆炸的可能性，进而使模型提前终止训练。而且，在模型准确率逐渐收敛、训练损失趋于平缓后，进一步增加网络深度会导致损失值反向增大，产生神经网络的退化现象。ResNet 通过引入残差模块，解决梯度消失和损失退化问题，使得模型可以设计得更深。残差模块结构如图 2-4 所示。

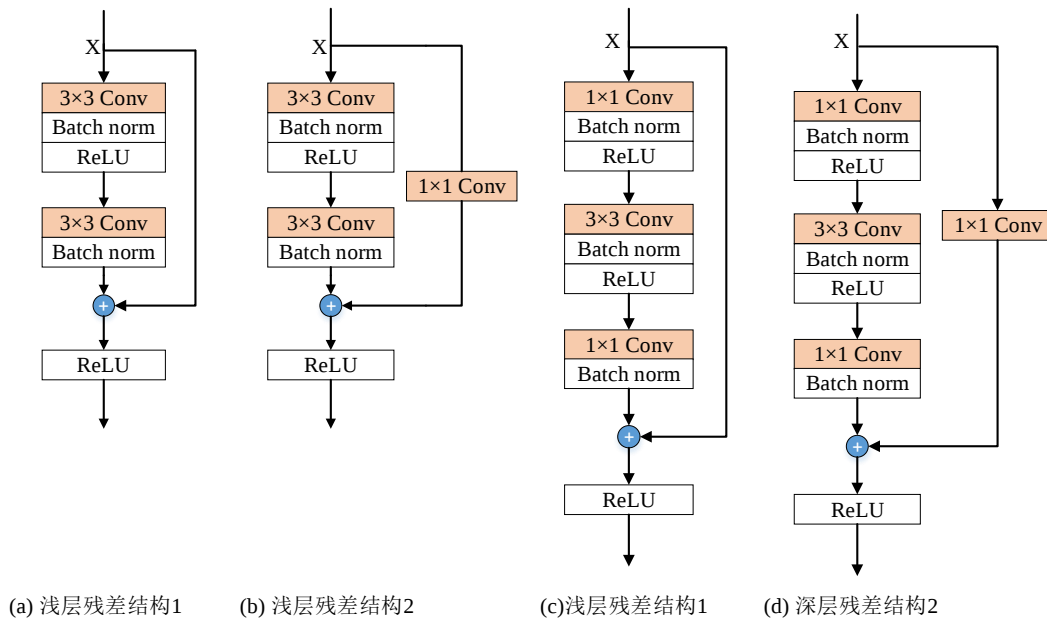


图 2-4 残差模块结构图

ResNet 网络采用堆叠基础残差模块的方式加深网络，常见的有 18 层、34 层、50 层、101 层以及 152 层网络结构。18 层和 34 层模型采用图 2-4 中(a)、(b)所示残差模块，50 层及以上模型采用图 2-4 中(c)、(d)所示残差模块。残差模块利用跳跃连接的思想，将浅层特征添加至深层卷积，有利于模型跨层级优化，缓解训练中的退化问题。

### 2.2.2 MobileNet 缺陷分类网络结构

MobileNet 作为一种轻量级的卷积神经网络，旨在为移动和嵌入式视觉应用提供高效的计算<sup>[55]</sup>。而 MobileNetV2 和 MobileNetV3 分别在 MobileNet 轻量化思想的基础上优化模型结构，进一步提升了模型检测精度等性能<sup>[58]</sup>。

#### 1) 深度可分离卷积(Depthwise Separable Convolution, DSConv)

传统卷积的卷积核通道数与输入特征矩阵通道数相同，且卷积输出特征矩阵通道与卷积核的个数相等。而深度卷积(Depthwise Convolution, DWConv)的卷积核通道始终为 1，且卷积输入特征矩阵通道数与卷积核个数和卷积输出矩阵通道数一致。DSConv 由 DWConv 和逐点卷积(Pointwise Convolution, PWConv)组成<sup>[58]</sup>，PWConv 等同于普通 $1 \times 1$ 卷积，如图 2-5 所示。DWConv 对每个输入的通道单独使用卷积核提取通道特征，然后将得到的各通道特征运用逐点卷积，生成具有卷积核个数的特征图，实现对输出通道数的调整。

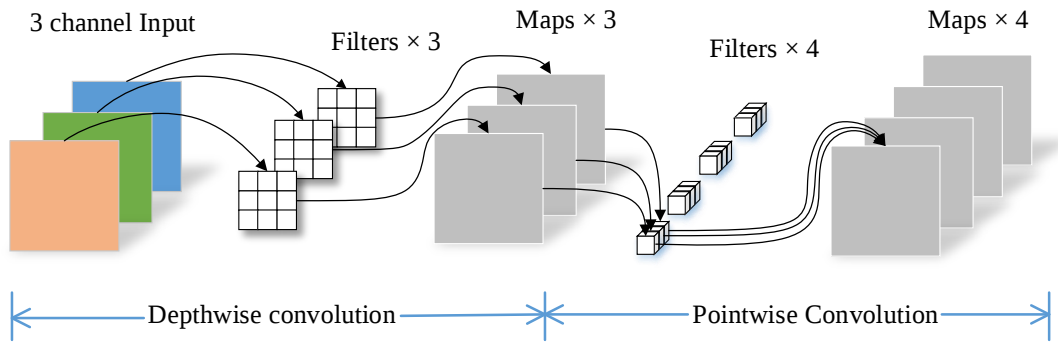


图 2-5 深度可分离卷积示意图

与传统标准卷积相比，DSConv 在计算量方面更具优势。假设特征图的高宽一致，则 DSConv 计算量与标准卷积计算量之比如下式所示：

$$\frac{D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F}{D_K \times D_K \times M \times N \times D_F \times D_F} = \frac{1}{N} + \frac{1}{D_K^2} \quad (2.1)$$

其中， $K$ 、 $F$  分别表示卷积核和特征图， $N$  为卷积核个数， $M$  为通道维度， $D$  表示卷积核或特征图的高和宽，具体而言， $D_K$ 、 $D_F$  分别表示卷积核、特征图

的高和宽， $D_F \times D_F \times M$  表示输入特征尺寸大小， $D_K \times D_K \times M \times N$  表示  $M$  个通道下  $N$  个卷积核的计算量。

如果 DSConv 的卷积核大小为  $3 \times 3$ ，则由式(2.1)可得，其计算量约为普通卷积的  $1/9$ 。因此深度可分离卷积可以大大提高算法训练和推理的速度。

## 2) 倒残差结构(Inverted Residuals)

倒残差结构的核心思想是对传统的残差进行变更，以适应模型轻量化的要求。传统的残差结构先经过卷积降维再升维，最后与原始输入叠加进行输出；倒残差结构先经过标准卷积进行通道扩增，再经过 DSConv 进行特征提取，并调整通道以降低维度，同时引入非线性变换，最后与原始输入叠加进行输出。两种残差结构区别如图 2-6 所示。

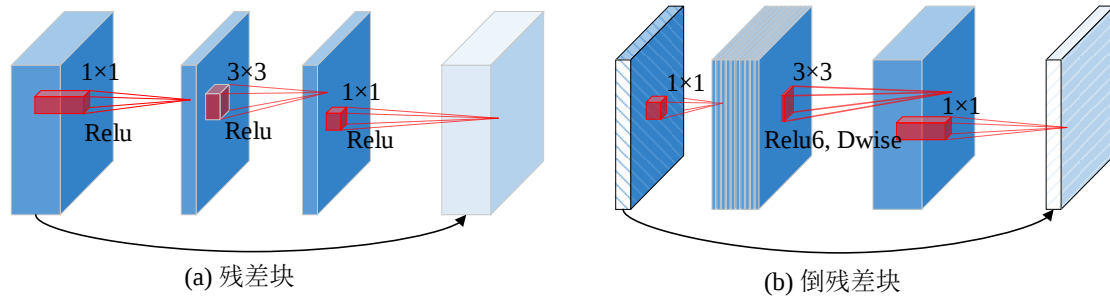


图 2-6 残差块与倒残差块

倒残差结构具有诸多优点：①使用瓶颈深度可分离操作，在减少计算量的同时提高算法计算效率；②引入 ReLU6 非线性激活函数，使模型在低精度计算时也具有较好的鲁棒性；③  $3 \times 3$  标准卷积具有较好的感受野和特征提取能力；④残差有助于深度网络特征信息的传递，缓解梯度消失问题的同时有利于模型泛化性能的提升。

## 3) SE(Squeeze and Excitation)注意力

MobileNetV3 中的倒残差模块在 DWConv 和 PWConv 之间融入 SE 注意力模块<sup>[58]</sup>。SE 注意力工作原理如图 2-7 所示。

首先，利用全局平均池化压缩深度卷积输出特征，并将其转化为全局描述符，捕捉输入特征中每个通道的全局信息；然后，通过 2 个全连接层分别将全局描述符映射为较小的维度和原始通道数，引入 H-sigmoid 激活函数进行归一化，得到每个通道的权重；最后，对通道权重与输入特征图进行乘积操作，得到加权注意力的特征图。

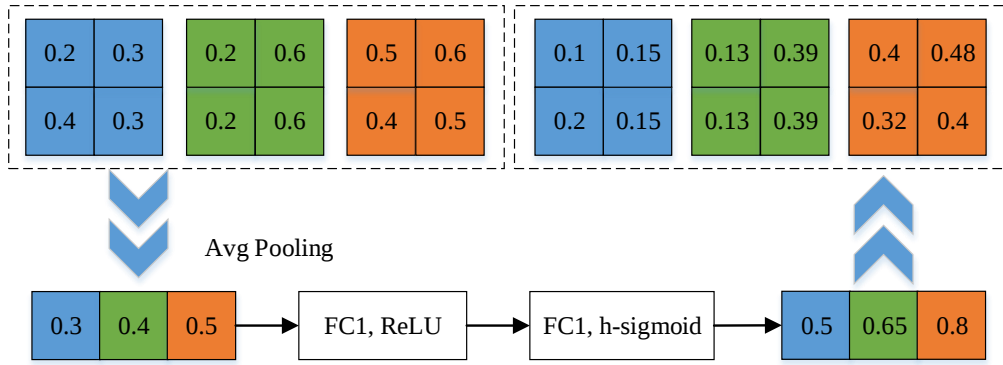


图 2-7 SE 注意力示意图

### 2.2.3 ShuffleNet 缺陷分类网络结构

2017 年 7 月 4 日发布的 ShuffleNet 是一种用于图像分类的卷积神经网络，通过一种通道重排的模块化设计，减少了模型参数量<sup>[56]</sup>。相比 MobileNet，ShuffleNet 在 ImageNet 数据集上取得了更优的性能，在降低计算复杂度和提升检测速度的同时，使模型的准确性也得以提升。在传统的卷积神经网络模型中， $1 \times 1$  的卷积将消耗大量的计算成本，而 ShuffleNet 网络利用逐点分组卷积 (Pointwise Group Convolution, PGConv) 对网络中存在的  $1 \times 1$  的卷积进行分组。此外，ShuffleNet 网络采用通道混洗 (Channel Shuffle, CS) 方法对各个通道的信息进行融合，从而提高模型特征表达能力，解决卷积分组导致的不同通道信息隔离的问题。上述 2 种机制的原理如图 2-8 所示。

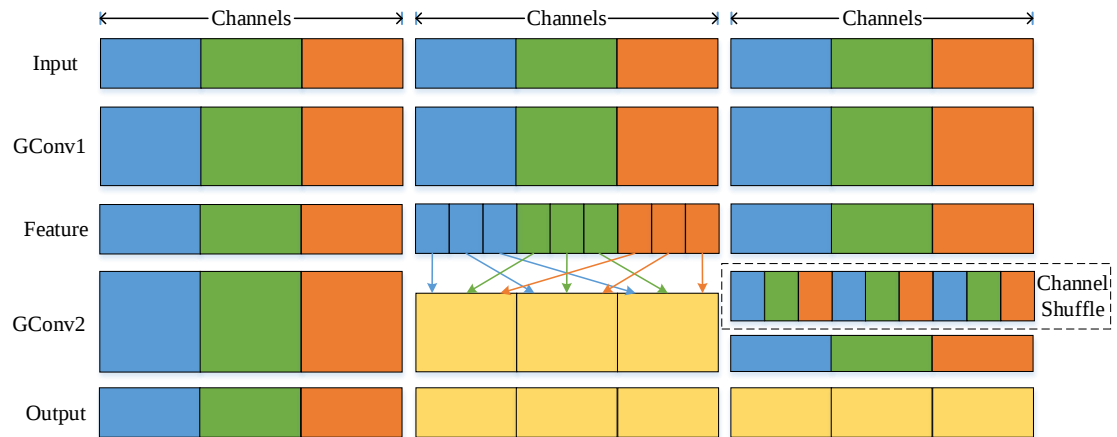


图 2-8 分组卷积及通道混洗原理图

经过通道分组操作，同组中的通道信息存在相似性，对输出特征具有限制和约束性。因此，引入通道交换的思想丰富分组后的信息，解决分组特征不交融的问题，在极大减少模型计算量的同时维持模型检测精度。

2018 年，改进版本 ShuffleNetV2 的提出，在 ShuffleNet 的基础上引入通道分离和逐通道瓶颈结构的设计<sup>[59]</sup>，进一步降低模型参数量、提升检测准确率。

### 2.2.4 EfficientNet 缺陷分类网络结构

EfficientNet 对神经网络模型扩展方法进行研究，利用 NAS(Neural Architecture Search)网络搜寻方法对模型输入图像分辨率、网络深度和宽度的相关参数进行优化设置<sup>[57]</sup>。

模型宽度的扩增，有助于模型获取更为复杂的语义特征，但仅仅增加模型宽度，难以对深层特征进行学习；模型深度的扩展，有助于将复杂特征纳入网络学习，但过深的网络模型不仅增加了计算量，而且导致模型训练难度加大，难以收敛；输入图像分辨率的增大，能为模型提供更为细致的特征进行学习，但同时减缓了模型训练速度，降低模型增益。因此，平衡模型输入图像分辨率、模型宽度和深度至关重要。EfficientNet 网络结构以 MBConv 模块为 Stage 基础模块，该模块受 MobileNetV2 和 MobileNetV3 中倒残差模块思想的启发，构成模型的主特征提取模块，其基本结构如图 2-9 所示。

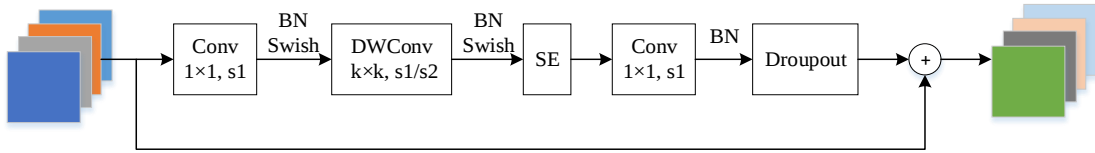


图 2-9 MBConv 结构示意图

BMConv 模块对倒残差模块进行了改进：一方面，采用 Swish 激活函数替换传统的 ReLU 激活函数，在准确率上取得更优的效果；另一方面，引入 SE 注意力机制。BMConv 采用  $1 \times 1$  卷积对输入特征维度进行扩增，扩增倍数取决于卷积核个数；当瓶颈结构维度扩增倍数  $n=1$  时，则不进行  $1 \times 1$  卷积层的运算，以减少模型计算量。BMConv 中的 Shortcut 连接，仅在输入特征矩阵与输出特征矩阵维度信息一致时才存在。

EfficientNet 对输入图像的分辨率、模型深度和宽度三个维度进行复合缩放，其缩放方式如下：

$$N = \odot_{i=1 \dots S} F_i^{L_i}(X_{\langle H_i, W_i, C_i \rangle}) \quad (2.2)$$

其中， $N$  表示分类模型； $\odot$  表示连乘； $X$  表示第  $i$  个 Stage 的输入特征矩阵； $F_i$  为 BMConv 运算操作； $L_i$  为网络深度参数，表示  $F_i$  在第  $i$  个 Stage 中重复的次数； $C_i$  为网络宽度参数，表示输入特征矩阵的通道数； $\langle H_i, W_i, C_i \rangle$  为网络的分辨率参数，表示输入特征矩阵，而  $H_i$ 、 $W_i$  为特征矩阵的高度和宽度。

## 2.3 改进 MobileNetV3 的自立袋封装缺陷分类检测

根据分类模型组成, 分析 MobileNetV3 网络结构: 首先, Stem 层采用一个大小为  $3 \times 3$ , 步距为 2 的标准卷积层, 进行特征预提取; 其次, 主特征提取层采用结合 DSConv 的倒残差和线性瓶颈结构, 扩增特征通道, 细化图像特征, 进一步压缩模型参数量, 平衡检测准确性和检测速度; 与此同时, 在倒残差结构中嵌入 SE 模块, 自适应缩放特征图, 增强模型特征表征能力; 最后, 利用  $7 \times 7$  池化和  $1 \times 1$  卷积实现分类器构建。此外, MobileNetV3 使用 ReLU6 激活函数引入非线性的同时, 采用平滑可微的 Swish 激活函数, 更多地保留输入信息。

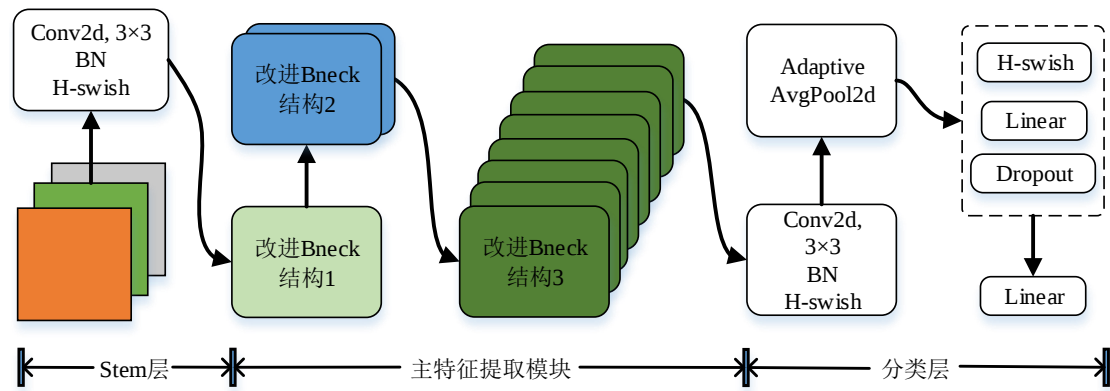


图 2-10 改进 MobileNetV3 网络示意图

MobileNetV3 网络根据模型的深浅分为 MobileNetV3-Large 和 MobileNetV3-Small 两种变体<sup>[58]</sup>, 依据自立袋封装缺陷检测任务低延时高实时性的要求, 本文选取高效的轻量级神经网络架构 MobileNetV3-Small 作为基底模型, 并从网络结构和网络设计策略方面进行改进, 引入 CA 注意力替换 SE 注意力<sup>[60]</sup>, 兼顾通道信息和方向相关的位置信息, 以优化网络 Bneck 结构。改进的网络结构如图 2-10 所示, 根据 Stem 层、主特征提取模块和分类层将网络模型分为三个部分, 其中主特征提取模块的改进 Bneck 结构有 3 种类型, 分别为改进 Bneck 结构 1、改进 Bneck 结构 2 和改进 Bneck 结构 3, 在模型中分别重复 1 次、2 次和 8 次。结构 1、结构 3 和结构 2 的区别在于结构 2 未融合 CA 注意力机制; 结构 1、结构 2 和结构 3 的区别在于结构 3 第一层卷积使用 H-swish 非线性激活函数, 而结构 1 与结构 2 采用 ReLU 激活。此外, 为避免模型过拟合、提高模型泛化能力, 分类层中引入 Dropout 层。改进模型的各层输入维度、卷积操作、扩展通道、输出通道、是否融合注意力、非线性激活类别、卷积步长等信息如表 2-2

所示。

表 2-2 改进 MobileNetV3 网络各层结构组成表

| Input             | Operator         | exp size | out  | CA | NL | s |
|-------------------|------------------|----------|------|----|----|---|
| $224^2 \times 3$  | conv2d, 3×3      | -        | 16   | -  | HS | 2 |
| $112^2 \times 16$ | Bneck1, 3×3      | 16       | 16   | √  | GL | 2 |
| $56^2 \times 16$  | Bneck2, 3×3      | 72       | 24   | -  | GL | 2 |
| $28^2 \times 24$  | Bneck2, 3×3      | 88       | 24   | -  | GL | 1 |
| $28^2 \times 24$  | Bneck3, 5×5      | 96       | 40   | √  | GL | 2 |
| $14^2 \times 40$  | Bneck3, 5×5      | 240      | 40   | √  | GL | 1 |
| $14^2 \times 40$  | Bneck3, 5×5      | 240      | 40   | √  | GL | 1 |
| $14^2 \times 40$  | Bneck3, 5×5      | 120      | 48   | √  | GL | 1 |
| $14^2 \times 48$  | Bneck3, 5×5      | 144      | 48   | √  | GL | 1 |
| $14^2 \times 48$  | Bneck3, 5×5      | 288      | 96   | √  | GL | 2 |
| $7^2 \times 96$   | Bneck3, 5×5      | 576      | 96   | √  | GL | 1 |
| $7^2 \times 96$   | Bneck3, 5×5      | 576      | 96   | √  | GL | 1 |
| $7^2 \times 96$   | conv2d, 1×1      | -        | 576  | √  | HS | 1 |
| $7^2 \times 576$  | pool, 7×7        | -        | -    | -  | -  | 1 |
| $1^2 \times 576$  | conv2d, 1×1, NBN | -        | 1024 | -  | HS | 1 |
| $1^2 \times 1024$ | conv2d, 1×1, NBN | -        | k    | -  | -  | 1 |

注：NL 表示非线性激活，HS 表示 H-swish 激活，GL 表示 GELU 激活<sup>[61]</sup>

### 2.3.1 GELU 激活函数

MobileNetV3 网络采用 ReLU 和 H-swish 激活函数的组合，一同为模型引入非线性，使模型具有逼近表示更为复杂函数关系的能力。然而，ReLU 分段线性激活函数线形不光滑，在零点处为间断不可导，在一定程度上影响模型性能；而且 ReLU 作为确定性非线性函数，在模型搭建时需要附加随机正则以增强模型泛化能力。而 H-swish 激活函数的非线性相对较弱，对复杂特征模式的表达不足，可能会导致模型梯度爆炸问题。为缓解上述两种激活方式的缺陷，本节引入一种具有平滑非线性特性，且能更好避免梯度消失的高斯误差线形单元 (Gaussian error linear unit, GELU)<sup>[61]</sup>，保持输入信息，同时以输入的概率统计量作为提供随机正则性的激活函数，无需额外插入正则项，即可缓解模型训练过拟合问题，优化网络模型的训练过程。GELU 表达形式如下：

$$\text{GELU}(x) = x * P(X \leq x) = x * \Phi(x) \quad (2.3)$$

其中， $\Phi(x)$  表示  $x$  高斯正态分布的积累函数。

$\Phi(x)$  与右尾函数  $Q(x)$  有关系如式(2.4)所示：

$$\Phi(x) = P(X \leq x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{x^2}{2}} dx = 1 - Q(x) \quad (2.4)$$

设误差函数表示为  $erf(x)$ ，则：

$$erf(x) = 1 - 2Q(\sqrt{2}x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-x^2} dx \quad (2.5)$$

因此， $\Phi(x)$  可以化简为：

$$\Phi(x) = \frac{1}{2} \left[ 1 + erf\left(\frac{x}{\sqrt{2}}\right) \right] \quad (2.6)$$

进而， $GELU(x)$  可以近似表示为：

$$GELU(x) = \frac{1}{2} x \left\{ 1 + \tanh \left[ \sqrt{\frac{2}{\pi}} (x + 0.044715x^3) \right] \right\} \quad (2.7)$$

或，

$$GELU(x) = x\sigma(1.702x) \quad (2.8)$$

其中， $\sigma(\cdot)$  表示 Sigmoid 函数。

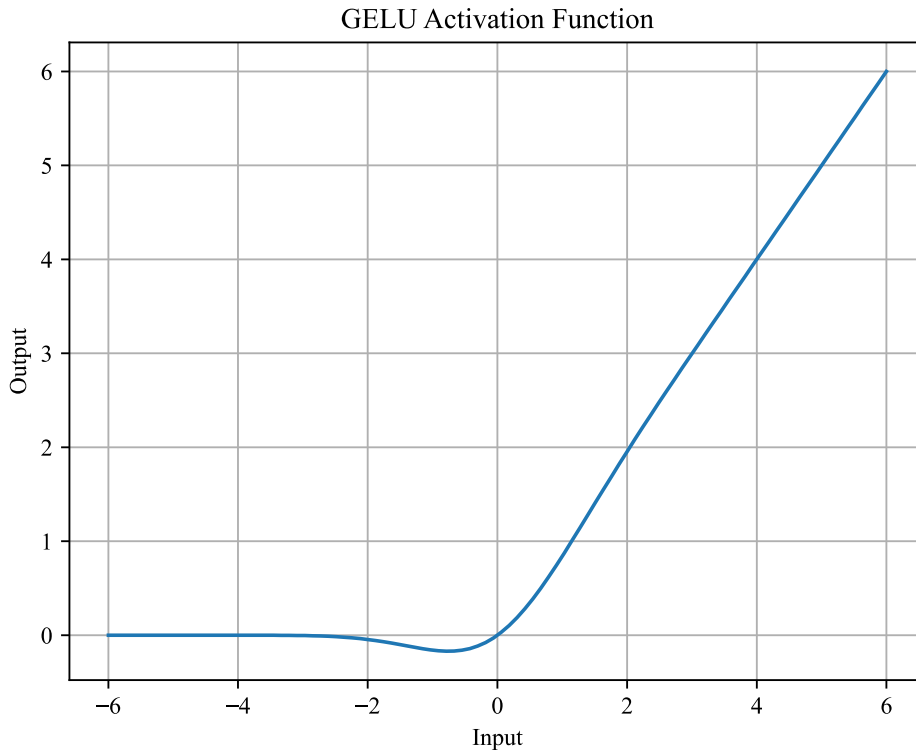


图 2-11 GELU 激活函数曲线

GELU( $x$ ) 激活函数曲线如图 2-11 所示，当输入值  $x > 0$  且较大时，GELU( $x$ ) 与 ReLU 激活函数相似，输出为线性；当输入值为  $x < 0$  且较小时，GELU( $x$ ) 输出值为 0；当输入值  $x$  接近于 0 时，GELU( $x$ ) 的输出为非线性，连续性良好。

GELU 激活函数与 ReLU、H-swish 激活函数一同作用于改进的 MobileNetV3 模型，其中 ReLU 或 H-swish 与 GELU 激活函数搭配一同作用于主特征提取部分 Bneck 结构，而 H-swish 激活函数还作用于模型 Stem 层与分类层。

### 2.3.2 CA 注意力机制

MobileNetV3 网络使用 SE 注意力机制对 Bneck 结构特征提取操作进行特征融合，然而 SE 注意力仅赋值通道信息权重，对位置信息关注不足，难以实现单一通道不同位置信息的融合。为此，本节使用 CA 注意力机制替换 SE 注意力，提升模型对空间特征的关注度，优化分类检测精度和模型大小。

CA 注意力机制从水平和竖直两个方向对输入特征进行全局池化，有助于模型对特征行和列的全局进行感知，更好地捕捉细节特征。CA 注意力机制在强化位置信息作用的同时削弱干扰信息的影响，利用确定的位置信息对特征图之间各个通道的关系和长期依赖性进行编码，有助于缺陷检测位置特征信息的提取。CA 注意力机制结构如图 2-12 所示。

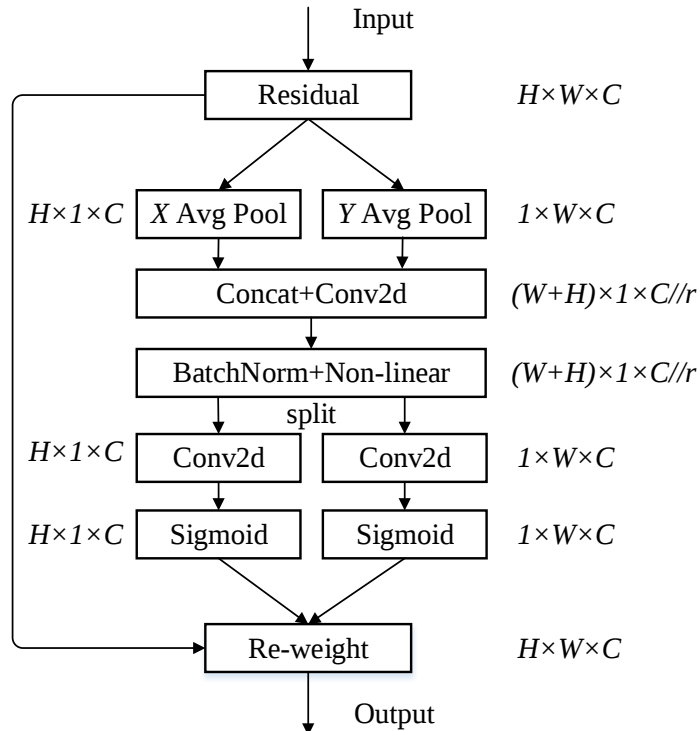


图 2-12 CA 注意力机制结构图

CA 注意力模块的输入输出张量大小一致。首先，将各通道特征沿着水平  $X$  和竖直  $Y$  两个方向平均池化，在  $X$  方向的平均池化对特征图每行求均值，拼接每行所得均值获取水平方向注意力向量，如图 2-13 中图(a)所示；在  $Y$  方向的平均池化对输入特征每列求均值，拼接每列所得均值获取竖直方向注意力向量，如图 2-13 中图(b)所示。

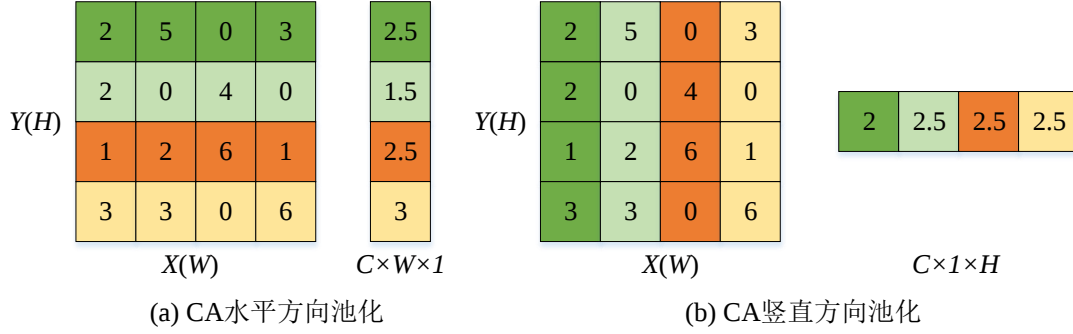


图 2-13 CA 注意力机制水平与竖直池化示意图

$X$  和  $Y$  方向的池化过程，如公式(2.9)、(2.10)所示：

$$Z_c^X(h) = \frac{1}{W} \sum_{0 \leq i < W} x_c(h, i) \quad (2.9)$$

$$Z_c^Y(w) = \frac{1}{H} \sum_{0 \leq i < H} x_c(j, w) \quad (2.10)$$

其中， $Z_c^X(h)$ 、 $Z_c^Y(w)$ 表示经两个不同方向池化生成的特征； $W$ 、 $H$ 分别表示特征图的宽度和高度，即  $X$  和  $Y$  方向像素点的个数； $h$ 表示特征图某行高度， $i$ 表示特征图某点宽度， $w$ 表示特征图某列宽度， $j$ 表示特征图某点高度， $c$ 为特征图某通道， $x$ 为输入特征， $x_c$ 为输入特征某通道的特征图。

经  $X$  和  $Y$  方向上的平均池化，生成大小为  $H \times 1 \times C$ 、 $1 \times W \times C$  尺寸的特征向量组，将  $1 \times W \times C$  转置并与  $H \times 1 \times C$  进行拼接，使用  $1 \times 1$  卷积和激活操作对通道进行  $r$  倍压缩( $r$ 为缩减系数，常取 16)，如式(2.11)所示。

$$f = \delta(F_1([Z^X, Z^Y])) \quad (2.11)$$

其中， $f$ 为经过转置、拼接和卷积激活操作获取的特征矩阵，其尺寸为  $(H+W) \times 1 \times C // r$ ； $Z^X$ 、 $Z^Y$ 分别表示所有通道水平、竖直平均池化所得特征向量组； $F_1$ 表示卷积维度压缩操作； $\delta$ 表示批次归一化和 H-swish 激活操作。

然后，在空间维度上，split 操作将特征矩阵分为  $H \times 1 \times C / r$  和  $1 \times W \times C / r$  的特征向量组，利用  $1 \times 1$  卷积进行  $r$  倍升维，再利用 Sigmoid 激活函数获取注意

力权重向量。

$$g^x = \sigma(F_x(f^x)) \quad (2.12)$$

$$g^y = \sigma(F_y(f^y)) \quad (2.13)$$

其中， $f^x$ 、 $f^y$  表示  $x$  与  $y$  方向 split 操作， $F_x$ 、 $F_y$  表示卷积升维， $\sigma$  表示 Sigmoid 激活函数， $g^x$ 、 $g^y$  分别表示  $x$  与  $y$  方向注意力权重向量。

最后，融合  $x$  与  $y$  方向权重向量的 CA 注意力最终输出为：

$$y_c(i, j) = x_c(i, j) \times g_c^x(i) \times g_c^y(j) \quad (2.14)$$

其中， $g_c^x(i)$ 、 $g_c^y(j)$  分别表示通道  $c$  中  $x$  与  $y$  方向注意力权重向量， $x_c(i, j)$  表示通道  $c$  中特征， $y_c(i, j)$  表示通道  $c$  中 CA 注意力输出。

### 2.3.3 改进 Bneck 结构

MobileNetV3 网络的 Bneck 块整合了 DSConv、线形瓶颈倒残差结构和轻量级 SE 注意力机制，根据使用 DSConv 大小是  $3 \times 3$  还是  $5 \times 5$ 、卷积后接激活函数是 ReLU 还是 H-swish、是否使用 SE 注意力等因素，将 Bneck 结构划分为 3 种类型，如图 2-14 所示。

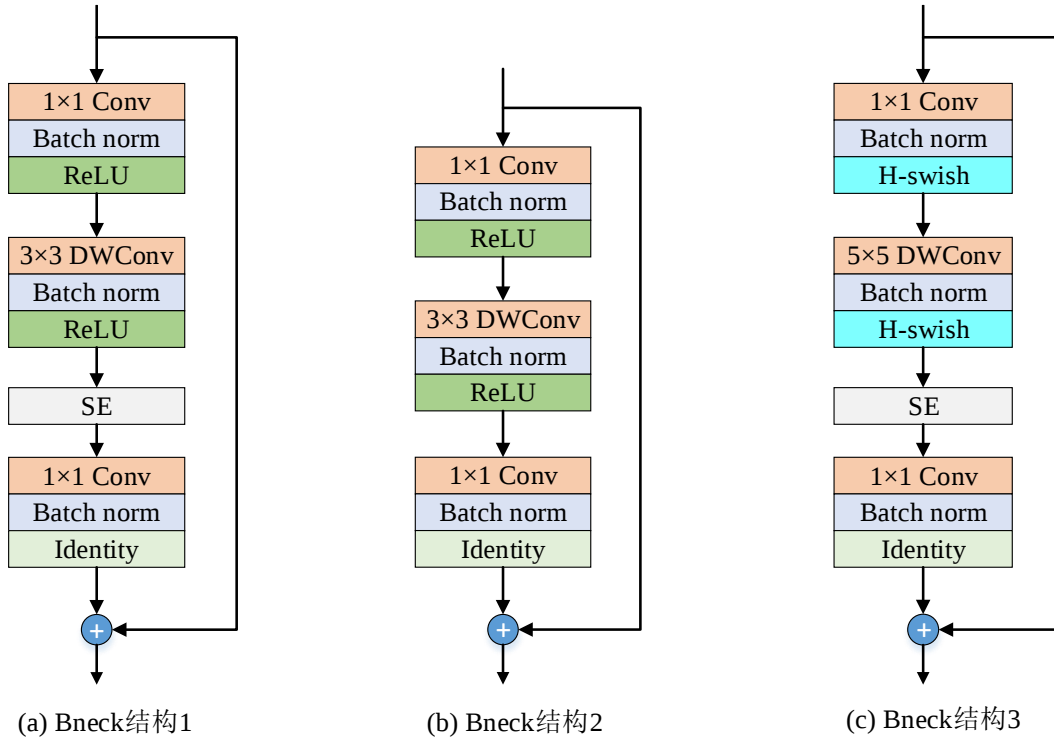


图 2-14 原始 Bneck 块的 3 种结构示意图

原始的 Bneck 结构利用普通  $1 \times 1$  卷积对输入特征进行升维，选取 ReLU 或 H-swish 激活函数；然后采用 DWConv 进行特征提取，再次选取 ReLU 或 H-swish 激活函数，并选择是否采用 SE 注意力机制；最后，使用  $1 \times 1$  卷积对特征

进行降维，与输入特征进行叠加作为 Bneck 块的输出。

改进的 Bneck 结构，受到 Swin Transformer 模型<sup>[62]</sup>与 ConvNeXt 网络<sup>[63]</sup>的启发，将第一层  $1 \times 1$  卷积与第二层的 DWConv 位置调换，使用 DWConv 对输入特征进行升维，利用  $1 \times 1$  卷积进行特征进一步提取，并在第一层的卷积后使用可供选择的激活函数，即 ReLU 或 H-swish。为避免网络训练过程梯度消失和梯度爆炸，在第二层  $1 \times 1$  卷积后统一采用 GELU 激活函数；其次，选择是否使用 CA 注意力对特征图进行位置信息融合；最后，使用  $1 \times 1$  卷积对特征图进行降维，并与输入特征进行叠加作为 Bneck 块的输出。改进的 Bneck 结构与原始 Bneck 结构的 3 种搭配方式一一对应，如图 2-15 所示。

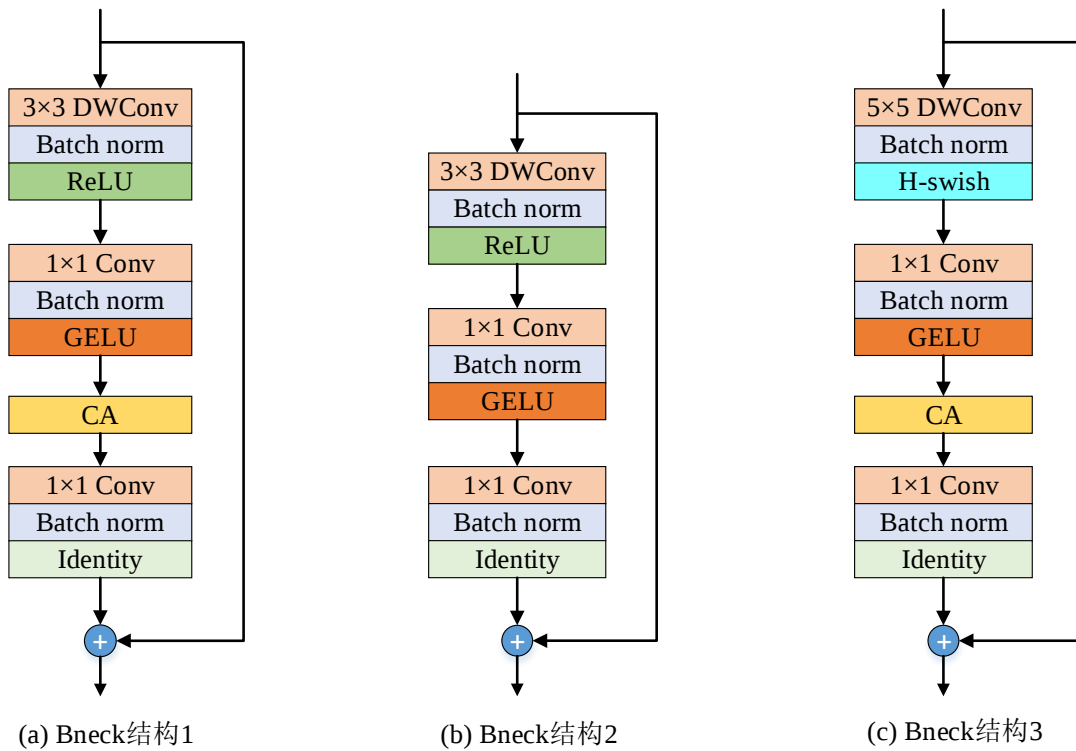


图 2-15 改进 Bneck 块的 3 种结构示意图

改进的 Bneck 结构在自立袋封装缺陷分类检测模型中的组合方式见图(2-10)，分类模型共含有 11 个 Bneck 块，一同构成分类模型的主特征提取模块，按先后顺序分别为一个 Bneck 结构 1、两个 Bneck 结构 2、八个 Bneck 结构 3。

## 2.4 实验结果分析

本节对改进的 MobileNetV3 模型，进行实验验证与分析，共分为三个部分：①介绍自立袋封装分类模型分别采用的评价指标；②从改进 Bneck 结构和引入 CA 注意力两方面进行消融实验，研究改进模型检测精度和模型大小；③将改进

模型与各经典分类模型进行实验对比，探讨改进模型对自立袋封装缺陷分类的检测效果是否具有优越性。

实验采用的硬件配置为 AMD Ryzen 7 5800H (3.20 GHz) 16GB 处理器，NVIDIA GeForce RTX 3070 8GB 显卡；操作系统为 Windows 10，软件平台为 PyCharm；使用深度学习框架为 Pytorch1.10.0，Python 版本为 3.8，在 CUDA11.3 和 cuDNN8.21 上进行加速训练。实验参数的设置：batch size 为 16；训练 epochs 为 500；优化器为 Adam；训练的学习率下降方式采用余弦策略，设置初始学习率 0.001，最小学习率 0.00001，衰减权重 0.0005。

#### 2.4.1 评价指标

基于监督学习的自立袋封装缺陷分类检测模型中，含有凸形正样本、凸形褶皱、凸形未对齐、M 形正样本、M 形褶皱、M 形严重褶皱和 M 形未对齐，共 7 种分类类别。当检测某一类别时，则其它类别视为该类别的负样本。因此，针对该类别出现 4 种检测结果：真阳性(*True Positive, TP*)、假阳性(*False Positive, FP*)、假阴性(*False Negative, FN*)、真阴性(*True Negative, TN*)。由上述 4 种检测结果可以组成混淆矩阵，如表 2-3 所示。

表 2-3 混淆矩阵构成示意图

| 混淆矩阵 |                 | 真实值             |                 |
|------|-----------------|-----------------|-----------------|
|      |                 | <i>Positive</i> | <i>Negative</i> |
| 预测值  | <i>Positive</i> | <i>TP</i>       | <i>FP</i>       |
|      | <i>Negative</i> | <i>FN</i>       | <i>TN</i>       |

根据表 2-3 中的 *TP*、*FP*、*FN*、*TN* 可以计算出评价分类模型的不同性能指标。各指标计算公式如下：

$$Precision = \frac{TP}{TP + FP} \quad (2.15)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.16)$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (2.17)$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.18)$$

其中, *Precision* 为精确率, 表示预测值和真实值均为 *Positive* 的样本数与所有预测值为 *Positive* 的样本数之比; *Recall* 为召回率, 表示预测值和真实值均为 *Positive* 的样本数与所有真实值为 *Positive* 的样本数之比;  $F_1$  表示精确率和召回率加权平均值; *Accuracy* 为准确率, 表示预测正确的样本数与所有样本数之比。

此外, 检测类别平均精确率(*mean Precision, mPre*)、平均召回率(*mean Recall, mRe*)也常作为衡量模型检测精度的补充指标; 参数量(*Params*)表示网络模型包含的参数数量值, 常反映所构建网络模型的复杂度; 模型大小表示模型参数的总大小, 可视为模型存储空间的大小, 单位为 MB 或 GB。

#### 2.4.2 消融实验

为探究 Bneck 结构中 DWConv 位置、第二层卷积 GELU 激活函数、以及 CA 注意力替换 SE 注意力对 MobileNetV3 自立袋封装缺陷分类模型的影响, 本节对上述影响模型性能的三种因素进行 4 组消融实验, 分别为: ①DWConv 位于 Bneck 块第二层卷积、第二层卷积后采用 ReLU 或 H-swish、第二层卷积后使用 SE 注意力机制; ②DWConv 位于 Bneck 块第二层卷积、第二层卷积后采用 ReLU 或 H-swish、第二层卷积后使用 CA 注意力机制; ③DWConv 位于 Bneck 块第一层卷积、第二层卷积后采用 ReLU 或 H-swish、第二层卷积后使用 CA 注意力机制; ④DWConv 位于 Bneck 块第一层卷积、第二层卷积后采用 GELU 激活函数、第二层卷积后使用 CA 注意力机制。

在自立袋封装缺陷分类数据集上, 进行上述消融实验, 并根据各类别召回率的结果绘制混淆矩阵, 结果如图 2-16 所示。从图中可以看出, 相较于实验 1, 实验 2、实验 3 在牺牲“凸形未对齐”类别一定召回率的同时, 实现对“M 形正常”类别、“凸形正常”类别召回率的提升; 相较于实验 2 和实验 3, 实验 4 牺牲“凸形正常”类别召回率, 实现对“凸形未对齐”召回率的补偿; 以实验 1 结果为基准, 实验 4 在牺牲更小的“凸形未对齐”类别的召回率的条件下, 较实验 2、实验 3 更好地提升其他类别召回率。实验表明, Bneck 结构中 DWConv 放在第一层卷积、且在第二层卷积后采用 GELU 激活函数和选择 CA 注意力机制进行特征融合, 可以取得自立袋封装缺陷检测 7 个类别召回率更高、更均衡的检测效果, 使模型更好识别各类别样本。

根据混淆矩阵, 绘制消融实验训练过程各类别的平均准确率变化曲线, 如图 2-17(a)所示。可以看出, 实验 2、实验 3 及实验 4, 相较于实验 1 的原始

MobileNetV3 分类模型，能够更加迅速地收敛，且收敛准确率更高，表明后三种实验模型改进的有效性。

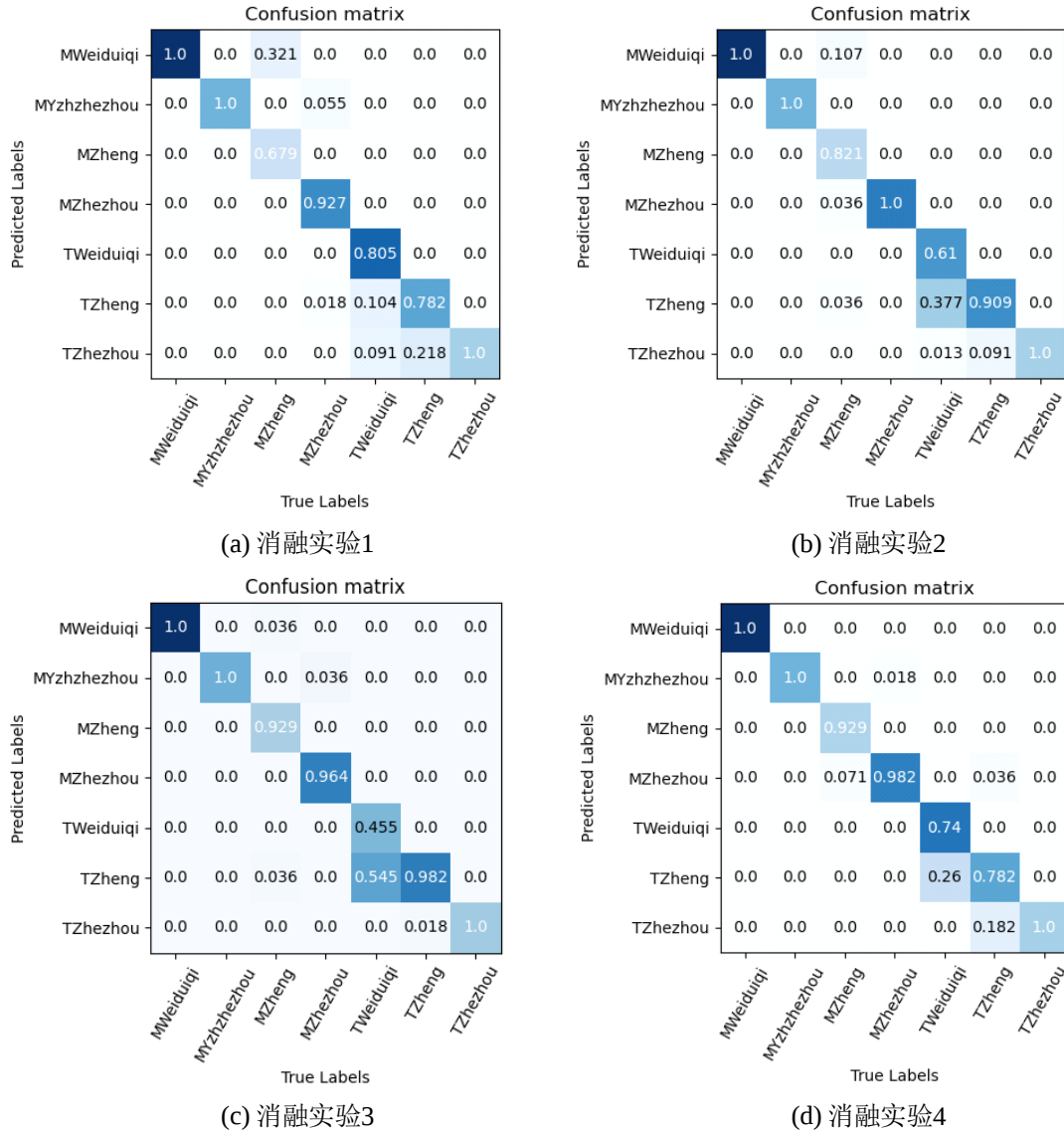
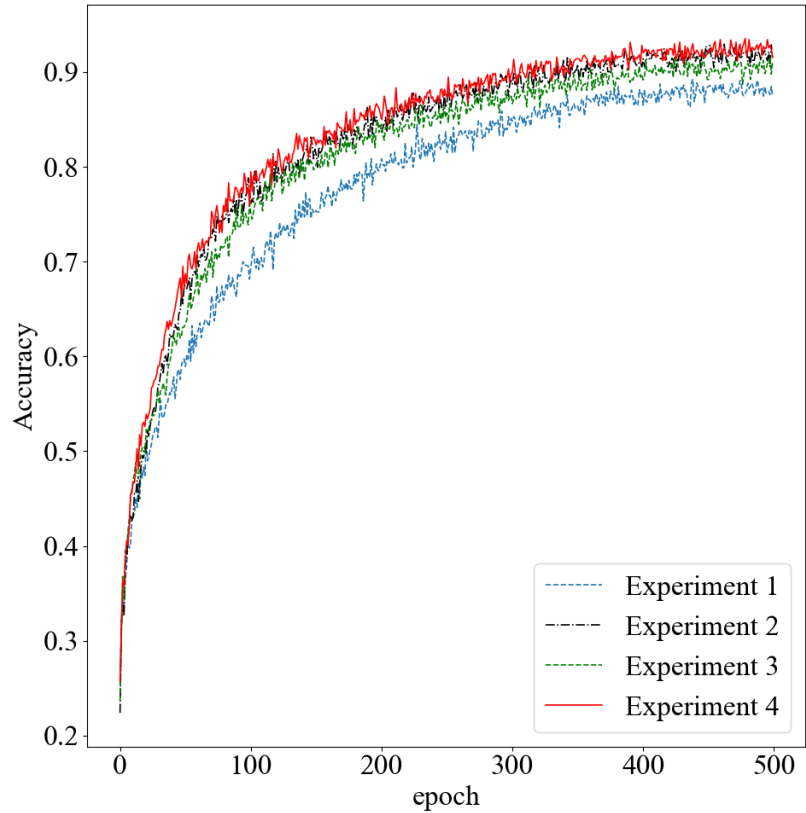


图 2-16 消融实验混淆矩阵图

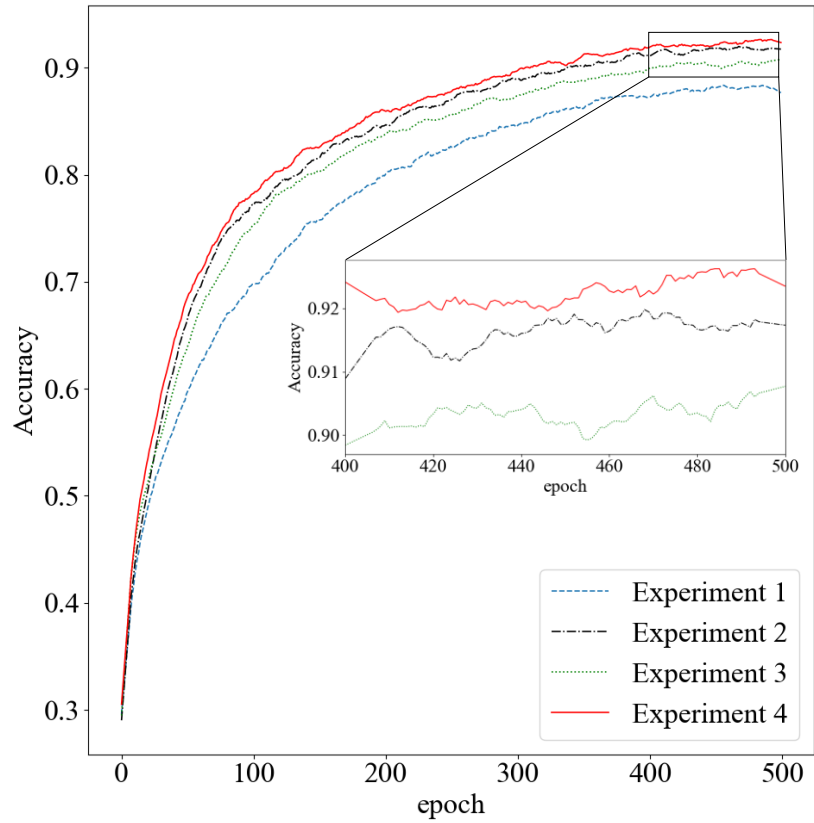
然而，直接可视化消融实验原始准确率数据，存在明显的收敛震荡现象，进而影响实验 2、实验 3 与实验 4 准确率曲线图的对比。为此，本节采用 Savitzky-Golay 滤波器对输入准确率数据进行平滑处理，具体操作为：调用 scipy 库中的“`savgol_filter(x, window_length, polyorder)`”函数，其中  $x$  表示输入的准确率数据；`window_length` 表示滤波窗口长度，实验中取 14；`polyorder` 表示拟合多项式阶数，实验中取 1。

经平滑处理的消融实验准确率曲线如图 2-17(b)所示。从准确率收敛速度和最终收敛数值的角度看，实验 2、实验 3、实验 4 实验结果均优于原始模型的实

验 1，而消融实验 4 又优于实验 3 和实验 2。



(a) 消融实验训练精度原始结果



(b) 消融实验训练精度平滑结果

图 2-17 消融实验训练过程准确率曲线

为进一步探讨三种改进 Bneck 结构的优劣，对自立袋封装缺陷检测分类模型的 Bneck 结构进行 4 组消融实验，结果如表 2-4 所示。

表 2-4 DWConv 位置、GELU 激活函数与 CA 注意力消融实验结果

|               |              | 实验 1  | 实验 2  | 实验 3         | 实验 4         |
|---------------|--------------|-------|-------|--------------|--------------|
| DWConv 位置     | 第一层          |       |       | √            | √            |
|               | 第二层          | √     | √     |              |              |
| 第二层卷积<br>激活函数 | GELU         |       |       |              | √            |
|               | ReLU/H-swish | √     | √     | √            |              |
| 注意力机制         | CA           |       | √     | √            | √            |
|               | SE           | √     |       |              |              |
| 评价指标          | Acc/%        | 88.83 | 88.83 | 87.15        | <b>90.22</b> |
|               | mPre/%       | 89.21 | 91.33 | <b>92.63</b> | 90.53        |
|               | mRe/%        | 88.47 | 90.58 | 90.67        | <b>91.89</b> |
|               | Params/M     | 1.53  | 1.16  | <b>1.11</b>  | <b>1.11</b>  |
|               | 模型大小/MB      | 5.82  | 4.42  | <b>4.23</b>  | <b>4.23</b>  |

从表 2-4 可以看出，第一层卷积为 DWConv、第二层卷积层后采用 GELU 激活函数和引入 CA 注意力进行特征融合时，模型在 Acc、mRe、Params、模型大小指标上取得最优，在数值上分别达到 90.22、91.89、1.11 和 4.23，表明消融实验 4 不仅正确分类性能最佳，而且实现模型参数量最小化，使得模型复杂度极大程度降低、模型占用内存最小，在一定程度上轻量化了模型，有助于模型的迁移和应用。虽然实验 4 在 mPre 指标上较实验 3 略差，但也相较原始 MobileNetV3 模型有 1.32%提升。

#### 2.4.3 对比实验

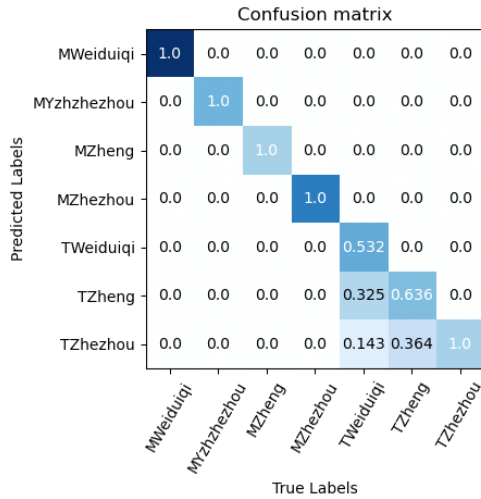
为进一步探索并验证改进模型对自立袋封装缺陷分类检测效果的优劣，将改进模型与 ResNet18、ShuffleNetV2-x1、MobileNetV2、EfficientNetV2-s 等分类模型进行实验对比，实验结果如表 2-5 所示。

由表 2-5 可知，改进 MobileNetV3 模型在自立袋封装缺陷检测的测试数据集上表现最佳，能够在模型参数量和模型大小分别保持 1.11M 和 4.23MB 的低水平条件下，实现模型测试 Acc、mPre、mRe、F1 性能指标的最优。表明在自立袋封装缺陷检测分类任务中，改进的 MobileNetV3 模型相较于其他经典模型具有复杂度更低、检测精度更高、更轻量化、更易部署的优点。

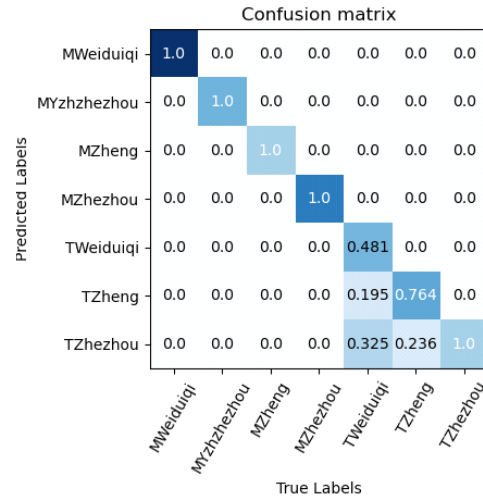
表 2-5 改进 MobileNetV3 模型的对比实验结果

| 模型               | Acc/%        | mPre/%       | mRe/%        | F1/%         | Params/M    | 大小/MB       |
|------------------|--------------|--------------|--------------|--------------|-------------|-------------|
| ResNet18         | 84.36        | 86.54        | 88.13        | 87.32        | 11.18       | 42.65       |
| ShuffleNetV2-x1  | 85.20        | 88.02        | 89.2         | 88.61        | 2.28        | 8.69        |
| MobileNetV2      | 86.87        | 89.20        | 90.44        | 89.82        | 2.23        | 8.52        |
| EfficientNetV2-s | 81.84        | 84.63        | 83.56        | 84.09        | 21.46       | 81.86       |
| MobileNetV3      | 88.27        | 89.92        | 90.24        | 90.08        | 1.53        | 5.82        |
| 本章方法             | <b>90.22</b> | <b>90.53</b> | <b>91.89</b> | <b>91.20</b> | <b>1.11</b> | <b>4.23</b> |

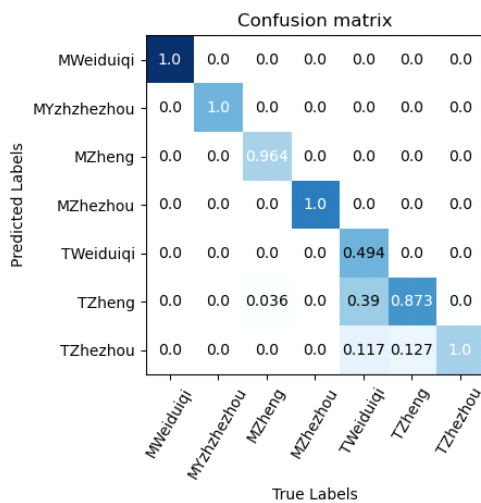
为进一步对比经典模型与改进模型对 7 个自立袋封装缺陷类别的检测效果，分别绘制各经典模型每个类别召回率的混淆矩阵，如图 2-18 所示。



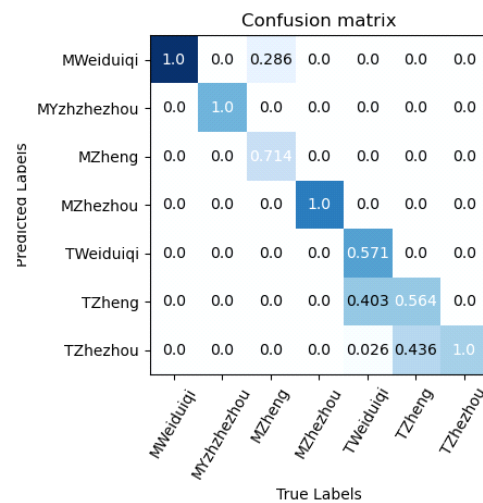
(a) Resnet18 混淆矩阵



(b) ShuffleNetV2-x1混淆矩阵



(c) MobileNetV2混淆矩阵



(d) EfficientNetV2-s混淆矩阵

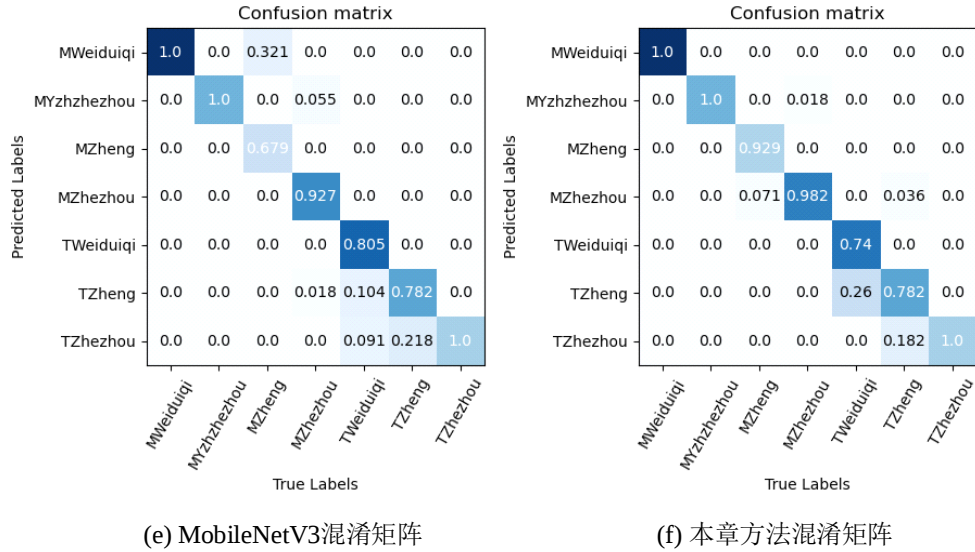


图 2-18 各经典模型自立袋封装缺陷检测混淆矩阵

从图中混淆矩阵可以看出，针对 M 形未对齐、M 形严重褶皱、M 形正常、M 形褶皱和凸形褶皱的 5 个类别分类，虽然 ResNet18、ShuffleNetV2-x1、MobileNetV2 模型检测效果良好，但是针对凸形未对齐和凸形正常类别，分类效果均具有较大误差。同时，EfficientNetV2-s 与 MobileNetV3 模型分别在 M 形正常、凸形未对齐 2 个类别的检测效果不佳。而本章方法在原始模型基础上，实现了上述两个类别检测误差的降低，7 个类别的平均召回率达到最高。

## 2.5 本章小结

本章首先根据监督学习理论要求，从包装设备生产企业实地采集自立袋封装缺陷图像，进行筛检、扩增等操作，获取自立袋封装缺陷分类检测数据集；其次，从缺陷检测分类网络模型的结构设计出发，介绍了经典的分类模型架构；然后，针对 MobileNetV3 模型的 Bneck 块，从 DWConv 位置、是否使用 GELU 激活、是否使用 CA 注意力替换 SE 注意力等角度进行改进；最后，利用 Acc、mPre、mRe、Params 与模型大小等性能指标，分别通过消融实验和对比实验，验证了改进模型在实现轻量化的同时，达到更高的缺陷检测精度。

## 第 3 章 基于无监督学习的自立袋封装缺陷检测

基于监督学习的自立袋封装缺陷检测分类方法通常需要采集和处理诸多的缺陷图像，而自立袋的缺陷种类繁多，难以完整获取，因此研究基于无监督学习方法的自立袋封装缺陷检测符合实际需求。本章在建立自立袋封装分割数据集的基础上，研究无监督缺陷检测结构，探讨和设计了一种基于 MB 正负样本差异特征的无监督自立袋封装缺陷检测方法，通过消融实验、对比实验和泛化实验，验证该方法的有效性、实用性和适用性。

### 3.1 构建自立袋封装分割数据集

常见的缺陷检测数据集是一种包含正常样本和缺陷掩码的分割数据集，如 MVTec AD 数据集<sup>[64]</sup>等。MVTec AD 数据集作为一种用于表面缺陷检测的经典基准数据集，常用于检验各种缺陷检测模型性能。本章也将在实验分析部分，使用该数据集进行模型泛化性能的验证。

MVTec AD 数据集包含纹理(Textures)和物体(Objects)两个大类，分别包含地毯(carpet)、栅格(grid)、皮革(leather)、瓷砖(tile)、木材(wood)，以及瓶口(bottle)、电缆(cable)、胶囊(capsule)、榛子(hazelnut)、金属螺母(metal nut)、药丸(pill)、螺丝(screw)、牙刷(toothbrush)、晶体管(transistor)、拉链(zipper)，共涉及 15 个类别 73 种表面缺陷，每个类别约 300 张图片，共 5354 张图像，其中训练与验证集 3629 张图片，测试集 1725 张。图像分辨率以 1024×1024 为主，为适应网络模型训练，常将输入图像大小统一为 256×256。

本节根据缺陷检测 MVTec AD 数据集的结构，针对不同颜色的凸形和 M 形缺陷图像构建自立袋封装缺陷分割数据集(Doypack-Defects)。

#### 3.1.1 EISeg 获取缺陷图像分割

Doypack-Defects 数据集原始图像来自 2.1 节自立袋缺陷分类数据集，利用 EISeg 软件对原始图像进行缺陷分割处理。首先，载入通用场景训练的模型权重；然后，添加缺陷类别标签名称与分割区域颜色，按 MVTec AD 数据集掩码格式，分割区域颜色选为白色；最后，加载缺陷图像文件夹，对缺陷对象逐一标注。每张缺陷图像标注完毕后生成的 3 张 png 格式图像，分别存储缺陷语义分割掩码标签文件、缺陷前景可视化图像和白色掩码标签文件，如图 3-1 所示。



图 3-1 缺陷图像与 EISeg 分割生成 3 种图像的对比

### 3.1.2 Doypack-Defects 数据集构成

构建 Doypack-Defects 数据集时，从分割标注生成的图像中选择白色掩码标签文件，与原始图像搭配构成该类别测试集和真实掩码图像。数据集中包含 M 形红色(M\_Red)、M 形白色(M\_White)、凸形粉色(Tu\_Pink)等 3 种类别自立袋的封口正常图像、缺陷异常图像以及缺陷异常的分割图像。3 种自立袋封装正常图像如图 3-2 所示。



图 3-2 三种自立袋封口正常样本展示

按照 MVTec AD 数据集的结构格式，构建 Doypack-Defects 数据集，其组成结构如表 3-1 所示。其中，test 和 train 分别表示测试集和训练集，ground\_truth 为测试集缺陷图像对应的分割掩码图像集；good 为无缺陷的正常样本，MWeiduiqi、MYzhzhezhou 分别为 M 形未对齐缺陷、M 形严重褶皱缺陷；TuWeiduiqi、TuZhezhou 分别表示凸形未对齐缺陷、凸形褶皱缺陷；M\_Red、M\_White 和 Tu\_Pink 三个类别训练集、测试集与测试集对应掩码集分别共有 89、72、62 张图像，数据集共有 223 张图像。

表 3-1 Doypack-Defects 数据集组成结构

| 数据集             | 一级文件夹   | 二级文件夹        | 三级文件夹       | 图像/张         |            |   |
|-----------------|---------|--------------|-------------|--------------|------------|---|
| Doypack-Defects | M_Red   | ground_truth | MWeiduiqi   | 9            |            |   |
|                 |         |              | MYzhzhezhou | 8            |            |   |
|                 |         |              | good        | 10           |            |   |
|                 |         | test         | MWeiduiqi   | 9            |            |   |
|                 |         |              | MYzhzhezhou | 8            |            |   |
|                 |         |              | train       | good         | 45         |   |
|                 | M_White | ground_truth | MWeiduiqi   | 8            |            |   |
|                 |         |              | good        | 10           |            |   |
|                 |         |              | MWeiduiqi   | 8            |            |   |
|                 |         | train        | good        | 46           |            |   |
|                 |         |              | Tu_Pink     | ground_truth | TuWeiduiqi | 6 |
|                 |         |              |             |              | TuZhezhou  | 6 |
|                 | good    | 6            |             |              |            |   |
|                 | test    | TuWeiduiqi   |             | 6            |            |   |
|                 |         | TuZhezhou    |             | 6            |            |   |
|                 |         | train        |             | good         | 32         |   |

### 3.2 基于无监督 MB 方法的缺陷检测网络

本节首先对无监督缺陷检测方法中的 MB 理论<sup>[39]</sup>进行详细阐述，分析该理论缺陷检测的优劣；在此基础上，将采用 MB 思想缺陷检测方法的自编码器<sup>[65]</sup>和 U-Net 结构<sup>[66]</sup>进行对比；最后选取 U-Net 结构，搭建基于 MB 思想的无监督缺陷检测模型。

#### 3.2.1 基于 MB 的缺陷检测原理

MB 思想最先发端应用于重构方法的缺陷检测模型，这类重构方法不是直接利用编码器压缩输出的低维特征进行图像重构，而是额外构建一个 MB 存储空间用于存储正常无缺陷样本或特征。在训练阶段，输入为正常无缺陷的样本，需要对 MB 存储项进行更新；在测试阶段，输入为正负样本，需要查询 MB 中的样本，以重构误差或者相似度比较进行异常检测。

由于重构思想无监督缺陷检测对输入图像数据分布假设不够精确、对输入图像模式敏感、难以处理复杂的数据结构，且难以确定重构误差的合适阈值，基于 MB 思想的无监督检测的发展越来越倾向于使用相似度比较或异常分数指标进行缺陷位置的定位。因此，使用 MB 思想的缺陷检测模型也可以归属为基于嵌入方法的缺陷检测。采用 MB 存储库对输入特征进行更新和对比的无监督缺陷检测优缺点如表 3-2 所示。

表 3-2 基于 MB 思想嵌入方法的无监督缺陷检测优缺点总结

| 优缺点 | 概述     | 说明                                      |
|-----|--------|-----------------------------------------|
| 优点  | 消除重构误差 | 限制卷积神经网络泛化性能，解决基于重构方法异常缺陷样本也被重构较好的问题。   |
|     | 自适应性   | 自适应地学习正常图像数据的分布，能够检测多类型缺陷。              |
|     | 可扩展性   | 可以根据需要进行扩展，用于加大嵌入特征向量的存储，为处理大规模数据提供可能。  |
|     | 高效性能   | 通过将输入特征存储在 MB 中，可以快速地进行相似性比较和缺陷检测。      |
| 缺点  | 限制推理速度 | 在推理阶段具有复杂的特征匹配过程，限制模型推理速度。              |
|     | 依赖性高   | 检测性能与嵌入模型的质量有关，如果嵌入模型设计不合理，将导致检测结果不够准确。 |

### 3.2.2 自编码器与 U-Net 结构

基于 MB 嵌入方法的无监督缺陷检测模型以自编码器或 U-Net 网络为主体结构。自编码器于 1986 年提出，在 2011 年引入卷积层，用于提升图像数据分割处理效果，其应用越来越广泛。自编码器最显著的特点是由编码器和解码器组合构成，编码器将输入图像压缩为低维特征表示，然后解码器利用低维特征进行图像还原<sup>[65]</sup>。而 U-Net 结构通过借用自编码器思想，采用编码器与解码器架构，实现图像分割任务。不过其核心思想是在两者之间引入跳跃连接，使网络能够在仅采用少量图像训练的基础上，实现像素点类别的预测，极大提高图像分割能力<sup>[66]</sup>。

由于无监督缺陷检测模型训练通常采用分割数据集，因此本章在构建自立袋缺陷分割数据集的基础上，在基于 MB 无监督自立袋缺陷检测模型中采用 U-Net 结构。U-Net 网络起初仅由卷积、上采样和下采样构成，后来卷积层普遍添加 BN 层，以加速网络训练并提高模型泛化性能。以 ResNet18 模型为编码器的 U-Net 结构为例，其结构如图 3-3 所示。

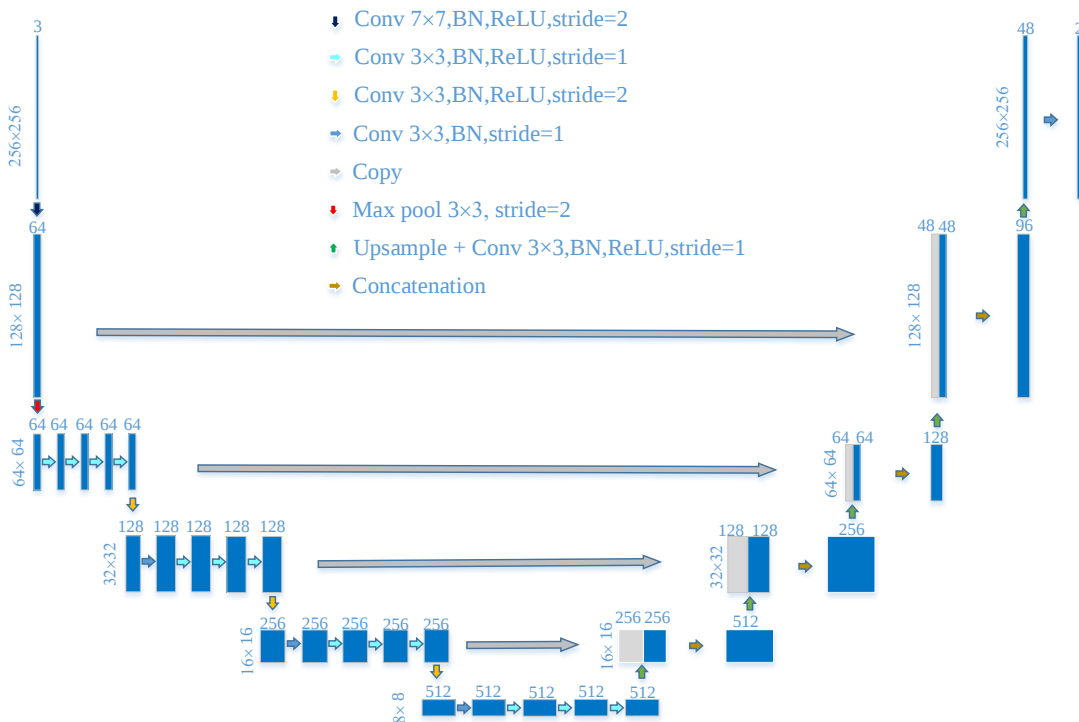


图 3-3 以 ResNet18 模型为编码器的 U-Net 结构

图 3-3 中，深色块和浅色块表示特征图，特征图上方的数字表示该特征图的通道数，特征图左侧数字表示特征图的宽和高，同一行特征高宽维度一致。可以看出 U-Net 网络通过最大池化或卷积对输入图像进行特征维度收缩，然后

利用最大池化和连接操作逐次对特征进行升维，最后提取图像预测掩码信息。

### 3.2.3 结合 MB 方法和 U-Net 网络的缺陷检测结构

根据上述 MB 思想以及 U-Net 分割网络架构，设计基于 MB 嵌入方法的无监督缺陷检测统一结构，如图 3-4 所示。

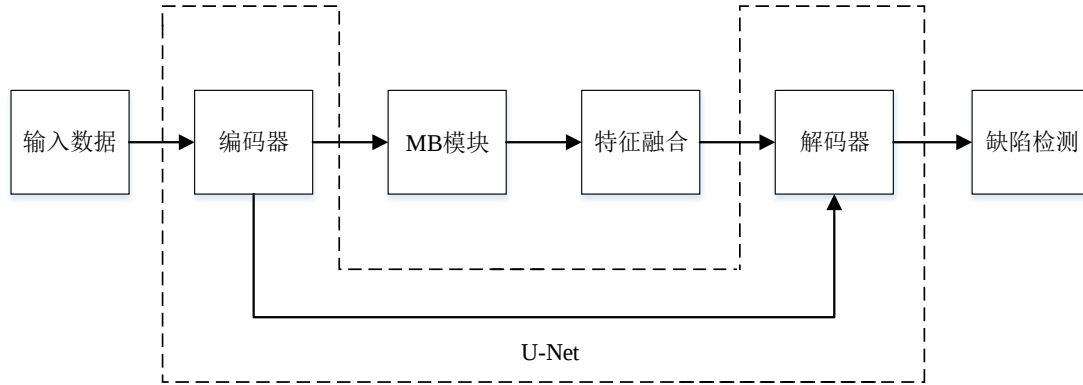


图 3-4 基于 MB 方法 U-Net 网络的缺陷检测结构

针对图 3-4 中各模块的说明如下：

**输入数据：**无监督缺陷检测任务的特征是输入为无标签图像，且仅提供少量缺陷图像和掩码进行模型测试。因此，在训练阶段输入数据仅为无缺陷的正常样本，采用不同的异常模拟策略生成人工异常缺陷；在测试阶段，输入数据为无缺陷正常样本和各类缺陷样本。

**编码器：**此处的编码器与后文的解码器通过跳跃连接，构成 U-Net 结构。在无监督缺陷检测中，编码器常采用在 ImageNet 数据集上训练过的具有预训练权重的网络模型。

**MB 模块：**将正样本图像经编码器输出的中层特征存储在 MB 中，用于训练和测试期间编码器中层特征的更新和比较，也可以存储差异信息，用于特征融合输入解码器。

**特征融合：**将跳跃连接解码器之前的编码器输出特征或 MB 模块匹配后的特征进行多尺度的交叉融合，有助于提取易于分割的有效信息。

**解码器：**此处的解码器与前文的编码器通过跳跃连接，组成 U-Net 结构。解码器无需与编码器的结构完全对称，但最终上采样生成一个通道为 2 的与原图像大小一致的特征图，以反映分割效果。

**缺陷检测：**通过计算测试样本与正常样本在解码器输出特征图上的距离差值，作为获取缺陷区域的异常分值，设定异常分数阈值用于缺陷检测。

### 3.3 基于 MB 正负样本差异特征的无监督自立袋封装缺陷检测

基于 MB 正负样本差异特征的无监督自立袋封装缺陷检测继承于存储库与异常模拟模型的方法，在关注模型精确度的同时，采用参数量较少的 ResNet18 编码器融于 U-Net 架构，以兼顾缺陷检测模型的轻量化。具体模型结构如图 3-5 所示。

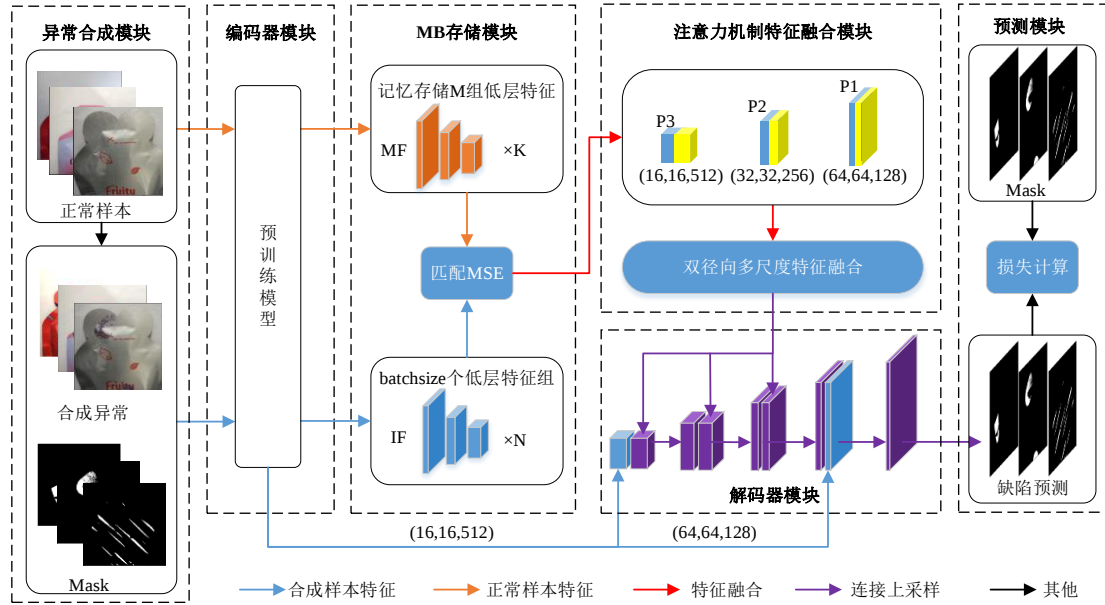


图 3-5 MB 正负样本差异特征缺陷检测框架

模型包含异常合成、编码器、特征存储、注意力机制特征融合、解码器以及预测等共 6 个模块，分别用于实现不同功能。异常合成模块利用正样本合成异常进而模拟缺陷样本，并与正样本一同作为编码器的输入；编码器模块用于提取正负样本特征，底层特征和高层特征传入解码器，中间特征传入 MB 存储模块；MB 存储模块通过匹配正负样本获取特征差异信息用于特征融合；注意力特征融合模块采用 CA 注意力和双径向特征融合多尺度特征；解码器模块利用融合后特征与编码器特征经连接和上采样恢复样本图像大小；预测模块计算生成预测图像与掩码 Mask 的损失指导模型训练。

编码器采用 ImageNet 预训练的 ResNet18 网络，并为保持 MB 存储库中正负样本低层信息的一致性，冻结前三层预训练模型参数。编码器输出特征从低层到高层的维度依次是： $128 \times 128 \times 64$ 、 $64 \times 64 \times 64$ 、 $32 \times 32 \times 128$ 、 $16 \times 16 \times 256$ 、 $8 \times 8 \times 512$ 。最高层特征经卷积上采样后，将特征维度  $16 \times 16 \times 512$  传入解码器模块，参与恢复检测图像大小的解码操作。本节将依次

介绍基于正负样本差异特征双径向融合缺陷检测模型的其余模块：异常模拟模块、特征存储模块、注意力特征融合模块和预测模块。

### 3.3.1 负样本合成模块

异常模拟的缺陷图像优劣关乎模型训练效果。受 DRAEM 异常合成思想的启发<sup>[45]</sup>，本文异常模拟模块在基底模型异常模拟方法基础上<sup>[53]</sup>，在掩膜和结构异常源获取过程中添加不同的图像增强策略。如图 3-6 所示，利用正样本合成异常模拟图像的具体流程分为三个阶段。

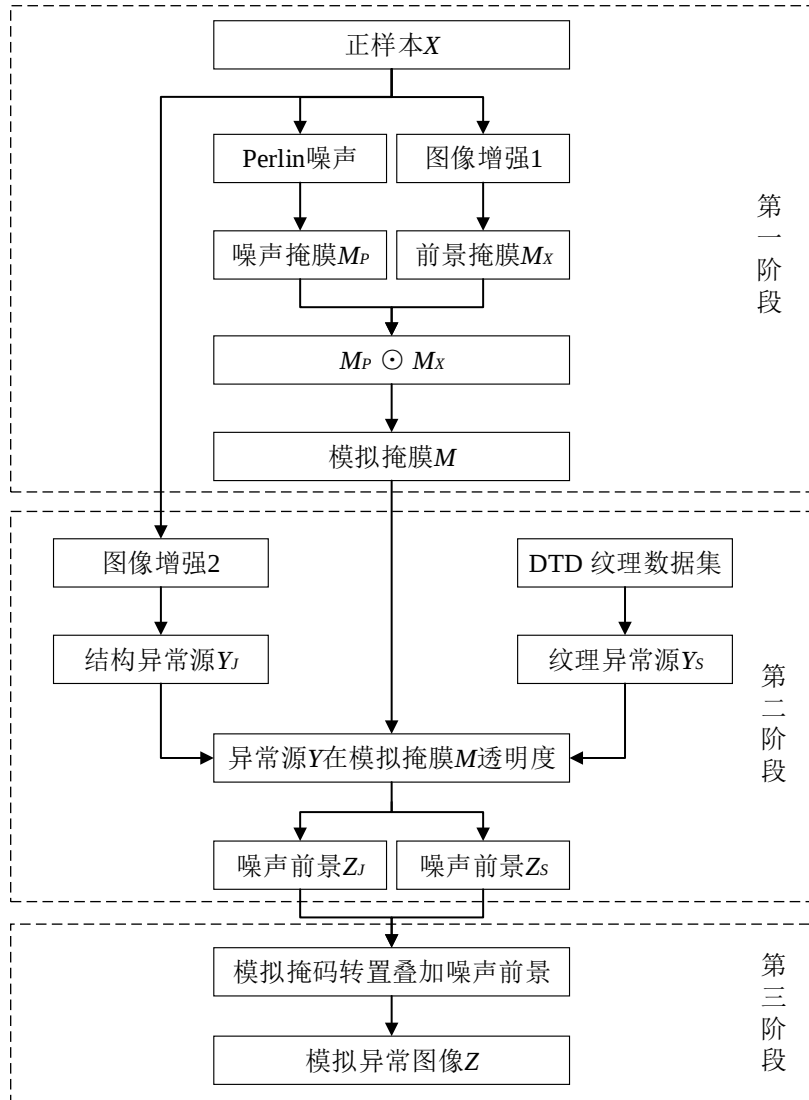


图 3-6 合成异常图像流程图

首先，在实际的生产制造过程中获取的缺陷图像正负样本常含有繁杂无关的背景信息，若不能对背景信息进行有效的剔除，目标物体的异常模拟将产生较大误差，进而影响模型训练结果。因此，在模拟异常合成图像的第一阶段，针对背景深浅不同的情况，分别采用叠加正样本或叠加正样本灰度化后再二次

叠加的简单图像增强方式，有效扩大图像目标物体与背景的像素差值，从而利于 OTSU 自适应阈值分割<sup>[67]</sup>获取图像前景信息生成前景掩膜  $M_x$ ；同时使用与正样本大小一致的 Perlin 噪声，经阈值化和旋转等处理后生成噪声掩膜  $M_p$ ； $M_x$  与  $M_p$  按元素相乘得到模拟掩膜  $M$ 。

在第二阶段，从 DTD 数据集<sup>[68]</sup>中选取图片作为异常合成的低层纹理，即丝织纹理异常源  $Y_s$ 。同时，添加原始图像的纹理信息，使异常模拟合成图像更趋近实际缺陷图像。为此，对正样本进行亮度、对比度、饱和度、锐度、色调以及转置等方面的增强，并在增强后分成  $8 \times 8$  栅格随机排列，重新组合成与正样本大小一致的图像，生成结构异常源  $Y_j$ 。然后，引入透明因子  $\beta$  平衡正样本  $X$  和异常源  $Y$  数据，并与模拟掩膜  $M$  融合，生成带有异常源噪声信息的前景图像  $Z_j$  或  $Z_s$ ，增加模拟异常与实际情况的相似性，具体公式如下：

$$\begin{cases} Z_j = \beta(M \odot Y_j) + (1 - \beta)(M \odot X) \\ Z_s = \beta(M \odot Y_s) + (1 - \beta)(M \odot X) \end{cases} \quad (3.1)$$

其中， $\odot$  表示按元素相乘，透明因子  $\beta$  在区间  $[0.15, 1]$  随机取值，可以视为对训练数据集的一种数据增强，扩展数据集的多样性。

在第三阶段，经模拟掩码  $M$  转置与正样本按元素相乘，得到将前景像素信息置零的图像，在此基础上，叠加异常源噪声前景图像  $Z_j$  或  $Z_s$ ，使 0 像素点与噪声前景像素点相加，获取模拟异常图像  $Z$ ，如公式(3.2)所示：

$$\begin{cases} Z = (1 - M) \odot X + Z_j \\ Z = (1 - M) \odot X + Z_s \end{cases} \quad (3.2)$$

经过上述三个阶段的异常模拟过程，合成模型训练所需带有缺陷的图像。以本章 2.1 节自制自立袋缺陷分类数据集中 M\_Red 和 Tu\_Pink 的 2 个类别，对添加丝织异常源模拟流程中各阶段生成掩膜和图像进行展示，如图 3-7 所示。

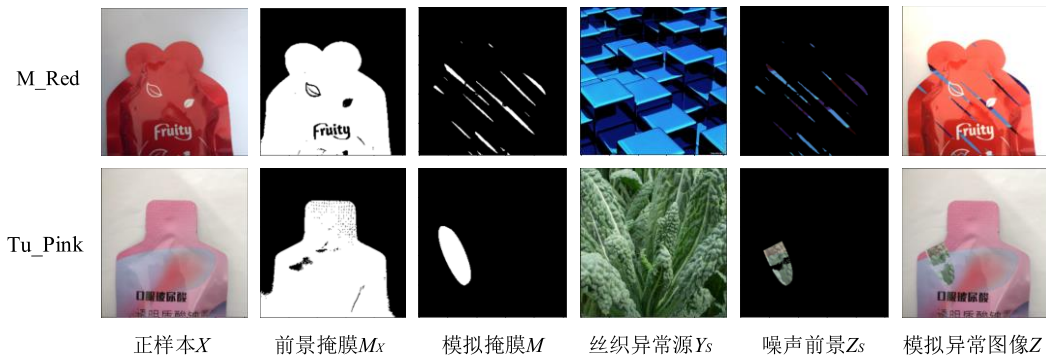


图 3-7 异常模拟图像生成展示

### 3.3.2 特征存储模块

结合存储库嵌入的思想，特征存储模块对正样本模式形成记忆，在模型训练和推理过程中计算输入特征与正样本模式的差异，并将这种差异融入注意力机制和多尺度特征融合模块，最终将多方处理的差异特征传递至解码器。本文采取一种简单有效的均方与差方误差结合的方式提取差异特征。

首先，正样本模式应当记忆样本中有效细节信息，摒弃抽象化的信息碎片，同时约束分辨率以保证检测速率。因此，本研究从训练集的每个类别中随机选取  $K$  个正样本，经预训练模型提取中间 3 层特征作为  $K$  组记忆存储特征，每组特征维度信息分别为： $64 \times 64 \times 64 \times K$ 、 $32 \times 32 \times 128 \times K$ 、 $16 \times 16 \times 256 \times K$ 。在训练和推理过程中，经编码器提取输入图像特征后，通过计算 3 个中间层特征与所有正样本记忆特征的均方误差，获得输入图像与存储信息的差异信息。针对输入的 1 张图片和存储库中的第  $k$  组正样本特征，差异信息  $DI_k$  由下式计算：

$$DI_k = \frac{1}{64^2} \sum_{i=0}^{64^2} [MF_k(64) - IF(64)]^2 + \frac{1}{32^2} \sum_{i=0}^{32^2} [MF_k(32) - IF(32)]^2 + \frac{1}{16^2} \sum_{i=0}^{16^2} [MF_k(16) - IF(16)]^2 \quad (3.3)$$

其中，64、32 和 16 分别表示不同特征图尺度大小； $MF_k(64)$ 、 $MF_k(32)$ 、 $MF_k(16)$  分别表示第  $k$  组不同尺度的存储特征， $IF(64)$ 、 $IF(32)$ 、 $IF(16)$  为输入图像不同尺度特征。

其次，计算最佳存储特征与输入图像特征差方损失。 $K$  组存储特征对应  $K$  组差异信息，遍历获取最小的差异信息，得到此差异信息所对应的存储特征作为最佳存储特征。该差异信息一定程度上反映出输入图像与最佳存储样本相似度最高。最佳差异信息则由最佳存储特征和输入图像特征计算每个像素点差方损失构成，该差异信息在某一区域的值越大，在输入图像相应的区域为缺陷的概率越大。输入图像特征  $IF$  与记忆存储特征  $MF$  的最佳差异特征计算如下式：

$$DI_{best} = \begin{cases} [MF_{best}(64) - IF(64)]^2, & c = 64 \\ [MF_{best}(32) - IF(32)]^2, & c = 128 \\ [MF_{best}(16) - IF(16)]^2, & c = 256 \end{cases} \quad (3.4)$$

其中， $c$  为特征通道数； $MF_{best}(64)$ 、 $MF_{best}(32)$ 、 $MF_{best}(16)$  为记忆存储的不同

尺度的最佳特征； $DI_{best}$  为最佳差异特征，维度为  $64 \times 64 \times 64$ 、 $32 \times 32 \times 128$ 、 $16 \times 16 \times 256$ 。

最后，最佳差异特征  $DI_{best}$  与输入图像特征  $IF$  在通道维度上连接，将  $64 \times 64 \times 128$ 、 $32 \times 32 \times 256$ 、 $16 \times 16 \times 512$  的  $P_1$ 、 $P_2$ 、 $P_3$  多尺度特征送入注意力机制特征融合模块。

### 3.3.3 注意力特征融合模块

经过特征存储模块，获取由输入特征  $IF$  和最佳差异特征  $DI_{best}$  连接而得的多尺度特征  $P_n$ 。直接使用多尺度特征  $P_n$  将会产生特征冗余的问题，而且会增大模型计算量，导致推理速度减慢。为缓解这一问题，在 PANet 等路径聚合方法的启发下<sup>[69]</sup>，采用一种利用 CA 注意力和双路径聚合的方法融合多尺度特征，组成注意力特征融合模块。其结构如图 3-8 所示。

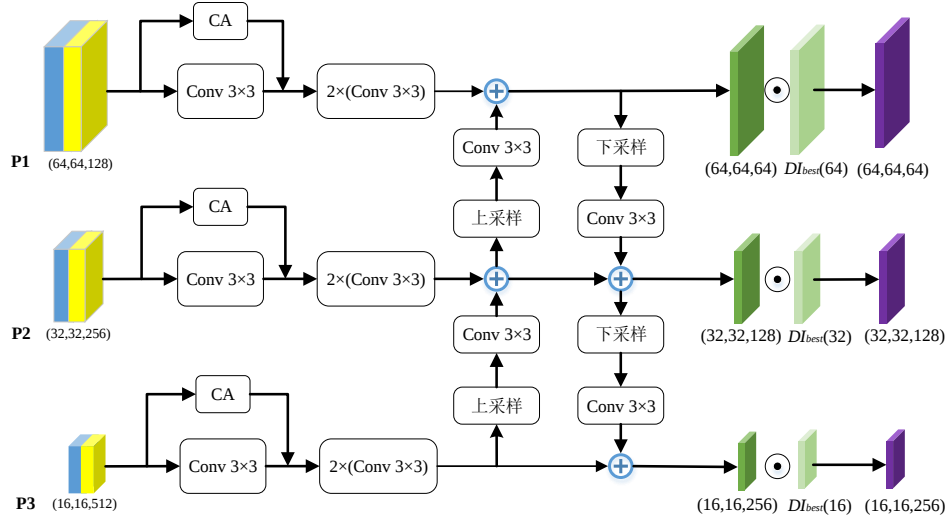


图 3-8 注意力特征融合模块结构图

输入注意力特征融合模块的多尺度特征  $P_n$ ，利用整合通道信息和方向位置信息的 CA 注意力，分别获取  $P_1$ 、 $P_2$  及  $P_3$  不同通道的关联信息，并加权不同维度特征。如图 2-12 所示，CA 注意力的输入输出不改变特征维度大小，因而在此基础上使用  $3 \times 3$  卷积与注意力输出按元素相乘，得到特征聚焦后的多尺度特征维度分别为： $64 \times 64 \times 128$ 、 $32 \times 32 \times 256$ 、 $16 \times 16 \times 512$ 。然后每个尺度特征经过 2 层  $3 \times 3$  卷积实现通道减半，进一步提高特征表达能力，卷积后特征尺度分别为： $64 \times 64 \times 64$ 、 $32 \times 32 \times 128$ 、 $16 \times 16 \times 256$ 。接着，进行双径向特征融合操作：使用双线性上采样或下采样对不同尺度大小的特征图进行高宽对齐，使用  $3 \times 3$  卷积对通道数量进行 2 倍压缩或扩充，保持高层特征到低层特征方向聚

合的特征维度信息一致，并对聚合特征执行按元素相加运算，实现高层到低层或低层到高层的特征聚合。经过双路径聚合操作后，不同尺度特征的维度信息为： $64 \times 64 \times 64$ 、 $32 \times 32 \times 128$ 、 $16 \times 16 \times 256$ 。

最后，为充分利用均方与差方损失计算获取的差异信息，更好地分辨和定位缺陷区域，采用空间注意力特征图与路径聚合特征图按元素相乘的方法，将缺陷位置信息相关的最佳差异特征再次融合。不同尺度的空间特征计算如公式(3.5)所示。

$$\begin{cases} S(16) = \frac{1}{256} \sum_{i=1}^{256} DI_{best}(16)_i \\ S(32) = \frac{1}{128} \left[ \sum_{i=1}^{128} DI_{best}(32)_i \right] \odot S(16)^{up} \\ S(64) = \frac{1}{64} \left[ \sum_{i=1}^{164} DI_{best}(64)_i \right] \odot S(32)^{up} \end{cases} \quad (3.5)$$

其中， $S(16)$ 、 $S(32)$ 、 $S(64)$ 表示高宽为16、32、64的空间注意力特征图；256、128及64表示不同尺度大小特征图的通道数； $DI_{best}(16)_i$ 、 $DI_{best}(32)_i$ 、 $DI_{best}(64)_i$ 表示不同尺度的第*i*个通道特征图； $S(16)^{up}$ 、 $S(32)^{up}$ 表示对 $16 \times 16$ 、 $32 \times 32$ 的空间注意力特征图进行2倍的双线性差值上采样，保持特征信息的完整性。

### 3.3.4 预测模块

预测模块针对解码还原的图像进行阈值分割，以生成预测掩码，并将预测掩码与模拟生成的掩码进行损失计算，量化预测结果，指导模型优化方向。预测模块的损失函数通过对L1损失、Focal损失以及Dice损失加权获取。其中，L1损失与L2损失相比，可以保留更多的边缘信息；Focal损失有助于缓解预测图像中正常区域与缺陷区域的不平衡问题，使模型集中训练难以训练的样本上，进而提高损失分割精度；Dice损失常用于图像分割任务，能够对少数类别的模型预测结果进行更好的评估，同时具有一定边缘细节信息保留能力。利用预测分割图像 $\hat{M}$ 与模拟异常掩码图像 $M$ 分别计算上述3种损失，公式如下：

$$L_{l1} = |M - \hat{M}| \quad (3.6)$$

$$L_f = -\alpha_t (1 - p_t)^\gamma \log(p_t) \quad (3.7)$$

$$L_d = 1 - \frac{2|M \cap \hat{M}|}{|M| + |\hat{M}|} \quad (3.8)$$

其中,  $L_{l1}$ 、 $L_f$ 、 $L_d$  分别表示 L1 损失、Focal 损失及 Dice 损失;  $p_t$  表示像素预测概率值, 若  $p$  表示像素类别 0 或 1 的预测概率, 当某像素对应模拟掩码  $M$  该位置像素为 1 时  $p_t = p$ , 当某像素对应模拟掩码  $M$  该位置像素为 0 时  $p_t = 1 - p$ 。

结合三种损失的优点, 对以上三种损失赋权值, 组成加权损失, 在训练过程中通过不断学习降低该损失, 优化模型。

$$L = \lambda_{l1}L_{l1} + \lambda_fL_f + \lambda_dL_d \quad (3.9)$$

其中,  $L$  为模型总的加权损失,  $\lambda_{l1}$ 、 $\lambda_f$ 、 $\lambda_d$  分别表示 L1 损失、Focal 损失及 Dice 损失加权超参数, 且满足条件  $\lambda_{l1} + \lambda_f + \lambda_d = 1$ 。

### 3.4 实验结果分析

基于 MB 正负样本差异特征的无监督自立袋封装缺陷检测模型的实验验证与分析, 共分为四个部分: ①介绍无监督自立袋封装分割模型采用的评价指标; ②从异常模拟数据增强和双径向特征融合两方面进行消融实验, 研究所提模型缺陷分割效果; ③将所提模型与各经典模型进行实验对比, 探讨所提模型检测是否具有优越性; ④使用 MVTec AD 数据集对所提模型进行泛化实验。

实验采用的硬件配置为 AMD Ryzen 7 5800H (3.20 GHz) 16GB 处理器, NVIDIA GeForce RTX 3070 8GB 显卡; 操作系统为 Windows 10, 软件平台为 PyCharm; 使用深度学习框架为 Pytorch1.10.0, Python 版本为 3.8, 在 CUDA11.3 和 cuDNN8.21 上进行加速训练。实验参数的设置: batch size 为 8; 训练迭代次数为 5000; 存储特征组数  $K$  取 30; Perlin 噪声阈值为 0.5; 优化器为 AdamW; 训练的学习率采用余弦退火预热重启策略, 设置最大初始学习率 0.003, 最小学习率 0.0001, 预热比重 0.1, 衰减权重 0.0005。

#### 3.4.1 评价指标

实验以接受者操作特征曲线下面积 (Area Under the Receiver Operating Characteristic Curve, AUROC) 为评价指标, 其取值范围在 0 到 1 之间, 数值越大表示模型越好。AUROC 表示 ROC 曲线线下面积的大小, 而 ROC 曲线是以真阳性率 (True Positive Rate, TPR) 为纵坐标, 假阳性率 (False Positive Rate, FPR) 为横坐标的曲线,  $TPR$  和  $FPR$  计算如式 (3.10)、(3.11) 所示。

$$TPR = \frac{TP}{TP + FN} \quad (3.10)$$

$$FPR = \frac{FP}{FP + TN} \quad (3.11)$$

其中， $TP$ 为真正例， $FN$ 为假反例， $FP$ 为假正例， $TN$ 为真反例，它们由混淆矩阵得出。

图像级 AUROC 综合考虑模型在多个类别上的分类能力，提供一个全局的性能度量；而像素级 AUROC 考虑图像中每个像素点的分类结果，在像素级别上评估模型对不同缺陷的分割能力。实验分别从图像级别和像素级别对缺陷检测结果进行评估，更加综合全面地评价模型的性能。

### 3.4.2 消融实验

为探究异常模拟数据增强和双径向特征融合对缺陷分割效果的优劣，本节在自立袋分割数据集 Doypack-Defects 上进行消融实验，以图像级、像素级 AUROC 指标，研究数据集中 Tu\_Pink、M\_White 与 M\_Red 三个类别无监督缺陷检测分割效果。实验结果如表 3-3、表 3-4 和表 3-5 所示。

表 3-3 Tu\_Pink 类别缺陷消融实验检测效果

|      |           | 实验 1  | 试验 2  | 实验 3  | 实验 4         |
|------|-----------|-------|-------|-------|--------------|
| 措施   | 基本模型      | √     | √     | √     | √            |
|      | 异常模拟增强    |       | √     |       | √            |
|      | 双径向特征     |       |       | √     | √            |
| 评价指标 | 图像级 AUROC | 0.639 | 0.889 | 0.771 | <b>0.917</b> |
|      | 像素级 AUROC | 0.776 | 0.870 | 0.885 | <b>0.878</b> |

表 3-4 M\_White 类别缺陷消融实验检测效果

|      |           | 实验 1  | 试验 2  | 实验 3  | 实验 4         |
|------|-----------|-------|-------|-------|--------------|
| 措施   | 基本模型      | √     | √     | √     | √            |
|      | 异常模拟增强    |       | √     |       | √            |
|      | 双径向特征     |       |       | √     | √            |
| 评价指标 | 图像级 AUROC | 1     | 1     | 1     | 1            |
|      | 像素级 AUROC | 0.553 | 0.583 | 0.611 | <b>0.728</b> |

表 3-5 M\_Red 类别缺陷消融实验检测效果

|      |           | 实验 1  | 试验 2  | 实验 3  | 实验 4         |
|------|-----------|-------|-------|-------|--------------|
| 措施   | 基本模型      | √     | √     | √     | √            |
|      | 异常模拟增强    |       | √     |       | √            |
|      | 双径向特征     |       |       | √     | √            |
| 评价指标 | 图像级 AUROC | 1     | 1     | 1     | 1            |
|      | 像素级 AUROC | 0.874 | 0.881 | 0.856 | <b>0.893</b> |

由上表可知，对 3 类缺陷而言，同时添加异常模拟和双径向特征融合方法，能实现基底模型评价指标保持和提升；M\_Red 缺陷消融实验表明，添加双径向特征效果并不理想，但是与异常模拟数据增强操作一同作用于基底模型时，会提升像素级 AUROC 指标；M\_White 缺陷消融实验，证实了两种改进实验的有效性，同时像素级 AUROC 仅为 0.728，表明模型对该类别的分割效果仍具有提升空间。以消融实验 4 作为模型在 Doypack-Defects 数据集上的最终实验结果。3 种缺陷无监督分割实验结果如图 3-9 所示。



(a) Tu\_Pink缺陷消融实验4分割结果



(b) M\_White 缺陷消融实验4分割结果



(c)M\_Red 缺陷消融实验4分割结果

图 3-9 基于 MB 正负样本差异特征的无监督自立袋封装缺陷检测效果

### 3.4.3 对比实验

为进一步探索并验证基于 MB 正负样本差异特征的检测模型对自立袋封装缺陷分类与分割任务的检测效果，本节在消融实验结果的基础上，利用所提模型与基于重构思想的 Draem<sup>[45]</sup>、基于嵌入归一化流方法的 CS-Flow<sup>[46]</sup>、基于嵌入导师学生方法的 STPM<sup>[48]</sup>等经典异常检测模型进行对比实验。实验在自制 Doypack-Defects 数据集上进行训练，以获取图像级和像素级 AUROC 性能指标进行对比，实验结果如表 3-6、表 3-7 所示。

表 3-6 Doypack-Defects 数据集缺陷检测经典模型图像级 AUROC 对比表

| 类别      | 文献[45] | 文献[46] | 文献[48] | 本章方法         |
|---------|--------|--------|--------|--------------|
| Tu_Pink | 0.778  | 0.667  | 0.778  | <b>0.917</b> |
| M_White | 0.762  | 0.719  | 0.775  | <b>1.000</b> |
| M_Red   | 0.871  | 0.662  | 0.741  | <b>1.000</b> |

表 3-7 Doypack-Defects 数据集缺陷检测经典模型像素级 AUROC 对比表

| 类别      | 文献[45] | 文献[46] | 文献[48]       | 本章方法         |
|---------|--------|--------|--------------|--------------|
| Tu_Pink | 0.852  | 0.817  | <b>0.934</b> | 0.878        |
| M_White | 0.705  | 0.640  | 0.686        | <b>0.728</b> |
| M_Red   | 0.716  | 0.721  | 0.850        | <b>0.893</b> |

各模型关于自立袋缺陷分割检测效果对比如图 3-10 所示。



图 3-10 检测效果对比图

由上表以及缺陷分割对比图可知，在 Doypack-Defects 数据集三个类别的识别检测中，本章方法图像级 AUROC 均为最高，表明所提模型对缺陷的分类检

测效果更优、准确性更高；而在像素级 AUROC 指标上，本章方法在 M\_White 和 M\_Red 类别检测上取得最优值，在 Tu\_Pink 类别分割检测上低于 STPM 模型，表明在模型分割细粒度上，所提模型仍具有提升空间，但综合分割效果良好。

#### 3.4.4 泛化实验

为进一步研究所提模型适用性，在 MVTec AD 纹理数据集上进行泛化实验，获取图像级和像素级 AUROC 性能指标，并将其与经典的缺陷检测算法进行对比。实验结果如表 3-8、3-9 所示。

表 3-8 MVTec AD 数据集缺陷检测经典模型图像级 AUROC 泛化实验对比表

| target  | 文献[51]       | 文献[70] | 文献[71]       | 本章模型         |
|---------|--------------|--------|--------------|--------------|
| carpet  | <b>0.984</b> | 0.928  | 0.931        | 0.977        |
| grid    | 0.898        | 0.473  | <b>0.999</b> | 0.997        |
| leather | 0.988        | 0.954  | <b>1.000</b> | <b>1.000</b> |
| tile    | 0.959        | 0.965  | 0.934        | <b>1.000</b> |
| wood    | 0.990        | 0.959  | 0.986        | <b>0.999</b> |
| average | 0.964        | 0.856  | 0.970        | <b>0.995</b> |

表 3-9 MVTec AD 数据集缺陷检测经典模型像素级 AUROC 对比表

| target  | 文献[51] | 文献[70] | 文献[71]       | 本章模型         |
|---------|--------|--------|--------------|--------------|
| carpet  | 0.988  | 0.663  | 0.983        | <b>0.948</b> |
| grid    | 0.936  | 0.566  | <b>0.975</b> | 0.963        |
| leather | 0.990  | 0.615  | <b>0.995</b> | 0.987        |
| tile    | 0.917  | 0.632  | 0.905        | <b>0.991</b> |
| wood    | 0.900  | 0.644  | 0.905        | <b>0.949</b> |
| average | 0.946  | 0.624  | 0.953        | <b>0.968</b> |

由上表可知，在本章模型在 MVTec AD 纹理数据集的缺陷检测效果最优，与其它 3 种缺陷检测算法相比，图像级和像素级 AUROC 平均性能指标最高，分别达到 0.995 和 0.968，表明所提模型在纹理缺陷的分类与分割效果良好。同时，证实了本章模型不仅适用于自立袋封装缺陷检测，也适用于纹理图像缺陷检测任务，验证了模型具有良好的可移植性和泛化性能。此外，为了直观展示基于 MB 正负样本差异特征双径向融合的无监督缺陷检测方法泛化效果，在 MVTec AD 数据集 15 个类别上进行异常检测并可视化异常预测掩码，如图 3-11 所示。图中每 3 张图像为一组，从左至右分别展示缺陷类别原始图像、缺陷掩

码、缺陷预测掩码叠加原始图像。

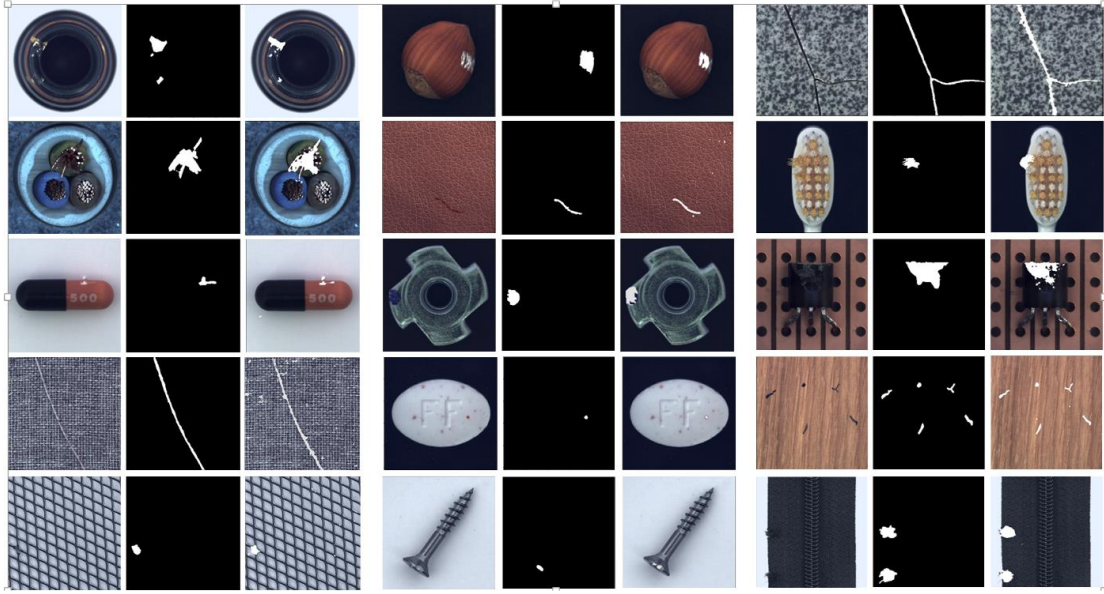


图 3-11 MVTec AD 数据集上的检测效果

### 3.5 本章小结

本章以自立袋封装缺陷分类数据集为基础，使用 EISeg 软件获取缺陷图像掩码，构建用于无监督自立袋封装缺陷分割的 Doypack-Defects 数据集；介绍基于 MB 思想的无监督缺陷检测方法，以及该方法常采用的自编码器与 U-Net 结构，并给出融合 MB 方法和 U-Net 网络无监督缺陷检测的组成架构；在此基础上，探讨并设计了一种采用异常模拟和正负样本差异特征双径向融合策略的无监督分割网络模型；最后，利用图像级和像素级 AUROC 为评价指标，在 Doypack-Defects 数据集上，分别进行消融实验和对比实验，对所提模型的有效性和先进性进行验证，并通过 MVTec AD 纹理数据集验证了其良好的泛化能力。

## 第 4 章 自立袋封装缺陷检测系统设计

本章在自立袋封装缺陷检测算法研究的基础上，设计缺陷检测系统。首先，结合实际生产需求分析系统设计要求；在此基础上，研究缺陷检测系统架构；最后，根据检测系统架构完成缺陷检测软件界面的搭建。

### 4.1 缺陷检测系统设计要求

构建自立袋封装缺陷检测系统，需要理清系统实现的目标，形成具有生产实际特性的设计需求。首先，本文研究的对象是物料填充后自立袋封装的缺陷，检测袋形包括凸形自立袋和 M 形自立袋，缺陷类别包含褶皱、严重褶皱、未对齐等。视觉采集图像背景是否单一、光线是否均匀、自立袋表面是否有污渍或反光等构成自立袋表面是否含有缺陷的因素。通过对包装设备生产流程和自立袋封装调试的调研，实际工业生产将对自立袋封装缺陷视觉检测提出以下要求：

1) 稳定性：控制系统工作的基本前提。系统需要具有在各种情况下均能保持正常运行的能力，不会因外部干扰或内部错误而导致系统崩溃或失效。尤其在检测算法上，应尽量优化图像采集过程，使检测图像清晰而不受上述影响因素干扰；

2) 准确性：反映系统检测与执行的误差。为了防止包装运输泄露问题的发生，系统缺陷检测应保持高准确率和低误检率与漏检率，将包装产线中所有缺陷自立袋准确识别并剔除；

3) 快速性：反映系统响应动态过程的能力。为适配包装设备生产速度，检测系统应当具有自立袋包装设备相当的检测速度，视觉检测系统应当具有每秒 2~3 个产品以上的检测速度，对缺陷检测算法和设备算力有较高要求。

4) 良好的人机交互：实现人工与系统的信息交流与互动。缺陷检测系统需要集成为具有人机交互功能的软件，该软件应实现移植和部署便捷化、功能管理和调试模块化、操作界面简洁化。

总之，视觉缺陷检测系统在人机交互界面启动检测后，需要将不同袋形不同缺陷稳定准确地识别，并在交互界面中显示，同时将识别信号传至 PLC 等控制器分析处理，由控制器发出信号，再由执行机构快速剔除缺陷自立袋。

## 4.2 系统架构设计

本节在明确缺陷检测系统设计要求的基础上，对缺陷检测系统的架构进行设计。以 DS-180DS 型号包装机为自立袋缺陷检测系统研究的基础，该型号包装生产设备集复合材料热封成形、物料注入、热封封口及成品输出为一体，实现液体、粉末、固体颗粒等样式的食品、日化和医药产品的快速包装。该型号包装设备示意图如图 4-1 所示。



图 4-1 DS180-DS 自立袋包装机示意图

简化 DS180-DS 设备机械结构，分析自立袋包装生产作业流程，其示意图与流程图如图 4-2、图 4-3 所示。示意图中标记有 20 个操作装置，并与作业流程图中的标号一致。自立袋包装的生产流程分成五个阶段，分别为：放膜成形阶段、塑封成形阶段、裁剪牵引阶段、开袋填充阶段以及上封与成品输出阶段。

**放膜成形阶段：**将预制的复合材料薄膜进行拉放，经多个滚轴整理，进行冲孔操作，然后经引导装置使包装膜成形，使用色标传感器对每次薄膜拉放尺度进行测量，确保每次放膜尺寸一致。

**塑封成形阶段：**依次通过预留特定凹槽的加热棒对已成形的塑料薄膜的底部、左侧和右侧进行加热塑封，形成保留上端口未封的连体袋。

**裁剪牵引阶段：**将连体袋中间位置经易撕口装置进行冲口操作，然后利用剪切机构对自立袋进行裁剪，形成单个的自立袋，使用伺服牵引和夹袋牵引分别实现对连体袋和单体袋的迁引和传递。

**开袋填充阶段：**利用前后吸盘对自立袋进行开袋，然后采用吹气装置使自立袋鼓起，方便灌装设备将物料进行定量填充，根据包装速度的要求可以选择采用单填充或多填充装置。

上封与成品输出阶段：经过预留特殊凹槽的加热棒对已填充的自立袋进行上封操作，上述流程完整正确无误的运行后将输出成品。

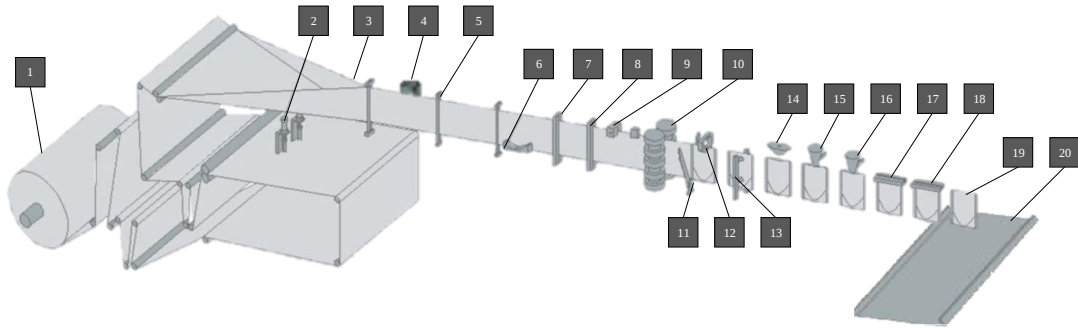


图 4-2 DS180-DS 自立袋包装设备作业示意图

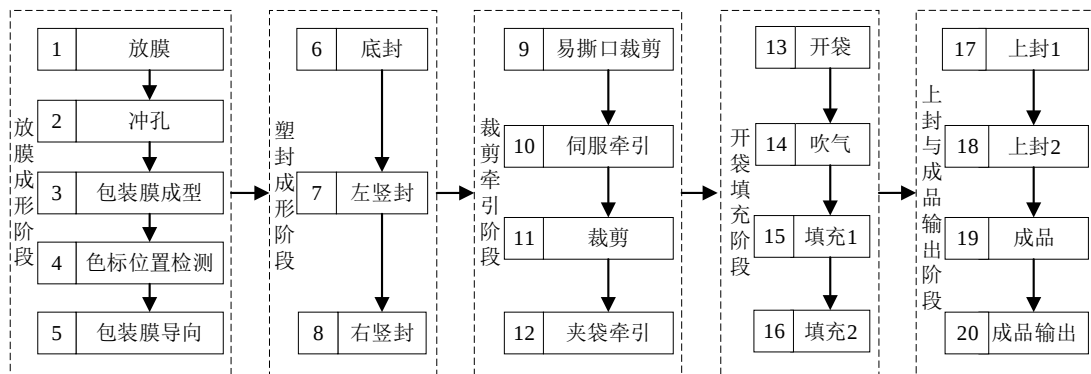


图 4-3 DS180-DS 自立袋包装设备作业流程图

缺陷检测系统的构建根植于 DS180-DS，但略有不同。首先，根据本文研究的缺陷检测算法所涉及的缺陷类别，包装设备在上封与成品输出阶段，加热棒凹槽应当为凸形凹槽或 M 形凹槽；其次，缺陷检测系统不仅涉及缺陷识别还需要实现对缺陷自立袋的剔除。因此，上封与成品输出阶段的成品输出需要电控的剔除机构，可以选取单摆臂作为成品自立袋剔除与否的执行机构。

自立袋封装缺陷检测系统主要包含 3 个部分：图像采集模块、缺陷检测模块以及控制与执行模块。在上封与成品输出阶段，设置均匀光源并采用工业摄像头和光电传感器对图像信息进行捕捉获取，具体的视觉检测位置如图 4-1 所示；然后在计算单元中将获取图像信息进行去噪等图像预处理操作，并采用已研究的自立袋封装分类算法和无监督缺陷检测模型对输入图像进行缺陷的识别；最后，根据缺陷检测模块的识别结果，选择直接成品打包或废品剔除，如检测识别结果为“缺陷”，则控制单元将输出控制信号驱动电机带动摆臂，实现缺陷自立袋的剔除任务。自立袋封装缺陷检测系统架构及原理图，如图 4-4 所示。

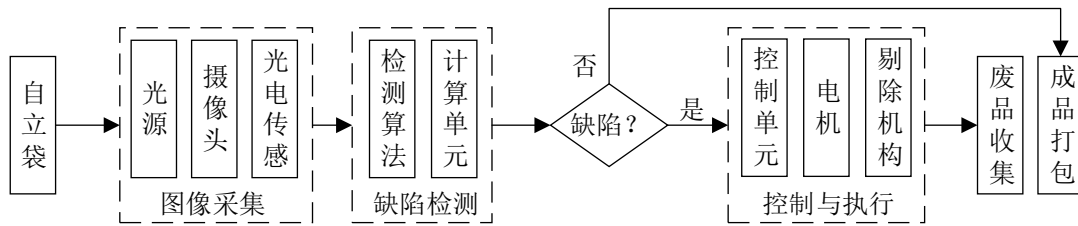


图 4-4 自立袋封装缺陷检测系统架构及原理图

图像采集模块相当于整个系统的眼睛，需要精心设置：首先，需要考虑自立袋的大小以及袋到摄像头之间的距离，对工业相机的分辨率与部署位置的选择提出了要求；其次，为保证拍摄图像不反光、袋形特征明显，需要为图像的捕捉设置均匀的光源，可以采用灯罩带有磨砂效果的环形 LED 光源作为照明设备；此外，为补充自立袋的位置信息，光电传感器也是必要的。

缺陷检测模块中最关键的设备为计算单元，在工业生产中，它通常由工业电脑代替。作为缺陷检测系统的信息处理中心，用于选取调用各种缺陷分类或分割模型对图像采集模块获取的图像进行缺陷检测，将得到的缺陷识别结果反馈到系统控制单元。

控制与执行模块中使用 PLC 控制器作为控制单元，当缺陷检测模块检测结果为“缺陷”时，PLC 控制器输出执行信号驱动电机控制剔除机构将缺陷自立袋剔除。

### 4.3 软件系统开发

软件开发是对缺陷检测系统信息的总结，本节分别从软件系统开发环境、软件工作流程和人机交互界面三个部分进行阐述。

#### 4.3.1 软件系统开发环境

自立袋封装缺陷检测软件系统基于 Python 语言开发。因软件面向包装设备厂商，使用数量不多，且考虑到 C/S 架构软件易于开发、理解和使用，同时具有强大数据操作处理能力的特性，因此，所设计的软件系统采用 C/S 体系架构。

操作系统界面是在 Pycharm 2021.3、Python3.8 的集成开发环境中，利用 PyQt5 和添加的 QTDesigner、PyUIC、PyRIC 等扩展工具进行设计。具体而言：①Qt Designer 直接通过选择和放置按钮、标签、文本框等 Qt 控件，设置控件属性和布局进行 GUI 界面的设计，仅使用少量手动编程代码进行补充设计，所设

计界面保存为 ui 格式文件；②PyUIC 将 ui 格式文件转化为包含各种组件和布局的 py 格式文件，方便在 Python 中使用和控制界面；③利用 PyRIC 将 PyQt5 的图像、图标以及样式表等资源文件整合到可执行文件中，简化系统界面的分发过程，提高应用程序性能与可移植性。此外，使用 sys、cv2、numpy 等软件库，搭建本地客户端，用于数据交互和显示界面。所使用的底层数据库是 MySQL。软件开发软硬件如表 4-1 所示。

表 4-1 软件开发环境

|    | 名称     | 说明                                |
|----|--------|-----------------------------------|
| 软件 | 操作系统   | Window10                          |
|    | 开发环境   | Pycharm 2021.3                    |
|    | 学习框架   | Pytorch                           |
|    | 扩展工具   | QTdesigner、PyUIC、PyRIC            |
|    | 环境依赖   | PyQt5                             |
|    | 架构依赖   | C/S                               |
| 硬件 | 处理器    | AMD Ryzen 7 5800H (3.20 GHz) 16GB |
|    | 显卡     | NVIDIA GeForce RTX 3070 8GB       |
|    | 硬盘空间   | 500G                              |
|    | USB 接口 | USB3.0 控制总线                       |

#### 4.3.2 软件工作流程

根据自立袋封装缺陷检测系统架构设计的缺陷检测软件，其工作流程图，如图 4-5 所示。首先，利用光电传感器判别是否检测到自立袋，如自立袋进入测试范围，则启动均匀光源并打开摄像头；其次，利用计算单元加载分类或分割缺陷检测模型，并进行缺陷检测；然后，判别检测对象是否存在缺陷，如无缺陷，则判断检测合格，检测结束；如存在缺陷，则判断检测不合格并激活控制单元，控制剔除机构将缺陷自立袋剔除；随后，利用光电传感器判断缺陷目标是否成功剔除，如剔除成功则检测完成，进程结束，如剔除失败则转入控制单元，再次输入控制信号，由 PLC 控制器驱动电机带动剔除机构将缺陷自立袋剔除。

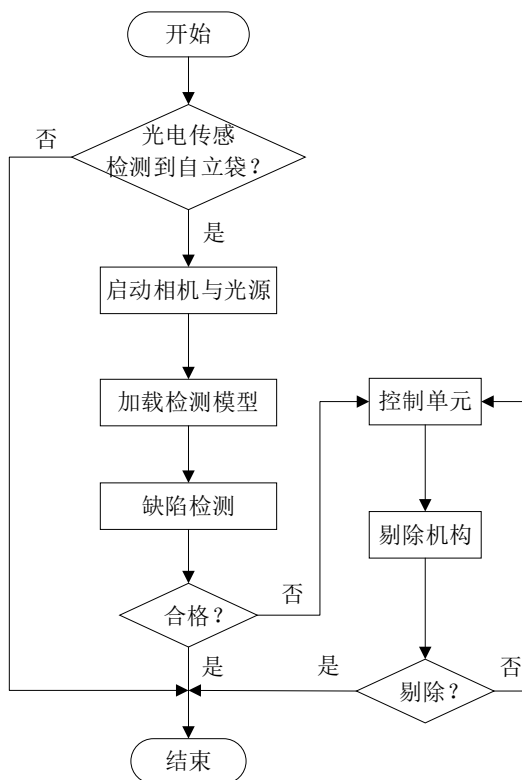


图 4-5 自立袋封装缺陷检测系统软件工作流程图

### 4.3.3 人机交互界面

#### 1) 注册界面与登录界面

注册界面与登录界面的设置是为了保护自立袋缺陷检测系统的用户信息，非用户授权人员不能使用。注册界面和登录界面是进入系统前的开始界面，两种界面均设置同一封面，在右侧界面点击注册或登录，右侧界面上半部分分别跳至注册或登录信息页。两界面信息页不同的是，注册页需要再次“确认密码”，而登录页仅需输入一次密码。注册界面与登录界面如图 4-6、图 4-7 所示。



图 4-6 自立袋封装缺陷检测系统软件注册界面



图 4-7 自立袋封装缺陷检测系统软件登录界面

## 2) 缺陷检测界面

在用户注册登录后，软件进入缺陷检测界面。该界面包含检测模型、检测模式与个人信息的设置，以及检测视图和缺陷识别结果的展示。自立袋封装缺陷检测系统检测模型的选择如图 4-8 所示。



图 4-8 自立袋封装缺陷检测系统检测模型选择

缺陷检测界面嵌入改进 MobileNetV3、MobileNetV3、EfficientNetV2-s、ShuffleNetV2-x1 以及 ResNet18 等分类模型，用于对自立袋封装缺陷的检测。软件具有多种缺陷检测模式，除了用于包装设备调用工业相机实现的实时检测模式外，还包括用于工控机存储的单张图像、图像文件夹，以及缺陷检测的视频模式。自立袋封装缺陷检测系统单张图像检测效果的演示，如图 4-9 所示。点击检测模式中“打开图片”按钮，软件系统打开本地文件浏览，在选择待检测

的自立袋样本后，软件系统将待检测图像在检测界面“视图”窗格中显示，并根据所选择的检测模型，快速调用改进的 MobileNetV3 分类模型，对所选待检测样本进行分类检测，并将分类预测类别以及准确率结果，在检测界面“识别结果”中显示。自立袋封装缺陷检测软件调用的改进 MobileNetV3 模型代码见附录。



(a) M形未对齐缺陷检测示例



(b) M形严重褶皱缺陷检测示例

图 4-9 自立袋封装缺陷检测系统检测效果演示

#### 4.4 本章小结

本章以 DS180-DS 型号包装设备自立袋生产流程为研究主线，在明确自立袋封装缺陷检测系统设计需求的基础上，在包装设备上封与成品输出阶段引入均匀光源、工业相机与光电传感等图像信息采集装置和电机驱动的摆臂剔除机构，设计自立袋封装缺陷检测系统架构；然后根据系统架构设计软件的工作流程，使用 Pycharm 集成开发环境搭建软件注册登录以及缺陷检测的交互界面。

## 第 5 章 总结与展望

### 5.1 总结

自立袋作为当前食品、日化以及医药等领域的主要包装形式，受众范围大，传播范围广，为避免长途运输对自立袋表面及袋内物料的不良影响，确保产品的安全性，在产品生产端检测并剔除缺陷自立袋以提高产品良品率成为问题的关键。本文首先针对封装缺陷检测中准确率较低的问题，采用监督学习的分类模型研究自立袋封装缺陷检测；其次针对监督学习缺陷样本较少以及模型泛化性能较低的问题，研究一种基于 MB 正负样本差异特征自立袋封装缺陷检测算法；最后，根据所研究缺陷检测算法和自立袋包装设备，设计自立袋封装缺陷检测系统，开发系统软件交互界面。具体研究内容如下：

1) 研究了一种监督学习自立袋缺陷分类检测算法。首先从工厂实地采集 DS180-DS 包装设备在调试过程产生的缺陷样本，经过筛检、压缩旋转、明暗变换等图像预处理操作构建自立袋封装缺陷检测分类数据集；然后详述分类网络模型结构；在此基础上，从引入 GELU 激活函数、CA 注意力机制着手，对 MobileNetV3 网络进行改进，并探讨深度卷积在 Bneck 块中不同位置的影响；最终，经过消融实验和对比实验验证改进方法的有效性和先进性。

2) 探讨了一种无监督学习自立袋缺陷分割检测算法。利用 EISeg 软件获取缺陷样本的分割掩码，构建 Doypack-Defects 分割数据集；总结无监督检测中 MB 方法的网络结构，采用基于 MB 方法 U-Net 网络结构模型为基底模型；利用负样本合成模块、特征存储模块、注意力特征融合模块以及预测模块 4 个模块搭建一种基于 MB 正负样本差异特征的无监督自立袋封装缺陷检测模型；在 Doypack-Defects 数据集上进行消融对比实验研究，在 MVTec AD 纹理数据集上进行模型泛化能力的研究，结果显示所提模型具有实际应用价值。

3) 设计了自立袋封装缺陷检测系统。在明确系统设计目标与要求的基础上，调研 DS180-DS 型包装设备工作流程，研究自立袋封装缺陷检测系统架构，在包装机器上封与成品输出阶段添加图像采集设备和成品缺陷剔除设备。然后，利用 Qt 框架进行软件系统设计，分别搭建系统注册登录与检测界面，并调用所研究的自立袋封装缺陷检测分类模型对检测效果进行可视化。

## 5.2 展望

本文从监督学习和无监督学习两个角度，利用自制数据集对自立袋封口缺陷检测进行研究，并开发缺陷检测软件。虽然取得了一定的成效，但是尚有值得改进之处需要深入剖析探讨，如：

- 1) 自制的自立袋封装缺陷检测分类数据集中，采用数据增强扩充数据集以弥补数据量采集不足的问题，可能会导致数据同质化严重；
- 2) 监督学习策略难以收集和覆盖全部缺陷样本，导致监督学习的模型在泛化能力上有待提升；
- 3) 无监督学习策略在检测中易受数据分布的影响，若数据分布发生较大变化，需要调整缺陷检测模型以重新适应数据分布；
- 4) 针对自立袋缺陷的检测，仅涉及包装的上封流程，并没有考虑包装的竖封流程和裁剪流程所出现的缺陷。

## 参考文献

- [1] 中国包装行业年度运行报告(2022)[R]. 北京: 中国包装联合会, 2023.
- [2] 侯茗萱, 李家豪, 东为富等. 食品生态包装材料的研究进展[J]. 塑料包装, 2023, 33(04): 1-6+49.
- [3] 康军凯, 潘晓军, 茅周祎等. 软包装自立袋在食品行业中的应用和发展[J]. 饮料工业, 2014, 17(10): 49-52.
- [4] 戈明辉, 张俊, 陆慧娟. 基于机器视觉的食品外包装缺陷检测算法研究进展[J]. 食品与机械, 2023, 39(09): 95-102+116.
- [5] 程锦锋, 方贵盛, 高惠芳. 表面缺陷检测的机器视觉技术研究进展[J]. 计算机应用研究, 2023, 40(04): 967-977.
- [6] Kuang H L, Ding Y R, Li R F, Liu X H. Defect detection of bamboo strips based on LBP and GLCM features by using SVM classifier[C]//2018 Chinese Control And Decision Conference (CCDC), IEEE, Shenyang, China, 09-11 June, 2018: 3341-3345.
- [7] 孙浩然, 李雅雯, 韩有军等. 基于拓扑滤波与改进 Canny 算子的锂离子电池电极缺陷检测[J]. 储能科学与技术, 2022, 11(10): 3297-3305.
- [8] 刘晓阳, 刘晶, 张向阳等. 基于二维 Gabor 滤波器的胶带撕裂检测[J]. 工矿自动化, 2021, 47(04): 103-107.
- [9] 吕宁, 肖剑, 高健等. 基于改进 MRF 的冲压件轮廓缺陷图像分割算法[J]. 锻压技术, 2022, 47(04): 101-109.
- [10] Jia L, Chen C, Xu S, et al. Fabric defect inspection based on lattice segmentation and template statistics[J]. Information Sciences, 2020, 512: 964-984.
- [11] 胡跃明, 李璐, 罗家祥. 高密度柔性集成电路基板的表面氧化缺陷检测[J]. 计算机集成制造系统, 2020, 26(09): 2379-2387.
- [12] 谢为俊, 魏硕, 王凤贺等. 基于机器视觉的胡萝卜表面缺陷识别方法研究[J]. 农业机械学报, 2020, 51(S1): 450-456.
- [13] 包从望, 胡才梦, 张彩虹等. 基于改进 Canny 算子的齿轮缺陷检测边缘检测算法[J]. 组合机床与自动化加工技术, 2023, (01): 83-86+91.
- [14] 赵民, 武晓龙. 石材表面的线类缺陷检测[J]. 现代制造工程, 2017, (02): 113-116.
- [15] Pasadas D J, Baskaran P, Ramos H G, et al. Detection and classification of defects using ECT and multi-level SVM model[J]. IEEE Sensors Journal, 2019, 20(5): 2329-2338.
- [16] 于一深, 苏宇锋, 高建设. 基于 KNN 分类的涂胶质量检测算法[J]. 组合机床与自动化加工技术, 2023, (07): 127-130.
- [17] 柴子凡, 赵丽琴, 蒋海波等. 基于 Hough 变换和 K-means 聚类的玻璃瓶分模线缺陷检测[J]. 机械设计与研究, 2023, 39(03): 171-174.
- [18] 邓泉, 叶蔚, 谢睿等. 基于深度学习的源代码缺陷检测研究综述[J]. 软件学报, 2023, 34(02): 625-654.
- [19] 方宇伦, 陈雪纯, 杜世昌等. 基于轻量化深度学习 VGG16 网络模型的表面缺陷检测方法

- [J]. 机械设计与研究, 2023, 39(02): 143-147.
- [20] Apostolopoulos I D, Tzani M A. Industrial object and defect recognition utilizing multilevel feature extraction from industrial scenes with Deep Learning approach[J]. Journal of Ambient Intelligence and Humanized Computing, 2023, 14(8): 10263-10276.
- [21] 薛勇, 王立扬, 张瑜等. 基于 GoogLeNet 深度迁移学习的苹果缺陷检测方法[J]. 农业机械学报, 2020, 51(07): 30-35.
- [22] Gong Y, Luo J, Shao H, et al. Automatic defect detection for small metal cylindrical shell using transfer learning and logistic regression[J]. Journal of Nondestructive Evaluation, 2020, 39: 1-13.
- [23] He Y, Song K, Meng Q, et al. An end-to-end steel surface defect detection approach via fusing multiple hierarchical features[J]. IEEE transactions on instrumentation and measurement, 2019, 69(4): 1493-1504.
- [24] 夏禹, 肖金球, 翁玉尚. 基于改进 Faster-RCNN 的偏光片表面缺陷检测[J]. 光学技术, 2021, 47(06): 695-702.
- [25] 马美荣, 李东喜. 基于 RetinaNet 的手机主板缺陷检测研究[J]. 计算机工程与科学, 2020, 42(04): 673-682.
- [26] Zhang J, Xu J, Zhu L, et al. An improved MobileNet-SSD algorithm for automatic defect detection on vehicle body paint[J]. Multimedia Tools and Applications, 2020, 79: 23367-23385.
- [27] 李建明, 杨挺, 王惠栋. 基于深度学习的工业自动化包装缺陷检测方法[J]. 包装工程, 2020, 41(07): 175-184.
- [28] 卢俊哲, 张铖怡, 刘世鹏等. 面向复杂环境中带钢表面缺陷检测的轻量级 DCN-YOLO[J]. 计算机工程与应用, 2023, 59(15): 318-328.
- [29] 王陶然, 王明泉, 张俊生等. 基于 Mask R-CNN 的轮毂缺陷分割技术[J]. 国外电子测量技术, 2021, 40(02): 1-5.
- [30] 李原, 李燕君, 刘进超等. 基于改进 Res-UNet 网络的钢铁表面缺陷图像分割研究[J]. 电子与信息学报, 2022, 44(05): 1513-1520.
- [31] Huang Y, Jing J, Wang Z. Fabric Defect Segmentation Method Based on Deep Learning[J]. IEEE Transactions on Instrumentation and Measurement, 2021, 70: 1-15.
- [32] 周倩倩, 刘汉林, 陈维锋等. 基于 Deeplabv3+ 的排水管道缺陷检测与语义分割[J]. 中国给水排水, 2022, 38(13): 22-27.
- [33] Jiang H, Zhang X, Tao C, et al. Casting defect region segmentation method based on dual-channel encoding-fusion decoding network[J]. Expert Systems with Applications, 2024, 247: 123254-12370.
- [34] 袁非牛, 章琳, 史劲亭等. 自编码神经网络理论及应用综述[J]. 计算机学报, 2019, 42(01): 203-230.
- [35] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks[J]. Communications of the ACM, 2020, 63(11): 139-144.

- [36] Papamakarios G, Nalisnick E, Rezende D J, et al. Normalizing flows for probabilistic modeling and inference[J]. *Journal of Machine Learning Research*, 2021, 22(57): 1-64.
- [37] Bergmann P, Fauser M, Sattlegger D, et al. Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings[C]//*Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*. IEEE, Seattle, WA, USA, 16-21 June, 2020: 4183-4192.
- [38] Ruff L, Vandermeulen R, Goernitz N, et al. Deep one-class classification[C]// *Proceedings of the 35th International Conference on Machine Learning (ICML)*. PMLR, Stockholm, Sweden, 10-15 July, 2018: 4393-4402.
- [39] Peng P, Li Z. Visual anomaly detection via partition memory bank module and error estimation[J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2023, 33(8): 3596-3607.
- [40] 李毅, 许知建, 王逸涵等. 基于分块重构 GAN 的铁路隧道漏缆卡扣缺陷检测[J]. *铁道标准设计*, 2024, 68(02): 176-184.
- [41] 常江, 管声启, 师红宇等. 基于改进生成对抗网络和 MobileNetV3 的带钢缺陷分类[J]. *激光与光电子学进展*, 2021, 58(04): 221-226.
- [42] Tsai D M, Fan S K S, Chou Y H. Auto-Annotated Deep Segmentation for Surface Defect Detection[J]. *IEEE Transactions on Instrumentation and Measurement*, 2021, 70: 1-10.
- [43] 唐善成, 陈明, 王瀚博等. 采用变分自编码器的无监督压敏电阻表面缺陷检测[J]. *计算机集成制造系统*, 2022, 28(05): 1337-1351.
- [44] 代朝东, 许国良, 毛骄等. 基于无监督网络的手机屏幕缺陷分割方法[J]. *激光与光电子学进展*, 2021, 58(20): 375-384.
- [45] Zavrtanik V, Kristan M, Skočaj D. Draem-a discriminatively trained reconstruction embedding for surface anomaly detection[C]//*Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Montreal, QC, Canada, 10-17 October, 2021: 8330-8339.
- [46] Rudolph M, Wehrbein T, Rosenhahn B, et al. Fully convolutional cross-scale-flows for image-based defect detection[C]//*Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Waikoloa, Hawaii, USA, 3-8 January, 2022: 1088-1097.
- [47] Wu H, Köhler J, Noé F. Stochastic normalizing flows[J]. *Advances in Neural Information Processing Systems*, 2020, 33: 5933-5944.
- [48] Wang G D, Han S M, Ding E, et al. Student-Teacher Feature Pyramid Matching for Anomaly Detection[C]//*British Machine Vision Conference (BMVC)*, BMVA, Virtual, 22-25, November, 2021: 1-14
- [49] Cerqueira V, Gomes H M, Bifet A, et al. STUDD: A student-teacher method for unsupervised concept drift detection[J]. *Machine Learning*, 2023, 112(11): 4351-4378.
- [50] 杨旭东. 基于深度学习的烟用胶囊外观缺陷检测技术研究 [D]. 中国科学技术大学, 2023.

- [51] 李亚舟, 曹江涛, 姬晓飞. 融合改进 Padim 建模和 ResNet 网络的喷涂质量检测算法[J]. 电子测量与仪器学报, 2022, 36(11): 91-97.
- [52] Roth K, Pemula L, Zepeda J, et al. Towards total recall in industrial anomaly detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, New Orleans, LA, USA, 19-24 June, 2022: 14318-14328.
- [53] MemSeg: A semi-supervised method for image surface defect detection using differences and commonalities[J]. Engineering Applications of Artificial Intelligence, 2023, 119: 105835.
- [54] Wu Z, Shen C, Van Den Hengel A. Wider or deeper: Revisiting the resnet model for visual recognition[J]. Pattern recognition, 2019, 90: 119-133.
- [55] Sinha D, El-Sharkawy M. Thin mobilenet: An enhanced mobilenet architecture[C]//2019 IEEE 10th annual ubiquitous computing, electronics & mobile communication conference (UEMCON). IEEE, New York, USA, 10-12 October, 2019: 0280-0285.
- [56] Zhang X, Zhou X, Lin M, et al. Shufflenet: An extremely efficient convolutional neural network for mobile devices[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Salt Lake City, USA, 18-12 June, 2018: 6848-6856.
- [57] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks[C]//International conference on machine learning (ICML). PMLR, Los Angeles, USA, 9-15 June, 2019: 6105-6114.
- [58] Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]//Proceedings of the IEEE/CVF international conference on computer vision (ICCV). IEEE, Seoul, Korea (South), 27 October -2 November, 2019: 1314-1324.
- [59] 胡斌汉, 李曙. 基于改进 ShuffleNetV2 的轻量级花色布匹瑕疵检测[J]. 计算机系统应用, 2023, 32(04): 161-169.
- [60] Hou Q, Zhou D, Feng J. Coordinate attention for efficient mobile network design[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. IEEE, Nashville, TN, USA, 20-25 June, 2021:13708-13717.
- [61] 沈希忠, 谢旭. 带钢表面缺陷的 RepVGG 网络改进及其识别[J]. 现代制造工程, 2023, (05): 121-126.
- [62] Liu Z, Lin Y, Cao Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows[C]//Proceedings of the IEEE/CVF international conference on computer vision (ICCV). IEEE, USA, 10-17 March, 2021: 10012-10022.
- [63] Liu Z, Mao H, Wu C Y, et al. A convnet for the 2020s[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, New Orleans, USA, 21-24 June, 2022: 11976-11986.
- [64] Bergmann P, Fauser M, Sattlegger D, et al. MVTec AD--A comprehensive real-world dataset for unsupervised anomaly detection[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR). IEEE, Long Beach, USA, 18-20 June, 2019: 9592-9600.

- [65] 来杰, 王晓丹, 向前等. 自编码器及其应用综述[J]. 通信学报, 2021, 42(09): 218-230.
- [66] Siddique N, Paheding S, Elkin C P, et al. U-net and its variants for medical image segmentation: A review of theory and applications[J]. Ieee Access, 2021, 9: 82031-82057
- [67] 黄梦涛, 连一鑫. 基于改进 Canny 算子的锂电池极片表面缺陷检测[J]. 仪器仪表学报, 2021, 42(10): 199-209.
- [68] Cimpoi M, Maji S, Kokkinos I, et al. Describing textures in the wild[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Columbus, USA, 24-27 June, 2014: 3606-3613.
- [69] Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE, Salt Lake City, USA, 18-12 June, 2018: 8759-8768.
- [70] Cohen N, Hoshen Y. Sub-image anomaly detection with deep pyramid correspondences[J]. arXiv preprint arXiv:2005.02357, 2020: 1-17.
- [71] Yi J, Yoon S. Patch svdd: Patch-level svdd for anomaly detection and segmentation[C]//Proceedings of the Asian conference on computer vision (CVPR). IEEE, USA, 14-19 June, 2020: 1-12.

## 附录

改进 MobileNetV3 模型部分代码如下:

```
from typing import Callable, List, Optional

import torch

from torch import nn, Tensor, tanh

import numpy as np

from torch.nn import functional as F

from functools import partial

from torch.hub import load_state_dict_from_url

__all__ = ['mobilenet_v3_large', 'mobilenet_v3_small']

model_urls = {
    'mobilenet_v3_large': 'https://download.pytorch.org/models/mobilenet_v3_large-8738ca79.pth',
    'mobilenet_v3_small': 'https://download.pytorch.org/models/mobilenet_v3_small-047dcff4.pth',
}

class h_sigmoid(nn.Module):
    def __init__(self, inplace=True):
        super(h_sigmoid, self).__init__()
        self.relu = nn.ReLU6(inplace=inplace)

    def forward(self, x):
        return self.relu(x + 3) / 6

class h_swish(nn.Module):
    def __init__(self, inplace=True):
        super(h_swish, self).__init__()
        self.sigmoid = h_sigmoid(inplace=inplace)

    def forward(self, x):
        return x * self.sigmoid(x)
```

```
class CoordAtt(nn.Module):  
    def __init__(self, inp, oup, reduction=32):  
        super(CoordAtt, self).__init__()  
        self.pool_h = nn.AdaptiveAvgPool2d((None, 1))  
        self.pool_w = nn.AdaptiveAvgPool2d((1, None))  
        mip = max(8, inp // reduction)  
        self.conv1 = nn.Conv2d(inp, mip, kernel_size=1, stride=1, padding=0)  
        self.bn1 = nn.BatchNorm2d(mip)  
        self.act = h_swish()  
        self.conv_h = nn.Conv2d(mip, oup, kernel_size=1, stride=1, padding=0)  
        self.conv_w = nn.Conv2d(mip, oup, kernel_size=1, stride=1, padding=0)  
  
    def forward(self, x):  
        identity = x  
        n, c, h, w = x.size()  
        x_h = self.pool_h(x)  
        x_w = self.pool_w(x).permute(0, 1, 3, 2)  
        y = torch.cat([x_h, x_w], dim=2)  
        y = self.conv1(y)  
        y = self.bn1(y)  
        y = self.act(y)  
        x_h, x_w = torch.split(y, [h, w], dim=2)  
        x_w = x_w.permute(0, 1, 3, 2)  
        a_h = self.conv_h(x_h).sigmoid()  
        a_w = self.conv_w(x_w).sigmoid()  
        out = identity * a_w * a_h  
  
    return out
```

```

class GELU(nn.Module):
    def __init__(self):
        super(GELU, self).__init__()

    def forward(self, x):
        return 0.5 * x * (1 + torch.tanh(np.sqrt(2 / np.pi) * (x + 0.044715 * torch.pow(x,3))))

class Partial_conv3(nn.Module)

    def __init__(self, dim, n_div, forward):
        super().__init__()

        self.dim_conv3 = dim // n_div

        self.dim_untouched = dim - self.dim_conv3

        self.partial_conv3 = nn.Conv2d(self.dim_conv3, self.dim_conv3, 3, 1, 1, bias=False)

        if forward == 'slicing':
            self.forward = self.forward_slicing

        elif forward == 'split_cat':
            self.forward = self.forward_split_cat

        else:
            raise NotImplementedError

    def forward_slicing(self, x: Tensor) -> Tensor:
        # only for inference

        x = x.clone() # !!! Keep the original input intact for the residual connection later

        x[:, :self.dim_conv3, :, :] = self.partial_conv3(x[:, :self.dim_conv3, :, :])

        return x

    def forward_split_cat(self, x: Tensor) -> Tensor:
        # for training/inference

        x1, x2 = torch.split(x, [self.dim_conv3, self.dim_untouched], dim=1)

        x1 = self.partial_conv3(x1)

```

```
x = torch.cat((x1, x2), 1)

return x


def _make_divisible(ch, divisor=8, min_ch=None):
    if min_ch is None:
        min_ch = divisor
    new_ch = max(min_ch, int(ch + divisor / 2) // divisor * divisor)
    # Make sure that round down does not go down by more than 10%.
    if new_ch < 0.9 * ch:
        new_ch += divisor
    return new_ch


class ConvBNActivation(nn.Sequential):
    def __init__(self,
        in_planes: int,
        out_planes: int,
        kernel_size: int = 3,
        stride: int = 1,
        groups: int = 1,
        norm_layer: Optional[Callable[..., nn.Module]] = None,
        activation_layer: Optional[Callable[..., nn.Module]] = None):
        padding = (kernel_size - 1) // 2
        if norm_layer is None:
            norm_layer = nn.BatchNorm2d
        if activation_layer is None:
            activation_layer = nn.ReLU6
        super(ConvBNActivation, self).__init__(nn.Conv2d(in_channels=in_planes,
            out_channels=out_planes,
            kernel_size=kernel_size,
            stride=stride,
```

```
padding=padding,  
groups=groups,  
bias=False),  
norm_layer(out_planes),  
activation_layer())
```

```
class SqueezeExcitation(nn.Module):  
    def __init__(self, input_c: int, squeeze_factor: int = 4):  
        super(SqueezeExcitation, self).__init__()  
        squeeze_c = _make_divisible(input_c // squeeze_factor, 8)  
        self.fc1 = nn.Conv2d(input_c, squeeze_c, 1)  
        self.fc2 = nn.Conv2d(squeeze_c, input_c, 1)  
  
    def forward(self, x: Tensor) -> Tensor:  
        scale = F.adaptive_avg_pool2d(x, output_size=(1, 1))  
        scale = self.fc1(scale)  
        scale = F.relu(scale, inplace=True)  
        scale = self.fc2(scale)  
        scale = F.hardsigmoid(scale, inplace=True)  
        return scale * x
```

```
class InvertedResidualConfig:
```

```
    def __init__(self,  
        input_c: int,  
        kernel: int,  
        expanded_c: int,  
        out_c: int,  
        use_ca: bool,  
        activation: str,  
        stride: int,
```

```
        width_multi: float):

    self.input_c = self.adjust_channels(input_c, width_multi)

    self.kernel = kernel

    self.expanded_c = self.adjust_channels(expanded_c, width_multi)

    self.out_c = self.adjust_channels(out_c, width_multi)

    self.use_ca = use_ca

    self.use_hs = activation == "HS" # whether using h-swish activation

    self.stride = stride


@staticmethod
def adjust_channels(channels: int, width_multi: float):

    return _make_divisible(channels * width_multi, 8)


class InvertedResidual(nn.Module):

    def __init__(self,

                 cnf: InvertedResidualConfig,

                 norm_layer: Callable[..., nn.Module]):

        super(InvertedResidual, self).__init__()

        if cnf.stride not in [1, 2]:

            raise ValueError("illegal stride value.")

        self.use_res_connect = (cnf.stride == 1 and cnf.input_c == cnf.out_c)

        layers: List[nn.Module] = []

        activation_layer = nn.Hardswish if cnf.use_hs else nn.ReLU

        # depthwise

        if cnf.expanded_c != cnf.input_c:

            layers.append(ConvBNActivation(cnf.input_c,
```

```
        cnf.input_c,
        kernel_size=cnf.kernel,
        stride=cnf.stride,
        groups=cnf.input_c,
        norm_layer=norm_layer,
        activation_layer=activation_layer))

# expand
layers.append(ConvBNActivation(cnf.input_c,
                               cnf.expanded_c,
                               kernel_size=1,
                               # norm_layer=norm_layer,
                               activation_layer=activation_layer))

if cnf.use_ca:
    layers.append(CoordAtt(cnf.expanded_c, cnf.expanded_c))

# project
layers.append(ConvBNActivation(cnf.expanded_c,
                               cnf.out_c,
                               kernel_size=1,
                               norm_layer=norm_layer,
                               activation_layer=nn.Identity
                               ))

self.block = nn.Sequential(*layers)

self.out_channels = cnf.out_c

self.is_strided = cnf.stride > 1

def forward(self, x: Tensor) -> Tensor:
```

```
result = self.block(x)

if self.use_res_connect:
    result += x

return result
```

```
class MobileNetV3(nn.Module):
    def __init__(self,
                  inverted_residual_setting: List[InvertedResidualConfig],
                  last_channel: int,
                  num_classes: int = 1000,
                  block: Optional[Callable[..., nn.Module]] = None,
                  norm_layer: Optional[Callable[..., nn.Module]] = None):
        super(MobileNetV3, self).__init__()

        if not inverted_residual_setting:
            raise ValueError("The inverted_residual_setting should not be empty.")
        elif not (isinstance(inverted_residual_setting, List) and
                  all([isinstance(s, InvertedResidualConfig) for s in inverted_residual_setting])):
            raise TypeError("The inverted_residual_setting should be List[InvertedResidualConfig]")

        if block is None:
            block = InvertedResidual

        if norm_layer is None:
            norm_layer = partial(nn.BatchNorm2d, eps=0.001, momentum=0.01)

        layers: List[nn.Module] = []

        # building first layer
        firstconv_output_c = inverted_residual_setting[0].input_c
        layers.append(ConvBNActivation(3,
                                       firstconv_output_c,
                                       kernel_size=3,
                                       stride=2,
```

```
        norm_layer=norm_layer,
        activation_layer=nn.Hardswish))

# building inverted residual blocks
for cnf in inverted_residual_setting:
    layers.append(block(cnf, norm_layer))

# building last several layers
lastconv_input_c = inverted_residual_setting[-1].out_c
lastconv_output_c = 6 * lastconv_input_c
layers.append(ConvBNActivation(lastconv_input_c,
                                lastconv_output_c,
                                kernel_size=1,
                                norm_layer=norm_layer,
                                activation_layer=nn.Hardswish))

self.features = nn.Sequential(*layers)
self.avgpool = nn.AdaptiveAvgPool2d(1)
self.classifier = nn.Sequential(nn.Linear(lastconv_output_c, last_channel),
                                nn.Hardswish(inplace=True),
                                nn.Dropout(p=0.2, inplace=True),
                                nn.Linear(last_channel, num_classes))

# initial weights
for m in self.modules():
    if isinstance(m, nn.Conv2d):
        nn.init.kaiming_normal_(m.weight, mode="fan_out")
        if m.bias is not None:
            nn.init.zeros_(m.bias)
    elif isinstance(m, (nn.BatchNorm2d, nn.GroupNorm)):
        nn.init.ones_(m.weight)
        nn.init.zeros_(m.bias)
    elif isinstance(m, nn.Linear):
        nn.init.normal_(m.weight, 0, 0.01)
```

```
nn.init.zeros_(m.bias)
```

```
def _forward_impl(self, x: Tensor) -> Tensor:
```

```
    x = self.features(x)
```

```
    x = self.avgpool(x)
```

```
    x = torch.flatten(x, 1)
```

```
    x = self.classifier(x)
```

```
    return x
```

```
def forward(self, x: Tensor) -> Tensor:
```

```
    return self._forward_impl(x)
```

```
def mobilenet_v3_small(pretrained=False, progress=True, num_classes: int = 1000,
```

```
                        reduced_tail: bool = False) -> MobileNetV3:
```

```
    width_multi = 1.0
```

```
    bneck_conf = partial(InvertedResidualConfig, width_multi=width_multi)
```

```
    adjust_channels = partial(InvertedResidualConfig.adjust_channels, width_multi=width_multi)
```

```
    reduce_divider = 2 if reduced_tail else 1
```

```
    inverted_residual_setting = [
```

```
        # input_c, kernel, expanded_c, out_c, use_se, activation, stride
```

```
        bneck_conf(16, 3, 16, 16, True, "RE", 2), # C1
```

```
        bneck_conf(16, 3, 72, 24, False, "RE", 2), # C2
```

```
        bneck_conf(24, 3, 88, 24, False, "RE", 1),
```

```
        bneck_conf(24, 5, 96, 40, True, "HS", 2), # C3
```

```
        bneck_conf(40, 5, 240, 40, True, "HS", 1),
```

```
        bneck_conf(40, 5, 240, 40, True, "HS", 1),
```

```
        bneck_conf(40, 5, 120, 48, True, "HS", 1),
```

```
        bneck_conf(48, 5, 144, 48, True, "HS", 1),
```

```

        bneck_conf(48, 5, 288, 96 // reduce_divider, True, "HS", 2), # C4
        bneck_conf(96 // reduce_divider, 5, 576 // reduce_divider, 96 // reduce_divider, True, "HS",
1),
        bneck_conf(96 // reduce_divider, 5, 576 // reduce_divider, 96 // reduce_divider, True, "HS", 1)
    ]
    last_channel = adjust_channels(1024 // reduce_divider)
    lastconv_input_c = inverted_residual_setting[-1].out_c # C5
    lastconv_output_c = 6 * lastconv_input_c
    model = MobileNetV3(inverted_residual_setting=inverted_residual_setting, last_channel =
last_channel)
    if pretrained:
        state_dict = load_state_dict_from_url(model_urls['mobilenet_v3_small'], model_dir =
'./model_data', progress=progress)
        model.load_state_dict(state_dict)

    if num_classes!=1000:
        model.classifier = nn.Sequential(nn.Linear(lastconv_output_c, last_channel),
            nn.Hardswish(inplace=True),
            nn.Dropout(p=0.2, inplace=True),
            nn.Linear(last_channel, num_classes))

    return model

```

## 攻读学位期间发表的学术论文

- [1] **高港**, 魏利胜, 唐绍语. 一种基于改进 YOLOv5 单目测距方法研究[J]. 安徽工程大学学报, 2023, 38(01): 42-48. (三类)
- [2] **Gao Gang**, Wei Lisheng, Hu Quangan. Research on Image Lightweight Binocular Ranging[C] // 2023 3rd International Conference on Computer, Control and Robotics (ICCCR). IEEE, ShangHai, China, 24-26 March, 2023: 82-86. (EI: 20233414619462)
- [3] Xiang Shiyu, Wei Lisheng, **Gao Gang**. Colonic Polyp Segmentation Algorithm Based on Improved SegFormer Network[C] // 2023 3rd International Conference on Computer, Control and Robotics (ICCCR). IEEE, ShangHai, China, 24-26 March, 2023: 101-105. (EI: 20233414619461)
- [4] Zhu Shengbo, **Gao Gang**, Wei Lisheng. Classification Detection of Packaging Bag Defects Based on Improved MobileNetV3[C] // 2024 14nd Asian Control Conference (ASCC). (EI, 已录用)
- [5] **高港**, 魏利胜, 朱圣博. 正负样本差异特征双径向融合的自监督缺陷检测方法[J]. 电子测量与仪器学报(CSCD, 已录用: B2307064)
- [6] 朱圣博, 魏利胜, **高港**, 郑泊文. 基于改进 YOLOv8 的光学遥感小型船舶检测算法 [J]. 电子测量与仪器学报(CSCD, 已投稿: B2407348)

## 攻读学位期间获得的奖励

- [1] 获得 2023-2024 学年校研究生一等奖学金
- [2] 获得 2022-2023 学年校研究生一等奖学金
- [3] 获得 2021-2022 学年校研究生三等奖学金
- [4] 获得 2021-2022 年度“优秀学生干部”称号
- [5] 获得 2022 年校“优秀共产党员”称号
- [6] 获得 2023 年中国传感器创新创业大赛(华东赛区)一等奖
- [7] 获得 2023 年中国传感器创新创业大赛(全国)三等奖
- [8] 获得 2022 年校第八届“互联网+”大学生创新创业大赛研究生创意组三等奖

## 致 谢

丝绦嫩芽又绿，新燕暖春还来。行文至此，我的毕业论文即将画上句点，三年的研究生生涯即将落下帷幕，回望过去，一丝欣喜，几多感慨，更多的是感谢。

随风潜入夜，润物细无声。首先我要感谢的是导师魏利胜教授和校外导师刘小琿老师。三年来，魏老师一直是律人先律己，教导我们身心健康是第一位的，给我们在科研上树立求真务实、科学严谨的学习态度，交给我们跟踪学术前沿的方法、总结学术文献的方式，与之交流，总是能够引导我关注到关键所在，寥寥数语而又能拨云见日，找到解决问题的方向；在实习上，感谢魏老师为我争取到论文课题相关的实习机会，帮助我将毕业课题推进下去的同时，补贴生活；在就业上，感谢魏老师从个人规划、城市发展以及时代潮流等方面帮助我下定决心选择 offer；感谢魏老师三年来对我无微不至的关怀和教导。

风华正茂，友谊长存。其次，我要感谢 509 实验室所有同学。感谢张倩、汪瑞、马晓东师兄师姐给予我学习和就业方向的指导；感谢项诗雨、唐绍语、刘瑞、王瑞仁同门对我的督促和提携；感谢朱圣博、陶新杰、李明、王勤等师弟师妹的真诚陪伴。星光不负赶路人，愿所有同学都能学业有成，前程似锦。

哀哀父母，生我劬劳。感谢我的父母，您们赐我生命，伴我成长，给我无忧无虑的学习和生活提供依靠，爱的无私而无言，没有一份付出可以被忘记，也没有一份恩情可以被辜负。

最后，对在百忙之中评审这篇论文的各位专家学者致以特别真诚的感谢。

高港

安徽工程大学电气工程学院

二〇二四年三月

尚德敏学 唯实惟新

