基于AI的图像识别技术在UI自动化测试中的应用研究

杨 欢

（南京市产品质量监督检验院（南京市质量发展与先进技术应用研究院），江苏 南京 210019）

摘 要 随着移动互联网与5G/6G网络的深度融合，传统基于DOM定位的UI测试方法在跨平台、跨网络环境下面临维护成本高、适应性差等技术瓶颈。通过对YOLOv8目标检测算法进行优化，引入多尺度特征金字塔与深度可分离卷积的轻量化架构，结合SSIM相似度计算的状态转换机制，提出视觉驱动测试技术。该技术优于现有主流方法，在多个企业产品部署中显著降低测试成本并提高缺陷检出率，为AI技术在软件测试领域的应用提供参考。

关键词 UI自动化测试；图像识别；YOLOv8；深度学习；跨平台测试

中图分类号：TP181；TP311.53 **文献标志码**：A **DOI**：10.3969/j.issn.2097-3365.2026.03.011

0 引言

随着5G/6G通信技术的商用部署与移动互联网应用的深度融合，软件产品迭代周期从月度缩短至周级，在多网络环境、多终端形态的复杂测试场景下，UI自动化测试面临严峻挑战[1]。传统的基于DOM树和XPath定位的测试方法，在面对频繁出现的界面变更情况时，其脆弱性表现得十分明显，它的维护成本比较高，而且跨平台的复用率比较低[2-3]，现有的图像识别框架，如Airtest，虽然在一定程度上缓解了元素定位的问题，但是它依赖固定的模板匹配，在处理动态内容的时候准确率仅仅能达到60%～70%，很难契合企业的实际需求[4]。相关研究表明，智能化技术的引入给UI测试提供了新的突破，深度学习在图像分割与语义理解等方面拥有巨大的潜力，RPA与AI协同合作能够为多目标测试提供可行的解决方案[5-7]。针对现有方法在效率—精度权衡及复杂场景适应性方面的技术局限，提出基于改进YOLOv8的视觉驱动测试技术，融合卷积神经网络的深度特征提取与SIFT特征匹配的几何约束，提升多尺度UI元素的识别能力与定位稳定性，实现跨Android、iOS平台的统一执行。

1 技术方法与架构设计

1.1 基于改进YOLOv8的UI元素检测

移动应用界面控件分布密集、尺寸多样、背景复杂，对目标检测算法提出更高要求。卷积神经网络在UI元素分类中展现出优异的特征提取能力[8]，基于此选取YOLOv8作为基础架构。针对小尺寸控件特征响应弱、密集分布遮挡严重、长宽比集中等技术挑战，进行三项关键改进，在特征金字塔中增加高分辨率特征层保留细节信息用于小图标、按钮识别，采用K-means聚类优化锚点配置适应不同控件尺寸，应用深度可分离卷积替代标准卷积层以减小模型参数量与计算复杂度。训练采用迁移学习与增量训练相结合的策略，先使用COCO预训练权重初始化，再在UI控件数据集上增量训练，利用随机裁剪、颜色抖动、MixUp等数据增强技术提升模型泛化能力。

1.2 分层测试执行技术

元素识别仅构成自动化测试技术链路的起点，技术难点在于将识别结果转化为有效的测试操作序列并管理复杂的多状态流程转换。传统图像测试方法受限于僵化的脚本架构，难以适应动态应用程序接口[9]。采用分层技术设计，将测试执行逻辑解耦为感知层（UI元素检测与OCR文本提取）、决策层（有限状态机管理与SSIM相似度计算）、执行层（平台API封装与异常恢复机制）三个模块。跨越这些层的模块借助接口来实现通信，感知层负责捕捉接口信息，这一层除了调用UI元素检测模型来识别交互控件之外，还运用OCR技术提取文本数据，并且基于颜色直方图计算页面主题位移，借助多个特征呈现多维页面信息。决策层利用有限状态机模型管理测试流，状态转换依靠页面相似度计算，采用结构相似度指数（SSIM）进行判定，当SSIM值低于0.85时，判定为页面发生显著变化，触发状态转换逻辑。执行层接收决策层下发的测试指令并转换为平台原生控制命令。针对5G/4G网络切换

作者简介：杨欢（1991-），男，硕士研究生，工程师，研究方向：电子信息、软件质量、AI算法。

场景下页面加载时延的不确定性，设计自适应等待机制与异常恢复技术，等待机制通过动态监测页面加载状态调整超时阈值，异常恢复机制覆盖网络超时重连与应用崩溃恢复。参考AI数据采集技术[10]，执行层集成操作日志、屏幕截图、性能指标等多维数据采集能力，为缺陷回溯分析提供支撑。分层测试执行技术架构设计如表1所示。

表1详细描述了各层的功能职责、关键技术以及模块间的接口定义。这种分层架构不仅提高了可维护性，还为后续的功能扩展预留了接口，能够灵活适配不同类型的测试需求。

1.3 跨平台测试技术

Android与iOS平台在渲染引擎与交互协议层面存在显著技术差异，构成跨平台测试的核心障碍[11]。采用三层抽象技术架构，视觉抽象层基于与平台无关的特征描述符实现UI元素的统一表征，交互抽象层通过Adapter模式将统一操作原语映射至各平台原生API，验证抽象层提供多样化的断言机制。分辨率自适应技术采用基于锚点的相对定位方式将绝对坐标归一化为比例坐标，对动态区域采用SIFT特征匹配实现尺度不变定位，确保测试脚本的跨平台移植能力。

2 技术实现与应用验证

2.1 测试平台构建与实现

测试平台基于Python 3.8、PyTorch 1.12、OpenCV 4.5构建分布式技术架构，硬件配置采用Intel至强Gold 6248R处理器与NVIDIA RTX 3060显卡，测试设备集群包含10台Android终端，覆盖Android 8.0至13.0版本跨度。技术架构由识别模块（YOLOv8推理引擎）、执行模块(测试指令转换与平台API调用)、报表模块(测试结果可视化）三部分组成，模块间通过RabbitMQ消息队列实现异步解耦通信。集成Jenkins Pipeline与GitLab的持续集成流水线，基于Webhook机制实现代码提交触发的自动化回归测试。

2.2 实验设计与性能测试

实验数据集从电商、社交、金融、工具、游戏五类主流移动应用中采集15 000张UI界面截图，涵盖不同网络环境（5G/4G/WiFi）与分辨率的多样化场景，标注包括按钮、输入框、图标等15类常见UI控件。为全面评估改进YOLOv8模型的性能优势，设计对比实验选取Selenium（基于DOM解析）、Appium（基于原生控件识别）、Airtest（基于模板匹配）三种主流测试方法作为基准。在测试环境方面，统一配置成Ubuntu 20.04操作系统，并且让所有的算法都在相同的硬件条件之下运行。为减少随机误差，每项实验重复进行5次并取平均值。实验从元素识别准确率、执行效率、跨平台适应性、维护成本等关键维度进行综合评估，具体测试结果如表2所示。

数据如表2所示，人工智能模型实现82.7%识别率和68%跨平台应用，只需图片即可定位出控件，将原本几天完成脚本开发时间压缩到几小时以内。

2.3 企业应用实践与效果评估

技术方案在电商平台与金融应用的实际部署中，电商回归测试的周期显著缩短，金融兼容性测试通过锚点相对定位与SIFT特征匹配有效降低了人工成本。部署采用分布式架构，测试节点通过RabbitMQ消息队列调度并利用GPU加速推理。应用过程中遇到的主要挑战包括第三方广告等动态内容的识别稳定性问题，通过设置排除检测策略予以解决；5G与4G网络切换导致的时延波动则依靠自适应超时机制动态调整阈值来应对。如表3所示，该技术在11款产品的部署中实现年度成

表1 分层测试执行技术架构设计

架构层次	功能职责	关键技术	接口定义	数据流向
感知层	1. 界面状态实时获取 2.UI元素识别与定位 3. 文本信息提取 4. 界面主题分析	YOLOv8模型（80 ms/帧）、OCR技术、颜色直方图分析、多维特征提取	getUIElements()、extractText()、analyzeTheme()、getPageFeatures()	截图数据→特征向量
决策层	1. 测试流程管理 2. 页面状态判断 3. 测试路径规划 4. 异常检测与处理	FSM状态机、SSIM相似度计算（阈值0.85）、路径规划算法、智能等待机制	calculateSimilarity()、planTestPath()、handleException()、stateTransition()	特征向量→操作指令
执行层	1. 操作命令执行 2. 平台适配 3. 异常恢复 4. 数据采集记录	平台API封装、重试机制、日志系统、性能监控	click(x,y)、swipe(x1,y1,x2,y2)、input(text)、collectData()	操作指令→执行结果

表 2 四种测试框架综合性能对比

评估指标	Selenium	Appium	Airtest	本文方法
元素识别准确率（%）	依赖 DOM 结构	87.2（原生控件） 62.3（自定义控件）	68.5	82.7
平均执行时间（ms）	85	126	145	92
跨平台复用率（%）	30	42	55	68
动态内容处理（%）	不支持	45.2	52.6	71.0
维护成本指数	100	85	72	62
模型 / 框架大小	15.3 MB	28.6 MB	42.3 MB	15.2 MB
初始配置时间（小时）	2	4	3	8
支持的平台数量	1（Web）	2（Android/iOS）	3（多平台）	3（多平台）

（注：维护成本指数以 Selenium 为基准 100，数值越低表示维护成本越低。）

表 3 企业实际应用效果统计

应用场景	测试类型	测试规模	原耗时	现耗时	效率提升	缺陷检出率变化	年节省成本
电商 APP	功能回归测试	800 用例 / 天	16 小时	4.5 小时	71.9%	+18%	48 万元
金融 APP	兼容性测试	35 机型	6 人天	2.8 人天	53.3%	+15%	28 万元
社交 APP	UI 界面测试	500 用例	12 小时	5.2 小时	56.7%	+12%	待统计
合计（11 款产品）	综合	月均 3 万用例	—	—	52%	+16.5%	76 万元

本节约 76 万元、测试效率提升 52%、投资回报率 1.9 倍。实践表明，视频流 UI 的识别准确率为 68%（静态界面达 82.7%），主要受压缩算法与运动模糊影响，存在进一步优化空间；首次部署需 15 个工作日完成模型调优。

3 结束语

针对移动应用 UI 测试在跨平台适配与维护成本方面的技术挑战，提出基于改进 YOLOv8 的视觉驱动测试技术。通过引入高分辨率特征层、K-means 聚类优化锚点配置、深度可分离卷积轻量化等技术，结合有限状态机与 SSIM 相似度计算的智能决策机制，实现元素识别准确率 82.7%、平均响应时延 92 ms、跨平台复用率 68%。在电商、金融、社交等 11 款产品的部署中，测试效率提升 52%，年度成本节约 76 万元（ROI 达 1.9 倍）。实践中发现视频流 UI 的识别准确率为 68%（静态界面达 82.7%），主要受压缩算法与运动模糊影响，首次部署需 15 个工作日完成调优。未来将探索 Transformer 架构增强时序特征建模能力，引入预训练视觉模型提升复杂场景语义理解精度。

参考文献：

[1] 刘维维 . 人工智能技术在移动终端自动化测试中的应用 [J]. 软件导刊 ,2021,20(02):59-62.
[2] 余锦润 , 杨丹君 , 李波波 . 基于位图识别的 UI 自动化测试研究和应用 [J]. 自动化与仪表 ,2021,36(03):90-94.
[3] 袁之国 . 基于 UI 建模的移动应用自动化测试工具的设计与实现 [D]. 北京 : 北京邮电大学 ,2024.
[4] 胡莉琼 , 詹夏城 , 唐健玲 , 等 . 基于 Airtest 框架的图像识别自动化测试研究与应用 [J]. 工业控制计算机 ,2024,37(07):109-110.
[5] 丁磊 . 智能技术在软件设计自动化中的应用 [J]. 电子技术 ,2024(09):338-339.
[6] 黄非 . 图像智能识别及其在自动化测试的应用 [J]. 现代信息科技 ,2025,09(06):67-70.
[7] 李晖 , 辛华 . 基于 RPA+AI 的多目标自动化软件测试系统设计 [J]. 粘接 ,2025,52(07):144-146.
[8] 中国工商银行股份有限公司 . 基于卷积神经网络的用户界面测试结果分类方法及装置 :CN202110484586.3[P].2021-07-06.
[9] 厦门星纵物联科技有限公司 . 一种图像自动化测试的方法 , 装置 , 系统及存储介质 :CN202211225800.4[P].2023-01-31.
[10] 北京墨云科技有限公司 . 一种基于 AI 的自动化渗透测试系统的数据收集方法 :CN202010303510.1[P].2023-08-11.
[11] 彭飞 , 王建 , 寇超 , 等 . 基于图像的航天业务软件 Web UI 自动化测试方法 [J]. 测控技术 ,2024,43(09):21-27.