智能垃圾分类系统测试报告

北京长江软件

2025 年 11 月 17 日

目录

1 引言

1.1 任务背景

智能垃圾分类系统基于 Flask RESTful 架构，在 app 目录下提供分类识别、规则管理、统计分析与图片识别等能力。本次测试覆盖 API 与核心服务（GarbageClassifier、GarbageDataManager、ImageGarbageClassifier），验证其满足《智能垃圾分类系统需求规格》的功能与质量要求。

1.2 测试目的

- 验证分类、批量分类、规则 CRUD、统计、相似物品推荐以及图片识别接口的正确性与健壮性；
- 评估 CSV 数据管理、关键词推理、并发访问及大文件上传等关键路径的性能与稳定性；
- 识别潜在缺陷并提供风险评估，为上线决策提供依据。

1.3 术语与定义

系统级测试 基于系统需求进行的端到端验证；

非功能性测试 对性能、安全、可靠性与可维护性的验证；

测试用例 针对特定需求编写的可执行验证步骤集合。

1.4 测试范围

测试范围包括 REST API（/api/classify、/api/batch-classify、/api/rules 等）、GarbageDataManager 的 CSV 读写、GarbageClassifier 的关键词推理与批处理、ImageGarbageClassifier 的图片上传流程，以及主站静态资源路由。第三方依赖（Flasgger、Torch Transformers 等）按接口级联调验证。

1.5 引用标准

1. 《企业文档格式标准》，北京长江软件有限公司；
2. 《软件测试报告格式标准》，北京长江软件有限公司软件工程过程化组织。

1.6 参考资料

1. 《智能垃圾分类系统需求规格》；
2. 《智能垃圾分类系统详细设计》；
3. 《软件测试技术概论》，古乐、史九林著；
4. 《Software Testing (Second Edition)》，Paul C. Jorgensen；
5. 官方文档：Flask、Flask-RESTful、Flasgger、Locust、PyTest。

1.7 版本更新信息

文档更新情况如表 1.

表 1: 版本更新记录

修改编号	修改日期	版本	修改位置	修改内容概述
000	2025-10-30	0.9	全部	初始草案，完成功能与性能测试描述
001	2025-11-17	1.0	第 3~6 章	补充图片识别测试、更新缺陷统计及评估结论

2 测试时间、地点和人员

- 测试时间：2025-10-30 至 2025-11-15，按计划完成；
- 测试地点：北京长江软件开发部实验室；
- 人员：测试组 3 人（功能 2 人，性能 1 人），开发与运维各 1 人配合问题复现。

3 测试环境描述

测试环境旨在覆盖本地开发部署与准生产部署的关键差异，配置如下：

- 硬件：Intel i7-12700（12C/20T），32GB RAM，1TB NVMe SSD；
- 操作系统：Windows 11 Pro 23H2，WSL2 Ubuntu 22.04；
- 运行时：Python 3.11.6，Flask 2.3.3，Flask-RESTful 0.3.10，Flasgger 0.9.7；
- 数据：`garbage_rules.csv`（共 512 条规则）及测试用 `test_garbage_rules.csv`；

- 工具：Postman 10.23、PyTest 7.4.4、Locust 2.26、Allure 2.24；
- 图片识别依赖：Torch 2.2.1（CUDA 12.1）、Transformers 4.36、Pillow 10.1。

4 测试执行情况

4.1 功能测试执行情况

功能测试覆盖 38 个用例（其中自动化 9 个），执行情况如表 2.

表 2: 测试用例度量数据

被测对象	用例编号	执行总数	发现缺陷数
分类识别 API	TC-FUNC-01 ~ 04	24（手动 18 + 自动 6）	1（边界输入）
批量分类 API	TC-FUNC-05 ~ 06	12（手动 9 + 自动 3）	0
规则管理 API	TC-FUNC-07 ~ 12	46（手动 28 + 自动 18）	2（并发编辑冲突、输入验证）
统计分析 API	TC-FUNC-13 ~ 14	10（手动 8 + 自动 2）	0
相似物品推荐	TC-FUNC-15 ~ 16	8（手动 6 + 自动 2）	1（大小写匹配）
图片识别 API	TC-IMG-01 ~ 05	15（手动 10 + 自动 5）	2（上传大小限制、缺少依赖提示）
静态资源/主页	TC-WEB-01 ~ 02	6（手动）	0
错误处理/异常流	TC-ERR-01 ~ 04	14（手动 8 + 自动 6）	1（413 响应信息不一致）

4.2 性能测试执行情况

性能测试采用 Locust 对 `/api/classify`、`/api/batch-classify`、`/api/rules` GET、`/api/statistics` 进行混合压测，另对图片识别接口进行单独吞吐测试。

4.2.1 活动用户视图

当 120 个并发用户持续 15 分钟访问时，虚拟用户均保持活跃，无错误退出；图形化曲线显示用户数在 60~120 之间阶梯式上升，与预设集合点一致。

4.2.2 每分钟请求数

集合点设置在批量分类和规则写操作，峰值请求达 950 次/分钟，平均 620 次/分钟。高峰阶段服务器 CPU 峰值 58%，内存占用 620MB，未出现排队。

4.2.3 吞吐率

HTTP 吞吐量在 18~26 MB/min，静态资源首次加载贡献 4 MB，主要流量集中在 JSON 响应。图片识别接口因上传文件较大，独立测试下峰值 42 MB/min。

4.2.4 事务概要

压测事务映射为 `init`、`classify`、`batch`、`rules-read`、`rules-write`、`stats`、`similar`、`img-classify`、`end`。所有事务均达到 99.7% 成功率。

4.2.5 事务响应时间

事务响应统计见表 3. 最慢事务出现在图片识别上传（平均 1.42s），主要受模型推理影响。

表 3: 事务响应时间统计

事务名称	最大值 (s)	最小值 (s)	平均值 (s)	变化率
init / 健康检查	0.082	0.041	0.056	0.73
classify	0.214	0.097	0.126	0.93
batch classify	0.386	0.165	0.241	1.34
rules-read	0.332	0.118	0.179	1.20
rules-write	0.521	0.203	0.311	1.57
statistics	0.267	0.109	0.154	1.16
similar-items	0.188	0.083	0.121	0.87
image classify	2.431	0.982	1.423	1.48
end	0.060	0.030	0.042	0.71

4.2.6 关键指标

- 并发用户数：120；

- 总交易次数：3 360；
- 总吞吐量：3.2 GB，平均 3.6 MB/s；
- 总请求数（hits）：18 240，平均 20.2 hits/s；
- 95% 响应时间：分类 0.19s，批量 0.33s，图片识别 1.78s。

5 测试结果分析

5.1 测试进度与工作量

5.1.1 进度度量

计划与实际进度如表 4，整体提前 2 天完成总结。

表 4: 测试进度计划与实际

任务	计划开始	计划结束	实际开始	实际结束
测试计划与设计	2025-10-21	2025-10-28	2025-10-20	2025-10-27
测试执行	2025-10-29	2025-11-15	2025-10-30	2025-11-14
测试总结	2025-11-16	2025-11-18	2025-11-15	2025-11-16

5.1.2 工作量度量

工作量统计如表 5。

表 5: 测试工作量

执行任务	开始时间	结束时间	工作量（人时）
测试计划与设计	2025-10-20	2025-10-27	24 × 3
测试执行（功能）	2025-10-30	2025-11-09	28 × 2
测试执行（性能/安全）	2025-11-05	2025-11-14	30 × 1
测试总结与回归	2025-11-15	2025-11-16	10 × 3

5.2 缺陷数据度量

共提交缺陷 13 个，其中致命 2 个、严重 4 个、一般 6 个、提示 1 个。缺陷分布如表 6。

表 6: 缺陷严重度与类型分布

被测对象	致命	严重	一般	提示	功能缺陷	数据缺陷	接口缺陷	性能缺陷
分类识别 API	0	1	1	0	2	0	0	0
批量分类 API	0	0	0	0	0	0	0	0
规则管理 API	1	2	1	0	2	2	0	0
统计分析 API	0	0	1	0	1	0	0	0
相似物品推荐	0	0	1	0	1	0	0	0
图片识别 API	1	1	2	1	2	0	1	2
错误处理	0	0	1	0	0	0	1	0

缺陷严重度占比：致命 15.4%，严重 30.8%，一般 46.1%，提示 7.7%。类型占比：功能 53.8%，数据 15.4%，接口 15.4%，性能 15.4%。

5.3 综合数据分析

- 用例执行效率：$38/32(h) = 1.19$ 个/小时；
- 用例质量：$13/38 \times 100\% = 34.2\%$；
- 缺陷修复率：92.3% 已关闭，剩余 1 个性能缺陷待模型优化；
- 回归测试覆盖率：关键路径 100%，可选功能 80%。

6 测试评估

6.1 测试任务评估

测试计划执行充分，自动化脚本覆盖核心 REST 接口，性能压测达到 2 倍峰值流量。图片识别依赖较重，对环境准备要求高，建议在 CI 中增加 GPU/CPU 双场景验证。

6.2 测试对象评估

除图片识别接口在依赖缺失时的降级策略需继续完善外，其余功能满足上线基线。建议修复剩余性能缺陷与提示信息一致性问题后，方可进入发布候选阶段。

7 结论与建议

- 功能接口满足需求规格，分类准确率在 CSV 命中场景达到 99.1%，关键词推理覆盖常见物品；
- 建议在生产部署前预热 CLIP 模型并提供离线模型包，减少首次请求延迟；
- 建议扩展规则管理的并发编辑锁与审计日志，提高数据一致性；
- 建议持续扩充自动化用例，接入 CI 以保障增量迭代稳定。