

## **ИНТЕГРАЦИЯ КЛАССИФИКАТОРА МОБИЛЬНЫХ ПОЛЬЗОВАТЕЛЬСКИХ ИНТЕРФЕЙСОВ ДЛЯ ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ ИССЛЕДОВАТЕЛЬСКОГО ТЕСТИРОВАНИЯ**

*Семеняк Георгий Алексеевич*

*Студент*

*Факультет ВМК МГУ имени М. В. Ломоносова, Москва, Россия*

*E-mail: georgeseменяк@gmail.com*

*Научный руководитель — Турдаков Денис Юрьевич*

Автоматизированное исследовательское тестирование мобильных приложений направлено на обнаружение критических ошибок и воспроизведение пользовательского опыта. Одной из основных задач при таком тестировании является эффективное покрытие ключевых состояний приложения. Традиционные стратегии обхода интерфейса, например поиск в глубину, выполняют систематическое исследование, однако часто приводят к избыточным действиям на экранах с повторяющимися элементами или вспомогательными страницами.

В данной работе рассматривается подход к повышению эффективности исследовательского тестирования за счёт интеграции классификатора экранов пользовательского интерфейса. Классификатор анализирует изображение текущего экрана приложения и определяет его тип. Полученные метки используются модулем приоритизации действий, который выбирает подходящую стратегию взаимодействия с интерфейсом.

Классификатор экранов был предварительно обучен на расширенном и переразмеченном наборе данных, сформированном на основе открытых наборов мобильных интерфейсов RICO [2] и Enrico [3].

Предлагаемый метод предполагает использование различных стратегий в зависимости от типа экрана. Основные стратегии взаимодействия включают:

- для экранов с повторяющимися элементами интерфейса (например, списков или клавиатур) выполняется взаимодействие только с одним представителем группы элементов, что позволяет избежать избыточных действий;
- для экранов входа или регистрации используется автоматический ввод заранее подготовленных данных, что позволяет продолжить исследование без ручного вмешательства;

- всплывающие окна и обучающие экраны пропускаются, чтобы не расходовать тестовые действия на элементы, не содержащие функциональности приложения;
- для некоторых экранов выполняется выход со страницы, если дальнейшее взаимодействие может привести к бесполезному обходу интерфейса или взаимодействию с внешними сервисами;
- для экранов загрузки применяется стратегия ожидания, позволяющая дождаться полной отрисовки интерфейса;
- для экранов, содержащих основные элементы управления и настройки приложения, выполняется полный обход дерева пользовательского интерфейса.

Эффективность предложенного подхода оценивалась на наборе мобильных приложений из различных предметных областей. В эксперименте использовались приложения для бронирования гостиниц, обработки фотографий, электронной коммерции, управления финансами, чтения новостей и заказа еды.

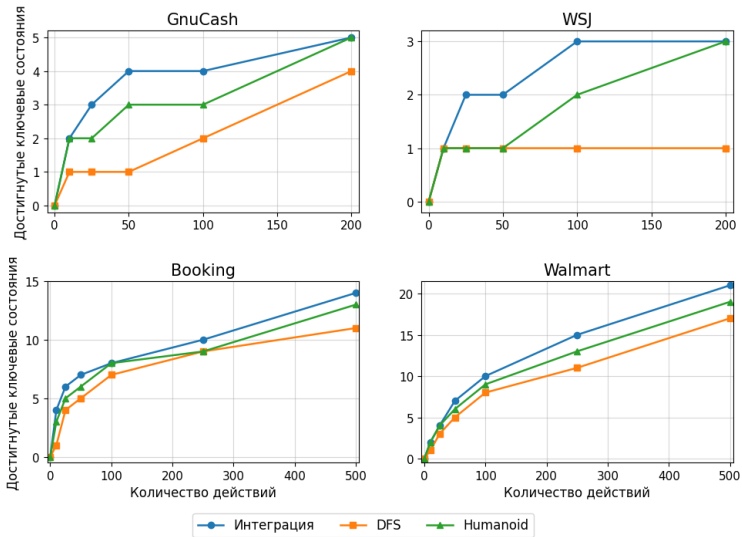
Сравнение проводилось с двумя базовыми стратегиями: традиционным поиском в глубину и подходом на основе машинного обучения, аналогичным модели Humanoid [1], обученной на больших наборах пользовательских взаимодействий.

Основной метрикой служило число покрытых ключевых состояний приложения — страниц, представляющих основные функциональные состояния приложения. Результаты эксперимента показывают, что предложенная стратегия интеграции классификатора превосходит классическую стратегию поиска в глубину и демонстрирует улучшенные результаты по сравнению с ML-подходом.

Преимущество предложенного метода особенно заметно в интерфейсах с большим количеством повторяющихся элементов, например в списках товаров интернет-магазинов или новостных лентах. За счёт исключения избыточных действий система быстрее достигает новых состояний интерфейса и эффективнее покрывает функциональность приложения.

Таким образом, предложенная интеграция классификатора пользовательских интерфейсов позволяет повысить скорость достижения ключевых состояний и увеличить общее покрытие функциональности приложения при исследовательском тестировании мобильных систем.

## Иллюстрации



Сравнение покрытия ключевых состояний приложения при использовании различных стратегий исследовательского тестирования.

## Литература

1. Li Y., Yang Z., Guo Y., Chen X. Humanoid: A Deep Learning-Based Approach to Automated Black-box Android App Testing // 2019 IEEE/ACM 34th International Conference on Automated Software Engineering (ASE). IEEE, 2019. P. 1070–1073.
2. Deka B., Huang Z., Franzen C., Hibschman J., Afergan D., Li Y., Nichols J., Kumar R. Rico: A Mobile App Dataset for Building Data-Driven Design Applications // Proceedings of the 30th ACM Symposium on User Interface Software and Technology (UIST). New York, USA, 2017. P. 845–854.
3. Leiva L. A., Hota A., Oulasvirta A. Enrico: A Dataset for Topic Modeling of Mobile UI Designs // Proceedings of the 22nd International Conference on Human-Computer Interaction with Mobile Devices and Services. Oldenburg, Germany, 2020.