**УДК 004.51, 004.89**

# ПОИСК ЭФФЕКТИВНОГО МЕТОДА ДЕТЕКЦИИ ЭЛЕМЕНТА GUI ПО ЕГО ПРИЗНАКАМ

**Костенко К.Д.** (ИТМО), **Аминов Н.С.** (ИТМО)
**Научный руководитель – кандидат технических наук, доцент Ватьян А.С.** (ИТМО)

**Введение.** Современные графические пользовательские интерфейсы (GUI) являются основным инструментом для взаимодействия пользователей с электронными устройствами и информационными системами. Например, в 2024 году более 4 миллиардов человек во всем мире используют смартфоны [1]. Современные системы требуют все более универсальных и удобных методов анализа графических интерфейсов. Для реализации интеллектуального помощника с функцией пошаговых графических подсказок «Teept» требовалось разработать систему для поиска элементов на экране, совместимую с мультимодальными LLM. Было принято решение использовать алгоритмы компьютерного зрения, как наименее требовательный к ресурсам при разработке метод.

**Основная часть.** Задачей была разработка системы, комбинирующей использование API ChatGPT и методы компьютерного зрения, способной для каждого шага инструкции обводить необходимый элемент интерфейса на экране поверх остальных приложений. Решение предполагает объединение возможностей LLM и детекции элементов по их визуальным и текстовым признакам.

Изначально была предпринята попытка использования проекта GroundingDINO [2] для детекции нужного элемента. Проект привязывает описания к изображениям, "заземляя" их и позволяя получить область, в которой находится объект на изображении по его описанию. Однако в ходе тестирования выявилась невозможность использования проекта без дополнительного обучения. Исходное обучение на датасете COCO не дало модели достаточно информации для понимания концепции графических элементов, их визуальных образов и текста на них. Отсутствие возможности поиска элемента по включенному в описание тексту оказалось серьезной проблемой. В дальнейшем предпринималась попытка применить к модели fine-tuning для улучшения результатов, однако имеющиеся датасеты не охватывали достаточного разнообразия графических элементов, чтобы обучить модель их различать. В связи с этим, в условиях ограниченного времени и бюджета, было принято решение воспользоваться другими проектами. Сначала была произведена попытка использования других разработок на основе GroundingDINO с интеграцией языковых моделей, как, например, это реализовано в проекте DetGPT [3]. Однако проблема поиска и идентификации текстовых элементов сохранялась.

Для повышения точности определения элементов, содержащих текст, и улучшения распознавания вообще было принято решение использовать OCR (Optical Character Recognition) для поиска элементов с текстом и классические методы компьютерного зрения для детекции остальных объектов. Такое решение уже было реализовано в проекте UIED [4] и требовало относительно небольших доработок, таких как подключение Tesseract OCR [5], настройки входных и выходных данных. В нем для детекции нетекстовых элементов используются градиентные карты и сегментация через бинаризацию.

В рамках работы Teept инструкции по поиску элемента поступают от LLM, на основе которой работает система. Передаётся информация о том, следует ли искать элемент по тексту через OCR или по его описанию с использованием методов компьютерного зрения, а также сам текст или описание элемента. Для проверки схожести изображения и описания используется нейросеть CLIP [6], в которой реализованы два энкодера — один для текста, другой для изображений — чтобы преобразовывать оба типа данных в общее векторное пространство. Для поиска по тексту используется библиотека fuzzywuzzy [7], которая применяет расстояние Левенштейна для нечеткого сравнения строк. Также для улучшения

результатов экран делится на 9 равных блоков, и вместе с инструкциями GPT возвращает информацию о том, в каком из них находится искомый элемент, что помогает убрать лишние элементы, сократить время обработки и повысить точность.

**Выводы.** В работе рассмотрен процесс разработки системы анализа GUI и подбор методов для ее реализации. Нынешнее решение, хоть и показывает существенно лучшие результаты по сравнению с предыдущими, все еще требует немалой доработки и далека от совершенства, однако была создана модульная, универсальная система, которая послужит основой для дальнейших разработок в сфере анализа графического интерфейса.

**Список использованных источников**:


1. Statista. Number of smartphone users worldwide from 2014 to 2029. – 2024.
2. Liu S. et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection //European Conference on Computer Vision. – Cham: Springer Nature Switzerland, 2024. – С. 38-55.
3. Pi R. et al. Detgpt: Detect what you need via reasoning //arXiv preprint arXiv:2305.14167. – 2023.
4. Xie M. et al. UIED: a hybrid tool for GUI element detection //Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. – 2020. – С. 1655-1659.
5. Smith R. An overview of the Tesseract OCR engine //Ninth international conference on document analysis and recognition (ICDAR 2007). – IEEE, 2007. – Т. 2. – С. 629-633.
6. Radford A. et al. Learning transferable visual models from natural language supervision //International conference on machine learning. – PMLR, 2021. – С. 8748-8763.
7. Levenshtein V. I. Binary codes capable of correcting deletions, insertions, and reversals //Proceedings of the Soviet physics doklady. – 1966.