and finds the closest match [3]. The third level engages computer vision for visual search on the screenshot, effective when the DOM hierarchy has changed significantly but the visual representation remains similar. For training such models, large datasets containing tens of thousands of annotated mobile UI screenshots are used. CNNs learn to recognize typical components – buttons, text fields, toggles, lists – regardless of styling [1]. Transformer architectures additionally form a "semantic map" of the interface by accounting for contextual relationships between elements. Furthermore, reinforcement learning (RL) agents can autonomously explore the application's state space, detecting edge-case crashes, performance bottlenecks, and layout anomalies that static scripts might miss [1]. Integration with CI/CD is realized through time-bounded exploration sessions triggered after deterministic regression tests pass. Results are analyzed for trends, creating a feedback loop that directs engineering attention toward the most defect-prone modules.

As a conclusion, transitioning to ML-based automated testing significantly enhances the resilience and scalability of mobile software quality assurance. By embedding intelligent models into the CI/CD workflow, development teams can accelerate release cycles while maintaining high standards of reliability [3]. Empirical evidence suggests that self-healing mechanisms reduce test maintenance time by 60–80%, and intelligent exploratory testing detects 25–40% more defects compared to traditional methods. This approach minimizes the manual burden of script maintenance, with ML complementing rather than replacing traditional testing – deterministic tests provide baseline coverage while intelligent agents extend it to areas unreachable by static scripts. A promising further direction is integrating large language models (LLMs) for generating test scenarios in natural language and converting them into executable code, democratizing access to automated testing and closing the gap between business requirements and technical implementation.

References

1. E. Collins, A. Neto, A. Vincenzi, and J. Maldonado, "Deep reinforcement learning based Android application GUI testing," in Proc. XXXV Brazilian Symp. Software Engineering (SBES), Sep. 2021, pp. 186–194.
2. Apple Inc., "XCTest: User Interface Testing," Apple Developer Documentation. Available: https://developer.apple.com/documentation/xctest/user_interface_tests.
3. S. Saarathy, S. Bathrachalam, and B. Rajendran, "Self-healing test automation framework using AI and ML," Int. J. Strategic Management, vol. 3, no. 3, pp. 45–77, 2024.

AI-DRIVEN AUTOMATION FOR CAMERA-DEPENDENT TESTS AND IMAGE PROCESSING IN EMBEDDED SYSTEMS

Vorozhko M., Hlavchev D.
National Technical University “Kharkiv Polytechnic Institute”, Kharkiv, Ukraine

Modern embedded systems, ranging from surveillance IoT devices to automotive driver-assistance modules, increasingly rely on integrated cameras and complex image processing pipelines. Validating these systems involves rigorously testing video streaming features, image signal processing (ISP) algorithms, and camera driver stability under various environmental conditions. Traditionally, assessing visual outputs—such as verifying the absence of visual artifacts, frame drops, or incorrect color rendering—has required significant manual intervention or the use of fragile pixel-matching scripts. The integration of Artificial Intelligence (AI), specifically machine learning and computer vision techniques, provides a robust solution for automating these camera-dependent tests, enabling continuous and reliable evaluation of visual data without human oversight [1, 2].

The purpose of the report is to explore the application of AI models as automated test oracles for embedded image processing software.

By deploying convolutional neural networks (CNNs) and advanced computer vision algorithms within the testing framework, it is possible to automatically evaluate the quality and accuracy of video streams generated by embedded devices.

This AI-driven approach allows for the dynamic detection of anomalies in image processing modules, ensuring that any firmware updates affecting camera functionalities are thoroughly and automatically verified before deployment [3, 4].

A critical component of this methodology is the use of synthetic data generation and metamorphic testing for image-centric embedded software. Since acquiring physical test data for every possible lighting condition or camera angle is impractical, AI models can be utilized to apply controlled distortions, noise, or lighting alterations to a baseline set of test images. The embedded system's image processing algorithms are then executed using these modified inputs. Subsequently, a trained AI evaluator analyzes the processed outputs to determine if the system successfully handled the visual noise or if it produced unexpected artifacts. Furthermore, standard automated metrics, such as Structural Similarity (SSIM), can be integrated alongside deep learning evaluators to provide a comprehensive, quantitative assessment of image degradation and processing accuracy [5].

Implementing AI-driven visual testing [6] within a continuous integration (CI) pipeline fundamentally transforms how camera-dependent features are maintained. When a developer commits changes to video streaming protocols or image rendering code, the automated suite captures the output from the target hardware's camera interface. The AI model processes this output in real-time, flagging anomalies such as jitter, latency spikes, or compression errors that traditional unit tests might miss. This continuous, intelligent monitoring ensures high fidelity in video streaming applications and drastically reduces the time spent on manual visual QA.

As a conclusion, the utilization of AI for automating camera-dependent tests represents a significant advancement in embedded software quality assurance. By replacing subjective manual inspections with objective, AI-driven visual evaluations, development teams can confidently scale their testing efforts for complex image processing systems. This not only accelerates the release cycles for camera-equipped embedded devices but also ensures a consistently high standard of visual data processing and streaming reliability. Moreover, as embedded vision systems continue to grow in complexity and deployment scale, AI-driven test automation becomes not merely a convenience but a necessity for maintaining quality at pace with development. Future work may further explore the integration of lightweight, on-device inference models that perform self-diagnostics directly within the embedded hardware, eliminating the dependency on external evaluation infrastructure. Ultimately, the convergence of machine learning, computer vision, and continuous integration practices establishes a new paradigm for robust, scalable, and intelligent quality assurance in embedded systems engineering.

References

1. J. M. Zhang, M. Harman, L. Ma, and Y. Liu, "Machine Learning Testing: Survey, Landscapes and Horizons," *IEEE Trans. Softw. Eng.*, vol. 48, no. 1, pp. 1–36, Jan. 2022, doi: 10.1109/TSE.2019.2962027.

2. V. H. S. Durelli et al., "Machine Learning Applied to Software Testing: A Systematic Mapping Study," *IEEE Trans. Rel.*, vol. 68, no. 3, pp. 1189–1212, Sep. 2019, doi: 10.1109/TR.2019.2892517.

3. S. Segura, G. Fraser, A. B. Sanchez, and A. Ruiz-Cortés, "A Survey on Metamorphic Testing," *IEEE Trans. Softw. Eng.*, vol. 42, no. 9, pp. 805–824, Sep. 2016, doi: 10.1109/TSE.2016.2532875.

4. S. Ajorloo, A. Jamarani, M. Kashfi, M. H. Kashani, and A. Najafizadeh, "A systematic review of machine learning methods in software testing," *Appl. Soft Comput.*, vol. 162, p. 111805, 2024, doi: 10.1016/j.asoc.2024.111805.

5. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.

6. V. Garousi, N. Joy, Z. Jafarov, A. B. Keleş, S. Değirmenci, E. Özdemir, and R. Zarringhalami, "AI-powered software testing tools: A systematic review and empirical assessment of their features and limitations," *arXiv preprint* arXiv:2409.00411, 2024.

РОЛЬ АУДІО-ВІЗУАЛЬНОГО ПОЄДНАННЯ ДЛЯ НАВІГАЦІЇ РОБОТІВ У ДИНАМІЧНОМУ СЕРЕДОВИЩІ

Лозицький А., Главчев Д.
Національний технічний університет «Харківський політехнічний інститут»,
Харків, Україна

Рухомі об’єкти, такі як люди або інші роботи, створюють значні перешкоди для візуальної одометрії, оскільки вони перекривають статичний фон і призводять до збоїв у побудові карт [1]. Традиційні системи розпізнавання об’єктів на основі глибокого навчання вимагають великих обчислювальних ресурсів і потужних графічних процесорів (GPU), що обмежує їх використання на мобільних роботах [2]. Вирішенням стає застосування локалізації джерела звуку (SSL) для виявлення шумових перешкод [1]. Напрямок звуку проєктується на RGB-D зображення камери, формуючи обмежувальну зону навколо об’єкта, після чого значення глибини в цій зоні анулюються [1]. Це дозволяє системі витягувати візуальні ознаки