International Journal of Multidisciplinary Research and Growth Evaluation.

# Adaptive Testing For Real-Time Object Recognition in AR Applications

**Santosh Kumar Jawalkar**
Independent Researcher, Texas, USA

* Corresponding Author: **Santosh Kumar Jawalkar**

**Article Info**

**ISSN (online):** 2582-7138
**Volume:** 04
**Issue:** 05
**September-October 2023**
**Received:** 16-08-2023
**Accepted:** 12-09-2023
**Page No:** 1139-1144

**Abstract**
3D Object recognition is one of the most important factors behind Augmented Reality (AR) applications to provide immersive and interactive experiences. They optimize recognition accuracy and responsiveness despite varying lighting conditions, motion blur, occlusions, and time constraints. We propose a viewing synthesis and performance training framework for adaptive AR object detection. They would generate the tests, and as per simulated and real-world contexts, they would check the smooth recognition in action-oriented dynamic environments. Performance optimization may encompass model compression, edge computing, adaptive learning, hardware acceleration and energy-aware computing to maximize real-time processing with retained accuracy. The study highlights how AR applications can leverage AI-enabled adaptive learning and self-optimizing. The advancements in federated learning, multi-sensor fusion and 5G illuminating AR frameworks will usher in seamless navigation experiences across various fields (education, retail, geolocation) and enable real-time object recognition capability.

**DOI:** https://doi.org/10.54660/.IJMRGE.2023.4.5.1139-1144

**Keywords:** Augmented Reality (AR), real-time object recognition, adaptive testing, automated test case generation, model optimization, edge computing, machine learning, computer vision, hardware acceleration, performance benchmarking

## 1. Introduction

Augmented Reality (AR) is an emerging technology that finds a wide range of applications in gaming, navigation, industrial robotics, and healthcare [1]. Due to its utility to enhance user experiences and interactive solutions across different use cases, AR applications can superimpose virtual objects over the physical environment. Another important aspect of AR technology is real time object recognition, which enables systems to identify and track objects in their surroundings [2]. This is crucial for smooth interactions with AR applications that require recognizing and detecting objects. But ensuring that kind of reliability in the fast and unpredictable environments is anything but simple. Significant differences in light, motion blur, occlusions, and detailed backgrounds can drastically change recognition performance and will lead to inconsistent AR experiences [3]. Static assessment techniques commonly applied in traditional testing paradigms for object recognition systems fall short in addressing the dynamic characteristics of AR environments. No standardized datasets or controlled test conditions account for the complicated and not isolated variety between different cases where lighting is dynamic, objects are moving too fast, and backgrounds are cluttered. Furthermore, AR applications must operate in real time, which requires high processing speed and low-latency detection. These challenges necessitate this adaptive testing framework, used to assess recognition systems for AR objects under varying and variable conditions [4].

Automated Test Case Generation Another critical aspect of adaptive testing is the automated generation of test cases. Manually creating test scenarios that capture each combination of a large scale of conditions such as lighting conditions, motion speeds and object orientations would take too much time and are infeasible for large scale systems in the use cases involving the generation of synthetic data for testing and validation [5]. Automated test case generation via AI/ML techniques can facilitate synthetic data creation and the emulation of real-world environments in a controlled setting [6]. These procedures enable validation of object detection algorithms on a complete history of occurrences and ensure AR applications stay reliable and performant using various user behaviors [7]. A different key approach is the improvement of algorithms involving AR object detection [8]. Computational Efficiency: AR applications are mostly run on resource-constrained devices, specifically smartphones, AR headsets, and wearables, so computational efficiency is very critical [9]. High-performance metrics with accuracy with models of object recognition Abundant approaches such as model compression [6], edge computing [12], and fine-tuning algorithms are deployed for improving efficiency while preserving recognition precision. These procedures are essential to AR applications, allowing it to truly offer its intricate and immersive experiences without worrying too much about delays and hardware constraints [11, 12].

The adaptive testing approaches aim at fast recognition of real-time object at different illumination and moving environment and automizing to create the test case and optimizing effective performance improvement in object detection [13]. Data-driven testing frameworks that work with real-world variables are very important, and this work is an excellent contribution to the challenges of managing AR object recognition robustness and testing [14]. Moreover, their work helps to improve the performance of AR-based applications across a range of industries thereby enabling the robustness and efficiency of object detection to operate amidst different operating environments.

## 2. Challenges in AR object recognition

Real-time object detection and recognition technologies serve as the foundational features of interactivity and immersion in Augmented Reality (AR) applications [1, 16]. But they are difficult because of how variable the conditions in the real world are. Unlike simulation environments that might be used for object detection training [2, 15], unsimulated settings are characterized by variable and dynamic lighting, motion changes occlusions and background complexity. Such variables are vital in recognition which could cause hazards by providing false or late recognitions [17]. Considering that AR systems rely on real-time object recognition, two key aspects must be addressed to ensure reliability: the challenges of obtaining real-time information and the need for adaptive testing methodologies that can dynamically adapt to the application context [3, 18].

### A. Lighting Variability

Most detection models depend on visual features (in most cases color, shape and texture) thus lighting conditions can be decisive for object recognition in AR. But real-world lighting conditions vary greatly: bright daylight, dim indoor conditions, artificial illumination [19]. For example, moving from day to indoor environments or encountering a shadow suddenly impact object recognition algorithms [20]. Moreover, reflections and sunlight can deform objects properties which also lead to the challenge of detection. Establishing a pioneer path that traditional computer vision models have struggled with, usually with extensive retraining on diverse, domain-specific lighting datasets [3, 12]. Techniques like adaptive illumination models and deep learning-based brightness correction are also utilized to address these limitations, but extensive testing is needed to validate their effectiveness under various lighting conditions [4, 8, 11].

### B. Motion & perspective variations

Since both users and their environments are dynamic in Augmented Reality (AR) applications, real-time object recognition poses a challenge. Like motion blur, when objects move fast, it loses detail and becomes hard to detect [12]. Moreover, captured objects may look different based on their viewing angle or distance from the camera, which leads to the necessity for the recognition models to generalize over different object perspectives. Most AR systems presume the availability of features through feature extraction, which inevitably segues into the failure of detection systems at extreme motion ranges, resulting in incorrect or missed detections [14]. AR object recognition algorithms need to adapt to this challenge by implementing motion-adaptive models that help mitigate blur and perspective shift. We use optical flow estimation and CNNs trained on multi-angle datasets to achieve accurate recognition in dynamic environments [15]. Nonetheless, testing these models across different motion scenarios is an essential step in the testing process [16].

### C. Background complexity & occlusions

Many real-world applications present challenges where backgrounds are complicated and in motion, where the object detection model needs to separate the target object from the surroundings. Background clutter with similar textures or color can result in false positives, where the system identifies an incorrect object [4, 17, 19]. Also, occlusions happen when some objects are partially obscured by other elements, leading to a shortage of visible information for detecting the object. This is especially problematic for AR navigation and gaming applications, where users interact with multiple objects in real time [20]. But before you can get to that sweet augmentation part, you do have to exercise a little bit of caution: A solid AR object recognition system will rely on background segmentation and wear occlusion to help improve detection accuracy [2, 12, 15]. More advanced depth-sensing technologies and semantic segmentation models separate items from their context, which reduces word sensitivity and helps the technology to recognize what you mean in challenging contexts [5, 8].

### D. Adaptability to environmental changes

AR systems coordinate with environmental changes such as lighting differences or weather situation changes and object repositioning [9]. For instance, traditional object recognition applications would assume that the input images come from a controlled environment, even one where the objects it learns are all about like one APPEARANCE: However, AR applications need to keep updating their understanding of the environment [10]. This adaptation is important in use cases like AR-based navigation where users could walk through dynamic areas, and AR augmentations like industrial products where geometry could change very frequently [5, 12]. These methods of testing must simulate changing environments in real-time to observe how well the object recognition models adapt [13]. For example, continual learning models and on-the-fly dataset augmentation are techniques that enhance adaptability by allowing AR applications to stay accurate and responsive in a wide range of real-world scenarios [16].

### E. Security & privacy concerns in AR recognition

With continued widespread usage of AR applications, security and privacy ramifications surrounding object recognition are becoming more important. AR systems typically rely on the processing of visual data, meaning where data is captured from user surroundings, which can cause privacy concerns and illicit data collection [17]. Furthermore, adversarial attacks on object recognition models can produce results to influence and cause the detection result to change, which can present security risks. Adversarial examples can be used to manipulate AR systems by making modifications to objects that lead to misrecognition [9, 18]. In addition, to address the security issues mentioned above, security mechanisms such as encrypted AR object data processing, adversarial training, and secure AR object recognition framework validation should be integrated into AR object recognition frameworks [19]. Security tests should be part of those protocols that need to be applied to the software to check how much the software would withstand against an attack and whether the data of its user is safe or not [20].

### F. The need for adaptive testing frameworks

With the host of challenges impacting AR object recognition, standard testing scenarios fail to capture system performance in any meaningful way for the real world. Automated test case generation, real-time scenario simulations, performance benchmarks under various conditions are built into these frameworks [5, 7]. Developers can use AI-driven testing methodologies to improve the reliability of AR applications and refine recognition algorithms for real-world scenarios [8]. Chapters ahead will cover methods to realize adaptive testing and performance optimization strategies leading to improved object recognition of AR [10].

### G. Conceptual constraints & real-time performance

AR applications need to execute the object detection tasks on time, in real-time, to deliver seamless experiences for users. Real-time processing, however, can be computationally demanding, especially on mobile devices and AR headsets [12]. Processing high-resolution images as well as the detection models can be expensive which results in a delay in recognition [14]. Techniques like model pruning, quantization, and even edge computing are used to optimize these models so as to retain performance while ensuring accuracy [15]. Why Model Compression is Important and How to Approach It Model compression aims to decrease the footprint of your recognition models while retaining their predictive power so that they can be executed on devices with limited resources [16]. Also, offloading computations to edge servers or the cloud balance performance needs. Real time performance can be platform dependent due to different runtime hardware constraints, testing models under varied settings is crucial for such applications [18, 20].

## 3. Testing methodologies for robust object recognition

**Table 1:** Comparison of testing methodologies for AR object recognition

| Testing Methodology | Description | Advantages | Limitations |
|---|---|---|---|
| Manual Testing | Human testers evaluate object recognition accuracy in different environments. | Real-world validation, user experience assessment. | Time-consuming, limited scalability. |
| Automated Testing | Software-driven test case execution using predefined scenarios. | High efficiency, repeatability, scalability. | May not fully replicate real-world complexity. |
| Synthetic Data Testing | AI-generated datasets simulate real-world conditions for testing. | Cost-effective, diverse scenarios. | May lack real-world unpredictability. |
| Simulation-Based Testing | Virtual environments replicate real-world conditions for controlled evaluation. | Safe, repeatable, adaptable. | Computationally intensive. |
| Field Testing | Real-world testing in live environments under varying conditions. | Realistic, comprehensive validation. | Expensive, time-consuming. |

**Table 2:** Automated test case generation techniques for AR object recognition

| Technique | Description | Use Cases | Challenges |
|---|---|---|---|
| Synthetic Data Augmentation | AI-generated variations in lighting, angles, and occlusions. | Improving model robustness, expanding dataset diversity. | Requires high-quality data generation models. |
| Scenario-Based Testing | Predefined real-world scenarios (e.g., different lighting conditions, motion variations). | Testing AR applications in specific user environments. | Needs comprehensive scenario coverage. |
| Real-Time Simulation | AR engine-based testing using dynamic virtual environments. | Evaluating real-time recognition performance under motion and occlusions. | High computational cost, hardware dependency. |
| Randomized Test Case Generation | Automated generation of varied test conditions. | Uncovering edge cases and improving model generalization. | May require extensive validation to ensure relevance. |

**Table 3:** Performance metrics for evaluating AR object recognition models

| Metric | Definition | Importance | Limitations |
|---|---|---|---|
| Accuracy | Percentage of correctly recognized objects. | Measures overall recognition reliability. | May not account for false positives and negatives. |
| Precision | Ratio of true positive detections to total detected objects. | Evaluates correctness of positive identifications. | Can be misleading in imbalanced datasets. |
| Recall (Sensitivity) | Ratio of true positive detections to total actual objects present. | Indicates ability to detect all relevant objects. | May not reflect model's tendency to produce false positives. |
| F1-Score | Harmonic mean of precision and recall. | Balances accuracy and completeness. | Does not capture computational efficiency. |

| | | | |
|---|---|---|---|
| Processing Time | Time taken for the model to recognize an object. | Essential for real-time applications. | May vary depending on hardware specifications. |
| Memory Usage | Amount of system memory required for model inference. | Important for AR devices with limited resources. | Higher memory usage can affect AR performance. |

**Table 4:** Optimization strategies for real-time object recognition in AR

| Optimization Strategy | Description | Benefits | Challenges |
|---|---|---|---|
| Model Compression | Reducing model size while maintaining accuracy. | Enhances efficiency for real-time applications. | Risk of losing recognition accuracy. |
| Edge Computing Integration | Processing object recognition on local devices instead of cloud servers. | Reduces latency, improves response time. | Requires powerful edge devices. |
| Pruning & Quantization | Removing redundant model parameters and using lower-precision computations. | Improves processing speed and memory efficiency. | Can lead to minor accuracy losses. |
| Algorithm Fine-Tuning | Optimizing hyperparameters based on test results. | Adapts model to real-world scenarios. | Computationally expensive. |
| Adaptive Learning Models | Continual learning using real-time data updates. | Improves model adaptability over time. | Requires robust training infrastructure. |

**Table 5:** Testing frameworks and tools for ar object recognition

| Framework/Tool | Functionality | Advantages | Limitations |
|---|---|---|---|
| OpenCV | Open-source library for computer vision and object recognition. | Widely used, extensive community support. | Requires manual tuning for AR applications. |
| TensorFlow Lite | Optimized deep learning framework for mobile and AR applications. | Supports efficient real-time inference. | Limited support for complex AR scenarios. |
| Unity ARKit/ARCore | Provides AR simulation environments for testing. | Realistic simulation of AR interactions. | High hardware requirements. |
| YOLO (You Only Look Once) | Fast object detection algorithm used for real-time recognition. | High-speed recognition, suitable for AR. | Can struggle with small object detection. |
| ML Kit by Google | Machine learning toolkit for on-device AR applications. | Easy integration with mobile AR apps. | Less flexible for custom deep learning models. |

## 4. Performance optomization strategies for AR object recognition

Real-time object recognition is a critical standard for AR based applications to provide users a seamless experience in gaming, navigation and industrial automation. Yet, maintaining high recognition accuracy, while efficiently computing the similarity measures is a challenging task. On smartphones and head-mounted displays, AR devices generally possess limited processing power and battery life, thus optimizing object detection algorithms is crucial. These techniques often target at computational complexity reduction, processing speed enhancement, and memory consumption improvement, keeping recognition accuracy intact or enhancing recognition rate. Methods for Optimizing Performance: Depending on the target application, real-time performance is paramount, especially if objects need to be detected while the camera is live, detecting movement or large objects in dynamic environments.

### A. Model compression and pruning

Such deep learning models applied in an AR object recognition context are typically heavy weight, requiring substantial computing resource and memory. Various model compression techniques (i.e., pruning and quantization) scale down these models without sacrificing recognition accuracy. Pruning is the process of removing redundant parameters of neural networks, thus reducing model complexity and inference time. This method does follow-up of unnecessary computations. On the other hand, quantization downscales the numerical precisions, enabling the model operations to run efficiently even on a limited processing unit hardware. Quantization replaces floating-point operations (32-bit) with int8 (8-bit) operations, which speeds it up dramatically, all while keeping a decent amount of accuracy. This optimization methods can greatly benefit mobile AR applications as they demand speed and lightweight models for real-time recognition.

### B. Edge computing for low-latency processing

Traditional augmented reality applications will resort to cloud computing for object recognition, thus leading to a delay in recognition due to network transfer. The advent of edge computing helps mitigate this problem since computation tasks are transferred from the centralized server to local devices, thereby shortening response instances and offering better real-time performance. AR applications can handle object recognition with low latency by processing data on the device, which makes the interaction more natural. Current AR devices are equipped with cutting-edge hardware required to boost deep learning inference-per-edge, e.g., NPUs/neural processing units, and GPUs/graphics processing units. Using this approach helps to reduce the reliance on the connectivity to the network and provides a great user experience even in contexts where no online connectivity is available, or the amount of available bandwidth is low. But, in order to do edge computing, efficient resource management is required to trade-off power consumption versus computational performance.

### C. Adaptive learning models for real-time adaptability

Augmented Reality (AR) object recognition systems need to handle scenarios where objects may be present under varying lighting, orientations, and viewpoints, to account for dynamic surroundings. Adaptive learning models constantly update for real-time data to boost recognition precision. Adaptive systems, on the other hand — literally learn where the mistakes are from previously labelled datasets and are capable of ongoing improvement on object detection.

Purpose and Method: Transfer learning techniques help in fine-tuning pre-trained models for better in-domain model by training with domain-specific data, hence helping recognition in real world scenarios. Moreover, Self-supervised learning contributes to improving model robustness with large unlabeled datasets, while mitigating the dependency on labelled data. It helps ensure accurate and responsive AR applications in our fast-changing environments.

### D. Algorithm fine-tuning for performance optimization

[Intelligently optimizing the object detection algorithms to balance speed versus precision.] It is explaining how Hyperparameter optimization can elevate the performance of the model. Learning rate, batch size, and network architecture are some of the parameters that significantly influence the speed and accuracy of object recognition. Fine tuning consists of optimizing for these parameters based on broad testing through various AR environments. Background: Lightweight deep learning models such as Mobile Net, YOLO (You Only Look Once), and Efficient Net provide computational efficient frameworks for image detection without compromising the accuracy of detection. Before deploying an AR experience, real-time benchmarking during the development process can assist developers in pinpointing bottlenecks and optimizing the inference speed of neural networks, ensuring seamless interactions across various devices.

### E. Hardware acceleration for efficient AR processing

AR object recognition requires hardware acceleration to be used in real-time. Today's AR devices include hardware components dedicated to performing AI-driven functions. For deep learning inference and training, GPUs (Graphics Processing Units) and TPUs (Tensor Processing Units) parallelize these compute-intensive operations, which speeds up object detection. Moreover, Field Programmable Gate Arrays (FPGAs) or Application-Specific Integrated Circuits (ASICs) can be utilized to provide specialized hardware for AR processing. By offloading intensive processes to dedicated hardware solutions, it drastically lowers the latency involved, thus making AR applications more fluid and responsive. This will require developers to optimize their model better and also induce compatibility across AR platforms.

### F. Energy efficiency and battery optimization

The performance of mobile AR applications needs to be balanced with energy efficiency to prolong battery life. This can rapidly consume power and where, as a result, devices may not be practical to use for the given task due to these demand on system resources. Power-efficient deep learning models are developed to harmonize energy consumption and accurate classification. For example, dynamic voltage scaling (DVS) and adaptive computation scaling adjust power levels in response to workload requirements, which can be a significant energy saver. Integrating low-power AI chips in AR devices improves battery efficiency by eliminating redundant computations. Furthermore, energy-efficient optimizations keep AR applications running without draining a device's power leading to a better user experience.

### G. Enhancing real-time performance with parallel processing

By distributing computational tasks across multiple processing units, parallel processing techniques improve the efficiency of AR object recognition. For parallel recognition, multi-threading (both server and client side) and distributed computing help them do so, ultimately significantly reducing recognition latency. Batch processing collects and optimizes multiple frames to maximize inference performance across the entire batch. In 2015, they proposed a parallelization strategy to improve real-time recognition tasks so that AR can be smoothly executed on devices even in complicated scenarios. AR applications enhance experiences with fast response time and accurate detection by making use of multi-core processors and high-speed memory access.

### H. Trade-Offs between accuracy and performance in ar recognition

HP models usually use more processing power and can be high in latency and energy consumption. Lightweight models offer the efficiency with a decrease in detection performance. An option to mitigate this problem is to pick model architectures, optimization strategies, and devices that meet the needs AR applications. Performance benchmarking aids in evaluating these trade-offs, enabling AR systems to provide the highest recognition accuracy possible with no sacrifice of real-time responsiveness. Performance benchmarking helps evaluate these trade-offs so that AR systems can achieve optimal recognition precision without at the expense of real-time performance.

## 5. Conclusions & Future Research

### A. Conclusion

Adaptive testing and performance optimization can be the backbone of real time object recognition at the edge node in AR apps. It differentiates from standard computer vision as the object recognition under the AR environments should address the specific challenges contributed by the real-world scene variations, such as different lightness degrees, motion blurs, occlusions, and computation limitation conditions, which explicitly impact the recognition accuracy and the task's promptness. This motivates the need for automatic test case generation, a simulation-based testing pipeline, and a platform to compare AR object detection algorithms in-the-wild. Moreover, optimization methods like model pruning, edge computing, adaptive learning, and hardware acceleration facilitate real-time performance with minimal computational overhead. This combination of machine learning and AR leads to effective algorithms to provide high accuracy and efficiency in its wide applications in analytics, mapping, gaming, navigation, and robotics, etc. With continued advancements in AR, optimizing these methods will be essential for delivering integrated and engaging user experiences. Adaptive learning paradigms and real-time optimization frameworks will contribute to the scalability and reliability of AR-based object recognition systems.

### B. Future Direction

Indeed, AR object recognition should also direct future research on the development of AI-driven self-optimizing models that dynamically adjust to environmental variations. Reinforcement learning-based adaptive testing can improve robustness of object detection under real world settings. Furthermore, federated learning could facilitate privacy-preserving model updates by improving the recognition accuracy of AR without sacrificing the security of user data. The optimization of edge devices via lightweight neural networks will continue to improve real-time processing while decreasing dependency on cloud computing. A new direction could be multi-sensor fusion, involving LiDAR, depth sensors and visual computing to enhance object identification in dynamic settings. 5G and edge AI will also power future AR applications, enabling low-latency and real-time interaction. With increasing AR adoption across different industries, interdisciplinary team-ups between AI researchers, hardware engineers, and software developers will promote innovations, enabling AR object recognition systems to be more adaptive, efficient, and scalable.

**6. References**


1. Krauß V, Boden A, Oppermann L, Reiners R. Current practices, challenges, and design implications for collaborative AR/VR application development. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems; 2021. p. 1-15.
2. Tanaya M, Yang KK, Christensen T, Li S, O'Keefe M, Fridley J, Sung K. A framework for analyzing AR/VR collaborations: An initial result. In: 2017 IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA); 2017. p. 111-116.
3. Lacoche J, Villain E, Foulonneau A. Evaluating usability and user experience of AR applications in VR simulation. Front Virtual Real. 2022;3:881318.
4. Ashtari N, Bunt A, McGrenere J, Nebeling M, Chilana PK. Creating augmented and virtual reality applications: Current practices, challenges, and opportunities. In: Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems; 2020. p. 1-13.
5. Wang X, Liu S, Wang X, Xie J, Wang B, Wang Z. Research and test on key technologies of intelligent monitoring and control driven by AR/VR for fully mechanized coal mining face. J China Coal Soc. 2022;47(2):969-985.
6. Li K, Lake A. 31-3: Eyebox evaluation in AR/VR near-eye display testing. In: SID Symposium Digest of Technical Papers; 2019. p. 434-437.
7. Machała S, Chamier-Gliszczyński N, Królikowski T. Application of AR/VR technology in Industry 4.0. Procedia Comput Sci. 2022;207:2990-2998.
8. Nowacki P, Woda M. Capabilities of ARCore and ARKit platforms for AR/VR applications. In: Engineering in Dependability of Computer Systems and Networks: Proceedings of the Fourteenth International Conference on Dependability of Computer Systems DepCoS-RELCOMEX; 2019 Jul 1-5; Brunów, Poland. Springer International Publishing; 2020. p. 358-370.
9. Eisenberg E, Jensen J. Measuring and qualifying optical performance of AR/VR/MR device displays and addressing the unique visual requirements of transparent AR/MR displays. In: Optical Architectures for Displays and Sensing in Augmented, Virtual, and Mixed Reality (AR, VR, MR); 2020. p. 211-229.
10. Tan Y, Xu W, Li S, Chen K. Augmented and virtual reality (AR/VR) for education and training in the AEC industry: A systematic review of research and applications. Buildings. 2022;12(10):1529.
11. Sosnilo AV, Kreer MY, Petrova VV. AR/VR technologies in management and education. Upravlenie. 2021;9(2):114-124.
12. Koumaditis K, Venckute S, Jensen FS, Chinello F. Immersive training: Outcomes from small scale AR/VR pilot-studies. In: 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR); 2019. p. 1-5.
13. Kleiber M, Alexander T, Winkelholz C, Schlick CM. User-centered design and evaluation of an integrated AR-VR system for tele-maintenance. In: 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC); 2012. p. 1443-1448.
14. Pereira MF, Prahm C, Kolbenschlag J, Oliveira E, Rodrigues NF. Application of AR and VR in hand rehabilitation: A systematic review. J Biomed Inform. 2020;111:103584.
15. Piroozfar P, Farr ERP, Essa A, Boseley S, Jin R. Augmented reality (AR) and virtual reality (VR) in construction industry: An experiential development workflow. In: The Tenth International Conference on Construction in the 21st Century (CITC-10); 2018. p. 0-0.
16. Krogfoss B, Duran J, Perez P, Bouwen J. Quantifying the value of 5G and edge cloud on QoE for AR/VR. In: 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX); 2020. p. 1-4.
17. Beckmann J, Menke K, Weber P. Holistic evaluation of AR/VR-trainings in the ARSuL-project. In: INTED2019 Proceedings; 2019. p. 4317-4327.
18. Boutros F, Damer N, Raja K, Ramachandra R, Kirchbuchner F, Kuijper A. On benchmarking iris recognition within a head-mounted display for AR/VR applications. In: 2020 IEEE International Joint Conference on Biometrics (IJCB); 2020. p. 1-10.
19. Verhulst I, Woods A, Whittaker L, Bennett J, Dalton P. Do VR and AR versions of an immersive cultural experience engender different user experiences? Comput Human Behav. 2021;125:106951.
20. Pratama GN, Triyono MB, Hasanah N, Kenzhaliyev OB, Kosherbayeva AN, Kassymova GK, Azman MNA. Development of construction engineering teaching course by using VR-AR: A case study of Polytechnic in Indonesia. Int J Interact Mob Technol. 2022;17(14).