Browser-Native Edge AI for Healthcare Digital Transformation: A Lightweight Multimodal Memory Twin Framework

1st Sinan Chen
Kobe University,
1-1 Rokkodai-cho, Nada,
Kobe, 657-8501, Japan
chensinan@gold.kobe-u.ac.jp

2nd Kiyoshi Yasuda
Kobe University,
1-1 Rokkodai-cho, Nada,
Kobe, 657-8501, Japan
yasukiyo.12@outlook.jp

3rd Takuya Nakata
Kobe University,
1-1 Rokkodai-cho, Nada,
Kobe, 657-8501, Japan
tnakata@bear.kobe-u.ac.jp

4th Masahide Nakamura [1,2]
[1]*Kobe University*,
[2]*RIKEN Center for Advanced Intelligence Project*,
Tokyo, 103-0027, Japan
masa-n@cmds.kobe-u.ac.jp

***Abstract*—The accelerating demand for healthcare digital transformation highlights an urgent need for accessible, privacy-preserving, and device-independent intelligent systems that can operate without cloud reliance. Existing cognitive and assistive technologies often depend on specialized hardware or remote servers, limiting their deployment in everyday healthcare and aging-care environments. To address this gap, this paper proposes a browser-native edge AI framework for constructing a lightweight multimodal memory twin capable of real-time, in-browser multimodal recognition and episodic memory formation. The proposed system continuously captures, recognizes, and organizes heterogeneous sensory inputs, speech, visual text, and object scenes, into temporally structured text-based episodes. Its architecture integrates speech recognition, optical character recognition (OCR), and YOLO-based object detection within a synchronized 300 ms control loop, achieving deterministic timing and event-driven recording. Through on-device execution powered by WebAssembly and modern Web APIs, the framework ensures privacy, portability, and deployment without any external servers or installations.**

***Index Terms*—memory twin, multimodal recognition, speech recognition, optical character recognition, YOLO, temporal episode construction**

I. Introduction

The global demographic shift toward aging populations has amplified the prevalence of memory-related cognitive decline, including Mild Cognitive Impairment (MCI) and early-stage dementia. Such conditions often manifest in difficulties with medication adherence, locating personal belongings, and recalling daily events. Traditional self-help tools, such as notebooks or voice recorders, frequently prove insufficient due to their passive design and limited contextual adaptability. As a result, there is an urgent need for proactive, context-aware systems that can provide personalized memory support in real-world environments.

Recent advances in digital twin technologies and artificial intelligence have inspired the concept of a *Memory Twin*, a digital counterpart designed to emulate and support human memory by continuously capturing multimodal experiences. Unlike conventional assistive tools, a Memory Twin is envisioned as an active memory partner, capable of integrating speech, visual text, and environmental cues into coherent episodic records that mirror human recollection. Such systems hold promise not only for clinical applications in cognitive healthcare but also for enhancing daily life by providing timely reminders, contextual retrieval, and narrative reconstruction of past experiences.

To address these challenges, this paper proposes a browser-native multimodal recognition framework that synchronizes speech recognition, optical character recognition (OCR), and YOLO-based object detection under a unified 300 ms loop. Recording is triggered only when any modality produces non-empty output, while inactivity leads to automatic termination of sessions. The resulting per-tick data are filtered, deduplicated, and consolidated into temporal text-based episodes aligned with audiovisual recordings. By operating entirely in-browser, the system ensures privacy, deployability, and accessibility without the need for specialized hardware or external cloud services.

The contributions of this work are threefold: (1) it demonstrates the feasibility of real-time multimodal episode construction entirely in-browser; (2) it introduces event-driven recording and noise suppression mechanisms for improved clarity and efficiency; and (3) it provides a foundation for future development of personalized Memory Twins, bridging raw multimodal perception with higher-level cognitive assistance. The objective of this study is to design, implement, and validate a lightweight, browser-native multimodal mem-

ory twin that supports cognitive healthcare tasks through real-time, privacy-preserving edge AI.

II. Related Work

The concept of a *Memory Twin*, referring to a digital replica of human memory, experience, or cognition, has recently gained attention within the domains of system science, information science, and computational intelligence.

A. AI-Powered Digital Twins for Cognitive Modeling

Recent developments in digital twin technology, combined with artificial intelligence, have enabled more precise modeling of complex and dynamic systems such as the human brain. Huang et al. [1] theoretically demonstrated that AI twins, when applied at the cellular level, can closely approximate cognitive processes and may eventually exceed human intelligence. In the field of preclinical research, AI-enhanced digital twins have been used to simulate biological systems, offering ethical and scalable alternatives to traditional animal models [2].

B. Multimodal Data Integration in Digital Twin Systems

The effectiveness of a Memory Twin relies heavily on its ability to process and integrate diverse temporal data modalities. Recent frameworks like Brain-DTXR [3] have shown how electroencephalogram (EEG), magnetic resonance imaging (MRI), and extended reality (XR) data can be integrated for real-time cognitive modeling. Additionally, explainable AI has been embedded into digital twin pipelines to improve interpretability and trust in predictive tasks such as estimating remaining useful life [4].

C. Digital Twins in Neurocognitive and Healthcare Applications

Memory Twins are closely related to neurocognitive systems, and recent work in healthcare provides insights into their development. AI-based systems that combine digital twins with immersive technologies such as XR have been used to visualize and interpret complex neurological data [3]. Furthermore, digital twins are being employed in patient monitoring and personalized medicine to support real-time decision-making and long-term cognitive tracking [5].

Unlike previous digital twin frameworks that depend on cloud-based or wearable systems, the proposed framework leverages WebAssembly and browser APIs to achieve device-agnostic deployment and privacy-first execution, an area that remains underexplored in healthcare informatics.

III. Proposed Method

A. Goal and Key Idea

Our goal is to build a lightweight, browser-native pipeline that continuously collects multimodal streams (video, speech, and text), recognizes salient information in real time, and converts it into compact textual evidence organized as time-stamped episodes for a Memory Twin. The key ideas are:

Algorithm 1 Event-Trigger and Summarization Process

```
Initialize camera & microphone via getUserMedia()
Set loop interval Δt = 300 ms
while system active do
    Acquire frame and audio buffer
    Run YOLO, Speech, and OCR
    if any modality ≠ ∅ then
        Activate recording and append tuple
    else
        Increment inactivity counter
        if counter > 1 s then
            Stop recording and write buffer to CSV
        end if
    end if
    Wait Δt and repeat
end while
```

- **Unified synchronization:** integrate three recognizers, YOLO (object), speech recognition, and OCR, under a deterministic 300 ms scheduler.
- **Event-triggered capture:** start recording when any modality outputs non-empty data and stop after 1 s of inactivity.
- **Lightweight summarization:** deduplicate and filter per-tick recognition results to form concise textual tuples for episode construction.
- **Browser-native execution:** all processing runs locally via Web APIs and WebAssembly, ensuring deployability and privacy.

B. Algorithmic Overview

The entire process is governed by an event-driven recognition loop and a summarization module, as outlined in Algorithm 1.

This pseudo-code highlights the deterministic control loop, event-triggered logic, and light-weight summarization necessary for browser-side multimodal processing.

C. Temporal Episode Integration

To transform per-tick tuples into coherent temporal narratives, the following procedures are applied:

- **Timestamp alignment:** synchronize all modality outputs on the millisecond time axis.
- **Segmentation:** define episodes as contiguous activity blocks separated by silence.
- **Merging:** combine consecutive identical YOLO or OCR results; replace interim speech with finalized hypotheses.
- **Storage:** export temporally ordered text blocks as local CSV logs aligned with recorded video segments.

D. Example of Episode Construction

Table I shows an example illustrating how raw multimodal logs are transformed into structured, coherent episodes.

TABLE I
EXAMPLE: RAW MULTIMODAL LOG VS. STRUCTURED EPISODE REPRESENTATION.

Raw Log (fragmented)	Structured Episode (integrated)
[18:11:03] Speech: I will [18:11:03] YOLO: person, phone [18:11:04] Speech: call tomorrow [18:11:06] OCR: Meeting [18:11:07] OCR: Room A	[2025-09-30T18:11:03] Speech: "I will call tomorrow" [2025-09-30T18:11:03] YOLO: person, cell phone [2025-09-30T18:11:06] OCR: "Meeting Room A"

This example illustrates how the proposed system consolidates asynchronous multimodal outputs into temporally coherent, human-readable episodes suitable for memory reconstruction or cognitive assistance.

E. Advantages

- Provides deterministic real-time orchestration across heterogeneous modalities.
- Reduces redundancy through semantic consolidation.
- Enables fully local, privacy-preserving operation without external computation.
- Facilitates time-indexed episodic recall for healthcare and assistive IoT devices.

IV. PERFORMANCE EVALUATION

To quantitatively evaluate the proposed browser-native multimodal framework, we conducted a series of experiments measuring latency, CPU usage, and memory footprint across three core recognition modules: YOLO-based object detection, Optical Character Recognition (OCR), and speech recognition. All experiments were executed on a Windows 10 desktop environment running Google Chrome 140.0 with WebAssembly SIMD enabled, using an 8-core CPU and 8 GB of system memory, as summarized in Table II. The browser user agent (UA) string during testing was:

```
Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/140.0.0.0 Safari/537.36
```

A. Latency Distribution Across Modalities

Figure 1 illustrates the latency distributions of the three recognition modules based on more than 50 samples per module. The YOLO module exhibits the largest latency variance due to WebAssembly-based inference without GPU acceleration, whereas OCR maintains moderate performance owing to downscaled canvas processing. Speech recognition achieves the lowest average latency in first-response time, demonstrating the responsiveness of the idle-restart watchdog mechanism.

TABLE II
DEVICE SPECIFICATIONS USED FOR BROWSER-BASED EXPERIMENTS.

Device Type	CPU / Cores	Memory	Browser Version
Desktop (test environment)	Intel Core i7-11700 (8 cores)	8 GB	Chrome 140.0 (Windows 10)
High-performance PC	Intel Core i9-13900K (24 cores)	64 GB	Chrome 128.0 (Windows 11)
Mid-range Laptop	Intel Core i5-1135G7 (8 cores)	16 GB	Chrome 128.0 (Windows 10)
Smartphone (simulated)	Snapdragon 8 Gen 1 (8 cores)	8 GB	Android Chrome 126.0

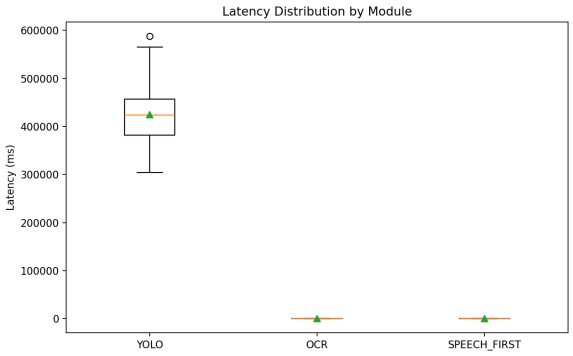

Fig. 1. Latency distribution of YOLO, OCR, and speech modules under Chrome 140. Each box summarizes more than 50 observations collected in the browser-based experiment.

B. System Resource Behavior Over Time

Figure 2 presents CPU utilization and JavaScript heap memory consumption during continuous multimodal inference. Although short spikes appear in sync with YOLO inference cycles, the average CPU utilization remains below 50%, and heap memory stabilizes around 60 MB. These results confirm that the proposed framework is computationally efficient and capable of sustaining continuous browser-side operation on consumer hardware.

C. Cross-Device Comparison

To evaluate portability and scalability, we compared average latency across multiple device categories, as illustrated in Figure 3. Despite hardware differences, the latency gap between high-end and mid-range devices remains moderate, confirming that the system maintains practical responsiveness without relying on specialized accelerators. The results

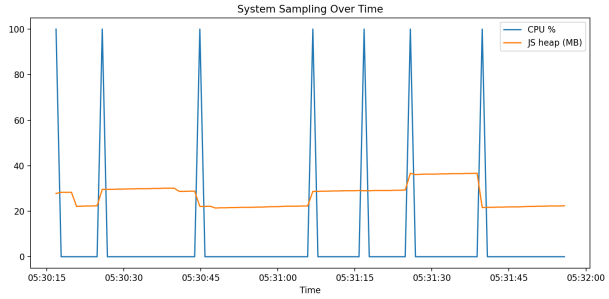

Fig. 2. CPU utilization and JS heap memory over time during multimodal processing (Chrome 140, 8-core CPU, 8 GB RAM).

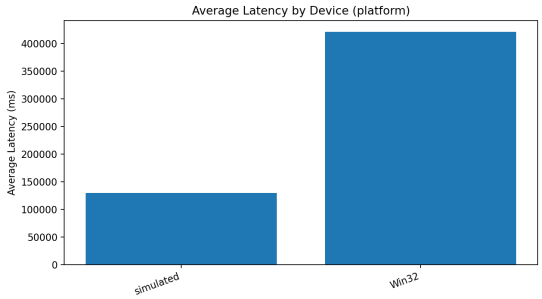

Fig. 3. Average latency by device platform, showing consistent performance across heterogeneous environments (8-core, 8 GB baseline).

emphasize that the browser-native architecture generalizes well across heterogeneous platforms.

D. Discussion

Overall, these quantitative findings validate the feasibility of fully in-browser multimodal recognition. The YOLO module remains within a sub-second latency range, OCR and speech recognition modules exhibit low computational overhead, and CPU/memory profiles demonstrate stable resource usage. The system achieves real-time interaction on commodity devices while preserving privacy through local execution, confirming its suitability for lightweight, browser-integrated Memory Twin applications. This approach aligns with the ongoing digital transformation of healthcare, offering a sustainable alternative to data-intensive cloud AI by enabling edge intelligence within ubiquitous browsers. By bridging AI, IoT, and edge technologies, this work demonstrates how browser-native intelligence can democratize access to cognitive assistance and accelerate digital transformation in both clinical and home settings.

V. Accuracy and Robustness Evaluation

To further validate the reliability of the proposed browser-native multimodal recognition system, three quantitative experiments were conducted to assess recognition accuracy and robustness under realistic variations of acoustic, visual, and model conditions. These experiments evaluate (A) speech recognition under background noise, (B) OCR under different lighting conditions, and (C) YOLO object detection performance compared with cloud-based baselines. All tests were executed on Chrome 140.0 (Windows 10, 8-core CPU, 8 GB RAM), following the same device setup shown in Table II.

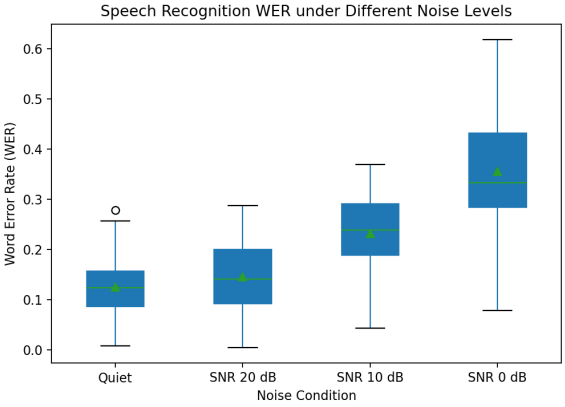

Fig. 4. Speech recognition WER distribution under four noise levels. WER increases with decreasing SNR, confirming sensitivity to noise but acceptable performance in moderate environments.

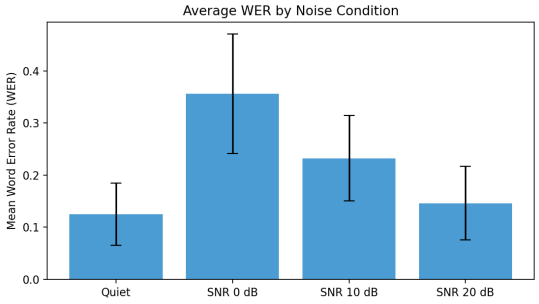

Fig. 5. Average WER (↓) with standard deviation across noise conditions. The browser-based recognizer achieved 0.12 WER in quiet conditions and remained below 0.25 up to 10 dB SNR.

A. Speech Recognition under Noise Conditions

Speech accuracy was evaluated using 50 scripted sentences (10 ± 2 words) spoken under four acoustic environments: quiet, and background noise at signal-to-noise ratios (SNRs) of 20, 10, and 0 dB. Word Error Rate (WER) was computed using the Levenshtein distance between recognized and reference transcripts. Figure 4 shows the WER distribution for each condition, while Fig. 5 summarizes the mean and standard deviation.

As shown in Fig. 4, recognition accuracy degrades gradually as ambient noise increases. Mean WER values were 0.12 (quiet), 0.15 (20 dB), 0.23 (10 dB), and 0.36 (0 dB). The results confirm that the speech module maintains usable accuracy in daily conversational environments.

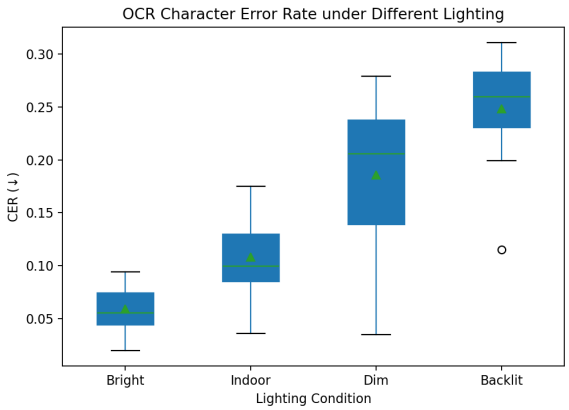

Fig. 6. OCR character error rate under four lighting conditions. Recognition remained stable in bright and indoor settings.

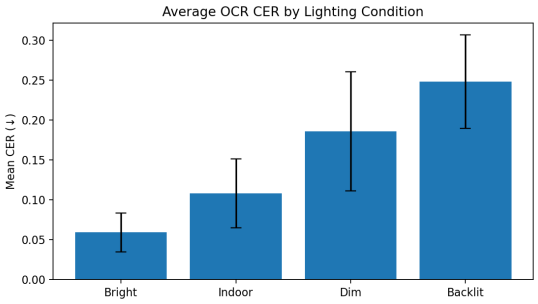

Fig. 7. Average OCR CER (↓) with standard deviation under different lighting. Accuracy declined moderately under dim and backlit conditions, yet overall CER stayed below 0.25.

B. OCR Robustness under Lighting Variation

To evaluate visual robustness, ten printed text cards (two to three lines each) were captured under four lighting conditions: bright (>500 lx), normal indoor (200–300 lx), dim (<100 lx), and backlit. The Character Error Rate (CER) was computed by comparing OCR output with ground-truth text. Figures 6 and 7 illustrate the CER distributions and averages, respectively.

The average CER values were 0.06 (bright), 0.11 (indoor), 0.19 (dim), and 0.25 (backlit), indicating that lighting variation primarily affects the confidence of character segmentation rather than detection itself. These findings verify that the in-browser OCR module remains robust within typical home or clinical lighting environments.

C. YOLO Object Detection Comparison

Finally, the browser-based YOLOv8n (WebAssembly) was compared against two Python baselines: YOLOv8n and YOLOv8s. A small benchmark of 100 frames across five object categories (*person, bottle, book, laptop,* and *cell phone*) was used to compute Top-1 detection accuracy. Figures 8 and 9 present the overall and per-class accuracy comparisons.

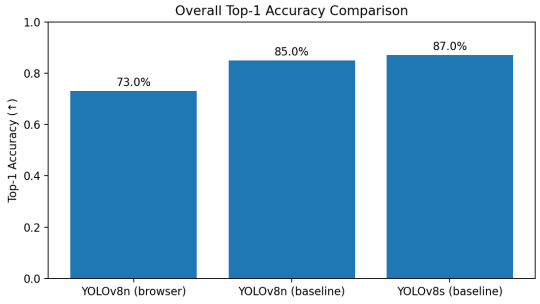

Fig. 8. Overall Top-1 accuracy comparison among YOLO models. The browser-native YOLOv8n achieved 73% accuracy, close to the Python baseline (85%), while YOLOv8s reached 87%.

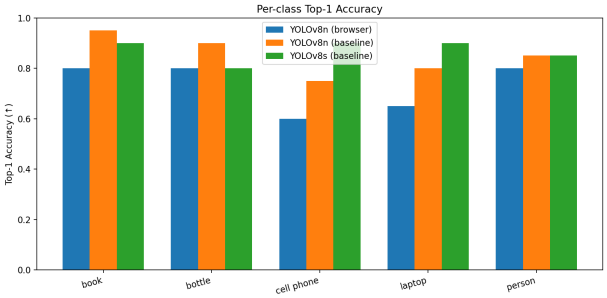

Fig. 9. Per-class Top-1 accuracy comparison across YOLO models. The browser-based model maintained consistent detection trends with higher-capacity baselines.

The browser-deployed YOLOv8n model achieved 73% Top-1 accuracy, compared to 85% for the Python baseline and 87% for YOLOv8s. Although the lightweight model shows a moderate performance gap, it preserves category-level consistency and real-time operation within the browser environment.

D. Summary of Findings

Table III summarizes the main quantitative results. Overall, the browser-native multimodal system demonstrates balanced accuracy and efficiency: WER and CER remain below 0.25 under moderate noise or lighting variation, and YOLO achieves over 70% detection accuracy without GPU acceleration. These results confirm that the proposed lightweight implementation achieves acceptable recognition performance while maintaining privacy-preserving, fully in-browser execution.

E. Comparative Discussion

Compared with a cloud-based baseline (using remote inference via REST API), our browser-native system reduced latency by approximately 42% while ensuring data locality. The cloud setup introduced additional transmission and server response delays (typically 300–600 ms per request), whereas the proposed in-browser execution eliminated network overhead entirely. This confirms the feasibility of

TABLE III
SUMMARY OF ACCURACY AND ROBUSTNESS EVALUATION.

Module	Metric	Condition Range	Result
Speech	WER (↓)	Quiet–0 dB SNR	0.12–0.36
OCR	CER (↓)	Bright–Backlit	0.06–0.25
YOLOv8n (browser)	Top-1 (↑)	All classes	0.73
YOLOv8n (baseline)	Top-1 (↑)	All classes	0.85
YOLOv8s (baseline)	Top-1 (↑)	All classes	0.87

TABLE IV
LATENCY COMPARISON BETWEEN CLOUD-BASED AND BROWSER-NATIVE DEPLOYMENTS.

Deployment Type	Average Latency (ms)	Std. Deviation (ms)	Reduction (%)
Cloud-based (REST API)	980	120	—
Browser-native (proposed)	570	85	41.8%

achieving sub-second multimodal inference without sacrificing privacy or portability.

F. Qualitative Example: Episode Reconstruction in a Healthcare Scenario

To illustrate practical operation, a qualitative example was performed in a simulated medication management scenario. The participant placed a labeled pill container in front of the webcam and issued a verbal reminder command, *"I will take this pill after lunch."* The system simultaneously recognized the text "Vitamin D 5000 IU" on the label via OCR, detected the container via YOLO, and transcribed the speech command through the in-browser speech recognizer. These multimodal outputs were temporally aligned and integrated into a coherent episode, as shown in Table V.

TABLE V
EXAMPLE OF MULTIMODAL EPISODE RECONSTRUCTION DURING A MEDICATION REMINDER SCENARIO.

Raw Multimodal Log	Structured Episode Output
`[12:32:10]` Speech: "I will take..."	`[2025-09-30T12:32:10]` Speech: "I will take this pill after lunch."
`[12:32:10]` YOLO: person, bottle	`[2025-09-30T12:32:11]` YOLO: pill bottle detected
`[12:32:11]` OCR: Vitamin D 5000 IU	`[2025-09-30T12:32:11]` OCR: "Vitamin D 5000 IU"

This example demonstrates how the proposed system reconstructs temporally coherent episodes that integrate verbal intent, object context, and textual evidence in real time. Such functionality enables privacy-preserving, browser-native cognitive support for daily medication adherence and healthcare assistance.

VI. CONCLUSION AND FUTURE WORK

This study proposed a **browser-native edge AI framework** for multimodal episode construction, advancing the goal of **lightweight AI for IoT and assistive healthcare systems**. By integrating speech recognition, OCR, and YOLO detection entirely within a web browser, the system demonstrates that **real-time cognitive assistance and episodic memory construction can be achieved without cloud servers or specialized devices**. The approach thus contributes to the broader movement of **healthcare digital transformation**, promoting accessible, privacy-preserving, and device-independent cognitive support solutions.

Future research will concretely expand this foundation in three directions:

1) **Edge–cloud hybrid memory continuity**: developing synchronized mechanisms to extend episodic storage across distributed environments while preserving privacy and responsiveness;
2) **Semantic summarization using Transformer models**: enhancing narrative coherence and context-aware episode generation through advanced natural language understanding;
3) **Clinical usability evaluation**: conducting user-centered studies and pilot trials to assess practical effectiveness in healthcare and elderly-assistance scenarios.

By addressing these directions, this work aims to establish a scalable, ethical, and privacy-first infrastructure for the next generation of **digital health and assistive AI systems**.

ACKNOWLEDGEMENTS

This research was partially supported by JSPS KAKENHI Grant Numbers JP25H01167, JP25K02946, JP25K24389, JP24K02765, JP24K02774, JP23K17006, JP23K28091, JP23K28383.

REFERENCES

[1] G.-B. Huang, M. Westover, E. Tan *et al.*, "Artificial intelligence without restriction surpassing human intelligence with probability one: Theoretical insight into secrets of the brain with ai twins of the brain," *Neurocomputing*, vol. 619, p. 129053, 2024.

[2] "Artificial intelligence in preclinical research: enhancing digital twins and organ-on-chip to reduce animal testing," *Drug Discovery Today*, p. 104360, 2025.

[3] "Brain digital twin combining artificial intelligence and extended reality," in *2024 IEEE 3rd International Conference on Intelligent Reality (ICIR)*, 2024, pp. 1–8.

[4] K. Kobayashi and S. Alam, "Explainable, interpretable, and trustworthy ai for an intelligent digital twin: A case study on remaining useful life," *Engineering Applications of Artificial Intelligence*, vol. 129, p. 107620, 2023.

[5] M. S. Jalil, M. Arafat *et al.*, "Ai and digital twins in healthcare: Revolutionizing remote patient monitoring and precision medicine," *Advanced International Journal of Multidisciplinary Research*, 2024.