Recognizing and Recording Grasped Objects for Assisting Elderly at Home to Find Missing Items

Toshinori Shindo*, Takuya Nakata†, Sinan Chen†, Sachio Saiki‡, Kiyoshi Yasuda* and Masahide Nakamura†
*Graduate School of Engineering, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe 657-8501, Japan
Email: shintoshi@es4.eedept.kobe-u.ac.jp, yasukiyo.12@outlook.jp
†Center of Mathematical and Data Sciences, Kobe University, 1-1 Rokkodai-cho, Nada-ku, Kobe 657-8501, Japan
Email: tnakata@bear.kobe-u.ac.jp, chensinan@gold.kobe-u.ac.jp, masa-n@cmds.kobe-u.ac.jp
‡School of Data and Innovation, Kochi University of Technology, 185 Miyanokuchi, Tosayamada, Kami, Japan
Email: saiki.sachio@kochi-tech.ac.jp

***Abstract*—In Japan, the number of elderly people and dementia patients is on the rise. They often face the issue of forgetting where they placed their belongings. When they are unable to remember the location of an item, not only do they themselves have to search for it, but their family members and caregivers are also involved in the search, increasing the burden on everyone involved. Therefore, reducing this burden has become a significant challenge. To address this issue, this study proposes a "Grasped Object Recognition and Recording Service" that utilizes cameras installed in the room and image recognition technology to identify and record objects grasped by individuals. The implementation focused on the recognition and recording of the grasped objects. As a result, the system successfully achieved a practical level of accuracy in recognizing and recording the objects held by people in the room.**

***Index Terms*—forgetting the location of objects, elderly people, dementia, large language model.**

I. Introduction

It is estimated that the proportion of elderly people in Japan is high and the number of dementia patients will continue to increase in the future [1]. As a result, it is expected that the number of elderly people and dementia patients who forget where they left their belongings will increase. Finding where they left their belongings is a burden not only for the elderly people but also for their family members and caregivers.

A related study, SenseCam, is a wearable camera that automatically records daily activities to assist memory [2]. In this study, patients with memory impairment actually used the SenseCam and found it to be effective in helping them recall everyday events. However, the purpose of this study was to assist memory in daily life, and not specifically to locate objects.

The purpose of this study is to reduce the time and effort required for elderly people and patients with dementia to find where they put things, which is caused by forgetting where they put things. To achieve this goal, we propose a "Grasped Object Recognition and Recording Service" that recognizes and records objects grasped by a person in a room using a fixed-point camera and image recognition technology, and makes the information available for viewing. To achieve the purpose of this research, the proposed method takes the following four approaches.

A1: Hand Image Extraction from Video Stream
Crop images around a person's hand using a posture estimation model on the video stream from a fixed-point camera installed in the room.

A2: Object Recognition via GPT-4o
The image around the person's hand acquired in A1 is sent to GPT-4o, a Large Language Model (LLM) capable of image input, for recognition of the object grasped by the person.

A3: Storing Recognition Results and Other Information in Database
Store the recognition results obtained in A2 and other information useful for locating objects in a database.

A4: Query and Browse Forgotten Object Data
Using the data stored in A3, the system enables users to query about objects they wish to find and to browse information about those objects.

In order to evaluate the usefulness of the above approach, we conducted an experiment by implementing functions A1 through A3. In the experiments, we verified two points: (1) whether the implemented part can actually recognize the grasped object correctly, and (2) whether extracting hand images contributed to the improvement of recognition accuracy.

II. Preliminaries

A. Increase in the Number of Elderly People and Dementia Patients

According to the "White Paper on Aging Society in 2024" [1], the total population of Japan was 124.35 million as of October 1, 2023, of which 36.23 million were aged 65 and over, and the aging rate was 29.1 percent. The aging rate is estimated to continue to rise, reaching 30.8 percent by 2030, 34.8 percent by 2040, and 37.1 percent by 2050. According to the survey conducted from 2022 to 2005, the number of the elderly aged 65 years and over with dementia was estimated to be 4,443,200 with a prevalence rate of 12.3%, and the number of the elderly with mild cognitive impairment (MCI) was estimated to be 5,585,000 with a prevalence rate of 15.5%. In 2040, the number of the elderly aged 65 years or older with dementia is estimated to be 5,842,000, with a prevalence rate of 14.9%, and the number of the elderly with MCI is estimated to be 6,128,000, with a prevalence rate of 15.6%.

B. The Problem of Elderly and Dementia Patients Forgetting Where to Put Things

One of the symptoms of age-related forgetfulness in the elderly is forgetting where they put things [3]. Even if a person forgets where he or she has placed an object, the memory of having placed the object still exists [4], and it is possible to find the object on one's own, but it is time-consuming and labor-intensive.

One of the symptoms of dementia is that the person forgets to put things away or misplaces them, and is constantly searching for them [5]. In many cases, forgetting where to put things due to symptoms of dementia is not even recognized by the patient because the experience of putting things down itself is lost [4]. Therefore, family members and caregivers often have to search for things on behalf of the patient, which takes time and effort on their part.

C. PoseNet

PoseNet is the official pose detection model of TensorFlow.js, a JavaScript library for implementing and executing machine learning [6] [7]. The model can estimate the posture of a single person or multiple people in an image or video. The model's posture estimation results include the coordinates of 17 key points on the human body, such as the wrists and hips, in the image or video, and their confidence scores.

D. Large Language Model

Large Language Model (LLM) is pre-trained, large-scale statistical language model based on neural networks. LLM is capable of understanding language and generating responses to incoming instructions and questions (prompt sentences), and is also capable of reasoning about tasks for which it have no knowledge, using few examples and prior learning knowledge [8].

E. GPT-4o

GPT-4o is an LLM model developed by OpenAI that can accept voice, image, and video input in addition to text, and can answer questions about the input images, for example [9]. GPT-4o provides an API and can be used by external programs.

F. Related Study

In this section, we introduce SenseCam: A Retrospective Memory Aid as a related study [2]. The purpose of this research is to develop a device called SenseCam, a wearable still-image camera, to automatically record daily life and assist memory. The device is worn around the neck and automatically records visual information. Recording is based on time and events, and the built-in sensor detects changes in the surrounding environment and automatically takes pictures. In an experiment, patients with memory impairment used SenseCam and were able to recall everyday events by viewing the recorded images.

III. Proposed Method: Grasped Object Recognition and Recording Service

A. Objective and Approach

The objective of this study is to reduce the time and effort required for elderly people, dementia patients, their families, and caregivers to search for misplaced objects in a room. To achieve this objective, we propose the "Grasped Object Recognition and Recording Service" (GORRS). As described in Section II-B, elderly people and dementia patients often forget where they placed objects. To address this issue, GORRS automatically recognizes and records objects grasped by individuals in a room using fixed-point cameras and image recognition technology. This information is made available to users, making it easier to locate objects even if their placement is forgotten. GORRS must meet the following two requirements:

R1: Automatically recognize and record objects grasped by individuals in the room

Elderly people and dementia patients often find it difficult to remember the placement of objects for extended periods. Additionally, they may face challenges in handling digital devices. Therefore, GORRS should automatically recognize and record objects grasped by individuals in the room.

R2: Allow users to query and browse information about recorded objects

Elderly people, dementia patients, their families, and caregivers often spend significant time and effort searching for misplaced objects. By utilizing the recorded information about objects, this burden can be reduced. Thus, GORRS should enable users to query and browse the recorded information.

To meet these requirements, this study adopts the following four approaches. Each approach implements a function to satisfy the requirements.

- A1: Hand Image Extraction from Video Stream
- A2: Object Recognition via GPT-4o
- A3: Storing Recognition Results and Other Information in Database
- A4: Query and Browse Forgotten Object Data

B. Overall Architecture

The overall architecture of the system is shown in Figure 1. The system consists of GORRS and users. GORRS uses PoseNet and GPT-4o to recognize and record objects grasped by users. Users can query GORRS about misplaced objects and browse information about those objects.

C. A1: Hand Image Extraction from Video Stream

As a preliminary step, A1 extracts images around a person's hands from the video stream of a fixed-point camera installed in the room. This process aims to improve the recognition accuracy of GPT-4o by increasing the proportion of the grasped object in the image.

First, a webcam is installed on the wall or ceiling to capture the entire living area in the room. Next, PoseNet is applied to the video stream at regular intervals T [s] to estimate the

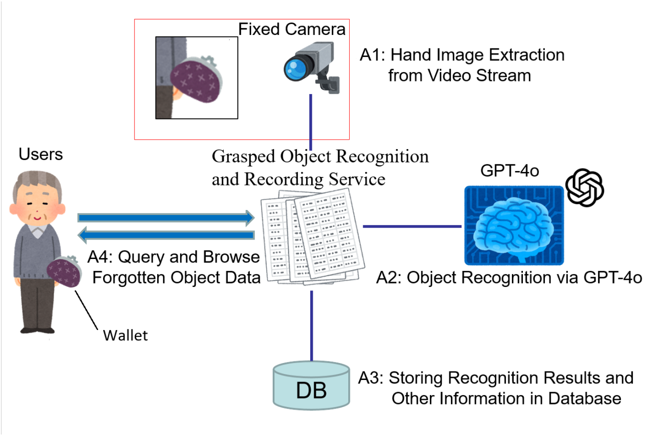

Fixed Camera
A1: Hand Image Extraction from Video Stream
Users
Grasped Object Recognition and Recording Service
GPT-4o
A4: Query and Browse Forgotten Object Data
A2: Object Recognition via GPT-4o
Wallet
DB
A3: Storing Recognition Results and Other Information in Database

Fig. 1. Overall Architecture of the System.

posture of one person in the room. If the confidence scores of both the wrist and elbow of each hand exceed a certain threshold, images around the hands are extracted as square images. To ensure that the grasped object fits within the image, the position and side length of the square are adjusted.

The position of the square is centered at a certain distance offset from the wrist coordinates in the direction from the elbow to the wrist. By doing so, the center of the cropped image is closer to the palm, allowing the entire grasped object to fit within the image.

The side length of the square is determined by multiplying the distance between the left shoulder and left hip by an adjustment factor α, provided the confidence scores of the left shoulder and left hip exceed a certain threshold. If the calculated value is smaller than the minimum side length *minSide* [pixel], or if the confidence scores of the left shoulder and left hip are below the threshold, the side length is set to *minSide* [pixel].

D. A2: Object Recognition via GPT-4o

In A2, the images obtained in A1 are sent to GPT-4o via an API for object recognition, and the results are retrieved.

GPT-4o, an LLM capable of image input, can infer and describe objects without prior training by using other contexts. Additionally, by inputting prompts along with images, the output can be controlled, such as asking for the characteristics of the object.

The images obtained in A1 are sent to GPT-4o along with the prompt, then the recognition results are obtained as output from GPT-4o. The intention behind the prompt is as follows:

- “What are you holding in your hand? Answer in one word.”: This part prompts GPT-4o to recognize the object being held in the image and respond with the object name as a single word. Answering in a single word prevents unnecessary output.
- “If ambiguous, describe the characteristics of the object.”: This part ensures that even if GPT-4o fails to recognize the object name, it can describe the characteristics of the object, enabling queries in GORRS.
- “If holding nothing, answer ‘none’. Do not include line breaks.”: This part fixes the output when holding nothing and prevents line breaks in the output, making data organization in GORRS easier.

E. A3: Storing Recognition Results and Other Information in Database

In A3, the recognition results obtained in A2, along with other information such as timestamps, are saved in a database for both hands. Storing this data in a database enables efficient management and facilitates user queries and the provision of information useful for locating objects.

The columns of the hand_data table, along with their purposes, are listed below:

- user_id: User ID. Links grasped objects to users.
- time: Timestamp of data saving. This is almost equal to the time the object was grasped. Enables time-based queries.
- x_l, x_r, y_l, y_r: x and y coordinates of the left and right wrists, respectively. Enables browsing of visual information about the location.
- message_l, message_r: The recognition results of the grasped objects. Enables queries based on object names or characteristics.
- response_l, response_r: Information such as the number of tokens used. Provides information about the usage status of GPT-4o.
- image_l, image_r: Images around the left and right hands, respectively, extracted in A1. Users can browse the images to identify the actual grasped objects.

F. A4: Query and Browse Forgotten Object Data

In A4, the data saved in A3 is used to allow users to query GORRS about misplaced objects and browse information that helps locate them.

The query function allows users to query the service about misplaced objects in three ways and obtain candidates:

- Query by object name or characteristics: Allowing queries by characteristics increases the likelihood of obtaining information about the objects users are searching for.
- Query by time: Allowing queries by time enables users to query even if they forget the object name or fail to find candidates through queries by object name or characteristics.
- Query by history: This function enables users to browse information about objects for which candidates were not found through queries by object name or characteristics or by time.

Candidates obtained from queries are displayed on the screen if multiple candidates exist. Each candidate displays the object name or characteristics, the timestamp, and the image of the object.

The information browsing function allows users to browse the object name or characteristics, timestamp, image, and location in the room for candidates obtained from queries.

The location in the room is displayed as a point on the video stream of the fixed-point camera using the coordinates saved in the database. Visually displaying the location where the grasped object was recorded makes it easier for users to find misplaced objects.

IV. Implementation

A. Implementation Overview

In this study, the parts of the proposed method corresponding to A1 through A3 were implemented.

The following technologies were used for the implementation:

- Programming Languages: Python 3.13.0 [10], JavaScript, HTML, CSS
- Libraries and Tools: FastAPI [11], SQLAlchemy [12], TensorFlow.js [6], PoseNet [7], gpt-4o-2024-11-20 [9]
- External Database: MySQL [13]

In this study, the MX BRIO webcam from logicool [14] was used. The main specifications of this camera are as follows:

- Field of View (FOV): 90 °
- Sensor Resolution: 4K
- Maximum Frame Rate: 60FPS 1080p / 30FPS 4K

B. Implementation of A1, A2, and A3

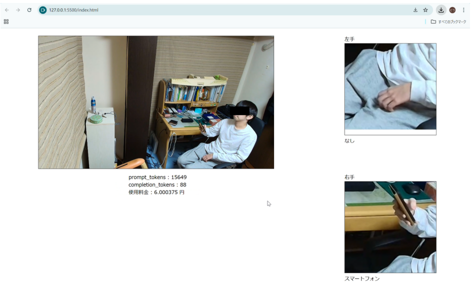
HTML and JavaScript were used to implement function A1. The video stream from the webcam was set to 1080p resolution and a frame rate of 30 fps. The recognition interval was set to T = 4.0 s, the adjustment factor $\alpha = 0.8$, and the minimum side length *minSide* = 150 pixels.

Python and FastAPI were used to implement function A2, and an API was created to enable JavaScript to send prompt sentences and images to GPT-4o and receive output.

Python, FastAPI, SQLAlchemy, and MySQL were used to implement function A3, and an API was created to enable JavaScript to save data to the hand_data table in MySQL.

C. Other Implementation Parts

To verify the operation of the implemented components, a preview screen was created to view the video stream from the webcam, the extracted images, and the recognition results on a browser. An example of the implemented preview screen is shown in Figure 2.


[redacted]

prompt_tokens : 15649
completion_tokens : 88
使用料金 : 6.000375 円

Fig. 2. An Example of the Implemented Preview Screen.

V. Evaluation Experiment

A. Overview and Content of the Experiment

The purpose of the evaluation experiment is to verify whether the implemented part of GORRS, which recognizes and records objects grasped by individuals, can correctly recognize and record objects grasped by individuals in a room. The evaluation experiment sets the following RQs (Research Questions) and draws conclusions:

RQ1: Can the system correctly recognize and record objects grasped by individuals in the room?

RQ2: Does extracting hand images contribute to improving recognition accuracy?

RQ1 can be further divided into the following two RQs:

RQ1-1: When individuals in the room are actually grasping objects, are those objects correctly recognized and recorded?

RQ1-2: When individuals in the room are not grasping objects, is "none" correctly recognized and recorded?

To conduct the evaluation experiment, the webcam described in Section IV-A was installed in the room. The installation environment was set to a height of $1.85\,\mathrm{m}$, a high angle, and a shooting range of $4.58\,\mathrm{m}^2$. The objects used in the experiment are shown in Figure 3.

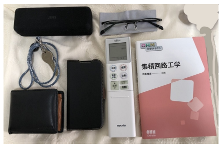


Fig. 3. Objects Used in the Experiment.

1) Experiment on RQ1-1 and RQ1-2: For the experiment on RQ1-1 and RQ1-2, the system was operated for a series of actions performed by an individual in the room. The series of actions taken by the individual are as follows:

1) Enter the room and place the key on the shelf.
2) Sit on a chair, take out a wallet from the pocket, and place it on the desk.
3) Take out a smartphone from the pocket and operate it.
4) Take out glasses from the glasses case on the desk, wipe them with a glasses cloth, and wear them.
5) Operate the smartphone while leaving the room.

In the experiment on RQ1-1, the recognition results were classified as recognition success when they matched the grasped object, recognition failure when they were clearly different, and feature response when the characteristics were described. The number of occurrences of each was investigated. The number of recognition successes included cases where the recognition results did not misrepresent the essence of the object, such as answering "case" for a glasses case. The recognition success rate was defined as the ratio of recognition successes to the number of recognition executions for each grasped object. In this experiment, considering that multiple recognitions are executed for one grasped object, it was judged that half of them being correctly recognized would be sufficient for practical use, and a recognition success rate of 50% was set as the standard.

In the experiment on RQ1-2, the number of occurrences was investigated when "none" was recognized when the individual was not grasping an object and when something other than "none" was output.

2) Experiment on RQ2: For the experiment on RQ2, the effects of the presence or absence of extracting hand images was compared. In this experiment, the recognition accuracy was compared between the images of the entire room and the images of the extracted hand surroundings when the individual was grasping a smartphone, remote control, and book. These three types of objects were selected as grasped objects to compare using various sizes of objects. The definitions of recognition success, recognition failure, and feature response were the same as in the experiment on RQ1-1, and cases where the grasped object was completely cut off during extraction were excluded from the comparison.

B. Experimental Results

1) Experimental Results on RQ1-1: A graph summarizing the experimental results on RQ1-1 is shown in Figure 4. Note that glasses and glasses cloth were grasped simultaneously, so they were aggregated as one type.

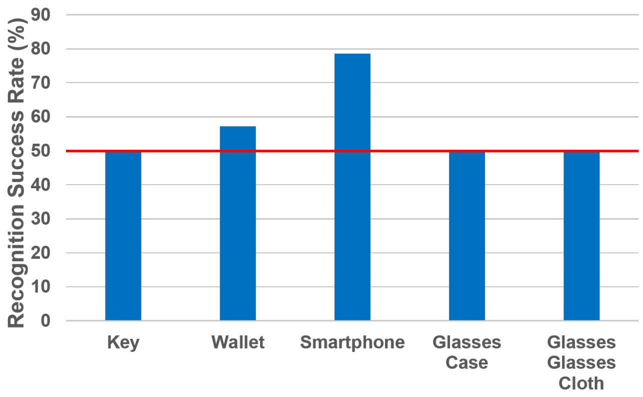


Fig. 4. Recognition Success Rate for Each Grasped Object.

As shown in Figure 4, the recognition success rate was above 50% for all grasped objects.

Additionally, the breakdown of the results of recognition execution for each grasped object, categorized into recognition failure due to misrecognition by GPT-4o and complete cutoff of the grasped object during image extraction, is summarized in Table I.

TABLE I
BREAKDOWN OF RECOGNITION RESULTS FOR EACH GRASPED OBJECT

Grasped Object	Executions	Success	Misrecognition	Cutoff	Feature
Key	4	2	1	1	0
Wallet	7	4	0	2	1
Smartphone	14	11	0	3	0
Glasses Case	4	2	1	1	0
Glasses Glasses Cloth	20	10	5	1	4

Examples of images for recognition success, feature response, misrecognition, and complete cutoff during extraction are shown in Figures 5, 6, 7, and 8, respectively.

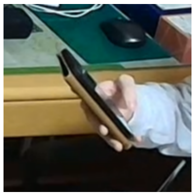


Fig. 5. Successfully Recognized Grasped Smartphone.

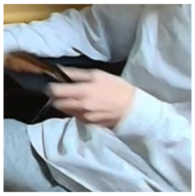


Fig. 6. Feature Described for Grasped Wallet.

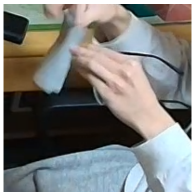


Fig. 7. Misrecognized as Controller for Grasped Glasses and Glasses Cloth.

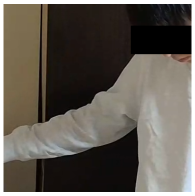

[redacted]

Fig. 8. Completely Cutoff Grasped Key.

2) Experimental Results on RQ1-2: A graph summarizing the experimental results on RQ1-2 and an example of an image for recognition failure are shown in Figures 9 and 10, respectively.

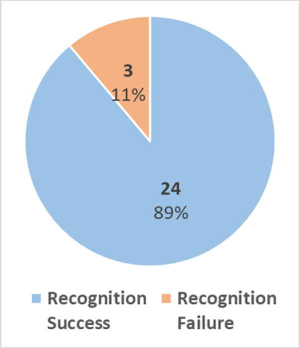


Fig. 9. Classification of Recognition Results When Not Grasping Objects.

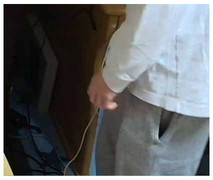


Fig. 10. Example Misrecognized as Code.

3) Experimental Results on RQ2: Tables summarizing the recognition results for each grasped object in each case are shown in Tables II and III.

TABLE II
RECOGNITION RESULTS FOR EACH GRASPED OBJECT WITHOUT IMAGE EXTRACTION

Grasped Object	Success	Failure	Feature
Smartphone	15	1	0
Remote Control	4	7	2
Book	6	3	0

TABLE III
RECOGNITION RESULTS FOR EACH GRASPED OBJECT WITH IMAGE EXTRACTION

Grasped Object	Success	Failure	Feature
Smartphone	16	3	0
Remote Control	11	1	0
Book	6	1	0

VI. Discussion

A. Discussion on Experimental Results for RQ1-1

From Figure 4, it can be seen that the recognition success rate meets the standard for all grasped objects. Considering that, in general, multiple recognition attempts are made while a single object is being grasped, if the recognition success rate is 50%, the probability of successfully recognizing an object at least once in four attempts exceeds 90%. Although feature responses are not included in the recognition success rate, they can be used for queries based on object names or characteristics. Therefore, since the recognition success rate meets the standard for all grasped objects and feature responses can provide supplementary information, the implemented part can be evaluated as having practical recognition accuracy.

Referring to Table I, the main cause of recognition failure for grasped objects such as keys, wallets, smartphones, and glasses cases is complete cutoff during image extraction. For these objects, when cutoff does not occur, they are almost always correctly recognized, and cases where characteristics are described are few. On the other hand, for glasses and glasses cloth, the number of misrecognitions by GPT-4o and cases where characteristics are described account for about half of the recognition executions, which is clearly higher than for other objects. This result suggests that GPT-4o may have difficulty recognizing certain objects in this study's method. Additionally, referring to Figure 8, when the grasped object is cutoff, the center of the extracted image is not the palm but a completely different position. This suggests that the main cause of cutoff is errors in the posture estimation results. Therefore, improving the accuracy of posture estimation could further increase the proportion of data saved that can be queried based on object names or characteristics.

B. Discussion on Experimental Results for RQ1-2

From Figure 9, the recognition success rate is about 90%, indicating that the implemented part can recognize hands not grasping objects with high accuracy. However, considering the purpose of GORRS, further improvement is necessary to prevent user confusion. Additionally, referring to the example in Figure 10, the cause of misrecognition when not grasping objects is likely due to the overlap of the hand with background objects in the image, making it appear as if the background object is being grasped.

C. Discussion on Experimental Results for RQ2

Comparing Tables II and III, the recognition accuracy of the remote control significantly increases. This result suggests that when using GPT-4o for recognizing grasped objects, extracting hand images can significantly improve the recognition rate for specific objects.

D. Future Challenges

The immediate challenge is to complete the implementation of A4, the unimplemented part of the proposed method, and verify how useful it is for finding misplaced objects. Additionally, although the implemented part has practical recognition accuracy for grasped objects, there is room for improvement, and further enhancement of recognition accuracy is necessary. Furthermore, since GPT-4o is not a local LLM, privacy issues will also need to be considered in the future.

VII. Conclusion

This study proposed GORRS. The background of this study is the increasing number of elderly people and dementia patients and the burden of forgetting the placement of objects on themselves, their families, and caregivers. The purpose of this study is to reduce this burden. The key idea is to use fixed-point cameras and image recognition technology to recognize and record grasped objects, making it easier to locate misplaced objects. The implementation focused on the part that recognizes and records grasped objects.

Future challenges include implementing the unimplemented part and conducting evaluation experiments to verify the usefulness of GORRS, as well as improving the recognition accuracy of grasped objects.

Acknowledgment

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

References

[1] G. o. J. Cabinet Office, "White paper on aging society 2024," 2024, https://www8.cao.go.jp/kourei/whitepaper/w-2024/zenbun/06pdf_index.html (Accessed on 2025/04/29).

[2] S. Hodges, L. Williams, E. Berry, S. Izadi, J. Srinivasan, A. Butler, G. Smyth, N. Kapur, and K. Wood, "Sensecam: A retrospective memory aid," in *Proceedings of the 8th International Conference on Ubiquitous Computing (UbiComp 2006)*, ser. Lecture Notes in Computer Science, vol. 4206. Springer Berlin, Heidelberg, 2006, pp. 177–193.

[3] N. C. for Geriatrics and Gerontology, "What is the difference between forgetfulness and dementia?" 2024, https://www.ncgg.go.jp/dementia/about/007.html (Accessed on 2025/04/29).

[4] T. L. I. Company, "What is the difference between forgetfulness due to aging and dementia?" 2022, https://www.taiyo-seimei.co.jp/net_lineup/colum/ninchi/002.html (Accessed on 2025/04/29).

[5] N. C. of Neurology and J. Psychiatry, "Dementia," https://kokoro.ncnp.go.jp/disease.php?@uid=WwE9LLpYbVZTIDMI (Accessed on 2025/04/29).

[6] T. Team, "Tensorflow.js: Machine learning for javascript developers," 2024, https://www.tensorflow.org/js?hl=en (Accessed on 2025/04/29).

[7] D. Oved, I. Alvarado, and A. Gallo, "Real-time human pose estimation in the browser with tensorflow.js," TensorFlow Medium, 2018.

[8] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," arXiv preprint, 2024, arXiv:2402.06196. [Online]. Available: https://arxiv.org/abs/2402.06196

[9] OpenAI, "Hello gpt-4o — openai," 2024, https://openai.com/index/hello-gpt-4o/?utm_source=chatgpt.com (Accessed on 2025/04/29).

[10] P. S. Foundation, "Python 3.13.1 documentation," 2025, https://docs.python.org/3.13/ (Accessed on 2025/04/29).

[11] S. Ramírez, "Fastapi," https://fastapi.tiangolo.com/ (Accessed on 2025/04/29).

[12] SQLAlchemy, "Sqlalchemy documentation — sqlalchemy 2.0 documentation," 2025, https://docs.sqlalchemy.org/en/20/ (Accessed on 2025/04/29).

[13] Oracle, "Mysql," https://www.mysql.com/ (Accessed on 2025/04/29).

[14] Logicool, "Specifications - mx brio," https://support.logi.com/hc/en/articles/18849231040023-Specifications-MX-BRIO (Accessed on 2025/04/29).