Hand–object interaction recognition based on visual attention using multiscopic cyber-physical-social system

Adnan Rachmat Anom Besari [a,b,1,*], Azhar Aulia Saputra [a,2], Wei Hong Chin [a,3], Kurnianingsih [c,4], Naoyuki Kubota [a,5]

[a] Department of Mechanical Systems Engineering, Graduate School of Systems Design, Tokyo Metropolitan University, Tokyo 191-0065, Japan
[b] Department of Information and Computer Engineering, Politeknik Elektronika Negeri Surabaya, Sukolilo, Surabaya 60111, Indonesia
[c] Department of Electrical Engineering, Politeknik Negeri Semarang, Jl.Prof. Sudarto, Tembalang, Semarang 50275, Indonesia
[1] anom@pens.ac.id; [2] aa.saputra@tmu.ac.jp; [3] weihong@tmu.ac.jp; [4] kurnianingsih@polines.ac.id; [5] kubota@tmu.ac.jp
* corresponding author

ARTICLE INFO

Article history
Received August 26, 2022
Revised November 14, 2022
Accepted April 13, 2023
Available online May 2, 2023

Keywords
Telemedicine
First-person vision
Hand-eye coordination
Independent rehabilitation
Occupational therapy

ABSTRACT

Computer vision-based cyber-physical-social systems (CPSS) are predicted to be the future of independent hand rehabilitation. However, there is a link between hand function and cognition in the elderly that this technology has not adequately supported. To investigate this issue, this paper proposes a multiscopic CPSS framework by developing hand–object interaction (HOI) based on visual attention. First, we use egocentric vision to extract features from hand posture at the microscopic level. With 94.87% testing accuracy, we use three layers of graph neural network (GNN) based on hand skeletal features to categorize 16 grasp postures. Second, we use a mesoscopic active perception ability to validate the HOI with eye tracking in the task-specific reach-to-grasp cycle. With 90.75% testing accuracy, the distance between the fingertips and the center of an object is used as input to a multi-layer gated recurrent unit based on recurrent neural network architecture. Third, we incorporate visual attention into the cognitive ability for classifying multiple objects at the macroscopic level. In two scenarios with four activities, we use GNN with three convolutional layers to categorize some objects. The outcome demonstrates that the system can successfully separate objects based on related activities. Further research and development are expected to support the CPSS application in independent rehabilitation.

1. Introduction

Rehabilitation is required for patients recovering from neurological illnesses, particularly hand stroke. However, roughly two-thirds of hand stroke survivors have visual impairments because of visual field loss, double vision, and perception issues [1]. Vision problems may lead to different adverse outcomes, including grasping objects because of their limited range of motion and vision. Individuals find it more convenient to receive rehabilitation treatment at home rather than in a hospital setting. However, visiting patients' homes is difficult during the COVID-19 outbreak. Telemedicine allows therapists to monitor rehabilitation patients via online therapy or video visits [2]. Because of limited therapeutic staff or privacy concerns, patients can not use this method indefinitely. Cyber-physical-social system (CPSS) [3] seamlessly integrates physical and social space in cyberspace. This technology can deliver valuable information which not available from typical physical sensors.

Hand movement progress in poststroke patients has been studied using cyber-physical systems (CPS) [4]. This study has been conducted to track individual hand therapy. Hand monitoring can be done in

two ways: contact and noncontact. Most studies employ the contact method, where patients wear a device in their hands. In rehabilitation research, wearable hand devices are commonly used as contact methods. It provides accurate data by utilizing a variety of sensors, including flex, accelerometers, and hall-effect sensors [5]. However, this strategy has drawbacks, such as high equipment costs and uncomfortable use.

Consequently, scientists are investigating new possibilities using noncontact techniques, such as low-cost computer vision systems. Still, suppose this method does not consider some social aspects, such as dealing with privacy issues, getting justification from a clinical background, and understanding user experiences to provide better benefits. In that case, it may fail in user acceptance. Noncontact techniques for detecting human action recognition [6] still deal with complicated processing, multiple interpretations, and privacy concerns. Hence, an egocentric vision like smart glasses is required to address these issues [7]. Egocentric vision has the advantages of reducing privacy concerns, monitoring mobile devices, and attracting attention during activities. Hand–object interaction (HOI) recognition using egocentric vision should be studied further in rehabilitation applications [8].

HOI recognition is increasingly being used in post-stroke therapy to track patients' progress [9]. This recognition study outperforms full-body human interaction detection. This vision analyzes images captured by an on-body camera, such as smart glasses or an action cam. Still, HOI recognition is more straightforward because only hands and objects are detected, especially when using egocentric vision [10]. However, when using this vision, the expansion of HOI recognition encounters several challenges, particularly in 2D images. First, researchers frequently encounter data on hand and object invalid contact detection. This issue arises because 2D image data cannot express depth data. As a result, the system could not pinpoint the precise location of the hand and objects. Numerous approaches can address the issues, such as learning with the interaction point [11]. However, these approaches are limited to recognizing a single object type, and problems may arise when detecting many objects.

Furthermore, the application of this method is limited to identifying hands and objects but not considering persons with visual impairment. Because of their visual impairment, patients struggle to make physical hand movements and apply their recognition ability. The egocentric vision that focuses on HOI recognition has the potential to monitor both physical and cognitive rehabilitation simultaneously. This vision system, for example, could use hand skeletal estimations and the kinematic finger model to assess the person's physical condition. Besides that, specific cameras are outfitted with eye tracking to gather data on the user's visual attention. Gaining user experience and developing cognitive abilities are required. Therefore, while handling objects, it's crucial to pay attention to hand movements and vision.

The study's primary contributions are: (1) We apply egocentric vision and feature extraction to observe hand posture at the microscopic level. We use three layers of graph neural network (GNN) as a feature-based classifier to differentiate 16 grasp poses when interacting with objects. (2) At the mesoscopic level, we use active perception to validate HOI recognition with eye tracking in the task-specific reach-to-grasp cycle. To identify the connection of the hand with an object, we use the distance between the fingertips and the center of an object as inputs to a multi-layer gated recurrent unit (MGRU) based on recurrent neural networks (RNN) architecture. (3) We implement a cognitive ability at the macroscopic level by incorporating visual attention. In two different scenarios, we use the object relation as the input of a GNN node classifier with three convolutional layers to separate objects based on related activities. This method's output indicates the object's relationship in activities based on personal behavior.

The structure of this article is as follows. Section 2 discusses the research in hand rehabilitation monitoring and our proposed method for improving HOI recognition based on visual attention. Section 3 discusses the findings and assesses the efficacy of the proposed framework. Finally, Section 4 discusses the research's findings and some future directions.

2. Method

Recent research trends in hand movements analysis using egocentric vision have been widely used to advance the field of rehabilitation. Some researchers are interested in introducing the patient's hand behavior, for example, by using fingertip detection when using a therapy ball [12], monitoring hands in spinal cord injury patients [13]–[15], and stroke patients [9] [16]. Other research focuses on computational problem solutions, such as the high cost of additional equipment and pixel-level observations [17] and overcoming occlusion, inference, and contact [18]. Many studies employ egocentric approaches, like GoPro wearable cameras or datasets like Deeplab-VGG16, EgoHand, EPIC-ADL, and multi-datasets [19]. Existing research focuses solely on physical hand evaluation. It is uncommon to find studies that combine physical hand and cognitive abilities [20], such as in hand-eye coordination research for predicting the "next active object" shortly [21]. Table 1 shows the research on hand rehabilitation with egocentric vision in the last 5 years.

Table 1. The research of hand rehabilitation with egocentric vision (2018–2022)

No.	Research	Application (Sensor Types/Dataset)	Methods
1.	Qurratu'aini *et al.* [12]	Fingertips gripping a therapy ball for hand recovery. (HD Logitech C615 Web Camera with 1920 × 1080)	Speeded Up Robust Features (SURF) descriptors, K-mean clustering, and Support Vector Machine (SVM).
2.	Li *et al.* [17]	Overcoming expensive equipment and pixel-level annotations. (Deeplab-VGG16 Dataset)	Un-supervised hand segmentation using a fully convolutional neural network (FCN).
3.	Likitlersuang *et al.* [13]	Monitoring spinal cord injury patients' hand usage at home. (GoPro Hero4 with 1920 × 1080/30fps)	Fast R-CNN (hand detection), Contour Selection (hand segmentation), and Functional Measure Extraction (interaction detection).
4.	Visée *et al.* [14]	Home monitoring of SCI patient's upper limb function. (GoPro Hero4 with 1920 × 1080/30fps)	YOLOv2 (hand detection), DAT (hand tracking), and Random Forest Classifiers (interaction detection).
5.	Xu *et al.* [16]	Developing a low-cost technology to monitor stroke patient hand motions and gestures. (EgoHand datasets)	CNN-based hand motion and gesture detection.
6.	Tsai *et al.* [9]	Evaluating hand functions after a stroke. (GoPro Hero 5 with 1280 × 720/30fps)	YOLOv2 (hand detection), UNET (hand tracking); Random Forest Classifiers (interaction detection).
7.	Jiang *et al.* [21]	Visual attention and hand posture to predict the next active object. (EPIC and ADL Dataset)	The deep neural network model combines visual and hand cues.
8.	Bandini *et al.* [16]	Assessing hand usage at home after a cervical spinal cord injury. (GoPro Hero 5 Black (1280 × 720/30fps)	Deep learning model (hand localization), HOID-Net (interaction detection), statistical analysis.
9	Lee *et al.* [18]	Detecting hand motion tracking with robustness against occlusion, interference, and contact. (Stereo camera, intelligent gloves, and IMU)	Visual-Inertial Skeleton Tracking (VIST)
10.	**Our proposed method**	**Develop HOI based on visual attention using multiscopic CPSS for physical-cognitive rehabilitation support. (Tobii Pro 3 Eye tracker 1920 × 1080/30fps)**	**GNN graph classifier (grasp pose classification), MGRU (multivariate time-series for HOI classification), and GNN node classifier (object classification).**

Previous work in our laboratory focused on nonverbal communication for socially integrated robot companions using directed learning [22]. This study examines person's intents and abilities when reaching for and gripping objects. However, determining individual's preferences for using the robot companion as a third person is difficult [23]. Occlusion limits the third-person perspective, which depends on many different viewpoints. From an egocentric standpoint, it is critical to support the current system. With a case study on the Chopsticks Manipulation Test, we examined the significance of combining finger joint angle estimation and a visual attention measurement in hand rehabilitation [24]. Our previous work used a multiscopic method to address dynamic locomotion in a legged robot [25]

and simulation for human-robot interactions [26]. We propose a multiscopic approach for developing a CPSS for HOI recognition based on visual attention based on this experience. Fig. 1 depicts the framework of HOI recognition based on visual attention using multiscopic CPSS. The subsections explain GNN, microscopic, mesoscopic, and macroscopic levels. Each subsection outlines methodology, algorithm, and development in detail.

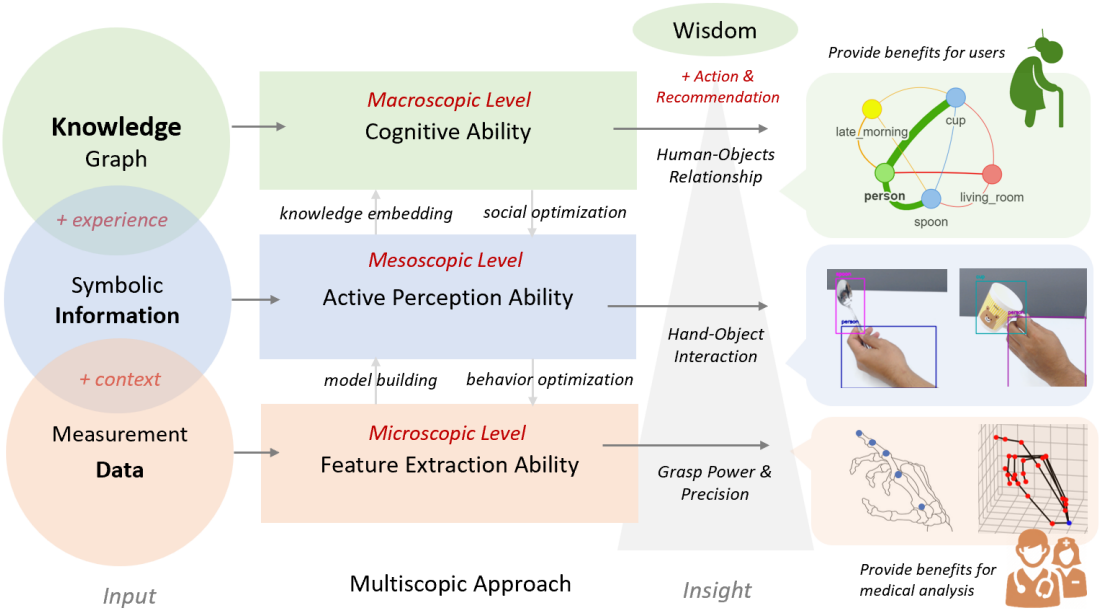


Fig. 1. The framework of HOI recognition is based on visual attention using multiscopic CPSS

Promoting independent hand rehabilitation from the ground up is critical, from the physical to the cognitive. This study presents the CPSS framework, which employs the multiscopic method to address the following technical issues: (a) Classify hand data at the microscopic level utilizing feature extraction abilities; data from vision sensors will be the input for this level. Using symbolic information, we develop a feature extraction capability to estimate hand and finger posture. (b) Using active perception ability, extract HOI at the mesoscopic level. At the mesoscopic level, this information will be used as input. As a knowledge graph, we will build active perceptions' ability to categorize HOI. (c) Discover the object relationship via macroscopic cognition ability. At this level, the graph data will be used as input. We hone our cognitive abilities to demonstrate recommendations based on human behavior.

2.1. Graph Neural Networks

GNN is neural network architecture used to learn the representations of graphs data and has become a popular learning model for prediction tasks on nodes, graphs, and links. The basic idea of GNN is to learn suitable graph data representation for neural networks [27]. Before we talk about GNN, we should look at the basic mathematics of graph structure data in computer science. A graph G can be a part of set atributed graphs $\mathcal{G}$. GNN use all graph information as an input, including the node features and the connections stored in the adjacency matrix. A graph G is defined by the following equation:

$$G = (V, E, X),\ G \in \mathcal{G} \tag{1}$$

Where $V = \{v_1, \ldots, v_n\}$ is a set of nodes and $E = \{e_{a,b}, \ldots, e_{i,j}\}$ is a set of ordered couples representing the connection between two nodes belonging to V. Each node comes with $X = \{x_v\}$ as a set of node attributes where $v \in V$. GNN output new representation called embeddings for each node. These node embeddings contain the structural and feature information of other nodes in the graph. The embeddings can finally be used to perform predictions. We embed each node through several rounds of message passing. This paradigm can be broken down into (a) initialization, (b) aggregation, and (c) update. We initialize each node v at layer $k = 0$ as the first round of message passing with the following equation:

$$h_{G,v}^{(0)} = x_v, \ v \in V \tag{2}$$

Suppose $h_{G,v}^{(k)}$ represents the node embeddings for some vertex v at layer k-th, where node feature x_v of all nodes $v \in V$ in graph G. Second, we do the aggregation function for each node v with the following equations:

$$m_{G,v}^{(k)} = f_{Agg}^{(k)}\left(h_{G,u}^{(k-1)}\right), \ 1 \le k \le K. \tag{3}$$

$$= \frac{1}{|N(v)|}\sum_{u \in N(v)} W_{i,j} h_{G,u}^{(k-1)}, \ i \ne j, \ 1 \le i,j \le |V|. \tag{4}$$

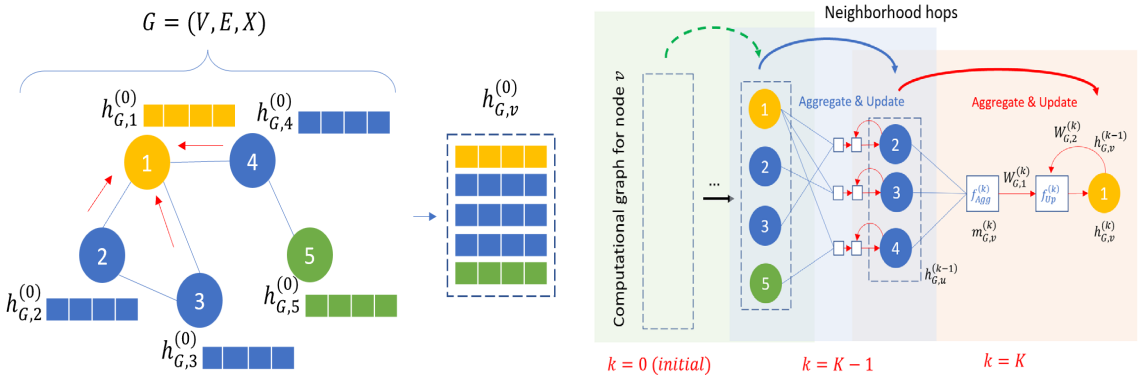
We utilize the next step of the neural message passing scheme where we localized node v from their neighbors $N(v)$. The node feature $h_u^{(k-1)}$ of all nodes $u \in N(v)$ in graph G are iteratively aggregating and stored in $m_{G,v}^{(l)}$ as aggregation function $f_{Aggregate}^{(k)}$. The $N(v) \subset V$ denotes the neighborhood of $v \in V$. The aggregation function performs a significant role and is shared by all nodes within an iteration. An average, degree-normalized sum or coordinate-wise min or max may also replace the sum.

Third, we employ the last step of the neural message passing scheme where the node feature $h_{G,v}^{(k-1)}$ of all nodes $v \in V$ in graph G are iteratively updated by the aggregating result from their neighbors $N(v)$ with the following equations:

$$h_{G,v}^{(k)} = f_{Up}^{(k)}\left(h_{G,v}^{(k-1)}, m_{G,v}^{(k)}\right). \tag{5}$$

$$= \sigma\left(W_{i,i} h_{G,v}^{(k-1)} + m_{G,v}^{(k)}\right), \ 1 \le i \le |V|\,. \tag{6}$$

The update function $f_{Up}^{(k)}$ is usually a weighted combination with learnable weight matrices. The update functions are implemented as a fully connected layer that alternates linear transformations and coordinate-wise nonlinear activations σ such as *ReLU*, *tanh,* or *sigmoid*. Fig 2 shows the graph representation including the graph with node feature and message passing mechanism.



(a) graph with node feature (b) message passing mechanism

Fig. 2. Graph representation

The final node representation $h_{G,v}^{(K)}$ is the last layer, possibly concatenated with a linear classifier. If we want to make a graph-level prediction, all node embeddings can be aggregated into a unified graph embedding $H_G^{(K)}$ with f_{Read}. The most popular method is to take the average of node embeddings. We add up all the node features of all nodes $h_{G,v}^{(K)}$ in the K-th layer and then dividing by the number of nodes, as the following equation:

$$H_G = f_{Read}\left(h_{G,v}^{(K)}\right). \tag{7}$$

$$= \frac{1}{|V|}\sum_{v \in V} h_{G,v}^{(K)}. \quad (8)$$

In the end, we use a linear transformation based on a fully connected layer with W_{proj} as weight projection and *arg max* function to determine which class corresponds to the graph input with the highest probability, as the following equation:

$$y_i = H_G W_{proj} \quad (9)$$

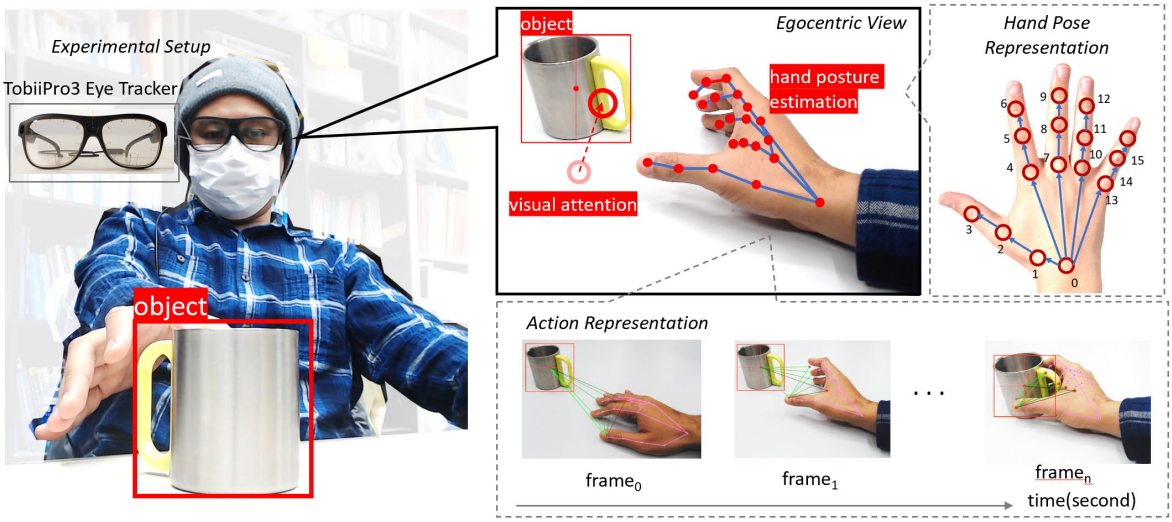
$$\hat{y} = arg \max_i y_i \quad (10)$$

We utilize Pytorch Geometric (PyG) as the GNN development framework. Our design uses high-level graph computation with PyG to teach scene interpretation. Because our graph classification datasets are small, we do a mini-batch for the graphs before inputting them into a GNN to guarantee full GPU utilization. PyG automatically takes care of batching multiple graphs into a single giant graph.

2.2. Microscopic Level: Feature Extraction Ability

The experimental setup and hand pose classification comprise the feature extraction ability at the microscopic level. We use egocentric vision with smart glasses facing the table and the objects. A participant facing down directly at the object wore Tobii Pro Glasses 3 smart glasses [28]. The smart glasses have a 1920 × 1080-pixel resolution and a frame rate of 25 frames per second. To capture hands and objects, this camera is used in 16:9 scene camera format with a wide field of view of 106° with 95° horizontal and 63° vertical. Visual attention or awareness is predicted to be appropriately detected when the interaction between the hand and the object falls within the field of view range.

We used the YOLOv5 [24] model to extract the object's location from picture frames represented by the bounding box and labels. To improve this detection, the Simple Online Real-time Tracking (SORT) [29] technique was used. This framework excels at representation learning and applying it to object recognition and tracking applications. We employ MediaPipe [30] hand tracking to obtain estimated hand posture data. A construction designed specifically for complex perceptual channels that use rapid real-time inference. We only utilize hand posture prediction as supporting data to validate HOI recognition in our approach. Object detection and hand estimation provide us with two pieces of information. First, we can look for an object in a specific image, and then we can pinpoint the precise location of the hand and its feature in the two-dimensional image. Fig. 3 depicts the experiment's design, which includes the experimental setup of the systems and the implementation of an egocentric view of HOI recognition with visual attention.



(a) Experimental setup (b) Egocentric vision with attention

Fig. 3. Design of the experiment

Many hand grasp variants and orientations are collected. We standardize the angular data to eliminate outliers. Then, we perform three positions in the vector-to-joint-angle conversion to obtain each finger feature. This transformation is accomplished by converting these three-dimensional coordinate points into an angle. Below is the equation used to calculate an angle from two vectors in three-dimensional coordinates:

$$\theta = arccos\left(\frac{\overrightarrow{AB}\cdot\overrightarrow{BC}}{\|\overrightarrow{AB}\|\|\overrightarrow{BC}\|}\right) \tag{11}$$

We can compute $\overrightarrow{AB}$ and $\overrightarrow{BC}$ if we have the coordinates for three points (A, B, and C). The angle obtained by $A \rightarrow B \rightarrow C$ employing the right-hand rule from B continues using dot products, whereas $\|\overrightarrow{AB}\|$ determines the length of $\overrightarrow{AB}$, $\|\overrightarrow{BC}\|$ determines the size of $\overrightarrow{BC}$, and θ (theta) is the angle formed by two vectors. And then, we can get the dot product $\overrightarrow{AB}\cdot\overrightarrow{BC}$ as well as the lengths $\|\overrightarrow{AB}\|$ and $\|\overrightarrow{BC}\|$. After all, by replacing the equation, we rearrange the formula for determining θ. These joint angle values are used for features in the data graph. We represent the relationship between joints in a directed graph (digraph).

We use this GNN to classify 16 manipulation grasping poses. We acquire 130 data for every grasp pose in various orientations. Thus, we have 2080 graphs with divisions of 1600 for training and 480 for testing. Our data graph consists of 16 nodes consisting of all nodes connected by 15 edges. We utilize a graph input layer consisting of 15 joint angle nodes and one wrist node as the center of the finger connection. The layer of output consists of 16 nodes representing grasping posture. We train a final classifier on graph embedding. Before applying the final classifier on top of the graph, we employ Rectifier Linear Unit (ReLU) activation function to create localized node embeddings. We use three layers of GNN with the same training cycle: construct an optimizer, feed the model inputs, compute the loss, and optimize using autograd. A linear transformation layer and *argmax* function are used to classify and determine which class of grasp posture corresponds to the input with highest probability. We discuss training and testing outcomes in the results and discussion section. Fig. 4 shows the microscopic level design using a three-layer GNN for hand pose classification.

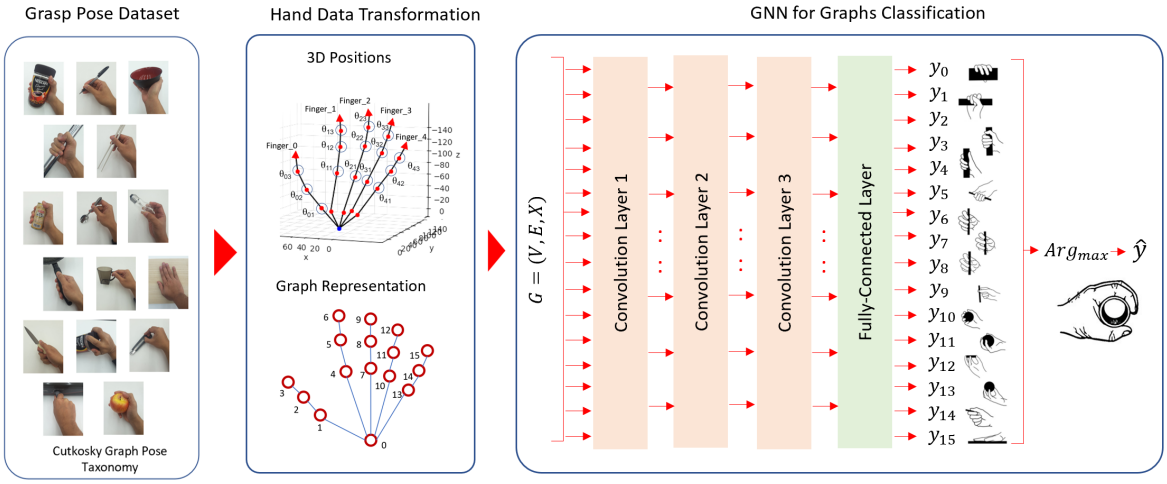


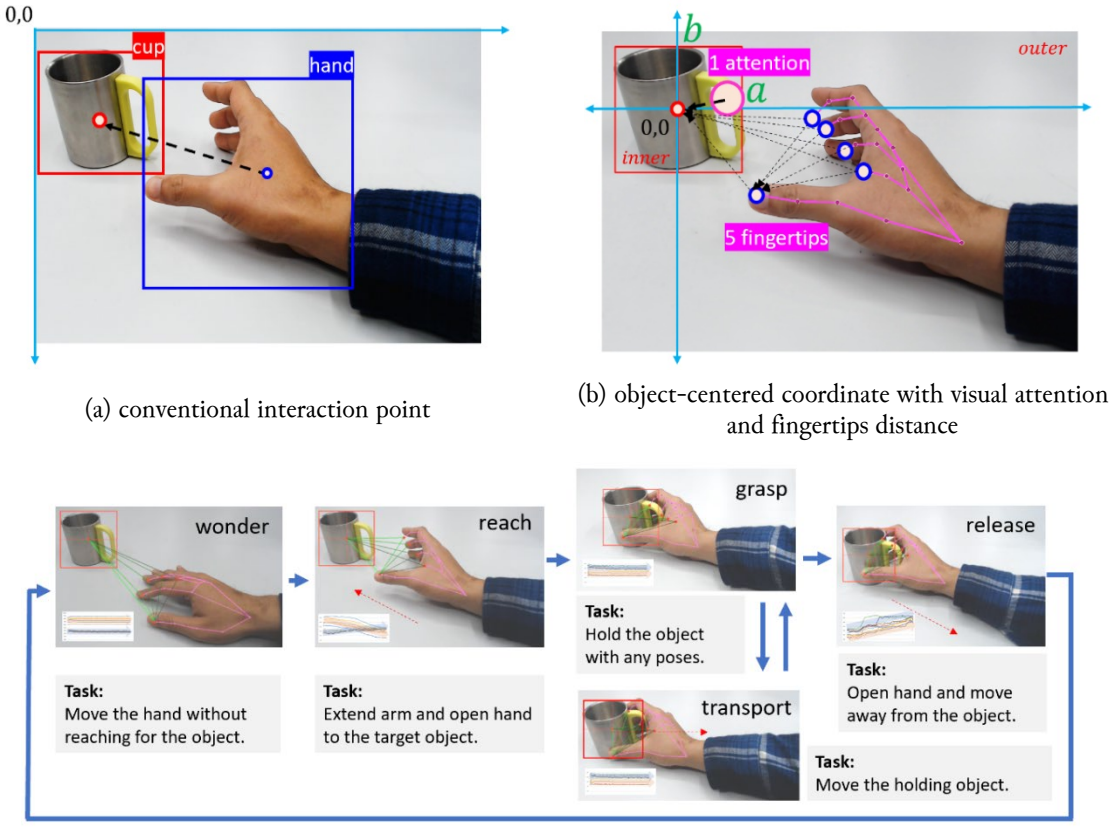
Fig. 4. The microscopic level design uses a three-layer GNN for hand pose classification

2.3. Mesoscopic Level: Active Perception Ability

This subsection describes the phases of active perception ability development at the mesoscopic level. These phases are object-centered coordinate transformation and validation of HOI recognition using the task-specific reach-to-grasp cycle. We accomplished a coordinate transformation centered on the object to simplify the validation procedure and get fewer data [31]. The goal of centering the item is to relocate the image coordinates' initials (0,0) to the middle of the object. We obtain a new center and identify the position of the new coordinates for every new frame. This method uses for any pixel (x_n, y_n) in the image plane. The joint finger position in the new coordinate plane contains

the object's length (a_0) and width (b_0). The inner and outer borders of the object are acquired using this additional bounding box location information. Then, using the Pythagorean equation, we can estimate the distance between a point (a_n, b_n) to the center of the object coordinates (0,0). The distance d_n must be determined between the fingertip or finger joint and the object-centered coordinate. To get these properties, we then utilize a_n, b_n, and d_n to validate the HOI transformation.

Validation of HOI recognition is restricted to the reference grasp of the approved hand usage section in ICF for hand rehabilitation, notably in the case of the reach-to-grasp cycle [32]. The procedure has four distinct tasks, each of which is defined separately. The "wonder" task, which indicates that the user moves the hand without reaching for the object, is the first task displayed as the starting status. The second task is the "reach" task, which requires the subject to extend his arm and open his hand to the object. The third task is the "grasp," in which the participant holds the object in any position. The task switches to a new state called transport when the user moves the holding object. The fourth task is the "released" task, which occurs when the individual pulls their open palm away from the item. Fig. 5 compares the traditional method to object-centered coordinates with visual attention in HOI recognition.



(a) conventional interaction point

(b) object-centered coordinate with visual attention and fingertips distance

(c) phases of the task-specific reach-to-grasp cycle

Fig. 5. The comparison method

In our initial stage, we utilize a single object as a reference. We picked a medium-sized cup with various hand postures. We use ten features: five elements of each fingertip's distance to the center of the object $(d_o, d_1, d_2, d_3, d_4)$, four elements of each fingertip's distance to the thumb fingertip (e_1, e_2, e_3, e_4), and one visual attention (f_0). We obtain 50 frames per sample using our computer specs, which becomes our benchmark for estimating the length of a data stream. We analyze 10 data points in each picture capture series to obtain a real-time result. In our neural networks, we use this data as input for the learning system.

We use an RNN architecture to classify multivariate time series using an MGRU [32]. The total number of layers, input size, hidden size, and the number of recurrent layers are some variables that can

be changed in this design. To compute each element in every layer of an MGRU, we calculate as the following functions:

$$r_t = \sigma(W_{i,r}x_t + b_{i,r} + W_{h,r}h_{(t-1)} + b_{h,r}), \tag{12}$$

$$z_t = \sigma(W_{i,z}x_t + b_{i,z} + W_{h,z}h_{(t-1)} + b_{h,z}), \tag{13}$$

$$n_t = \sigma_h(W_{i,n}x_t + b_{i,n} + r_t * (W_{h,n}h_{t-1} + b_{h,n})), \tag{14}$$

$$h_t = (1 - z_t) * n_t + z_t * h_{t-1}. \tag{15}$$

The time is symbolized by t. The hidden states are represented by h_t; the inputs are characterized by x_t, the hidden states of the layers at $t-1$ is expressed by h_{t-1} or the early hidden states at the initial time o, and r_t, z_t, n_t are the resets, updates, and new gates. The sigmoid function is represented by and $*$ symbolized by the product. In the MGRU, the input $x_t^{(l)}$ of the l-th layer $(l >= 2)$ is the hidden states $h_t^{(l-1)}$ of the previous layers multiplied by dropout, $\delta_t^{(l-1)}$ where each $\delta_t^{(l-1)}$ is a Bernoulli random variable 0 with a probability of dropout. Fig. 6 illustrates the mesoscopic level design for multivariate time-series classification using MGRU-RNN.

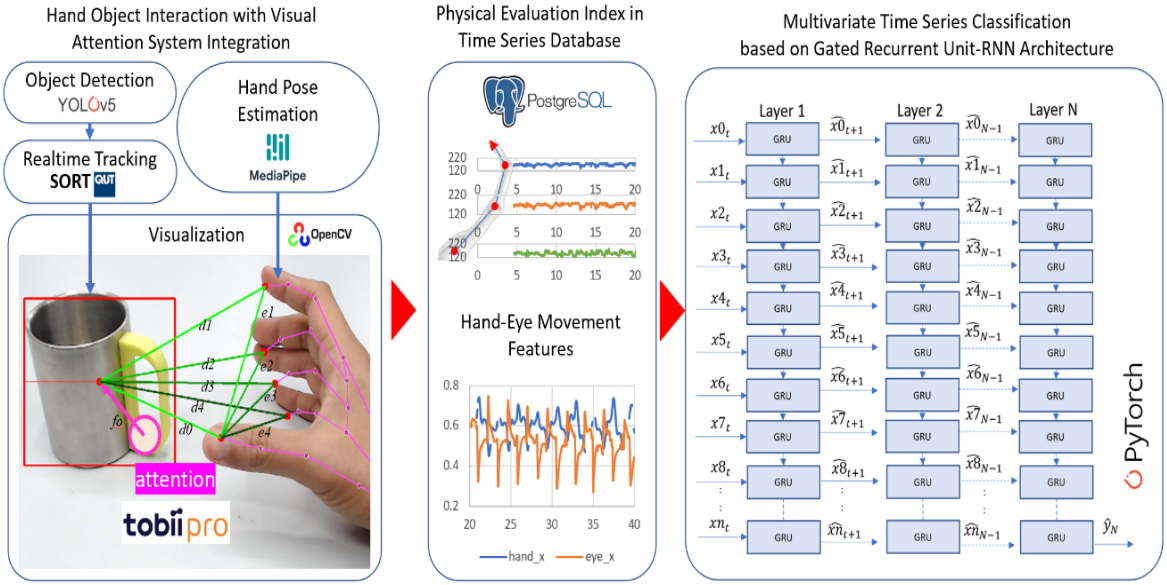


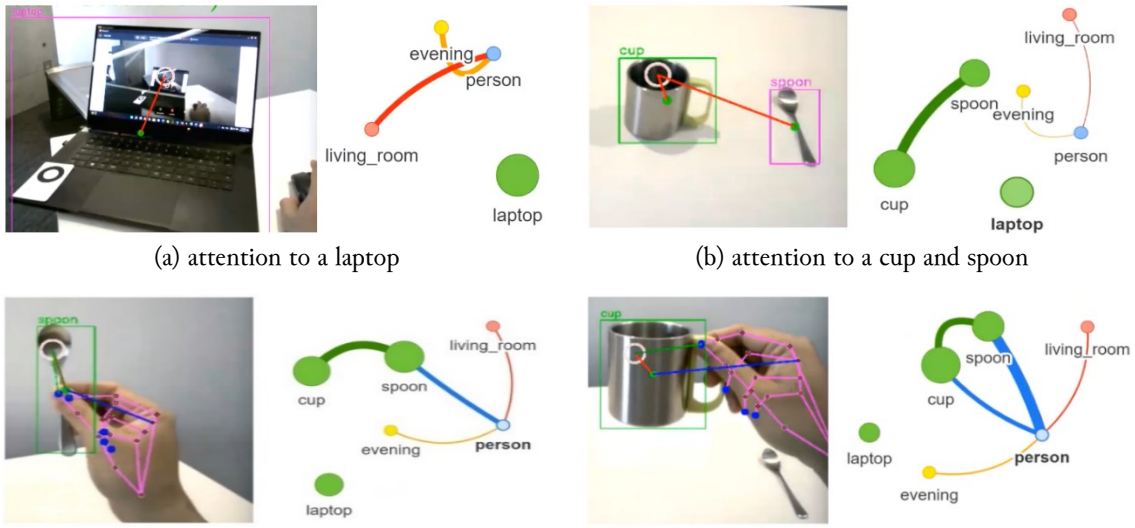
Fig. 6. The mesoscopic level design for multivariate time-series classification using MGRU-RNN

After building the MGRU-based RNN architecture, the next step is to generate a dataset to evaluate each action on HOI recognition. For each sequence, we collect 1–2 seconds of video with a minimum of 50 frames per sample. We compiled a collection of 100 videos of hands interacting with various objects. We shot the video using the following division: 25 data for the wonder, 25 data for the reach, 25 data for the grabbing job involving transportation, and 25 data for the release. We randomly divided the training and validation data into an 80:20 ratio. We decided this distinction was appropriate because the data we obtained was subjective. This experiment includes a responder. The system utilizes 80 movies and 20 videos for instruction. The training and testing outcomes are then discussed in the results and discussion section.

2.4. Macroscopic Level: Cognitive Ability

This subsection investigates the active perception ability development at the mesoscopic level. Using vision-based data collection, we create datasets for an object detection algorithm. The goal is to compile a list of captured objects at a given time. The Tobii Pro eyewear eye tracker sensor analyzes how the human eye responds to different environmental stimuli. Two scenarios are used to demonstrate human interaction with items found in everyday life. The scenario includes table activities, such as eating and

working. This experiment utilizes eleven different objects. In each scenario, we choose a participant to perform these exercises in sequence for approximately 5 minutes. Every time, the camera records these interactions and stores them in the Neo4j graph database. We create a system that generates a network of nodes and their interactions. The system computes each item's frequency of occurrence and relationships with others. The system then deduces the relationship between the objects and adds them to the graph's edge components. Fig. 7 depicts the lifelog graph dataset generation as the input of the macroscopic level.



(a) attention to a laptop (b) attention to a cup and spoon

(c) interaction with a spoon (d) interaction with a cup

Fig. 7. Lifelog graph dataset generation at macroscopic level

The collected data will be converted into graph data structures. This data representation is increasingly being used to detect connections between nodes. When two objects are connected, the edges show how they relate. Symbolic logic is used to develop a graph. This paradigm's information can be processed by computer programs and stored in graph database structures. As in previous works, we create a graph structure from the collected data to demonstrate the relationship [32]. The spring model is used to create a graph structure from this data. The features can then be assigned to network nodes and edges. GNN recently broadened the scope of datasets using graph-based topologies. We use Kipf and Welling's approach to the GNN convolutional framework to perform node classification [33]. The convolution layer is implemented by passing in the node feature representation and the graph connectivity representation. The macroscopic level design for object classification using the GNN node classifier with three convolutional layers is shown in Fig. 8.

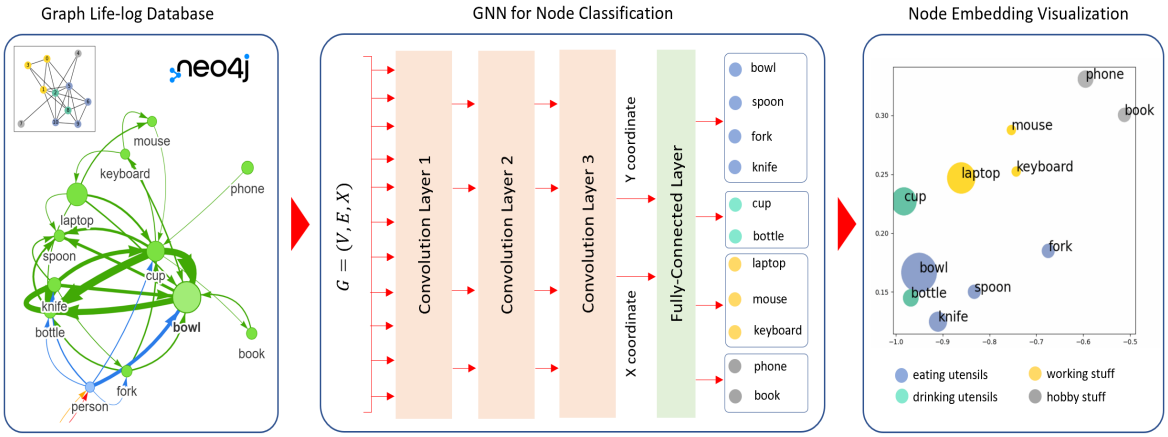


Fig. 8. The macroscopic level design for object classification using the GNN node classifier with three convolutional layers

We initialize the building blocks and define the flow of our network as a forward function. We present three convolution layers that aggregate 3-hop neighborhood information around each node. This layer reduces the node feature dimensionality from the number of the node to the number of the class (i. e. ,$11 \rightarrow 4 \rightarrow 4 \rightarrow 2$). *Tanh* nonlinearity is used to improve each convolutional layer. After that, we apply a single linear transformation as a classifier to map our nodes to the classes. We return both the final classifier output and the final node embeddings produced by the GNN convolution layer. The node embeddings are then processed by passing the initial node feature X and graph connectivity information V to the model and visualizing with two-dimensional embeddings. It generates an embedding of nodes that closely resembles the structure of the graph before training our model weights. Nodes of the same color are already closely clustered in the embedding space. Before training, the weights of our model are completely randomized. This indicates the conclusion that GNNs introduce a solid inductive bias, leading to similar embeddings for nodes close to each other in the input graph.

We train our network parameters using information about the activities of each node in the graph. We train the model by adding some objects with labels. Because our model is differentiable and parameterized, we observe how the embeddings react. We define a semi-supervised or transductive learning procedure by training against one node per class but are allowed to use the complete input graph data. We use a loss criterion to define our network architecture and start a gradient optimizer. Each round consists of a forward and backward pass to compute the gradient parameters of the loss derived from the forward pass. We compute node embeddings for all of our nodes, but only the training nodes are used to calculate the loss. This is implemented by filtering the output of the classifier out and ground-truth labels data to only contain the nodes in the training mask. The GNN model's three convolutional layers successfully separate the objects and classify the majority of the nodes.

This experiment focused on object classification in our everyday lives. The scenario creates a graph of ten daily items and their relationships in two scenarios of four activities. Next, the object is divided into two groups. In this semi-supervised learning scenario, only a person and single objects are labeled. Next, we discuss the training and testing outcomes in the results and discussion section.

3. Results and Discussion

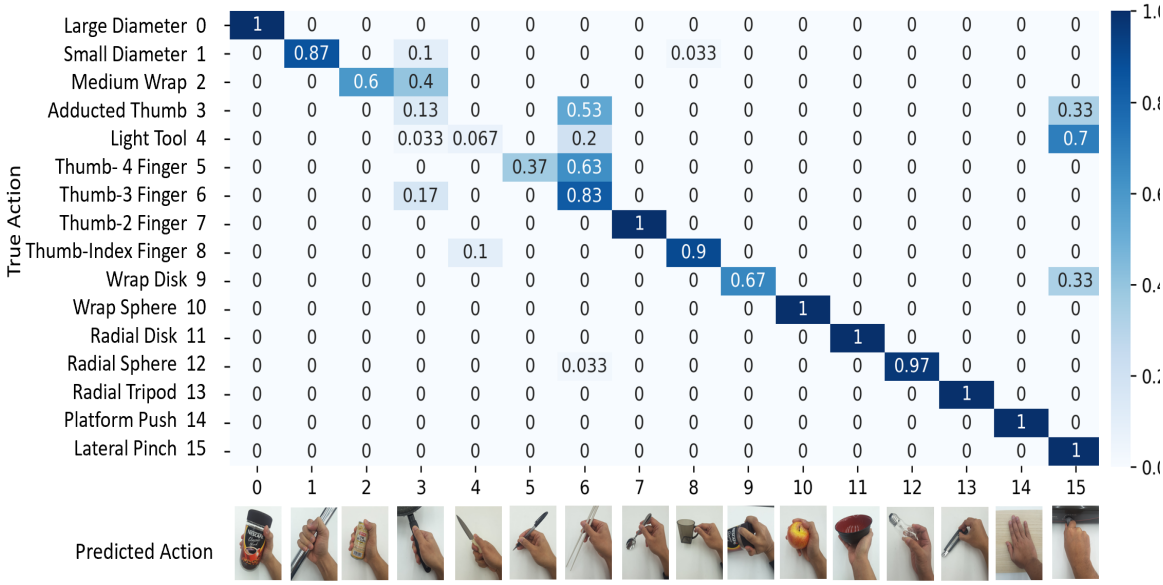
This part describes the results obtained at the microscopic, mesoscopic, and macroscopic levels. Then, we discuss the key points that must be emphasized in the current multiscopic system development for future improvement.

3.1. Discussion on Microscopic Level

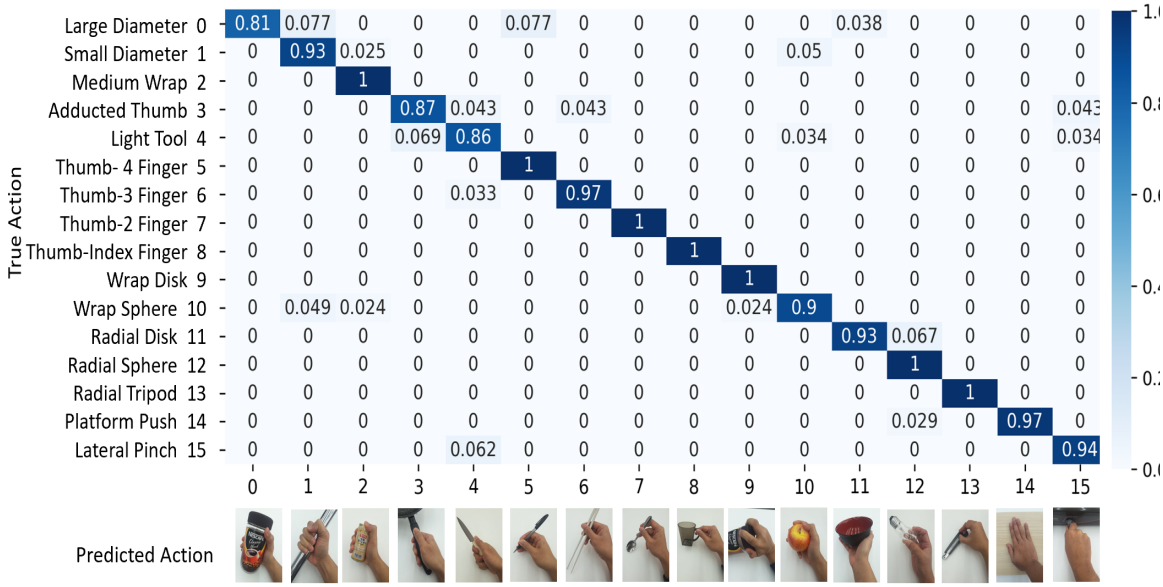
We have developed a microscopic level by designing feature extraction ability using an egocentric vision to observe hand and finger posture. The GNN learning results for classifying 16 grasp poses in a directed graph structure were reported. During training, the grasp acquisition collects input from the system. The loss was less than 0.1 after 1000 epochs. The experiment result demonstrated that the GNN for supervised classification is considered enough to be discussed. The performance of the classification is then validated during testing. The approach integrates the testing dataset into the model. GNN graph classifiers are evaluated by comparing the accuracy of our model's predictions to traditional models such as multi-layer perceptron (MLP). The confusion matrix of grasp poses classification at the 1000^{th} epoch of GNN compared to MLP is shown in Fig. 9.

All training for the GNN and MLP architectures has been completed for categorization. We used the learning outcomes model in the testing dataset to detect grasp posture. The testing accuracy for the 16 grasps pose using the GNN model is 94.87%. This result outperformed the accuracy of the MLP network, which scored 78.75% in our previous work [34]. This discovery demonstrates that the gathered dataset of grasp postures contains essential characteristics. We could see that MLP is considered to fail to recognize three classes properly, namely, in classes 3, 4, and 5 (adducted thumb, light tool, thumb-4 finger). The results show that adding a three-layer GNN to the MLP can improve it. This result indicates that GNN has a pretty good accuracy value, between 0.8 to 1, to classify 16 grasp poses. However, there is no way to know how well the model has trained each hand grasp posture. We must test the model on

diverse datasets to estimate its success rate. Hence, we must first decide whether we need power or precision to compare each grasp. These assignments have a variety of similar grab poses. For future developments, the deep learning application in grasp analysis requires hyperparameter optimization. Detailing specific grab postures might boost their accuracy.



(a) MLP testing accuracy (77.75%)



(b) GNN testing accuracy (94.87%)

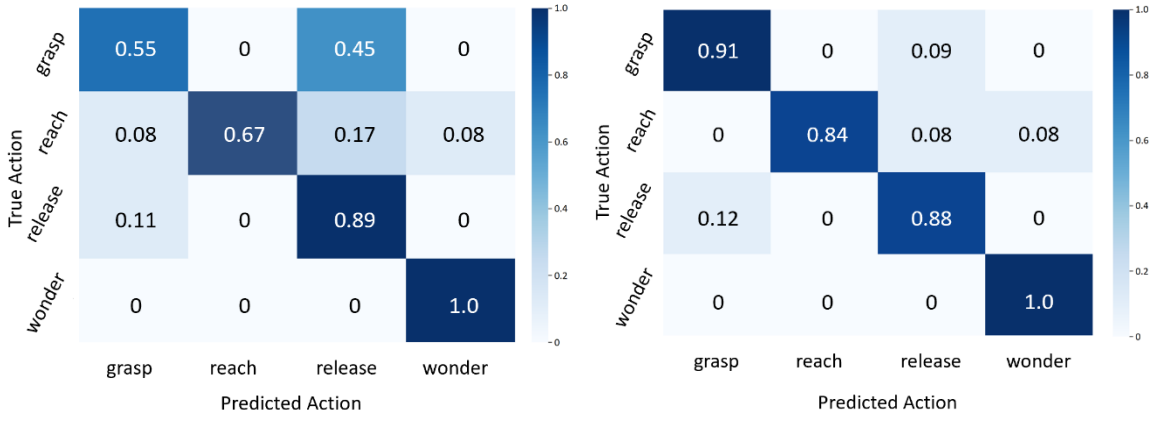
Fig. 9. The confusion matrix of grasp poses classification at the 1000th epoch

We summarize the basics microscopic level. It is reported in this section that the grasp pose categorization uses angle features. The egocentric vision with a single camera and eye tracker sensor module produced a homogeneous hand skeleton model in the form of a directed graph structure. The grasping posture was observed while the hand interacted with the item that generated the data. The data was collected and covered into angular attributes. The suggested approach was then tested using real-world grab position datasets. The categorization of grasp poses has been tested in a real-time application. Personal datasets, particularly from rehabilitation patients, are required to recognize grasp posture with multiple options for practical applications. In the future, we will employ the suggested technology to acknowledge a person's behavior when grasping the object in rehabilitation.

3.2. Discussion on Mesoscopic Level

Several experiments were carried out to test the proposed frameworks at the mesoscopic level. First, we conducted a single-participant task-specific reach-to-grasp cycle experiment. We extended the egocentric vision feature extraction capabilities discovered in our earlier research [34]. To make it easier to extract these characteristics, we used object-centered coordinate transformation to make it simple to extract these characteristics. Nonetheless, significant technical issues with the extraction occurred throughout the system's deployment. The first issue we discovered was that MediaPipe's hand position evaluation predicted only one frame. Object identification using YOLOv5 in conjunction with object tracking produces less accuracy in certain gripping poses because it does not use previous data. Hand and finger tracking with estimation filters, such as those found in the Leap Motion Controller [35], could be used to address this issue. Another issue is that the collected data is in 2D pixel units, whereas the egocentric approach is in three dimensions. Despite its limitations, the RGB camera may provide consistent results if the captured range is as far as the hand can reach, eliminating the need for precise data, such as millimeters. As a result, additional research may be conducted to improve the spread of this low-cost application.

Second, we evaluate the testing results regarding active perception ability in the task-specific reach-to-grasp action. We trained three RNN models five times: vanilla-RNN, MGRU, and long short-term memory (LSTM). With an average training time of 13.03 seconds (20.48 seconds), the MGRU is expected to outperform the RNN and LSTM in the 50th epoch. All RNN-based learning systems are well classified. The system's accuracy yields the best recognition results, with MGRU averaging 97.0% and 94.0% for LSTM. In this experiment, the MGRU outperformed the standard RNN. MGRU uses fewer tensor operations and takes less training time than LSTM. The results of the three RNN-based models, however, are nearly identical. In real-time testing, the average accuracy of MGRU is 90.75% and LSTM is 77.75%. The confusion matrix of HOI recognition at the 50th epoch of MGRU compared to LSTM is shown in Fig. 10.



(a) LSTM (b) MGRU

Fig. 10. The confusion matrix of HOI recognition at the 50th epoch

We investigate active perception capability by using an MGRU-based RNN to solve a multivariate time-series classification problem. By benchmarking multivariate time-series classification studies, this MGRU solves the vanishing and rising gradient problem of traditional RNNs [36]. MGRU is rated higher than vanilla-RNN and LSTM. Several studies show that for a simple model, the MGRU model integrates quickly and improves time-series identification performance. Compared with traditional algorithms, like the RNN and the LSTM, this learning technique improves accuracy. Even though we only use a few features for training, we achieve adequate accuracy.

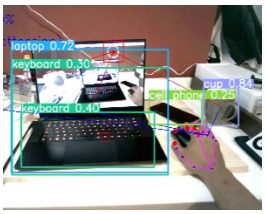
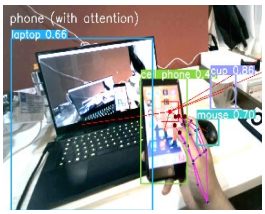
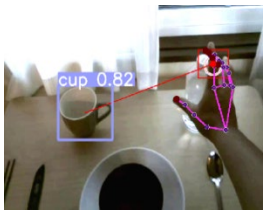
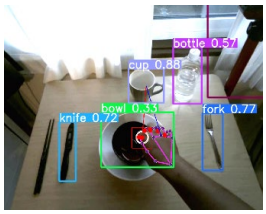
The overall mesoscopic level is summarized. For the task-specific reach-to-grasp cycle, we investigated visual attention to obtain information about hand action from objects. In this study, a new concept for independent rehabilitation in a patient with grasping and vision problems was developed. To

investigate HOI in real-world activities, we developed an egocentric viewpoint. Using active perception and hand skeleton model estimation, we successfully created object grasp detection. Then, we used RNN in conjunction with an MGRU-based architecture to classify essential hand activities. We analyzed our approach using a new dataset for object-grasping behavior. Our research has shown that our proposed method accurately verifies HOI recognition. In the future, we will work with the recommendation system [37] to solve nonstandard grasp pose and object affordance problems. We hope that this study will be used as the foundation of the macroscopic level for diverse and multi-object hand rehabilitation.

3.3. Discussion on Macroscopic Level

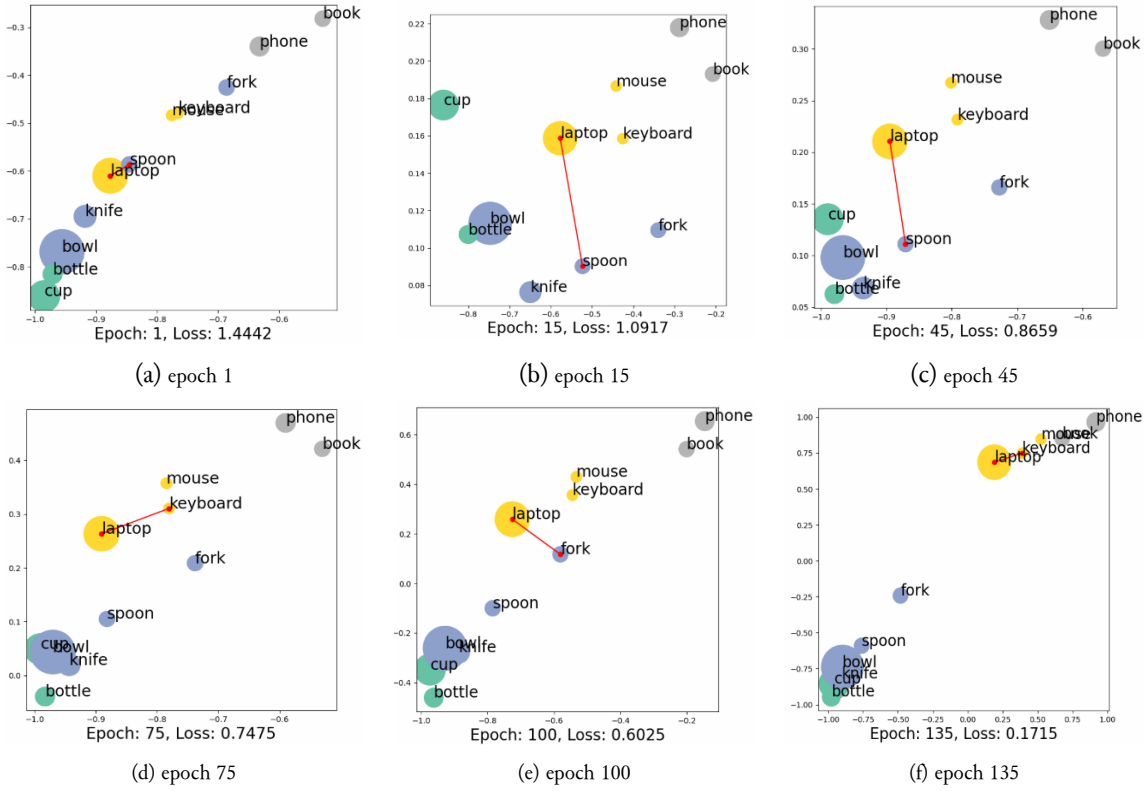
We present the macroscopic level learning stage findings in this subsection. We discovered that a GNN algorithm is intended to learn the graph data. Because the network has a small number of nodes and edges, we perform graph learning on each object in several daily activities and compare it with other objects [38]. Table 2 depicts data collection in four activities: (a) working using a laptop, (b) playing with a phone, (c) pouring water from a bottle into a cup, and (d) eating from a bowl. After a few epochs, the GNN learning system could perform semi-supervised classification in four cases. The graphic representation is used to calculate the distance between objects.

Table 2. Data collection in two scenarios with four different activities

Scenario A		**Scenario B**	
Activity 1	***Activity 2***	***Activity 3***	***Activity 4***
			
Working with a laptop	Playing with a phone	Pouring water into a cup	Taking food from a bowl

These findings imply that the relationship between objects in each scenario influences their class position. The experimental result for classifying some objects using the GNN node classifier is shown in Fig. 11. This diagram shows how the initial epoch can generate node embeddings similar to the graph structure. In the embedding space, nodes of the same color are close together, though some objects still overlap with other classes. The red line represents the laptop's relationship to the closest related object. The three-layer GNN model separated the communities linearly and correctly classified the nodes. Objects from Activities 1 and 2 appear close together as objects in Activities 3 and 4. As a result, after the 100th epoch, it is clear that GNN can separate two clusters that are far apart. This occurs because working on laptops and playing with phones occur in the same work environment as eating and drinking at the dining table.

The entire macroscopic level is summarized. We addressed graph learning research for object classification in the application. Based on the accumulated graph, we can observe how the system evolves. All interactions can be described as knowledge domains at the macroscopic level. A GNN application with weights on each attribute is required for input graphs with various contexts [39]. This method's goals are for semi-supervised categorization in daily activities related to objects. This system requires more dispersed personal datasets for real-world applications. This object classification method should be tested in several daily activities in real time. These technologies will be developed as a result of this work for future cognitive rehabilitation. Environmental constraints must be considered to tailor the rehabilitation system to specific issues. We hope that this study will help therapists and researchers by providing information unavailable in the clinic. We hope to collect patient samples for further validation and use this technology for rehabilitation in the future.



(a) epoch 1 (b) epoch 15 (c) epoch 45

(d) epoch 75 (e) epoch 100 (f) epoch 135

Fig. 11. The experimental result for classifying some objects using the GNN node classifier

4. Conclusion

This paper proposed HOI recognition based on visual attention using multiscopic CPSS. Feature extraction capacity utilizing an egocentric vision has been designed to observe hand and finger posture at the microscopic level. The GNN successfully enhanced the MLP in classifying hand grasp pose with 94.87% average accuracy. At the mesoscopic level, an active perception ability has been proposed to validate HOI recognition with eye tracking in the task-specific reach-to-grasp cycle. Objects with hand skeletal tracking were combined as inputs to MGRU, which is based on RNN architecture and has 90.75% average accuracy in categorizing hand interactions with objects. At the macroscopic level, cognitive ability has been implemented by adding visual attention to describe human behavior when interacting with multiple objects. GNN node classifiers can differentiate between two scenarios with four main activities. The outcome demonstrates that the system can successfully separate some objects based on related activities. Further research is expected to benefit independent rehabilitation support and boost community self-efficacy.

Acknowledgment

The authors would like to acknowledge the scholarship support provided by the Japan Ministry of Education, Culture, Sports, Science, and Technology (MEXT). This work was partially supported by Japan Science and Technology Agency (JST), Moonshot R&D, with grant number JPMJMS2034, and Tokyo Metropolitan University (TMU) Local 5G research support.

Declarations

Author contribution. All authors contributed equally and approved the final paper.
Funding statement. This research is funded by Japan Science and Technology Agency (JST), Moonshot R&D, with grant number JPMJMS2034, and TMU local 5G research support.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

Data and Software Availability Statements

We develop applications using Python 3.8, Windows 11 operating system, and open-source frameworks, including OpenCV 4.6.0 for standard computer vision applications, YOLOv5 and SORT for tracking objects, and Mediapipe 0.8.10.1 for tracking hands. For the learning environment, we use Pytorch 1.8.2 with several additional graph learning features utilizing Pytorch Geometric 2.0.4. We employ the Tobii Glasses 1.12.11 software for the eye tracker sensor reader and the RTSP protocol. We use PostgreSQL as the time-series database and Neo4j as the graph database. The code and data set can be accessed at https://github.com/anom-tmu/hoi-attention.

Appendix

Appendix 1 shows the commonly used notations in this paper.

Appendix 1. Commonly used notations.

Notations	Descriptions
G	A graph $G \in \mathcal{G}$.
$\mathcal{G}$	The set of graphs.
V	The set of nodes in a graph.
v, u	A node $v, u \in V$
X	The set of node features in a graph
x_v	A feature vector in a node v
E	The set of edges in a graph
$e_{i,j}$	An edge $e_{i,j} \in E$
$N(v)$	The neighbors of a node v
$h_{G,v}$	The embedding vector of a node v in a graph G
$m_{G,v}$	The embedding vector of aggregation result
H_G	The embedding vector of a graph G
W	The set of weight / learnable model parameter
f	A function
k, K	The layer index
t, T	The time step/interation index
i, j	The dimension of weight matrix
$\sigma(\cdot)$	The activation function
$\lvert\cdot\rvert$	The length of a set

References

[1] T. Singh, C. M. Perry, S. L. Fritz, J. Fridriksson, and T. M. Herter, "Eye Movements Interfere With Limb Motor Control in Stroke Survivors," *Neurorehabil. Neural Repair*, vol. 32, no. 8, pp. 724–734, Aug. 2018, doi: 10.1177/1545968318790016.

[2] M. Szekeres and K. Valdes, "Virtual health care & telehealth: Current therapy practice patterns," *J. Hand Ther.*, vol. 35, no. 1, pp. 124–130, Jan. 2022, doi: 10.1016/j.jht.2020.11.004.

[3] P. Wang, L. T. Yang, J. Li, J. Chen, and S. Hu, "Data fusion in cyber-physical-social systems: State-of-the-art and perspectives," *Inf. Fusion*, vol. 51, pp. 42–57, Nov. 2019, doi: 10.1016/j.inffus.2018.11.002.

[4] A. Laghari, Z. A. Memon, S. Ullah, and I. Hussain, "Cyber Physical System for Stroke Detection," *IEEE Access*, vol. 6, pp. 37444–37453, Jun. 2018, doi: 10.1109/ACCESS.2018.2851540.

[5] A. Rashid and O. Hasan, "Wearable technologies for hand joints monitoring for rehabilitation: A survey," *Microelectronics J.*, vol. 88, pp. 173–183, Jun. 2019, doi: 10.1016/j.mejo.2018.01.014.

[6] A. A. Saputra, A. R. A. Besari, and N. Kubota, "Human Joint Skeleton Tracking Using Multiple Kinect Azure," in *2022 International Electronics Symposium (IES)*, Aug. 2022, pp. 430–435, doi: 10.1109/IES55876.2022.9888532.

[7] M. Dousty and J. Zariffa, "Tenodesis Grasp Detection in Egocentric Video," *IEEE J. Biomed. Heal. Informatics*, vol. 25, no. 5, pp. 1463–1470, May 2021, doi: 10.1109/JBHI.2020.3003643.

[8] M. Cai, K. Kitani, and Y. Sato, "Understanding hand-object manipulation by modeling the contextual relationship between actions, grasp types and object attributes, " pp. 1-14, July. 2018. [Online]. Available at: https://arxiv.org/abs/1807.08254v1.

[9] M.-F. Tsai, R. H. Wang, and J. Zariffa, "Identifying Hand Use and Hand Roles After Stroke Using Egocentric Video," *IEEE J. Transl. Eng. Heal. Med.*, vol. 9, pp. 1–10, 2021, doi: 10.1109/JTEHM.2021.3072347.

[10] A. R. A. Besari, A. A. Saputra, W. H. Chin, N. Kubota, and Kurnianingsih, "Hand-Object Interaction Detection based on Visual Attention for Independent Rehabilitation Support," in *2022 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2022, vol. 2022-July, pp. 1–6, doi: 10.1109/IJCNN55064.2022.9892903.

[11] T. Wang, T. Yang, M. Danelljan, F. S. Khan, X. Zhang, and J. Sun, "Learning Human-Object Interaction Detection Using Interaction Points," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 4115–4124, doi: 10.1109/CVPR42600.2020.00417.

[12] D. Qurratu'aini, A. Sophian, W. Sediono, H. Md Yusof, and S. Sudirman, "Visual-Based Fingertip Detection for Hand Rehabilitation," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 9, no. 2, p. 474, Feb. 2018, doi: 10.11591/ijeecs.v9.i2.pp474-480.

[13] J. Likitlersuang, E. R. Sumitro, T. Cao, R. J. Visée, S. Kalsi-Ryan, and J. Zariffa, "Egocentric video: a new tool for capturing hand use of individuals with spinal cord injury at home," *J. Neuroeng. Rehabil.*, vol. 16, no. 1, p. 83, Dec. 2019, doi: 10.1186/s12984-019-0557-1.

[14] R. J. Visee, J. Likitlersuang, and J. Zariffa, "An Effective and Efficient Method for Detecting Hands in Egocentric Videos for Rehabilitation Applications," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 28, no. 3, pp. 748–755, Mar. 2020, doi: 10.1109/TNSRE.2020.2968912.

[15] A. Bandini, M. Dousty, S. L. Hitzig, B. C. Craven, S. Kalsi-Ryan, and J. Zariffa, "Measuring Hand Use in the Home after Cervical Spinal Cord Injury Using Egocentric Video," *J. Neurotrauma*, vol. 39, no. 23–24, pp. 1697–1707, Dec. 2022, doi: 10.1089/neu.2022.0156.

[16] J. Xu, P. Mohan, F. Chen, and A. Nurnberger, "A Real-time Hand Motion Detection System for Unsupervised Home Training," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct. 2020, vol. 2020-Octob, pp. 4224–4229, doi: 10.1109/SMC42975.2020.9283261.

[17] Y. Li, L. Jia, Z. Wang, Y. Qian, and H. Qiao, "Un-supervised and semi-supervised hand segmentation in egocentric images with noisy label learning," *Neurocomputing*, vol. 334, pp. 11–24, Mar. 2019, doi: 10.1016/j.neucom.2018.12.010.

[18] Y. Lee, W. Do, H. Yoon, J. Heo, W. Lee, and D. Lee, "Visual-inertial hand motion tracking with robustness against occlusion, interference, and contact," *Sci. Robot.*, vol. 6, no. 58, Sep. 2021, doi: 10.1126/scirobotics.abe1315.

[19] G. Kapidis, R. Poppe, and R. C. Veltkamp, "Multi-Dataset, Multitask Learning of Egocentric Vision Tasks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE Computer Society, 2021, pp. 1–1, doi: 10.1109/TPAMI.2021.3061479.

[20] K. Hesseberg, G. G. Tangen, A. H. Pripp, and A. Bergland, "Associations between Cognition and Hand Function in Older People Diagnosed with Mild Cognitive Impairment or Dementia," *Dement. Geriatr. Cogn. Dis. Extra*, vol. 10, no. 3, pp. 195–204, Dec. 2020, doi: 10.1159/000510382.

[21] J. Jiang, Z. Nan, H. Chen, S. Chen, and N. Zheng, "Predicting short-term next-active-object through visual attention and hand position," *Neurocomputing*, vol. 433, pp. 212–222, Apr. 2021, doi: 10.1016/j.neucom.2020.12.069.

[22] R. Tanaka, J. Woo, and N. Kubota, "Nonverbal Communication Based on Instructed Learning for Socially Embedded Robot Partners," *J. Adv. Comput. Intell. Intell. Informatics*, vol. 23, no. 3, pp. 584–591, May 2019, doi: 10.20965/jaciii.2019.p0584.

[23] M. Yani, A. R. A. Besari, N. Yamada, and N. Kubota, "Ecological-Inspired System Design for Safety Manipulation Strategy in Home-care Robot," in *2020 International Symposium on Community-centric Systems (CcS)*, Sep. 2020, pp. 1–6, doi: 10.1109/CcS49175.2020.9231354.

[24] A. R. A. Besari, A. A. Saputra, W. H. Chin, Kurnianingsih, and N. Kubota, "Finger Joint Angle Estimation With Visual Attention for Rehabilitation Support: A Case Study of the Chopsticks Manipulation Test," *IEEE Access*, vol. 10, no. September, pp. 91316–91331, 2022, doi: 10.1109/ACCESS.2022.3201894.

[25] A. A. Saputra, K. Wada, S. Masuda, and N. Kubota, "Multi-scopic neuro-cognitive adaptation for legged locomotion robots," *Sci. Rep.*, vol. 12, no. 1, p. 16222, Sep. 2022, doi: 10.1038/s41598-022-19599-2.

[26] K. Oshio, K. Kaneko, and N. Kubota, "Multi-scopic Simulation for Human-robot Interactions Based on Multi-objective Behavior Coordination," in *International Workshop on Advanced Computational Intelligence and Intelligent Informatics*, 2021, no. Iwaciii, pp. 3–8. [Online]. Available at: https://iwaciii2021.bit.edu.cn/docs/2021-12/b3d6c84e7e244c6e89cf502ed15cdc17.pdf.

[27] P. Pradhyumna, G. P. Shreya, and Mohana, "Graph Neural Network (GNN) in Image and Video Understanding Using Deep Learning for Computer Vision Applications," in *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Aug. 2021, pp. 1183–1189, doi: 10.1109/ICESC51422.2021.9532631.

[28] Y. J. R. De Kloe, I. T. C. Hooge, C. Kemner, D. C. Niehorster, M. Nyström, and R. S. Hessels, "Replacing eye trackers in ongoing studies: A comparison of eye-tracking data quality between the Tobii Pro TX300 and the Tobii Pro Spectrum," *Infancy*, vol. 27, no. 1, pp. 25–45, Jan. 2022, doi: 10.1111/infa.12441.

[29] H. Fu, L. Wu, M. Jian, Y. Yang, and X. Wang, "MF-SORT: Simple Online and Realtime Tracking with Motion Features," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11901 LNCS, Springer, 2019, pp. 157–168, doi: 10.1007/978-3-030-34120-6_13.

[30] V. Chunduru, M. Roy, D. R. N. S, and R. G. Chittawadigi, "Hand Tracking in 3D Space using MediaPipe and PnP Method for Intuitive Control of Virtual Globe," in *2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC)*, Sep. 2021, pp. 1–6, doi: 10.1109/R10-HTC53172.2021.9641587.

[31] A. R. Anom Besari, W. H. Chin, N. Kubota, and Kurnianingsih, "Ecological Approach for Object Relationship Extraction in Elderly Care Robot," in *2020 21st International Conference on Research and Education in Mechatronics (REM)*, Dec. 2020, pp. 1–6, doi: 10.1109/REM49740.2020.9313877.

[32] R. Volcic and F. Domini, "The endless visuomotor calibration of reach-to-grasp actions," *Sci. Rep.*, vol. 8, no. 1, p. 14803, Oct. 2018, doi: 10.1038/s41598-018-33009-6.

[33] A. Pareja *et al.*, "EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs," *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 04, pp. 5363–5370, Apr. 2020, doi: 10.1609/aaai.v34i04.5984.

[34] A. R. Anom Besari, A. A. Saputra, W. H. Chin, N. Kubota, and Kurnianingsih, "Feature-based Egocentric Grasp Pose Classification for Expanding Human-Object Interactions," in *2021 IEEE 30th International Symposium on Industrial Electronics (ISIE)*, Jun. 2021, vol. 2021-June, pp. 1–6, doi: 10.1109/ISIE45552.2021.9576369.

[35] A. Vysocký *et al.*, "Analysis of Precision and Stability of Hand Tracking with Leap Motion Sensor," *Sensors*, vol. 20, no. 15, p. 4088, Jul. 2020, doi: 10.3390/s20154088.

[36] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Min. Knowl. Discov.*, vol. 31, no. 3, pp. 606–660, May 2017, doi: 10.1007/s10618-016-0483-9.

[37] H. Hanafi, N. Suryana, and A. S. H. Basari, "Dynamic convolutional neural network for eliminating item sparse data on recommender system," *Int. J. Adv. Intell. Informatics*, vol. 4, no. 3, p. 226, Nov. 2018, doi: 10.26555/ijain.v4i3.291.

[38] R. Tanaka, J. Woo, and N. Kubota, "Action Acquisition Method for Constructing Cognitive Development System Through Instructed Learning," in *2019 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2019, vol. 2019-July, pp. 1–6, doi: 10.1109/IJCNN.2019.8852180.

[39] G. H. Martono, A. Azhari, and K. Mustofa, "An extended approach of weight collective influence graph for detection influence actor," *Int. J. Adv. Intell. Informatics*, vol. 8, no. 1, p. 1, Mar. 2022, doi: 10.26555/ijain.v8i1.800.