OPUS-HSO

Repositorium der Hochschule Offenburg
https://opus.hs-offenburg.de

Enhancing Independence through Intelligent Robotics: An AI Driven Assistive Robotics Interface

Katrin-Misel Ponomarjova, Thomas Wendt, Anke Fischer-Janzen, Sinan Emre Süme, Bastian Kayser

Zitiervorschlag im APA Stil:

Ponomarjova, K.-M., Wendt, T., Fischer-Janzen, A., Süme, S. E., & Kayser, B. (2024). Enhancing Independence through Intelligent Robotics: An AI Driven Assistive Robotics Interface. In J. Weber & J.-P. Lauffenburger (Eds.), *UR-AI2023 : The Upper-Rhine Artificial Intelligence Symposium : Artificial Intelligence for Time Series, Robotics and Beyond*. https://urai2023.sciencesconf.org/data/pages/book_urai2023_en_2024.pdf

Abstract

Applying methods in artificial intelligence to the field of assistive robotics has the potential to increase the independence of people with disabilities. The usage of AI to realize a shared control in this context is controversial, due to the high complexity of everyday tasks and the needed safety requirements. This paper presents the development of a user interface for AI-driven assistive robotic arms (ARA) that aims to assist people with physical disabilities in performing daily activities. This interface allows the user to select object manipulation tasks based on the objects recognized in a live video stream. Further, we compare several state-of-the-art, real-time object detection models to facilitate automatic robotic control. The results demonstrate the feasibility of the model and its potential integration into the overall robotic system.

Nutzungsbedingungen

Dieses Dokument wird unter diesen Bedinungen zur Verfügung gestellt:
Urheberrechtlich geschützt
Für weitere Informationen siehe:
https://rightsstatements.org/page/InC/1.0/

Kontakt

Hochschule Offenburg | Bibliothek
Badstraße 24
77652 Offenburg
Telefon: (0781) 205-240
E-Mail: bibliothek@hs-offenburg.de
www.hs-offenburg.de/bibliothek

Enhancing Independence through Intelligent Robotics: An AI-Driven Assistive Robotics Interface

Katrin-Misel Ponomarjova[1], Thomas M. Wendt[1], Anke Fischer-Janzen[1], Sinan Süme[1], Bastian Kayser[2]

[1] Offenburg University of Applied Sciences
katrin-misel.ponomarjova@hs-offenburg.de
[2] Cadida Software GmbH
kayser@cadida.de

Abstract. Applying methods in artificial intelligence to the field of assistive robotics has the potential to increase the independence of people with disabilities. The usage of AI to realize a shared control in this context is controversial, due to the high complexity of everyday tasks and the needed safety requirements. This paper presents the development of a user interface for AI-driven assistive robotic arms (ARA) that aims to assist people with physical disabilities in performing daily activities. This interface allows the user to select object manipulation tasks based on the objects recognized in a live video stream. Further, we compare several state-of-the-art, real-time object detection models to facilitate automatic robotic control. The results demonstrate the feasibility of the model and its potential integration into the overall robotic system.

Keywords: Assistive Robotics; Real-Time Object Detection; Human-Robot Interaction.

1 Introduction

The growing need for technological assistance in caretaking leads to increased developments in the field of assistive robotics. Currently, assistive robotic systems offer the user the possibility to directly control an ARA through various control inputs, such as eye tracking [1]. Physically impaired people with paraplegia or tetraplegia can be assisted with assistive robotic arms (ARA). Few models are available on the market, such as JACO, MICO (Kinova) or the iARM (Assistive Innovations bv). These ARA can be mounted on an electric wheelchair or on the side of the bed for bed bound patients. Reasons for such paralysis can be spinal injuries, stroke and various diseases such as cerebral palsy, amyotrophic lateral sclerosis and multiple sclerosis. According to recent statistics published by the German Federal Statistical Office, 7,8 million people in Germany have a severe disability of which approximately 1,6 million have a spinal or limb disability [2].

The design of interfaces to support these people is a crucial part of making a system usable and accessible, and presents a variety of challenges. Combining an ARA with other technologies, such as smart home appliances, lead to new opportunities. For example, the work of Brunete et al. shows that the user could interact over a tablet with individual input interfaces to control a mobile platform, a robotic arm, as well as IoT devices such as the heating, lighting, and shutters [3]. Furthermore, ARA can be controlled by joysticks, speech recognition, head or eye tracking, or even Brain Computer Interfaces (BCI). With the exception of joystick and voice control, most of these solutions are controlled via a display or tablet that provides the directional control of the robotic arm or selection of individual tasks the robot is capable of performing [4].

In this context the question arises as to whether usability, measured e.g. by ease of use [5], can be facilitated through the use of AI by displaying selectable items for interaction on these displays. Available state-of-the-art computer vision models enable us to detect objects in real time. In addition, advances in robot automation allow robots to generalize control tasks such as object manipulation to novel objects and environments.
In addition, a compromise must be found between the autonomy of the robot control and the controllability and interpretability by the user. Current work in AI and robotics shows promising results for the use of multimodal large language models (LLMs) to control robot agents [6, 7,8]. Here, tasks are given in natural language and the robot navigates and executes the given tasks based on the visual input. While this approach is promising for a variety of different use cases, most of the identified systems do not provide any visual feedback on which task or target objects have been identified or will be interacted with. In the context of assistive robotics, this can lead to a feeling of being at the mercy of a "black box" model.

In this work, we aim to develop a system that focuses on usability and accessibility by hiding the underlying complexity of robot control. This can be achieved through the use of an object detection model which simplifies the control and allows the user to accomplish complex tasks of everyday life.

2 Methods

2.1 System overview

To meet the needs of paraplegic people, the HIRAC (Hardware Independent Robotic Assistance Controller) project is developing a robotic system that enables people to perform activities of daily living.

The system consists of an ARA that can be mounted on an electric wheelchair or at the bedside. The different components communicate via the Robot Operating System 2 (ROS2 Humble Hawksbill). A camera is mounted on the robotic arm, allowing the user to move the field of view, which is necessary for bed-bound users. The touch display is mounted in the user's line of sight and displays a live video stream for ease of use.

2.2 Design of the User Interface (UI)

The four principles of accessibility outlined in the Web Content Accessibility Guidelines can be applied to the user interface presented in this work. They include perceptibility, usability, understandability, and robustness [9]. They resemble the seven principles of universal design presented by Story, which were developed as a benchmark for product design so that "people of all ages and abilities" can use them without the need of adaptation [5]. They consist of Equitable Use, Flexibility in Use, Simple and Intuitive Use, Perceptible Information, Tolerance for Error, Low Physical Effort, and Size and Space for Approach and Use [5]. Based on these principles, a requirements analysis was conducted to determine the critical design elements for a UI with high usability satisfaction. Existing solutions for this use case were identified and compared to this system. For the target group of paraplegic and tetraplegic persons, refinements in the accessibility were made. Regarding these insights, a mockup of the user interface and the user flow are presented in the following.

2.3 Evaluation methodology of the object detection network

The YOLO algorithm has become widely adopted in a variety of applications where real-time inference is required. It performs exceptionally well in terms of inference speed and detection

accuracy when compared to other real-time object detection algorithms such as Faster R-CNN (Faster Region based Convolutional Neural Networks) and SSD (Single Shot Detection) [10]. In addition, previous versions of YOLO have been successfully integrated with ROS2 [11], which is an essential part of the system. For these reasons, we decided to focus our attention on it.

Proposed method

Projects requiring real-time detection must consider the trade-off between accuracy and speed. Larger models tend to be slower in terms of FPS [12], but generally show better performance in terms of mAP (Mean Average Precision).

Therefore, we compare two different model sizes, nano ("n") and small ("s"), from YOLOv5 [13] and YOLOv8 [14] (both pre-trained on the MS COCO dataset), trained with and without data augmentation. A Nvidia RTX A2000 Laptop GPU (4096MiB) is used for training and evaluation.

As for the evaluation metrics used in this comparison, the mAP50-95 allows us to assess the precision of the detections, while the FPS will allow us to measure the real-time processing speed of the model. The FPS in this paper is calculated on the basis of average speed of the detection in milliseconds when processing the validation set, as shown in Equation (1). These metrics, as well as the Average Precision and Recall, can be found in Tab. 2.

$$FPS = \frac{1000}{preprocess_{ms} + inference_{ms} + postprocess_{ms}} \quad (1)$$

Dataset preparation

We create a dataset of images and annotations of keys, pens, cups and glasses by merging three datasets [15, 16, 17]. Initial poor performance in the "pen" class due to class imbalance was addressed by increasing the number of examples in this class from three additional datasets [18, 19, 20]. The final dataset used for training can be found on Github [21].

A data augmentation strategy is then implemented consisting of the modifications that showed the highest performance improvement: crops of up to 50% and Gaussian blur of up to 2px. This allowed us to generate 4198 additional images in the training set (Tab. 1).

As the objects to be detected are close and large, it is possible to train on smaller image sizes to decrease the training time. The images are trained over 10 epochs to see initial results. The image size used to train these networks is 320x320, then increased to 640x640 for the best performing networks trained over 40 epochs.

Table 1. Distribution of the dataset into training, validation and test sets

	Original			Data aug.
Class	*Train (70%)*	*Valid (20%)*	*Test (10%)*	*Train (70%)*
Keys	603	193	86	1809
Cup/Glass	1062	268	138	3186
Pen	434	139	75	1302
Total	**2099**	**600**	**299**	**6297**

3 Results

3.1 Design of a user interface for Assistive Robotics

After identifying the target group, a requirements analysis was carried out. Regarding the outcomes of Story [5], it is theorized that the user's cognitive workload can be reduced by automating the grasping process and task completion. With regard to the other benchmarks mentioned above, we focused on the usability and accessibility of the system. Several challenges were identified during the design process:

- How to present the available tasks to the user?
- How to ensure that the object detection model is able to identify the object and enable the task execution?
- What safety measures are necessary to ensure the safety of the user?
- What measures need to be taken to ensure usability and accessibility?

Safety regulations were not tested on the design, due to its early state. Controlling a robot at such a short distance to the user by AI raises new challenges and questions. Regulations of AI application are currently debated by the European Commission [22].

First, the tasks were determined. Activities of Daily Living (ADL) are describing a set of everyday tasks for self-maintaining. They are listed by the International Classification of Functioning, Disability and Health (ICF) [23]. Severe physically disabled people mostly need assistance from another person for these tasks, as robots are not able to help in some ADL such as bathing. We narrowed down the range of tasks which are feasible for the robot such as picking and placing objects, pouring a drink and handing it to the user, as well as handing food to the user. With these tasks, basic independence can be regained.

Secondly, the object detection model which will be presented in the next section is able to handle a moving workspace, considering that the ARA will be mounted on an electric wheelchair. The position of the robot has to be adapted to present the desired object. This leads to the importance of directional control and a fixed robot observation orientation. This was achieved by arrows on the sides, top and bottom of the touch screen visible in Fig. 1 which move the robot in the x- and y-planes. The robot's end effector is tilted around the y-axis at an angle of 35° to provide a bird's eye view of the scene. When the required object is detected in the scene, the system displays the bounding box around the object.

The bounding boxes are clickable. Once one is selected, the available tasks are presented to the user. When a task is selected, a progress bar is displayed representing the time the robot needs to complete the task.

Thirdly, rudimental safety features must be considered, such as a stop button that interrupts the task itself. Furthermore, measures such as a reset button to bring the robot back into position are needed to free the arm for movement or to resolve inadequate robot positions.

Lastly, the usability and accessibility of the system was examined. The need to eliminate seizure triggers was identified and implemented. Basic settings such as adapting the font size, adding pictograms to the texts, simple language and the possibility of feedback were included. Specific settings for the robot include the ability to adjust the speed, which was mentioned by participants in a previous study. It was considered important to design the UI in such a way that other input modalities, such as eye tracking, could be implemented at a later stage to ensure accessibility, for example for people with locked-in syndrome. All identified features are shown in the user flow in Fig. 1.

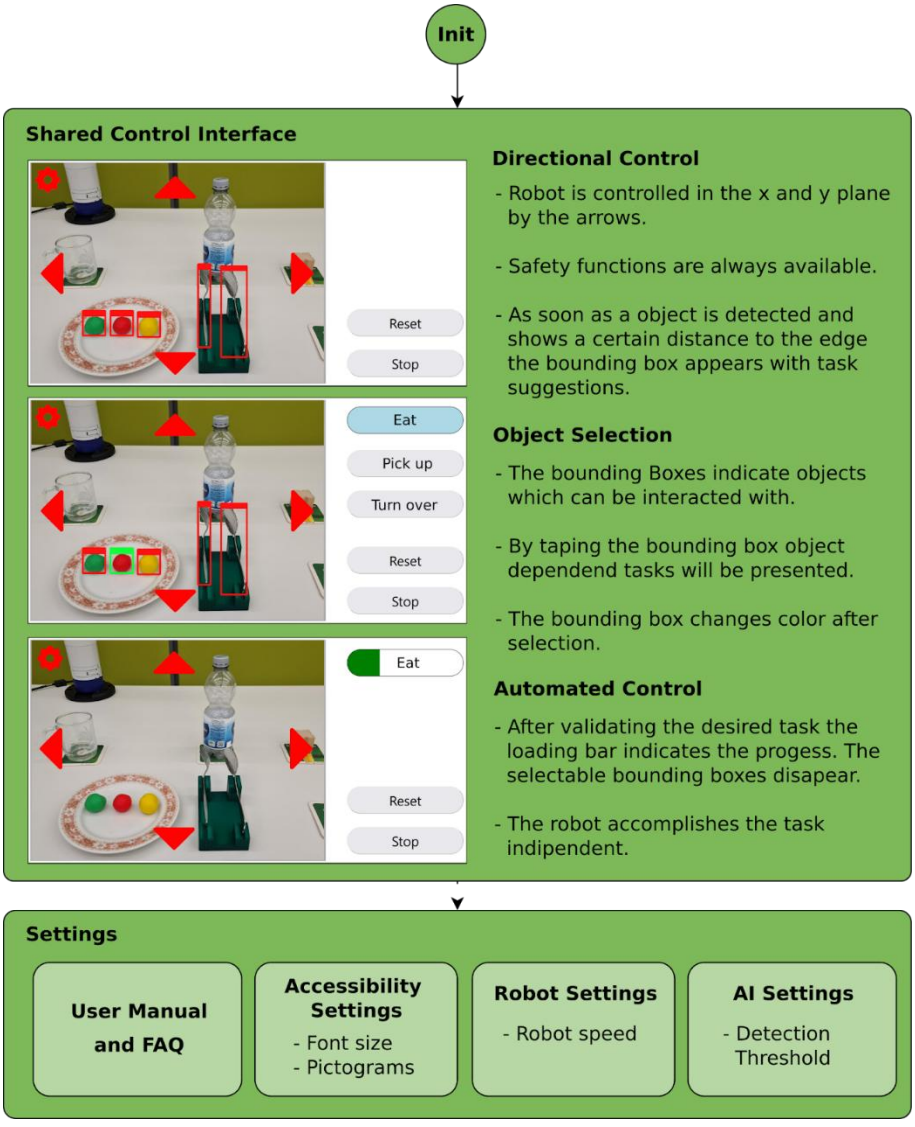

Fig. 1. User flow and mockup design of the user interface

3.2 Performance evaluation of computer vision models

Tab. 2 shows the results of training each network over 10 epochs with and without augmented data. Except for YOLOv8 trained on augmented data, there is a noticeable decrease in FPS as the network size was increased from “nano” to “small”. There is also a noticeable improvement in mAP for models trained on augmented data. The last rows represent the training of YOLOv8s and YOLOv5n over 40 epochs on augmented data.

Table 2. Results of training various size networks on datasets with and without augmentation

	Image size	Aug.	P	R	mAP50-95: all	mAP50-95: keys	mAP50-95: pen	mAP50-95: cup	mAP50-95: glass	FPS
YOLOv5n	320px	No	0.959	0.928	0.703	0.684	0.572	0.766	0.791	434.78
YOLOv5s	320px	No	0.94	0.928	0.71	0.717	0.54	0.773	0.808	260.55
YOLOv8n	320px	No	0.942	0.919	0.702	0.682	0.563	0.766	0.798	454.55
YOLOv8s	320px	No	0.936	0.934	0.723	0.725	0.576	0.795	0.795	333.88
YOLOv5n	320px	Yes	0.952	0.945	0.739	0.712	0.654	0.789	0.802	370.37
YOLOv5s	320px	Yes	0.941	0.947	0.738	0.736	0.619	0.788	0.81	336.63
YOLOv8n	320px	Yes	0.953	0.93	0.741	0.734	0.631	0.791	0.807	370.37
YOLOv8s	320px	Yes	0.945	0.949	0.748	0.751	0.633	0.8	0.809	419.90
YOLOv5n (40 ep.)	320px	Yes	0.954	0.944	0.759	0.753	0.661	0.8	0.82	434.78
YOLOv5n (40 ep.)	640px	Yes	0.954	0.954	0.775	0.782	0.666	0.825	0.83	487.8
YOLOv8s (40 ep.)	640px	Yes	0.951	0.953	0.77	0.783	0.635	0.824	0.838	135.14

YOLOv5n was particularly fast to train due to its small size and the dimensions of the training images (320x320px). The model achieved comparably high FPS on the validation set. Even when the size of the training images was increased to 640px, the training time was still about five times shorter than for YOLOv8s. This is an advantage in the future, as it could allow for faster parameter and dataset optimization.

Fig. 2. Predictions on images from outside the dataset, the confidence of the object detections were pen 0.9, cup 0.88 and keys 0.92.

Considering the high FPS achieved with YOLOv5n (trained on 640x640px images over 40 epochs), the short training time and the improved mAP across classes, it is the object detection model we have chosen for this project. Fig. 2 shows exemplary predictions by YOLOv5n on images from outside of our dataset, demonstrating the model's ability to generalize.

4 Discussion

4.1 Discussion on user flow and design elements of the UI

The user flow shown in Fig. 1 shows the steps the user has to take to perform a task with the robot. The design of the interface was chosen to present buttons and the video image as large as possible to improve the usability for paralyzed people and to ensure the usability with other input methods such as eye tracking or BCIs. A reason is the low accuracy of head-mounted eye tracking devices. In BCIs flickering buttons are sometimes presented at different frequencies [24]. These differences are visible in the recorded brain activity data and a selection of the button can be made. As the eye is also constantly moving, large buttons provide a larger field in which the user could focus. For this reason, the directional control buttons were placed within the video frame.

The overlap of bounding boxes and arrows would create a new challenge. The risk of pressing the wrong button, known as the Midas Touch Problem in eye tracking and gaze recognition, has to be eliminated. As a precaution, the clickable bounding boxes are only displayed if the object is a certain distance from the arrows. This also leads to a better view of the object, which is helpful for trajectory planning. Before completing the task, the user is also prompted to confirm their selection. To avoid annoying behavior by appearing and disappearing bounding boxes, the detection threshold of the object can be manipulated within a certain range. This results in either faster detection of the objects which can be selected by the user or less flickering, which can reduce seizure triggers.

Finally, the approach only allows the user to click on one bounding box and execute a task with it. Complex tasks such as pouring a glass of water could be represented by selecting multiple bounding boxes e.g., glass and bottle. However, in the real world, the glass and the bottle would be seldom in the picture at the same time. A pouring task is represented as grasping the bottle, remembering that a bottle was grasped and as soon as the user selects the cup after the user moves it into the field of view, the pouring task is proposed.

4.2 Discussion on the evaluation and results of the object detection algorithm

The object detection model was able to achieve satisfactory results even after initial difficulties with, for example, the "pen" class. The object itself has a more challenging shape to detect in comparison to cups, for example. In addition to this, training data on pens was difficult to come by.

More generally, it was interesting to note that an increase in network size from "nano" to "small" did not always translate to an increase in performance. Some possible reasons for this are that the data was limited, the dataset not as diverse as would be ideal, and as the complexity of the model increased with size. It may not be large and diverse enough to support the model, potentially leading to the drop in performance. In addition, the bounding boxes were not all manually verified in each image. Some bounding box errors were corrected, such as the bounding boxes not fitting tightly enough around each object, leading to an increase in

performance, but if any noisy or incorrect labels remained, a larger model would tend to amplify the impact of the errors, leading to a decrease in performance.

The main bottleneck to improve the performance of the object detection model would be the quality of the dataset, which could be solved in the future by creating a new dataset using frames from a stream of robotic arm in motion as this would lead to more domain-aware data than the available datasets, taking into account realistic angles the field of view could have from the perspective of a camera mounted on a robotic arm. It would also be possible to integrate objects such as specific types of cups commonly used in assistive care.

It is planned to combine the object detection network with the DeepSORT tracking algorithm. This type of algorithm would enable stable tracking of objects, even in situations where there is temporary partial or total occlusion of an object. When the object exits the field of view, its existence would not be instantly forgotten in a case where there are multiple objects of the same class available, for example. Solving these challenges will provide new insight in the design of AI-driven ARA user interfaces.

5 Conclusion

In this work we presented a user flow adapted for the use by physically impaired people. By automating task execution with a robot, it is assumed that the cognitive load of the user can be reduced. This shared control was realized by using object detection and linking tasks to the object classes. The best model was determined by evaluating FPS and mAP of different variations of YOLOv5 and YOLOv8. After training the models with data from various databases, YOLOv5n showed the best performance for this application. Part of future work is the development of an extended number of available tasks. Further, the usability will be tested with users and feedback will be collected.

6 Acknowledgement

The results were developed within the context of the ZIM project HIRAC, funded by the Federal Ministry for Economic Affairs and Climate Action.

References

1. Dragomir, A., Pana, C.F., Cojocaru, D., Manga, L.F.: Human-Machine Interface for Controlling a Light Robotic Arm by Persons with Special Needs, doi: 10.1109/ICCC51557.2021.9454664, 2021.
2. Destatis 2022. O. V., Press release No. 259 of 22 June 2022: 7.8 million severely disabled people living in Germany, Statistisches Bundesamt (Hrsg.), published 22.06.2022. URL: https://www.destatis.de/DE/Presse/Pressemitteilungen/2022/06/PE22_259_227.html, last accessed on 22.09.2023.
3. Brunete, A., Gamabo, E., Hernando, M., and Cedazo, R.: Smart Assistive Architecture for the Integration of IoT Devices, Robotic Systems, and Multimodal Interfaces in Healthcare Environment, Sensors 21 (6), DOI: 10.3390/s21062212, 2021.
4. Fischer-Janzen, A.: Eye Tracking in Robot Control Tasks, URLS: https://github.com/AnkeLinus/EyeTrackingInRobotControlTasks.git, https://doi.org/10.5281/zenodo.10003313, published: 13.10.2023, last accessed on 13.10.2023.
5. Story, M.F.: Maximizing Usability: The Principles of Universal Design, Assistive technology, 10, (1), DOI: 10.1080/10400435.1998.10131955, 1998.

6. Brohan, A., Brown, N., Carbajal, J., et al.: RT-1: Robotics Transformer for Real-World Control at Scale. arXiv, 13. December 2022. https://doi.org/ 10.48550/arXiv.2212.06817, 2022.
7. Driess, D., Xia, F., Sajjadi, M.S.M., Lynch C. et al. "PaLM-E: An Embodied Multimodal Language Model". arXiv, 6 March 2023. http://arxiv.org/abs/2303.03378.
8. Vemprala, S., Bonatti, R., Bucker, A., and Kapoor, A.. "ChatGPT for Robotics: Design Principles and Model Abilities", o. J.
9. Thesmann, S.: Interface Design, Ed. 2, Springer Vieweg, Wiesbaden, ISBN: 365803856X, 2016.
10. Srivastava, S., Divekar, A.V., Anilkumar, C., naik, I., Kulkarni, V. and Pattabiraman V.: "Comparative analysis of deep learning image detection algorithms". J Big Data 8, 66 (2021). https://doi.org/10.1186/s40537-021-00434-w.
11. Ivorra, E., Ortega, M., Catalán, J.M., Ezquerro, S., Lledó, L.D., Garcia-Aracil, N., and Alcañiz, M.: "Intelligent Multimodal Framework for Human Assistive Robotics Based on Computer Vision Algorithms", 2018, DOI: 10.3390/s18082408.
12. Li, C., Li, L, Geng, Y., Jiang, H., Cheng, M., Zhang, B., et al.: "YOLOv6 v3.0: A Full-Scale Reloading". arXiv, 13 January 2023. https://arxiv.org/pdf/2301.05586.pdf.
13. Jocher, G.: YOLOv5 by Ultralytics, version 7.0, available at: https://github.com/ultralytics/yolov5, DOI: 10.5281/zenodo.3908559, last accessed on 06.07.2023, 2020.
14. Ultralytics: YOLOv8, available at: https://github.com/ultralytics/ultralytics, last accessed on 06.07.2023.
15. Tazas y vasos Dataset, Vision, Open Source Dataset, available at https://universe.roboflow.com/ vision-gxbqw/tazas-y-vasos, Roboflow Universe, Roboflow, 2022, last accessed on 2023-07-04.
16. Emad, S.: Keysdetection Dataset, Open Source Dataset, available at https://universe.roboflow.com/ shehab-emad-n2q9i/keysdetection/dataset/2, Roboflow Universe, Roboflow, 2023, last accessed on 2023-07-12.
17. Menshikov, S.: Pen Dataset, Open Source Dataset, available at https://universe.roboflow.com/ sergey-menshikov/pen-suawu, Roboflow Universe, Roboflow, 2022, last accessed on 2023-07-12.
18. pen-ufs_dataset, Open Source Dataset, available at https://universe.roboflow.com/pen-rgrwx/pen-ufs, Roboflow Universe, Roboflow, 2022, last accessed on 2023-10-11.
19. Dwi: pen-ctcrc_dataset, Open Source Dataset, available at https://universe.roboflow.com/dwi/ pen-ctcrc, Roboflow Universe, Roboflow, 2022, last accessed on 2023-10-11.
20. taitora8002@yahoo.co.jp: YOLO_PEN Dataset, available at https://universe.roboflow.com/ taitora8002-yahoo-co-jp/yolo_pen/dataset/1, Roboflow Universe, Roboflow, 2022, last accessed on 2023-10-11.
21. Ponomarjova, Katrin-Misel: HIRAC-dataset, URL: https://github.com/katrinmisel/HIRAC.git, published: 14.10.2023, last accessed on 14.10.2023.
22. Proposal for a Regulation of the European Parliament and of the Council, Laying Down Harmonized Rules on Artificial (Artificial Intelligence Act) and Amending Certain Union Legislative Acts, available: https://www.ce-richtlinien.eu/richtlinien/KI/Richtlinie/1Proposal-KI.pdf, last accessed on 09.10.2023
23. Rehadat, ICF, Aktivitäten und Partizipation, https://www.rehadat-icf.de/de/klassifikation/ aktivitaeten-partizipation/, last accessed on 22.09.2023.
24. Clark, A., and Ahmad, I.: Interfacing with Robots without the use of Touch and Speech in PETRA 2021, June 29-July 2, 2021, Corfu, Greece, DOI: 10.1145/3453892.3461330, 2021.