Efficient Task Organization with Commonsense Knowledge for Human-Robot Collaborative Tasks

Swagnik Roychoudhury
New York University

Aparna S. Varde
Montclair State University

Efficient Task Organization with Commonsense Knowledge for Human-Robot Collaborative Tasks

Swagnik Roychoudhury
Department of Computer Science
New York University
New York, New York
sr6474@nyu.edu

Aparna S. Varde
School of Computing
Montclair State University
Montclair, New Jersey
vardea@montclair.edu

Weitian Wang
School of Computing
Montclair State University
Montclair, New Jersey
wangw@montclair.edu

***Abstract*—We present a new and innovative approach called *DISCERN (Detection Image System with Commonsense Efficient Ranking Network)* to "discern" object selection priority designed for human-robot collaborative tasks. Our approach utilizes a combination of standard image models, a commonsense knowledge base (CSKB), a vision language model, and custom priorities derived from human intuition to determine an optimal order for the robot's actions. *DISCERN* is a competitive solution to extensive training or learning from human demonstrations and works out-of-the-box with effective results and minimal resources, hence implying low algorithmic complexity and high execution efficiency. We validated the proposed approach in a typical human-robot collaborative home dining table cleaning task, although they can be applied to any household setting. Experimental results and evaluations demonstrate that the developed DISCERN has significantly better performance than baseline methods.**

***Index Terms*—AI & Robotics, CSK, Commonsense Reasoning, Human-Robot Collaboration, Task Planning, Sustainable AI, XAI**

I. Introduction

Countless advances in machine learning and robotics made in the last few years have brought robots closer to achieving human-like cognition. However, the multiple processes involved for a robot to perform a task, such as vision, task organization, and task execution, each require extensive training and resources to function. Learning from Demonstration (LFD), or Imitation Learning can face difficulties, including the complexity of data collection, inability to generalize, dependency on demonstration quality, and susceptibility to policy drift if the data doesn't span the full state space.

Additionally, complex algorithms in machine learning often lack explainability, hence the reasoning for their actions can be hard to understand. As a result, it is harder for humans to trust robots in human-robot collaborative tasks [1]. While recent advances in Natural Language Processing (NLP) have enabled greater explainability in robotics via Large Language Models (LLMs) [2], they face many challenges that make their adoption in robots difficult, such as inaccurate outputs, bias and misconceptions, and higher (and possibly prohibitive) hardware and resource requirements [3].

The challenges of speed, reliability, hardware requirements, resource consumption, data collection, bias, generalization, general difficulty in recurrent training, and policy drift could potentially be solved or reduced if only robots had basic common sense analogous to humans. Enter DISCERN, an innovative method to "discern" object selection priority in household robotic tasks, as illustrated in Fig. 1. DISCERN is a novel approach that integrates image models, a commonsense knowledge base (CSKB), a vision language model (VLM), and custom priorities to optimize task ordering in domestic settings. It enhances the robot's decision-making by infusing intuitive, commonsense-based human judgment to enhance explainability, hence impacting XAI (explainable AI).

DISCERN deploys acquired commonsense knowledge (CSK) [4] in line with modern-day machine intelligence [5]. CSK in DISCERN reduces the need for pre-training via LFD or neural models that need huge training data and consume excessive energy. This is vital in domestic settings, e.g. dorm room/home, where resource consumption and energy efficiency can be crucial. Given DISCERN's capacity to function with one-time training coupled with knowledge-base usage, it has relatively low algorithmic complexity and thus low energy consumption for model training and execution. Hence, it can make broader impacts on Sustainable AI and Responsible AI.

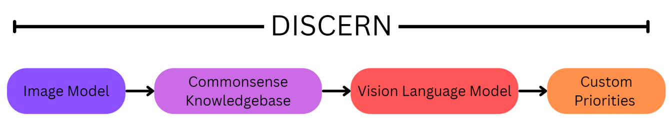

Fig. 1. High-level overview of the DISCERN approach.

Setting up image models and linking them to CSKBs poses unique challenges, including ensuring accurate object categorization and leveraging human intuition effectively. Translating an arbitrary feature space from an image model to a lower-dimensional embedding using a commonsense knowledge base in a manner that prevents leakage and misrepresentation proved difficult, but eventually that is precisely what allowed our robotic agent to use commonsense for household tasks.

While DISCERN makes partial use of a vision language model, the VLM only serves as a minor and optional refinement in the task ordering process. DISCERN also uses an image model but only for basic object detection; note that the typical number of objects in a given setting (e.g. household)

978-1-6654-7345-3/22/$31.00 ©2024 IEEE

is of the order of 100s, as opposed to millions of full images (e.g. of scenes in households) on which excessive pre-training may otherwise be needed to make robots function adequately. Hence, DISCERN heads more towards optimal behavior.

II. Related Works

Previous research has focused on using imitation learning and LfD for a robot to complete a task, e.g. IRIS [6] decomposes the task into parts, such that the robot imitates small tasks and combines parts of sub-optimal solutions to achieve successful task completions. While systems like IRIS can work with sub-optimal data, arriving at the most efficient solution still remains a gap. Dependency on training data means the algorithm faces difficulty in generalizing to unfamiliar contexts; lack of real-time feedback necessitates further training to improve such models. Image mining with multidisciplinary facets has been addressed in many works, e.g. [7], [8], and can be pertinent to robotics as well [9], [10].

Other works include terminology evolution [11] that can be vital in making AI systems understand current contexts. More recent work language processing has enabled the addition of language models in task organization in household settings. Since they are based on natural language, LLMs provide more explainability in actions, e.g. Ocker et al. explore LLMs to extract CSK, aiming to bridge the gap between implicit human understanding and robotic execution by populating ontologies with action patterns [12]. Their approach highlights challenges in reliably extracting context-specific and actionable knowledge. While LLMs can extract large volumes of general knowledge, they struggle with consistency and specificity, making it difficult to apply the extracted information effectively in dynamic real-world scenarios.

CSK in robots enhances their ability to perform tasks by adding cognitive reasoning [13]. For instance, Conti et al. [14] propose a CSK-HRC framework that harnesses commonsense knowledge in human-robot collaboration to improve robot action planning. CSK-Detector [15] is an approach deploying human-like common sense to categorize images in a task-relevant manner based on the most vital objects detected in them. The Robo-CSK-Organizer system [16] adapts CSK to organize detected objects in pertinent locations with high explainability. Our work DISCERN expands on such frameworks to optimize robotic operations in dynamic environments. It improves human-robot trust and efficiency. Hence it can positively impact greenness, analogous to next-generation data centers [17], making robots energy-efficient & sustainable.

III. The DISCERN Approach

We propose an approach *DISCERN* (Detection Image System with Commonsense Efficient Ranking Network) to "discern" object selection priority in robotics. It works as follows.

A. Algorithm Design

The main algorithm of DISCERN has four parts. The first part is the object detection and localization via an image model. We compared ResNet50 [18], YOLO-world [19], DETR [20], and DETIC [21], and chose DETIC (trained with LVIS [22]) due its advantages of mask segmentation.

The second part is context detection, object categorization, and attribute assignment via a commonsense knowledge base. By detecting the context of the objects and categorizing them, we can embed an arbitrary output size from our image model into a pre-defined set of categories, as in Fig. 2. This allows us to assign default attributes by the object's category without having to hard-code the attributes for every possible object class in our image model, which would be greatly time-consuming and inefficient (e.g. DETIC outputs over 1200 classes). The categories used in our home dining table setup are *container* (for food and drink), *utensil*, *cookware*, *furniture*, and *decoration*. While many common sense knowledge bases exist, such as Cyc [23] and OMCS [24], we found that ConceptNet [25] suited our needs the best due to its logical paths and general to specific inferences.

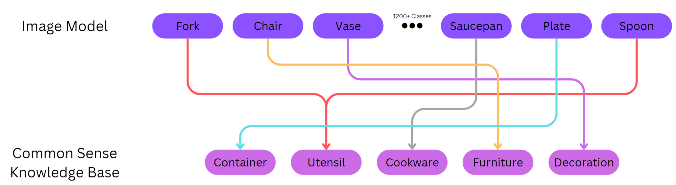

Fig. 2. Categorization from Image Model (input) to CSK Model (output)

Context detection discerns the environment, e.g. "kitchen", "bedroom" or "bathroom", using the objects detected by the image model, which is important to properly assign default attributes to each object. After pruning ConceptNet to include only relevant nodes/edges and relations, making traversal more efficient, we perform a modified breadth-first search (BFS) on ConceptNet using the following edges: /r/AtLocation, /r/HasContext, /r/HasProperty, /r/IsA & /r/RelatedTo.

Categorization occurs similarly, although with extra constraints. For example, certain decorative items, such as a vase, might be misclassified as a "container". However, the "container" category is reserved for food/drink-related containers, so a vase object should be categorized as decoration. To prevent such leaks from occurring, we also check if "food" or "liquid" are immediate neighboring nodes before assigning the "container" category. This ensures that only food containers (via the "food" node) and liquid containers (via the "liquid" node) are categorized as a "container".

Once one of the target categories is found in our modified BFS query, we note the depth and finish the rest of the neighboring nodes at that depth before ending our search. Any other targets that are found at the same depth are collected, and the category is the mode (most found) target in the list. This prevents misclassification due to the alphabetical traversal of the search, as exemplified in Fig. 3.

Once we have categorized each object into one of the five predefined categories, we assign default attributes unique to each category. The assignable attributes can be found in the "Robot Attributes" of Table I. All objects in the same category are assigned the same attributes. The distance attribute is a placeholder for depth sensor data in future experiments and is estimated using an elliptical distance field from the camera.

Part 3 of the algorithm is an optional state-aware refinement of the default attributes. For each object, we input the default attributes and an extracted image of the object into a VLM, which then refines specific attributes according to the object's state. In our experiments, we used GPT4o as our VLM.

Finally, the objects' attributes go through a formula designed by us to mimic human-like commonsense priorities based on size, danger to the human, distance, and other attributes to finalize an order. Empirically, the formula prioritizes objects that are heavier and larger, followed by those posing a danger to the user, and those closer to the robot.

Objects are divided into three lists: close, far, and ignore. Objects that are under the decoration or furniture category are ignored, as the robot should not be touching tables, chairs, and decorative items on/around the table. Objects that are beyond the robot's reach, determined by the distance to the centroid of the object and an empirical threshold, are present in the "far" list. Both the close and far lists are sorted according to our common-sense mimicking formula, and the final order constitutes the output. Fig. 4 is a visual depiction of ordering on a sample image. Green boxes depict objects within the robot's reach, orange boxes represent objects outside the robot's reach, and red boxes are objects that should not be touched by the robot (i.e. decorative items and furniture). Each box is labeled with a number, which is that object's rank in the object priority queue as determined by our algorithm. [1]

B. Simulation Environment Development

The simulation for DISCERN's execution is made with the Unity Game Engine. It can be roughly divided into three parts. The first part is the general scene setup. We used assets from the Unity Asset Store to simulate a realistic home dining setting and texturing of assets for better contrast.

TABLE I
ATTRIBUTES USED BY HUMAN AND ROBOT AGENTS

Human Attributes		Robot Attributes	
Size	Weight	Size	Weight
Fragility	Sharpness	Fragility	Sharpness
Value	Handedness	Value	Flexibility
Permanent		Filled	Distance

The second part is the cluster of props, where we once again use assets from the Unity Asset Store to simulate objects that would be found in the real world. The scene is setup as shown in Fig. 5. We assign each prop a set of attributes, which are later used to differentiate the time and effort needed to move different objects. The attributes are exclusively accessed by our human agent; the robot agent does not have access to these attributes, and instead estimates them using our algorithm.

[1]DISCERN Code + Simulation: https://github.com/GOTWIC/DISCERN

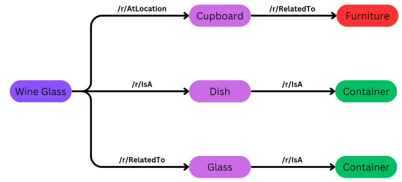

Fig. 3. The first target found is not necessarily the most correct category. Stopping at the first valid category can result in misclassification.

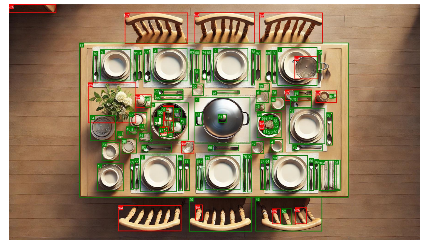
Fig. 4. A visual representation of our algorithm's output on a sample home dining table image (Input Image Courtesy of GPT4o).

They are similar to those we assign to various categories in our approach; see Table I for a full list.

Two notable attributes are the *handedness* and *permanent* attributes. *Handedness* is the number of hands (one/two) needed to clear a specific item. The *permanent* attribute is a boolean value representing whether or not an item is removable. Certain items, e.g. decorations have this value=1.

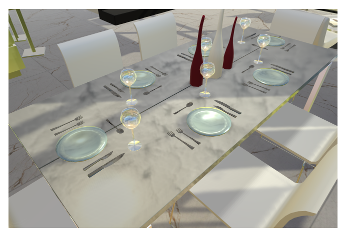
Fig. 5. Home Dining Table Simulated

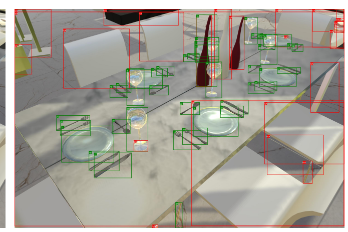
Fig. 6. DISCERN Output Visualized

In the third part, we include our human and robot agents. The human agent clears the table via an intuitive method by which humans normally clear tables - starting from the smallest items to the largest. Depending on the handedness of the next item, the human agent can pick up one or two items each time (for example, a fork and a spoon can be picked up together with one hand, but two pots cannot). In the simulation, we set the default time for the human agent to clear an item to be three seconds, corresponding to the approximate 6-foot distance between the home dining table and the kitchen. Then the default time is modified using the item's attributes, such as its weight and sharpness, using the values in Table II to simulate the extra time it would take a human to move such objects with greater care. The penalty is applied whenever the attribute goes above the activation threshold.

Unlike the human agent, the robot agent does not have internal access (through the Unity Engine's scripting capabilities) to know what objects are on the table, nor their assigned attributes, similar to the real world. The robot agent in the simulation is only able to "take a picture" of what it sees, and remove objects from the table. The picture is then processed

TABLE II
TIME PENALTIES FOR HUMAN AGENT BASED ON ATTRIBUTES OF ITEMS (PER 0.1 ATTRIBUTE INCREASE)

Attribute	Time Penalty (%)	Activation Threshold
Weight	13	0.3
Size	7	0.3
Sharpness	6	0.5
Fragility	10	0.7

through DISCERN, and the object priority order and location of each object are returned to the robot agent. However, the robot agent does not have a list of objects, so there is no way to directly corroborate the output of our algorithm (bounding boxes in 2D space) to the objects on the table (objects in 3D Space), which is similar to a real-world robot, as localizing objects that it detects is not an instant or obvious task.

Hence, the method by which the robot removes objects from the table is by shooting multiple raycasts within the bounding box returned by DISCERN, hopefully hitting the object we would like to remove. Failure to locate an object with the raycast is an intentional behavior meant to simulate the robot failing to pick up an object based on the image it is looking at. If the raycast successfully hits the object on the table, then it is then removed from the table. If a segmentation mask of the object is available (which DETIC supports), the robot agent will shoot raycasts inside the segmentation mask instead. Since the human agent has internal access to the list of props on the table, it knows when a prop has been removed by the robot, and updates its own priorities accordingly. If the robot agent misses its raycast, it skips that object and attempts to remove the next item instantly. The robot agent may miss its raycast if the object has already been removed by the human agent. Similar to the human agent, the robot has a removal time of three seconds. It can only remove one object at a time (and has no notion of handedness).

Additionally, if the robot attempts to pick up an object with a permanent value of 1 (i.e. a decoration or furniture, either intentionally or accidentally), then the object is not removed but the three-second cool-down is still in effect. In real life, if a robot tries to pick up an object it's not supposed to, it may need to be redirected towards something else.

The simulation ends when all objects have been removed, and the total execution time is displayed. The human and robot agents do not visually remove objects in our simulation - instead, the agents are model-less and the items removed by the agents are listed in the simulation's user interface. Physical models inside the simulation are purely for show.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Experimental Setup

For the simulation, we conduct five different tests.

1) Human Agent Only
2) Robot Agent Only (Random Object Ordering)
3) Human + Robot with Random Object Priority Ordering
4) Human + Robot with Algorithmic Object Priority Ordering, VLM Disabled
5) Human + Robot with Algorithmic Object Priority Ordering, VLM Enabled

The first two experiments seek to show comparative performances between the human and robot agents when working alone. While they do not prove or disprove the efficacy of our algorithm, they can be used to: A) make sure our simulation setup is reasonable, and B) compare the efficiency of human-robot collaboration vs a single party. Since the robot-only experiment may not detect every object, the simulation stops when the robot removes all the objects it detected.

The third experiment serves as the baseline to which we compare our algorithm. In the baseline, the robot agent takes an image, finds the location of objects in the image, and shoots raycasts at the objects to remove them, but does so in random order. The fourth and fifth experiments use DISCERN to prioritize objects and are compared to the baseline by measuring the total time taken for the collaborative task. All experiments are repeated five times with averages taken.

In addition to the main experiment to test our algorithm, we test different image models in our simulation. For each model, we measure the time it takes for the simulation to finish in both solo and collaborative tasks, as well as note down the number of penalties administered.

B. Results

Table III shows a comparison of the different image models. We see that both YOLOv5 and YOLO-WORLD have the lowest number of penalties. However, both YOLO Models miss more objects in the scene when compared to other models. Since there are 42 objects in the scene and each object has a fixed three-second removal time, removing all objects takes at least 126 seconds. As both YOLO models are under that threshold by a significant margin, their performance is not as desirable. Since the YOLO model is unable to detect larger objects, sparse detection means less clutter and thus fewer raycast misses, giving YOLO a lower penalty count.

DETR performs the worst out of the four models. While its detection capabilities are on par with DETIC, DETR's output results in the robot agent making significantly more mistakes than the other models due to the quality of its bounding boxes.

DETIC performs the best out of all the models. Since it can detect more objects than the YOLO models, it is more useful in collaboration with the human agent. Furthermore, due to DETIC's mask segmentation functionality, the robot agent is able to shoot raycasts with higher accuracy and less error margin in a more cluttered environment. Both of these contribute to DETIC having the fastest collaboration time.

Table IV shows the result of our five experiments. While the robot by itself takes 37% longer than the human robot, both agents working together (with the robot executing in random order) finish the task 65% faster than the human alone. However, using DISCERN, the collaborative task is 110% faster than the human alone. When comparing DISCERN against the baseline, DISCERN performs 27% faster.

While the VLM did not have a significant impact on the specific simulation in which we conducted our experiments,

TABLE III
COMPARISON OF IMAGE MODELS

Model	Robot Only Time (s)	Robot Only Penalties	Collaborative Time (s)	Collaborative Penalties
YOLOv5xu	109.1	2.8	51.1	2.2
YOLO-WORLD	108.6	2.8	51.3	2.8
DETR	155.9	12.4	79.5	13.4
DETIC	135.3	6.6	47.4	3

TABLE IV
SIMULATION RUNTIMES

Experiment	Avg. Time (s)	Human Ct.	Robot Ct.
Human Only	99.2	42	0
Robot Only	136.2	0	38
Collaborative (Random)	60.3	29	13
Collaborative (CSK)	47.7	30	12
Collaborative (CSK+VLM)	47.4	30	12

we note that in scenarios where there are many heavy items, the VLM will help the robot agent prioritize objects in the same category but with different attributes.

V. CONCLUSIONS AND FUTURE WORK

We propose an approach called *DISCERN* to *discern* robotics environments and help with effective task organization, by integrating an image detection model and a CSKB fine-tuned with a VLM, followed by adding custom priorities based on our human common sense. Through simulation experiments (on a home dining table), we demonstrate that DISCERN significantly enhances the efficiency of robot-assisted tasks in household environments. Our collaborative approach, leveraging both human and robotic agents, proves 27% faster than random object priority ordering, showcasing the potential of commonsense knowledge integration in improving task performance. Additionally, the inclusion of the vision language model further refines the robot's understanding and interaction with objects, leading to more accurate and context-aware task execution. These results underscore the value of combining AI and human factors in developing robots for complex tasks in dynamic environments [26], [27]. In the future, we plan to explore more advanced concepts such as learning without memorization [28], unlearning [29] in machines, and choosing data [30] for transfers needed in such tasks. They can be useful in our AI & robotics applications.

ACKNOWLEDGEMENTS

This work is supported by the NYU Dean's Undergraduate Research Fund under Grant 7669616542 and in part by the National Science Foundation under Grant CNS-2117308, Grant CMMI-2338767, and Grant CNS-2018575.

REFERENCES

[1] G. Charalambous, S. Fletcher, and P. Webb, "The development of a scale to evaluate trust in industrial human-robot collaboration," *Intl. J. of Social Robotics*, vol. 8, 04 2016.
[2] Y. Ye, H. You, and J. Du, "Improved trust in human-robot collaboration with chatgpt," *arXiv preprint arXiv:2304.12529*, 2023.
[3] F. Zeng, W. Gan, Y. Wang, N. Liu, and P. S. Yu, "Large language models for robotics: A survey," *arXiv preprint arXiv:2311.07226*, 2023.
[4] S. Razniewski, N. Tandon, and A. S. Varde, "Information to wisdom: Commonsense knowledge extraction and compilation," in *ACM WSDM*, pp. 1143–1146, 2021.
[5] N. Tandon, A. S. Varde, and G. de Melo, "Commonsense knowledge in machine intelligence," *SIGMOD Rec.*, vol. 46, no. 4, pp. 49–52, 2018.
[6] A. Mandlekar, F. Ramos, B. Boots, S. Savarese, L. Fei-Fei, A. Garg, and D. Fox, "Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data," *arXiv preprint arXiv:1911.05321*, 2020.
[7] A. Varde, E. Rundensteiner, G. Javidi, E. Sheybani, and J. Liang, "Learning the relative importance of features in image data," in *IEEE ICDE workshops*, pp. 237–244, 2007.
[8] D. Karthikeyan, A. S. Varde, and W. Wang, "Transfer learning for decision support in covid-19 detection from a few images in big data," in *IEEE Big Data*, pp. 4873–4881, 2020.
[9] O. Obidat, J. Parron, R. Li, J. Rodano, and W. Wang, "Development of a teaching-learning-prediction-collaboration model for human-robot collaborative tasks," in *2023 IEEE CYBER*, pp. 728–733, 2023.
[10] I. Jacoby, J. Parron, and W. Wang, "Understanding dynamic human intentions to enhance collaboration performance for human-robot partnerships," in *2023 IEEE MIT URTC*, pp. 1–6, 2023.
[11] A. Kaluarachchi, D. Roychoudhury, A. S. Varde, and G. Weikum, "Sitac: discovering semantically identical temporally altering concepts in text archives," in *ACM EDBT*, pp. 566–569, 2011.
[12] F. Ocker, J. Deigmöller, and J. Eggert, "Exploring large language models as a source of common-sense knowledge for robots," *arXiv preprint arXiv:2311.08412*, 2023.
[13] J.-P. Töberg, A.-C. Ngonga Ngomo, M. Beetz, and P. Cimiano, "Commonsense knowledge in cognitive robotics: a systematic literature review," *Frontiers in Robotics and AI*, vol. 11, 2024.
[14] C. Conti, A. Varde, and W. Wang, "Human-robot collaboration with commonsense reasoning in smart manufacturing contexts," *IEEE TASE*, vol. 19, pp. 1784–1797, 07 2022.
[15] I. Chernyavsky, A. S. Varde, and S. Razniewski, "Csk-detector: Commonsense in object detection," in *IEEE Big Data*, pp. 6609–6612, 2022.
[16] R. Hidalgo, J. Parron, A. Varde, and W. Wang, "Robo-csk-organizer: Commonsense knowledge to organize detected objects for multipurpose robots," in *IEMTRONICS, Springer*, pp. 654–668, 04 2024.
[17] M. J. Pawlish and A. S. Varde, "A decision support system for green data centers," in *ACM CIKM's PIKM workshop*, pp. 47–56, 2010.
[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.
[19] T. Cheng, L. Song, Y. Ge, W. Liu, X. Wang, and Y. Shan, "Yolo-world: Real-time open-vocabulary object detection," *arXiv preprint arXiv:2401.17270*, 2024.
[20] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," *arXiv preprint arXiv:2005.12872*, 2020.
[21] X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, and I. Misra, "Detecting twenty-thousand classes using image-level supervision," *arXiv preprint arXiv:2201.02605*, 2022.
[22] A. Gupta, P. Dollár, and R. Girshick, "Lvis: A dataset for large vocabulary instance segmentation," *arXiv:1908.03195*, 2019.
[23] D. B. Lenat, "Cyc: a large-scale investment in knowledge infrastructure," *Commun. ACM*, p. 33–38, nov 1995.
[24] P. Singh, T. Lin, E. T. Mueller, G. Lim, T. Perkins, and W. L. Zhu, "Open mind common sense: Knowledge acquisition from the general public," in *CoopIS and ODBASE 2002*, p. 1223–1237, 2002.
[25] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," *arXiv preprint arXiv:1612.03975*, 2018.
[26] C. Hannum, R. Li, and W. Wang, "A trust-assist framework for human–robot co-carry tasks," *Robotics*, vol. 12, no. 2, p. 30, 2023.
[27] J. Parron, T. T. Nguyen, and W. Wang, "Development of a multimodal trust database in human-robot collaborative contexts," in *2023 IEEE UEMCON*, pp. 0601–0605, 2023.
[28] V. Feldman, "Does learning require memorization? a short tale about a long tail," in *ACM SIGACT Symp. Theory Comp.*, pp. 954–959, 2020.
[29] L. Bourtoule, V. Chandrasekaran, C. A. Choquette-Choo, H. Jia, A. Travers, B. Zhang, D. Lie, and N. Papernot, "Machine unlearning," in *2021 IEEE Symp. Security & Privacy)*, pp. 141–159, 2021.
[30] N. M. Sepahvand, V. Dumoulin, E. Triantafillou, and G. K. Dziugaite, "Data selection for transfer unlearning," *arXiv preprint arXiv:2405.10425*, 2024.