Robotic Perception

Deep Learning End-to-End Strategies for Object Affordance in Robotic Manipulation

RO57010: Literature Research

Leonoor Verbaan

Robotic Perception

Deep Learning End-to-End Strategies for Object Affordance in Robotic Manipulation

by

Leonoor Verbaan

Student Name	Student Number
Leonoor Verbaan	5415721

Supervisor:	Dr. Ir. Yke Bauke Eisma
Daily supervisor:	Remco van Leeuwen
Project Duration:	September, 2023 - May, 2023
Faculty:	Faculty of 3mE, Delft

Cover:	https://theinnovator.news/2019-10-22_man-and-machine-885dbc02ddaa/
Style:	TU Delft Report Style, with modifications by Daan Zwaneveld

TUDelft

Contents

Abstract

keywords
Object affordance, end-to-end systems, deep learning models, voltage rack, robotic arm, Microsoft HoloLens

This study investigates the optimal deep learning strategies for robotic manipulation in electrical engineering tasks, such as fuse switching in voltage racks, with a focus on alleviating the workload of workers and enhancing task execution efficiency. Central to this literature research is the exploration of state-of-the-art deep learning models for implementing object affordance in robotic manipulation. By integrating a Microsoft HoloLens and a robotic manipulator, the research addresses the critical need for robust, safe, and adaptable systems capable of performing complex tasks like fuse switching while keeping the operator in the loop. The methodology involves a dual analysis: a conceptual analysis of object affordance approaches and a meta-analysis of the latest deep learning models for end-to-end object affordance. The findings reveal that for simpler tasks, the Grasping Siamese Mask R-CNN (GSMR-CNN) is optimal, while for more advanced tasks, a Large Language Model (LLM) with a Convolutional Neural Network (CNN) backbone can be used to generate multiple action affordances. Moreover, end-to-end auto encoders or deep reinforcement learning methods are effective as well but increase complexity in the system. These models offer the necessary generalizability, adaptability, and accuracy for task execution. Concluding, this literature research proposes an end-to-end object affordance approach by selecting deep learning models based on the analyses. It recommends further research to refine these models for specific applications, and improve their generalizability, thereby enhancing safety and efficiency in operations such as voltage rack maintenance.

Leonoor Verbaan
Delft, December 2023

1 Introduction

1.1. Background

In recent years, wearable augmented reality (AR) devices have gained significant attention in research for their ability to intelligently capture and convey information, especially to other technology such as robots. The integration of wearable augmented reality devices with industrial robots, especially robotic manipulators, is showing considerable potential. This advancement is leading to the implementation of new, state-of-the-art models and planners that are combined with these devices [1, 2, 3, 4]. This technology is particularly relevant in high-risk, technical fields, where precision and safety are essential. Especially in the electrical industry with companies such as Alliander [5] many processes can be automated and improved for safety. One of these specific tasks in the electrical industry that needs to be automated and improved for safety is the switching of fuses in voltage racks [6, 7]. The availability of engineers for this type of work is limited, leading to inefficiencies in task execution. This scarcity is largely due to the challenges faced by training programs in attracting engineers, as the work is not perceived as attractive, is physically demanding, and offers only moderate financial compensation.

Introducing innovative solutions, such as the use of wearable augmented reality devices – for instance, the Microsoft HoloLens [8] – combined with a robotic manipulator, could offer significant improvements. This combination has the potential to not only improve its appeal but also alleviate the burden on electrical engineers and enhance the efficiency of task execution. Moreover, this approach leverages advanced algorithms to enhance both safety and operational efficiency [9, 10]. Operators, through the AR interface, can remotely direct these manipulators, benefiting from a more intuitive and interactive way of managing these risky tasks [11]. These vision based robotic applications are commonly recognized as visuo-motor systems. These systems use deep-learning computer vision models that enhance their reliability and robustness [12, 13, 14].

1.2. Visuo-motor robot systems

Developing an effective visuo-motor system for robotic manipulators involves designing a perception and control system capable of interpreting environmental cues and executing precise actions. Central to these visuo-motor systems is the concept of **object affordance**, which enables robots to recognize and manipulate objects based on their functional regions and inherent properties. In robotic manipulation, understanding affordance is crucial as it enables robots to interact with and manipulate objects that they have not encountered before. Approaches to make such visuo-motor systems for object affordance vary from multi-stage methods, where separate stages train for different aspects of the task, to end-to-end systems where training is unified [15, 16]. Multi-stage methods have proven generalizability and effective real-world performance but face issues with cumulative errors across stages. In contrast, end-to-end approaches train policies in a unified manner, enhancing system adaptability and task performance. These approaches balance learning object affordance with decision-making strategies, leading to more effective systems [17, 18]. *Figure 1.1* displays a general structure of such an **end-to-end visuo-motor system** for object affordance.

Consequently, in the development of end-to-end visuo-motor systems for robotic manipulation, the role of **deep learning models** is essential for enhancing robot object affordance learning. These models, which include various neural network architectures, excel in interpreting detailed environmental data. The effective combination of these deep learning models forms a solid base for visuo-motor systems. Integrating the perception skills of models such as Neural Networks with the dynamic decision-making abilities of Reinforcement Learning models elevates the efficiency of these systems in complex tasks [19, 20].

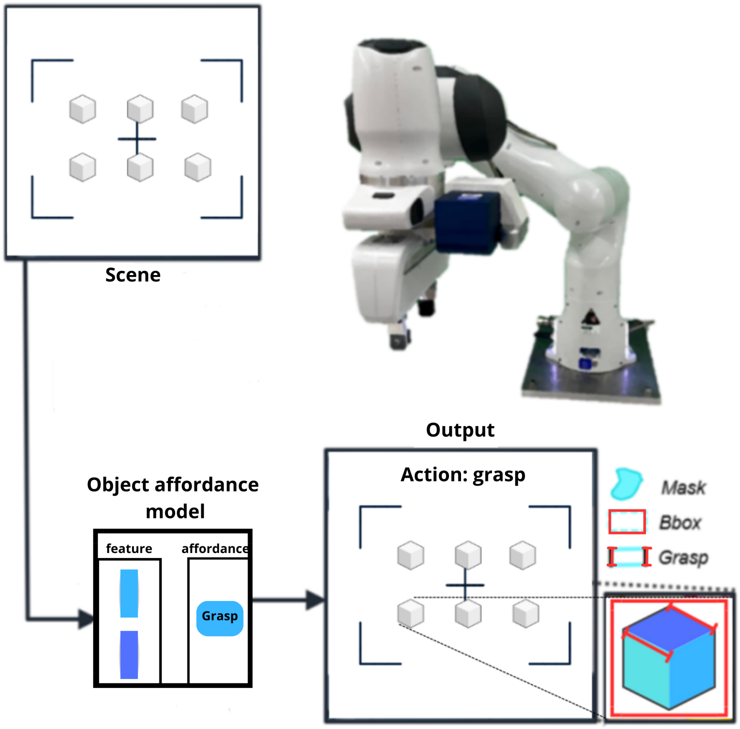

Figure 1.1: This Figure displays a general structure of an end-to-end visuo-motor system. The scene is presented from the camera on the Franka robot manipulator. The image data from the camera is the input for the end-to-end object affordance model that extracts object affordance from feature vectors. The output shows these features (for example segmentation mask, detection bbox and grasp location) and converts this visual input to an action. The cube is recognized as a graspable object.

1.3. Research gap and General Problem Statement

The research focuses on end-to-end visuo-motor systems for robotic manipulators in the context of object affordance [21]. The challenge lies in creating an end-to-end system that interprets environmental cues for tasks like operating voltage racks. While multi-stage methods are common, their error accumulation and lack of adaptability in varied environments highlight the need for more robust end-to-end systems. The proposed system aims not only to address these challenges but also to enhance task performance and efficiency. This enhancement is achieved by simplifying the object affordance learning pipeline and supporting electrical engineers through the operation process with a user-friendly Microsoft HoloLens interface. Thus, this literature research seeks to explore such systems, focusing on integrating perceptual affordance with decision-making processes to improve task performance, appeal and safety [21].

Therefore, this literature research paper aims to answer the following **research question**: *What is the most effective deep learning end-to-end strategy for object affordance in robotic manipulation, such as grasping, pulling and pushing, within the current state-of-the-art?*

In the upcoming chapters, this research delves into a comprehensive analysis of current research within this field. Initially, the research methodology and analytical framework are outlined in *Chapter 2*. This is followed by an exploration of various object affordance learning methods in *Chapter 3*. *Chapter 4* then shifts focus to compare different deep learning models based on object affordance. *Chapter 5* brings it all together, by providing a discussion and the conclusions drawn from our analyses to answer the stated research question. It also presents the future work and a research framework developed from this literature research, setting the direction for the remainder of the thesis. Finally, *Chapter 6* concludes this literature research.

2 Methods

2.1. System baselines

The experimental setup for our final framework features a Franka Emika Research 3 manipulator with RealSense D435i vision as the primary system for evaluating the diverse end-to-end object affordance systems for manipulation. The system will use a Microsoft's Hololens 2, an augmented reality device, as a user interface. This allows operators to control the manipulator and experience the end-to-end system through a digital interface. To address the research question effectively and evaluate the end-to-end system, some aspects need to be broken down and specified. Specifically, the task of fuse switching involves four key steps that can be defined with existing motion primitives (grasping, pulling, pushing): **(1) opening the fuse case door** (grasping, pulling), **(2) grasping the fuse** (grasping), **(3) removing the fuse** (grasping, pulling), and **(4) closing the fuse case door** (grasping, pushing). *Figure 2.1* displays these tasks with motion primitives. This literature research primarily concentrates on the application of deep learning strategies in robotic manipulation, with a focus on tasks like grasping, pulling, and pushing.

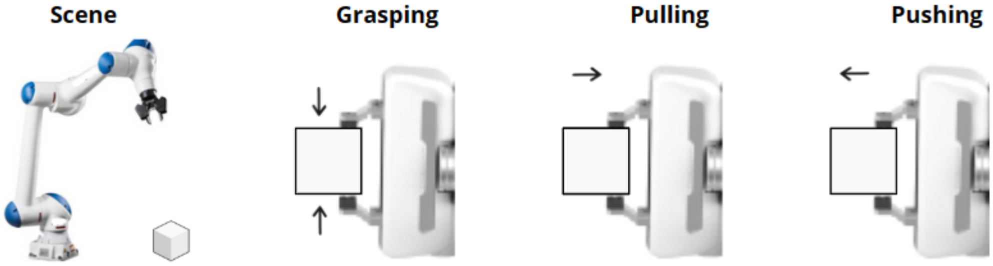

Figure 2.1: This figure suffices as a visualization of the motion primitives that are used to perform the task of fuse switching in voltage racks.

The approach of the following research is to break down the project in designing a system that is able to perform a **simple task**, such as any motion primitive separately. Thereafter, the system will be extended to perform the **advanced task** by implementing more of these motion primitives in sequence so the system can perform the four key steps. Therefore a proposal will be made that displays a system that generalize to both simple and advanced tasks (*Chapter 5*). Preliminary work for the implementation with the Microsoft HoloLens and the end-to-end system can be seen in the *Appendix A*. This includes a vision app that labels and collects data which can be used as an input for the end-to-end system. A complete representation of the system structure with Microsoft HoloLens implementation is displayed in *Figure 2.2*. The topic analysed in the literature research is depicted in the red box.

2.2. System and literature criteria

To effectively analyze the results, it is essential to establish the main criteria for the complete system as illustrated in *Figure 2.2*. The system must meet the following requirements:

*The system needs to be **adaptable** for both simple to advanced tasks* **[Criterion 1]**

*The system needs to be **generalizable** for both simple to advanced tasks* **[Criterion 2]**

*The system needs to be able to detect **object affordance** and learn **environmental representations*** in real-time. **[Criterion 3]**

*For the simple task the system has to be able to perform the **grasping** motion primitive, and for the advanced task the system has to be able to perform grasping in combination with **pulling or pushing.*** **[Criterion 4]**

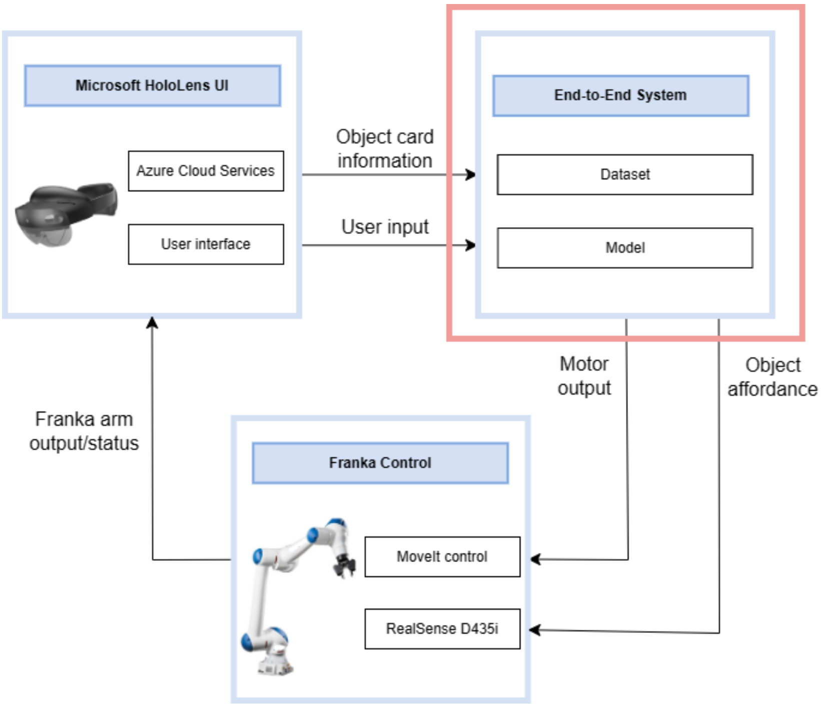

Figure 2.2: This Figure displays the complete system structure. In this system the Microsoft HoloLens functions as a UI and connection to the cloud services, then the End-to-End system produces the object affordance for the Franka arm control. This system consists of a feedback loop to the Microsoft HoloLens UI to visualize the motor and visual output. The topic of the literature research is depicted in the red box.

In order to find relevant literature, the research question is divided into the following sub-questions. The associated chapter relevant to the sub-question are marked in bold:

How can a robot learn object affordance in robotic manipulation? **[Chapter 3]**

What deep learning models can be used for implementing object affordance in robots? **[Chapter 4]**

How do these approaches and models apply to simple and specific advanced tasks? **[Chapter 5]**

What is the best model to use to effectively implement object affordance for grasping, pulling and pushing? **[Chapter 5]**

What kind of end-to-end system framework can be designed based on these approached and models? **[Chapter 5]**

2.3. Analysis methods

As a means to answer these aforementioned questions this literature research performs a conceptual analysis for defining proper object affordance learning methods (*Chapter 3*). Consequently, a meta-analysis is performed on deep learning models based on object affordance (*Chapter 4*). A more detailed visualization of this approach is shown in *Figure 2.3*. With the results of both analyses conclusions are drawn and a research framework is set up of an end-to-end object affordance system in the robot arm in combination with the Microsoft HoloLens (*Chapter 5*).

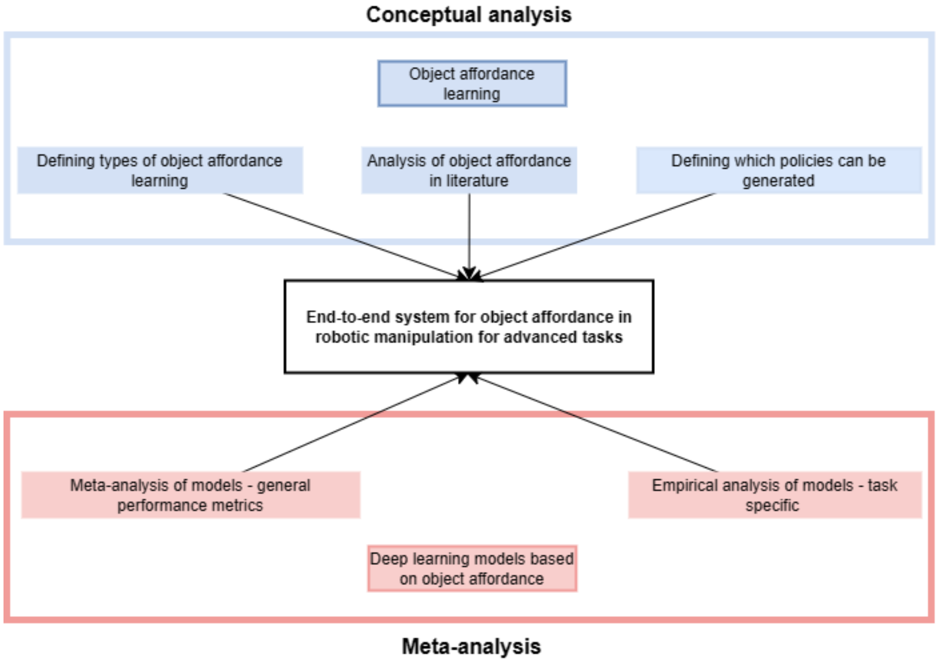
Conceptual analysis
Object affordance learning
Defining types of object affordance learning
Analysis of object affordance in literature
Defining which policies can be generated
End-to-end system for object affordance in robotic manipulation for advanced tasks
Meta-analysis of models - general performance metrics
Empirical analysis of models - task specific
Deep learning models based on object affordance
Meta-analysis

Figure 2.3: This literature research will approach this research question with two research methods: a conceptual analysis of object affordance learning and an meta-analysis of state-of-the-art deep learning models based on object affordance. In this Figure the analyses are broken down in the topics that are being covered.

3

Object affordance learning

3.1. The use of affordance in robotic manipulation tasks

The main theme in robotic affordance learning consist of the questions how and with what the robot is interacting with. The concept of affordance is essential for achieving advanced manipulation and interpretation similar to human capabilities. Object affordance learning for an end-to-end system can be classified on what kind of policy it can generate. The approaches that focus on grasping new objects are called **grasping affordance**. For instance, algorithms have been developed to identify proper grasp locations from images. These algorithms use logistic regression or deep learning approaches to detect the location and orientation suitable for grasping an object. This understanding of grasping affordance has been applied to tasks such as unloading objects from a dishwasher [22, 23, 24, 25]. The practice of understanding the relationship between an object's physical features and its functions for different activities is known as **task-oriented affordance**. It emphasizes the connection between an object's physical attributes and the specific functions these attributes enable, thereby focusing on the practical applications facilitated by these features. This understanding of task-oriented affordance has been applied in tasks such as tool clustering or identifying graspable parts of an object [26, 27, 28, 29, 30, 31, 32, 17]. The practice of defining affordance for manipulation using methods like ontology-based approaches is defined as **manipulation affordance**, which involves manually defining affordances based on the perception of the environment. This approach is used to determine the appropriate affordance for advanced or specific robot tasks like pushing a refrigerator door [33, 34, 35, 36]. Additionally, these affordances are used to plan task motion sequences while taking into account environmental factors such as obstacles and supporting features. These affordances are segmented for illustrating the different ways in which agents can interact with object based on the affordances they perceive. However, it is important to note that in practice, these categories can overlap and combined for more accurate and efficient task execution. For example an agent can make a visual understanding of what it is perceiving with geometric and semantic affordance properties, and use these visual affordances together with interacting with the environment to learn a certain action.

The following sections will address several approaches to teach object affordance to agents. Just as the task categories, these teaching approaches can also be combined for a working object affordance end-to-end system for manipulation.

3.2. Learning object affordance from trained properties

3.2.1. Inferring geometric and physical properties

An agent can gain insight of object affordance by studying the **geometric and physical properties** of both the object and its environment. This process involves recognizing and characterizing object features based on various aspects such as shape, structure, texture, material, surface, and contours [36, 37, 38, 39, 40].

Wu et al., for example, used a model that learns the distribution of complex 3D shapes across different object categories and arbitrary poses. From this distribution of shapes the agent discovers hierarchical compositional part representation automatically [38]. Sundermeyer et al., introduced a system for 3D orientation learning for 6D object detection where it provides an implicit representation of object orientations. This implicit representation can be used to visually understand shape and the scene for the agent [39]. Wen et al., proposed a model that robustly accumulates information into a consistent 3D representation capturing both geometry and appearance. The geometry includes pose and orientation and the appearance contains the visual texture of the object. The geometric features are found by segmenting the object and feature matching. The geometric features are then compared with a set of geometric features from a memory pool. So no prior knowledge of the object or interaction agent is needed to reconstruct the whole object in the scene [40]. All in all, these types of affordance learning focus more on obtaining information more directly useful for how to act.

3.2.2. Predicting semantic representations

Affordance can also be learned by an agent through **semantic properties** of the object and environment. In this context, object features are identified and characterized by methods such as category label predictions, semantic keypoints, part segmentations, relations with obstacles or objects, surfaces in the environment, etc [34, 32, 36, 33, 35, 41, 23, 42, 43]. The use of semantic representation prediction focuses on the concept of the agent understanding given perceptual input. For example, the agent can infer where buttons, handles, switches and levers are. For proper object inference these semantic representation predictions are an important element in deciding what kind of action needs to be done and how it should be performed [32].

Hu et al. introduced a system that can predict the functionality of a 3D object given in isolation. These functionalities are based on man-made objects and they are characterized by their human-object interaction and inter-object interactions. Consider scenarios such as a person sitting on a couch or a coffee cup resting on a table. The objects create a scene context that allows for the study of the central object's functionalities. The functionalities are defined by the interactions between the central object and its surrounding objects. This method learns object functionalities via a scene dataset which contains a set of interaction contexts [33]. Lee et al. proposed a model that uses probabilistic max-pooling to learn useful visual features such as object parts, edges and whole objects from scenes and objects themselves. Accordingly, hierarchical representations are learned from the input scene and object RGB images. This method learns object parts and hierarchical representations from unlabeled RGB images of objects and natural scenes [35]. Do et al. proposes a model that learns affordances based on part segmentations and category labels combined where each pixel of an object is assigned an affordance label based on its functionality. For example, a pixel might be labeled as part of a 'grasp' area on a mug. The network uses these labels in combination with RGB images to learn visual cues associated with different affordances. This method learns these part segmentation representations from a labeled RGB image dataset, where the affordances are labeled at the pixel level [41]. Finally, Levine et al. used an approach where grasping affordance is learned with the spatial relationship between the gripper and object in the scene. This system replicates hand-eye coordination, allowing the model to operate without needing exact calibration between the camera and the end-effector. Instead, it relies on visual cues to ascertain the spatial relationship between the gripper and objects in the scene that can be grasped. This method learns hand-eye coordination based on solely monocular camera images [23]. In short, all these type of affordance learning focus more on obtaining information on what the agent is acting on and having a general understanding of the scene.

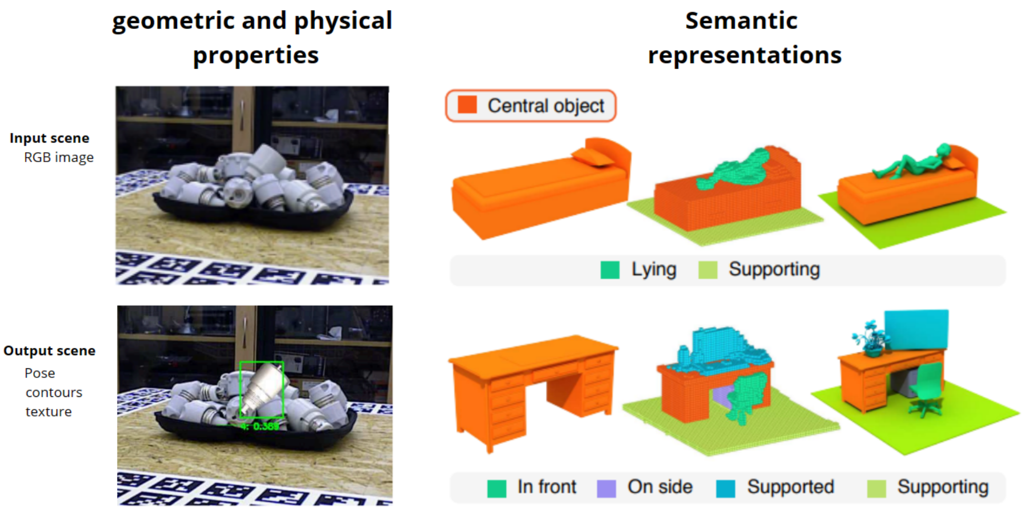

Figure 3.1: This Figure displays two approaches to learning object affordance from trained properties. (left) This is an example of an agent learning **geometric and physical properties** with RGB image data [38]. (right) This is an example of an agent learning **semantic representations** from a scene where one central object is taken as an input [33].

3.3. Learning object affordance by interactions and observations

3.3.1. Predicting by interaction

In addition to learning semantic and geometric properties, an agent can also learn affordance by interacting with the environment during execution of an end-to-end task. These learned affordance properties can be referred to as **interaction representations**. For example, Affordance detection often involves learning from human interaction. This means that studies have predicted types of affordances like pushable, drawable, or graspable by learning from image attributes through human demonstrations. These attributes can be learned through active exploration, simulation and real-world interactions [31, 32, 37, 24, 17, 44, 29, 45].

Geng et al. addressed the challenge of training a policy to manipulate 3D objects with diverse shapes, functionalities, and semantic affordances using reinforcement learning. This system learns object affordance by using contact information generated during reinforcement learning training to predict contact maps, which then guide the learning of appropriate interactions with objects. This means that the manipulation module of this system continuously collects data and the visual affordance module measures the likelihood of contact between agent and object to determine affordances. These manipulation affordances are then in real-time converted to an affordance map that the agent uses to predict its next action [17]. Gu et al. had a similar approach where object affordance is learned based on contact information during reinforcement learning in an asynchronous updating mechanism. The system develops its policy from the ground up by utilizing the data (observations, actions, and rewards) gathered by the robots, which is then used to update the policy network. The network learns which actions are more likely to be successful in achieving the task objectives, effectively learning the affordances of objects in its environment. The learner thread updates the policy based on the experiences collected by multiple robots, where this parallelization accelerates the learning process [44]. Another way for the agent to learn object affordance by interaction was introduced by Borja-Diaz et al. This studies used a learning from demonstration approach where a presumed expert teaches the agent to mimic policies from presumed expert demonstrations. So this system uses self-supervised visual affordance models derived from human teleoperated play data to inform and enhance policy learning and motion planning in the agent. This model can transform images into binary segmentation maps indicating regions that afford interaction and estimate 2D pixel coordinates of affordance region centers. This learned affordance information is then embedded within a reinforcement learning policy, guiding the robot to interact with objects [24]. Other studies such as Wu et al. and Mo et al. proposed a model that can learn manipulation affordance in simulation. Wu et al introduce a model that uses a reinforcement learning policy to explore various action trajectories within the simulation. It collects data from these interactions, which is then used to train the perception system. The perception system learns manipulation affordance, by observing the outcomes of these simulated interactions [31]. Mo et al. proposed a model that focuses on predicting per-pixel actionable information for manipulating 3D articulated objects, mapping pixels to actions using a learning-from-interaction approach. The system uses a hybrid data sampling strategy, starting with offline random action trajectories and then employing online interaction data points based on the network's predictions. This process allows the system to learn the manipulation affordances of different parts of an object by observing the results of its own actions within the simulation [32].

3.3.2. Predicting by observation

Another unique way to teach agents about possible actions through object affordance is by having them watch videos of how other agents interact with objects or the environment. These learned affordance properties can be referred to as **observational representations**. This method has been effective in teaching agents policies such as where to grasp or touch. The agent learns which areas are the most interacted with. Here object features are described through for example passive demonstration, amount of interactions, interaction surface segmentation, etc [37, 32, 30, 22, 46].

For example Thermos et al. developed a system that exploits the spatio-temporal nature of human-object interaction for affordance segmentation with only a small amount of supervision. In particular, an auto encoder is designed that is trained using ground-truth labels of only the last frame of the sequence, and is able to infer pixel-wise affordance labels in both RGB-D videos and static images. A soft attention component in the model extract is therefore able to extract interaction hotspots [30] [46]. Another system from Holomjova et al. used a deep learning model to identify target objects and determine grasp affordance accordingly. The system learns grasp affordance by observing multi-object scenes and learning from examples given in an initial dataset by extracting patterns corresponding to effective grasps. The model allows the agent to recognize and generalize to new object categories not seen during training because of its one-shot learning capability. This grasp affordance is learned from annotated (segmentation map and multiple hand-annotated grasp) RGB images of multi-object scenes with different object classes [22].

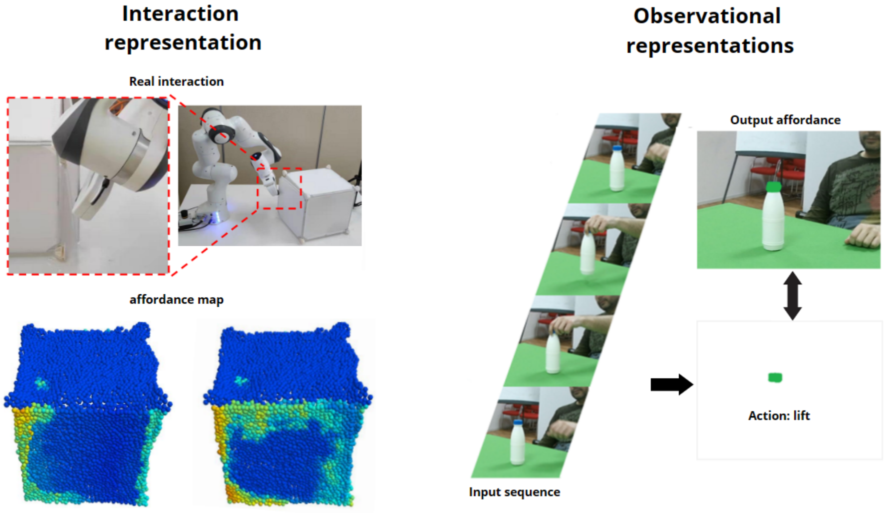

Figure 3.2: This Figure displays two approaches to learning object affordance from interactions and observations. (left) This is an example of an agent learning **interaction representations** that makes an affordance map based on the agent's interaction with the environment [29]. (right) This is an example of an agent learning **observational representations** where a video sequence is taken as an input to learn affordance on parts of an object [41].

The discussed approaches for learning object affordance can be summarized in the categories discussed in the sections of this Chapter. An overview of this conceptual analysis is displayed in *Table 3.1*. In this literature the focus is set on analysing prediction of policies, trained properties, and interactions and observations.

Object affordance learning		
Analysis	***Types***	***Literature***
Policies	Grasping affordance	[22, 23, 24, 25].
	Task-oriented affordance	[26, 27, 28, 29, 30, 31, 32, 17]
	Manipulation affordance	[33, 34, 35, 36]
Trained properties	Geometric and physical representations	[36, 37, 38, 39, 40]
	Semantic representations	[34, 32, 36, 33, 35, 41, 23]
Interactions and observations	Interaction representations	[31, 32, 37, 24, 17, 44, 29]
	Observational representations	[37, 32, 30, 22, 46]

Table 3.1: This Table displays the conceptual analysis and overview of object affordance learning literature.

Next to the overview the following *Table 3.2* presents the conceptual analysis of these object affordance learning properties. The advantages, disadvantages and input requirements are listed for each property.

Object affordance learning properties			
Representation	***Advantages***	***Disadvantages***	***Input requirements***
Geometric and physical	Comprehensive scene understanding. Automatic compositional representations. Easy to integrate with other representations.	Dependent on segmentation/feature matching. Increases complexity.	RGB-(D) data, 3D shape datasets.
Semantic	Comprehensive scene understanding. Automatic compositional representations. Pixel-level affordance learning.	Dependency of scene context. Difficulty in learning from unstructured data	RGB-(D) data, semantic labels.
Interaction	Real-time learning Simulation based exploration.	Dependence on quality of interaction data. Complex generalizability.	Contact information, teleoperated data, simulation.
Observational	Beneficial for multi-object scenes. Generalizable.	Dependence on quality of demonstration data.	RGB-(D) data, affordance labels.

Table 3.2: This Table displays the conceptual analysis with advantages, disadvantages, and input requirements for each object affordance learning type.

4

Deep Learning models based on object affordance

4.1. Overview of models

Deep learning models have significantly advanced the field of robotic manipulation, particularly in end-to-end object affordance learning. This chapter provides an overview of the state-of-the-art deep learning models that have advanced the field of robotic manipulation through object affordance learning. These models range from **Convolutional Neural Networks (CNNs)**, renowned for processing visual data, to sophisticated **Deep Reinforcement Learning (DRL)** models capable of learning complex policies for autonomous agents. Also included are **Generative Models (GMs)** that synthesize novel images, shapes of objects, and text, and other innovative neural network models that enhance the capabilities of end-to-end object affordance systems. The range of these kind of models relevant to this study are presented in the following overview:

- **Convolutional Neural Networks (CNNs)**
  CNNs are specialized for processing visual data and learning spatial relationships in object affordance tasks in an end-to-end manner [34, 32, 23, 38, 35, 33, 41].
- **Deep Reinforcement Learning Models (DRLs)**
  DRL models are known for their effectiveness in learning complex policies for agents and their flexibility from learning with not much prior knowledge [47, 19, 24, 31, 17].
- **Generative Models (GMs)**
  These models can imagine or synthesize novel shapes or poses of objects with desired affordances. Generative models can also learn relevant object affordances by combining generative modeling with a performance predictor, or by generating coherent and context awareness in a scene with text [39, 46, 48, 49, 50, 51].
- **Other Neural Network Models**
  These models are other more specialized models that are less common in the literature, however they still show interesting approaches such as using a knowledge base or attention layers [22, 40, 36].

These models demonstrate various approaches used in robotic manipulation, relevant to this literature research. They differ in how they determine object affordance for the agent. Some models, like CNNs, focus on identifying and extracting visual properties, while others, such as DRLs and GMs, link these properties to actions. Often elements of these models are combined to create an end-to-end object affordance system. The previous chapter analysed potential affordance properties for such systems, while this chapter examines which models are best for extracting these properties. The following sections will provide a more detailed examination of each model's distinct characteristics and uses.

4.1.1. CNN

Convolutional Neural Networks (CNNs) are suitable for extracting semantic and geometric features in a scene due to its ability to process and analyze visual data effectively, capturing the spatial relationships between objects and learning patterns indicative of various affordances from the training data. These networks are not only adept at extracting features with high efficiency but also excel in handling high-dimensional data, facilitating transfer learning, ensuring spatial invariance, and supporting real-time processing. This is why CNN structures are commonly deployed in end-to-end systems since they facilitate end-to-end learning, where raw input data can be directly mapped to outputs (like affordance labels) without needing manual feature engineering [34, 32, 23, 29].

Another type of CNN, referred to as the **Convolutional Deep Belief Networks (CDBNs)**, is an algorithm for unsupervised probabilistic deep learning. The CDBN is adept at handling high-dimensional voxel data (3D pixel data), making it suitable for detailed 3D shape analysis and recognition. It can learn the joint probability distribution of voxel data and object category labels, which is important for understanding complex 3D structures and their affordances. The CDBN offers several benefits for learning object affordances such as: learning complex spatial relationships, capability of handling high dimensional data, and it can learn deep hierarchical features with high and low-level representations [38, 35]. Additionally, **Deep Convolutional Neural Networks (DCNNs)** are predominantly used for pattern recognition in images and videos. The DCNs are suitable for object affordance as they can effectively process 3D object data, predict functionalities by learning from scene datasets, and generate interaction contexts that visually represent these functionalities. The use of a generative approach allows the model to create diverse scenes demonstrating the object's potential uses. Examples of modules in DCNs are a functional similarity module, generative module or a segmentation module. Advantages of using DCNs are that it can give very detailed functionality predictions, provide contextual representations of the scene, and perform segmentation and classification. [33, 41].

4.1.2. Deep Reinforcement Learning Models

Deep Reinforcement Learning Models (DRLs) mostly facilitate the agent to action-provoking decision making through a trial-and-error approach to achieve the optimal algorithmic model for a situation. Generally, DRL models have an integration of a reinforcement learning module which focuses on improving the manipulation capabilities of the agent with a reward function. This reinforcement learning module then works together with a visual affordance module to form a simplified end-to-end system which is self-supervised and generalizable [24, 17, 29, 52].

Deep Q Networks (DQNs) form another distinct category within DRL models. A DQN is a reinforcement learning algorithm designed for environments with discrete action spaces. It combines Q-learning with deep neural networks to handle complex, high-dimensional input spaces, such as visual input. In robotic manipulation, a DQN can be utilized for learning affordances like grasping or picking up objects, where the actions are discrete (for example, different ways to grasp an object). It helps the robot to understand which actions will maximize the potential of successfully manipulating an object based on its affordance properties. DQN is beneficial in scenarios where an agent needs to make decisions from a set of discrete actions. It can process and integrate complex visual data to make decisions. Its ability to learn from trial and error helps the agents to adapt to new objects and scenarios, improving their interaction capabilities over time. This means that these models generally don't need a lot of prior knowledge to learn a policy (self-supervised), allowing for more flexible and adaptable learning [47, 19, 27, 53]. Furthermore, **Deep Deterministic Policy Gradient models (DDPGs)** is a type of DRL model that simultaneously learn a Q-function and a policy. DDPG is a model-free, off-policy actor-critic algorithm, suitable for environments with continuous action spaces. It combines the benefits of DQN with the advantages of policy gradient methods, allowing for the handling of more complex, nuanced actions. In the context of affordance learning, DDPG is particularly useful for tasks requiring smooth, precise control, such as gently manipulating fragile objects or performing tasks that involve fine motor skills. DDPG excels in environments where the action space is continuous and high-dimensional. It allows robots to learn a wide range of behaviors and to execute these behaviors with a high degree of precision and fluidity [44, 19, 31].

4.1.3. Generative Models

Generative models (GMs) are a type of model that captures the data distribution, indicating the probability of specific examples. They are capable of creating various actions or scene representations, which assist an agent in learning about object affordances [39, 46, 50, 49, 51]

For instance, an **Augmented Autoencoder (AAE)** is a type of GM that is trained using Domain Randomization on synthetic views of 3D models, making it effective in handling object and view symmetries and generalizing to various environments. This approach is particularly suitable for 3D orientation estimation as it allows the model to implicitly learn representations of object orientations, avoiding the need for explicit mapping from images to object poses. The AAE offers several advantages in learning object affordance properties such as: it learns representations that are invariant to the significant differences between synthetic and real world data , it considers self-supervised learning, and handles symmetrical objects [39]. Furthermore, **Encoder Structures** are other types of GMs that have the ability to train a single end-to-end model directly on source and target input. An example of an encoder structure is the demo2Vec model which integrates a demonstration encoder and an affordance predictor [46]. It processes demonstration videos into low-dimensional vectors, capturing key details about human actions and object appearances. These vectors enable the model to predict interaction regions and action labels for target images, effectively translating complex human-object interactions from videos into action affordance labels. This model excels in generalizing learned affordances to new, unseen objects [46]. Furthermore, One of the most recent end-to-end GMs used for generating semantic knowledge and even robotic manipulation are the **Large Language Models (LLMs)**. For example the research of wake et al explores the use of OpenAI's ChatGPT for converting natural language instructions into executable robot actions [54]. The study proposes customized input prompts for ChatGPT that can be integrated with robot execution systems or visual recognition programs, adapt to various environments, and create action sequences [50]. This implementation is capable of not only generating robot actions but also comprehending and interpreting the context of the scene. The use of LLMs enhance adaptability and generalizability into the systems. Moreover, this approach ensures that the human is in the loop of operations ensuring task efficiency [49, 51].

4.1.4. Other Neural Network Models

Other models that are specialized for object affordance are **Grasping Siamese Mask R-CNN (GSMR-CNN)**, **Neural Object Fields (NOFs)**, and **Hierarchical Networks (HNs)**. The GSMR-CNN model is an extension of the Siamese Mask R-CNN, combining the components of Siamese Neural Networks (SNNs) and Mask R-CNN. It includes an additional branch for grasp detection in parallel to the original object detection head branches. This design allows for simultaneous identification of target objects with suitable grasps in a scene [22]. The GSMR-CNN model offers several benefits for learning object affordances, particularly for grasping such as: generalization to new object categories, efficient architecture due to fewer model parameters, and simultaneous object identification and grasp prediction. Moreover, the NOF is key for learning multi-view consistent 3D shape and appearance, which is crucial for understanding the full geometry of an object, particularly when part of it may be occluded or not immediately apparent from a single viewpoint. These kind of models are generalizable, adaptable, and have a dynamic learning process [40]. Lastly, HNs are iterative algorithms for creating networks which are able to reproduce the unique properties of a topology. An example of a HN model is **Parallel Deep Learning with Suggestive Activation (PDLSA)**. This model incorporates several brain operating principles derived from neuroscience and psychophysical studies. PDLSA is organized as a hierarchical network, combining hand-coded parallel levels (each corresponding to a unimodal feature type) with automatically learned serial levels through deep learning. PDLSA focuses on global affordance features based on shape, color, local texture, and semantic contexts. These kind of hierarchical are close to cognitive models which have the benefits of being robust, efficient in object categorization, and contain suggestive activation. Suggestive activation is a cognitive feature that involves feedback loops, where units completing computations faster can influence the processing of other units [36].

4.2. Analysis of models

In this section the aforementioned models and their object affordance strategies are compared on: **how they obtain affordance** (annotation, demonstration, simulation or other), **what kind of input information they need** (RGB-(D), depth, pointcloud or other), and **How these models are verified** (simulation, real-life or model testing). Next to these metrics the category of model is defined together with what type of affordance is produced with that model. **C** defines the scene context (such as trained features) and **P** defines the produced policies from the affordance features. The results of this analysis are displayed in *Table 4.1*. In order to address the research question appropriately and build an object affordance end-to-end system these metrics per system are objectively analysed and compared (*Chapter 5*).

Table 4.1: This is an analysis of the literature regarding CNNs, Deep Reinforcement Learning models, Generative models and other Neural Network models.

→		***How to obtain affordance***				***Input information***				***Verification***			
Models ***Papers***	Category	Annotation	Demonstration	Simulation	Other	RGB-(D)	Depth	Pointcloud	Other	Simulation	Real-life	Model testing	Affordance type
Moldovan et al. [34]	CNN			✓	Exploration	✓				✓	✓		P
Mo et al. [32]	CNN			✓		✓		✓		✓			P
Levine et al. [23]	CNN	✓	✓			✓					✓		P
Wu et al. [38]	CDBN	✓					✓					✓	C
Lee et al. [35]	CDBN	✓			Hierarchical	✓						✓	C
Hu et al. [33]	DCN	✓							3D-voxels			✓	C
Do et al. [41]	DCN	✓				✓						✓	C
Borja-Diaz et al. [24]	DRL		✓			✓				✓	✓		P
Wang et al. [29]	DRL			✓	Few-shot	✓		✓		✓	✓		P
Geng et al. [17]	DRL		✓	✓				✓		✓	✓		P
Zhang et al. [47]	DQN			✓		✓				✓			P
Lillicrap et al. [27]	DQN			✓		✓				✓			P
Gu et al. [44]	DDPG		✓					✓		✓	✓		P
Wu et al. [31]	DDPG			✓				✓		✓	✓		P
Sundermeyer et al. [39]	AAE			✓		✓					✓	✓	C
Fang et al. [46]	Encoder-Structure		✓			✓			Action-heatmap		✓		P
Thermos et al. [30]	AE	✓	✓			✓						✓	P
Holomjova et al. [22]	GSMR-CNN	✓	✓		Few-shot	✓					✓	✓	P
Wen et al. [22]	NOF				graph-optimization	✓					✓	✓	C
Varadarajan et al. [36]	PDLSA	✓				✓						✓	C
Wake et al. [50]	LLM	✓	✓		Few-shot	✓			Text		✓	✓	P

From this *Table 4.1* the systems that have real-life verification experiments and have the P affordance type are filtered to be compared for real-life success rate. These systems are analyzed further since these systems are more generalizable to the criteria of our research question. *Table 4.2* displays the average success rate across all policies. The average success rate is chosen as the primary metric, and is defined as the ratio between the number of successful executions and the number of attempts.

Table 4.2: This Table shows the average success rate across all policies in every model in real-life testing. Note that this average success rate is obtained differently for every model approach. This purely functions as an indication on how these models can perform for our end-to-end system.

Model success rate in real-life tests				
Papers	***Model***	***Learning***	***Average Success Rate***	***Policy***
Moldovan et al. [34]	CNN	Probabilistic	≈60%	grasp, pushing, tapping
Levine et al. [23]	CNN	Reinforcement	≈70%	Grasping
Borja-Diaz et al. [24]	DRL	Reinforcement	>80% (84-90%)	grasping, opening
Geng et al. [17]	DRL	Reinforcement	≈50%	opening door/pot lid, pick & place, pushing
Gu et al. [44]	DDPG	Reinforcement	≈100%	reaching, pulling , pick & place, pushing
Wu et al. [31]	DDPG	Reinforcement	>30%	pulling, pushing
Holomjova et al. [22]	GSMR-CNN	Supervised	<80% (76.4%)	grasping
Wake et al. [50]	LLM	Self-supervised	≈100%	grasp, opening, closing, moving, releasing

Moreover, from the *Table 4.1* the systems that have tested models and have the C affordance type are filtered to be compared on their model validation metrics (accuracy, recall, precision and AUC-ROC). These systems are analyzed in order to dissect elements for scene understanding. *Table 4.3* displays the scene representation models and their performance metrics. These elements can be eventually used in designing an object affordance end-to-end system to our context-specific application.

Table 4.3: This Table shows the performance metrics of the scene context models. Each model has is validated on another affordance regarding scene understanding. This is why the main validation results of these models have different metrics. The general validation metrics are indicated: accuracy (Acc), recall (Rec), and area under ROC curve (AUC-ROC). Moreover, the used dataset, way of learning and scene representations (rep) are given per model.

Model performance metrics								
Papers	***Model***	***Dataset***	***Learning***	***Classification***	***Acc***	***Rec***	***AUC-ROC***	***Scene***
Wu et al. [38]	CDBN	ModelNet	Supervised	10-clas 40-class	≈85.0% ≈77.0%		0.69 0.50	3D-rep
Lee et al. [35]	CDBN	Caltech-101	Unsupervised	3-class	≈65.0%		≈0.68	Hierarchy-rep
Do et al. [41]	DCN	IIT-AFF & UMD	Supervised	27-class 16-actionlabels	≈80.0%			Function-rep
Varadarajan et al. [36]	PDLSA	UW dataset	Supervised	14-class	≈88.0%			3D-rep
Hu et al. [33]	DCN	Extended ModelNet40	Supervised	25-class 18-actionlabels		≈0.70		Function-rep
Sundermeyer et al. [39]	CNN	3D model views	Self-supervised	one-shot-class		≈0.70		3D-rep
Wen et al. [40]	NOF	BEHAVE	Self-supervised	no-class			≈0.90	3D-rep

5

Discussion and future work

5.1. Discussion

In this Chapter the results from the conceptual analysis (*Chapter 3*) and the meta-analysis (*Chapter 4*) are discussed. These results are then combined to design an effective approach for end-to-end object affordance.

5.1.1. Discussion of conceptual and meta-analysis

The conceptual analysis is derived from *Chapter 3*, with the corresponding terms summarized in *Table 3.1*. From these concepts grasping affordance can be defined as the simple task that our system needs to perform since it focuses solely on the grasping task. Task-oriented affordance can be defined for both simple and advanced tasks because it can infer different actions on different features of the object. These can be both simple (only grasping) or extendable (grasping and pulling). Manipulation affordances are for advanced tasks and focus on object affordance produced task sequences more. Therefore, for the simple system it would be beneficial to focus on grasp and task-oriented affordance methods, whereas for advanced tasks a combination of the task-oriented and manipulation affordance can be used. In the case of object affordance learning as stated in *Table 3.2*, a combination of trained properties and observations/interactions is useful to gain full scene understanding and associating specific actions with it. For simple tasks geometric, physical and semantic representations are adequate since these improve scene understanding of the agent. However, a more robust system for advanced tasks benefits from merging these geometric, physical and semantic properties together with observational representations. This is due to the ability of observational representations to generalize to new object categories in scenes. Interaction representations are also beneficial for advanced tasks since these are mostly RL based and explore the environment to learn affordance in real-time. However, implementing these interaction and observational representations are mainly dependent on the quality of the interaction/demonstration data which adds another level of complexity to the system.

The meta-analysis is derived from *Chapter 4*, with the model literature analysis results stated in *Table 4.1*. From this analysis the models that fit the system criteria in *Chapter 2* can be filtered. Systems with multiple ways to obtain affordance are more robust and generalizable from simple to more advanced tasks. Moreover, the models taking RGB-(D) data as input are more favourable to implement with Microsoft HoloLens and robot arm vision. At last the models that contain affordance type P (produced policies from affordance features) are beneficial in direct implementation and testing of the system. These models that include pulling and grasping as affordance label can then be generalized to opening the fuse cases of the voltage racks. This Table indicates that deep reinforcement learning and generative models predominantly meet these criteria. It is important to note that most deep reinforcement learning models also incorporate a CNN backbone, thereby integrating scene context into their affordance labeling. Most CNN models generate the trained properties (geometrical,physical and semantic representations), whereas the deep reinforcement learning and generative models predict interaction regions and labels (interaction and observational representations). Therefore, to meet the criteria of the system it would be beneficial to incorporate a deep reinforcement learning or generative models to generalize from simple to advanced tasks with the learning element of the model while taking in RGB-(D) as input.

From these models in *Table 4.1* the validated models containing performance metrics are compared. The models that produce policies from affordance features are compared with their average success rate in real life tests and output policy labels as stated in *Table 4.2*, and the models that produce scene context features are compared with their associated model performance metrics (accuracy, recall, AUC-ROC) and their output representations as stated in *Table 4.3*. It is important to recognize that the performance metrics used to evaluate these systems and the datasets on which they are trained differ. This variation necessitates caution when making direct comparisons. Nevertheless, valuable insights can be extracted regarding the strengths and potential applications of each model. In the results of *Table 4.2* the deep reinforcement learning models (DRL and DDPG) show ranging performances which indicate that these models might be difficult to converge or generalize to certain environments or new policies. Moreover, the models with the higher average success rate score have reached convergence in their learning policies after a considerable amount of iterations. These models are too complex and non generalizable to our application. Other models such as the GSMR-CNN and other CNN structures show less complex and generalizable properties, however they are mostly generating simple policies such as grasping or pushing. The average success rate of these models is also sufficient (60% to 80%), which shows potential in applying in the simple task execution. When analyzing the performance metrics of the scene context models in *Table 4.3* the models with higher object and affordance classes show reasonable performance in accuracy, AUC-ROC and recall. The more classes and labels a model can differentiate, the more detailed its understanding of affordances. The higher scoring accuracy and AUC-ROC models generally produce 3D representations in the scene. However for our end-to-end system it is beneficial to look into immediate function representations as represented by the deep convolutional networks. Generative models such as LLMs and auto encoder models can be useful as an extension after implementing simple function representations since these can achieve an average success rate of 100% after multiple iterations for multiple policies. Supervised models display generally higher performance metric values, however unsupervised and self-supervised are advantageous in making a system that extends from simple to advanced tasks since it potentially reduces the need for labeled data. These findings are visualized in a comparison chart in *Table 5.1* where the models are compared to the criteria of the system mentioned in *Chapter 2*.

Table 5.1: In this Table the models are generally compared on their performance in object affordance learning. Criteria 1 (adaptability), 2 (generalizability), 3 (semantic, geometric, physical, observational and interaction representations) and 4 (single and multiple policy) are hereby covered for each model. Rep is the abbreviated version of representations. ✓represents low performance, ✓✓represents moderate performance, ✓✓✓represents high performance. Note that performance is defined from the analysis relative to the other models.

Model comparison for object affordance learning				
	CNNs	**DRLs**	**GMs**	**Other**
Adaptable	✓✓	✓✓✓	✓✓	✓
Generalizable	✓✓✓	✓	✓✓	✓✓
Learning semantic rep	✓✓✓	✓	✓✓	✓✓✓
Learning geometric and physical rep	✓✓✓	✓✓	✓✓	✓✓✓
Learning observational rep	✓✓	✓✓	✓✓✓	✓
Learning interaction rep	✓	✓✓✓	✓✓	✓
Single policy learning	✓✓	✓✓	✓✓	✓✓
Multiple policy learning	✓	✓✓	✓✓✓	✓

Based on the analyses the following conclusions can be drawn: The most efficient system for our application consists of a model that has RGB-(D) data as input, uses a CNN structured backbone for scene context with a semi-supervised/unsupervised/reinforcement/one-shot or generative learning element to extend to different situations and task labels. It would be the effective to implement models with direct policy or function representation output to generalize from simple to advanced tasks. Furthermore, the model needs to be directly implementable with the Franka arm and the Microsoft HoloLens as a UI for the object affordance end-to-end system. Therefore, a system with a CNN backbone and siamese network structure such as a GSMR-CNN are efficient for simple tasks such as grasping where the input is a real time RGB-(D) scene and an image of the object to grasp and output segmention masks, object detection and grasp affordance of the object. Moreover, a system with an LLM is efficient for advanced tasks such as grasping, pulling and pushing sequences where the input are RGB-(D) demonstration videos and output scene affordance and motor actions. The following sections presents the proposed systems based on this analysis.

5.1.2. Proposed end-to-end object affordance systems for simpler tasks

GSMR-CNN with scene context and grasping affordance - The proposed system with a Grasping Siamese Mask R-CNN (GSMR-CNN) has an additional branch for grasp detection in parallel to an object detection head branches. It takes as input the scene and an image of the target object. This model employs a shared feature approach, leading to fewer parameters and potentially faster training and inference times. It achieves a grasp accuracy of ≈80% and generalizes well to new object categories (self-supervised), making it ideal for real-time object recognition and manipulation. The model's grasp detection branch uses fully connected layers and outputs grasp pose parameters, trained using a multi-task loss function. This approach excels in executing a single action, like grasping, effectively. It achieves this through training on a specific dataset and using few-shot interactions to refine its performance. A visualization of the full end-to-end object affordance model is displayed in *Figure 5.1*. Note that this is an extension of the model visualization in *Figure 2.2*. The input and output images are the whole test setup of a voltage rack and its handle

5.1.3. Proposed model for advanced tasks

LLM with CNN backbone for dynamic scene context and task-affordance - The proposed system with an LLM and a CNN backbone separates the scene context (extracted with CNN) from the task actions (extracted with LLM). This system takes as input textual instructions and a video of the scene with interaction for learning. How to perform the action functions are extracted from the scene and the observations from the video data. The output is a sequence of actions that the user can alter by natural-language feedback. The use of LLMs in combination with a CNN is generalizable to multiple actions. By learning through observation while using few-shot interactions the system is adaptable to other scenes. According to the analysis the system can achieve an average success rate ≈100% after multiple iterations which makes in a sufficient system for performing advanced tasks. A visualization of the full end-to-end object affordance model is displayed in *Figure 5.2*.

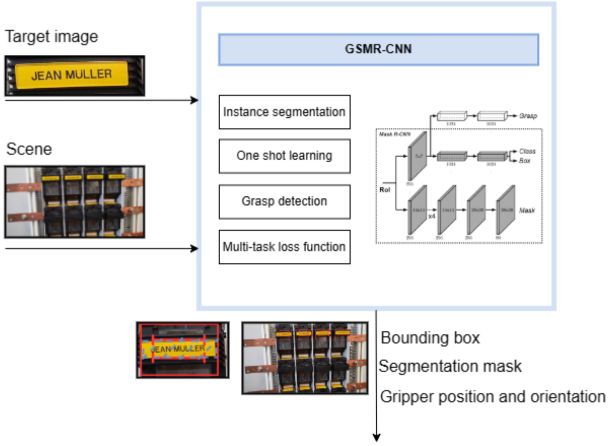

Figure 5.1: End-to-end object affordance system for simple tasks with GSMR-CNN.

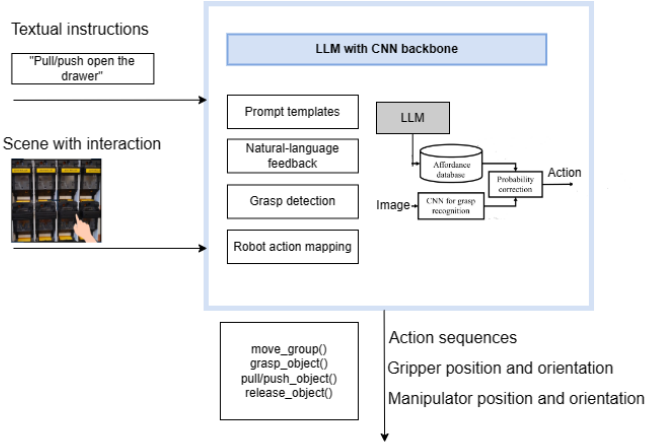

Figure 5.2: End-to-end object affordance system for advanced tasks using a LLM with CNN backbone.

5.2. Future work

For future work the end-to-end model from *Figure 5.1* is going to be implemented for simple grasping tasks and tested in real life. For extending the system to learning advanced tasks the end-to-end model from *Figure 5.2* is going to be implemented and tested in real life. Moreover, other extensions to the system could be made by using systems such as Generative Adversarial Networks (GANs). These are a type of model that uses a generator and a discriminator. The generator tries to create realistic data, while the discriminator tries to distinguish between real and fake data. The two networks compete with each other, improving the quality of the generated data [55]. Due to time constraints of the project these would be interesting approaches for future projects or extensions. On the basis of this analysis other model approaches such as the use of an auto encoder or a DRL can be used. A DRL can add a reinforcement learning element in the system which is beneficial in improving policy execution [29]. By adding an auto encoder it is possible to process frames of human-object interaction sequences and infers pixel-wise affordance label predictions. It would be interesting to combine this with the Microsoft HoloLens UI, where the operator can collect RGB-(D) video data of human-object interaction sequences to teach the manipulator scene context and to perform certain tasks [30].

6 Conclusion

Companies like Alliander aim to alleviate the workload of workers and enhance task execution efficiency while working on voltage racks by proposing robotic solutions such as the Microsoft HoloLens and robotic manipulators. Especially the switching of fuses in voltage racks is a physically intensive task and needs skilled electrical engineers in the field. Most robotic solutions deployed in the field use a general path planner with motion primitives or multi-stage approaches which are difficult to generalize to more advanced task sequences. Research in the combined areas of vision, action, and path planning is rapidly advancing, showcasing its vital role in the development of sophisticated designed systems. These systems use deep learning models in order to extract object affordance, which enables robots to recognize and manipulate objects based on their functional regions and inherent properties. Hence, this literature research seeks to address the research question: *What is the most effective deep learning end-to-end strategy for object affordance in robotic manipulation, such as grasping, pulling and pushing, within the current state-of-the-art?*.

This literature research aims to answer this question by assessing the different approaches of obtaining object affordance and analyse the capabilities and limitations of the existing end-to-end deep learning models. The analysis of these approaches and models involves a two-step strategy: firstly, a conceptual analysis of object affordance approaches as documented in the literature, and secondly, a meta-analysis of the latest state-of-the-art models for end-to-end object affordance. The performances of these approaches and models are assessed and a framework for an end-to-end object affordance system is proposed for simple and advanced tasks. The results of the analysis indicate that the state-of-the-art end-to-end systems contain a reinforcement learning element or an encoder structure for policy refinement, a CNN backbone for scene context (3D representations and semantics). The analysis suggests that for simpler tasks, a Grasping Siamese Mask R-CNN (GSMR-CNN) would be effective, particularly for tasks like object grasping, which require real-time RGB-(D) scene processing. For more advanced tasks involving sequences such as grasping, pulling, and pushing, a Large Language Model (LLM) with Convolutional Neural Network (CNN) backbone can be used to generate multiple action affordances. Moreover, end-to-end auto encoders or deep reinforcement learning methods are effective as well but increase complexity in the system. These models offer the necessary generalizability, adaptability, and accuracy for task execution.

In conclusion, this literature research addresses the most effective deep learning strategies for object affordance in robotic manipulation of advanced tasks on voltage racks. For simpler tasks, the GSMR-CNN model is adequate, while more complex tasks may require a LLM with CNN backbone. The advancement of robotic manipulation in technical settings depends on the continued development and integration of these end-to-end deep learning strategies for object affordance. It is recommended that further research focuses on refining this model for specific applications, improving their generalizability for enhanced safety and efficiency.

Bibliography

[1] J. Guhl, S. Tung, and J. Kruger, “Concept and architecture for programming industrial robots using augmented reality with mobile devices like microsoft hololens,” in *2017 22nd IEEE international conference on emerging technologies and factory automation (ETFA)*. IEEE, 2017, pp. 1–4.

[2] H. Liu, Y. Zhang, W. Si, X. Xie, Y. Zhu, and S.-C. Zhu, “Interactive robot knowledge patching using augmented reality,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1947–1954.

[3] S. Liu, X. V. Wang, and L. Wang, “Digital twin-enabled advance execution for human-robot collaborative assembly,” *CIRP annals*, vol. 71, no. 1, pp. 25–28, 2022.

[4] M. Ostanin and A. Klimchik, “Interactive robot programing using mixed reality,” *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 50–55, 2018.

[5] Alliander, “About Alliander,” Mar 2020. [Online]. Available: https://www.alliander.com/en/organisation/

[6] K.-B. Park, S. H. Choi, J. Y. Lee, Y. Ghasemi, M. Mohammed, and H. Jeong, “Hands-free human–robot interaction using multimodal gestures and deep learning in wearable mixed reality,” *IEEE Access*, vol. 9, pp. 55 448–55 464, 2021.

[7] J. Delmerico, R. Poranne, F. Bogo, H. Oleynikova, E. Vollenweider, S. Coros, J. Nieto, and M. Pollefeys, “Spatial computing and intuitive interaction: Bringing mixed reality and robotics together,” *IEEE Robotics & Automation Magazine*, vol. 29, no. 1, pp. 45–57, 2022.

[8] Microsoft, “Microsoft hololens official website,” https://learn.microsoft.com/en-us/hololens/, 2023, accessed: 2023-12-10.

[9] H. Fang, S. K. Ong, and A. Y.-C. Nee, “Robot programming using augmented reality,” in *2009 International Conference on CyberWorlds*. IEEE, 2009, pp. 13–20.

[10] G. Avalle, F. De Pace, C. Fornaro, F. Manuri, and A. Sanna, “An augmented reality system to support fault visualization in industrial robotic tasks,” *Ieee Access*, vol. 7, pp. 132 343–132 359, 2019.

[11] H. Picard, J. Verstraten, and R. Luchtenberg, “Practical approaches to mitigating arc flash exposure in europe,” in *PCIC Europe 2013*. IEEE, 2013, pp. 1–10.

[12] N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford *et al.*, “The limits and potentials of deep learning for robotics,” *The International journal of robotics research*, vol. 37, no. 4-5, pp. 405–420, 2018.

[13] M. Ostanin, S. Mikhel, A. Evlampiev, V. Skvortsova, and A. Klimchik, “Human-robot interaction for robotic manipulator programming in mixed reality,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2805–2811.

[14] J. Ni and V. Balyan, “Research on mobile user interface for robot arm remote control in industrial application,” *Scalable Computing: Practice and Experience*, vol. 22, no. 2, pp. 237–245, 2021.

[15] Y. Zuo, W. Qiu, L. Xie, F. Zhong, Y. Wang, and A. L. Yuille, “Craves: Controlling robotic arm with a vision-based economic system,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4214–4223.

[16] J. W. S. Chong, S. Ong, A. Y. Nee, and K. Youcef-Youmi, “Robot programming using augmented reality: An interactive method for planning collision-free paths,” *Robotics and Computer-Integrated Manufacturing*, vol. 25, no. 3, pp. 689–701, 2009.

[17] Y. Geng, B. An, H. Geng, Y. Chen, Y. Yang, and H. Dong, “End-to-end affordance learning for robotic manipulation,” *arXiv preprint arXiv:2209.12941*, 2022.

[18] H. Liu, Y. Zhang, W. Si, X. Xie, Y. Zhu, and S.-C. Zhu, “Interactive robot knowledge patching using augmented reality,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1947–1954.

[19] S. Amarjyoti, “Deep reinforcement learning for robotic manipulation-the state of the art,” *arXiv preprint arXiv:1701.08878*, 2017.

[20] J. Guo, P. Chen, Y. Jiang, H. Yokoi, and S. Togo, “Real-time object detection with deep learning for robot vision on mixed reality device,” in *2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech)*. IEEE, 2021, pp. 82–83.

[21] B. Fang, F. Sun, H. Liu, C. Liu, and D. Guo, *Wearable technology for robotic manipulation and learning*. Springer, 2020.

[22] V. Holomjova, A. J. Starkey, and P. Meißner, “Gsmr-cnn: An end-to-end trainable architecture for grasping target objects from multi-object scenes,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3808–3814.

[23] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection,” *The International journal of robotics research*, vol. 37, no. 4-5, pp. 421–436, 2018.

[24] J. Borja-Diaz, O. Mees, G. Kalweit, L. Hermann, J. Boedecker, and W. Burgard, “Affordance learning from play for sample-efficient policy learning,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 6372–6378.

[25] C. Pohl, K. Hitzler, R. Grimm, A. Zea, U. D. Hanebeck, and T. Asfour, “Affordance-based grasping and manipulation in real world applications,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9569–9576.

[26] S. Levine, C. Finn, T. Darrell, and P. Abbeel, “End-to-end training of deep visuomotor policies,” *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1334–1373, 2016.

[27] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning,” *arXiv preprint arXiv:1509.02971*, 2015.

[28] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, “Human-level control through deep reinforcement learning,” *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[29] Y. Wang, R. Wu, K. Mo, J. Ke, Q. Fan, L. J. Guibas, and H. Dong, “Adaafford: Learning to adapt manipulation affordance for 3d articulated objects via few-shot interactions,” in *European Conference on Computer Vision*. Springer, 2022, pp. 90–107.

[30] S. Thermos, P. Daras, and G. Potamianos, “A deep learning approach to object affordance segmentation,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 2358–2362.

[31] R. Wu, Y. Zhao, K. Mo, Z. Guo, Y. Wang, T. Wu, Q. Fan, X. Chen, L. Guibas, and H. Dong, “Vatmart: Learning visual action trajectory proposals for manipulating 3d articulated objects,” *arXiv preprint arXiv:2106.14440*, 2021.

[32] K. Mo, L. J. Guibas, M. Mukadam, A. Gupta, and S. Tulsiani, “Where2act: From pixels to actions for articulated 3d objects,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6813–6823.

[33] R. Hu, Z. Yan, J. Zhang, O. Van Kaick, A. Shamir, H. Zhang, and H. Huang, “Predictive and generative neural networks for object functionality,” *arXiv preprint arXiv:2006.15520*, 2020.

[34] B. Moldovan, P. Moreno, D. Nitti, J. Santos-Victor, and L. De Raedt, “Relational affordances for multiple-object manipulation,” *Autonomous Robots*, vol. 42, pp. 19–44, 2018.

[35] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Unsupervised learning of hierarchical representations with convolutional deep belief networks,” *Communications of the ACM*, vol. 54, no. 10, pp. 95–103, 2011.

[36] K. M. Varadarajan and M. Vincze, “Parallel deep learning with suggestive activation for object category recognition,” in *International Conference on Computer Vision Systems*. Springer, 2013, pp. 354–363.

[37] N. Yamanobe, W. Wan, I. G. Ramirez-Alpizar, D. Petit, T. Tsuji, S. Akizuki, M. Hashimoto, K. Nagata, and K. Harada, “A brief review of affordance in robotic manipulation research,” *Advanced Robotics*, vol. 31, no. 19-20, pp. 1086–1101, 2017.

[38] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, “3d shapenets: A deep representation for volumetric shapes,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912–1920.

[39] M. Sundermeyer, Z.-C. Marton, M. Durner, M. Brucker, and R. Triebel, “Implicit 3d orientation learning for 6d object detection from rgb images,” in *Proceedings of the european conference on computer vision (ECCV)*, 2018, pp. 699–715.

[40] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Müller, A. Evans, D. Fox, J. Kautz, and S. Birchfield, “Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 606–617.

[41] T.-T. Do, A. Nguyen, and I. Reid, “Affordancenet: An end-to-end deep learning approach for object affordance detection,” in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 5882–5889.

[42] L. E. Hafi, H. Nakamura, A. Taniguchi, Y. Hagiwara, and T. Taniguchi, “Teaching system for multimodal object categorization by human-robot interaction in mixed reality,” in *2021 IEEE/SICE International Symposium on System Integration (SII)*, 2021, pp. 320–324.

[43] L. El Hafi, H. Nakamura, A. Taniguchi, Y. Hagiwara, and T. Taniguchi, “Teaching system for multimodal object categorization by human-robot interaction in mixed reality,” in *2021 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2021, pp. 320–324.

[44] S. Gu, E. Holly, T. Lillicrap, and S. Levine, “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates,” in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3389–3396.

[45] K. Lotsaris, C. Gkournelos, N. Fousekis, N. Kousi, and S. Makris, “Ar based robot programming using teaching by demonstration techniques,” *Procedia CIRP*, vol. 97, pp. 459–463, 2021.

[46] K. Fang, T.-L. Wu, D. Yang, S. Savarese, and J. J. Lim, “Demo2vec: Reasoning object affordances from online videos,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2139–2147.

[47] F. Zhang, J. Leitner, M. Milford, B. Upcroft, and P. Corke, “Towards vision-based deep reinforcement learning for robotic motion control,” *arXiv preprint arXiv:1511.03791*, 2015.

[48] R. Mirjalili, M. Krawez, S. Silenzi, Y. Blei, and W. Burgard, “Lan-grasp: Using large language models for semantic object grasping,” *arXiv preprint arXiv:2310.05239*, 2023.

[49] C. Tang, D. Huang, W. Ge, W. Liu, and H. Zhang, “Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping,” *arXiv preprint arXiv:2307.13204*, 2023.

[50] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi, “Chatgpt empowered long-step robot control in various environments: A case application,” *arXiv preprint arXiv:2304.03893*, 2023.

[51] A. Murali, W. Liu, K. Marino, S. Chernova, and A. Gupta, “Same object, different grasps: Data and semantic knowledge for task-oriented grasping,” in *Conference on Robot Learning*, 2020.

[52] M. Khansari, D. Kappler, J. Luo, J. Bingham, and M. Kalakrishnan, “Action image representation: Learning scalable deep grasping policies with zero real world data,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3597–3603.

[53] A. Abdi, D. Adhikari, and J. H. Park, “A novel hybrid path planning method based on q-learning and neural network for robot arm,” *Applied Sciences*, vol. 11, no. 15, p. 6770, 2021.

[54] OpenAI, “Openai,” 2023, accessed: 2023-12-19. [Online]. Available: https://openai.com/

[55] S. Bender, T. Joseph, and J. M. Zöllner, “Cycle-consistent world models for domain independent latent imagination,” in *European Conference on Computer Vision*. Springer, 2022, pp. 561–574.

[56] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, “Segment anything,” *arXiv preprint arXiv:2304.02643*, 2023.

A
Appendix A

A.1. Prelimary setup

Test setup

The Franka Emika robot arm is mounted on a stable platform. For experiments for the simple task the arm is mounted next to an empty table. For experiments for the advanced task the arm is situated adjacent to the front face of the low voltage rack, ensuring optimal accessibility and visibility. An old low voltage rack from the company Alliander N.V. is used for testing. Furthermore the robot arm is provided with vision by adding a RealSense D435i depth camera to the part just above the gripper and a 3D printed mount.

Cloud computing platforms for storage and computing

Moreover, Azure cloud Services such as Azure Storage (including Azure Table Storage and Azure Blob Storage) and Azure Spatial Anchors are used. These cloud services are used for computing code and storage of information fetched by the Microsoft HoloLens device.

A.2. Vision app

The robot arm can be operated by a operator with a Microsoft HoloLens 2 device. The manual connection between the HoloLens and the task control of the arm can be done by making a manual vision application. In this application an object card can be made where the operator can manually make a label (with image, name, and description) and gather image data with this label. This data is thereafter stored in the Azure Storage which can be directly used for manual labeling for datasets or other inspection purposes. Furthermore, the location of this object with its label can be stored as an Azure Spatial Anchor. This location can then be retrieved from the storage for setting as goal location for the robotic arm control. With the computer vision button the user can start real-time object detection and segmentation with YOLOv8 of every-day objects [20, 56]. This information can all be gathered as input for the end-to-end control of the arm. The app is visualized in *Figure A.1*, where the left displays the app UI and the right the outputs of real-time object detection/segmentation.

Labeling The user can make an object card for labeling a certain object which includes an image, name, description and location.

Collecting image data The user can collect image data for an object card label.

Setting 3D location The user can set a 3D spatial anchor with the associated object card label.

Vision information With the computer vision button the user can extract object bounding boxes and masks.

Saving information to cloud Saving the object card information in the Azure Cloud.

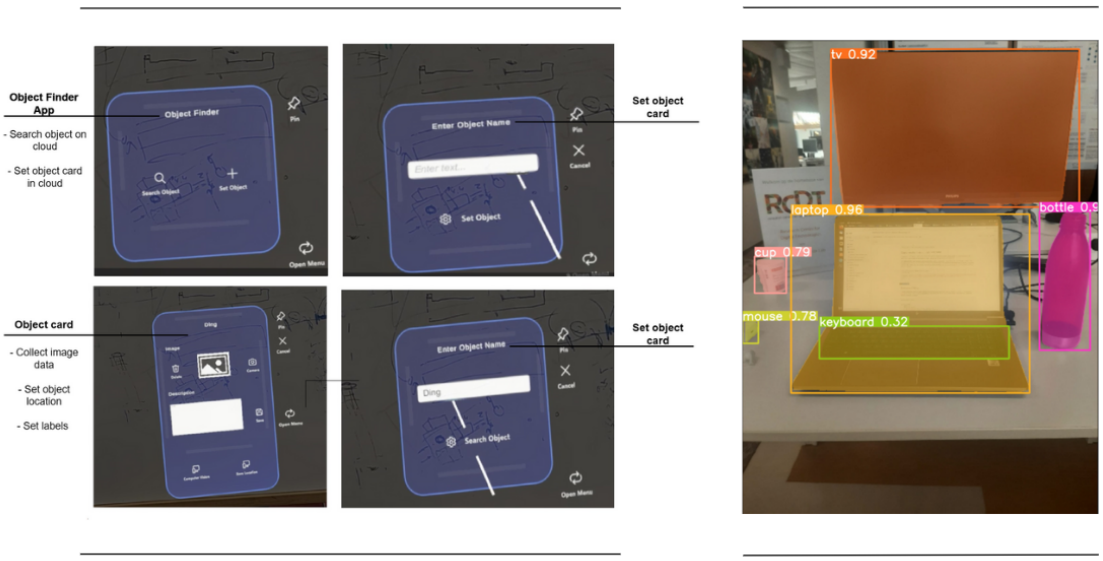

Figure A.1: This Figure displays the Vision App. This app can label images, collect image data via the HoloLens, set the location of the objects and perform computer vision (detection and segmentation) to collect object bounding boxes and masks.