Lancaster University

Optimising Brazing Processes through Learning-Based Visual Servoing and Collaborative Robotics

Haolin Fei, BEng (Hons)
School of Engineering
Lancaster University

A thesis submitted for the degree of
Doctor of Philosophy

October, 2023

Optimising Brazing Processes through Learning-Based Visual Servoing and Collaborative Robotics

Haolin Fei, BEng (Hons).
School of Engineering, Lancaster University
A thesis submitted for the degree of *Doctor of Philosophy.* October, 2023

Abstract

In manufacturing safety-critical components, brazing stands out for its ability to form strong, cost-efficient joints between dissimilar materials. Despite its importance, the brazing process often falls short in efficiency and precision due to its reliance on manual labour. Simultaneously, full automation, while enhancing certain operational aspects, lacks the adaptability and decision-making prowess inherent to human operators. This thesis addresses these challenges within the brazing process by advocating for a synergistic integration of robotics and artificial intelligence (AI) in a human-robot collaboration (HRC) framework. It uniquely combines human expertise with advanced machine capabilities, aiming to refine brazing operations beyond the reach of solely human or automated endeavours.

Central to the thesis is the development of a category-agnostic object localisation strategy. This technique enables robots to recognise and position brazing filler metal (BFM) across a diverse array of joint configurations without prior specific knowledge of the objects. By leveraging AI-driven insights, this approach significantly enhances operational precision and adaptability, illustrating its utility in complex assembly tasks where traditional methods fall short.

Building on this foundation, a learning-based visual servoing method is introduced. This innovative approach allows robots to dynamically adjust their actions in real-time based on visual feedback, navigating complicated environments and performing tasks with heightened accuracy. Such capability is crucial for ensuring the consistent placement of BFM under varying conditions, demonstrating a marked improvement in the process's reliability and efficiency.

Finally, an intuitive human-robot collaboration framework is proposed. This model is designed to seamlessly integrate the strengths of both humans and robots, facilitating a partnership that leverages the precision of automation and the judgement of human operators. Through examples such as collaborative adjustment of brazing parameters in response to real-time observations, the framework underscores the importance of human

insight in augmenting robotic capabilities.

This approach not only advances the brazing process by mitigating the reliance on skilled labour and enhancing safety standards but also lays a foundation for applications beyond brazing, highlighting the transformative potential of integrating human and robotic expertise in industrial processes.

Acknowledgements

I would like to express my deepest gratitude to the Lloyd's Register Foundation and Lancaster University for their generous sponsorship and unwavering support, which made this PhD research possible. I am particularly thankful to the Lloyd's Register Foundation, a charitable foundation dedicated to protecting life and property through its support of engineering-related education, public engagement, and research application. I am also grateful for the invaluable support provided by the National Structural Integrity Research Centre (NSIRC) and the Joining 4.0 Innovation Centre (J4IC), two exceptional institutions that foster collaboration between industry and academia. Additionally, I would like to extend my appreciation to the Human Robotics Group at Imperial College for their support during our collaboration.

I am deeply indebted to my academic supervisors, Dr Ziwei Wang, Prof Andrew Kennedy and Dr Yingtao Tian, for their invaluable guidance and assistance throughout my PhD research. Their expertise and mentorship have not only shaped my academic pursuits but have also had a profound impact on my personal growth. I am especially grateful to Dr Ziwei Wang for rescuing me from moments of confusion and providing me with clear direction in the field. Their unwavering enthusiasm and commitment to scholarship have inspired me to make meaningful contributions to the field. Furthermore, I would like to express my sincere appreciation to my industrial supervisors, Dr Nick Ludford and Dr Stefano Tedeschi. I am deeply impressed by their profound knowledge and exceptional management skills, as well as their strong connections with industry partners, research institutes, and universities. Their guidance and support have been instrumental in shaping the practical aspects of my research. I would also like to thank Prof Darren Williams, the director of J4IC, for his valuable support and insightful advice throughout my research journey. I am immensely grateful to Xiaofang and all my friends who have provided unwavering support during challenging times. Their encouragement, understanding, and companionship have been a constant source of strength and inspiration.

Finally, I would like to acknowledge the countless individuals, researchers, and institutions whose work and contributions have paved the way for my own research. Their dedication to advancing knowledge in the field has been instrumental in shaping my own academic journey.

Declaration

I declare that the work presented in this thesis is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. This thesis does not exceed the maximum permitted word length of 80,000 words including appendices and footnotes, but excluding the bibliography.

Haolin Fei

Haolin Fei
November, 2024

Publications

Fei, H., Kennedy, A., Wang, Z. (2023). Hybrid Approach for Efficient and Accurate Category-Agnostic Object Detection and Localization with Image Queries in Human-Robot Interaction, 49th Annual Conference of the IEEE Industrial Electronics Society (IECON).

Fei, H., Kennedy, A., Wang, Z. (2023). An Anthropomorphic Framework for Learning-Based Visual Servoing to Reach Unseen Objects, IEEE International Conference on Automation Science and Engineering (CASE).

Fei, H., Kennedy, A., Wang, Z. (2023). Robust Reinforcement Learning Based Visual Servoing with Convolutional Features, In 2023 International Federation of Automatic Control (IFAC) World Congress.

Sun, L., Ma, N., Xiao, B., Huang, Y., **Fei, H.**, Yeatman, E. (2023). Adaptive Robust Fault-Tolerant Regulation of Mechatronic Systems with Prescribed-Time Convergence, In 2023 International Federation of Automatic Control (IFAC) World Congress.

Fei, H., Wang, Z., Tedeschi, S., Kennedy, A. (2023). Boosting visual servoing performance through RGB-based methods. Robotic Intelligence and Automation, 43(4), 468-475.

Fei, H., Kennedy, A., Williams, D., Saxty, A., Wang, Z. (2023). Learning-Based Anthropomorphic Servoing Framework for Supernumerary Limb Object Reaching. IEEE International Conference on Robotics and Automation (ICRA).

Sun, L., Huang, Y., **Fei, H.**, Xiao, B., Yeatman, E. M., Montazeri, A., Wang, Z. (2023). Fixed-time regulation of spacecraft orbit and attitude coordination with optimal actuation allocation using dual quaternion. Frontiers in Robotics and AI, 10, 1138115.

Fei, H., Shijie, L., Xueqian, W., Liucheng, G., Stefano, T., Wang, Z. (2024). Seamless Robot Teleoperation: Intuitive Control through Hand Gestures and Neural Network Decoding. In 2024 World Congress on Computational Intelligence (WCCI).

Under-review Papers:

Wang, Z., **Fei, H.**, Huang, Y., Rouxel, Q., Xiao, B., Li, Z., Burdet, E. (2023). Learning to Assist Bimanual Teleoperation using Interval Type-2 Polynomial Fuzzy Inference. IEEE Transactions on Cognitive and Developmental Systems.

Fei, H., Tedeschi, S., Huang, Y., Kennedy, A., Wang, Z. (2023). Dynamic Hand Gesture-Featured Human Motor Adaptation in Tool Delivery using Voice Recognition. arXiv preprint arXiv:2309.11368.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Brazing plays a pivotal role in manufacturing safety-critical components, offering the unique capability to join dissimilar materials and form robust, cost-efficient bonds. This process is indispensable across a wide spectrum of industries, from aerospace to automotive, where the integrity of joint connections is non-negotiable [1, 2, 3]. Brazing is chosen over comparable techniques like welding due to its unique advantages in joining dissimilar materials, lower operating temperatures reducing thermal stress and distortion, and particular suitability for thin materials and complex assemblies. Unlike welding, brazing maintains the base material properties while creating strong joints, making it crucial for safety-critical components [4]. Despite its advantages, the efficiency and precision of brazing are significantly constrained by its reliance on manual techniques. In an era marked by Industry 4.0, with its emphasis on automation and smart manufacturing, the integration of robotics into brazing processes presents a promising avenue to enhance both accuracy and operational efficiency.

However, the current state of brazing automation faces several critical challenges. Firstly, there are significant limitations in autonomous capabilities. Existing automation in brazing is often restricted to repetitive common tasks, requiring specialised machines for specific operations. The unstructured environments typical in brazing operations pose significant challenges for traditional robotic systems, limiting their ability to adapt to varying workpieces and conditions. This limitation results in inflexibility and increased setup times when switching between different brazing tasks or product lines, hindering the overall efficiency of the manufacturing process.

Secondly, there is a strong dependency on human operators in the brazing process, particularly for tasks requiring dexterity and decision-making. This reliance not only leads to potential inconsistencies but also exposes workers to repetitive tasks that can cause fatigue and reduce overall efficiency. The industry is facing a growing shortage of skilled labor, further exacerbating the challenges of maintaining high-quality brazing operations. The dependence on human expertise introduces variability in process quality and productivity, making it difficult to achieve consistent results across different production runs.

Lastly, accessibility issues present a significant challenge in certain brazing scenarios. Some environments are unsafe or physically unreachable for human operators, such as those involving hazardous materials or spaces with severe limitations. The complexity of brazing techniques often requires extensive training periods, creating a steep learning curve that further compounds the skilled labour shortage. These accessibility issues not only pose safety concerns but also limit the potential for process optimisation in challenging environments, restricting the industry's ability to innovate and improve efficiency in more demanding applications.

To address these challenges, the brazing industry requires advancements in three key areas:

- Flexible Manufacturing Systems: There is a pressing need for intelligent robotic systems capable of navigating unstructured environments and adapting to various brazing tasks with minimal reprogramming. Such systems should be able to take over repetitive or physically demanding tasks, thereby improving consistency and reducing worker fatigue. The development of these systems would significantly enhance the adaptability of brazing operations to different product types and batch sizes.

- Human-Robot Co-Activity: The development of intuitive interfaces for human-robot interaction is crucial. These interfaces should enable seamless task handover between humans and robots, effectively combining the precision of automation with human expertise in decision-making and problem-solving. This collaboration has the potential to leverage the strengths of both humans and robots, leading to improved process efficiency and quality.

- Intuitive Teleoperation: For scenarios where direct human presence is impractical or unsafe, advanced teleoperation systems are required. These systems should be capable of interpreting human intentions accurately and provide intuitive control interfaces, allowing skilled operators to apply their expertise remotely. The development of such systems would not only enhance safety but also extend the reach of human expertise to challenging brazing environments.

By focusing on these areas, this research aims to develop comprehensive solutions that not only enhance the efficiency and quality of brazing processes but also address the broader challenges facing the manufacturing industry in the era of smart automation. The potential impact of such advancements extends beyond brazing, potentially setting new standards for human-robot collaboration in complex manufacturing processes.

1.2 Aims and Scope

The overall aim of this research is to develop an intelligent, flexible, and collaborative robotic system for brazing applications that addresses the three key challenges identified: limited autonomous capabilities, human dependency, and accessibility issues. To tackle these challenges, this research focuses on three corresponding areas, including flexible manufacturing systems, human-robot co-Activity, and intuitive teleoperation.

Specifically, the research aims to:

1) Develop flexible manufacturing systems to address Limited Autonomous Capabilities:

- Design and implement intelligent robotic systems capable of navigating unstructured brazing environments.
- Create adaptive algorithms that enable robots to handle various brazing tasks with minimal reprogramming.
- Enhance the system's ability to quickly adjust to different product types and batch sizes.

2) Establish effective human-robot co-activity to mitigate human dependency:

- Develop intuitive interfaces for seamless human-robot interaction in brazing processes.
- Create collaborative frameworks that optimise task allocation between humans and robots.
- Implement systems that leverage human expertise for complex decision-making while utilising robotic precision for repetitive tasks.

3) Implement intuitive teleoperation solutions to overcome accessibility issues:

- Design advanced teleoperation systems that allow skilled operators to perform complex brazing tasks remotely.
- Develop haptic feedback mechanisms to enhance operator sensory input in remote operations.

- Create intelligent interfaces that accurately interpret and execute human intentions in challenging or hazardous environments.

The scope of this research encompasses:

- Development of computer vision (CV) and machine learning (ML) algorithms for robotic perception in unstructured brazing environments.
- Design and implementation of human-robot interfaces (HRI) for collaborative brazing tasks.
- Creation of advanced teleoperation systems with enhanced feedback capabilities.
- Integration of these technologies into a cohesive system for brazing applications.
- Validation and testing of the developed systems in both simulated and real-world brazing scenarios.

1.3 Thesis Outline

The thesis structure is depicted in a logical flow given in Fig. 1.1.

This thesis begins with Chapter 1, which presents the fundamental motivation driving the enhancement of brazing processes through advanced robotics and artificial intelligence. The chapter establishes three critical challenges in contemporary brazing practices: limited autonomous capabilities, human dependency, and accessibility constraints. Following the problem formulation, it delineates the research aims and scope, setting the foundation for subsequent technical discussions.

The overview of the organisation is given here:

Chapter 2: Preliminaries

Chapter 2 lays the groundwork by introducing essential preliminaries and theoretical foundations. The chapter commences with brazing technology fundamentals before transitioning into intelligent manufacturing concepts. It then details the simulation frameworks vital for experimental validation, encompassing both hardware configurations and software architectures. Particular attention is paid to hand-eye calibration methodology and the implementation of digital twin technology for teleoperation applications.

Chapter 3: Literature Review

This chapter lays the foundation of the thesis by surveying the landscape of academic and industrial research related to brazing, with a particular focus on the integration of automation and robotics to enhance efficiency, precision, and safety in brazing operations. It starts by providing a comprehensive overview of traditional brazing techniques, highlighting their strengths and limitations, and underscores

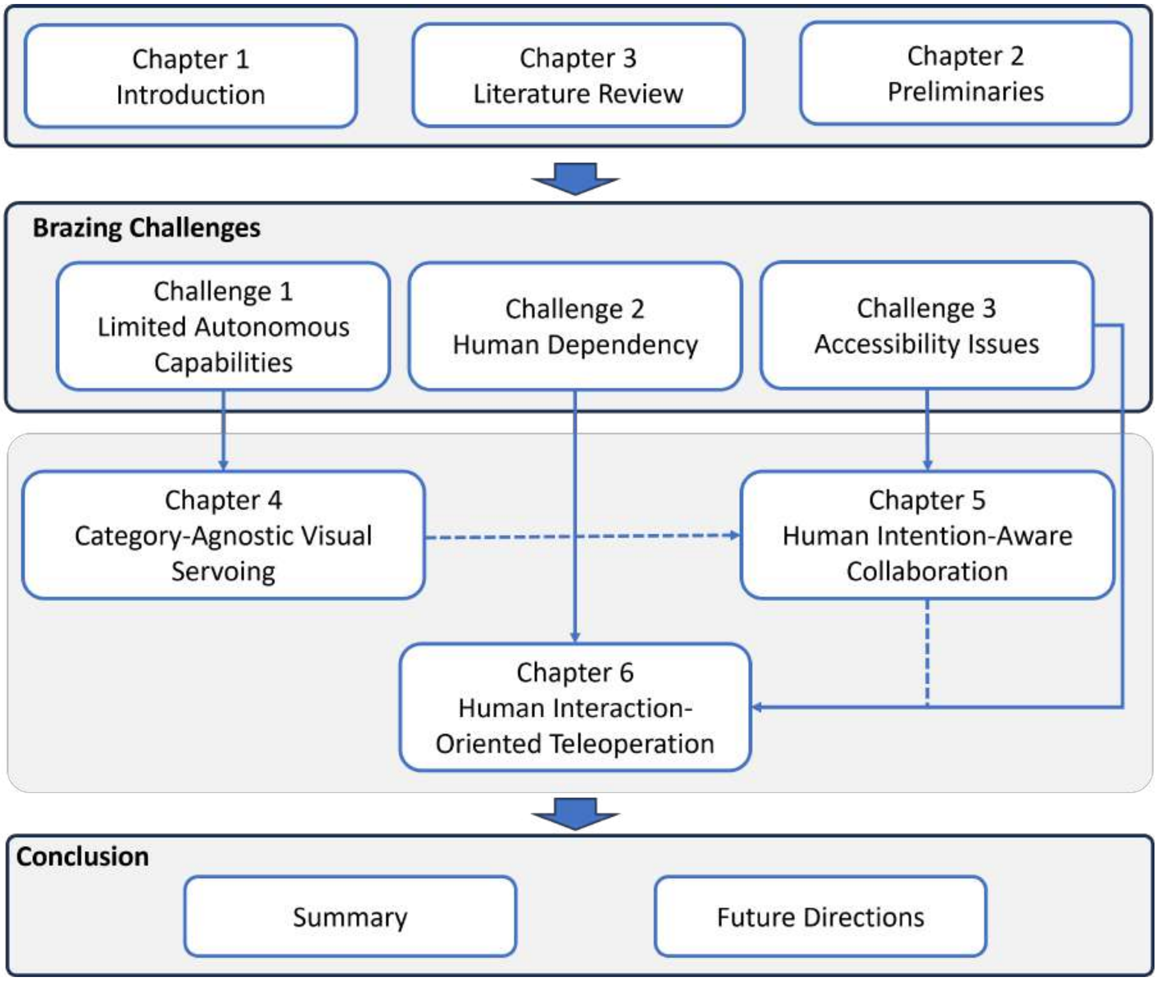


Figure 1.1. Research framework showing the progression from fundamental robot perception (Chapter 4) through human-robot collaboration (Chapter 5) to teleoperation (Chapter 6). The integration of visual servoing, collaborative control, and human-robot interfaces forms a cohesive approach to flexible manufacturing automation.

the critical need for innovation in this domain. The discussion then transitions to the exploration of intelligent manufacturing principles, illustrating their potential to redefine brazing processes through advanced automation and data-driven decision-making. Further, the chapter delves into the specifics of robotic applications in brazing, examining the role of robots in automating repetitive tasks such as braze pasting and component placement. This includes an analysis of current technologies, their operational frameworks, and the challenges they face in practical applications. Special attention is given to the concept of category-agnostic localisation and the importance of precise hand-eye calibration in ensuring the accuracy and reliability of robotic actions within the brazing context. The exploration continues by addressing learning-based visual servoing, a key component in enabling robots to adapt and respond to dynamic manufacturing environments. This section assesses various approaches, including reinforcement learning techniques, and their applicability to the challenges of brazing operations. It evaluates how these methods contribute to the development of more intelligent and autonomous robotic systems capable of complex decision-making and real-time adjustments. Then the focus shifts to the human-robot collaboration (HRC) in the context of brazing. It investigates frameworks and strategies for learning from human expertise, metrics for evaluating HRC efficiency, and models for adaptive collaboration that can accommodate the nuances of brazing tasks in both human-robot co-activity and teleoperation scenarios. The role of teleoperation is also explored as a means to extend human capabilities and enhance safety, particularly in hazardous working conditions.

Chapter 4: Category-Agnostic Object Localisation

Chapter 4 introduces category-agnostic visual servoing, representing a significant advancement in robotic perception and control. The chapter presents a comprehensive comparative analysis of visual servoing algorithms before introducing a novel hybrid approach for category-agnostic object detection and manipulation. This framework seamlessly integrates learning-based methods with traditional control strategies, validated through extensive experimentation in both simulated and physical environments.

Chapter 5: Human Intention-Aware Collaboration

Building upon enhanced robotic perception, Chapter 5 addresses human-robot collaboration through an intention-aware framework. The chapter details the development of dynamic control strategies that adapt to human intentions, incorporating real-time hand pose estimation and gesture recognition. This multimodal approach demonstrates significant improvements in collaborative task execution, particularly in manufacturing scenarios.

Chapter 6:Human Interaction-Oriented Teleoperation

Chapter 6 explores human interaction-oriented teleoperation, presenting both physical controller-based interfaces and gesture-based systems. The chapter details the integration of haptic feedback mechanisms and evaluates system performance across various teleoperation scenarios, emphasizing the practical implementation aspects in real-world applications.

The thesis concludes with Chapter 7, synthesising the research contributions while contextualising their significance within the broader robotics and manufacturing domains. This final chapter also presents a critical discussion of future research directions, particularly focusing on the evolution of flexible manufacturing systems and human-robot collaboration frameworks.

This outline guides the reader through the systematic progression of the thesis, from its inception to the realisation of novel solutions and their implications for intelligent manufacturing in brazing processes.

Chapter 2

Preliminaries

2.1 Introduction

This chapter engages in a comprehensive exploration of existing related works, aiming to establish the current state of knowledge in the field. This critical examination serves two primary purposes: firstly, to identify gaps, trends, and advancements in the existing body of literature, and secondly, to discern the methodologies and approaches employed in prior research endeavours. This systematic analysis of the available literature contextualises the research within the broader academic landscape and guides the formulation of a research framework that either builds upon, challenges, or refines existing theories and methodologies.

In this section, the following questions are aimed to be answered: 1) What is brazing? 2) How to braze? 3) What currently exists as a gap in the brazing landscape? 4) Which process in brazing can benefit greatly from robotics and AI technology? 5) What industrial/academic implications could addressing this gap have? By providing insights into these questions, this research can further narrows down the focus and refining the research scope.

2.2 Brazing Background

Brazing, a well-established manufacturing process, plays a crucial role in fabricating various safety-critical components. According to the definition by the American Welding Society [5], brazing encompasses a group of joining processes that achieve material coalescence by heating them to the brazing temperature and utilising a filler metal (solder) with a liquidus above 450°C and below the solidus of the base metals. One distinctive feature that sets brazing apart from other joining methods is the use of brazing filler metals to bond base materials. By employing a filler metal with a melting temperature lower than that of the base material, the base material remains solid during the brazing process. The advantages and disadvantages of the brazing process are presented in Table 2.1.

Brazing can be further categorised based on the brazing method, which encompasses three main techniques: open-air brazing, controlled atmosphere brazing, and laser brazing [4].

- **Open-air Brazing**: This method encompasses two primary approaches: torch brazing and induction brazing. Torch brazing utilises a welding torch as the heat source, while induction brazing employs a high-frequency electromagnetic field induction as a contactless and flameless heat source. Induction brazing offers several advantages, including localised heating, repeatability, and ease of automation, making it a preferred choice in various applications compared to torch brazing.

- **Controlled Atmosphere Brazing**: This technique is particularly suitable for joining oxidised materials. The brazing process takes place in a controlled atmosphere to prevent oxidation of the materials being brazed. Furnace brazing, conducted in a vacuum furnace, eliminates the need for flux during the brazing process, streamlining the post-braze cleaning process. Additionally, the temperature at each stage of the vacuum brazing process can be precisely controlled using computer programs, ensuring the production of high-quality brazed parts.

- **Laser Brazing**: Widely used in automotive manufacturing, laser brazing involves using a laser to melt the wire-form filler metal. Similar to laser beam soldering, laser brazing provides highly concentrated heat, resulting in minimal thermal deformation. This property makes it ideal for joining lightweight, appearance-conscious, and rigid components. As a result, laser brazing is commonly employed in joining automotive roofs, side panels, and trunk lids, where precision and aesthetic considerations are critical.

Achieving a high-quality braze joint hinges on four crucial factors that significantly enhance brazing quality, manufacturing productivity, and operator safety.

1. Proper Cleaning and Protection of Parts: Essential to the brazing process, regardless of the method employed, is the proper cleaning and protection of parts. This aims to prevent oxidation and contamination during brazing. Failure to address this aspect can result in oxidation-related defects, compromising joint integrity. Addressing this issue may involve advanced cleaning machines and stricter working environment rules, which fall beyond the scope of the study.

2. Design of the Parts: Thorough capillary actions into the joint depend on the design of the parts, requiring meticulous attention to joint, clearance, and fixture

Table 2.1. A list comparing the advantages and disadvantages of brazing.

Advantages	**Disadvantages**
Brazed parts suffer less from warping or overheating: the brazing process is operated at a lower temperature (compared with the welding process) which leads to lower possibilities of damaging base metals.	For non-permanent joint joining: the brazing process produces permanent joints which are usually irreversible and hard to restore to the previous form after joining. Mechanical fastening is enough and might be more economical and convenient than brazing in this particular situation.
Brazed joints are ductile: brazed joints can withstand more shock and vibration than other joining methods which property can be used in joining high sealing requirements materials.	For low strength joint joining: similarly, brazing can provide joints with large tensile strength. Joint has low demand on strength or leak tightness might consider other more economical joining methods.
Brazing can join dissimilar materials: for example, brazing is capable of joining ceramic to metal.	
The brazing process is relatively easy: brazing can be fast while it still dependent on human worker's skills.	
Brazing is an economical process.	
Brazing can produce strong joints: brazing can provide a stronger joint than the metal (nonferrous metals and steel) to be joined.	

design. Poor design choices, such as wide gaps or uneven joint clearances, can hinder capillary action and lead to brazing failure. Attention to design details is crucial, and advancing this factor involves process parameter optimisation, akin to an engineering problem.

3. Even and Precise Heating of Parts: Uneven heating can impede the flow of brazing filler metal, leading to defects such as incomplete penetration. Achieving precise and controlled heating is essential for the production of robust and reliable brazed joints. Enhancements in this area can be realised through the implementation of more accurate control algorithms and integrated sensor technology, benefiting from mature solutions available in both industrial and academic domains [6, 7] and related technologies, such as welding.

4. The Pasting of Brazing Filler Metal (BFM). The amount and location of BFM deposition influence the heating time required for the assembly and the weight of the joints. Manual brazing involves feeding the filler metal during the heating process adjacent to the joint, while automated brazing methods and furnace brazing may apply the filler metal before the in-phase brazing process [5]. However, industry solutions and academic contributions to this factor remain limited, primarily due to challenges in measuring and recording dynamic interactions between parts, joint designs, materials, and BFMs during dispensing. Both the accuracy and repetitivity of the robot and the experience of humans are essential, making this problem non-trivial. Combining robotics and AI has great potential in this process, as seen in its successful application in robot perception [8, 9, 10], control [11, 12, 13], and collaboration with humans [14, 15, 16].

To illustrate the integration of human expertise and robotic capabilities in brazing processes, Fig. 2.1 presents an example implementation of intelligent digitized brazing. This particular approach demonstrates one possible workflow encompassing three distinct phases: pre-pasting preparation, pasting execution, and post-process analysis. The example utilizes QR codes for process information management and incorporates both human demonstration and automated execution paths depending on task complexity. While this implementation shows one potential solution for combining human expertise with robotic precision, numerous other approaches and configurations are possible depending on specific application requirements and available technologies.

2.3 Intelligent Manufacturing Overview

After identifying the specific gap in the brazing landscape and recognising the substantial benefits that robotics and AI technology can bring to the braze pasting process, a fundamental question arises: How can these technologies contribute to its enhancement? This section delves deeper into this question through an exhaustive review of related technologies.

Intelligent manufacturing, often referred to as Industry 4.0, represents a paradigm shift in the way products are designed, produced, and delivered. At its core, it entails the seamless integration of cutting-edge technologies, data-driven processes, and human expertise to create a highly efficient and adaptable manufacturing ecosystem [17]. The significance of intelligent manufacturing cannot be overstated. It offers a means to elevate production processes to unprecedented levels of efficiency, quality, and flexibility. By harnessing the power of automation, data analytics, and human-machine collaboration, manufacturers can address the demands of a dynamic market, reduce costs, and enhance their global competitiveness. In recent years, the advancement of computer vision and deep learning algorithms has spurred technological innovation, particularly in the development of robust camera vision systems for rapid and precise welding. A notable example is the hybridisation of extreme learning machines and genetic algorithms, proposed in 2016 by Rong et al.

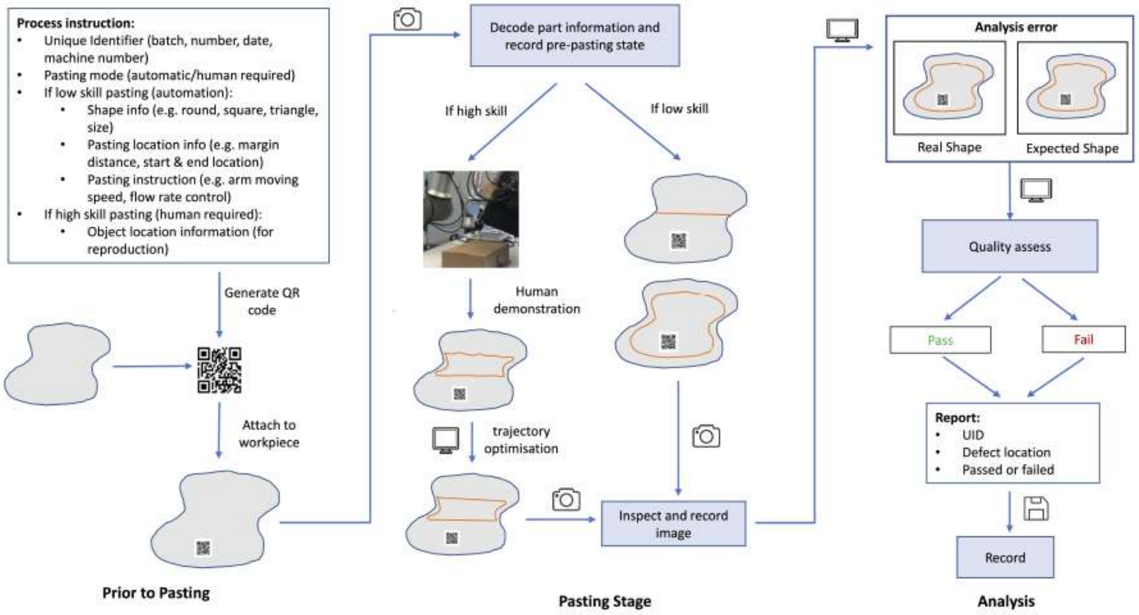

Process instruction:
• Unique Identifier (batch, number, date, machine number)
• Pasting mode (automatic/human required)
• If low skill pasting (automation):
• Shape info (e.g. round, square, triangle, size)
• Pasting location info (e.g. margin distance, start & end location)
• Pasting instruction (e.g. arm moving speed, flow rate control)
• If high skill pasting (human required):
• Object location information (for reproduction)
Generate QR code
Attach to workpiece
Prior to Pasting
Decode part information and record pre-pasting state
If high skill
If low skill
Human demonstration
trajectory optimisation
Inspect and record image
Pasting Stage
Analysis error
Real Shape
Expected Shape
Quality assess
Pass
Fail
Report:
• UID
• Defect location
• Passed or failed
Record
Analysis

Figure 2.1. An example implementation of digitized brazing process integration. The illustrated workflow shows one possible approach divided into three phases: Prior to Pasting (left), showing process encoding through QR codes; Pasting Stage (centre), demonstrating both human demonstration and automated execution paths; and Analysis (right), featuring vision-based quality assessment. This represents one of many possible configurations for integrating human expertise with robotic capabilities in brazing applications.

[18]. This innovative approach leverages published experimental data to construct a predictive model for the top and bottom dimensions of weld beads. By finely controlling welding parameters such as speed and wire feed rate, the extreme learning machine accurately forecasts the bead profile's top and bottom width. The genetic algorithm component further refines predictions, yielding minimised relative errors and improved quality.

- **Collaborative Robots**

In recent years, the domain of robotics has undergone a remarkable transformation. The conventional view of robots as rigid, pre-programmed machines that perform repetitive tasks in controlled environments has evolved into a far more sophisticated and adaptable paradigm. A new generation of robots, often referred to as "smart robots" or "learning robots," has emerged, setting the stage for a technological revolution in the field of robotics [19]. These learning robots, equipped with advanced machine learning and artificial intelligence techniques, possess the remarkable ability to acquire knowledge, adapt to changing environments, and make informed decisions. They can navigate complex and unstructured surroundings, interact seamlessly with humans, and perform tasks that require both precision and flexibility. This transformation is largely attributed to the advent of robot learning, a multidisciplinary field that marries the principles of robotics, machine learning, and artificial intelligence.

Robots can be broadly classified into two subsets: traditional robots and collaborative robots, also known as cobots. Traditional robots are designed to replace human workers in performing tasks. However, they face challenges related to safety regulations, which require them to be enclosed within cages, separating them from human operators. This isolation results in a discrete production process, which may not be optimal for processes that require close human involvement. Cobots, on the other hand, have emerged as part of the Industry 4.0 concept, providing manufacturing with intelligence and flexibility. They allow humans and robots to work in close proximity within a shared workspace [20]. The concept of cobots was initially introduced in [21], and they are defined as "robots intended for direct human-robot interaction within a shared space or where humans and robots are in close proximity."

Traditional industrial robots, designed for high-volume manufacturing, operate in isolated environments due to safety concerns and regulatory requirements. In contrast, collaborative robots (cobots) represent a distinct category of robotic systems engineered specifically for direct human-robot interaction within shared workspaces [22]. While both types serve manufacturing purposes, cobots incorporate advanced sensing and control features that enable safe operation alongside human workers without the need for physical barriers. The key distinction of cobots lies in their inherent safety features and adaptability to human presence. These include force/torque monitoring, collision detection, and speed/separation monitoring, all regulated under ISO/TS 15066:2016 safety standards. Unlike traditional industrial robots that prioritize speed and payload capacity, cobots emphasize safe interaction, teachability, and flexible deployment. This makes them particularly suitable for tasks requiring frequent reprogramming or human oversight.

In modern manufacturing environments, cobots serve as complementary tools rather than replacements for human workers. They excel in applications where complete automation is either impractical or undesirable, such as small-batch production or processes requiring human judgment. The integration of cobots enables manufacturing systems to leverage both the consistency of automation and human cognitive capabilities. When implemented in Human-Robot Collaboration (HRC) scenarios, these systems can achieve higher flexibility than fully automated solutions while maintaining better consistency than purely manual operations. Table 2.2 provides a detailed comparison between traditional industrial robots and cobots, highlighting their distinct characteristics and applications.

Table 2.2. Comparison between industrial robots and collaborative robots.

	Industrial robot	**Collaborative robot**
Flexibility	Not flexible. Fixed installation. Changes on a product line are limited and require a long time for reprogramming and testing.	Flexible. Allows frequent changes on the product line and can quickly adapt to different products. Suitable for small batch production and customised services.
Efficiency	Fast.	Slow.
Cost	Expensive. Requires additional expenditure on installation, design, customer training, and after-sales maintenance.	Relatively affordable.
Safety	Separate workspace. Usually fenced.	Can work alongside humans (Need to be designed in accordance with ISO/TS 15066:2016).
Deployment	Time-consuming design, setup, and customer training.	Easy to program. Many cobots can be programmed or reprogrammed in a short time using methods like hand guiding, coding, or 3D visualisation technology.
Intelligence	No self-awareness. Limited failure detection capabilities.	Real-time monitoring of load, location, and tactile pressure [23]. Easily incorporates artificial intelligence technologies.
Application scenario	Repeated and relatively fixed manufacturing lines with high demands for accuracy and payload.	Dynamic product lines with frequent changes. Particularly suited for procedures involving human workers and customised or intelligent systems.
Payload	High.	Low.

In summary, collaborative robots represent a significant advancement in the field of automation, offering flexibility, safety, and intelligence to manufacturing processes. They bridge the gap between human workers and automated systems, creating a harmonious and efficient working environment.

- **Industry-Academia Gap**

Large disparities persist between the industrial and research sectors in the field of collaborative robots. Industrial environments impose stricter constraints, while most research settings fail to adequately demonstrate the adaptability and versatility of cobots within partially unstructured work environments [24]. In [25], four critical gaps in current cobot research for industrial settings are identified: human safety, intuitiveness, adaptability, and employability.

- **Human Safety:** The paramount concern in industrial applications involving cobots is the safety of human operators. These environments typically necessitate close human-robot interaction, making comprehensive safety protocols essential.

- **Intuitiveness:** For cobots to achieve widespread adoption, they must be accessible to users without extensive technical training. Innovations like the integration of Human-Robot Collaboration (HRC) with mixed reality (MR) technologies, as explored in [26], show promise. By blending virtual and augmented realities, MR allows for an intuitive, interactive interface between humans and robots, potentially lowering the barriers related to training and programming.

- **Adaptability:** The capacity of cobots to adjust seamlessly to various tasks and environments is critical for their effective deployment across different industrial sectors. This flexibility not only enhances their utility but also maximises their operational efficacy, underscoring the need for systems that can easily transition between tasks with minimal downtime.

- **Employability:** Bridging the gap between theoretical cobot concepts and their practical application is crucial for realising their full potential in industrial settings. Digital Twin (DT) technology, a cornerstone of Industry 4.0, plays a vital role in this context by providing realistic simulations for training, testing, and validating cobot systems. Despite the high fidelity of these simulations, discrepancies between virtual and physical environments can pose challenges, especially in fine-tuning reinforcement learning algorithms, thus pointing to an essential area for further research and development.

This thesis directly addresses the notable gaps in the adaptability and intuitiveness of Cobots within the specific context of brazing operations. Current advancements in cobot technology, while promising, have not fully met the nuanced requirements of brazing tasks, which demand high levels of precision and flexibility. The research presented here aims to narrow this gap by developing a framework that improves cobots' ability to interact with human operators in a more intuitive manner and to adapt efficiently to the diverse challenges of brazing. By concentrating on these key issues, the thesis proposes a pragmatic approach that seeks to enhance the practical deployment of cobots in brazing, thereby contributing to the broader field of robotic automation in specialised manufacturing environments. This focused endeavours is intended to provide a solid foundation for the future integration of Cobots in complex tasks, emphasising a realistic assessment of current technologies and their potential for improvement.

2.4 Robot Control Overview

In this section, a brief introduction to different robot control methods are provided, highlighting their advantages, disadvantages, and application scenarios. In general, there are three ways to generate robot control: offline programming, Learning from Demonstration (LfD), and teleopearation.

Offline programming is one of the most basic function of robot control which normally programmed by scripts, through teach pedant or GUI. In comparison, LfD is a popular method for robots to learn new skills quickly, where a human operator manually guides the robot through the desired task. LfD serves as a significant part in robot automation that imparts human knowledge and skills to robots. LfD offers the advantage of intuitive and natural interaction, as the human can directly demonstrate the task using their own expertise. This method is particularly suitable for complex tasks that are challenging to program explicitly. However, LfD requires skilled human operators and may be time-consuming, as the robot needs to learn from multiple demonstrations to generalise the task. Additionally, the accuracy of reproducing the demonstrated task can be influenced by variations in human demonstrations. It is a compelling alternative to offline programming methods in complex environments where the task cannot be easily scripted or optimised [27].

There are two main categories of LfD: model-based methods, such as sampling-based motion planning [28] and trajectory optimisation [29], and model-free methods such as recurrent neural network-based methods [30, 31], CNN-based methods [32, 33, 34], and RL-based methods [35, 36]. While model-based methods have shown to be reliable and sample-efficient, achieving excellent results [37, 38], they are still limited by human performance. On the other hand, model-free methods have the ability to explore the state space and sometimes generate control policies that surpass human performance [39]. However, like other machine learning techniques, these methods also face challenges such as sample inefficiency [40], increasing difficulty of the task due to large action dimensionality [41], and poor performance in noisy demonstrations [42]. A basic demonstration of using LfD is given in Fig. 2.2, showcasing the seamless translation of human-guided expertise into robotic action in a box-opening tasks. Targeting the limitations of the two categories, recent studies have proposed new techniques for LfD. For instance, [43] proposed the idea of dividing any robot task into reaching and interaction stages, respectively. They found that humans are not very good at producing quality demonstrations when reaching, but excel at interaction, which is an embodiment of human knowledge. Following this idea, it is fair to say that a reaching method without specifying the target location is the first step toward

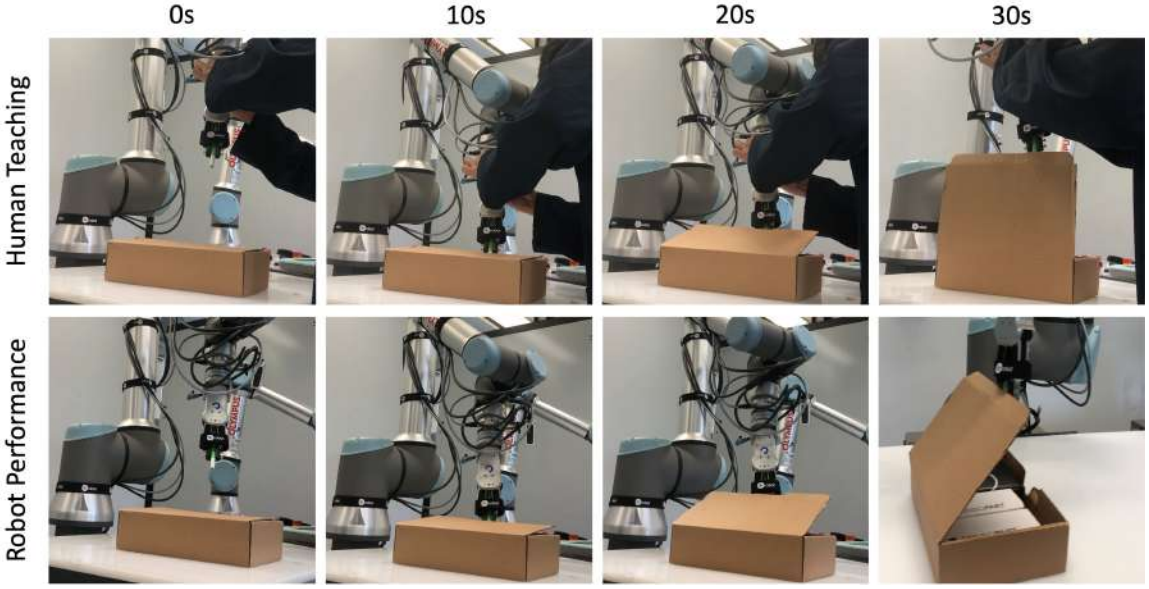


Figure 2.2. Demonstration of the box-opening task using Programming by Demonstration (LfD). The top panel shows a human operator guiding the robot through the task, while the bottom panel displays the precise execution of the task by the robot. Sequential images capture different stages of the process, from left to right.

true robot autonomy. A segmentation-based visual servoing method was introduced in the follow-up work [44], which does not rely on further training after a one-shot demonstration and can be deployed in new environments immediately.

On the other hand, teleoperation controls the robot remotely by a human operator. Teleoperation provides real-time control and immediate feedback, making it suitable for tasks that require human intervention or in hazardous environments. A teleoperation platform with two Phantom Omni hand controllers for preliminary testing and validation of remote bilateral control is established as in Fig. 2.3. It allows the human operator to leverage their perception and decision-making abilities, ensuring precise and adaptable robot behaviour. However, teleoperation relies heavily on the operator's skills and may not be scalable for complex tasks or long-duration operations. Latency and communication constraints can also affect the performance of teleoperated robots.

Apart from the aforementioned three mainstream approaches, there also exists hybrid approaches, combining multiple control methods, have also emerged. These approaches aim to leverage the strengths of different methods to overcome their individual limitations. For example, a hybrid approach could involve initial teleoperation for task demonstration, followed by LfD to refine the learned behaviour. Such hybrid methods can offer a balance between human expertise and autonomous learning, enabling efficient and adaptable robot teaching. Table 2.3) provides a comparison of different robot control methods.

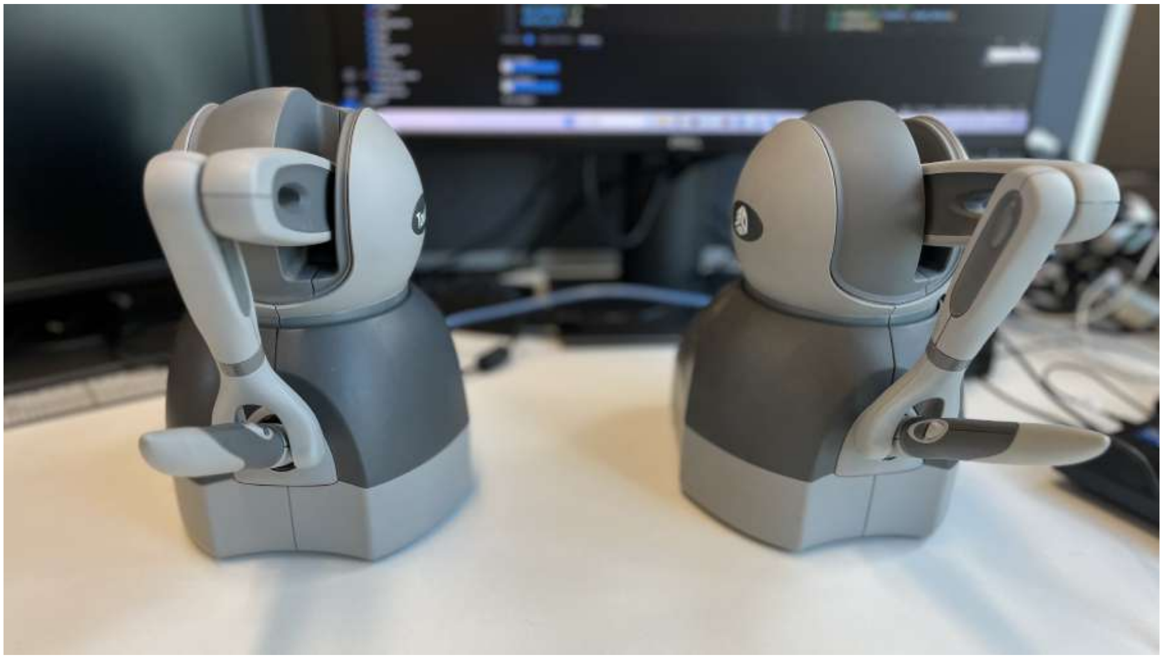


Figure 2.3. Dual hand-controller teleoperation platform used for preliminary teleportation testing and validation.

Table 2.3. Comparison of Robot Control Methods

Teaching Method	Advantages	Disadvantages	Applications
Offline Programming	Direct, suitable for simple and repeated tasks (with parameters clearly defined)	Requires programming skill, time-consuming, accuracy influenced by variations in demonstrations	Extensive experience learning/training time
LfD	Intuitive and natural interaction, suitable for complex tasks	Requires skilled operators, time-consuming, accuracy influenced by variations in demonstrations	Complex task programming, skill transfer
Teleoperation	Real-time control, immediate feedback, precise and adaptable behaviour	Relies on operator skills, may not scale for complex tasks, latency and communication constraints	Human intervention, hazardous environments
Hybrid Approaches	Balance between human expertise and autonomous learning, efficient and adaptable teaching	Complexity in combining methods, potential integration challenges	Task refinement, leveraging multiple methods

2.5 Reinforcement Learning Basis

Reinforcement Learning (RL) represents a distinct approach within the broader spectrum of machine learning techniques, characterised by its focus on learning optimal actions through trial and error to maximise rewards in a given environment. Unlike traditional machine learning algorithms, which often rely on large datasets to train models in a supervised manner, RL requires no prior knowledge about the system model, making it particularly suited for applications where explicit examples of correct behaviour are not available. This feature sets RL apart from other machine learning strategies, where the emphasis is on pattern recognition and prediction based on historical data. In comparison to deep learning, another subset of machine learning known for its ability to process and learn from large amounts of unstructured data, RL excels in scenarios requiring decision-making and policy optimisation. While deep learning algorithms excel in identifying patterns and making predictions from complex inputs, they are less equipped to directly address problems of sequential decision-making under uncertainty—a core strength of RL [45].

The application of RL in highly complex and dynamic environments, such as those encountered in robot manipulation and process control, has demonstrated substantial benefits. Research indicates that RL algorithms not only offer superior stability and computational efficiency compared to traditional optimal control algorithms [46] but also adapt effectively to changes in the environment, optimising actions based on the feedback received. This adaptability is a key advantage of RL, enabling it to outperform both conventional machine learning and deep learning algorithms in tasks that involve complex sequences of decisions and actions.

Given these considerations, RL is chosen for visual servoing in robotics due to its inherent ability to iteratively improve and find optimal solutions in dynamic environments without the need for predefined models. This capability makes RL particularly appealing for visual servoing applications, where the robot must adapt to varying visual inputs and physical conditions to perform tasks accurately. The choice of RL is justified by its proven track record in enhancing both the efficiency and effectiveness of robotic control systems, surpassing the capabilities of other machine learning and deep learning approaches in this specific context. The advantages of RL algorithms are summarised as follows:

- Resource Efficiency: RL algorithms have low online computational complexity [47].
- Inherent Adaptability: RL algorithms do not rely on predefined models.
- Handling Complexity: RL algorithms can process high-dimensional sensory inputs [48].
- High Performance: RL algorithms are capable of achieving results close to optimality.
- Simplified Hyperparameters: RL algorithms typically have fewer hyperparameters and are less sensitive to tuning.

Typically, RL algorithms are applied in complex environments that can be described by a Markov Decision Process (MDP). An MDP is a memory-less discrete stochastic process that models state transitions influenced by agent policies and environmental stochasticity. While RL holds great promise, it demands substantial training data, which can be challenging to generate or collect for robots due to safety hazards and energy costs associated with interactions in the physical world. However, recent research has explored methods to overcome data scarcity, including fine-tuning and boosting the training process through human demonstrations [40, 49, 50, 51]. Additionally, conventional data augmentation techniques used in deep learning can be adapted for training RL algorithms.

When designing reinforcement learning algorithms normally the following four parts should be considered:

- **Observation and State:** The process of observation and state plays a pivotal role in the design of reinforcement learning algorithms. This process represents the state of the agent within the Markov Decision Process (MDP). After each action is taken and the environment changes, observations are made to update the agent's understanding of its surroundings. These observations are fundamental, providing the agent with the information necessary to make informed decisions and optimise its policy.

- **Action:** In the domain of reinforcement learning, actions serve as the cornerstone of agent behaviour. They embody the choices made by the agent and dictate its interactions with the environment. The research is centred around the strategic design and execution of actions that guide the agent toward its predefined objectives.

- **Reward Function:** The design of the reward function profoundly influences the convergence of the reinforcement learning algorithm and represents a pivotal challenge in the field of robotics. This function acts as the guiding star for policy optimisation, offering feedback on the agent's performance. To address the issue of sparse binary rewards and enhance learning efficiency, this research have pioneered a data-driven reward function. This approach rectifies the sample efficiency problem by providing the policy with timely and meaningful feedback. It draws inspiration from human behaviour when reaching for an object with a specific gesture. Initially, coarse movements are executed when the target is distant, with fine adjustments coming into play as proximity increases, ensuring precision. While the reward function is not easily hand-crafted, it encapsulates the essence of the task's objectives. It provides continuous and task-aligned guidance for policy optimisation, effectively forming a closed loop with the agent's actions. This synergy significantly expedites the training process by ensuring the policy receives immediate feedback, thereby staying aligned with the task's objectives.

In recent years, reinforcement learning algorithms have emerged as promising methods for addressing this challenging control, decision making and optimisation tasks.

- **Control:** Chen et al. [52] demonstrated the use of an expectation maximisation-based RL algorithm to control tendon-driven serpentine manipulators, which excel in minimally invasive surgical tasks. These manipulators navigate confined spaces through keyhole incisions but are challenging to control due to non-linearities and model uncertainties. The proposed algorithm combined Learning from Demonstration and EM RL to teach these manipulators surgical tasks. Rajeswaran et al. [53] addressed the control of multi-fingered hands, a complex problem due to high dimensionality and numerous potential contacts. They introduced a combination of RL and imitation learning called demo augmented policy gradient (DAPG). DAPG effectively scaled up to complex manipulation tasks with a high-dimensional 24-degree-of-freedom hand in a simulated environment. Importantly, this algorithm integrated a small number of human demonstrations, significantly reducing the required robot learning time. Brito et al. [54] applied an RL algorithm for inspection trajectory control, allowing operators to interact and change action paths in real-time, improving actions iteratively. Schmidt et al. [55] introduced an architecture that enables the implementation of an RL framework on Programmable Logic Controllers (PLCs). This approach involves coupling non-real-time learning frameworks with the real-time environments of PLCs. Consequently, the need for external interfaces from the production unit is eliminated. This not only eliminates potential safety hazards associated with transferring models via internet connections but also reduces integration efforts. Such a solution holds the potential to save on equipment replacement costs and minimise downtime for factories aiming to upgrade their systems to align with the Industry 4.0 framework.

- **Decision Making:** Autonomous decision-making has become a significant area of focus within the realm of robot teleoperation and visual servoing, driven by the need for robots to navigate and adapt to a variety of tasks and environmental conditions with minimal human intervention [56, 57, 58]. This advancement enables robots to not only follow predefined tasks but also to dynamically adjust to changes in real-time, thereby increasing their utility in diverse operational scenarios. RL acts as a key method for developing such autonomous capabilities, which allows robots to learn and refine decision-making strategies through trial-and-error interactions with their environment. RL is particularly effective in motion prediction tasks [59, 60, 61], where it helps machines to derive optimal actions based on historical outcomes, thus enhancing their ability to make informed decisions in future scenarios. However, the practical application of RL and autonomous decision-making faces several challenges, notably in dealing with environmental uncertainties. These can include sensor inaccuracies and unpredictable movements within the robot's operational space. The reinforcement learning model's exploration-exploitation approach is designed to navigate these uncertainties, improving decision-making capabilities by learning from variable conditions. Despite these advancements, reliance on RL for robot teleoperation is not without its drawbacks. The approach demands significant volumes of training data to reach an acceptable level of performance, and in complex environments, developing an effective strategy might require extensive and costly data collection efforts [62]. Additionally, the training phase for

RL algorithms can be time-consuming, potentially limiting their application in situations demanding immediate responsiveness. Furthermore, there's a risk that RL models could develop strategies that are either unsafe or ethically questionable, especially when deployed in real-world settings [63, 64]. These concerns highlight the necessity for careful consideration of safety and ethical standards in the development and implementation of autonomous decision-making systems for robots

- **Optimisation:** Li et al. [65] applied multi-agent game theory in a non-zero-sum game framework to optimise performance in large-scale unknown industrial processes. Local RL-based optimisation addressed individual production index sub-problems, contributing to overall plant-wide performance. This strategy effectively navigated complex trade-offs in industrial settings, considering factors like cost, maintenance, quality, time, and labour. However, the inherent black-box nature of RL algorithms poses challenges in safety-critical industrial scenarios, requiring attention to the predictability and trustworthiness of robots. Rossi et al. [66] proposed an RL-based robot arm path planning approach aimed at reducing task completion time in collaboration with humans. They showed that unsupervised learning paradigms could produce similar or better results compared to annotated motion datasets, saving time and effort. Reinforcement learning was compared to other methods such as Programming by Demonstration (LfD), which, although intuitive, can be time-consuming. Zhu et al. [67] introduced the Dynamic Actor-Advisor Programming (DAAP) algorithm, aiming to minimise both task costs and constraint risks concurrently, with a focus on sample efficiency, safety, and scalability. Integrating this approach with lower-level safety measures shows promise for deploying RL in industrial settings. Jiang et al. [68] used RL to address optimal selection of process control inputs in the flotation process. The algorithm ensured optimal tracking of operational indices while maintaining inputs within specified bounds. Khader et al. [69] applied Q-learning-based RL to optimise surface mount technology (SMT) in PCB manufacturing by controlling stencil printing parameters. These implementations demonstrate RL's potential in enhancing process control and optimisation in industrial contexts.

In summary, when implementing RL algorithms in industrial settings, it's crucial to address the following challenges:

- **Observation Space Limitations:** The observation space of RL agents in real-world scenarios can be significantly smaller than the state space. Practical sensors may provide only partial state information.

- **Reward Design Complexity:** Designing appropriate reward functions for RL algorithms in physical environments can be intricate. Sparse rewards in real-world settings can substantially impact the convergence of RL algorithms.
- **Stability and Theoretical Framework:** The theoretical stability of RL algorithms is an ongoing area of development. Ensuring the robustness and safety of RL-based systems in industrial applications remains a concern.
- **Safety Concerns:** RL algorithms can sometimes act unpredictably, which is a significant safety consideration in critical industrial environments.

2.6 Simulation for Robot Learning

Simulated environments are gaining increasing attention for providing virtual representations of factory environments. As mentioned earlier in the literature review chapter, this virtual mirroring concept is significant for the training, testing, and validation of robots due to its safety advantages and its ability to generate large amounts of data without exposing real robots to potentially slow, expensive, and dangerous exploratory learning processes. However, these virtual environments have limitations. While they offer valuable data for analysis, they cannot perfectly replicate all the details of the physical world. This difference can significantly affect robot learning such as reinforcement learning algorithms. For example, in a simulation, an agent applying a reinforcement learning algorithm can observe the system as a whole, using this global observation as its own state to make decisions. However, in a real-world scenario, observations come from various sensors, each providing a different aspect of environmental information. Integrating data from these sensors forms the agent's observation. This means that the observation dimension is limited by the number and characteristics of sensors, providing only a partial observation compared to the all-encompassing view in a simulated environment. In this section, two main simulators: Pybullet and Gazebo, are introduced, each serves for specific purposes and tasks.

2.6.1 Simulation Setup

- **Docker**

To enhance the experiment reproducibility, software isolation, and ease of deployment for other research, a containerisation technology called Docker is used in the experiment. This section delves into the integration of Docker within the experimental setup, highlighting the benefits it brings to the research.

Benefits of Docker in robotics research:

- Experiment Reproducibility: Docker enables the encapsulation of the entire experimental environment, including all dependencies and software libraries, into a single container. This ensures the exact reproduction of experiments, mitigating compatibility issues and minimising the ”it works on my machine” problem.

- Isolation: Docker containers operate in isolated environments, preventing conflicts between software components. This is particularly valuable when multiple experiments with different requirements share the same hardware setup.

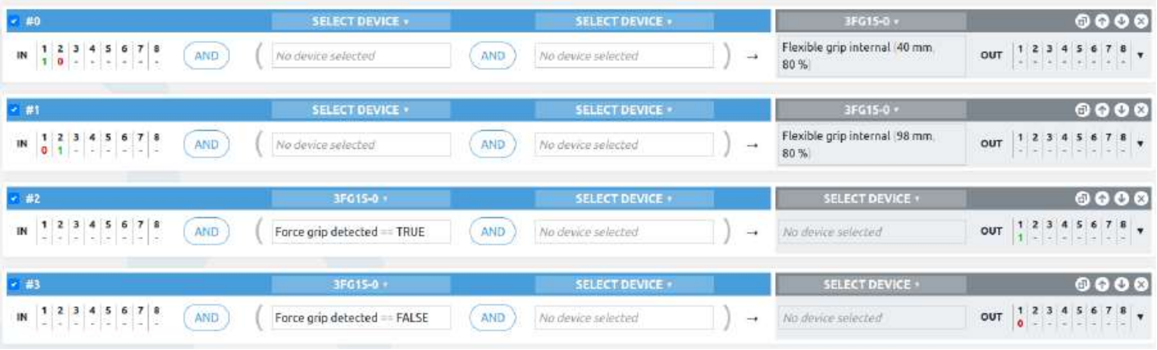
- Portability: Docker containers, being lightweight, can be effortlessly transferred between different systems, allowing seamless transitions of experiments across various computational platforms.

- Version Control: Docker facilitates version control of experiment environments. By creating and maintaining specific Docker images for each iteration, changes can be precisely tracked, and reverting to previous setups becomes feasible.

In the research conducted, Docker plays a crucial role in maintaining the integrity and reproducibility of the experiments. The following practices are adopted:

Firstly, containerised simulation environments are employed. Docker is used to create isolated simulation environments for the testing and validation of algorithms. These environments encapsulate simulation software, device drivers, and control modules, ensuring consistent simulations across various systems. Secondly, emphasis is placed on Dependency Management. All software dependencies, including ROS packages, libraries, and drivers, are containerised using Docker. This guarantees that experiments can be executed on any Docker-supported system, irrespective of the underlying system’s configuration. Finally, Experiment Packaging is emphasised. Each experiment is encapsulated as a Docker image, encompassing the essential codebase, datasets, and configuration files. This approach facilitates researchers in effortlessly pulling these images and launching experiments without concerns about compatibility issues.

However, it is important to acknowledge Docker’s limitation in providing serial port support, presenting challenges for devices with high-frequency data exchange or stringent communication requirements. For instance, difficulties were encountered when attempting to use the Phantom Omni, a haptic device with six degrees of freedom (DoF), within Docker or Windows Linux Subsystem (WSL). While techniques such as USB mapping (with usbipd) facilitated a connection between the device and the system inside the Docker container, they introduced significant latency in the data exchange process. Consequently, the device would disconnect within a minute during testing. Unfortunately, this issue remains unresolved as of the writing of this thesis.



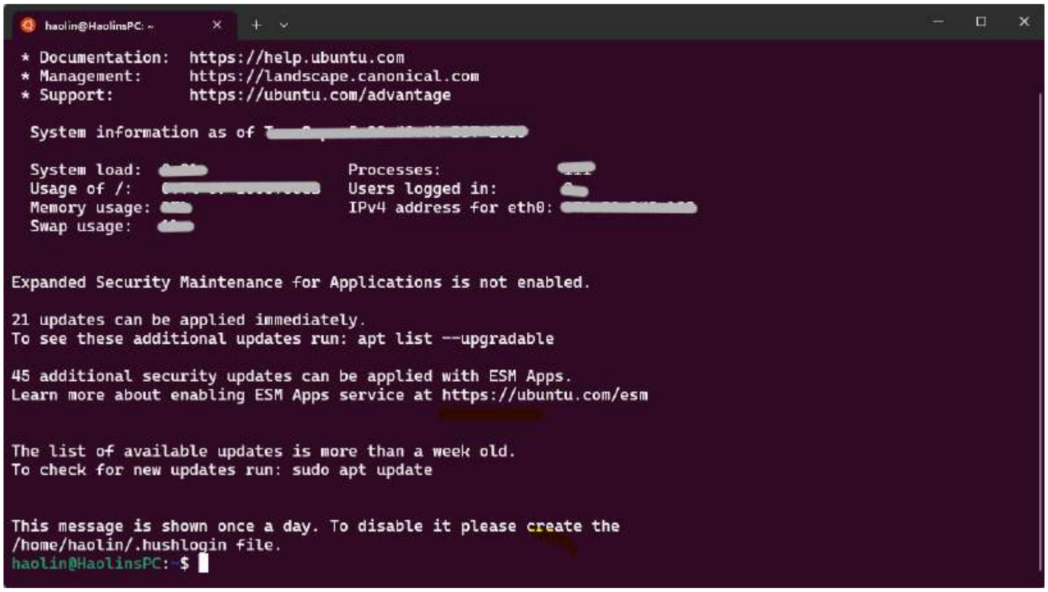
#0 — SELECT DEVICE — SELECT DEVICE — 3FG15-0
IN 1 2 3 4 5 6 7 8: 1 0 - - - - - - AND (No device selected AND No device selected) → Flexible grip internal (40 mm, 80 %) OUT 1 2 3 4 5 6 7 8: - - - - - - - -

#1 — SELECT DEVICE — SELECT DEVICE — 3FG15-0
IN 1 2 3 4 5 6 7 8: 0 1 - - - - - - AND (No device selected AND No device selected) → Flexible grip internal (98 mm, 80 %) OUT 1 2 3 4 5 6 7 8: - - - - - - - -

#2 — 3FG15-0 — SELECT DEVICE — SELECT DEVICE
IN 1 2 3 4 5 6 7 8: - - - - - - - - AND (Force grip detected == TRUE AND No device selected) → No device selected OUT 1 2 3 4 5 6 7 8: 1 - - - - - - -

#3 — 3FG15-0 — SELECT DEVICE — SELECT DEVICE
IN 1 2 3 4 5 6 7 8: - - - - - - - - AND (Force grip detected == FALSE AND No device selected) → No device selected OUT 1 2 3 4 5 6 7 8: 0 - - - - - - -

Figure 2.4. Gripper control program in Compute Box.



```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of [redacted]

  System load:  [redacted]              Processes:             [redacted]
  Usage of /:   [redacted]              Users logged in:       [redacted]
  Memory usage: [redacted]              IPv4 address for eth0: [redacted]
  Swap usage:   [redacted]


Expanded Security Maintenance for Applications is not enabled.

21 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

45 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


This message is shown once a day. To disable it please create the
/home/haolin/.hushlogin file.
haolin@HaolinsPC:~$
```

Figure 2.5. Example of WSL.

- **Windows Subsystem for Linux**

The Windows Subsystem for Linux (WSL) is a compatibility layer developed by Microsoft to run a Linux kernel interface and command-line tools directly on Windows 10 and 11. WSL provides a convenient way for developers to work with Linux-based tools and applications without the need for a dedicated Linux virtual machine or dual-boot setup. Here are reasons why WSL and Docker solution is desired in the experiment: firstly, WSL allows Windows users to execute Linux commands and run Linux applications directly on their Windows machines. It supports a wide range of Linux distributions (Ubuntu 20.04 is used for experiment). Secondly there is no virtualisation overhead. Unlike traditional virtual machines, WSL does not require resource-intensive virtualisation. It runs Linux binaries directly on the Windows kernel, resulting in minimal performance overhead, which is particularly advantageous when employing resource-intensive tools such as Gazebo and RViz. In addition, it seamlessly integrates with the Windows file system. This means that codes can be assessed inside windows system directly from the WSL command line, making it easy to work with files from both environments. In the experimentation, WSL played a crucial role in bridging the gap between Windows and Linux environments. Despite having access to an Ubuntu 20.04 computer, WSL command-line capabilities and the ability to execute Linux-based tools seamlessly alongside Windows applications are desired. This integration proved to be of paramount importance in the comprehensive management and orchestration of the diverse components comprising the experimental setup. These components encompassed a spectrum of devices, ranging from Windows-based to Linux-based, and WSL served as the unifying platform that streamlined their interactions. An illustrative depiction of WSL is presented in Fig. 2.5, offering a visual glimpse into the operational dynamics of this innovative compatibility layer.

2.6.2 Simulators Comparison

Gazebo Simulator

In the realm of intricate robot applications, ROS stands out with its suite of software libraries and tools. Gazebo, a simulator within ROS, is employed for the development of both the digital twin platform and an extended teleoperation platform. A Gazebo-based environment within ROS serves as a pivotal arena for generating and collecting robot data, thereby enhancing the richness of the training dataset. Additionally, ROS's moveit! planning environment is instrumental in facilitating complex motion planning, a crucial component of the research (see Fig. 2.9 (c)). Notably, the setup of ROS typically requires a Linux system, and in this research, ROS is implemented on Ubuntu 20.04, equipped with the full desktop version of the noetic release.

PyBullet Simulator

PyBullet, a user-friendly Python module, plays a pivotal role in robot learning. Widely acclaimed for its applications in physics simulation, robotics, and deep reinforcement learning [70]. This versatile toolkit, leveraging the Bullet Physics SDK, excels in loading articulated bodies from formats like Unified Robot Description Format (URDF) and Spatial Data File (SDF). Among its rich functionalities are forward dynamics simulation, inverse dynamics computation, forward and inverse kinematics, collision detection, and ray intersection queries, making it the platform of choice for various tasks in the research. Beyond its core physics simulation capabilities, PyBullet extends support to rendering by employing a

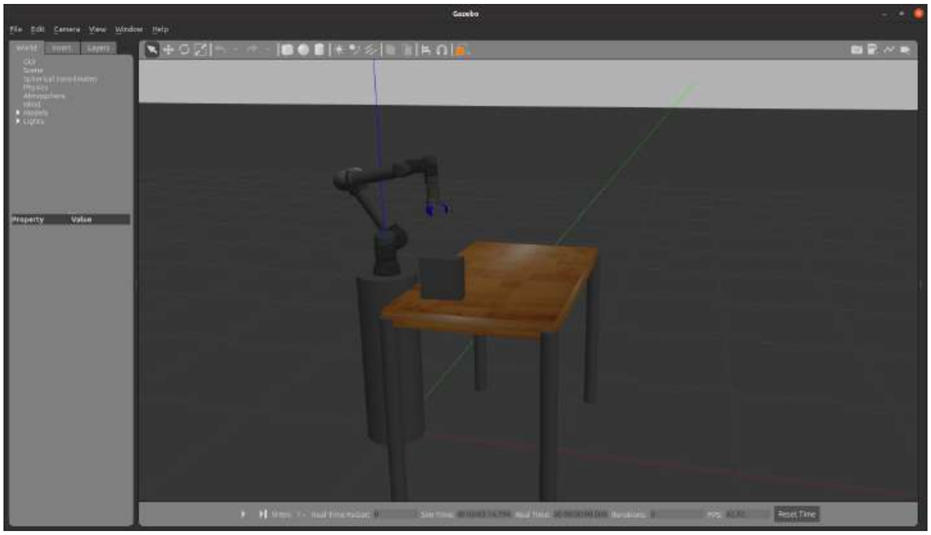


Figure 2.6. The Gazebo digital twin environment, coupled with Docker containers, serves as a bridge between virtual and physical environment, establishing the cornerstone for the integration of digital twins in the research.

CPU renderer and OpenGL visualisation, further accommodating compatibility with virtual reality headsets. Particularly noteworthy is PyBullet's tailored emphasis on reinforcement learning, rendering it uniquely well-suited for swift programming, algorithm training, validation, and testing. This strategic alignment seamlessly harmonises with the objectives of the robot learning segment, prioritising an accessible platform for algorithm training over meticulous rendering accuracy. Moreover, the simulation architecture is able to extend beyond PyBullet, incorporating the Robot Operating System (ROS) platform for a more comprehensive approach to simulation, data generation, and results demonstration. Table 2.4 provides a comprehensive comparison between PyBullet and Gazebo, aiding in the selection of the most suitable platform for specific research or application needs.This research employs ROS to establish a visualisation platform using rviz, providing an interactive environment for algorithm monitoring and refinement.

Table 2.4. Comparison between PyBullet and Gazebo.

Feature	**PyBullet**	**Gazebo**
Simulation Environment	Python-based physics simulator	Extensive robotics simulator within ROS
Community Support	Active community with regular updates and contributions.	Well-established community with extensive documentation and support.
Usage	Ideal for ML research and robotic control development.	Widely used for both research and industry applications, including robotic manipulation and navigation.
Application Areas	Physics simulation, robotics, deep reinforcement learning	Broad spectrum including robotics, autonomous systems, teleopeation and sensor simulation
SDK/Framework	Leverages Bullet Physics SDK, primarily designed for use with Python.	Integrated with ROS, which supports C++ and Python.
Functionality	Forward and inverse dynamics simulation, kinematics, collision detection, ray intersection queries	Multi-robot simulation, physics simulation, sensor simulation, dynamic environment creation
Visualisation	Limited visualisation capabilities	Robust visualisation using rviz
Motion Planning	Basic functionalities	Advanced motion planning through ROS moveit!
Data Generation	Suitable for certain applications	Extensive data generation capabilities within ROS
Real-world Physics Simulation	Focuses on accurate physics simulation for robotic applications.	Aimed at simulating robots in real-world scenarios, including physics and sensors.

2.6.3 Simulation Environment

To facilitate the safe and cost-effective training of robot learning algorithms, a foundational simulation environment has been established. In this rudimentary configuration, the objective is to enable the robot to explore its environment autonomously through visual information. The robot should perform tasks without prior knowledge of the object's characteristics, necessitating randomisation in the type and position of the objects. This simulation leverages the PyBullet simulator to emulate the robot learning task. The PyBullet simulator takes centre stage in emulating this autonomous learning task. Built upon the robust Bullet physics engine, PyBullet facilitates the seamless observation of state information for interactive objects, including task-specific items, the table, and the robot itself. This rich state information, encompassing spatial position, orientation, velocity, and acceleration of task objects, proves instrumental for training data-driven algorithms. It's worth noting that, due to inherent limitations in sensor capabilities regarding quantity and precision, only a subset of this state information can be utilised as input. In this foundational simulation configuration, a UR5e robot equipped with a ROBOTIQ 2F-85 gripper is employed for its user-friendly interface and widespread acceptance in contemporary research. The gripper's control is based on position, offering continuous input between 0 to 1. The robot maintains a fixed spatial position, situated above a worktable where objects are randomly generated. A snapshot of this simulation environment is depicted in Fig. 2.9 (a). This tailored setup is predominantly devised for the reinforcement learning based visual servoing chapter, emphasising the robot's autonomous controlling using visual information. Consequently, an RGB camera, mounted on the gripper and adhering to the eye-in-hand setting [71], is simulated to furnish observation data for the robot's learning endeavours.

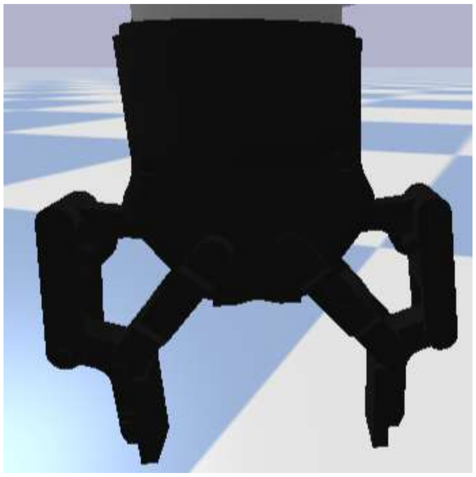


Figure 2.7. Robotiq 2F-85 gripper in simulation.

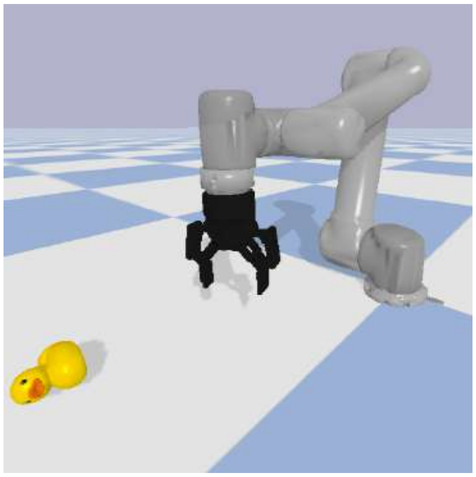
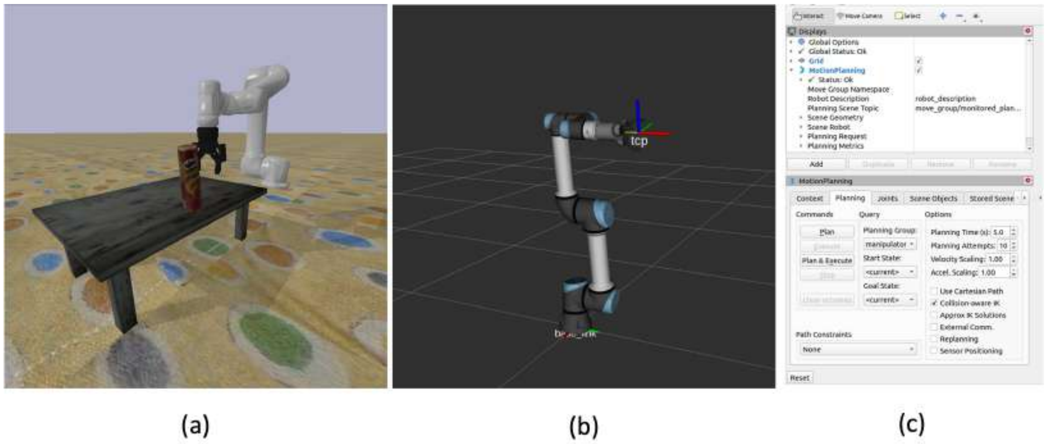


Figure 2.8. UR5e robot arm used in simulation.

2.6.4 Domain Randomisation

Simulations offer a safe and cost-effective means to teach robots, allowing for extensive exploration, learning from trial and error, rapid prototyping and facilitates the development of complex behaviours. However, the challenge lies in accurately transferring the learned behaviours from simulation to the real-world, as the simulation may not capture all the dynamics and uncertainties present in the physical environment. Domain randomisation is an effective technique for domain transfer of



Move Group Namespace
Robot Description: robot_description
Planning Scene Topic: move_group/monitored_plan...
Scene Geometry
Scene Robot
Planning Request
Planning Metrics

Planning Group: manipulator
Start State: <current>
Goal State: <current>
Path Constraints: None

Planning Time (s): 5.0
Planning Attempts: 10
Velocity Scaling: 1.00
Accel. Scaling: 1.00
□ Use Cartesian Path
☑ Collision-aware IK
□ Approx IK Solutions
□ External Comm.
□ Replanning
□ Sensor Positioning

Figure 2.9. Simulation and Visualisation Components: (a) PyBullet simulation environment, illustrating the virtual setting for experimentation. (b) Rviz visualisation of the robot, providing a dynamic and interactive display. (c) MoveIt motion planning.

learnt knowledge from simulation environment to real-world environment [72]. The domain randomisation includes randomising visual shapes of the scene, lightning, camera position, camera parameters and etc.. With the help of domain randomisation, the learnt policy will focus less on the appearance of the object and the scene while focus more on the geometric patterns. In addition, the randomness of the scene can lead to an improved robustness when dealing with an unfamiliar scene in real-world. This research randomises the texture of table, ground, and the appearance of task objects by applying texture images extracted from DTD texture database [73]. The eye-on-hand camera is in fixed position, but the experiment randomises the FoV of the camera by adding a small random number to the pre-set FoV value.

2.7 Hand-Eye Calibration

2.7.1 Hand-Eye Calibration Basis

In the realm of robotics, the precise coordination of vision systems and robotic manipulators is a critical component of numerous applications. Whether it's picking and placing objects with high accuracy, recognising and tracking objects in an environment, or even guiding a robotic surgical instrument, the synchronisation of cameras and robot arms is indispensable. This synchronisation is achieved through a process known as hand-eye calibration.

- **Forward Projection Process**

The transformation of 3D real-world points into a 2D image, a process commonly employed in image processing and computer vision applications, is known as forward projection. This section delves into the intricacies of this transformation, encompassing the crucial stages of converting from a world coordinate system to camera coordinates, subsequently to image coordinates, and finally, to pixel coordinates. The objective is to understand the fundamental concepts and mathematical foundations underpinning this process.

- **World Coordinates to Camera Coordinates**

The initial phase of forward projection involves the conversion of 3D real-world points, defined in the world coordinate system, to the camera coordinate system. This transformation is achieved through external camera parameters, comprising the rotation matrix (R) and the translation matrix (T). These parameters play a pivotal role in aligning multiple camera coordinate systems with a common world coordinate system. Detailed methodologies for achieving this transformation, including the specific procedures and calculations involved, will be elaborated upon in the subsequent chapter, featuring Algorithm 2 for the eye-in-hand configuration and Algorithm 1 for the eye-to-hand setup.

To enhance the clarity of the upcoming description, a distinct representation is employed compared to the previous section. In mathematical terms, the conversion from world coordinates (X_w, Y_w, Z_w) to camera coordinates (X_c, Y_c, Z_c) can be articulated as follows:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = [R|T] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{2.1}$$

, where $[R|T]$ represents the concatenation of the rotation matrix R and the translation matrix T. The outcome is a set of 3D points represented in the camera coordinate system.

- **Camera Coordinates to Image Coordinates**

The next step in the forward projection process involves the projection of 3D camera coordinates onto the 2D image plane. This crucial transformation is facilitated by the camera's intrinsic parameters, with a specific focus on the focal length denoted as (f). To perform this transformation, a perspective transformation is utilised, taking into account the 3D camera coordinates (X_c, Y_c, Z_c) and producing the resulting 2D image coordinates (u, v). The perspective transformation can be mathematically expressed as follows:

$$u = f \cdot \left(\frac{X_c}{Z_c} \right) \tag{2.2}$$

$$v = f \cdot \left(\frac{Y_c}{Z_c} \right) \tag{2.3}$$

Here, (u, v) represents the image coordinates, while (X_c, Y_c, Z_c) represents the camera coordinates. The focal length (f) serves as a scaling factor in this projection, defining the precise relationship between real-world units and image units.

- **Image Coordinates to Pixel Coordinates**

The final step in the forward projection process involves converting image coordinates (u, v) into pixel coordinates (x, y) while considering the camera's intrinsic parameters, which include the optical centre (O_x, O_y) and sampling rates (S_x, S_y). This conversion from image coordinates to pixel coordinates is mathematically expressed as:

$$x = S_x \cdot u + O_x \tag{2.4}$$

$$y = S_y \cdot v + O_y \tag{2.5}$$

Here, the terms (O_x, O_y) account for the optical centre's shift, and the sampling rates (Sx, Sy) are scaling factors applied during the imaging process. It's noteworthy that in many depth cameras, including Intel Realsense D435i used in the experiments, camera coordinates to image coordinates and image coordinates to pixel coordinates are seamlessly integrated into the camera's intrinsic matrix, streamlining the entire projection and affine transformation process.

2.7.2 Hand-Eye Calibration Setup

Hand-eye calibration is essential for a robot to understand the connection between its movements and the camera's observations. This process involves defining a mathematical model that links two key coordinate systems: the robot's end effector and the camera mounted on it. By establishing this relationship, the robot gains the ability to accurately interpret the spatial positioning of objects as seen through the camera's lens. Consequently, this alignment between the robot's perception and its actions facilitates effective interaction with the surrounding environment.

Two primary scenarios guide hand-eye calibration, each with distinct considerations:

- **Eye-to-Hand Calibration:** In this scenario, the camera is positioned externally to the robot's end effector. This configuration is common in scenarios where the camera is fixed in place, and the robot interacts with the environment. The aim is to determine the transformation between the camera's coordinate system and the robot's base coordinate system. This transformation is typically represented by the equation: $C_H = T * E_B$, where C_H represents the transformation between the Camera and the robot's end effector, T is the transformation matrix representing the end effector's pose relative to the base and E_B represents the transformation between the camera and the end effector. The detailed procedure for this calibration process is provided in Algorithm 1.

- **Eye-in-Hand Calibration:** In this scenario, the base and various joints of the robot, as well as the end effector, are already defined through a URDF. Here, the goal is to determine the transformation between the camera and the robot's end effector. This transformation is typically represented by the equation: $C_H = E_C \times C_E$, where E_C represents the transformation between the end effector and the camera andC_E represents the transformation between the camera and the robot's base. The detailed procedure for this calibration process is provided in Algorithm 2.

Algorithm 1 Eye-to-Hand Calibration Procedure

```
Require: Calibration object with known 3D features
Require: Images of the object from various poses
Require: Recorded robot end-effector poses
  Initialise empty lists: $\boldsymbol{T}_{\text{camera-to-object}}$, $\boldsymbol{T}_{\text{camera-to-robot}}$
  for each captured image do
    Estimate $\boldsymbol{T}_{\text{camera-to-object}}$ from image features
    Calculate $\boldsymbol{T}_{\text{camera-to-robot}}$ using robot pose data
    Extract rotation matrix $R$ and translation vector $t$ from $\boldsymbol{T}_{\text{camera-to-robot}}$
    Append $R$ and $t$ to $\boldsymbol{T}_{\text{camera-to-object}}$ and $\boldsymbol{T}_{\text{camera-to-robot}}$
  end for
  Aggregate data in $\boldsymbol{T}_{\text{camera-to-object}}$ and $\boldsymbol{T}_{\text{camera-to-robot}}$
  Calculate the final $\boldsymbol{T}_{\text{eye-to-hand}}$ using the aggregated data
  return $\boldsymbol{T}_{\text{eye-to-hand}}$
```

In both scenarios, the calibration process typically involves placing a distinctive marker on the robot's end effector and then using algorithms and specialised software to compute the transformation between the camera and the end effector. This transformation, crucial for achieving precise coordination, enhances the robot's capability to accurately interact with and understand its surroundings.

Algorithm 2 Eye-in-Hand Calibration Procedure

```
Require: Calibration object with known 3D features
Require: Images of the object from various poses
Require: Recorded robot end-effector poses
  Initialise empty lists: $\boldsymbol{T}_{\text{camera-to-object}}$, $\boldsymbol{T}_{\text{camera-to-robot}}$
  for each captured image do
    Estimate $\boldsymbol{T}_{\text{camera-to-object}}$ from image features
    Calculate $\boldsymbol{T}_{\text{camera-to-robot}}$ using robot pose data
    Extract rotation matrix $R$ and translation vector $t$ from $\boldsymbol{T}_{\text{camera-to-robot}}$
    Append $R$ and $t$ to $\boldsymbol{T}_{\text{camera-to-object}}$ and $\boldsymbol{T}_{\text{camera-to-robot}}$
  end for
  Aggregate data in $\boldsymbol{T}_{\text{camera-to-object}}$ and $\boldsymbol{T}_{\text{camera-to-robot}}$
  Calculate the final $\boldsymbol{T}_{\text{eye-in-hand}}$ using the aggregated data
  return $\boldsymbol{T}_{\text{eye-in-hand}}$
```

- **Aruco Markers**

Aruco markers, which abbreviate "Augmented Reality University of Cordoba markers," are binary square-based reference markers that find extensive use in camera pose estimation. These markers offer several distinct advantages, including simplicity, speed, and robust detection capabilities. An Aruco marker comprises a square shape distinguished by a wide black border and an inner binary matrix, which carries a unique identifier or ID. The prominent black border aids in swift marker detection in images, while the inner binary code serves identification, error detection, and correction purposes. The size of an Aruco marker determines the dimensions of its internal matrix; for instance, a 4x4 marker contains a 16-bit binary pattern.

In essence, Aruco markers operate as a form of encoding, much like the familiar QR codes encountered in the daily lives. However, differences in their encoding methods result in variations in information storage, capacity, and application. With four distinctive corner points and internal binary coding, a single Aruco marker supplies sufficient information for mapping from a two-dimensional realm to a three-dimensional space. This capability facilitates exploration of projection relationships between these two worlds, making it invaluable for applications such as pose estimation and camera calibration.

OpenCV's ArUco module comprehensively supports Aruco markers, encompassing their creation, detection, and use in tasks like pose estimation and camera calibration. Additionally, this module provides features such as marker boards. In this section, the primary focus centres on the creation and detection of Aruco markers.

To employ Aruco markers for spatial coordination estimation, the following steps are typically followed:

- **Camera Calibration:** Before utilising Aruco markers for spatial coordination estimation, precise camera calibration is indispensable. This calibration process ensures accurate knowledge of the camera's intrinsic parameters, including focal length, distortion coefficients, and principal point. OpenCV offers tools for camera calibration, which are crucial for precise spatial coordination estimation.

- **Marker Detection:** This process involves identifying and detecting Aruco markers within the camera's field of view. It entails locating the marker's corners and decoding its unique ID. The distinctive black border and inner binary matrix contribute to the robustness and reliability of this detection process. The image coordinates of the marker's corners form the basis for subsequent computations.

- **Pose Estimation:** After Aruco marker detection, pose estimation techniques come into play to determine the marker's position and orientation within the camera's coordinate system. This involves solving the Perspective-n-Point (PnP) problem, which calculates the transformation matrix between the marker and the camera. This matrix contains translation and rotation components, enabling the transformation of the marker's coordinates to the camera's coordinate system.

- **Spatial Coordination Calculation:** With the information obtained from pose estimation, it becomes possible to calculate the spatial coordinates of the Aruco marker. These coordinates are represented in a 3D Cartesian coordinate system. The translation component of the transformation matrix provides the marker's position, while the rotation component describes its orientation in 3D space. The detailed steps will be introduced in the next section.

The results, which employ OpenCV for the real-time detection and localisation of Aruco markers in each frame of the camera video stream, are presented in Fig. 2.10. The estimated poses of the markers are calculated, and their corresponding axes are drawn.

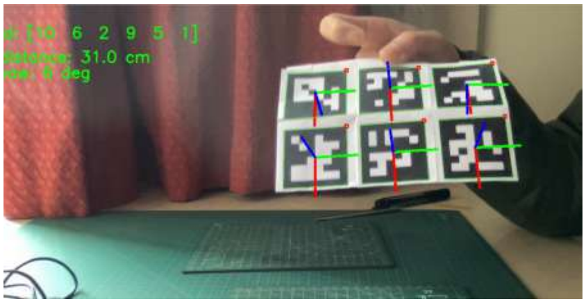


Figure 2.10. Visualisation of Aruco Markers Detection and Pose Estimation.

In summary, Aruco markers serve as fiducial markers, facilitating the estimation of spatial coordinates within a camera's field of view. This process involves camera calibration, marker detection, pose estimation, and spatial coordination calculation. The simplicity and robustness of Aruco markers, combined with the capabilities of computer vision libraries like OpenCV, render them indispensable for spatial coordination estimation for the tasks.

- **Eye-to-Hand Calibration**

This section outlines the practical process of performing eye-to-hand calibration of an UR5e robotic arm and a RealSense camera, which is run on Ubuntu 20.04 system.

The calibration setup involves the following key steps:

1. **Hardware Configuration:** Prepare the UR5e robotic arm within a controlled environment, ensuring it is correctly configured. Position the RealSense camera optimally to capture the workspace. Utilise ArUco markers as reference points.

2. **Software Preparation:** Configure the software environment, including the installation of ROS (Robot Operating System) on the Ubuntu 20.04 system. Install the RealSense SDK to interface with the camera. Ensure the UR5e robotic arm driver is installed and configured. Finally, install the 'easy_handeye' package to streamline the calibration process.

3. **Calibration Procedure:** This involves attaching ArUco markers to the robotic arm's end effector. Initialise the RealSense camera to capture RGB images. Move the robotic arm through different poses within its workspace, ensuring visibility of the markers, and simultaneously capture corresponding RGB images. UEmploy the 'easy_handeye' package to compute the transformation matrix, representing the spatial relationship between the camera's coordinate system and the robot's end effector (see Fig. 2.11).

4. **Verification and Refinement:** After obtaining calibration results, verify the accuracy of the transformation. Assess the robotic arm's ability to interact with the environment based on the calibrated data. If necessary, refine calibration parameters for optimal precision.

In contrast to the idealised pinhole camera model, which works well for small field-of-view scenarios, real-world cameras, especially those designed for capturing wide-angle or fish eye images, often employ convex lenses. While the pinhole camera model accurately represents imaging in situations with a small field of view, where lines in the 3D world map to lines in the image, it falls short in cases involving wide-angle or fish eye lenses, leading to distortion in the images. This distortion manifests as the

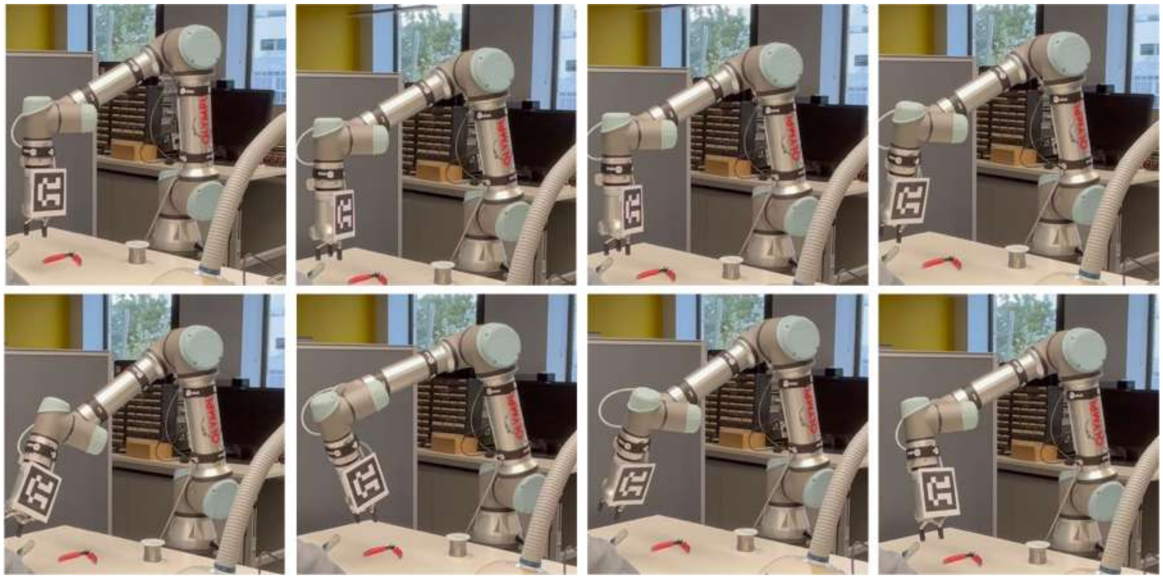


Figure 2.11. The calibration process unfolds from left to right and top to bottom. This sequence visually details the procedure, with AR marks attached to the robot hand.

curvature of straight lines from the 3D world when projected onto the image plane. Consequently, the assumptions made for coordinate transformations in the pinhole model no longer hold, necessitating the introduction of camera distortion parameters and the application of distortion correction techniques.

Camera lens distortion primarily falls into two categories: radial distortion and tangential distortion, with other forms of distortion having minimal impact and often being neglected.

- **Radial distortion:** Radial distortion arises due to the shape of the camera lens, and it becomes more pronounced as one moves towards the edges of the lens. This type of distortion results in straight lines in the real world appearing curved in the captured image.
- **Tangential Distortion:** Tangential distortion, on the other hand, is caused by misalignment between the lens and the image sensor (CMOS or CCD). With advancements in camera manufacturing processes, this type of distortion has become less common, and it is typically not a significant concern in modern camera systems.

When dealing with cameras that have a small FoV, such as smartphone cameras, the distortion can often be neglected due to the pinhole camera approximation, where straight lines in the 3D world remain straight in the image. However, in scenarios requiring wide-angle or fish eye cameras, distortion becomes a critical factor, necessitating distortion correction.

There are various methods for correcting camera distortion, with two common approaches being:

- **Lens Calibration:** This method involves capturing calibration images with known patterns (e.g., checkerboards) and analysing the distortions in these images to estimate the camera's distortion parameters. Once these parameters are known, they can be used to correct distortion in subsequent images taken with the same camera.
- **Software-Based Correction:** In situations where calibration is not feasible or practical, software-based distortion correction can be applied. This method involves the post-processing of captured images using distortion correction algorithms. These algorithms use mathematical models to remove distortion effects from the images.

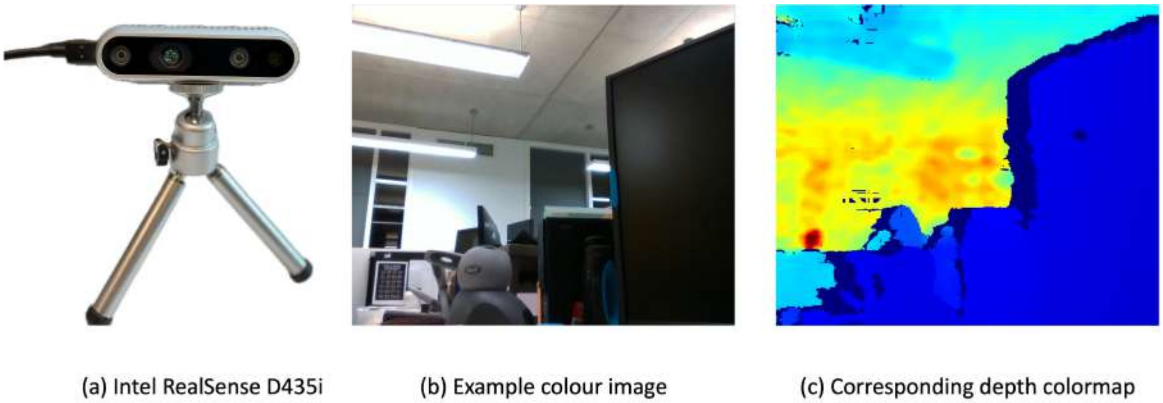
Distortion correction is crucial for applications such as computer vision, image stitching, and photogrammetry, where accurate geometric relationships between objects in the real world and their representation in images are essential. By accounting for and correcting lens distortions, the images captured by wide-angle or fisheye cameras can accurately represent the geometry of the scenes they capture.

Depth cameras, also known as depth-sensing cameras or 3D cameras, are a category of imaging devices designed to capture depth information in addition to traditional 2D images. These cameras have become increasingly popular in various fields, including computer vision, robotics, augmented reality, and more. Depth cameras offer valuable characteristics, including precise depth perception crucial for tasks like object recognition, 3D mapping, and obstacle avoidance, while also incorporating RGB imaging for comprehensive perception. They supply real-time data for swift decision-making and control, and their compact size facilitates seamless integration into diverse devices and systems. Depth cameras works on the basis of active depth sensing, where they emit a form of energy, typically infrared light, and subsequently measure the time it takes for this emitted energy to return after interacting with objects in the surrounding environment. Depth cameras can be categorised into two main types based on their principle of operation: the Time-of-Flight (ToF) approach and the Structured Light Projection approach.

- **Time-of-Flight:** Time-of-flight (ToF) depth cameras emit short bursts of infrared light and measure the time it takes for these light pulses to bounce back to the camera's sensor. This time measurement is directly proportional to the distance between the camera and the object. ToF cameras use specialised sensors, such as photodiodes, to capture the returning light and convert it into depth information.

- **Structured Light Projection:** Structured light projection involves projecting a known pattern, often a grid or series of stripes, onto the scene using an infrared projector. The camera then captures the deformed pattern as it interacts with objects in the environment. By analysing the deformation of the pattern, the camera can compute depth information.

In the experimental setup, depth cameras, specifically the Intel RealSense D435i, played a vital role in various tasks. These cameras provided depth information that allowed us to create detailed 3D reconstructions of the environment and keypoint position calculation. This was particularly valuable for robot navigation, object manipulation, and hand-eye calibration tasks. The image of the Intel RealSense D435i, employed in the experiment, is depicted in Fig. 2.12. The depth cameras' real-time capabilities ensured that the robotic system could react promptly to changes in the environment. Additionally, the cameras' integration with the ROS simplified their incorporation into the robotics framework, enabling seamless data access and processing.


(a) Intel RealSense D435i

(b) Example colour image

(c) Corresponding depth colormap

Figure 2.12. Image of the Intel RealSense D435i used in the experiment along with an illustrative example of a colour/depth image captured by the sensor.

2.8 Simulation for Teleoperation

2.8.1 Digital Twin Basis

In the domain of intelligent manufacturing, digital twin (DT) technology serves as a critical asset in streamlining and refining manufacturing processes. Essentially, digital twins are virtual replicas mirroring real-world systems or processes. These comprehensive models encapsulate the intricacies of the physical system, aiming to closely imitate its complexities. The core function of a DT is to offer an environment that replicates the real world, fostering a space for experimentation, testing, and analysis without the need to directly impact the physical system. Developing and validating robotic systems pose significant challenges. Real-world testing can be both hazardous and expensive, especially when working with intricate systems or novel algorithms. However, the creation of a virtual duplicate of the system in simulation allows for the safe and cost-effective testing and refinement of algorithms. This facet holds particular significance for complex systems demanding multiple trial runs, enabling rapid iterations and improvements in approach. Utilising digital twin technology significantly expedites the development process, ensuring the rapid and safe evolution of reliable and effective robotic systems.

Digital twin can be regarded as a tripartite framework, comprising the digital layer, the physical layer, and the cognitive layer. The digital layer serves as the foundation, encompassing crucial elements such as data ingestion, involving the collection, aggregation, and processing of information. This layer also embraces simulations through virtual models, analytical tools, and visualisation methods such as human interfaces. The physical layer incorporates tangible assets and robots from the real world, along with sensor data gathered through data collection and human input. The layer is further fortified by actuator control for managing control signals and algorithms. The Illustration of the Three Layers and Corresponding Components in a Digital Twin Framework is given in Fig. 2.13.

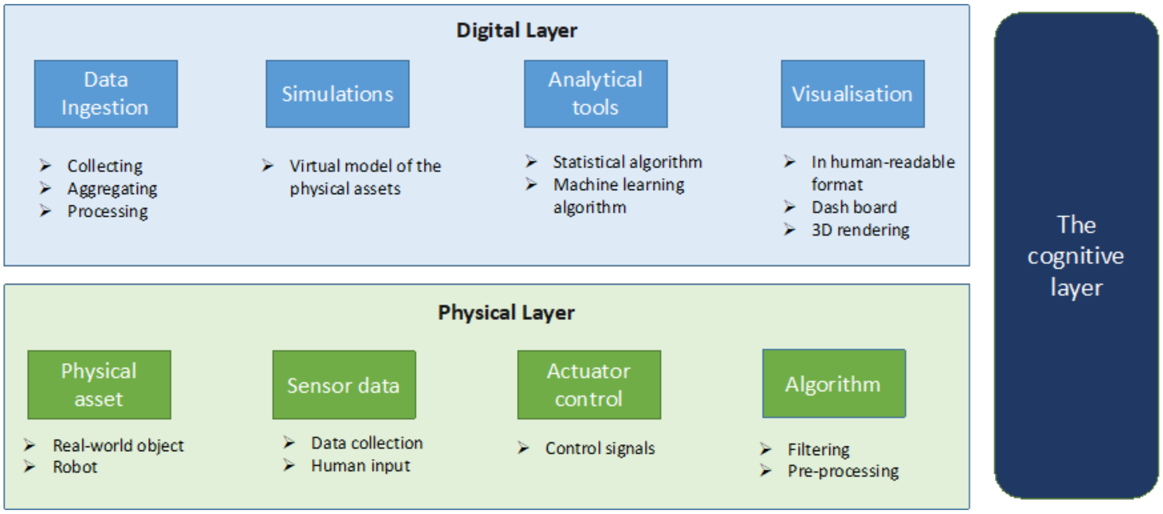


Figure 2.13. Digital Twin Framework Layers and Corresponding Components.

DT has various communication modes that can be chosen depend on the specific use case and requirements of the system. Different modes offer varying levels of real-time interaction, data flow, and control, making them suitable for a wide range of applications across industries like manufacturing, healthcare, and aerospace. Here are a few key communication modes that are relevant to DT systems in intelligent manufacturing settings:

- **Synchronous Communication:** In this mode, the digital twin operates in real-time coordination with its physical counterpart. It constantly receives and sends data, ensuring that both systems remain synchronised. This mode is crucial for applications where immediate feedback and response are necessary, such as industrial control systems.

- **Asynchronous Communication:** Asynchronous communication allows data to be exchanged between the DT and the physical system without strict real-time constraints. It's often used for scenarios where a slight delay is acceptable, such as remote monitoring and data logging.

- **Bi-Directional Communication:** This mode enables data exchange in both directions, allowing the DT to send data to the physical system and vice versa. This two-way communication is vital for applications that involve decision-making and action implementation in both domains, like collaborative robotics.

- **Unidirectional Communication:** In this mode, data flows in only one direction, typically from the physical system to the DT. It's often employed for data acquisition and analysis, where the DT continuously receives information from the real-world system for monitoring and modelling.

- **Sensor Data Streaming:** This communication mode involves real-time streaming of sensor data from the physical system to the DT. This data can include information from cameras, environmental sensors, or any other data source. It's crucial for applications that require the DT to have up-to-date information about its real-world counterpart.

- **Control and Command Communication:** This mode is employed when the DT is used to control or guide the actions of the physical system. It allows the DT to send commands and control signals to the physical system, facilitating actions like remote robotic control or automated process adjustments.

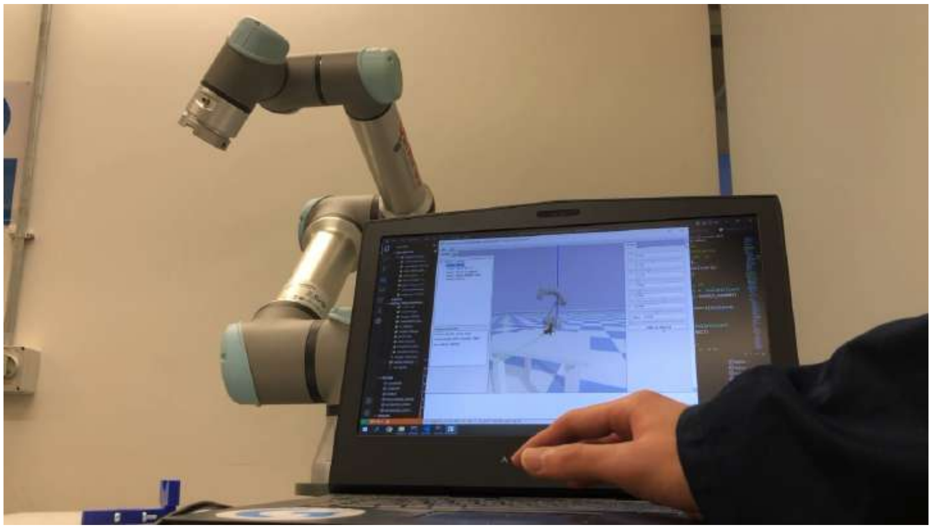


Figure 2.14. Digital twin deployment with PyBullet

2.8.2 Digital Twin Setup

For digital twin setup and demonstration, this research creates two experimental platforms each serves for different purpose and tasks requirement.

First of all, a PyBullet-based environment is created for deployed in a Virtual Reality (VR) environment with an HTC Vive or Oculus Rift VR device with the Valve Steam SDK installed. This DT implemented in PyBullet operates in unidirectional communication mode, facilitating data flow in two different directions. This DT environment works in two communication modes. The first mode involves data transmission from the physical to the digital realm, termed the ”monitoring mode”. In this mode, the information flows from the real-world system to the digital twin, allowing for continuous observation, analysis, and feedback generation within the virtual environment. This process is integral for real-time monitoring and evaluation. Conversely, the second mode, termed the ”following mode,” functions by relaying information from the digital twin back to the physical system. Here, the virtual environment initiates instructions and commands, guiding the actions or adjustments in the real-world system based on the analysis conducted within the digital twin. This mode enables the real-world system to adapt and react to the insights and decisions generated within the digital counterpart. The flowchart depicted in Fig. 2.15 illustrates the working principles of both modes: the monitoring mode (left) and the following mode (right).

In addition to the PyBullet, another environment utilising the Gazebo simulator with a configuration akin to the PyBullet simulator. This platform is established for safe algorithm validation and lays the groundwork for future explorations in virtual reality and teleoperation. In contrast to the PyBullet simulation setup outlined earlier, this platform prioritises the emulation of real-world scenarios over efficient learning algorithm training. Within this simulated environment, Gazebo functions as the digital twin, replicating the brazing process, as depicted in Fig. 2.6. It enhances the capability to conduct experiments and gather data under controlled, repeatable conditions, establishing a secure digital twin for previewing and predicting actions before real-world execution. Docker serves as the foundational technology, facilitating component encapsulation and consistent deployment across diverse environments.

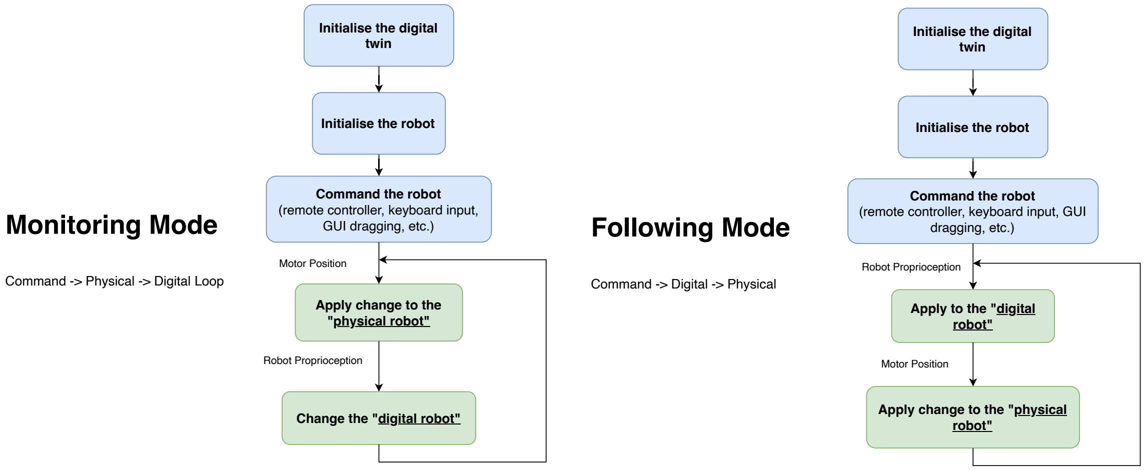

Monitoring Mode
Command -> Physical -> Digital Loop
Initialise the digital twin
Initialise the robot
Command the robot (remote controller, keyboard input, GUI dragging, etc.)
Motor Position
Apply change to the "physical robot"
Robot Proprioception
Change the "digital robot"
Following Mode
Command -> Digital -> Physical
Initialise the digital twin
Initialise the robot
Command the robot (remote controller, keyboard input, GUI dragging, etc.)
Robot Proprioception
Apply to the "digital robot"
Motor Position
Apply change to the "physical robot"

Figure 2.15. Flowchart of the monitoring mode (left) and the following mode (right).

This platform integrates human expertise with robotic capabilities for efficient and intuitive interactions and human demonstrated data collection for algorithm training. The connection between the DT and the tangible world is achieved with controllers, which can take various forms such as the Phantom Omni hand-controller, a PS2 controller, space mouse or even simple manipulation using a 2D mouse. These controllers transmit commands to the Gazebo environment, instigating actions in the simulation. This intricate interplay precisely mimics the behaviour of a real-world brazing system, providing insights for refining the intelligent manufacturing processes. The environment contains a UR5e robot equipped with an OnRobot 3-finger gripper. In addition, the extensibility of the environment accommodates the addition of multiple robotic arms, thereby facilitating multi-agent collaborative endeavours. Within this environment, the constituent elements encompass the robotic platform, diverse objects, and a meticulously replicated workspace mirroring the configuration of the real-world experiment, thereby ensuring faithful representation. The visual representation of this digital twin is aptly captured in Fig. 2.14, illustrating its dynamic operational framework.

Chapter 3

Literature Review

3.1 Visual Servoing Overview

3.1.1 Robot Vision

Robot perception refers to the process of extracting visual information from the environment to enable robots to interact with their surroundings effectively. Recent years have witnessed a notable increase in academic research pertaining to robot perception, which involves leveraging a diverse range of sensors to facilitate comprehending and interpreting the surrounding environment [8, 74, 75, 76]. With the advancements in computer vision and ML techniques, robots can now recognise, interpret, and make decisions based on the perception information gathered [77]. This has resulted in the widespread use of robot perception in various applications, including the navigation of mobile robots, providing context-awareness for service robots [78], robot arm manipulation [79], manufacturing [80, 81], mobile robots [82], transportation guidance for logistic [83], and many more.

3.1.2 Visual Servoing

The ability to distinguish between task objects and other objects and precisely reach them to perform tasks is a hallmark of human dexterity. While robots have made significant strides in recent years, replicating this level of performance remains a formidable challenge. Current robots often rely on accurate target positions and inverse kinematics techniques to determine the appropriate joint configuration for reaching objects [84]. Unfortunately, this approach typically requires prior knowledge of the desired target location, making the system vulnerable to failures caused by even minor variations. Therefore, it is crucial to develop a more robust and adaptive approach that can handle the complexity of real-world scenarios.

To overcome the constraints posed by pre-defined target positions, robots can utilise sensors to gather environmental perception data. This information enables robots to deduce target locations or estimate real-time errors, contributing to the development of a more resilient closed-loop system. Various feature detection algorithms have been proposed to accurately distinguish the task object from other distractions in the environment, including SIFT [85], SURF [86], FAST [87], ORB [88], and others. In recent years, convolutional neural networks (CNNs) have been widely applied for feature extraction in a variety of vision tasks, such as object detection [89, 90], recognition [91, 92], and tracking [93, 94]. CNNs are preferred for their robustness in dealing with occlusion, deformation, and other changes in object appearance. Additionally, recent studies have shown that enhancing the CNN's encoding of shape information can further improve the performance of template matching [95].

Nevertheless, the dependability of these systems is frequently impacted by environmental intricacies. Factors such as variations in ambient light or distractions within the robot's field of view can disrupt detection outcomes, posing challenges in designing perception algorithms that consistently operate in diverse environments. In the context of industry and early-stage research, a structured and controlled environment is often necessary to compensate for interference [96]. This may involve pre-defining the workspace, designing specific detection algorithms based on the 3D features of task objects or the environment, and setting up a simple background [97]. However, such an approach sacrifices flexibility and adaptability, and may not be suitable for daily tasks or small-batch production with frequent changeovers. Moreover, handcrafting a model for object recognition and localisation is time-consuming and impractical for every object in daily life. To achieve complex environment robot flexible manufacturing, robots can benefit from using vision-based control, also known as visual servoing, to increase the flexibility and adaptability of the system and enable the robot to perform a wide range of tasks, including those that were not explicitly programmed beforehand.

Visual servoing utilises visual feedback from a camera to control the motion and position of a robot arm, making it a versatile and adaptable approach for human-robot interaction [98, 99, 8, 100] and teleoperation [101, 102, 103, 104, 105]. While visual servoing is just one application of robot perception, it is a critical one for many robotic applications that require accurate and precise control of robot arm movement. One of the major advantages of visual servoing is its ability to work in unstructured environments, as it can adapt to changes in the environment and the objects being manipulated. Moreover, it can operate without requiring a precise or prior model of the robot's dynamics, making it more flexible and easier to implement in practice.

3.1.3 Category-Agnostic Localisation

As discussed in the last paragraph, human beings possess a remarkable innate ability to swiftly locate and identify objects within cluttered environments by presenting with a query image so that it does not need prior knowledge of the specific object. Transferring this capability to a robot arm, which can exploit object features in an image as cues, can be a significant advancement towards the finding and reaching of arbitrary unknown objects in unfamiliar settings, which thereby thrives versatile interaction types [106, 107] and the robot application in extreme environments [108, 109]. The introduction of query images provides an intuitive means of interaction with robots, allowing humans to communicate their intentions using the most instinctive form of information, namely visual information. In this context, human-robot interaction (HRI) plays a crucial role as a "third eye-hand agent", augmenting the robot's capabilities and assisting humans in accomplishing complex tasks [110]. The potential of such capabilities holds immense promise across a wide range of robot-assisted tasks, including table cleaning, factory sorting, and warehouse fetching. However, achieving this level of capability remains a highly challenging endeavour for robots.

Prior research advancements in the field of computer vision have made significant contributions toward enabling robots in unstructured environments. Numerous studies have explored the application of computer vision algorithms for object detection, classification, and segmentation [88, 111, 112, 113] in various scenarios, such as agriculture [114, 115], autonomous driving [116, 117], and indoor navigation [118, 119]. Semantic segmentation techniques, in particular, have been widely employed to accurately segment and classify objects [120, 118, 114, 121]. State-of-the-art models trained on large-scale datasets or fine-tuned with additional data have achieved impressive performance in object recognition tasks [122, 123]. However, these approaches are limited by the fixed number of detectors and struggle to generalise to scenarios involving objects with rich shapes, colours, and textures, such as daily life and manufacturing tasks. Also, these techniques often rely on specific models trained on extensive datasets or fine-tuned with additional data, encompassing diverse classes of objects. Unfortunately, these approaches face limitations due to their fixed number of classes, which hampers their direct application to brazing or other manufacturing tasks involving objects with arbitrary shapes. Follow by this, a research question comes out: to steer the robot with vision information, how can the robot know the target position without pre-defining a fix location?

To address this gap, research on category-agnostic methods has emerged to enhance robotics perception in more diverse environments. A milestone in category-agnostic segmentation is the work by [124], which can output category-agnostic masks and generalise to unseen categories. In the context of autonomous driving, [125] proposed a class-agnostic multi-object tracking method that handles rich and arbitrary object classes. [126] introduced an Mask-RCNN-based method for segmenting arbitrary objects using depth images. Another class-agnostic segmentation network based on dense feature comparison was presented in [127]. In comparison to the proposed method, which also involves a query image for comparison, the aforementioned methods require one annotated training image, while the method does not rely on manual annotations. Another line of research has focused on developing robotic grasping systems that exploit cues to achieve object manipulation and grasping. Approaches that utilise both query and image inputs to directly predict the desired robot actions have been proposed in [128, 44]. While these techniques show promise, they often rely on closed-loop recurrent control to compensate for estimation errors, which can be impractical in scenarios where trajectory planning is crucial and requires a separation of the estimation and control processes.

3.1.4 Category-Agnostic Servoing

Visual servoing can be categorised based on control methods into two branches: image-based visual servoing (IBVS) and position-based visual servoing (PBVS). In PBVS, vision data is used to reconstruct the 3D pose of the robot and generate a kinematic error in Cartesian space [129]. On the other hand, IBVS generates an error directly from image plane features. While PBVS often requires three-dimensional reconstruction [130], which is highly sensitive to camera calibration parameters, IBVS is known for its robustness with respect to camera calibration accuracy and stability under noisy conditions, which makes it well-suited for operation in unstructured environments where it can adapt to changes and operate without requiring a precise model of robot dynamics [131].

However, controlling the robot with IBVS can be challenging since the action is applied directly to the image plane. To address this issue, data-driven methods have gained increasing attention in various fields, including manufacturing [132], robotics [133, 134, 106], and optics [135]. Among these methods, RL has shown great potential in generating control policies without requiring a model of the system as discussed in the last section. By learning implicitly from a pre-specified reward function and optimising the reward through interactions with the environment, RL can converge to a learned policy that maximises the reward [136].

IBVS tasks can be addressed using various image processing and computer vision techniques. These tasks can be divided by whether the image has depth information. Depth information provides a detailed representation of the scene's three-dimensional geometry, allowing for a more accurate understanding of the scene's structure and spatial relationships between objects [137, 138]. In comparison, RGB images have become increasingly popular for robot applications due to their widespread availability, high-resolution imaging capability, and ability to provide colour information for object detection [139], recognition [140] and classification. However, relying solely on RGB information can pose challenges and limitations, such as occlusion, changes in illumination, and noise, which can adversely affect the accuracy and reliability of the control system [141]. Despite these challenges, researchers have continued to explore ways to improve the performance and robustness of visual servoing systems that use only RGB information. For example, [142] proposes an attention-based network that is still able to estimate depth information even when an object is partially obstructed from view, which can lead to incomplete RGB information. Direct visual odometry methods have also been used to compensate for illumination changes, which can cause significant variations in RGB information, leading to misinterpretation or incorrect interpretation of visual data [143]. Additionally, semantic segmentation is utilised in the RGB-based approach to deal with noisy and unstructured environments [144].

IBVS techniques can be classified according to the type of sensor used, with RGB cameras being a popular and cost-effective option for low-cost robot systems [8]. However, using RGB as the sole perception source for a robot system poses several challenges. One significant challenge is noise. Optical sensor noise, which can be caused by various factors such as electronic noise, environmental noise, or motion blur, can introduce significant errors into the control system, leading to reduced accuracy and decreased performance. Other challenges and limitations of using only RGB information for visual servoing include the lack of depth information, limited field of view, and sensitivity to variations in camera pose and calibration. To overcome these challenges and limitations, researchers have developed various techniques that use additional sources of visual data or modalities, such as depth or infrared information, to enhance the performance and robustness of visual servoing systems.

Nevertheless, utilising RGB information exclusively for visual servoing presents several distinct advantages, rendering it a competitive choice for numerous robot systems. One significant advantage is that RGB cameras are widely available, relatively inexpensive, and can provide high-resolution images, making them a popular choice for many robotic applications. RGB information can also provide valuable colour information, which can be useful for object recognition and tracking. Moreover, using RGB information for visual servoing can be computationally efficient, as it requires minimal preprocessing of the visual data. This can be especially important for real-time applications where rapid response times are critical. Despite these advantages, it is important to acknowledge the challenges and limitations of relying solely on RGB information for visual servoing and explore ways to overcome them to improve the accuracy and reliability of robotic systems.

RGB camera based IBVS techniques can be broadly categorised into three main branches: ML based approaches, feature-based approaches, and hybrid approaches. ML based approaches, such as YOLO [111], Faster R-CNN [145], Mask R-CNN [146], RetinaNet [147], and SSD-6D [148], take an image as input and output the object location in either the image domain or the real-world coordinate system. These algorithms are trained to recognise different types of objects and can adapt to variations in object appearance and lighting. Transfer learning techniques can also be employed to fine-tune these methods for specific tasks. On the other hand, feature-based approaches use extracted features such as edges, corners, and SIFT features [85] to estimate the object pose and location. In contrast to semantic segmentation, which classifies every pixel in the image, template matching uses predefined templates of objects to find their matches and predict the object bounding box. Each template is created by extracting specific features from a set of images containing the target object. Hybrid approaches combine the strengths of both feature-based and ML based approaches by leveraging the robustness of feature-based approaches to handle occlusions and changes in lighting while utilising the adaptability of ML-based approaches to handle variations in object appearance.

One of the challenges in visual servoing is the design of a robust controller that can handle estimation errors caused by the feature extractor and control the robot arm in the image plane. End-to-end methods have been explored to estimate the pose for visual servoing in previous works [34, 149, 33, 32, 30]. However, directly deriving the control policy from visual information can lead to a tremendous state or action space, which is often challenging for exploration, especially when dealing with sparse rewards, and the algorithm may fail to converge. Many end-to-end methods train the perception and control systems jointly [150, 31, 151, 152]. The method processes the raw visual information and controls the robot in two separate modules without any prior knowledge of the task object and the environment, rather than using observed images as input to infer the control command [31] or utilising the adapted perception information to drive the robot with traditional control law [34].

Additionally, the distinct dynamics of the real world and simulation world pose challenges when transferring models trained in simulation to the real world. This has led to the emergence of the ”sim-to-real” research field. In a study by Pinto et al. [153], an actor-critic training algorithm was proposed to address this challenge. The critic network was trained on full states, while the actor network used rendered images. This method reduces the need for expensive and potentially dangerous real-world training processes. Compared to traditional simulator-based reinforcement learning policy training, this approach minimises performance degradation when transitioning from simulated observations to real-world observations.

To derive a mapping from the raw perception information to robot control strategies, RL has attracted growing interest. RL has shown great potential in decision-making [154], control [102, 155], and achieving superhuman performance in various games [48, 156]. However, applying RL to robot manipulation is not straightforward due to several challenges. Firstly, the real-world robot environment has limited access to the states of objects within it, such as their position, speed, and acceleration, leading to partial observation. Secondly, the spatial complexity of the real world is high, while RL sample efficiency is low, resulting in high training costs. Thirdly, safety must be considered when training the robot, and direct training on a real-world robot can be energy-inefficient and potentially hazardous to its surroundings during exploration. In recent years, training in a simulation environment has become a more conventional approach due to its efficiency and safety in data collection and algorithm training. However, the reality gap between the simulated environment and its physical counterpart poses a new challenge when transferring the trained policy to the physical robot. Thus, simulation-to-real-world (Sim2Real) problems remain a trending research topic in this area, aiming to narrow the performance gap between simulation and the real-world [157, 158, 159].

3.2 Human-Robot Collaboration Overview

Human-Robot Collaboration (HRC) is a research area within the field of human-robot interaction (HRI) that focuses on designing, implementing, and evaluating robotic systems involving direct interaction between humans and robots [160]. HRC has offered a compelling synergy between humans and machines in various domains [161], where the need for efficient and seamless interactions between humans and robots becomes increasingly imperative [162]. Understanding the dynamics of HRC requires examining the various types of relationships that can exist between humans and Cobots. [163] classified these relationships into four types, as depicted in Fig. 3.1. These include the ”independent” relationship, where the operator and Cobot share the same space but perform different tasks; the ”simultaneous” relationship, where they work on the same or different workpieces together; the ”supportive” relationship, where they collaborate towards a common goal; and the ”sequential” relationship, where they sequentially work on the same piece.

In the specific context of this research project, the objective is to design a robotic system to optimise the brazing process, specifically the brazing pasting process. While automated machines can improve efficiency and repeatability, they often face challenges in achieving universal integration within complex manufacturing scenarios. When human operators are involved alongside automated machines, achieving full automation becomes impractical. For example, in the vacuum furnace brazing process, although the brazing itself can be automated, pre- and post-brazing processes still require manual intervention. Additionally, the hand-off time between human operators and automated machines, as well as speed matching issues, introduce complexity to the system. In such cases, a more flexible manufacturing model is needed, and HRC offers a potential solution. By incorporating Cobots into the production line, it becomes possible to leverage their collaborative abilities to assist human workers. This approach enables the seamless integration of human skills and

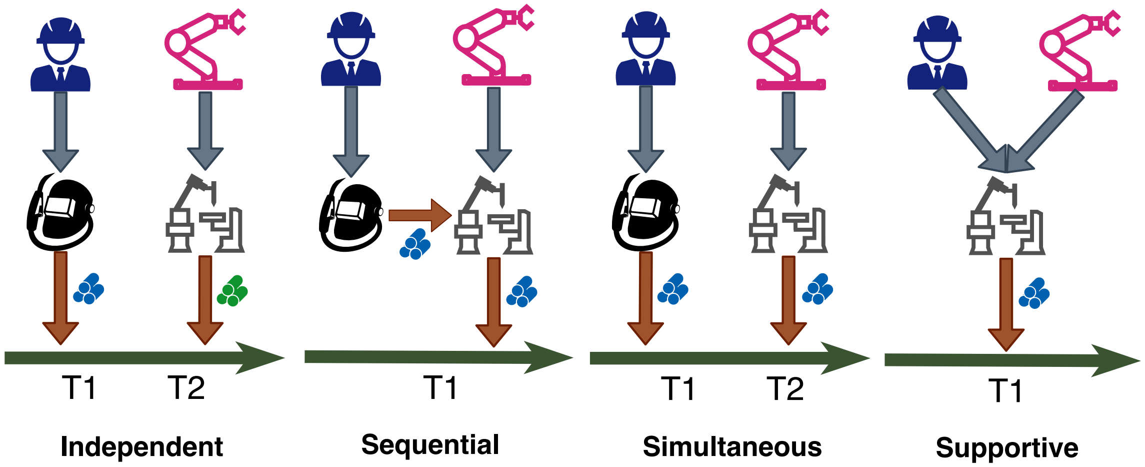


Figure 3.1. Classification of Cobot-human relationships [163].

robot precision, addressing the limitations of fully automated production lines and offering flexibility in scenarios where complete automation is challenging or unrealistic. This section provides an overview of the human-robot interfaces, HRC control, and metrics for evaluating HRC in the context of the robot pasting tasks and propose a framework that allows human to collaborate with robot intuitively.

3.2.1 Human-Robot Interfaces

In recent years, there has been a surge of interest in the development of HRC devices and methods. One of the challenges faced by HRC is designing robots that can collaborate with humans in a natural and intuitive way [107]. Humans have evolved to interact with other humans, and designing robots that can interact with humans in a way that feels natural and intuitive is a difficult task. The widespread adoption of HRC technology faces a critical bottleneck: the reliance on dedicated, often expensive, external controllers. The high cost of such controllers limits the widespread adoption of robot teleoperation to some extent.

Existing HRI predominantly rely on joystick or button-based interfaces [164, 165]. These interfaces, while effective, pose cognitive demands and lack the innate intuitiveness characteristic of natural human interactions. Conversely, the pursuit of enhanced precision and sophisticated hand controllers complicates the learning curve for new users, rendering the technology more economically burdensome. For example, various studies have focused on evaluating or devising devices that utilise hand controllers to provide physical interaction for robot control [166, 167]. While these controllers offer accuracy, they tend to be costly and lack intuitiveness, as they do not directly map human intentions to robot control. In scenarios where robotic movements adhere to relatively straightforward and non-complex trajectories, the intricate controls inherent in conventional hand controllers may prove excessive. This conundrum hinders broader adoption, particularly in applications where streamlined and easily accessible control interfaces could significantly enhance task performance [168].

Vision-based methods offer a relatively affordable and straightforward deployment option compared to other approaches. Among these methods, hand gesture-based teleoperation stands out as an intuitive means of controlling robots. Advancements in artificial intelligence and computer vision have led to improvements in hand detection, hand gesture recognition [169, 170, 171], and hand pose estimation [172, 173, 174] algorithms, making them increasingly accurate, faster, and more robust, which makes them a more common choice in robot teleoperation. For example, [164] proposed a dual camera based hand gesture recognition system for surgical robot teleoperation. [175, 175] program the robot through static hand gestures. However, these methods typically utilise hand gestures solely for commanding or for a specific purpose, which still does not fully convey human intent, such as omitting control over the robot's velocity or the direct control of the robot. Hand gesture-based approaches proposed in [176] and [177] directly map hand gestures to the robot's dexterous hand movements, offering intuitive and flexible control. However, these methods require users to continuously demonstrate the desired movements, which can be inaccurate and fatiguing. Additionally, a subset of these devices lack force feedback, hindering the operator's ability to perceive the robot's interaction with the environment; while those that do possess force feedback are vulnerable to hacking or tampering, introducing potential safety hazards to both the robot and its operator , especially in situations involving critical aspects of human life, such as remote surgery . Their relatively intricate nature also impedes their timely deployment in scenarios involving contingencies or demanding high temporal responsiveness.

Virtual-Reality (VR) devices can provide an immersive HRI environment [178, 179]. However, traditional VR devices are not always suitable for human operators to perform tasks in all application scenarios owing to factors such as cybersickness. Furthermore, VR devices are often costly, which also restricts the adoption of robot teleoperation. Additionally, existing VR devices primarily emphasise offering an immersive experience to the operator, while VR systems generally introduce latency, which may impede the operator's control of the robot and fail to achieve satisfactory performance in terms of operator interaction. Compared to VR, Augmented Reality (AR) typically has lower device requirements. AR experiences can be achieved by using smartphones or tablets with AR applications, which makes AR more accessible and user-friendly [180, 181]. Additionally, this approach reduces the likelihood of motion sickness that can occur in VR experiences. AR technology requires accurate perception and tracking of the real-world environment, as well as the integration of virtual content with it. This presents technical challenges for both hardware and software development, including the need for high-precision sensors and algorithms. Therefore, despite the intuitiveness and naturalness of VR and AR, it still cannot avoid the reliance on large datasets and high-precision sensors.

In contrast, devices that rely on human signals, such as gloves [182], inertia sensors [183, 184, 185, 186], motion capture systems, brain-computer interfaces (BCI) [187, 188], voice recognition [189], hand gesture recognition [175, 190], and electromyographic (EMG) signal devices, facilitate the mapping of human intention to robot control, thereby simplifying the transfer of intention from human to robot. Although these approaches offer intuitiveness and direct mapping of human intention to robot control, they often require complex and customised devices, resulting in high costs that hinder widespread adoption. Additionally, despite their intuitive nature, these devices still necessitate training time for operators to familiarise themselves with the environment.

In conclusion, there is a pressing need for a cost-effective, intuitive HRC interface that bridges the gap between human intent and robot action. The most ideal natural interaction state between humans and robots for brazing or other similar scenarios is a seamless, fluent, and intuitive interactive experience that resembles natural communication with another human being. Intuitive means requiring no complex instructions or learning curves, streamlining the training process for users and facilitating a seamless and natural interaction between humans and robots. Cost-effective means liberating users from the reliance on expensive, specialised controllers through the utilisation of readily available technology.

3.2.2 Human-Robot Collaboration Control

HRC control strategy can be categorised into two primary approaches: predictive human motion modelling [191, 192, 193] and reactive strategies.

Reactive strategies prioritise real-time adaptation to human input without explicit modelling [194]. Conversely, predictive human motion modelling techniques utilise ML and predictive modelling to enable machines to proactively respond to human movements [107, 195, 105, 196, 197, 198]. The aim is to create more anticipatory and responsive interactions, enhancing the intuitiveness of HRC, which allows robots to learn directly from task data to achieve optimal parameters for various applications [199]. Through the training of ML algorithms, the model can learn and capture the patterns and regularities of human motion from a large amount of data. This enables the model to cope with different people's motion habits and styles, and has a certain generalisation capability. By leveraging predictive human motion modelling techniques, robots can enhance their collaboration with humans. They can anticipate human's next actions, thereby facilitating improved coordination and alignment of their own behaviours. This capacity for HRC fosters increased work efficiency and safety, while also fostering cooperation and mutual trust between humans and robots. As a result, it significantly contributes to the advancement and wider adoption of human-robot collaboration.

However, these conventional methods come with their own set of limitations. One notable issue is the reliance on additional devices and signal processing, including technologies such as EMG [200, 201, 202, 203], electroencephalography (EEG) [204], or physiological signal-based approaches [205, 206, 207]. While these techniques have shown promise in certain contexts, they introduce elevated costs and increased complexity into the interaction setup. Users are often required to wear or employ these devices, which can be cumbersome and detract from the natural flow of interaction. Furthermore, large datasets are often required for training in HRC [208], which can be time-consuming and resource-intensive. This process, although crucial for the machine to understand human intentions, may not capture the full range of behaviours. Consequently, these systems may struggle to adapt beyond their training data, limiting their versatility in real-world settings. Thus, there's a need for more intuitive, adaptable, and less cumbersome approaches to HRC that align better with natural human communication and require minimal additional equipment or extensive data collection.

3.2.3 Human-Robot Collaboration Metrics

The current landscape of research in HRC can be distilled into two primary threads. The first centres on simplifying the programming and instructions for Cobots, while the second thread explores more advanced forms of interaction by equipping Cobots with semantic understanding capabilities or AI-driven anticipation skills [209]. Both of these threads represent the cutting edge of academic research in HRC. At the heart of these two threads lies a fundamental shift—from humans adapting to machines to machines adapting to humans [210]. This transformation underscores the overarching goal of HRC research: to enhance intelligence and digitisation in human-robot collaboration.

An examination of recent research trends in Cobot-related concepts [23] reveals a predominant focus on developing specific aspects of Cobot capabilities. These endeavours highlight five distinctive dimensions of Cobots:

- **Mobility:** The ability to move the robot from one location to another.
- **Intelligence:** Cobots should possess awareness of their surroundings and generate feedback.
- **Connectivity:** This pertains to both human-Cobot communication and Cobot system communication.

- **Actuation:** The capability to execute safe and dynamic trajectories.
- **Human-Centredness:** Cobots should provide support to human operators from both physical and mental perspectives.

These five dimensions serve as crucial metrics for assessing and evaluating HRC systems. The most extensively researched areas in HRC encompass proposing frameworks for Cobot deployment, mathematical modelling of the Cobot environment, and comprehensive surveys. Among the diverse applications of Cobots, assembly processes have garnered substantial attention, with particular emphasis on enhancing human-robot communication. This examination underscores that HRC research is primarily driven by industry needs, highlighting the importance of aligning research with practical industrial requirements and technological innovation. Given the broad applicability of HRC across various industries, the establishment of standardised or widely accepted design principles or frameworks can significantly aid in guiding the design, development, and integration of Cobots. In recent years, such frameworks have gained attention, with one prominent example outlined in [22]. This framework comprises three layers, each featuring two complementary viewpoints:

- **System Viewpoint:** Comprising contextual and conceptual aspects.
- **Embodiment Viewpoint:** Encompassing logical and physical facets.
- **Detail Viewpoint:** Focusing on risk and safety requirements.

Within each layer, four distinct starting points for consideration exist: data, function, interconnection, and motivation. This matrix-style framework offers a comprehensive evaluation of Cobot design and implementation, facilitating assessments of the comprehensiveness of system design concepts.

From an embodiment design perspective, the architecture of Cobots can be categorised into various modules. [211] introduced a three-phase cyclical framework representing three core, interconnected modules essential for Cobots as illustrated in Fig. 3.2. In this taxonomy, every phase of the circle corresponds to perception, skills, and behaviours. These three modules constitute the foundational components of a Cobot's operation. The perception phase focuses on gathering essential information about the surrounding environment and determining target points. Once the working space and tasks are perceived and defined, Cobots proceed to compose the individual skills module, forming complex motion plans. These motions are then executed by the behaviour module, which includes actuators, leading to changes in the state of the system. This cyclical process offers a coherent paradigm for both the logical and physical design of Cobots.

In alignment with the aforementioned taxonomy [211], the literature on HRC can be classified into three tracks:

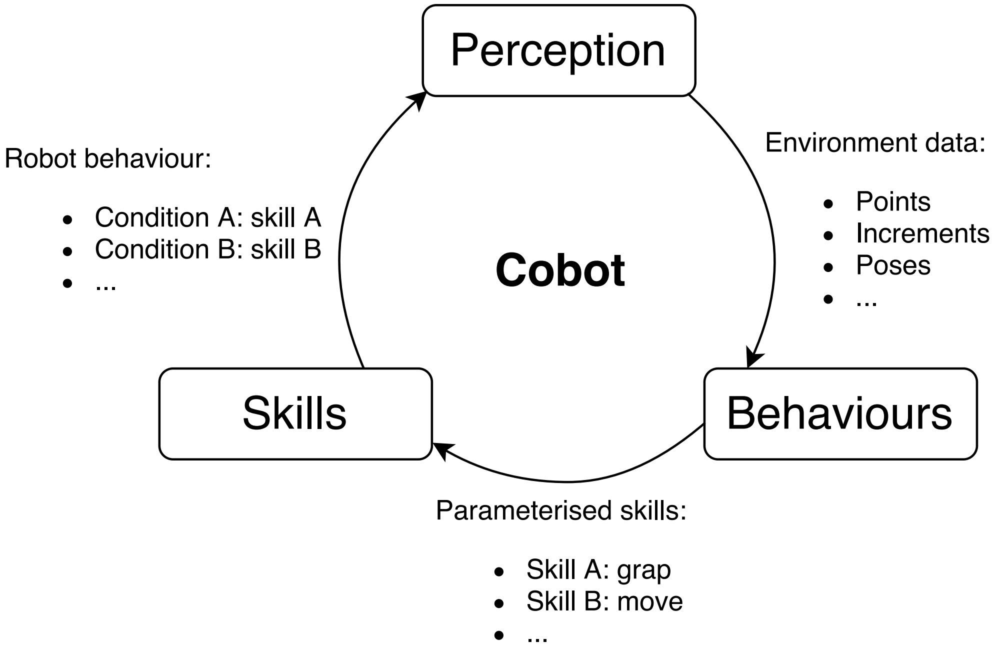


Figure 3.2. Schematic Representation of Intelligent Cobot Architecture [211]

- **Perception Design Track:** The perception module of a cobot plays a pivotal role in acquiring essential environmental information. Enhanced perception leads to heightened contextual awareness, thus bolstering the cobot's intelligence. Furthermore, ensuring the safety of human operators hinges on the effectiveness of cobot perception.

  Sensors serve as the primary means of environmental perception and can significantly impact HRC system performance. Integrating data from diverse sensors poses a challenge, which can be mitigated through the use of an Internet of Things (IoT) Multi-sensor Data Fusion (MDF) platform. This platform, utilising data from various sensors like infrared, WiFi, and sub-THz cameras, offers real-time cobot environmental perception [22].

  Nevertheless, sensor limitations introduce their own set of challenges. Research by [212] highlights significant latency in vision-based sensor systems, particularly concerning human movement and position detection, compared to control and actuation systems. Addressing sensor latency is critical in ensuring the safety of collaborative human-robot environments, particularly in vision-based HRC systems sensitive to task completion times.

- **Skills Design Track:** Skills encompass a set of actions performed by the robot to execute specific aspects of a task. For example, driving a nail into a hole involves several sub-actions like recognising a hammer, picking it up, identifying the target nail, picking up the nail, positioning it correctly, and finally, driving it into the hole. Skills can be considered as the building blocks that cobots use to execute tasks.

Robots can acquire these skills through various methods. Two predominant methods in industrial robotics are code-based programming and human demonstration. In [213], a framework for intricate sanding of complex surfaces was proposed, combining standard stiff position-controlled industrial manipulators with trajectory generation techniques derived from computer-aided design (CAD) and Programming by Demonstration.

ML techniques have gained traction in skill acquisition due to their ability to handle complex tasks. [214] introduced a hybrid approach that combined imitation learning with Q-learning-based reinforcement learning to enable a humanoid robot to perform collaborative tasks with humans. Other methods involve human demonstration and kinesthetic teaching, as seen in [215], and monocular camera-based learning, as demonstrated in [216]. However, data requirements and computing limitations can affect the widespread adoption of these methods in manufacturing.

- **Behaviours Design Track:** Behaviours in cobots are composed of skills. To orchestrate a set of skills harmoniously, [217] proposed an object-centric framework for learning and sequencing robot manipulation skills based on demonstrations. This approach minimises manual modelling efforts and enhances the flexibility and reusability of learned skills.

  Safety is a paramount concern when designing cobot behaviours, encompassing aspects such as predictability, speed/torque limits, path planning, and obstacle avoidance. In [218], an adaptive motion planning system was introduced to enhance worker safety while increasing operational efficiency. This system incorporates a worker motion predictor and an online trajectory generator to minimise waiting times and avoid contact during irregular worker movements.

  Efficiency is another key consideration in behaviour design. Efficiency metrics encompass time, cost, and human energy. For assembly line operations, efficiency can be affected by task sequencing and tool switching frequency between humans and robots. [219] proposed an adaptive algorithm to reduce human-robot and robot-robot tool switches while ensuring operator safety through a transparent and comprehensible workflow. Ergonomics is also considered by dynamically adjusting robot behaviour according to the operator's position and preferences, as demonstrated in [220] using a two-layered genetic algorithm to optimise task distribution and sequencing in a shared workspace.

Chapter 4

Category-Agnostic Visual Servoing

4.1 Introduction

In the context of brazing processes, the objects that require localisation present a diverse and challenging landscape for robotic systems. These objects primarily include brazing filler metals (BFMs), flux materials, and the workpieces themselves. BFMs come in various forms such as wire, paste, powder, or preformed shapes, each with unique visual and physical characteristics. For instance, brazing wires can be thin and reflective, making them difficult to detect against varying backgrounds. Paste and powder forms of BFM often lack distinct features and can blend with surrounding surfaces. Workpieces in brazing applications range from simple geometries to complex, irregular shapes, depending on the industry and specific application. These may include heat exchanger components, automotive parts, aerospace structures, or intricate electronic assemblies, each presenting its own set of challenges for localisation.

The pursuit of flexible manufacturing systems for brazing processes is significantly hampered by the limitations of traditional robotic vision and control systems when faced with this object diversity. Conventional object detection and localisation approaches often rely on predefined models or extensive training data for specific object categories. However, the wide variety of BFMs, fluxes, and workpieces encountered in brazing operations renders such category-specific methods inadequate. Furthermore, the dynamic nature of brazing environments, where object appearances can change due to heat application or flux coating, adds another layer of complexity. This chapter addresses these challenges by exploring category-agnostic object detection, localisation, and visual servoing techniques. These approaches are crucial for enabling robots to handle the diverse, often untrained, and irregularly shaped components common in flexible brazing manufacturing environments. The investigation is structured into three main parts:

1) Category-Agnostic Vision Algorithms Comparison: this part begins by comparing various visual servoing algorithms specifically designed for scenarios involving irregularly shaped objects. This comparative analysis provides a foundation for understanding the strengths and limitations of existing approaches in flexible manufacturing contexts.

2) Category-Agnostic Localisation: Building on this foundation, this part introduces a novel hybrid approach that integrates deep learning with feature-based methods. The proposed method aims to achieve robust category-agnostic object detection and localisation, enabling robots to handle a wide range of workpieces without prior training on specific shapes.

3) Category-Agnostic Visual Servoing: The final part of this chapter explores the development of visual servoing frameworks that leverage category-agnostic localisation capabilities. Two distinct approaches are investigated: image-based visual servoing (IBVS) and position-based visual servoing (PBVS). The IBVS framework utilises a feature extractor to embed image information, focusing on algorithmic robustness. In contrast, the PBVS framework estimates position difference and servo based on the estimated position difference. Both methods aim to enable robots to reach previously unseen objects without requiring calibration, addressing the challenge of operating in complex, less-informed environments. These frameworks are designed to allow robots to autonomously navigate and interact with diverse objects in unstructured environments, a key requirement for flexible manufacturing systems. The development of these visual servoing techniques represents a significant step towards enhancing the adaptability and efficiency of robotic systems in flexible manufacturing scenarios.

Throughout these investigations, this chapter focus on two critical aspects of flexible manufacturing: the visual perception of diverse objects and the servoing control to interact with these objects. The approach emphasises the development of systems that can adapt to varying workpieces and conditions with minimal reprogramming, directly addressing the challenge of Limited Autonomous Capabilities identified in the research aims. By developing these category-agnostic techniques, the chapter aim to enhance the flexibility and adaptability of robotic systems in manufacturing processes. The proposed methods not only improve the robots' ability to handle diverse workpieces. The results of the investigations demonstrate promising advancements in precision and accuracy, outperforming conventional methods and widely-used models. These improvements position the approach as an effective tool for enhancing the capabilities of flexible manufacturing systems, particularly in scenarios requiring adaptable and intelligent robotic vision and control.

The following sections will delve into the methodologies, experiments, and results of the investigations, providing a comprehensive analysis of the contributions to the field of flexible manufacturing systems through advanced visual servoing techniques.

4.2 Methods

4.2.1 Category-Agnostic Vision Algorithms Comparison

This section first introduce and compare the performance of different CV algorithms in the context of category-agnostics. This section compares three different approaches: a feature-based approach, a hybrid approach, and a machine-learning-based approach. To evaluate the performance of the approaches, experiments in a simulated environment using the PyBullet physics simulator (modified version of the basis environment) is conducted and introduced in the evaluation section. The experiments included different levels of complexity, including different numbers of distractors, varying lighting conditions, and highly-varied object geometry.

Some of the most representative algorithms for each category have been selected and tested respectively. For machine learning-based approaches, this research has tested a state-of-the-art semantic segmentation model, namely DeepLabv3+ [221]. For feature-based approaches, this research has tested SIFT [85] and ORB [88]. For the hybrid approaches, the method proposed in [222] has been tested. This algorithm comprises a convolutional neural network (CNN) extractor to extract features in both the template and captured image and compare their similarities using Normalised Cross-Correlation (NCC) [223]. The hybrid approach combines the strengths of both feature-based and machine learning-based approaches to achieve flexible object recognition and localisation.

The machine learning approach trains a deep learning model for semantic segmentation and uses connected component labelling to locate the object in the image. In contrast, the feature-based approach extracts distinctive features from the object and scene using Scale-Invariant Feature Transform (SIFT) and Oriented FAST and Rotated BRIEF (ORB) feature detectors and matches them to obtain correspondences between the two. The hybrid approach combines the strengths of both approaches by using the output of the semantic segmentation as a mask to limit the search for matching features, thereby improving efficiency and accuracy.

- **Machine Learning-Based Approach**

The machine learning-based approach uses a deep neural network to perform semantic segmentation of the image, followed by connected component labelling to locate the object. Specifically, this research employs a fine-tuned DeepLabv3+ [221] with a ResNet-50 [224] backbone for semantic segmentation. DeepLabv3+ is a state-of-the-art model that uses atrous convolution and a decoder module to refine the segmentation output. ResNet-50 is a widely used backbone architecture that has shown excellent performance in various computer vision tasks. The model on the collected dataset has been fine-tuned, which was generated in a simulator and included corresponding ground truth labels, using the cross-entropy loss function.

To accommodate the irregular shape and varying colour of the objects in the settings, this work trained the machine learning model to recognise background instead of object recognition. This approach allows the trained algorithm to detect any non-background objects within the field of view. During the inference stage, the trained model is applied to the input image I_m to generate a pixel-wise semantic segmentation map M, with each pixel classified into one of the pre-defined categories. The binary mask of the object is then obtained using a connected component labelling function, which represents the object's region in the image. Finally, a bounding box of the object is computed based on the binary mask, which is defined as the minimum rectangle that encloses the binary mask. This approach accurately localises the object in complex and cluttered scenes, and the use of semantic segmentation and connected component labelling makes the approach robust to variations in lighting, viewpoint, and object appearance, which makes it suitable for a wide range of applications.

- **Feature-Based Approach**

In addition to the machine learning-based approach, this research also explores feature-based approaches for object detection and location. Feature-based approaches involve extracting distinctive features from the object and matching them with the features extracted from the scene. The study compares two popular feature detection algorithms (i.e., SIFT and ORB). Both algorithms are widely used and have been verified to be robust to scale, rotation, and illumination changes. The SIFT algorithm detects key points in an image and computes descriptors for each key point based on the scale-space extrema. ORB, on the other hand, computes the descriptor using binary robust independent elementary features (BRIEF) with additional orientation information.

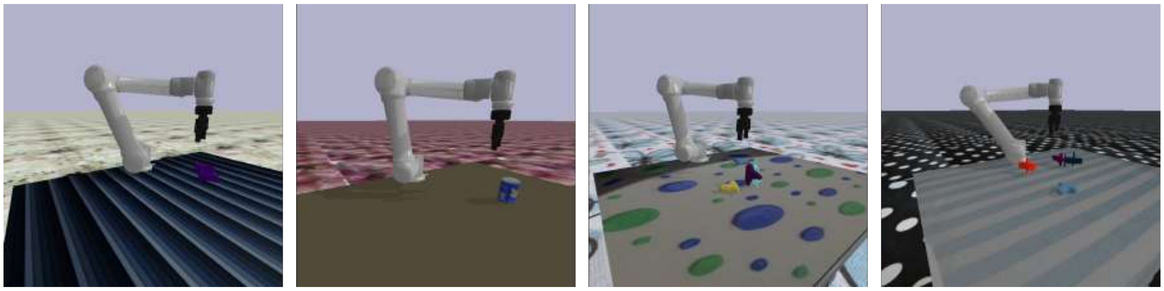


Figure 4.1. Sample scenes in the simulation environment, which includes a 6 degree-of-freedom (DoF) UR5e robot, a wrist camera, and randomly generated objects and backgrounds.

To locate the object using feature-based approaches, features were first extracted from both the object template and the input image using either SIFT or ORB. This work then matches the extracted features from the template and input images using brute-force matching, which results in a set of candidate correspondences. To obtain the final set of correspondences, Random Sample Consensus (RANSAC) algorithm is used to filter out any outliers. Finally, the method locates the object by drawing a bounding box around the set of matched points.

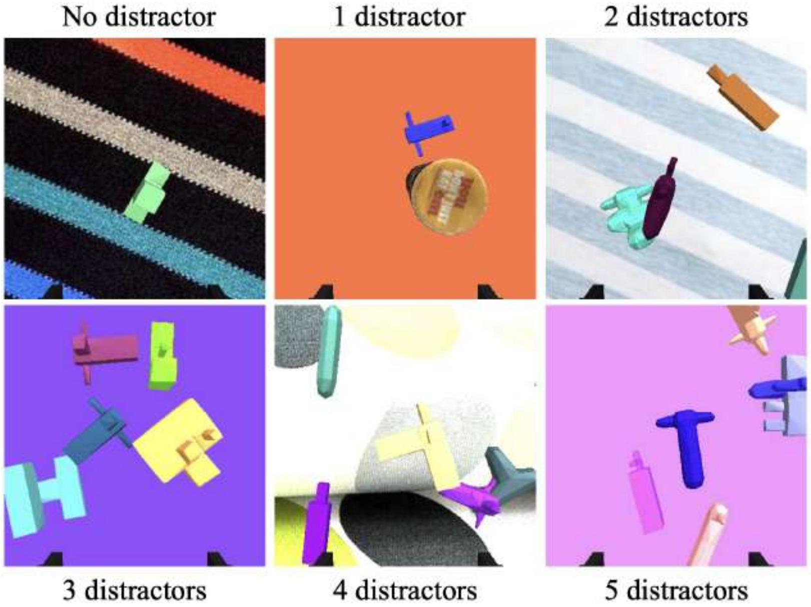


Figure 4.2. Variation of distractors in the image, ranging from 0 to 5, can affect the performance of IBVS.

It has been noted that feature-based approaches can be complementary to machine learning-based approaches, as they can provide an alternative means of object detection and location that may be more suitable for certain applications.

- **Hybrid Approach**

The hybrid approach combines the strengths of both machine learning and feature-based approaches to improve efficiency and accuracy. The definition of hybrid approach is vague. This research proposes a form that combines the use of a CNN for feature extraction with similarity comparison algorithms to locate the object in a lower dimension. The proposed method uses the VGG-16 backbone neural network to extract feature maps F_t and F_m from both the template image I_t and the input image I_i, respectively. The method first pass the template and input images through the VGG-16 network and extract the feature maps F_t and F_m from the last convolutional layer of the network. The feature maps are then normalised to have zero mean and unit variance. To locate the object in the input image, NCC [223] method is used to measure the similarity between the extracted feature maps as a function of their relative displacement. NCC is defined as the ratio of the cross-correlation of two images to the product of their standard deviations and is calculated as below:

$$\gamma = \frac{\sum_i \left[F_m - \bar{F_m}\right]\left[F_t - \bar{F_t}\right]}{\sqrt{\sum_i \left[Fm - \bar{F_m}\right]^2 \sum_i \left[F_t - \bar{F_t}\right]^2}}. \tag{4.1}$$

Here, $\bar{F_m}$ is the mean of F_m in the range under F_t, and $\bar{F_t}$ is the mean of F_t. The coordinate of the matching point (x_{max}, y_{max}) is located at the peak γ_{max} in the cross-correlation. This approach benefits from the ability of CNNs to extract high-level features from images and NCC's ability to accurately locate the object in a lower dimension, making it robust to scale, rotation, and illumination changes.

4.2.2 Category-Agnostic Localisation

Efficient and accurate object detection and localisation play a crucial role in enabling robots to understand and interact with their environment. Following the exploration of visual servoing methods, another question arises: How does the robot determine the target position without pre-defining a fixed location when it can be steered using visual information? This section proposes a two-stage method that takes an image query and a real-time image as input segments the target object and predicts the transformation from the query to the real-time image, providing valuable information for servoing and grasp planning. This method segments the target object and predicts the transformation from the query to the real-time image, providing valuable information for servoing and grasp planning. The advantage of this approach is that it can extract hidden information from the template, such as shape, colour, provided without learning the affordances of new objects, which makes the method more practical in brazing settings, where parts might be in random and irregular shapes, and able to extend to other tasks.

- **Problem Formulation**

The objective of this section is to develop a category-agnostic visual servoing method that enables robots to grasp arbitrary unknown objects using object images as queries. The proposed method aims to overcome the limitations of fixed-class approaches and eliminate the need for extensive datasets or manual labelling. By combining semantic segmentation and feature-based matching techniques, this research aims to enhance the perception and manipulation capabilities of robots in unstructured environments.

The problem can be mathematically defined as follows: Given an input image **I** containing both the scene and the target object, the task is to accurately localise and segment the target object within the image. Let **S** represent the segmented target object, which is a binary mask indicating the pixels belonging to the object and the background pixels. The method should be capable of handling arbitrary unknown objects with rich shapes, colours, and textures. Additionally, it should generalise well across different object categories without the need for specific training or prior knowledge about the objects.

Formally, the problem can be represented as finding the optimal segmentation mask $\mathbf{S}^*$ that maximises the objective function $f(\mathbf{S})$:

$$\mathbf{S}^* = \arg\max_{\mathbf{S}} f(\mathbf{S}) \tag{4.2}$$

where $f(\mathbf{S})$ captures the quality of the segmentation, considering factors such as object boundary accuracy, pixel-wise classification accuracy, and overall semantic consistency. The goal is to achieve high values of $f(\mathbf{S})$ for accurate and reliable object localisation and segmentation.

Furthermore, the method should be able to handle real-time applications and operate within the limitations of robotic systems, including limited computational resources, fast processing times, and low-latency requirements. The proposed method should provide a reliable and efficient solution for object detection and localisation tasks, enabling robots to interact with unknown objects in dynamic and unpredictable environments.

- **A Two-Stage Method**

To address the problem of reliable and efficient object detection and localisation, a two-stage approach consisting of object segmentation and transformation prediction is proposed. By separating the image domain prediction and spatial location control of the robot, interpretability and provide flexibility are enhanced, especially in scenarios where trajectory planning plays a crucial role. The proposed method is designed to provide a reliable and efficient solution that does not rely on closed-loop recurrent control or extensive training datasets, making it suitable for real-world applications with limited computational resources and manual labelling requirements.

In the architecture, as depicted in Fig. 4.14, the image and query pass through a Convolutional Neural Network (CNN), which outputs the foreground image and the background. The object segmentation stage utilises a pre-trained semantic segmentation model to obtain pixel-wise class predictions for the input image **I**. The segmented target object, denoted as **S**, is a binary mask that indicates the pixels belonging to the object and the background pixels. By leveraging deep learning and large-scale datasets, the segmentation model effectively distinguishes different object

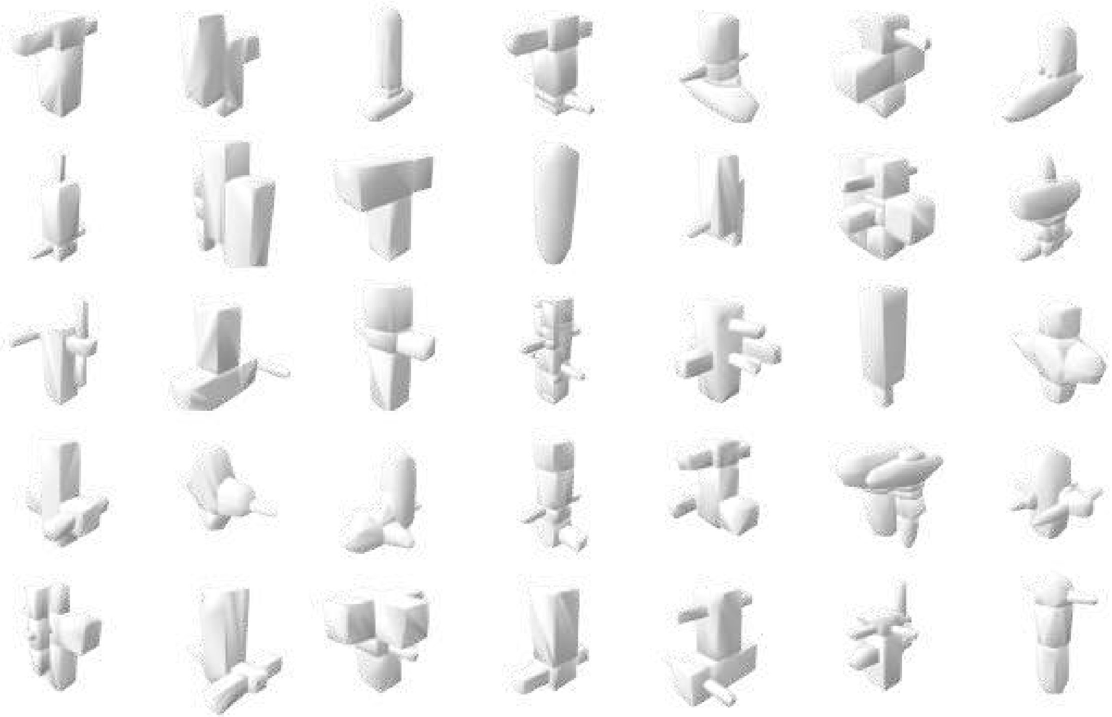


Figure 4.3. Example of randomly generated objects. These 3D CAD models are used for the generation of the data set and for the simulation real-time object.

regions within the image. To achieve category-agnostic segmentation, the model were modified to predict only two classes: objects and all other background. This approach can be seen as a background removal technique, allowing for the identification and isolation of objects for further processing.

Once the target object has been segmented, the transformation prediction stage aims to estimate the spatial location and orientation of the object with respect to the query image. This step involves comparing the segmented target object **S** with a provided template image using keypoint matching and homography estimation techniques. The estimated transformation matrix **T** maps the coordinates of the target object to the template image coordinates. The best match is determined based on the number of inlier matches and the quality of the estimated transformation. By analysing the spatial relationship between the template and the segmented object, the method can predict the object's location and orientation in the scene.

To evaluate the effectiveness of the proposed method, experiments were conducted using various test scenarios and evaluate the accuracy of object localisation. This work compared the performance of the method against existing category-specific approaches and analyse its adaptability to different object categories and environmental conditions. Additionally, this work explored the potential for automatic dataset generation to enhance the method's flexibility and adaptability in various robotic tasks. By providing a reliable and efficient solution for object detection and localisation, the method aims to empower robots to interact with unknown objects in dynamic and unpredictable environments.

4.2.3 Category-Agnostic Servoing

Image-Based Category-Agnostic Visual Servoing

IBVS poses a significant challenge for robotic systems, as it involves detecting the object and controlling the robot arm based on image feedback. These tasks are further complicated by various interference such as changes in ambient lighting, distractions, and background clutter. Recent research suggests that reinforcement learning is a promising approach to learning efficient control policies for such tasks. This section presents a novel reinforcement learning-based visual servoing approach for grasping unseen objects, which employs domain randomisation to bridge the reality gap between simulation and the real world. The full-state observation capability of the simulator and design an estimator is leveraged to predict the position difference between the robot and the object. The reinforcement learning agent is then trained to use this estimation information to control the robot's movements. The proposed framework mimics the closed-loop system of human perception and action, with the eyes (perception), hand (robot controller), and brain (RL agent) working in concert. Through extensive experimentation, it has been demonstrated that the framework is highly integrated, with each module complementing and enhancing the performance of the others.

- **Framework Overview**

Followed by the idea from the last chapter, the approach is based on a template matching algorithm that utilises the deep features extracted from a pre-trained Convolutional Neural Network (CNN) to track the object of interest. The proposed method processes the raw visual information and controls the robot in two separate modules: visual feedback and control. The visual feedback module extracts the deep features from the current image and compares them with the template features to obtain the error. The feature extraction module provides feedback for the RL controller, reducing the exploration difficulties for the RL algorithm and enabling faster convergence to a robust policy. The novel algorithm combines the robustness of both the backbone convolutional neural network (CNN) and the RL algorithm. The control module generates the control command based on the error and sends it to the robot controller to move the robot arm in the image plane towards the target object.

The proposed IBVS structure is illustrated in Fig. 4.4. It is designed to enable a robot to servo to a desired image without the need for further programming or prior knowledge about the exact location of the object or the image Jacobian. The framework consists of two main components: the RL agent and the environment that the RL agent interacts with. At each time step, the current extracted visual features $f(t)$ and the desired visual features $f^*(t)$ are used to calculate the error $e(t)$, which is the difference between $f(t)$ and $f^*(t)$. This error signal serves as the input to the controller used by the RL agent.

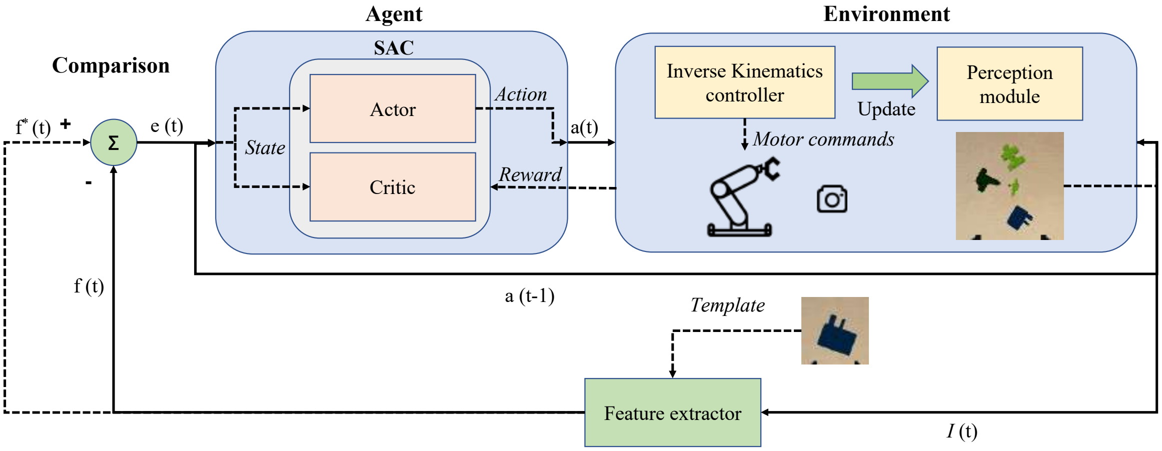


Figure 4.4. The proposed RL-based IBVS scheme. The agent, which applies SAC algorithm, receives as input an error estimation and the previous action. The environment receives the end-effector velocity as an input and calculate the corresponding motor command with inverse kinematics and takes an image after the agent action being executed. The feature extractor, maps the provided template image with the current image and estimates the feature.

The objective of feature extraction is to derive the error $e(t)$ in the image by extracting features such as the change of the object position compared with the last frame image. This $e(t)$ is then used for controlling the robot's motion. Accurately matching the task object in the current frame is crucial for feature extraction. There are generally four methods used for feature extraction in the computer vision field: template matching, feature matching, shape/outline detection, and data-driven matching. Traditional template matching requires translation consistency, and it is sensitive to deformation and other appearance changes. Feature matching is less sensitive to complicated deformation but requires complicated shapes to extract enough feature points, and can be time-consuming. Shape detection uses morphological transformations to detect the outline but is susceptible to noise. In

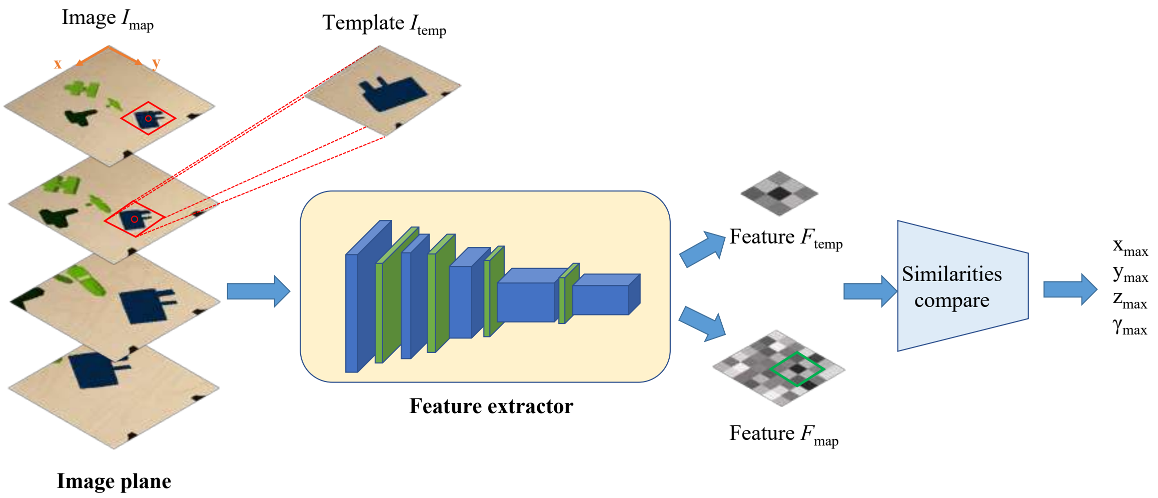


Figure 4.5. Architecture of the proposed IBVS framework. The robot moves in the world frame while capturing images at different time steps. In each time step, the captured image is pre-processed and fed into a feature extractor to extract deep features. The similarity is computed by comparing the template feature and the current map, which produces the estimated position of the object in the image plane and a confidence level.

this method, a CNN is used as the feature extractor. Compared to the other methods mentioned, the CNN feature extractor is less vulnerable to noise, and does not require intrinsic camera parameters or the use of hand-eye calibration to obtain the extrinsic camera parameters. In particular, this research is interested in a general model that can be applied directly to different unseen objects without human engineering, where there is no need to design hand-crafted models or fine-tune the parameters for a specific object.

To locate the target object in the current image, a deep CNN is used to extract feature vectors F_{temp} and $F_{map}(t)$ from a low-resolution template image I_{temp} and a full-resolution current image I_t, respectively. To compare the similarity between the two feature vectors, this experiment use normalised cross-correlation (NCC) [223], which is the same module used in the proposed category-agnostic visual servoing algorithm introduced in the last chapter. Specifically, the experiment computes the distance between F_{temp} and $F_{map}(t)$ as follows:

$$\gamma = \frac{\sum_i \left[F_{map} - \bar{F}_{map}\right]\left[F_{temp} - \bar{F}_{temp}\right]}{\sqrt{\left\{\sum_i \left[F_{map} - \bar{F}_{map}\right]^2 \sum_i [F_{temp} - \bar{F}_{temp}]^2\right\}}}, \tag{4.3}$$

where $\bar{F}map$ is the mean of $Fmap$ in the range under $Ftemp$ and $\bar{F}temp$ is the mean of the $Ftemp$. The coordinate of the matching point $(xmax, y_{max})$ is located at the peak γ_{max} in cross-correlation.

- **Reinforcement Learning based Controller**

The IBVS system aims to minimise the error between the current and target image features. This research addresses the problem of learning to move the robot to a desired location in an image plane without any prior knowledge of the object shape, pose information, or camera parameters. In the setup, only a single template image is provided, and any additional information provided by augmented reality markers or camera calibration is not available. Therefore, the feature Jacobian cannot be explicitly calculated. The problem is formalised as a Markov Decision Process (MDP) framework $< S, A, P_{s,a}, R >$ over discrete time steps in an environment E, where S, A, and R denote the sets of states, actions, and rewards. $P_{s,a}$ is the transition probability that describes the probability of the agent moving from state $s(t)$ to a new state $s(t{+}1)$ after taking the action $a(t)$. The final objective of the RL agent is to find a policy π that predicts an action $a(t) \in \mathcal{A}$ based on the observation of the state $s(t) \in \mathcal{S}$ at each time step t, which maximises the expected reward $\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$. The cumulative expected reward can be written as:

$$\mathbb{E}_{(s(t),a(t))\sim\rho_\pi}\left[\sum_t R\left(s(t), a(t)\right)\right]. \tag{4.4}$$

In the previous section, the feature extraction scheme that results in a higher dimensional feature value was introduced. The environment receives (x_{max}, y_{max}) and NCC score γ_{max} from the feature extractor, which are combined with the last step action $a(t-1)$ to form a 4-dimensional vector:

$$s(t) = \{e_x(t), e_y(t), \gamma(t), a(t-1)\}, \tag{4.5}$$

where $e_x(t)$ and $e_y(t)$ represent the normalised error of the template centre to the matched centre in the image plane. The reward function is defined as:

$$I_t = \begin{cases} 0, & e(t) \geq e_{thresh} \\ -1, & e(t) < e_{thresh} \end{cases} \tag{4.6}$$

This function calculates the error in the image plane, given by $e(t) = \sqrt{(e_x(t))^2 + (e_y(t))^2}$, and provides feedback based on whether the threshold value is exceeded.

- **Feature Extractor Model**

A NCC-based template matching approach is used, as proposed by [222], as the base for the perception module. This approach takes scale adaptive deep convolutional feature vectors from both the template and the current frame image using a pre-trained VGG-16 network and measures the similarity of the output features to locate the target. The feature extraction module is initialised with the template image of the task object, and use the maximal correlation value on the map to find the target location in the image plane. It is important to note that the peak correlation value of the same image is not always located at the centre of the image, so location at initialisation must be found this. Once the feature extraction module has located the target, it provides features to the reinforcement learning algorithm, which generates a motor command that moves the robot in the image domain.

- **Training Methodology using Soft Actor-Critic RL**

For training the agent, this research chooses Soft Actor Critic (SAC) [225], which is an off-policy maximum entropy RL algorithm. In addition to maximising the cumulative reward as described in Equation 4.4, SAC algorithm also aims to maximise the entropy of the action taken by the agent. The modified expected reward used in SAC is given as:

$$\mathbb{E}(s(t), a(t)) \sim \rho\pi \left[\sum_{t} R\left(s(t), a(t)\right) + \alpha H\left(\pi\left(\cdot \mid s(t)\right)\right) \right]. \tag{4.7}$$

Here, α is a temperature parameter that determines the trade-off between maximising the expected reward and maximising the entropy of the action distribution. This parameter is tuned to balance exploration and exploitation during training. During training, experience by running multiple episodes of the task is collected, each consisting of a sequence of state-action pairs. The experience is then used to update the policy and value function networks using stochastic gradient descent. The Q-values, Actor, and Value function are trained with the Adam optimiser using the same learning rate of 3e-4. The RL agent is trained and tested using an episodic RL setting. At the beginning of each episode, a random object is generated on the table. The robot arm moves towards the object to fetch the template image and initialise the feature extractor. After initialisation, the task scene is created, including resetting the robot motor to a fixed position and generating the target and distractors in the FoV of the wrist camera. Additional noise is added to the environment by varying the light direction, brightness, and diffusion coefficient. In each episode, the agent is rewarded according to Equation 4.6. Setting the error threshold e_{thresh} too large or too small can lead to task failures. The error threshold is set to 0.1. A looser definition for the threshold would lead to a jittery robot arm, as there is no penalty for the jitter, and a stricter definition would lead to longer exploration time before the algorithm converges. Early termination occurs in two scenarios: when the object is out of the FoV for ten time steps, and when the robot arm moves out of the preset boundaries. In both cases, the total reward is set to -200, which is the highest penalty term that one episode can receive.

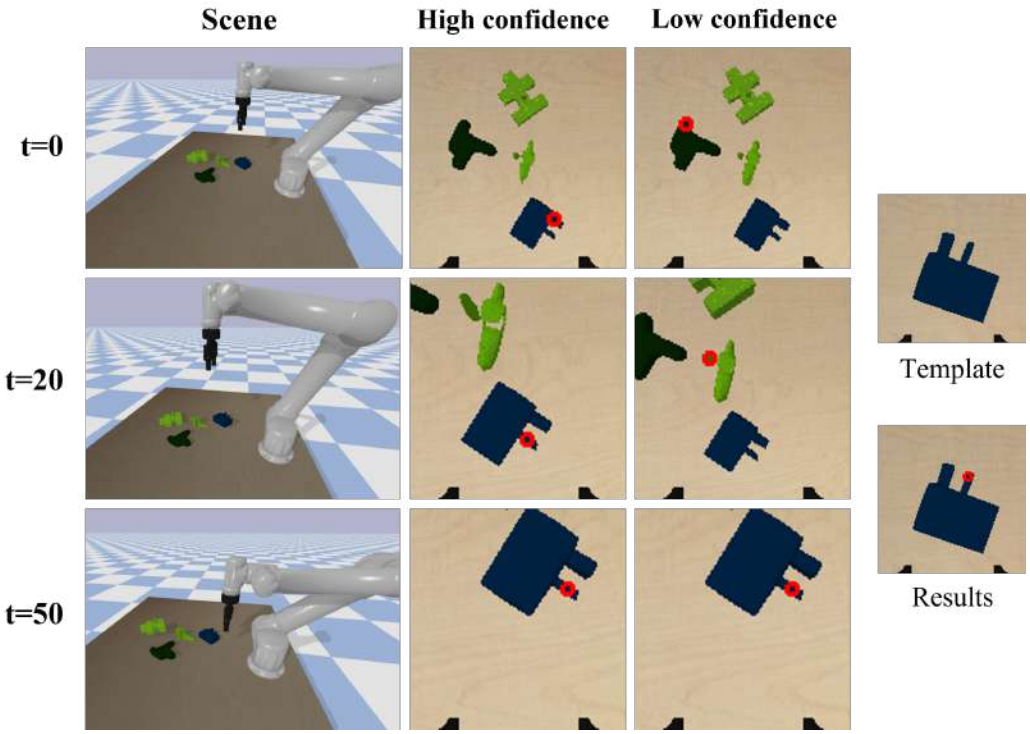


Figure 4.6. Examples of visual servoing process in simulation.

Position-Based Category-Agnostic Visual Servoing

In contrast to IBVS, PBVS plans the movement based on estimated robot's relative position. The framework consists of an eye" that estimates the approximate position of the target object, and a brain" that guides the robot arm to reach the object through iterations of eye estimation and action. To this end, an estimator to estimate the position difference between the target view and the current view is trained in an "eye-in-hand" setting. Furthermore, a reinforcement learning-based policy network is introduced to guide the robot arm with the estimation information. Unlike the prior IBVS framework, which relies on feature extraction and matching in the image space, this PBVS approach explores reducing reliance on hand-designed perception modules in an end-to-end fashion. By directly estimating 3D positions, it enhances the robot's versatility in complex environments but introduces more uncertainties and algorithmic challenges.

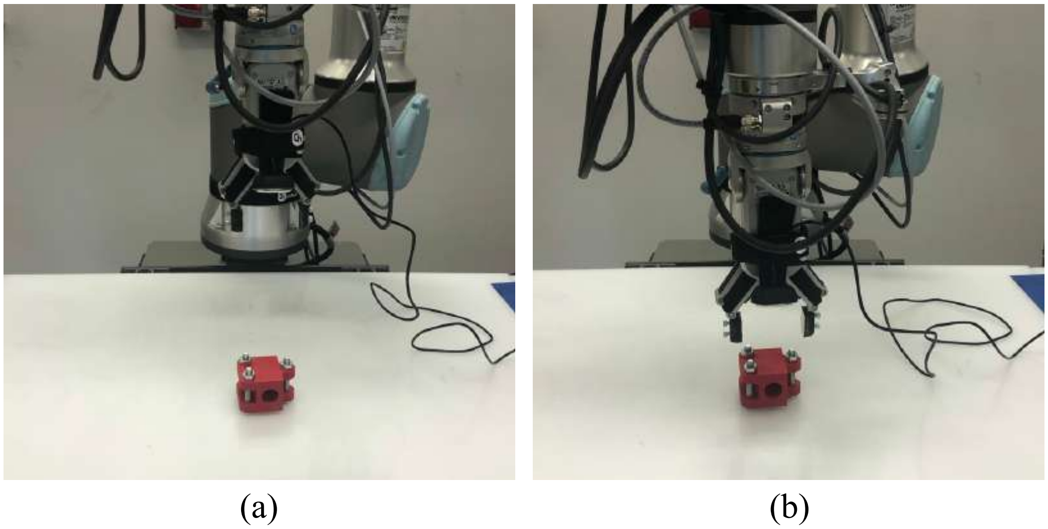


Figure 4.7. Real-world experimental setup for testing the proposed framework. (a) Initial setup before testing. (b) Experimental setup at the end of the testing phase.

To achieve this, the method aims to recognise the target object and to establish a relationship between actions and the changes in the image domain that result from those actions. As a result, the proposed framework can be divided into two interconnected components: (1) an estimator that calculates the positional difference between the current state and the target state, and (2) a policy network that uses the estimator's output to generate the robot's next action. Fig. 4.12 provides an overview of the approach.

- **The Estimator Module**

The estimator module is a key component of the proposed method, as it is responsible for estimating the position difference between the template image and the current image. Specifically, this module takes two images as input, the template image I_0 and the current image I_t, and is expected to output the difference vector which indicates the direction and magnitude of the required movement to reach the target. To achieve this, a CNN with two branches is used, where the first branch processes the template image and the second branch processes the current image. The two branches are then combined and passed through a fully connected layer to generate the final output. In this way, the estimator module establishes a mapping between the visual features of the observed state and the corresponding action required to move towards the target.

- **The Estimator Network Architecture**

The estimator module uses ResNet 50 [224] as the back-end and consists of the convolutional layers of ResNet 18 pre-trained on ImageNet to extract image features. Both the template image and the current image are fed through the same network, and the output image features of the two images are concatenated before being passed through three fully-connected (FC) hidden layers and an output layer. The three FC layers consist of 1024, 256, and 128 nodes, respectively, and use the ReLU activation function [226]. The output of the estimator module contains four nodes: an estimated position difference vector pointing to the goal position and the magnitude. The network is trained using smooth L1 loss, which is calculated by comparing the predicted pose difference y_{pred} with the ground truth pose difference y_{true}.

- **Training the Estimator**

Data collection and algorithm training are performed in a simulated environment. The process starts by randomly generating an object with a random position within the robot's field of view (FoV). The robot arm is then moved randomly to approach the object and capture the template image I_0. To ensure that the estimator learns only the relative spatial relationship between the images, instead of learning the camera parameters, the object is moved to a different position. Thus, the input to the estimator consists of two images captured at different positions with different camera relative locations. 10,000 such pairs of images are generated to form the training set. The training process consists of two phases. In the first phase, the pre-trained ResNet 18 model is used to initialise the feature extractor's weights. In the second phase, the entire estimator network is fine-tuned using the Adam optimiser [227] with a learning rate of 0.001 and a batch size of 32. The network is trained for 200 epochs with early stopping if the validation loss does not improve after 20 epochs. Smooth L1 loss, also known as the Huber loss, is used to calculate the difference between the predicted position difference vector and the ground truth position difference vector. The trained estimator network is then used to provide input observations to the policy network during the interaction with the environment.

- **RL For Policy Training**

To train the policy network, this method adopt Soft Actor-Critic (SAC) [228, 229] as the RL algorithm. SAC is built under the Maximum Entropy RL framework and has shown to be an effective off-policy RL algorithm for continuous control tasks. To further enhance the sample efficiency and stability of the training process, the Hindsight Experience Replay (HER) [230] technique is applied. HER allows the agent to learn from failures by relabelling the original transitions with states from past episodes in the replay buffer, which can lead to higher sample efficiency and faster convergence. As the goal is to reach a specific object, sparse and binary rewards are

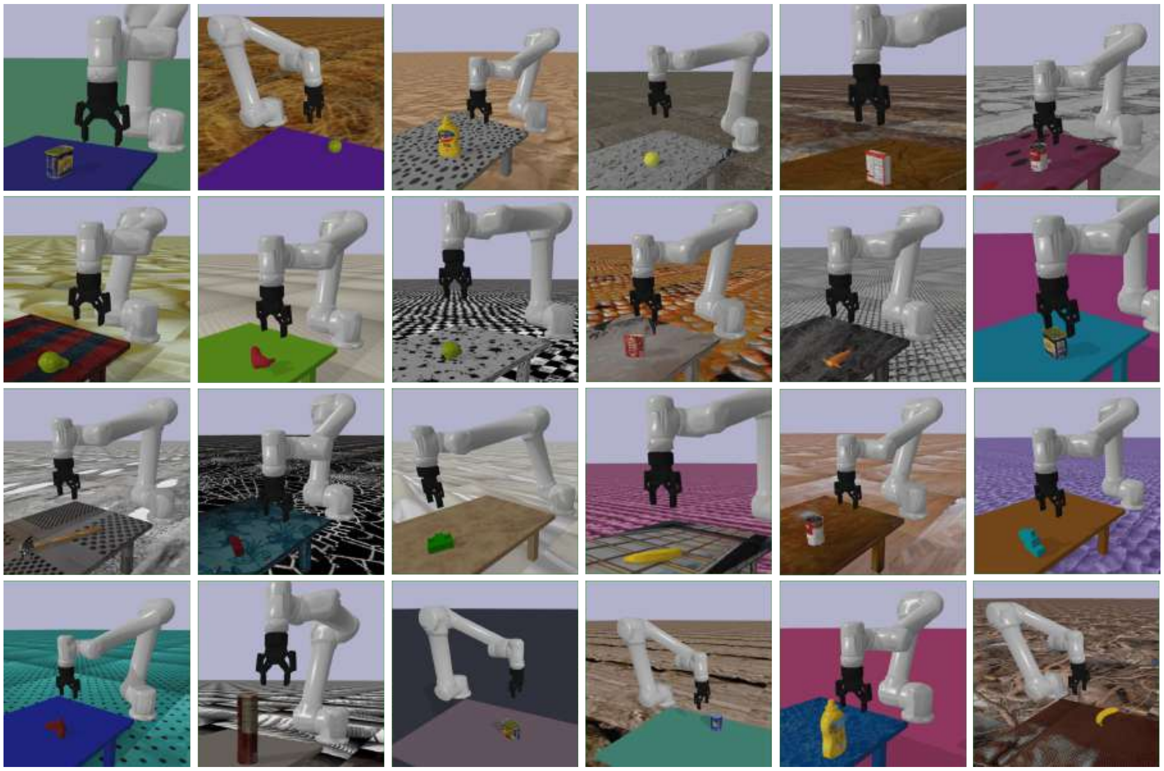


Figure 4.8. Simulated scenes with various task objects and textures were captured from both goal and random positions to generate a randomized dataset for training and evaluation.

used instead of carefully shaped rewards. This is particularly important in scenes with sparse rewards and high-dimensional state spaces, where it is difficult for the algorithm to receive meaningful feedback.

The observation space for the RL algorithm is the output of the estimator module, which is a low-dimensional representation of the spatial relationship between the goal position and the current position instead of raw images from the RGB camera. This approach significantly reduces the complexity and state dimension of the original task while retaining the critical information required for the robot to reach the goal position. By treating the prior estimator's output as the observation of the RL algorithm, the need for hand-eye calibration and carefully-tuned control algorithms is eliminated. Additionally, RL is capable of handling uncertainties in the environment, reducing the impact of potential errors in the estimator on the overall movement of the robot.

r_t is the reward received by the reinforcement learning agent at time step t. In this algorithm, the while loop continues until the reward is not equal to -1. $\boldsymbol{x}_z(t)$ is

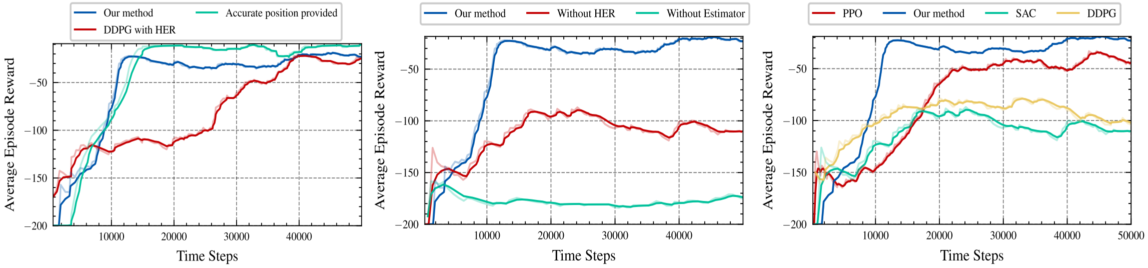


Figure 4.9. Left: compare the average episode reward of the proposed method with DDPG with HER replay buffer and replace the output of the estimator module with an accurate position difference. Middle: ablation experiments. Right: compare to other RL algorithms.

the height of the robot end-effector at time step t. This is used to determine if the robot has moved upward or not. The loop continues until the end-effector reaches the desired object position, which means it should stop moving upward and $\boldsymbol{x}_z(t)$ should not increase anymore. $\boldsymbol{x}_z(0)$ is the height of the robot end-effector at the initial position. This is used as a reference point to compare with $\boldsymbol{x}_z(t)$ and determine if the robot has moved upward or not.

4.3 Experiment

4.3.1 Category-Agnostic Vision Algorithms Comparison

This experiment utilised PyBullet [70] platform, as describe in the previous section, to evaluate the approach in simulation. The simulated environment consists of a 6 DoF UR5e robot and a wrist camera with a field of view of 60 degrees and an image size of 256 × 256 pixels. The robot is controlled using a Cartesian space position controller, and the simulation includes a set of randomly generated rigid objects or daily objects.

The generation of CAD objects begins with primitive geometric shapes serving as base components, where cubes, cylinders, and spheres are parametrically defined with randomly sampled dimensions within predefined ranges suitable for brazing applications. The shape composition phase combines multiple primitives using boolean operations including union, intersection, and difference to create complex geometries. Each object is composed from a varying number of primitive shapes, typically between two and five, arranged in a hierarchical composition tree. Spatial relationship constraints ensure the physical validity of the resulting objects while maintaining sufficient complexity to challenge the localisation algorithm.

These composed shapes subsequently undergo surface modifications through several transformations. A Perlin noise function applies random perturbations to surface points, creating natural variations in the object geometry. Smoothing operations maintain manufacturability while edge rounding and filleting operations enhance realism. The scale of these features varies randomly to test the algorithm's robustness across different size domains.

The background of the working area is generated randomly and it includes both pure colour backgrounds and texture backgrounds. The simulation environment is depicted in Fig. 4.1, which shows sample scenes with randomly generated objects and backgrounds. To evaluate the robustness of the algorithms to changes in the environment, the number of distractors in the scene and the lighting conditions were varied, including the distance between the light source and the object being rendered, the amount of ambient light, and the amount of diffuse and specular lighting in the scene.

This experiment compares the accuracy, efficiency, and robustness of the proposed IBVS techniques. The accuracy and robustness of the algorithms are evaluated using Intersection over Union (IoU), which is calculated by measuring the overlap between the predicted and ground truth bounding boxes. This work tested the algorithms on a set of 50 different objects with varying textures and backgrounds while controlling the camera parameters and environment. Additionally, to add difficulty to the evaluation, distractors were introduced by creating cluttered environments for 2D image-based algorithms. This work believes that these evaluations would provide valuable insights into the suitability of different algorithms for IBVS applications.

To evaluate efficiency, the processing time of different algorithms were tested, which directly affects the performance of the robotic system. In an IBVS system, the robot must be able to quickly and accurately locate the object of interest in the image frame and use that information to guide its motion toward the desired goal. If the object recognition and location process is slow, it can significantly degrade the overall performance of the system, leading to slower and less accurate robotic movements. This can be particularly problematic in applications where the robot needs to perform tasks in real-time or where there are time-sensitive constraints. Therefore, it is crucial to develop fast and efficient algorithms for object recognition and location that can meet the speed requirements of the IBVS system. The efficiency evaluation results are presented in Table 4.1, which reports the average processing time in seconds and the standard deviation across multiple trials.

The images in Fig. 4.2 illustrate the effect of adding different numbers of distractors to an image in an IBVS system. The images show examples of the same object with varying degrees of clutter and complexity, which can affect the speed and accuracy of object recognition and localisation algorithms. This is a demonstration of the importance of developing robust and efficient algorithms that can handle varying levels of clutter and background noise in IBVS applications.

Table 4.1. Efficiency Evaluation

Algorithm	Avg (%)	Std
Semantic	0.136	0.0979
ORB	0.0335	0.0672
SIFT	0.0198	0.00614
Hybrid	0.867	0.187

Table 4.2. Accuracy Evaluation with Different Numbers of Distractors (Accuracy %)

Algorithm	0	1	2	3	4	5
Semantic	91.5	-	-	-	-	-
ORB	83.9	57.2	49.2	29.9	32.7	4.63
SIFT	88.9	73.9	50.7	34.2	54.6	29.8
Hybrid	89.1	74.2	69.6	58.3	35.0	26.4

- **Data Generation and Dataset Preparation**

To ensure the reliability and validity of the experimental findings, a diverse and extensive dataset for object detection and localisation is created. The fine-tuning process of the pre-trained semantic segmentation model required collecting both RGB images and their corresponding labelled segmentation images. However, manual labelling can be a time-consuming task. To overcome this challenge, this experiment employed a combination of synthetic data generation techniques, real-world textures, and scanned 3D objects to automatically generate a large volume of template images, RGB images, and their segmented counterparts. The dataset encompassed a wide range of objects with varying shapes, sizes, textures, and colours. Variations in lighting conditions, backgrounds, and camera placements have also been introduced to realistically simulate various real-world scenarios. By utilising this diverse and rich dataset in the experiments, the performance of different IBVS methods under a variety of challenging conditions were able to comprehensively evaluated.

In addition, to efficiently train a category-agnostic semantic segmentation model, this experiment employs an automatic data generation process in simulator. The experiment created a synthetic training dataset comprising 10,000 images, image queries, and corresponding ground truth mask sets. The ground truth mask provides binary segmentation information, distinguishing between the objects and the background. Each image-query pair consists of random objects, including the target object, along with a variable number of distraction objects ranging from 1 to 6. This work generates a diverse range of background textures using domain randomisation techniques. This process ensures that the trained network can handle different background variations encountered in real-world scenarios. In addition, to

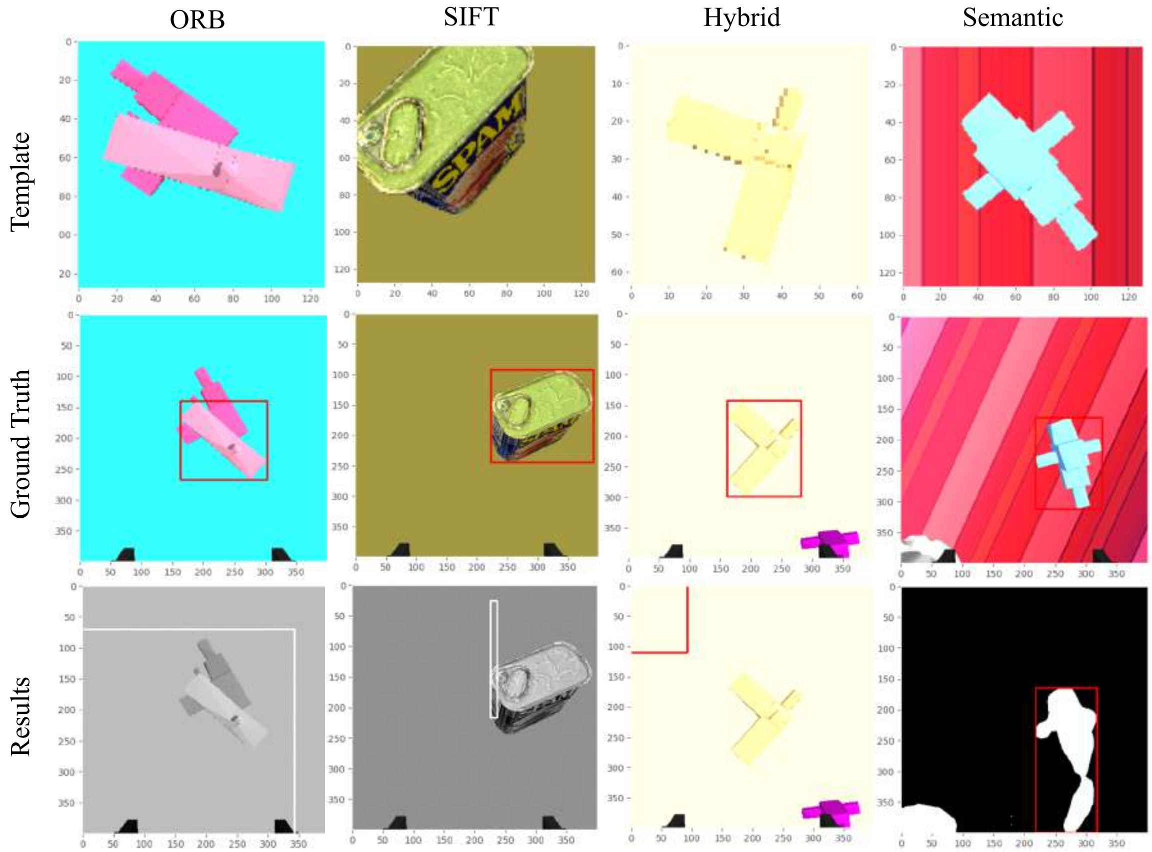


Figure 4.10. Failure Analysis.

enhance the robustness of the network and enable seamless transfer from simulation to reality, additional randomisation factors were introduced. These factors include object poses, camera poses, and light conditions. By varying these parameters, a more comprehensive training dataset was created that captures the variations encountered in practical environments.

To simulate realistic camera-in-hand robot settings, this work constrain the camera movement to the x-y plane, maintaining a top-down view. This restriction ensures that the camera's perspective resembles that of a robot operating in a real-world scenario. After generating the images, the distraction objects were removed from the scene, leaving only the target object. This work then generates a randomised picture of the target object with varying positions, orientations, and lighting conditions. Utilising the full state accessibility provided by the simulator, masks are computed that determine the set of pixels belonging to the target object in the RGB image. These computed masks serve as ground truth labels for training the network.

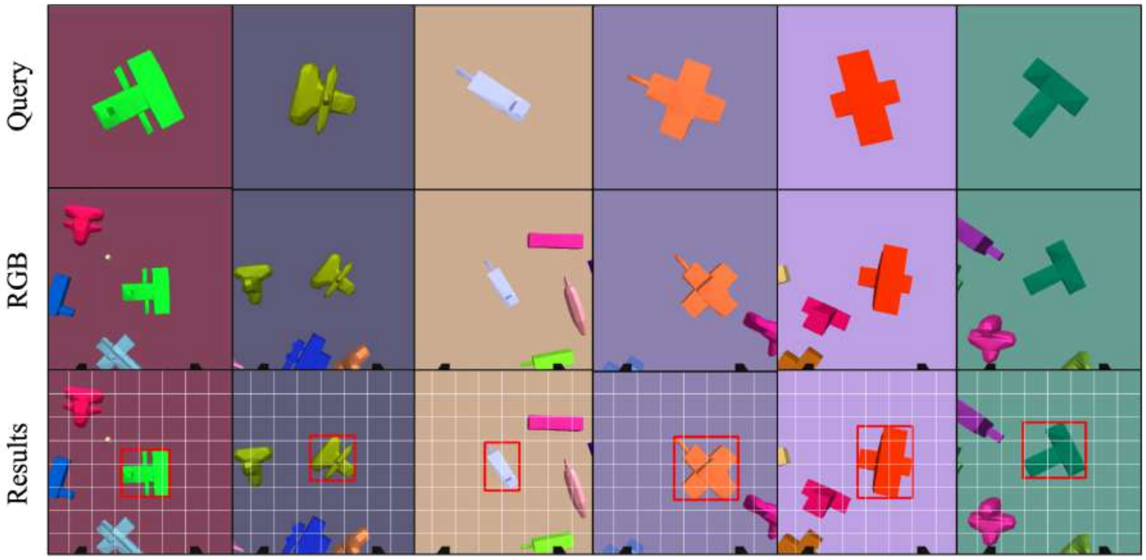


Figure 4.11. Example results for query object localisation in different random scenes. The predictions for the RGB image are boxed. Each input image represents a randomly generated scenario where arbitrary objects are added to an unseen background within the field of view.

Additionally, the ground truth bounding box of the target object was computed, providing precise spatial information for evaluation purposes. By leveraging automatic data generation, a diverse and extensive training dataset is ensured, facilitating the training of a category-agnostic semantic segmentation model capable of accurately localising target objects in real-world scenarios.

4.3.2 Category-Agnostic Localisation

In this section, the experimental setup used to evaluate the effectiveness of the proposed method for object detection and localisation is presented. A series of experiments are conducted to assess the performance, robustness, and generalisation capabilities of the method in various scenarios. The experiments involved both synthetic images and real-life images.

The experiments were conducted with the basic version PyBullet environment which details are given in the last chapter. This simulation environment consisted of a UR robot equipped with a camera system. This experiment utilised a combination of generated datasets and real-world images for evaluation.

To ensure that the network did not have prior exposure to the objects, unseen objects were introduced in the simulator. A total of 1,000 random objects were generated, with 200 objects used for dataset generation and 100 objects reserved for testing. Examples of the arbitrary-shaped objects are demonstrated in Fig. 4.3. The simulation testing scenario consisted of a table with randomly generated objects placed on it. The number of objects on the table was set to 5 or 6 to provide a challenging task.To validate the proposed method, the category-agnostic semantic segmentation results with the ground truth generated were compared using the information provided by the simulator.

For the real-world dataset validation, four different experimental conditions to thoroughly validate the performance of the method were conducted. Each experiment was designed with variations in background, light direction, light intensity, and object arrangement. By introducing these diverse conditions, this work aimed to increase the challenge and evaluate the robustness of the approach in different real-world scenarios. The variations in background provided a test for the method's ability to distinguish objects from complex and cluttered environments. Different light directions and intensities challenged the method's capability to handle variations in illumination conditions, simulating realistic scenarios where objects may be encountered under different lighting conditions. Additionally, the variations in object arrangement assessed the method's ability to accurately detect and localise objects in different spatial configurations. Through these experiments, valuable insights into the performance of the method across various challenging real-world conditions can be obtained, further validating its effectiveness and potential for practical applications. Additionally, the method's performance on the generated dataset is evaluated to assess its generalisation capabilities. The training dataset comprised 9,000 images, while 1,000 images were allocated for validation.

The proposed method utilised the DeepLab V3 model with a ResNet-50 backbone for improved efficiency. To adapt the ResNet-50 to the requirements, the number of output classes are reduced to two: foreground object and background. The model was implemented using the PyTorch framework and trained using the generated dataset. The Adam optimiser was used with a learning rate of 0.00001. The training process was conducted on an RTX 3060 GPU.

4.3.3 Category-Agnostic Servoing

Image-Based Category-Agnostic Visual Servoing

This section provides details on the implementation of the proposed IBVS method. PyBullet [70] is utilised to evaluate the approach in simulation. Similar to the basis simulation environment, the simulated environment includes a 7 degree-of-freedom UR5e robot. An additional wrist camera is included in this experiment with a field of view of 60° and an image size of 256 × 256 pixels. Before each episode of training, a template image of size 32 × 32 pixels is captured. The robot is controlled using a Cartesian velocity controller, and the simulation includes a set of randomly generated simple rigid objects.

The environment is wrapped as a goal-oriented environment that takes three inputs to form the state $\mathcal{S}$: the achieved goal (the current position), the desired goal (true object position), and the observation (information provided by the estimator). At each time step t, a live image from the camera is captured and fed into the estimator module along with the template image to obtain a predicted value. This value is passed to the RL algorithm to predict a robot motor command $\mathcal{A}$. By using the estimator output as the observation, the complexity of the task is reduced and the impact of potential errors from the raw images captured by the camera. Sparse rewards were used to specify the task aims implicitly. The RL agent receives a negative value if the distance to the desired position is larger than a preset threshold value, or 0 if it is smaller than the threshold value at each time step t.

- **Data Generation for IBVS**

Learning algorithms training requires the robot to interact with the environment. This exploration in environments usually takes a lot of time for training which is more particular in environments with spare rewards and large action or/and observation dimension. In recent years, some techniques [230] or sample-efficient algorithms [41] has been proposed to solve this problem. In the research, instead of using sparse rewards or hand-designed rewards, a deep learning-based reward function is proposed which provides abundant real-time feedback for the training of the RL algorithm.

Therefore, data must be collected to train this reward function. In this settings, as the observation is provided by the the on-hand camera, RGB image is the first choice for the input of the reward function. The target of this reward function is the improvement of the previous action. If the robot managed to move towards the target gesture, the reward will be positive, and accordingly, the reward will be negative when moving away. This means that the reward function will have at least three inputs: image of the target gesture, current image and previous image. The output is the estimated position and orientation differences which forms a seven element vector (three elements for the x-axis, y-axis and z-axis, and four elements orientation differences in quaternion).

To generate input-output pairs for the reward function, the following steps are undertaken:

At the initial/reset position, the robot arm's end effector vector is set to be perpendicular to the ground. During the data generation process, the robot arm's gesture should gradually move from the bottleneck position to the initial/reset position. This is done by setting the value of the displacement $\boldsymbol{x}(t)$ or ration as the value of a normal distribution function, which the expected value is proportional to the difference value between the final state $\boldsymbol{x}_{final}$ and the current state $\boldsymbol{x}_{current}(t)$:

Algorithm 3 Data generation and pre-processing

```
Initialise the environment and the Cobot. Generate the task environment with
  random texture and a random object for reaching.
repeat
    repeat
        Generate a random relative gesture (or pre-designed gesture) where the Cobot
          will move to.
        Take a image on this gesture which is used as the target image for this set of
          data.
        Record the target gesture.
    until The number of this dataset is satisfied.
until The total group number of the dataset is satisfied.
Output: Groups of dataset with different object, texture, and starting position.
```

$$\boldsymbol{x}(t) \sim \mathcal{N}\left(\mu(t), \sigma^2\right), \mu(t) = (\boldsymbol{x}_{final} - \boldsymbol{x}_{current}(t))/350, \tag{4.8}$$

where the proportional value is set to 350 to simulate a smooth return-to-centre speed. After the raw data is generated, data must be processed to form the input-output pairs to be able to be fed into the neural network. The input comprises three images: the target image, the current image, and the previous image. The reward function has two layers: the first estimates the gesture difference, and the second compares two images taken at different times, providing marks based on whether the robot moves closer to the target gesture. To align with this input paradigm, one image from a group that is not the target image of this group is taken out and concatenated with the target image of this group.

Position-Based Category-Agnostic Visual Servoing

The reaching process can be viewed as an interaction between the robot and the RL agent, as illustrated in Fig. 4.12. At each time step t, the agent takes an action to move the robot toward the target, based on the output of the estimator. The action is the desired velocity of the end-effector, which is converted into joint positions using inverse kinematics. Once the robot reaches a new position, it captures a new image and sends it to the estimator, which provides a new observation that is passed back to the agent. This forms a closed-loop scheme that is executed repeatedly. During the training stage, the agent receives a reward in addition to the state observation, which indicates the goal. The reward is calculated based on an accurate position, which is only available in the simulator.

- **PBVS Data Collection**

To train the estimator part, the image and the position difference pair were

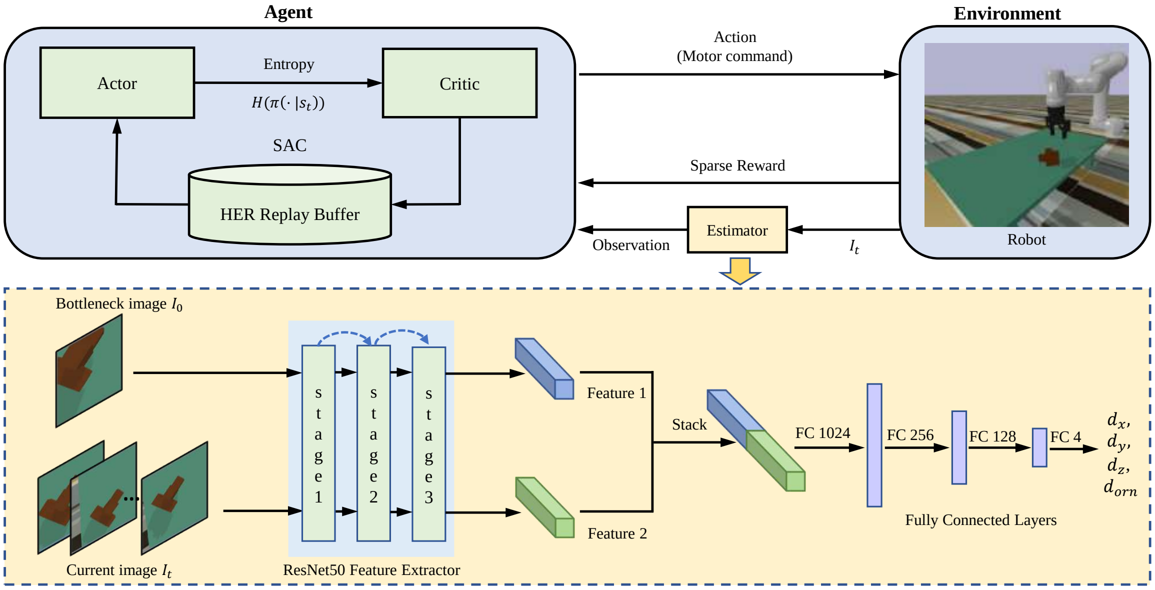


Figure 4.12. Overview of the proposed method consisting of two main parts: estimator training and policy training. The dataset for training the estimator is generated through human demonstrations (bottom). At each time step t, the estimator takes the template image I_0 and the current image I_t as inputs and provides the observation to the agent. The adapted estimator network is used to train the policy network while the agent interacts with the environment (top).

collected to form the training dataset. In specific, two images were collected, which are a template image of a random object and an image of the same object taken at another position, and their real-world distance to form an input-output pair.

Data is generated in the simulator and the data collection is run automatically. First, the robot and a random object are reset to a random initial position, and then the object is generated randomly on the table. Secondly, a template image is automatically collected. Starting from this position, the robot arm moves upward by sampling from a set of actions. These actions have different time-varying probabilities to ensure the robot moves upward overall. There are seven actions in the data collection process: staying still, moving along the plus and the minus x-axis, moving along the plus and the minus y-axis, and moving along the plus and the minus z-axis, respectively.

A collection process is defined, which includes the robot collecting data continuously in the environment with one initial human demonstration, as a group of data set $\mathcal{D}_{pose}$. When the step is less than 30, the probability of moving along the plus z-axis is set to a higher value, and the minus z-axis is set to a lower value to avoid colliding with the task object. Then, the probability of actions is the same except for moving along the plus z-axis, which is dependent on the step taken and a little bit higher than other actions to ensure covering as many regions as possible.

4.4 Results and Discussion

4.4.1 Category-Agnostic Vision Algorithms Comparison

Based on the experiments, it has been found that testing algorithms' success rates under various lighting conditions are critical to assess their robustness to environmental changes. In the experiments, this work evaluated the performance of the algorithms under different lighting conditions, including changes in the distance between the light source and the object, variations in the ambient, diffuse, and specular coefficients. Various real-world lighting conditions were simulated and the algorithms' ability to generalise were tested by controlling these parameters. To better illustrate the results, Blinn-Phong model were used to calculate the illumination for a given set of ambient, diffuse, and specular coefficients. The findings suggest that machine learning based approach (semantic segmentation and hybrid approach) performs better under diverse lighting conditions and have a better potential for deployment in real-world applications.

The results of the accuracy evaluation with different numbers of distractors are presented in Table 4.2. It shows the accuracy percentages of four IBVS algorithms (Semantic, ORB, SIFT, and Feature) under varying numbers of distractors. The results show that Semantic performs the best with an accuracy of 91.5% on average, even with five distractors. ORB, SIFT, and Feature have lower accuracy compared to Semantic. The accuracy decreases as the number of distractors increases for all algorithms. The highest decrease in accuracy is observed for ORB, which drops from 83.9% without any distractors to only 32.7% with four distractors. SIFT also shows a significant drop in accuracy as the number of distractors increases. These results demonstrate the effectiveness of hybrid approaches over other algorithms in cluttered environments.

The failure images are shown in Fig. 4.10. In analysing the failure examples of the tested algorithms, it was found that ORB and SIFT algorithms were prone to failure in situations where there were occlusions, lighting changes, or when the object of interest was rotated or scaled. On the other hand, the semantic segmentation algorithm failed when the background texture was complex and resembled objects, leading to confusion between the object and the background. The hybrid approach showed better performance than the other algorithms in terms of accuracy and robustness. However, it was found to be more computationally expensive and slower than the other algorithms. These findings suggest that while the hybrid approach may be suitable for applications where accuracy is paramount and computational resources are not a constraint, other algorithms such as ORB and SIFT may be more suitable for real-time applications where speed and efficiency are critical factors.

Although semantic segmentation cannot be directly used to match unseen objects with the provided template image, it is observed that semantic segmentation algorithms can be adapted to segment such objects with fine-tuning, which has a relatively low cost compared to the potential accuracy gains. Additionally, this work found that the output of semantic segmentation can be used as a mask to narrow down the search space for feature matching. Combining this approach with feature detectors can lead to a promising hybrid solution that reduces computational complexity while improving the accuracy of feature matching. This technique has the potential to enhance the performance of IBVS in robotic systems, especially in situations where computational resources are limited.

The efficiency evaluation results are shown in Table 4.1. The semantic method has the highest processing time, with an average of 0.136s, while the SIFT method has the lowest, with an average of 0.0198s. The ORB and hybrid methods have processing times of 0.0335s and 0.867s, respectively. The hybrid method, which combines deep learning and feature matching algorithms, has a higher processing time than the other methods but also achieves the highest accuracy, as shown in Tables 4.1 and 4.2. Overall, the results demonstrate the trade-off between accuracy and efficiency in IBVS systems and highlight the importance of developing algorithms that can achieve both high accuracy and fast processing times.

Although the experimental results demonstrate the effectiveness of these IBVS approaches, it is important to consider the limitations and potential challenges of each method when applied to real-world scenarios. For example, machine learning-based methods may be prone to false positives or missed detection in complex scenes with high variability. Additionally, feature-based approaches may struggle with scalability when applied to large datasets. Therefore, it is important for researchers and practitioners to carefully evaluate the strengths and weaknesses of each method before applying them in real-world scenarios.

However, it is crucial to acknowledge that the practicality of the RGB-based IBVS approach can vary depending on the specific environmental conditions. In scenarios where the environment is well-structured and the height of the background is known, depth cameras have the advantage of providing more precise object recognition and detection. Consequently, in cases where there is sufficient budget or familiarity with the environment, the RGB-based approach may be less competitive.

Furthermore, PBVS utilising deep learning models is an alternative solution for low-cost robot visual servoing. PBVS relies on inferring object pose estimation from 2D information, which may not be as accurate as IBVS which directly estimates errors in the image space. However, the advantage of PBVS lies in its image-to-Cartesian space mapping, which simplifies the design of control laws in the image domain and reduces dependency on camera location. As such, PBVS shows promise as a viable option for certain applications where accuracy requirements are less stringent and ease of control law design is a priority.

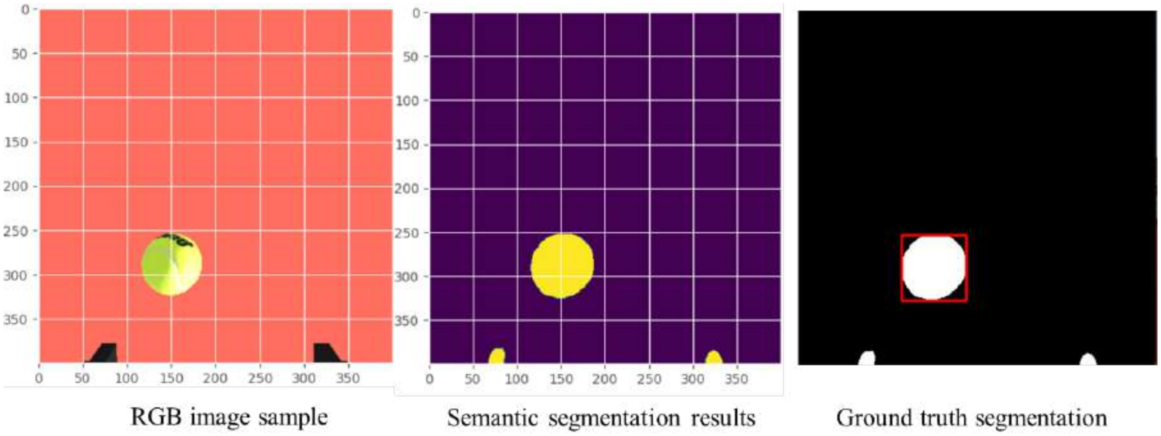


Figure 4.13. Example results of semantic segmentation, where each color represents a different object class.

4.4.2 Category-Agnostic Localisation

To evaluate the performance of the proposed method with image queries, several evaluation metrics were utilised, including recall and intersection over union (IoU). Recall indicates the ability to correctly detect and select the same object as the query image, while IoU measures the overlap between the predicted object region and the ground truth region. The IoU is calculated as:

$$IoU = \frac{|A \cap B|}{|A \cup B|}, \tag{4.9}$$

where A is the predicted region and B is the ground truth region. These metrics provide a comprehensive assessment of the method's accuracy and robustness.

The experiment analysed the segmentation results and compared them with the ground truth annotations to assess the effectiveness of the method in accurately localising and segmenting the target objects. Additionally, the method was compared with feature-based template matching methods (SIFT [231] and ORB[88]) and YOLOv3 [232] in a simulation environment, as shown in Table 4.3. For Yolov3, this work simply used the class on the query image as a hint to find the matched object. The results were lower than in other datasets, as the objects have arbitrary shapes and

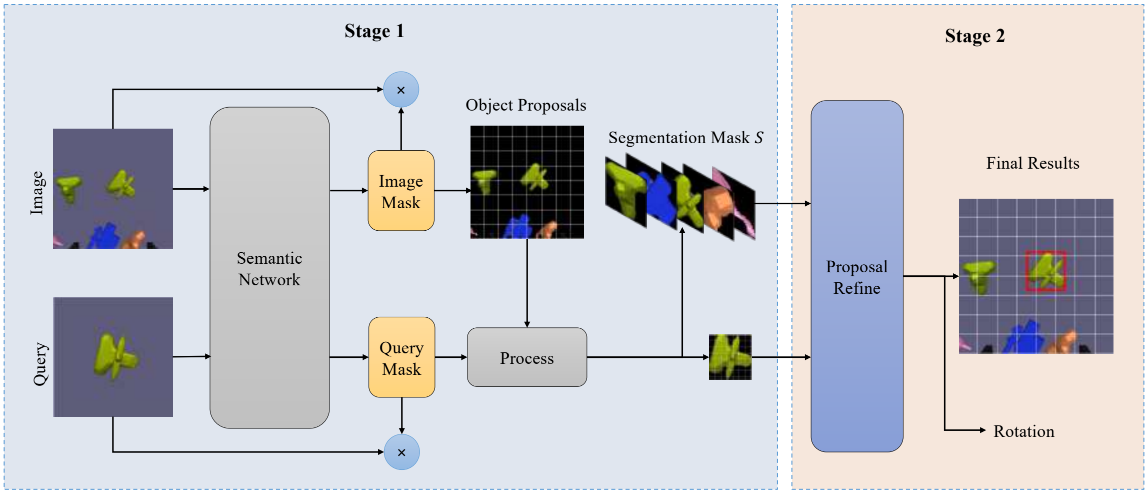


Figure 4.14. The proposed method contains two stages: At the stage 1, the input image and query are segmented by the foreground object and the background. Then the foreground is kept and used to mask the image and cropped into different proposals. The processed query and proposals are fed into the feature detection module in stage 2 to match the target object and estimate the rotation.

are harder to classify. Fig.4.11 showcases example results of query object localisation in various random scenes. The predictions for the RGB image are visually highlighted with bounding boxes. Each input image represents a unique scenario generated by randomly adding arbitrary objects to an unseen background within the field of view. This demonstrates the method's ability to handle diverse environmental conditions and accurately localise objects of interest.

Table 4.3. Comparison of average IoU and recall for each method. Porposed method achieves the highest performance.

Method	**Avg IoU**	**Avg Recall**
Proposed method	0.91	0.88
Feature-based methods	0.34	0.45
Yolov3	0.87	0.12

In addition to the simulated environment, this experiment also tested the method with real-world images. It is worth noting that the dataset used for training solely consists of synthetic images and has not seen any of the objects in the real-world experiments conducted. To further validate the robustness, the real-world experiments were carried out under different light conditions, various backgrounds, and different

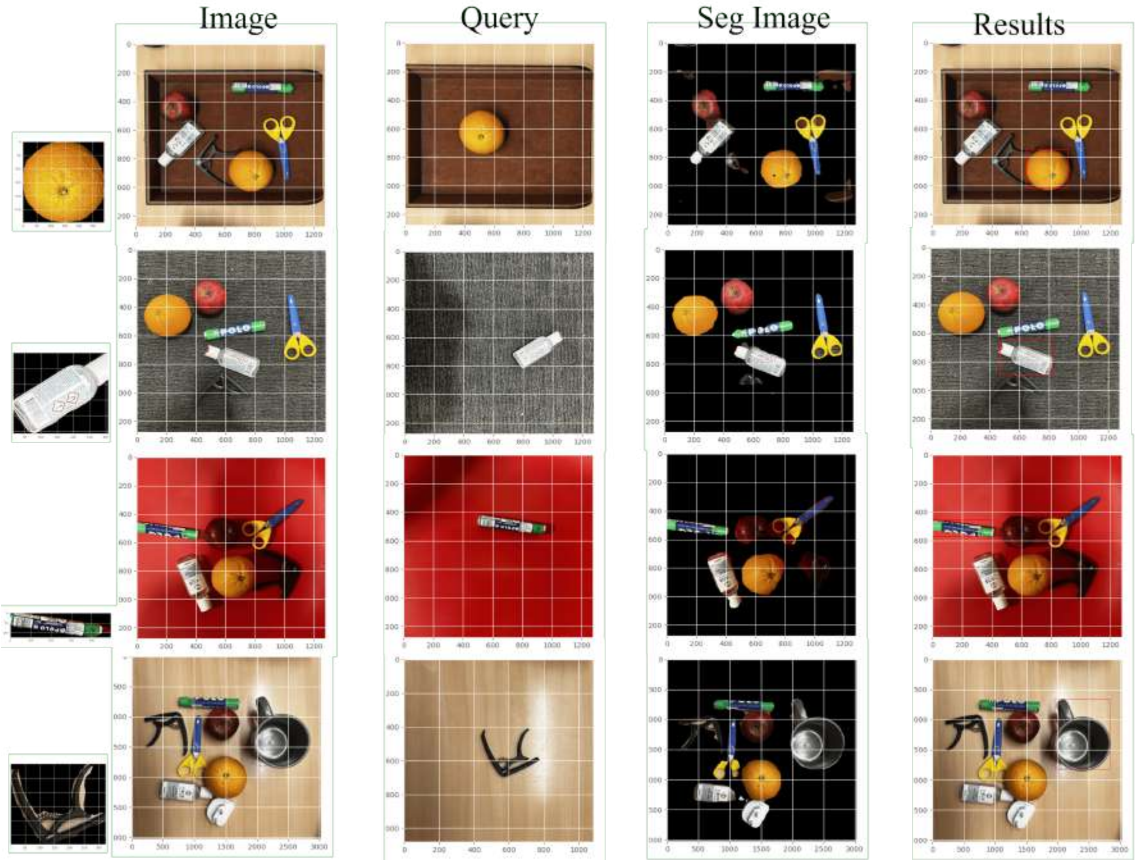


Figure 4.15. Real-world images for the validation of the tested algorithm. From left to right: cropped segmented query, images with different camera parameters, query, segmented mask, and the final prediction of the object's location.

image or query shapes (3024 by 3024 or 1276 by 1276 pixels). The results are shown in Fig. 4.15. Four example scenes are depicted, and the corresponding results are presented on the rightmost side. From left to right shows the cropped segmented query image, the input real-world image, the real-world image after masking by the generated mask, and the final results. The results suggest that the method can handle different camera parameters, diverse scenes with unseen arbitrary objects. However, it is worth noting that the segmentation might still predict multiple object segments on the same object, which can lead to localisation failures.

4.4.3 Category-Agnostic Servoing

The experiments aim to address two primary questions: (1) Can an RL agent learn to servo the robot in the image plane from deep features? and (2) How well do deep features perform compared to traditional template matching algorithms? To answer the first question, this experiment created a simulated visual servoing scene for training the control policy. The results are shown in Fig. 4.6, which depicts the simulation scene at different time steps, along with the template matching results with high and low confidence. The right-most two images show the template image and the estimated results with the target location marked in a red circle, respectively. Despite the occasional noisy and unreliable data, the learned visual servoing RL agent is capable of identifying the target object in the presence of other distractors. The RL-based agent can converge, and it exhibits stable behaviour around 25,000 episodes, as shown in the training curve (see Fig. 4.16).

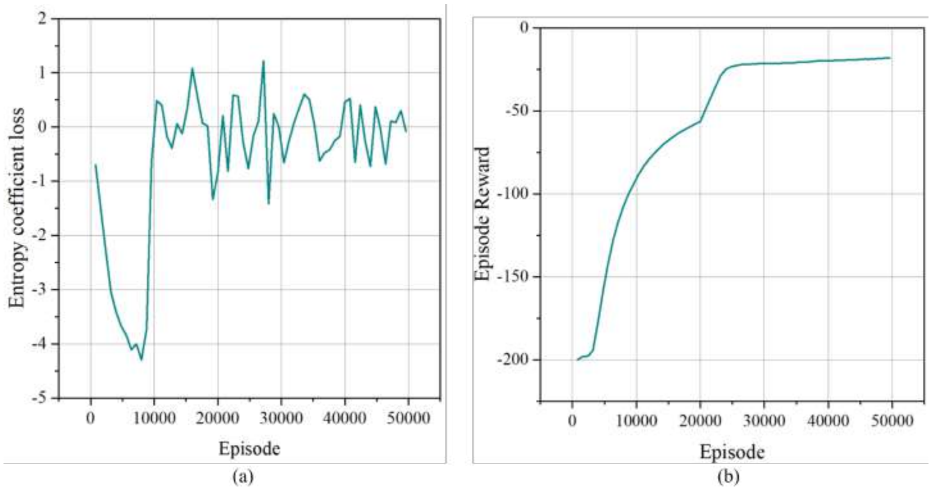


Figure 4.16. Training performance curves: (a) Accumulated reward versus training episodes for the visual servoing RL agent. (b) Entropy coefficient loss versus training episodes for the SAC algorithm.

Table 4.4. Results of comparison of the image matching.

Algorithm	Time (s)	Match rate (%)
SIFT	0.12	88.1
ORB	0.03	80.3
CNN	0.13	90.7

To answer the second question, this experiment conducted a comparison between the CNN-based template matching module and two traditional image matching techniques, namely SIFT and ORB. For this experiment, a fixed template image generated during the initialisation of the algorithm in the simulator is used, and compared with a map image taken by the robot in a random position while keeping all objects within the FoV. As the ground truth value is in the image domain, while only the world frame coordination of the object can be accessed via the simulator, a quick segmentation algorithm is designed to extract the target object from the image. Specifically, the simulator's capability of taking segmentation images is is used to create a mask of the template object and then applied a connected component labelling algorithm to find the target object in the segmented binary image. The results are presented in Table 4.4. This research did not take into account the initialisation time and training time of the CNN. The findings suggest that the CNN-based template matching method has a higher match rate and is more robust to deformation and occlusion compared to SIFT and ORB.

In order to train the model, this research tested 16 different 3D daily objects from [233] and 1000 randomly generated objects in the simulation. The simulation environment is built on the Bullet physics engine[234], with a simulated UR5e robot arm with a gripper and a wrist camera looking at the object on a table. The simulator is used for the data generation and the training of the estimator and the reinforcement learning agent. Domain randomisation (DR) is used for training and testing the method to boost the sim-to-real generalisation capability, where the scene background and table texture were selected randomly from the Describable Textures Data (DTD) set [235] and a randomly generated pure colour texture set. The camera is mounted on the end-effector of the robot, which provides RGB images ($256 \times 256 \times 3$) to the estimator. Camera-related parameters, such as light source direction, distance and intensities, the brightness of the reflection, and diffuse coefficient, are also randomised to train a more robust system. Fig. 4.8 shows example pictures of the randomised simulation environment with different task objects and textures taken from the goal position and other random positions.

In addition to the simulation environment, the method in a real-world scenario is tested at the same time. A 6-DoF UR10e robot with OnRobot force-torque sensor and an RG2 gripper mounted on the wrist is used. A Microsoft HD3000 camera is attached to the gripper to capture RGB images. Images taken from the camera have 640×480 resolution and are cropped and resized to $224 \times 224 \times 3$ to fit in the estimator's network. The maximum output of the reinforcement learning algorithm is scaled and mapped down to 0.01 m/s. For example, an output action 1 in the x-axis means a 0.01 m/s velocity motion in the x-axis. A smaller maximum velocity leads to a smoother and better motion but takes more steps to interact with the environment. The reinforcement learning agent and control are deployed on a PC connected to the robot with a cable connection. When interacting with the environment, the PC will send a command to the camera to capture images and pass them to the estimator module. The distance estimation is then passed to the reinforcement learning agent and predicts an output, which will be converted to the robot motor command.

The training profile of the estimator module is shown in Fig. 4.17. To test the RL module, the output of the estimator is replaced with an accurate position difference observation, which is assessed by the simulator, and compare the average episode reward to the method and a replacement of the RL agent of the method to the DDPG algorithm [236]. The result is demonstrated on the left of Fig. 4.9. All algorithms have been trained in the simulation environment with 50,000 steps. The algorithm with the estimator module shows a similar performance to the case where an accurate position difference has been given. To further validate the effect of the estimator and HER technique on the method, a thorough ablation study is conducted on the method and the results are shown in the middle of Fig. 4.9. To compare the learning curves of the method without the estimator, the estimator module is replaced with an end-to-end image input to the reinforcement learning algorithm. It can be observed that it is hard for the RL algorithm to figure out how to reach the target place with an end-to-end observation. The massive state dimension might impede the algorithm to dig out the relationship between motions in two figures and the output action.

Table 4.5. Performance comparison of different methods in both simulation and the real world. The numbers represent the success rate.

Method	Success (Sim)	Success (Real)
The method	0.96	0.87
The method (with accurate position)	1.00	N/A
DDPG + HER	0.93	0.40
SAC	0.70	0

In addition, the method to different RL algorithms with the same input from the estimator is compared. It can be observed that the method is able to reach a higher average episode reward with much fewer time steps. A higher average reward means fewer redundant moves performed by the robot. The right part of Fig. 4.17 depicts the critic loss of different RL algorithms in log scale, with the x-axis being the number of training steps the agent experiences, where the method and an alternative version with DDPG with HER replay buffer have a relatively small and convergent critic loss.

Table 4.5 provides a comprehensive overview of the performance of various methods, both in simulation and on a physical robot. Due to the inherent difficulty in accessing accurate relative position information, it is unable to conduct real-world experiments for the proposed method with precise position input. It is worth mentioning that the DDPG algorithm with the HER replay buffer exhibited

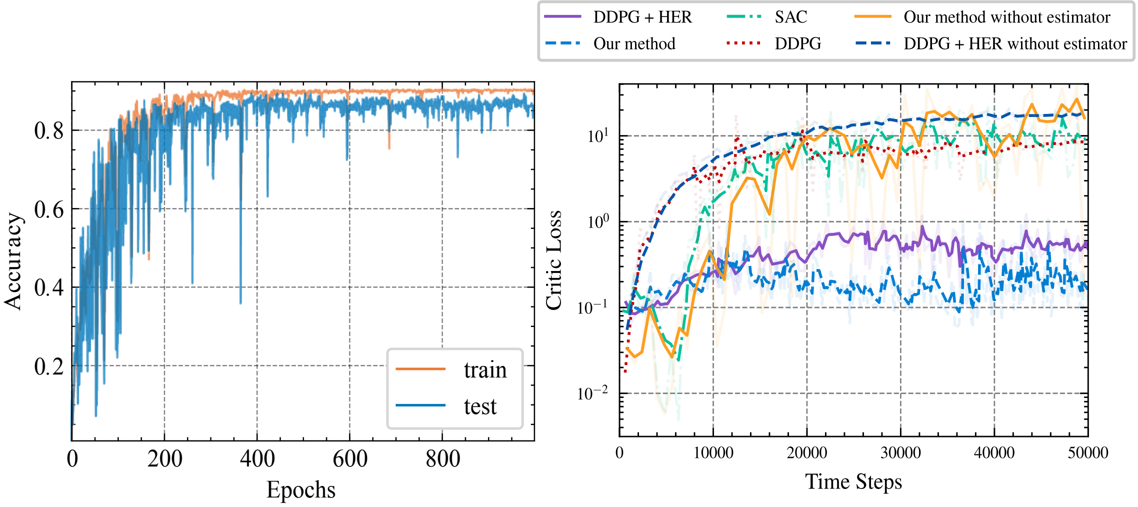


Figure 4.17. Left: the learning curve of the estimator module. Right: critic loss through iterations of training with different RL algorithms and techniques.

a noticeable decrease in performance during real-world experiments compared to simulation. Furthermore, the algorithm, initially trained on a UR5e robot arm, demonstrated transferability to a UR10e robot arm with distinct reach capabilities and maximum speeds, while maintaining consistent policy output.

4.5 Summary

In summary, this chapter presents two algorithms with the goal to develop a category-agnostic visual servoing method that leverages object images as queries to enable robots to grasp arbitrary unknown objects in unstructured environments. By combining object segmentation and transformation prediction, this research aims to overcome the limitations of fixed-class approaches and eliminate the need for extensive training datasets or manual labelling. The proposed method has the potential to enhance robotic perception and manipulation capabilities, enabling robots to perform a wide range of tasks in diverse real-world scenarios. In the first investigation of visual servoing methods for advancing robot perception and autonomy, a thorough evaluation and comparison of three distinct approaches were conducted: a feature-based approach, a hybrid approach, and a machine-learning-based approach. The empirical results unveil the machine-learning-based approach as the standout performer in terms of precision and resilience. It demonstrates the ability to detect and pinpoint objects within complex scenes, even amidst distractions and fluctuating lighting conditions. The hybrid approach displays promise but exhibits a lower tolerance to variations in lighting and object appearances. Meanwhile, the feature-based approach excels in straightforward scenarios but grapples when confronted with intricacies in more complex settings. The study underscores the supremacy of a hybrid algorithm that integrates a deep neural network into a feature detector for IBVS. This combination amplifies robustness in object detection and localisation, particularly in the presence of distractions and challenging lighting conditions.

Subsequently, based on the findings, a pioneering hybrid approach that amalgamates deep learning and feature-based methodologies is presented to tackle category-agnostic object detection and localisation using image queries, a pivotal capability for enabling robots to comprehend and engage with their surroundings. By capitalising on the strengths of both techniques, the approach achieves superior performance in the precise localisation and segmentation of objects, surpassing conventional feature-based template matching methods and the widely-adopted YoLov3. The proposed approach employs a category-agnostic semantic segmentation framework, segmenting objects based on their presence rather than specific categories. Rigorous quantitative assessments conducted on both synthetic and real-world datasets underscore the approach's exceptional accuracy and robustness across diverse scenarios, including objects with arbitrary shapes. These results collectively underscore the efficacy of the approach in bolstering object detection and localisation within the realm of visual servoing augmentation.

The investigation into computer vision methods has significantly elevated the robot's environmental sensing capabilities. Building upon this technological advancement, the research then seamlessly integrate the progress in visual servoing into a sophisticated learning-based robot control framework. The establishment and elucidation of a simulation framework for experimentation serve to provide readers with a nuanced comprehension of the platform where RL algorithms undergo testing and refinement. Additionally, the section outlines the process of simulation data generation, offering insights into the systematic collection of data crucial for training and evaluating RL algorithms within the simulated environment. Subsequently, a pioneering data-driven closed-loop robot control method is introduced that leverages both RL algorithms and insights gained from prior visual servoing exploration. This novel framework empowers the robot to operate without prior knowledge of the task object or the intrinsic camera parameters. A distinctive feature of this approach is its ability to servo the robot using only a single template image of the task object, showcasing the efficiency and adaptability of the reinforcement learning-based control strategy in dynamic and unpredictable scenarios.

The contribution of this chapter are:

- Providing a comprehensive comparison of three categories of robot arm visual servoing using only RGB information.

- Evaluating the methods on a dataset of rendered synthetic images captured by an RGB camera mounted on a robot arm in simulation.
- Analysing the strengths and weaknesses of each method and providing suggestions for future work.
- The development of a category-agnostic visual servoing approach utilising object images as queries.
- A two-stage methodology for segmenting target objects and predicting transformations between query and real-time images.
- Automatic dataset generation, negating the need for pre-existing datasets.
- Validation of the proposed algorithm across unforeseen objects in both simulated and real-world settings.
- Introducing an algorithm that showcases superior performance compared to both pure RL-controlled methods and traditional hand-designed feature extraction approaches.

Future work includes refining the segmentation process to improve the localisation accuracy and addressing the challenge of predicting multiple object segments on the same object. Additionally, exploring techniques for handling occlusions and improving the generalisation capabilities of the method to handle a wider range of real-world objects would be valuable directions to pursue.

Chapter 5

Human Intention-Aware Collaboration

5.1 Introduction

The previous chapter explored algorithms enabling robots to autonomously interact with their environment without learning the targets' category. While these systems demonstrate the viability of robots learning without human programming, they face limitations in environmental awareness and handling uncertainties, particularly regarding human factors. In addition, many manufacturing tasks, such as brazing and welding, require dynamic or complex trajectories that challenge current autonomous systems. This is to say that certain processes remain difficult to automate due to their dependence on human experience or operation, rendering automation potentially time-consuming, cost-ineffective, or technically challenging.

To address these limitations, integrating human expertise into autonomous systems has emerged as a promising approach. This paradigm explores human-robot interaction through physical collaboration and teleoperation, presenting both opportunities and challenges. This chapter examines human-robot co-activity, emphasising human intention recognition through human hand gestures. An innovative human robot co-activity framework that seamlessly integrates hand gesture and dynamic movement recognition, voice recognition, and a switchable control adaptation strategy is proposed. These modules provide a user-friendly approach that enables the robot to deliver the tools as per user need, especially when the user is working with both hands. Therefore, users can focus on their task execution without additional training in the use of HRI, while the robot interprets their commands. The proposed multimodal interaction framework is executed in the UR5e robot platform equipped with a RealSense D435i camera, and the effectiveness is assessed through a soldering circuit board task. The experiment results have demonstrated superior

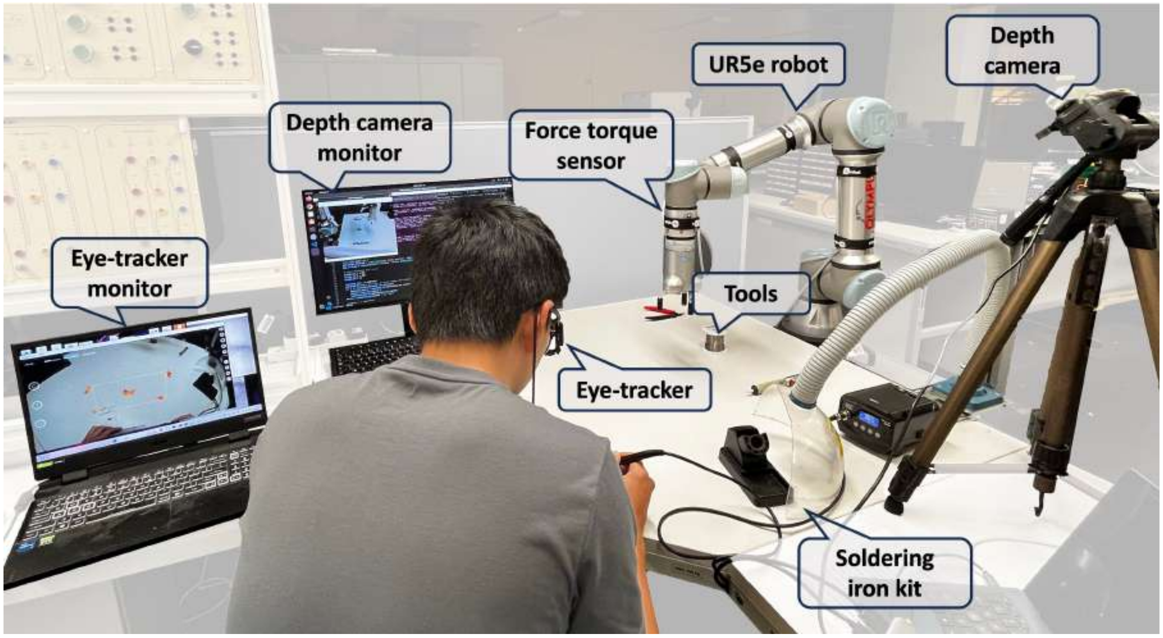


Figure 5.1. A practical validation platform designed to assess multi-modal interaction during the electrical circuit repair handover task.

performance in hand gesture recognition, where the static hand gesture recognition module achieves an accuracy of 94.3%, while the dynamic motion recognition module reaches 97.6% accuracy. Compared with human solo manipulation, the proposed approach facilitates higher efficiency tool delivery, without significantly distracting from human intents.

5.2 Method

Hand-gesture-based HRC presents a range of distinct advantages in HRC. Firstly, it capitalises on a mode of communication that comes instinctively to humans. Hand gestures constitute an integral part of everyday interactions, necessitating no specialised equipment or training, thereby reducing barriers to entry for users. Secondly, hand gestures facilitate non-verbal communication, which can be particularly advantageous in noisy or crowded environments where voice commands might prove less effective. They can also provide an additional layer of communication, allowing users to convey nuanced instructions and preferences beyond what can be expressed through words alone.

However, existing hand gesture-based HRC systems predominantly revolve around task-based processes [175, 190, 237], often overlooking the human factors. While these systems excel at deciphering specific gestures to trigger predefined actions, they frequently fall short of comprehending the broader context and the nuanced needs of the human user. Moreover, many of these systems rely on overt and exaggerated hand gestures [238, 239, 240], which can be unnatural and fatiguing for users, especially over extended periods. Such gestures may also lack the subtlety required for conveying intricate instructions or preferences effectively.

Motivated by the aforementioned challenges, this section presents a novel HRC framework that demonstrates exceptional utility when both hands of users are engaged in tasks, rendering the capacity to issue commands without concern for the robot's method of execution of paramount importance. This interaction paradigm aligns with the concept of the *supernumerary limb*, which allows users to extend their control through extra limbs or tools to manipulate the environment or interact with objects [241]. In this context, users can focus on their task execution, with the robot adeptly interpreting their intuitive gestures.

To this end, the framework leverages the visual information embedded in human hand gestures. By scrutinising both the form and motion of hand gestures, the system not only discerns the intended action but also gauges the urgency and precision required by the user. This approach empowers the robot to dynamically adapt to user needs, enabling the precise delivery of tools and objects, a feat previously challenging to achieve. Furthermore, the framework seamlessly integrate voice command capabilities into the interaction system. This enables users to instruct the robot effortlessly and naturally, eliminating the need for extensive training or adherence to predefined gestures. In this way, users can interact with the robot without additional interfaces such as screens or controllers.

- **Problem Formulation**

The primary objective revolves around the dynamic delivery of human-specified tools by the robot, along with the ability to adapt its delivery strategy in real-time based on the movements of the human hand gestures. This task surpasses the challenges posed by mere hand gesture recognition, as it demands the estimation of the 3D human hand pose and real-time adjustments to meet the user's requirements, ultimately minimising any distractions for the human user. To accomplish this, the robot relies on visual feedback, continuously recognising the presence of the human hand, estimating its 3D pose, and discerning the user's intention to alter the delivery strategy. This multifaceted process aims to maintain a seamless interaction with the human user while ensuring that the robot meets the user's specific needs.

The human-robot collaborated tool delivery framework is structured into two stages: the robot fetching and the tool delivering, each playing a crucial role in the overall process. Fig. 5.2 provides a visual representation of this framework. In the initial phase, the primary aim is to establish a seamless and intuitive mode of

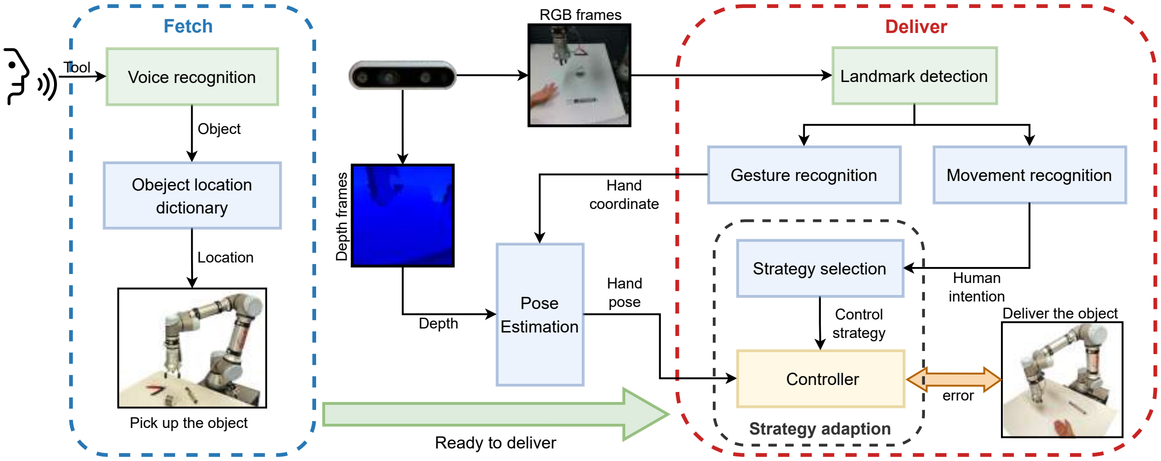


Figure 5.2. Schematic representation of the comprehensive framework for HRC in dynamic tool delivery. The framework encompasses two fundamental stages: robot fetching and tool delivery. In the robot-fetching stage, voice command recognition enables users to specify desired tools verbally. The robot employs visual feedback to recognise and track the user's hand, estimating its 3D pose and discerning the user's intention. In the tool-delivering stage, real-time hand pose estimation through a depth camera ensures precise tool delivery.

communication between the human user and the robot. To this end, this research deploys voice command recognition, enabling users to verbally specify their desired tool. Google's Speech Recognition [242] technology is used to convert spoken commands into actionable instructions for the robot. Following command processing, the robot initiates the tool retrieval process and proceeds to fetch the requested item. Subsequently, it awaits further directives from the human user. In the subsequent tool-delivering stage, the emphasis shifts to the robot's capacity to perceive and respond to the user's intentions. To begin, whether the human hand is within the robot's field of view is assessed. Upon detection, it is proceed to extract key human hand landmarks and subsequently pass this information to a hand gesture recognition network. This network determines if the human hand's gesture indicates readiness to receive an object. This estimation operates seamlessly, ensuring uninterrupted tracking and following of the human hand, thus enabling confident and precise tool delivery without the need for the human user to divert their attention from the process. Moreover, the system exhibits the flexibility to dynamically adjust its delivery strategy in real-time, responding to user-defined requirements. This adaptability enhances the overall efficiency of the user-robot interaction process, further emphasising its human-centric nature.

- **Hand Pose Estimation**

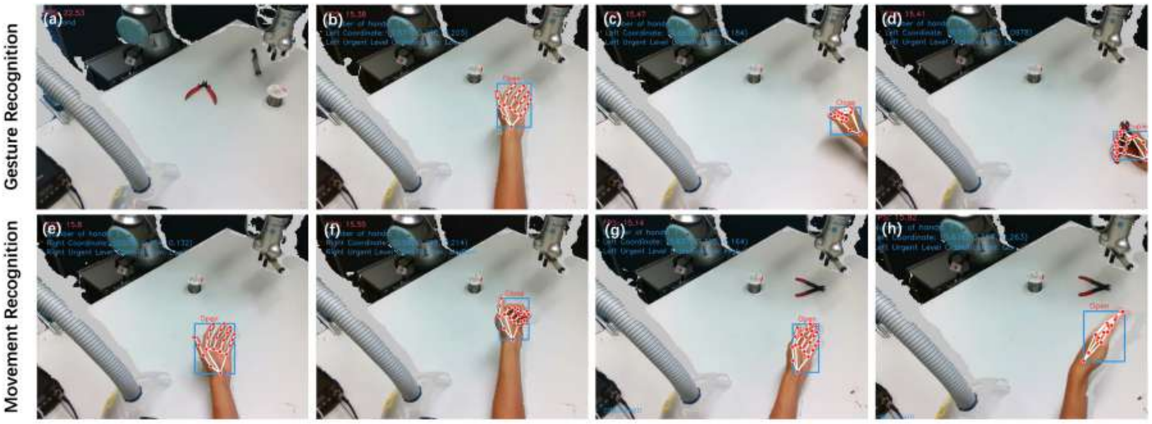


Figure 5.3. Sample results from gesture and hand movement recognition frames, illustrating various scenarios. To enhance clarity, depth and RGB images have been combined, with pixels corresponding to point cloud data beyond the defined range omitted. (a) No hand. (b) Open hand gesture. (c) Closed hand gesture. (d) Occupied hand gesture. (e) Low urgency hand movement. (f) Medium-urgency hand movement. (g) High-urgency hand movement. (h) 'Go away' hand movement.

A real-time hand pose estimation is implemented through a depth camera. This technology precisely identifies the exact position of the human hand's palm centre within the robot's base frame. Transforming pixel coordinates into the robot's base frame involves a series of crucial steps. Firstly, pixel coordinates (u and v) are deprojectd to 3D Cartesian coordinates (X, Y, and Z) in the camera frame using intrinsic parameters:

$$P_c = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = D \cdot \begin{bmatrix} \frac{u-u_c}{f_x} \\ \frac{v-v_c}{f_y} \\ 1 \end{bmatrix}, \tag{5.1}$$

where P_c represents the 3D point in the camera frame, f_x and f_y represent the camera's focal lengths, u_c and v_c represent the camera's principal points, and D is the depth value obtained from the camera. Following this pixel-to-3D point conversion, precise calibration between the camera's reference frame and the robot's end-effector frame is established. This calibration, known as "eye-to-hand calibration," accounts for any misalignment or offsets between the two frames. The transformation matrix $T_{\text{eye-to-hand}}$ is computed to convert the 3D points from the camera frame to the robot's end-effector frame. Lastly, the 3D point is translated from the robot's end-effector frame to its base frame, which is represented by $T_{\text{end-effector-to-base}}$. The overall transformation from pixel coordinates to the 3D point in the robot's base frame P_{base} is defined as:

$$P_{\text{base}} = T_{\text{end-effector-to-base}} \cdot T_{\text{eye-to-hand}} \cdot P_c. \tag{5.2}$$

- **Learn to Collaborate from Hand Gesture**

The collaborative learning process is initiated by leveraging the highly efficient Mediapipe pose detector, a tool developed by Google [243]. Renowned for its proficiency, this framework employs cutting-edge machine learning techniques to precisely identify and locate specific landmarks on the human hand. By integrating the Mediapipe pose detector into the framework, the method not only enhance interaction efficiency but also simplify the approach significantly. This streamlined approach not only improves training efficiency but also reduces computational demands. Additionally, it lessens the need for extensive training data, a common requirement when working directly with raw images.

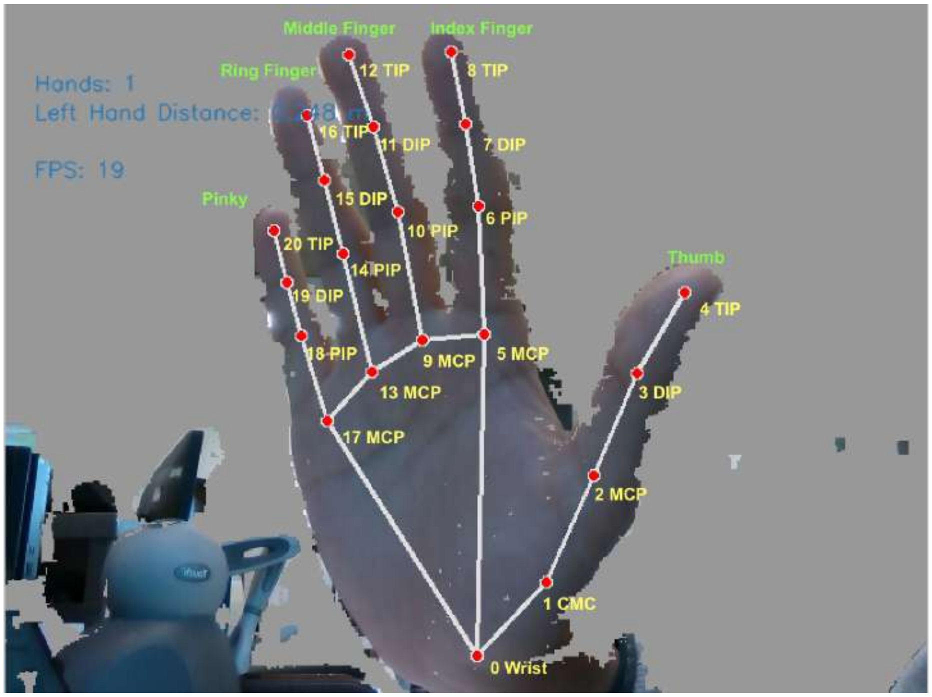


Figure 5.4. Example of hand keypoints.

One notable benefit is the reduction in input complexity. Unlike the challenges posed by using raw images as inputs, the Mediapipe pose detector simplifies the process. This streamlined approach not only improves training efficiency but also reduces computational demands. Additionally, it lessens the need for extensive training data, a common requirement when working directly with raw images. These advantages greatly contribute to the effectiveness of the collaborative learning process.

- **Gesture and Movement Recognition Network**

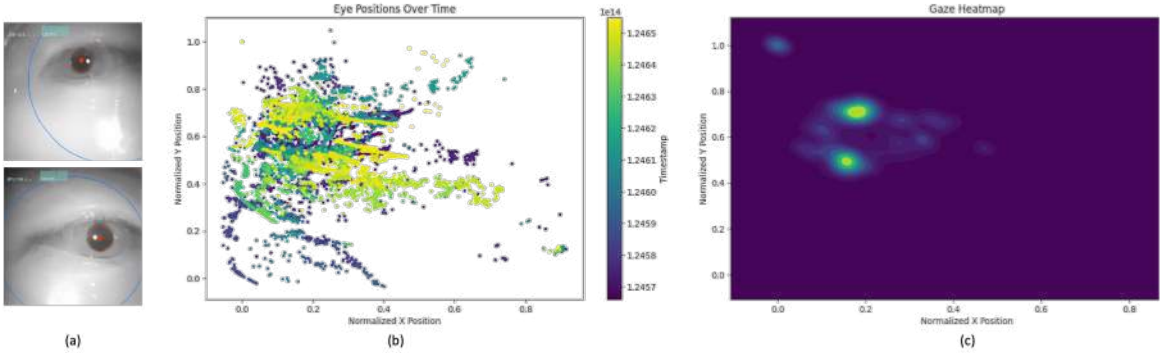


Figure 5.5. (a) Dual-camera images captured by the head-mounted PupilLabs Core eye-tracker. (b) Temporal evolution of gaze positions. (c) Heatmap representing gaze distribution based on eye tracking data.

Precise gesture recognition is paramount to facilitate seamless human-robot interaction within the HRC framework. It not only enables the identification of specific hand gestures but also provides insights into the user's status, such as hand occupation or openness for tool delivery. To accomplish this, this research employs a specialised neural network tailored explicitly for gesture recognition. This network, structured as a fully connected feedforward neural network, takes as input the 21 landmark points representing the user's hand pose. These landmarks undergo meticulous processing to yield precise classifications, enabling us to discern the specific gesture being executed. This gesture recognition network serves as a cornerstone of the framework, enhancing the interpretation of user commands and overall interaction quality.

In addition to static hand gestures, recognising human hand movements is crucial for achieving responsive human-robot collaboration, as it conveys dynamic information about the user's interaction preferences and other information. To address this need, this research introduces a neural network architecture that combines Long Short-Term Memory (LSTM) and Fully Convolutional Network (FCN) layers for movement recognition. This network operates on sequences of 30 frames, adeptly capturing temporal dependencies and spatial features within the hand movement data. The network's output provides precise movement classifications, serving as a pivotal criterion for mode switching within the HRC framework. These specialised neural networks, governing both gesture and movement recognition, empower the system to comprehensively interpret and respond to user actions, ensuring a natural and intuitive collaborative experience.

- **Dynamic Control Strategy Adaptation**

The approach involves switching between the Linear Quadratic Regulator (LQR) and the Proportional-Integral-Derivative (PID) controller based on real-time recognition of specific human hand movements using the Mediapipe framework and a trained neural network.

1. **Linear Quadratic Regulator (LQR)**: Linear Quadratic Regulator (LQR) control method is a fundamental component of the control strategy for precision target point tracking. It is particularly well-suited for applications where the system dynamics are modelled linearly, offering an effective means of optimising control effort while ensuring accurate tracking performance. The core objective of LQR is to determine an optimal control law that minimises a quadratic cost function, striking a balance between control effort and system performance. The cost function, denoted as J, is defined as follows:

   This choice aligns seamlessly with the inherent characteristics of the UR5e, which exhibits a stable linear time-invariant behaviour in the absence of external control inputs.

   $$J = \int_0^\infty \left(\mathbf{x}^T(t)\mathbf{Q}\mathbf{x}(t) + \mathbf{u}^T(t)\mathbf{R}\mathbf{u}(t)\right)\, dt, \tag{5.3}$$

   where $\mathbf{x}(t)$ represents the state vector of the robot, $\mathbf{Q}$ is a positive semidefinite weighting matrix that penalises state deviations, $\mathbf{u}(t)$ denotes the control input, and $\mathbf{R}$ is a positive definite weighting matrix that penalises control effort.

   The integral spans from 0 to ∞ in a continuous-time formulation. The system's dynamics are typically described by the linear time-invariant state-space representation:

   $$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{5.4}$$

   where $\mathbf{A}$ represents the state matrix, and $\mathbf{B}$ is the control input matrix. Crucially, the approach simplifies the system dynamics modelling, as it is recognised that the robot arm can be effectively considered as a linear time-invariant system. In this context, $\mathbf{A}$ matrix is set to all zeros and $\mathbf{B}$ as an identity matrix.

   The optimal control law $\mathbf{u}^*(t)$ can be derived by solving the Riccati differential equation:

   $$\dot{\mathbf{P}}(t) = -\mathbf{P}(t)\mathbf{A} - \mathbf{A}^T\mathbf{P}(t) + \mathbf{P}(t)\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}(t) - \mathbf{Q}, \tag{5.5}$$

   Once the state cost-to-go matrix $\mathbf{P}(t)$ is determined, the state feedback gain matrix $\mathbf{K}$ can be calculated through the solution of the continuous-time algebraic Riccati equation (CARE):

$$\mathbf{A}^T\mathbf{P} + \mathbf{PA} - (\mathbf{PB})\mathbf{R}^{-1}(\mathbf{B}^T\mathbf{P}) + \mathbf{Q} = 0. \tag{5.6}$$

Incorporating LQR into the control strategy allows us to achieve precise target point tracking while optimising control effort and ensuring stable operation, making it a fundamental element of the robotic control architecture.

2. **Proportional-Integral-Derivative (PID):** Proportional-Integral-Derivative (PID) control is another crucial pillar of the strategy for precision target point tracking with the UR5e robot arm. Renowned for its simplicity, robustness, and adaptability, PID control excels in a multitude of domains.

   PID control calculates the control input $u(t)$ based on a combination of three fundamental components:

   $$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau)\, d\tau + K_d \cdot \frac{d}{dt}e(t) \tag{5.7}$$

   where $u(t)$ is the control input, $e(t)$ represents the error between the desired setpoint $r(t)$ and the system's current state $y(t)$, and K_p, K_i, and K_d are the proportional, integral, and derivative gains, respectively.

   PID control's versatility has rendered it indispensable in numerous domains, including industrial automation, robotics, temperature regulation, and motor control, among others. Its simplicity and adaptability empower it to address a wide spectrum of control challenges, solidifying its role as a pivotal element in the control strategy for precise target point tracking.

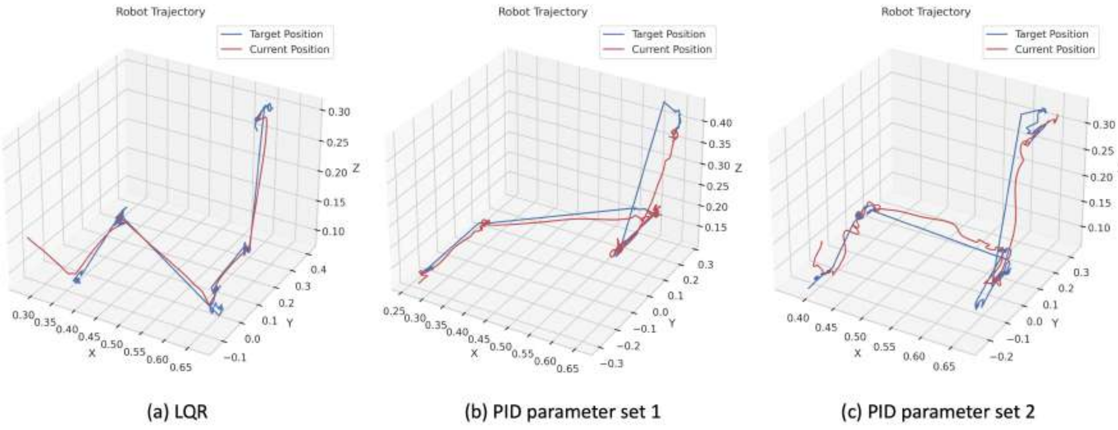
   Specifically, this research pay attention to gestures that indicate a request for a tool. When the user's hand assumes this specific gesture mode, the robot proceeds to initiate the tool delivery process.

   This structured approach to human-robot tool delivery ensures a seamless and efficient interaction between the user and the robot, with a clear division of labour between the robot fetching and delivering stages, all guided by voice commands and hand gestures to prioritise user intuitiveness and minimal distractions.

Following the identification of processes advantageous for the brazing domain, this research extends its exploration into how robots can autonomously acquire task capabilities through visual feedback through the literature review. Unlike previous automation scenarios characterised by predefined steps and processes, this study emphasises a more intricate challenge. It targets scenarios where small to medium batch production stands to gain the most, requiring robots to operate within unstructured or unfamiliar environments. In these settings, objects are not meticulously defined by engineers instructing the robot with specific movement commands. The literature review is entered on robot learning, seeking insights into the latest advancements in academia.

The incorporation of this dynamic control strategy mode switch module is vital due to the distinct strengths and weaknesses of LQR and PID control algorithms. LQR excels in delivering precise, optimal control but struggles with dynamic changes and nonlinearities, while PID offers versatility and robustness but may lack the precision of LQR. By integrating both controllers and leveraging the control strategy mode switch module, the framework ensures seamless adaptation. The robot can switch between LQR for tasks demanding precision and PID for scenarios requiring adaptability, thereby maximising performance across diverse HRC environments.

Dynamic control strategy adaptation relies on the real-time recognition of human hand movements, which serve as pivotal indicators for mode switching. The recognition process involves two key steps: utilising the Mediapipe framework for extracting landmarks from real-time visual input, thereby capturing the nuances of hand movements, and employing a trained neural network for urgency classification. This neural network classifies observed hand movements, designating certain gestures, such as the "give it to me" motion, as high urgency, necessitating a rapid transition to the more responsive PID control mode, while categorising other gestures as low urgency, allowing the system to maintain precision in LQR mode for routine tracking tasks. Fig. 5.6 showcases three distinct controllers applied to a robotic system. These controllers represent different strategies for regulating the system's trajectory, each with its own set of tuning parameters.



(a) LQR (b) PID parameter set 1 (c) PID parameter set 2

Figure 5.6. (a) LQR controller. (b) PID controller (Parameters: Kp=0.1, Ki=0.0, Kd=0.2). (c) PID controller (Parameters: Kp=0.1, Ki=0.02, Kd=0.25).

The criteria for switching between control strategies are intricately linked to the recognition of human hand movements:

High Urgency Gesture: When the neural network classifies a hand movement as high urgency (e.g., "give it to me" gesture), the system transitions from LQR to faster and more responsive PID control. This ensures rapid and accurate response to urgent requests.

Low Urgency or Medium Urgency Gesture: When low or medium urgency gestures are detected, the system operates in LQR mode with different velocities. LQR provides precision and stability for routine tracking tasks.

5.3 Experiment

This section provides an overview of the experimental setup designed to evaluate the proposed framework's performance. The primary aim of this experiment is to assess how effectively the framework facilitates seamless and intuitive interactions between humans and robots within a practical context.

The experiment was structured around a circuit repair task, carefully designed to comprehensively evaluate the capabilities of the HRC framework [1]. The experiments were approved by the Ethics Committee of Imperial College London (21IC7042). Each participant was informed about the experiment's purpose and protocol and signed a consent form before the experiment. They began by using a soldering iron to heat a designated pad on the circuit, emulating a common electronics repair scenario. Subsequently, participants instructed the robot to deliver a desoldering pump, which they skillfully use to gently desolder a malfunctioning electronic component. Following the desoldering process, participants requested the robot to provide a soldering wire, which they employed to solder a new electronic component onto the circuit. Finally, participants asked for a wire cutter from the robot to trim the excess length of the components' legs, thereby completing the circuit repair task.

To gauge the performance of the HRC framework comprehensively, a set of performance metrics was collected. This includes the performance of each recognition network and robot positional error, which is used to assess the accuracy of the robot's movements, quantifying deviations along the x, y, and z-axes between the intended target positions and the actual positions of the robot's end-effector. In addition, human gaze analysis was conducted using the PupilLabs eye tracker to examine whether participant attention is diverted or distracted by the robot during the task. Fig. 5.7 showcases the PupilLab Core eye tracker utilised in the experiment along with exemplar images captured during the analysis.

[1] The multimedia material is available at `https://sites.google.com/view/dhgfhma/home`

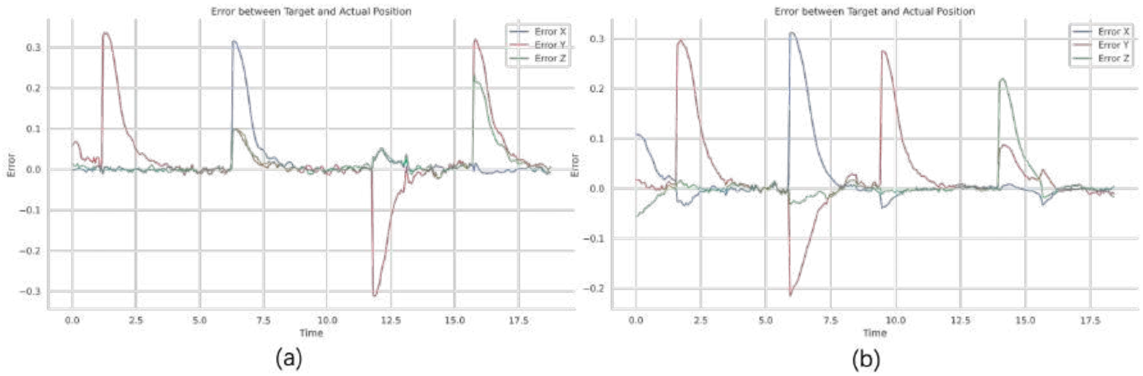


Figure 5.8. Robot positional error relative to the target position over time: (a) LQR control with the state matrix to all zeros and the control input matrix as an identity matrix. (b) PID control with proportional gains set to 0.1, integral gains set to 0, and derivative gains set to 0.2 for all x, y, and z axes.

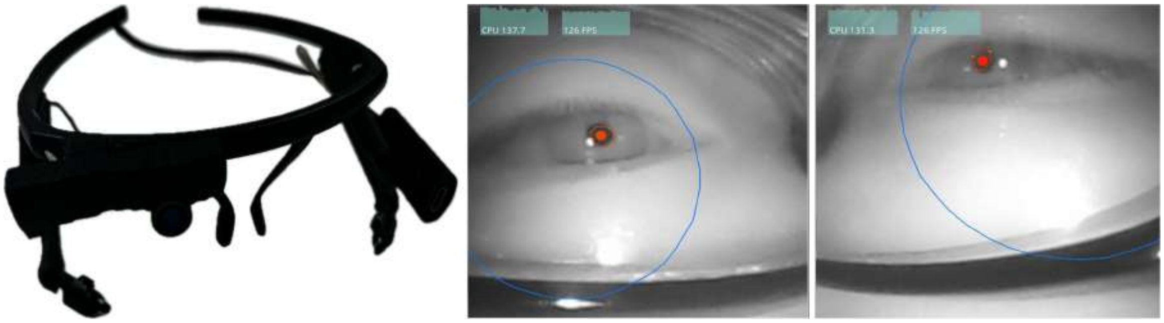
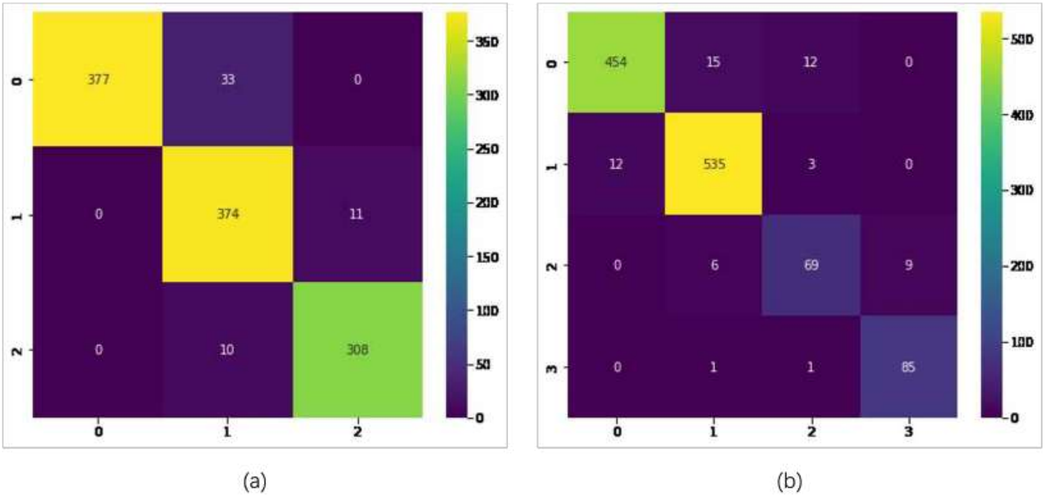


Figure 5.7. (a) PupilLab Core eye tracker. (b) Dual-camera images captured during eye tracking.

In the experimental setup, a 6 Degrees of Freedom (DoF) UR5e robot is employed for human interaction, facilitated by the RealSense D435i camera, which captured 640 × 480 RGB image frames for subsequent processing through the MediaPipe framework to collect landmark data. To ensure robust recognition, this research curated a diverse dataset of pre-defined hand gestures and movements, encompassing variations in angles, camera positions, and lighting conditions. This dataset was diligently preprocessed, involving trimming and normalisation, before training the recognition networks using PyTorch. Remarkably, the comprehensive framework operates seamlessly on an Intel i7-10510U CPU at a stable rate of 15Hz, ensuring an efficient and reliable interaction experience.

To ensure the safety of the experiment, several measures were implemented. A "virtual wall" was established as a virtual boundary, confining the robot to a



(a)

(b)

Figure 5.9. Trained recognition neural network heatmap: (a) Gesture recognition network heatmap. (b) Movement recognition network heatmap.

predefined workspace, preventing it from entering restricted or hazardous areas, and enhancing participant safety. A force / torque sensor was also mounted on the robot's end-effector, serving as an immediate stop mechanism, swiftly halting the robot's motion upon any unexpected collisions, further bolstering participant safety. Furthermore, stringent limits on joint velocity and acceleration were enforced to mitigate the risk of abrupt or erratic robot movements.

To attain the intended system response of PID control, meticulous adjustment of the parameters K_p, K_i, and K_d is required. K_p is set to 0.1, K_i to 0, and K_d to 0.2 in the experiment, which was fine-tuned to strike a balance between swift response, precision in steady-state conditions, and overall system stability. It is worth noting that the PID control module serves as an alternative means for regulating the robot behaviour, offering rapid response characteristics that are distinct from the LQR control module used in the system. The robot's response to the control strategies is illustrated in Fig. 5.8. It can be observed that although LQR is smoother and more stable at the steady state, the fine-tuned PID is faster.

5.4 Results

The outputs of the Gesture Recognition Network and Movement Recognition Network are presented in the form of heatmaps in Fig. 5.9. The Gesture Recognition Network heatmap illustrates the model proficiency in classifying hand gestures into three distinct categories: "open," "closed," and "occupied." Each category is represented by a unique heatmap, showcasing the network's ability to discern and accurately categorise these gestures based on the provided hand landmark data. The Movement Recognition Network heatmap, on the other hand, highlights the network's effectiveness in classifying hand movements into four distinct urgency categories: representing low, medium, and high urgency, and one movement "go away" that forces the robot back to the initial pose. These heatmaps provide insights into the network's capacity to interpret and classify dynamic hand movements, which is essential for real-time human intention recognition and mode switching within the HRC framework. The Gesture Recognition Network achieves an accuracy of 94.3%, while the Movement Recognition Network reaches an impressive accuracy of 97.6%.

The experiment evaluated mean pupil diameter, blink frequency, and fixation frequency to assess human workload, as established in [244]. Minor changes were observed in blink frequency, indicating that the robot intervention did not significantly distract from human intentions. However, changes in pupil diameter were observed, which increased from 2.54mm to 3.09mm, and blink frequency, which increased from 0.198 to 0.343 per second. These alterations suggest a moderate increase in human mental load due to the robot intervention, but overall, the impact on human intention and workload remained relatively low during the task.

Fig. 5.5 (a) depicts the temporal evolution of gaze positions over the course of the study. Each data point corresponds to the normalised gaze position of a participant, with the X-axis representing horizontal gaze coordinates and the Y-axis representing vertical gaze coordinates. Fig. 5.5(b) is the heatmap generated from eye tracking data. The figure utilised colormap to illustrate the participant's gaze behaviour throughout the experimental session and the density of gaze points across the screen. The heatmap provides valuable insights into the areas of interest and gaze distribution throughout the experimental task, shedding light on participant visual attention patterns. Analysis of gaze position trajectories enables a deeper understanding of how visual attention evolves in response to the robot movement and tasks.

To assess the performance of the framework, experiments under three primary conditions were conducted: with robot delivery (where the robot follows human hands and delivers objects), without robot delivery (the robot delivers objects to a fixed location upon voice command), and the framework without hand following and voice recognition (requiring users to fetch objects themselves), as shown in Table. 6.1. Notably, when hand following was incorporated, allowing the robot to adapt its movements based on the user's hand position and gestures, led to a significant reduction in the average task completion time. This result underscores the significance of hand following in streamlining interactions. Conversely, when both hand following and voice recognition were removed, placing the onus on users to fetch tools themselves, the completion time slightly increased. These findings indicate that the framework's adaptability and intuitiveness contribute to more efficient HRC. To gain a more comprehensive understanding of its performance, future experiments will explore additional metrics user satisfaction, and a comprehensive mental load analysis.

It is important to acknowledge that this study was conducted with a relatively small number of participants, and while the results are promising, further validation on a larger and more diverse sample is required to ensure the generalisability and feasibility of the system. However, it's essential to highlight that the experiment required a certain level of expertise in normal task execution. Consequently, considerable time and effort were invested in training the operators to perform the tasks effectively. This level of specialisation could pose challenges when recruiting subjects for similar experiments.

Additionally, the choice of objects used in the experiment was somewhat constrained to match the experimental setup, which may not fully represent the diversity of real-world scenarios. The primary focus of this work was to demonstrate the feasibility of the system through a demonstration-based approach. When considering tasks that require more generic applicability, such as object grasping, it becomes necessary to tailor the system to different gripper shapes and object recognition strategies, which extends beyond the scope of this study.

5.5 Summary

This chapter have identified that most collaborative approaches rely on intricate human-robot collaboration, which may lack the desired intuitiveness compared to natural limb control. Building upon this observation, an innovative HRC framework that seamlessly integrates hand gesture and dynamic movement recognition, voice recognition, and a switchable control adaptation strategy have been proposed. These modules present a user-friendly approach, empowering robots to provide tools as per user requirements, especially when users are engaged in tasks involving both hands. Consequently, users can focus on task execution without requiring additional training in human-machine interface operation, while the robot interprets their intuitive gestures.

In summary, the principal contributions of this work encompass:

- Introduction of an intuitive HRC system harnessing hand gesture-based information to discern human intents, allowing dynamic control strategy adaptation by the robot. This emphasises understanding human intents and preferences, yielding a natural, user-friendly approach without necessitating additional human-machine interface training.

- Utilisation of hand gesture pose information in conjunction with voice commands, enabling robots to adapt dynamically to user requirements and confidently deliver tools and objects, regardless of variations in hand position.

- Development of a multimodal validation platform that involves soldered circuit board tasks, offering real-time monitoring of user workload throughout the interaction.

Chapter 6

Human Interaction-Oriented Teleoperation

6.1 Introduction

Human-robot interfaces (HRI) stands at the forefront of technological advancement, offering a compelling synergy between humans and machine. The advantages of HRI are multifaceted, encompassing improved efficiency and productivity, enhanced safety in hazardous environments, accessibility for individuals with disabilities, and the potential for profound social and emotional connections with robotic companions[102, 245, 246]. It has gained significant attention in recent years due to its potential to enhance manufacturing processes by leveraging the combined strengths of humans and robots. Transitioning from automation through robot learning, the deeper integration of human involvement in these autonomous systems holds a new paradigm for incorporating human experience and knowledge. By establishing an intuitive and friendly interface for humans, human can operate robot for more complex tasks, preserving the experience of humans, while humans benefit from the efficiency brought about by the robots.

HRI has benefited users with higher efficiency towards interactive tasks. Nevertheless, most collaborative schemes rely on complicated human-machine interfaces, which might lack the requisite intuitiveness compared with natural limb control. It is also expected to understand human intent with low training data requirements. In response to these challenges, this chapter first introduces a Touch-Based Interface (TbI) design that is able to derive both direction and human forces exerted while considering human error by introducing a ball-shaped dead-zone. In addition, this research further explore TbI free teleoperation, which utilises hand gestures to enable remote interaction with hazardous environments, overcoming spatial constraints on human perception and manipulation. Most teleoperation systems rely on task-

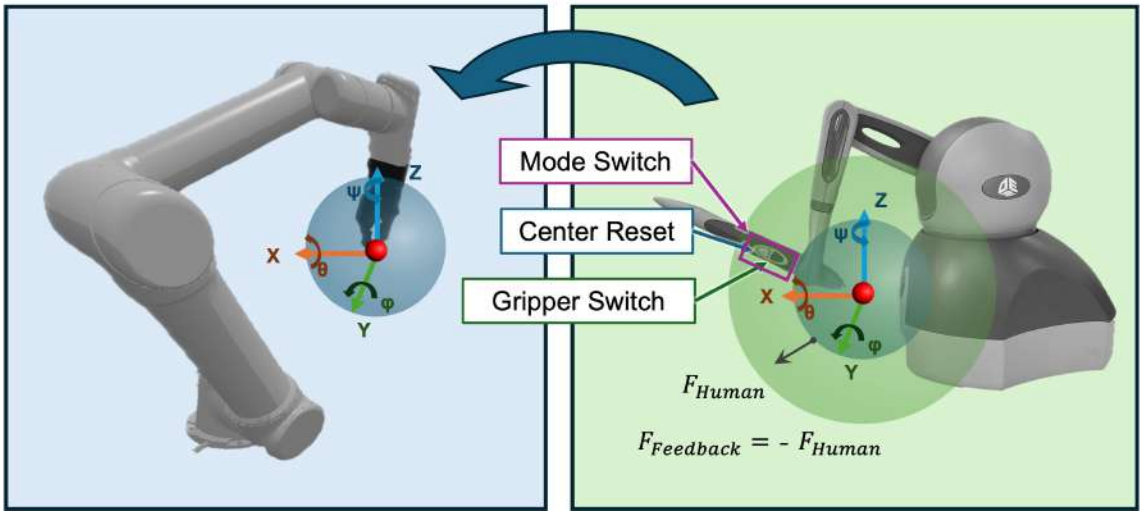


Figure 6.1. Teleoperation control scheme.

dependent interfaces to generate human instructions. This can lead to barriers in familiarising the robot's workspace and thus increase the training time for less experienced users. In order to address these problems, this research introduce a novel hand gestures based robot teleoperation method, eliminating the need for specialised controlling devices. Leveraging hand landmark detection and a neural network-based decoding algorithm, the system interprets hand movements to control robot velocity, offering a user-friendly solution to communicating with the robot. The trained model achieves a F2 score of 0.994 and outperforms algorithms in the collected dataset. Furthermore, the proposed method has been validated on a real-world Franka robot, achieving success rates of 100%, 80% and 86.7% across three manipulation tasks.

6.2 Methods

6.2.1 Physical Controller-Based Interface

To enhance the ease of use, this research have developed a haptic control mechanism, as shown in Fig. 6.1. The controller employs a velocity-based control scheme, which can be represented by the following equation:

$$V_{i,robot} = k_v \cdot d_{i,hand}, \tag{6.1}$$

where $i \in \{x, y, z\}$, $V_{i,robot}$ is the velocity of the end effector of the robot arm, $d_{i,hand}$ is the displacement of the tip on the pen of the haptic device, and k_v is the hand controller-to-robot velocity gain. The red ball in Fig. 6.1 represents the virtual origin point whose displacement is zero in each direction. The user can feel the force feedback when the pen tip moves out of the virtual blue ball, which is within the physical maximum extension of the controller, represented by the virtual green ball. The feedback force is given by:

$$F_{i,Feedback} = k_f \cdot d_{i,hand} + F_{initial}, \tag{6.2}$$

where $F_{initial}$ is the initial force that allows the user to feel a sense of boundaries. $F_{i,Feedback}$ is the feedback force on the user, which is equal in magnitude but opposite in direction to the force applied by the human on the haptic device, i.e., $F_{Feedback} = -F_{Human}$. This feedback force creates a sense of resistance when the user tries to move further. The combination of these two equations allows the user to experience greater resistance when expecting a larger robot arm moving speed.

In addition to the force feedback, the haptic device features two physical buttons on the pen. The grey button resets the current position as the virtual origin point, while the white button toggles the gripper between open and closed states. When both buttons are pressed simultaneously, the robot control mode switches between end-effector position control mode and orientation mode.

6.2.2 Hand-Gesture Based Interface

To address the challenges of a cost-effective, intuitive teleoperation approach that bridges the gap between human intent and robot action, this research propose a novel depth image information-based approach for the intuitive teleoperation of robots using human hands. This paradigm shift eliminates the need for specialised hand controllers by directly interpreting human hand gestures through a camera, reducing hardware costs and simplifying the user interface. By leveraging the inherent intuitiveness of hand gestures, the approach enables users to directly *show* the robot what to do, mimicking the ease and immediacy of real-world object manipulation. This opens up exciting possibilities for broader adoption of teleoperation technology, particularly in settings where simpler robot movements predominate.

The proposed system comprises three modules, as shown in Fig. 6.3. The raw images coming from an RGB camera are fed into the hand landmark detection module for hand landmark extraction. Then, landmark is scaled to image size and passed to the gesture to hand motion module. x-axis related gestures (Fig. 6.4 (a), (b)), y-z plane related gestures (Fig. 6.4 (c-f)) are recorded and fed into two different modules for direction and velocity decoding. X-axis related gestures are decoded based on the switching frequency between a hand open gesture and a half closed gesture. Then the calculated frequency is scaled and added to a moving window, which calculates the average value of the recent velocity to filter the velocity. Both handedness and the hand orientation are used to determine the direction of along x-axis. Similarly, y-z

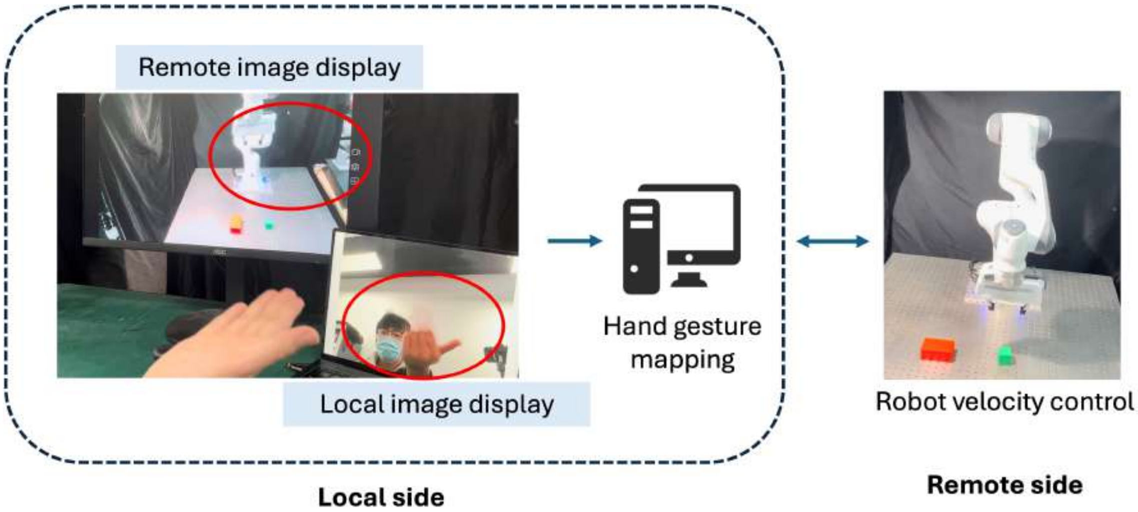


Figure 6.2. Key components of the proposed hand gesture-based velocity control teleoperation system. A monitor positioned in front of the user facilitates observation of the remote robot. An RGB camera facing the user captures hand gesture information, which is transmitted to a PC for processing, decoding, and command transmission. The remote robot, equipped with an RGB camera placed on the positive x-axis and facing towards it, receives the commands and moves accordingly.

axis related gestures decoded based on the index finger orientation. The aim of the method is to establish an intuitive way for human to control the robot. Therefore, this research would like to map the human's intention directly effect same way as the movement of the robot in the image, i.e. in a ”what you see is what you get” manner. For example, robot moves to the right side of in the image from the remote side camera when human points to the right. Different from the x-axis velocity decoding, the velocity of the y-z plane is based on the finger's back and forth speed, which ensures human to vary the velocity while giving the direction command at the same time. Similarly, a moving average window is used to get a smooth velocity.

The following notations describe the data stream procedure from data collection to the NN-based classification model training. One RGB camera is used at the local side to capture image containing the hand gesture from the operator, namely I_t^f, where t is the time and f is the number of frame in one second. This image is being resized to a smaller size for higher frame rate. Then, the image is feed in the hand landmark detection network, which is back-boned by mediapipe [243]. The output of the network will produce a set of landmarks of the hand in image frame:

$$L_t^f = \{(x_1, y_1) ..., (x_N, y_N)\}_t^f, \tag{6.3}$$

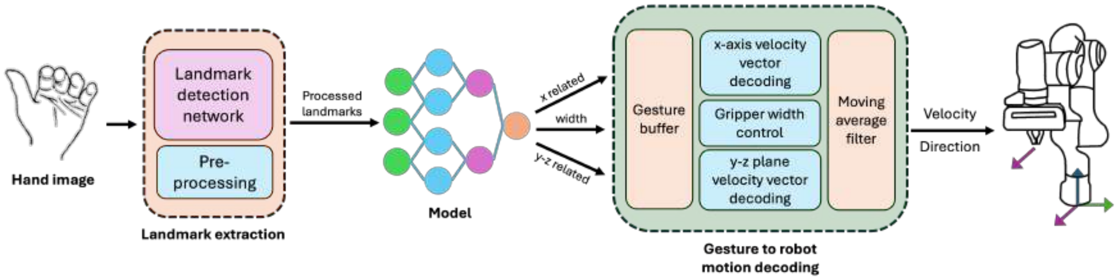


Figure 6.3. The data stream of hand gesture based robot teleoperation system.

where n is the index of the landmark, x_n, y_n are the image frame coordinate of key point n, and $N = 21$ is total number of key points on human hand. This set represented in a percentage form, which needs to be re-scaled with the shape of the image. The proposed gesture and movement algorithm detector will process L_t^f in a sliding window and map the corresponding robot moving direction in 6-axes $\{+x, -x, +y, -y, +z, -z\}$, speed in these 6-axes $\{v_{+x}, v_{-x}, v_{+y}, v_{-y}, v_{+z}, v_{-z}\}$, and the opening of the gripper d_{open}.

To guide the robot's velocity using human hand gestures, a crucial step involves establishing a mapping between the nuances of human hand motions and the corresponding robot velocity commands. This section elucidates the mechanism through which human hand gestures are translated into robot movement, aiming for an intuitively controlled trajectory in all spatial directions. To achieve this, the designed gesture set must encompass six distinct states, ensuring the versatility of the final robot movement. It is essential to note that enhancing gesture recognition accuracy necessitates careful consideration of the cumulative impact of various two-dimensional projections of gestures. Cumulativity refers to the potential loss and confusion of information resulting from projecting the length information of gestures onto a single point in one dimension. Consequently, the gesture control of robot movement is partitioned into two modes: y-z axis control and x-axis control.

For y-z plane control, a singular gesture—specifically, the pointing gesture—is employed to manipulate velocity in $\{v_{+y}, v_{-y}, v_{+z}, v_{-z}\}$. The pointing direction of the index finger determines the velocity's direction, with the intuitive mapping from the human operator's viewpoint. Here, pointing upwards corresponds to v_{+z}, downwards corresponds to v_{-z}, and pointing to the left corresponds to v_{-y}. This ensures that the remote robot displayed on the monitor moves in alignment with the human's perspective. For robot movement in x axis, this research introduce another set of hand movements labelled as ”come” and ”go.” As the remote side camera is placing in front of the robot (i.e. placing on the positive x-axis and facing at the base of the robot), the positive x-axis robot motion is represented by a waving movement. On the other hand, a ”drive away” movement represents negative x-axis motion. This two

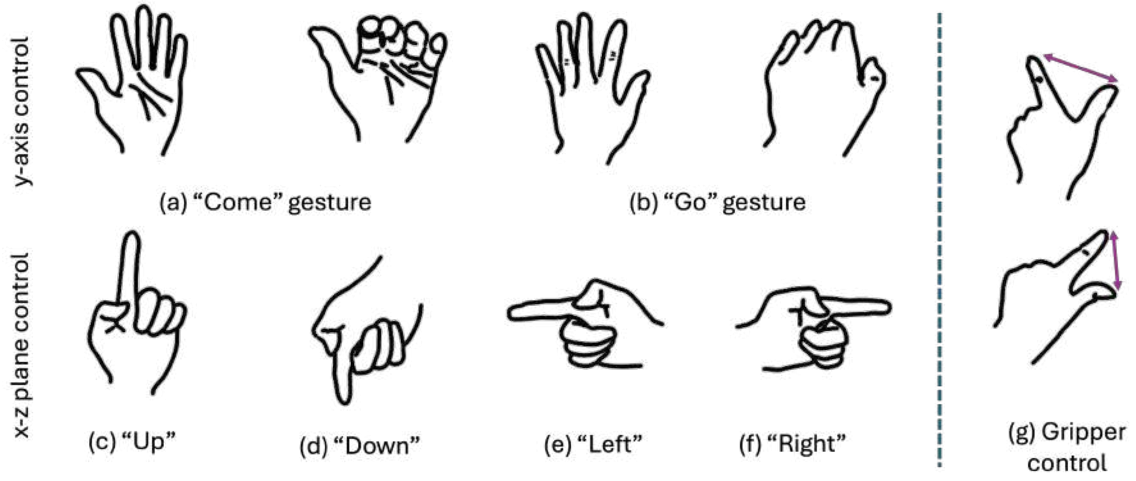


Figure 6.4. Gesture-based robot velocity control.

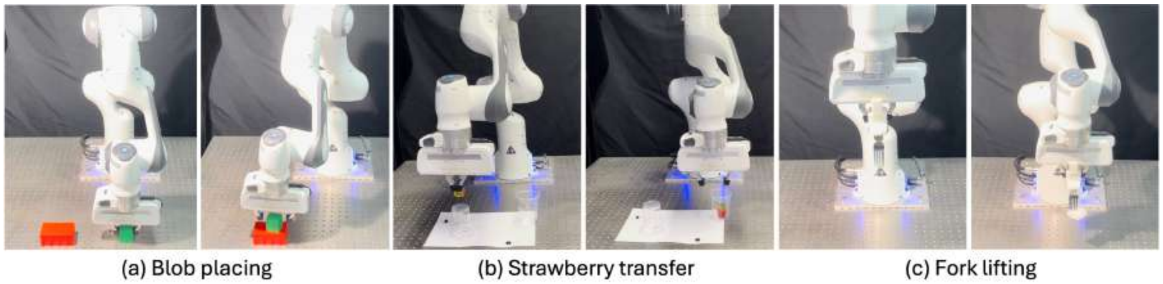


Figure 6.5. Illustration of three diverse manipulation tasks validating the effectiveness of the hand-gesture-based remote programming framework.

motions contains only two gestures and the interpretation of the motion is determined by the combination of handedness and palm facing orientation. If the palm is facing towards human, which normally means ”come” in human body language, the robot will moving towards human. Conversely, the ”drive away” movement would drive the robot away along x-axis. A standardised representation of hand gestures, depicting these movements, is illustrated in Fig. 6.4.

In addition to the velocity control mode, this research also incorporates gripper control through human gestures. The initiation of the gripper control mode is signalled by the gesture of opening the thumb and index finger. For direct gripper manipulation, this research establishes a mapping between the distance $d_{opening}$—measured between the thumb tip and index finger tip—and the gripper width. This allows users to effortlessly convey the gripper width by replicating the opening width of their own fingers. Several considerations are paramount in this context. Firstly, this research addresses the inherent physical differences among operators. Recognising that individuals possess varying finger opening widths, a direct mapping from physical opening to the gripper's width is deemed impractical. To ensure algorithmic robustness and adaptability across diverse individuals, the relative value of the index finger's length d_{index} in relation to the distance between the finger tips $d_{opening}$ is employed. When $d_{opening}$ equals d_{index}, the gripper achieves its maximum open width of 100%. Conversely, when the index finger tip contacts the thumb tip, the gripper width is set to the minimum open width:

$$d_{\text{gripper}} = \frac{d_{\text{opening}}}{d_{\text{index}}} \times 100\%. \tag{6.4}$$

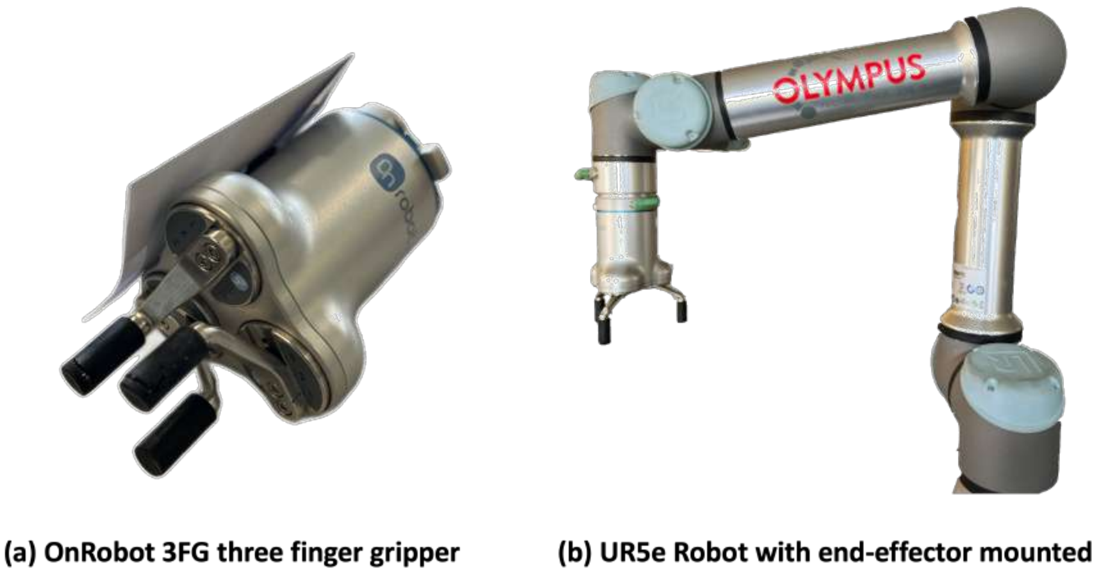
Secondly, to enhance stability and filter out potential jittering in measured values, a confirmation mechanism is introduced. Gripper movement only occurs when the hand maintains a relatively stable width. Additionally, a moving average algorithm is implemented to ensure smoother gripper control. This combined approach contributes to a more stable and reliable gripper control experience. It is worth noting that the method is designed to develop a generalised approach without additional constraints on the diversity of human operators, facilitated by a learning-based model. This ensures adaptability across various users and operational scenarios, promoting broader applicability and usability in real-world teleoperation tasks.

6.3 Experiment

6.3.1 Robot Hardware Setup

This research employs Onrobot RG2, ROBOTIQ 2F-85, and Onrobot 3FG15 grippers, each serving specific purposes. The Onrobot RG2 and ROBOTIQ 2F-85 grippers, widely embraced within the robotics community, offer robust parallel gripping capabilities. In contrast, the Onrobot 3FG15 gripper, illustrated in Fig. 6.6 (a), excels in securely grasping various cylindrical objects and irregular shapes, despite limited available documentation for control methods. A notable feature of the 3FG15 gripper is its automatic workpiece centring, ensuring rapid deployment with a stable grip, ideal for achieving precise placements. However, it's crucial to highlight the absence of an existing driver for direct PC control, and the gripper cannot be manipulated directly by the robot. Consequently, this thesis elaborates on the controlling methods developed, presenting a custom driver that empowers users to leverage Python scripts and ROS for efficient gripper control.

While the gripper's manual does suggest the availability of URScript, a script that facilitates communication between the UR robot and a PC, it is important to note that



(a) OnRobot 3FG three finger gripper

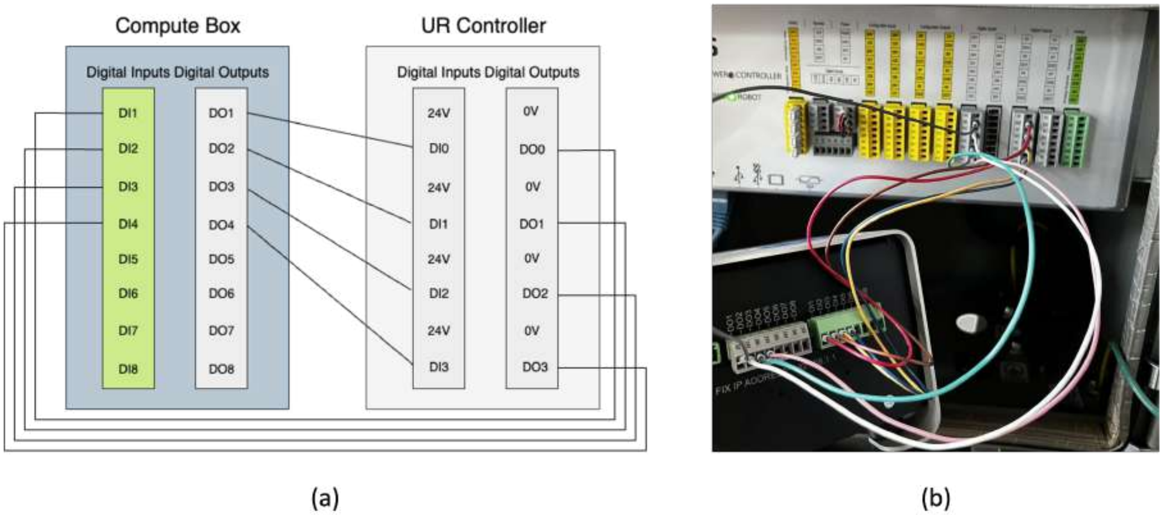
(b) UR5e Robot with end-effector mounted

Figure 6.6. Illustration of the experimental setup featuring the gripper (a) and the robot (b).

this script cannot be executed through a TCP/IP socket. This limitation has been confirmed by technical support from Universal Robots, who have clarified that the command is not an inherent function and, consequently, will not be executed in this manner. As a result, there are essentially two viable options for controlling the robot: the first involves utilising the teach pendant in conjunction with the manufacturer-provided functions, which is suitable when the objective is solely to control the robot and the gripper via the teach pendant. Although this option allows the controlling the robot and the gripper at the same time, it is not applicable when there is a need to command the robot through TCP/IP or require greater access to its functionalities. The second option is to control the gripper separately through the compute box controller, which is designed to read and configure sensors via Ethernet interface. This requires the gripper to connect directly to the compute box, then connect the compute box to the robot. However, this still does not allow the direct control to the gripper.

To simultaneously control the gripper and the robot, the hardware configuration process needs to be initiated, as depicted in Fig. 6.8. The setup entailed connecting the Tool data cable between the HEX-E/H QC (Quick Changer) and the Compute Box. The HEX-E/H QC serves as the intermediary interface linking the compute box with the gripper while furnishing 6-axis force and torque data across all six axes. Then, the connections are established using Ethernet cables: one between the robot controller and the compute box, and another between the compute box and a PC.

To configure the system, the DIP switch to position three needs to be adjusted, which activates static IP/DHCP Client mode in accordance with the datasheet guidelines. This configuration permitted access to the compute box's webclient through a web browser, using the assigned static IP address. It's essential to note that while the gripper can be controlled and monitored via the web client, direct script-based control through socket communication was not feasible. Nevertheless, the webclient offers the capability to control the gripper through WebLogic programs and utilise the readout GPIO (General-Purpose Input/Output) values as input. Then, it is able to transmit commands to the compute box through GPIO and leverage the compute box's internal program to regulate gripper operations.



(a)

(b)

Figure 6.7. (a) Schematic illustrating the circuit connection enabling remote control of the robot and gripper through the compute box. (b) GPIO connections of the compute box.

Controlling the gripper involves sending a command script to set the robot's GPIO port output, which is then connected to the Compute Box's digital input. To detect grasping, the robot's digital input needs to be connected to the Compute Box's digital output, as illustrated in Fig. 6.7. The configuration process for gripper control logic is accomplished through the web client interface, and the logic structure is depicted in Fig. 2.4.

6.3.2 Hand-Gesture Experiment Setup

Data was collected with 5 different gestures and collected 6000 of total images. The actual input of the classifier is the landmark which is irrelevant to the image.

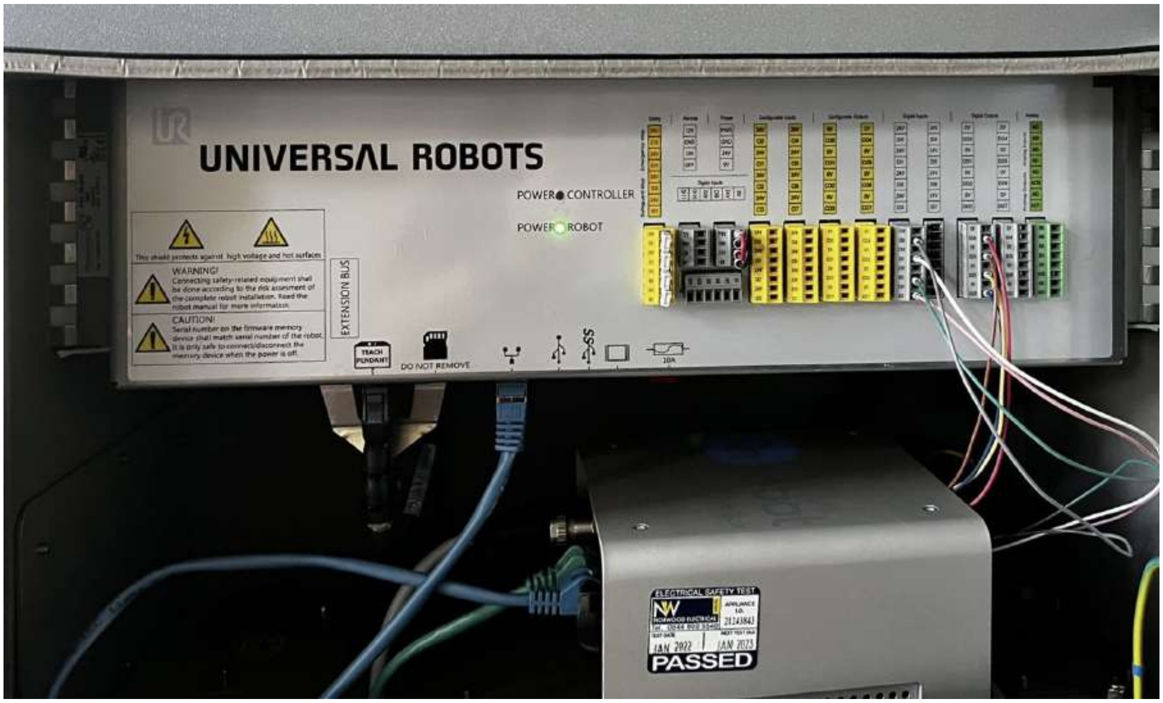
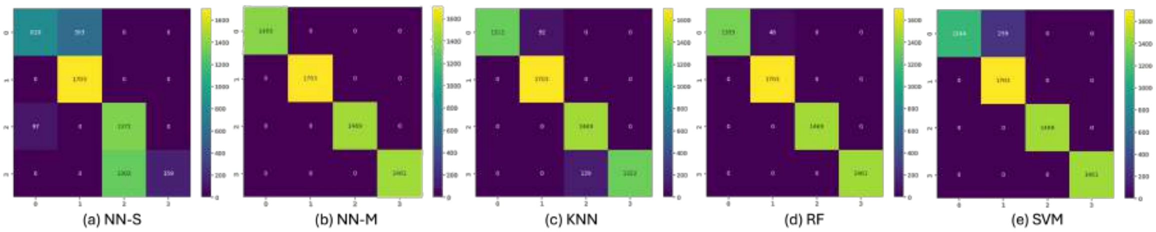


Figure 6.8. Demonstration of compute box connection to the robot.

Therefore, to make the algorithm more robust, dataset with different hand sizes and captured from different angles are collected. The hand gesture that belongs to the same category is first collected. The human hand stay the same gesture but with different angles. Meanwhile, the data collection program samples randomly until it reaches the set number of data. Then the next category is collected until the all categories data are collected.

This research conducted network exploration using 6036 sets of data across four different network structures or models. NN-S utilises a feedforward neural network architecture with two linear layers, two dropout layers, and Rectified Linear Unit (ReLU) activation functions. The input dimension is set to 21 by 2, representing the 21 key points of the hand. In NN-M, two additional hidden layers with ReLU activation functions are introduced. The optimal number of neurons and activation function in each model and layer have been determined through grid search. Additionally, K-nearest neighbour (KNN), Random forests, and Support Vector Machines (SVM), which are among the most common models for classification and have proven effective in mapping between landmarks and categories [247, 248], are tested. A grid search for the best pair of hyperparameters is conducted, given its sensitivity to hyperparameters. The grid search involved exploring parameter values, specifically C in 0.1, 1, 10, gamma in 0.1, 1, 10, kernel in linear, radial basis function



(a) NN-S (b) NN-M (c) KNN (d) RF (e) SVM

Figure 6.9. The confusion matrix of different models.

(rbf), polynomial, and epsilon in 0.1, 0.2, 0.5. The optimal hyperparameter pair is determined to be C equal to 1, gamma equal to 1, and utilising a polynomial kernel.

To validate the system, a real-world experiment with three manipulation tasks has been conducted. Experiments are conducted in an indoor environment. A 7 DoFs Franka Emika 3 robot with gripper is teleoperated by the human operator. A RealSense D435i camera is fixed and placed remotely, facing towards the robot so that it can provide the human operator with information of the working environment and the robot simultaneously. Another RGB camera is placed locally, with camera facing the human operator. Human operator is able to give commands while observing the robot's movement on the remote through a monitor. Fig. 6.2 displays the control scenario of human teleoperated the robot with hand gesture. The remote camera sent image information at a rate of 30 Hz, the image is being processed and output a human gesture command which sent from local PC via ROS node at a rate of 20 Hz to the remote robot. The local PC has an i7-12700k CPU.

6.4 Results

Table 6.1. Experimental results: average task completion time(s) under different conditions

Conditions	Avg Time (s)
w/o robot delivery	367
w robot delivery	**289**
w/o robot delivery and voice recognition	392

To determine the optimal number of training iterations for each neural network, a learning curve analysis has been conducted. This research examined the curves and identified the training iterations at which the model's performance exhibited the best results. Subsequently, the learning effectiveness of each network is evaluated at its respective optimal number of training iterations using confusion matrix.

Table 6.2. F2 scores and inference time of different model structures.

Models	**F2 Score**	**Time (ms)**
NN-S	0.948	**1.00**
NN-M	**0.994**	1.04
KNN	0.985	372
RF	0.992	3.10
SVM	0.953	64.7

Based on the confusion matrix presented in Fig. 6.9, it was observed that a considerable number of instances of Gesture 0 were erroneously identified as Gesture 1 in NN-S. The reason for the inability of the classification task accuracy to meet expectations is caused by insufficient adaptability to the complexity of the classification task. In addition to considering the accuracy of the model, for a comprehensive reflection of both recall and precision, the F2 score is computed for each model individually:

$$F_{score} = \frac{(1+\beta^2)\cdot \text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall}}. \tag{6.5}$$

In the context of robot control, the aim is to minimise the likelihood of missing control commands (false negatives) and to execute correct control commands promptly. Prioritising both safety and efficiency, this research emphasise capturing as many correct control commands as possible to mitigate the risk of critical actions being overlooked. Therefore, this research opts for the F2 score, setting β to 2. This priorities recall, ensuring the robot consistently executes necessary actions and minimising delays or errors caused by missed commands. The results of F2 scores of each model is presented in Table 6.1. This allowed us to assess the extent to which the models prioritise the important positive class.

Within the predefined set of four gestures, the inherent cumulative nature of gesture projections and the overlapping of key hand points for different gestures inevitably introduce complexity to the classification task. This complexity may lead to challenges in accurately identifying these gestures using the current depth of the neural network models. Consequently, this research considered augmenting the depth of the neural network to address this limitation. Specifically, NN-M has been expanded from the framework of NN-S by integrating additional Rectified Linear Unit (ReLU) activation functions between the hidden layers. The inclusion of activation functions between the hidden layers of a neural network introduces nonlinear mappings. Analysis of the confusion matrix indicates that increasing the depth of the neural network results in a discernible improvement in prediction accuracy. Moreover, KNN exhibits significant proficiency in accomplishing the classification task. However, it is noteworthy that the inference time of KNN is considerably longer than that of neural network-based methods, rendering it unsuitable for continuous robot control. Although the F2 score of the RF algorithm is comparable to that of the feed-forward neural network, it requires more processing time. The accuracy of SVM is relatively lower compared to KNN and RF, and its processing time does not meet the time requirements of the task.

Table 6.3. Success rate on three designed tasks.

Task	**Success Rate**
Blob placing	100%
Strawberry transfer	80%
Fork lifting	86.7%

To validate the effectiveness of the hand-gesture-based remote programming framework, this research conducted three manipulation tasks as depicted in Fig. 6.5. The first task involved grasping a blob and precisely dropping it into a designated bottle. In the second task, the objective was to grasp a fake foam strawberry with the appropriate force, transfer it, and place it inside a cup. Lastly, the third task required affixing a fork to the robot's end-effector and using it to lift a foam object off the table. The success rate of each task was measured, and the results are presented in Table 6.2. For consistency, each task was required to be completed within a 2-minute time frame, with only one trial allowed. In the blob-placing task, failure occurred if the green blob was dropped on the table or not positioned atop the red blob. For the strawberry transfer task, failure was indicated if the strawberry was not successfully grasped on the first attempt or if it was not placed inside the plastic cup. In the fork-lifting task, failure was recorded if the fork failed to lift the foam object in a single trial. Based on the experiment results, a notable limitation of the system is its remote spatial sensing capability, as users can only receive feedback from the camera, which can pose challenges when the task is self-occluded.

6.5 Summary

In conclusion, this chapter has introduced a TbI and a TbI-free system, dedicated to create ease-of-use human robot interfaces for reduced training time. The proposed interface has the potential to revolutionise the brazing process, reducing the reliance on skilled labour and improving the safety and quality of brazed joints. This research is highly relevant to the manufacturing industry, where the integration of cobots has become an increasingly popular trend in recent years. The proposed framework can serve as a starting point for future research in robot teleoperation, and could potentially be applied to other industrial processes beyond brazing.

In addition, this chapter introduced a novel approach for intuitive robot teleoperation using human hand gestures. By employing a neural network model to interpret hand movements into robot end-effector velocities, the system provides an intuitive and accessible solution, obviating the requirement for specialised controllers. Through experimentation with three manipulation tasks on an actual Franka robot, this research showcased the efficacy of the approach, attaining high success rates across tasks.

In summary, the principal contributions of this work encompass:

- A haptic control mechanism with force feedback and the ability to switch between velocity control and position control, providing users with enhanced flexibility during interaction.
- Development of a neural network-based hand landmark mapping algorithm that accurately translates human hand movements into corresponding robot commands, enabling precise control over the robot's velocity and direction simultaneously.

Chapter 7

Conclusion

7.1 Summary

This research starts with an examination of challenges within the brazing process, pinpointing the braze pasting phase as a critical area requiring enhanced flexibility and efficiency, especially in small/medium batch manufacturing. The inadequacies of existing robot frameworks in addressing this issue prompt a focus on robot vision, specifically the need for robots to navigate unseen, irregularly shaped objects. The exploration begins with the investigation of category-agnostic object detection and localisation algorithms, enabling the robot to swiftly extract pertinent information from complex backgrounds. Subsequently, the research delves into robot learning algorithms, with a specific emphasis on visual servoing. This endeavour aims to empower the robot to operate autonomously in unstructured environments, ensuring rapid deployment. Building on the foundation of improved robot perception and environmental sensing capabilities, the research incorporates human factors. This integration involves harnessing human intelligence and decision-making capabilities, culminating in the introduction of an intuitive Human-Robot Collaboration (HRC) framework. These interconnected domains contributing to the advancement of robotic intelligence and fostering seamless collaboration between humans and robots. The overarching objective is to propel robotics into practical applications within real-world manufacturing tasks, bridging the divide between theoretical advancements and their tangible implementation.

A comprehensive study of various feature extraction and matching algorithms are given for image-based visual servoing applications. The experimental results demonstrate that the hybrid approach, which combines deep neural networks with traditional feature detectors, achieves the highest accuracy and efficiency among the tested algorithms. Some of the limitations and failure cases of each algorithm have been also identified, which can guide future research in this area. While the proposed approach was developed specifically for visual servoing tasks, it has the potential to be applied to other tasks and scenarios as well. For instance, the utilisation of RGB-based robot arm visual servoing algorithms can demonstrate significance in the operations of unmanned aerial or autonomous underwater vehicles within diverse contexts. However, it is important to consider that the effectiveness of these algorithms may vary depending on the environment and specific task at hand. Variations in lighting conditions, and camera placement may impact the efficacy of object recognition and detection. It has also shown that semantic segmentation can be used to segment unseen objects with relatively low cost, and the output of the segmentation can be used as region of interests to limit the search space for feature matching, leading to a promising hybrid solution that reduces computational complexity while improving accuracy. The findings have implications for robotic systems that rely on IBVS, especially in situations where computational resources are limited. A hybrid approach is then presented that combines deep learning with feature-based methods for object detection and localisation. The method has demonstrated superior performance in accurately localising and segmenting objects, surpassing pure feature-based template matching methods and Yolov3, particularly in scenarios involving objects of arbitrary shapes. The evaluation results have not only highlighted the accuracy and robustness of the approach but also showcased its potential for enhancing human-robot interaction. By providing reliable and efficient object detection and localisation, the method empowers robots to effectively interact with unknown objects in dynamic and unpredictable environments, making it well-suited for applications in robotics, autonomous systems, and human-robot collaborations.

Building on enhanced perception capabilities, two distinct visual servoing approaches were developed. The image-based framework, utilizing deep feature extraction and reinforcement learning, achieved a 96% success rate in simulation and 87% in real-world scenarios. The position-based approach introduced an innovative estimator-policy network architecture, demonstrating superior adaptability to environmental variations while reducing the reliance on camera calibration.

In addressing human-robot collaboration, the research introduced a novel multimodal interaction framework integrating hand gesture recognition and voice commands. The framework's gesture recognition module achieved 97.6% accuracy for dynamic movements and 94.3% for static gestures, while maintaining real-time performance at 15 Hz on standard computing hardware. Analysis of human workload through eye-tracking metrics revealed only minimal increases in cognitive load (pupil diameter increase from 2.54 mm to 3.09 mm) during collaborative tasks.

The development of intuitive teleoperation interfaces culminated in two complementary systems: a haptic controller-based interface and a vision-based hand gesture system. The gesture-based system achieved F2 scores of 0.994 in controlled testing, with successful implementation demonstrated across three manipulation tasks (100% success in blob placing, 80% in object transfer, and 86.7% in precision manipulation), all while maintaining sub-millisecond latency (1.04 ms) in gesture recognition.

These technological advancements collectively establish a comprehensive framework for flexible manufacturing automation, particularly beneficial for small-batch production scenarios where traditional automation approaches prove impractical. The research not only addresses immediate challenges in brazing processes but also provides foundational methodologies applicable across various manufacturing domains requiring adaptive automation and human-robot collaboration.

7.2 Future Directions

Future research directions should focus on several key areas that could further enhance the capability and applicability of the developed framework.

- **Visual servoing:** Future research in the field of visual servoing should focus on enhancing the efficiency and robustness of these algorithms. Additionally, there is potential for exploring their applicability in other computer vision applications. Furthermore, future work will focus on further refining the segmentation process to address challenges such as predicting multiple object segments on the same object and handling occlusion. One promising direction is the integration of recent large models such as the Segment Anything Model (SAM), which can segment objects from input prompts like points or boxes, to generate masks for all objects in an image, offers exciting possibilities for robot systems. This integration should consider factors such as model size and computational resource requirements, as well as the identification of specific application scenarios. This will enable robots to benefit from more advanced vision capabilities and the continual advancement of the method holds promise for enabling more sophisticated and seamless HRIs in various domains.

- **Human-Robot Co-Activity:** The evolution of human-robot collaboration frameworks should advance toward predictive interaction models that anticipate human intentions before explicit gestures occur. This necessitates the development of context-aware systems that understand not just immediate actions but entire task sequences. Future research should explore the integration of hierarchical task planning with human behaviour modelling, enabling robots to proactively assist in complex manufacturing processes. Particular attention should be given to developing adaptive safety protocols that dynamically adjust based on task complexity and human cognitive load, moving beyond simple proximity-based safety measures to more nuanced, context-sensitive interaction paradigms.

- **Human robot collaboration and teleoperation:** A more intuitive, less cost and less sensor-dependent framework to make it more practical and more affordable to cope different scenarios. In addition to design a generalised framework, future work includes focusing on scenarios that are with more specific customised demands such as elderly care and medical scenarios. Future teleoperation systems should evolve toward reduced latency and enhanced haptic feedback while minimising hardware dependencies. Research should focus on developing novel compression algorithms specifically designed for robotic control signals, enabling high-fidelity teleoperation over standard network infrastructure. The integration of shared autonomy frameworks, where robots can intelligently assist remote operators while maintaining human oversight, presents a promising direction for enhancing operational efficiency. Additionally, the development of cross-modal feedback systems that can effectively communicate robot state through multiple sensory channels could significantly improve operator situational awareness without increasing cognitive load. Furthermore, the methods of HRC are expected to become more immersive and intuitive. Virtual reality (VR) and augmented reality (AR) technologies will be integrated into robot teaching methodologies, providing an even more natural and interactive way for humans to impart skills and knowledge to robotic systems. The synergy between AI, robotics, and human expertise will reach new heights, with a focus on user-friendly interfaces and expanded capabilities. Collaborative decision-making, where machines and humans work in tandem, each complementing the strengths of the other, will also be a promising future research focus. This collaborative intelligence will drive unprecedented levels of efficiency and innovation.

References

[1] Yong Kim, Kiyoung Park, and Sungbok Kwak. “A review of arc brazing process and its application in automotive”. In: *Int. J. Mech. Eng. Robot. Res* 5.246 (2016), pp. 246–250.

[2] Dušan P Sekulić. *Advances in brazing: science, technology and applications.* Elsevier, 2013.

[3] Devireddy Krishnaja, Muralimohan Cheepu, and D Venkateswarlu. “A review of research progress on dissimilar laser weld-brazing of automotive applications”. In: *IOP Conference Series: Materials Science and Engineering.* Vol. 330. IOP Publishing. 2018, p. 012073.

[4] Mel M Schwartz. *Brazing.* ASM international, 2003.

[5] American Welding Society C3 Committee on Brazing and Soldering. *Brazing handbook.* Miami: American Welding Society, 2011.

[6] Paul Kah et al. “Robotic arc welding sensors and programming in industrial applications”. In: *International Journal of Mechanical and Materials Engineering* 10.1 (2015), pp. 1–16.

[7] Baicun Wang et al. “Intelligent welding system technologies: State-of-the-art review and perspectives”. In: *Journal of Manufacturing Systems* 56 (2020), pp. 373–391.

[8] Andrea Bonci et al. “Human-robot perception in industrial environments: A survey”. In: *Sensors* 21.5 (2021), p. 1571.

[9] Christopher Crick et al. “Human and robot perception in large-scale learning from demonstration”. In: *Proceedings of the 6th international conference on Human-robot interaction.* 2011, pp. 339–346.

[10] Alexandros Iosifidis and Anastasios Tefas. *Deep learning for robot perception and cognition.* Academic Press, 2022.

[11] Oliver Kroemer, Scott Niekum, and George Konidaris. “A review of robot learning for manipulation: Challenges, representations, and algorithms”. In: *The Journal of Machine Learning Research* 22.1 (2021), pp. 1395–1476.

[12] Leonel Rozo, Pablo Jiménez, and Carme Torras. “A robot learning from demonstration framework to perform force-based manipulation tasks”. In: *Intelligent service robotics* 6.1 (2013), pp. 33–51.

[13] Shixiang Gu et al. “Deep reinforcement learning for robotic manipulation”. In: *arXiv preprint arXiv:1610.00633* 1 (2016), p. 1.

[14] Debasmita Mukherjee et al. “A survey of robot learning strategies for human-robot collaboration in industrial settings”. In: *Robotics and Computer-Integrated Manufacturing* 73 (2022), p. 102231.

[15] Francesco Semeraro, Alexander Griffiths, and Angelo Cangelosi. “Human–robot collaboration and machine learning: A systematic review of recent research”. In: *Robotics and Computer-Integrated Manufacturing* 79 (2023), p. 102432.

[16] Weitian Wang et al. “Facilitating human–robot collaborative tasks by teaching-learning-collaboration from human demonstrations”. In: *IEEE Transactions on Automation Science and Engineering* 16.2 (2018), pp. 640–653.

[17] Ray Y Zhong et al. “Intelligent manufacturing in the context of industry 4.0: a review”. In: *Engineering* 3.5 (2017), pp. 616–630.

[18] Youmin Rong et al. “Integrated optimization model of laser brazing by extreme learning machine and genetic algorithm”. In: *International Journal of Advanced Manufacturing Technology* 87.9-12 (2016), pp. 2943–2950. ISSN: 14333015. DOI: 10.1007/s00170-016-8649-6. URL: http://dx.doi.org/10.1007/s00170-016-8649-6.

[19] Abdelfetah Hentout et al. “Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017”. In: *Advanced Robotics* 33.15-16 (2019), pp. 764–799.

[20] Rinat Galin and Roman Meshcheryakov. “Review on human–robot interaction during collaboration in a shared workspace”. In: *International Conference on Interactive Collaborative Robotics*. Springer. 2019, pp. 63–74.

[21] Michael A. Peshkin et al. “Cobot architecture”. In: *IEEE Transactions on Robotics and Automation* 17.4 (2001), pp. 377–390. ISSN: 1042296X. DOI: 10.1109/70.954751.

[22] Ana M. Djuric, J. L. Rickli, and R. J. Urbanic. “A Framework for Collaborative Robot (CoBot) Integration in Advanced Manufacturing Systems”. In: *SAE International Journal of Materials and Manufacturing* 9.2 (2016), pp. 457–464. ISSN: 19463987. DOI: 10.4271/2016-01-0337. URL: https://www.jstor.org/stable/26267460.

[23] Yuval Cohen et al. “Deploying cobots in collaborative systems: major considerations and productivity analysis”. In: *International Journal of Production Research* (2021). ISSN: 1366588X. DOI: 10.1080/00207543.2020.1870758. URL: https://doi.org/10.1080/00207543.2020.1870758.

[24] Shirine El Zaatari et al. “Cobot programming for collaborative industrial tasks: An overview”. In: *Robotics and Autonomous Systems* 116 (2019), pp. 162–180. ISSN: 09218890. DOI: 10.1016/j.robot.2019.03.003. URL: https://doi.org/10.1016/j.robot.2019.03.003.

[25] F. Sherwani, Muhammad Mujtaba Asad, and B. S.K.K. Ibrahim. “Collaborative Robots and Industrial Revolution 4.0 (IR 4.0)”. In: *2020 International Conference on Emerging Trends in Smart Technologies, ICETST 2020* 0 (2020). DOI: 10.1109/ICETST49965.2020.9080724.

[26] Mikhail Ostanin et al. “Human-robot interaction for robotic manipulator programming in Mixed Reality”. In: *Proceedings - IEEE International Conference on Robotics and Automation* (2020), pp. 2805–2811. ISSN: 10504729. DOI: 10.1109/ICRA40945.2020.9196965.

[27] Harish Ravichandar et al. “Recent advances in robot learning from demonstration”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020), pp. 297–330.

[28] Gu Ye and Ron Alterovitz. “Guided motion planning”. In: *Robotics research*. Springer, 2017, pp. 291–307.

[29] Alex X Lee et al. “Unifying scene registration and trajectory optimization for learning from demonstrations with application to manipulation of deformable objects”. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2014, pp. 4402–4407.

[30] Rouhollah Rahmatizadeh et al. “Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration”. In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 3758–3765.

[31] Fereshteh Sadeghi et al. “Sim2real viewpoint invariant visual servoing by recurrent control”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4691–4699.

[32] Stephen James, Andrew J Davison, and Edward Johns. “Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task”. In: *Conference on Robot Learning*. PMLR. 2017, pp. 334–343.

[33] Aseem Saxena et al. “Exploring convolutional networks for end-to-end visual servoing”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 3817–3823.

[34] Quentin Bateux et al. “Visual servoing from deep neural networks”. In: *arXiv preprint arXiv:1705.08940* (2017).

[35] Ondrej Biza et al. “Action priors for large action spaces in robotics”. In: *arXiv preprint arXiv:2101.04178* (2021).

[36] Bo Xiao et al. “Optimization for Interval Type-2 Polynomial Fuzzy Systems: A Deep Reinforcement Learning Approach”. In: *IEEE Transactions on Artificial Intelligence* (2022), pp. 1–12. DOI: 10.1109/TAI.2022.3187951.

[37] Ci-Jyun Liang, Vineet R Kamat, and Carol C Menassa. “Teaching robots to perform quasi-repetitive construction tasks through human demonstration”. In: *Automation in Construction* 120 (2020), p. 103370.

[38] Maria Kyrarini et al. “Robot learning of industrial assembly task via human demonstrations”. In: *Autonomous Robots* 43 (2019), pp. 239–257.

[39] Michelle A Lee et al. “Guided uncertainty-aware policy optimization: Combining learning and model-based strategies for sample-efficient policy learning”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 7505–7512.

[40] Tim Brys et al. “Reinforcement learning from demonstration through shaping”. In: *Twenty-fourth international joint conference on artificial intelligence*. 2015.

[41] Ashvin Nair et al. “Overcoming exploration in reinforcement learning with demonstrations”. In: *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2018, pp. 6292–6299.

[42] Yang Gao et al. “Reinforcement learning from imperfect demonstrations”. In: *arXiv preprint arXiv:1802.05313* (2018).

[43] Edward Johns. “Coarse-to-fine imitation learning: Robot manipulation from a single demonstration”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 4613–4619.

[44] Eugene Valassakis et al. “Demonstrate Once, Imitate Immediately (DOME): Learning Visual Servoing for One-Shot Imitation Learning”. In: *arXiv preprint arXiv:2204.02863* (2022).

[45] Yaser Keneshloo et al. “Deep reinforcement learning for sequence-to-sequence models”. In: *IEEE transactions on neural networks and learning systems* 31.7 (2019), pp. 2469–2489.

[46] Rui Nian, Jinfeng Liu, and Biao Huang. "A review On reinforcement learning: Introduction and applications in industrial process control". In: *Computers and Chemical Engineering* 139 (2020), p. 106886. ISSN: 00981354. DOI: 10.1016/j.compchemeng.2020.106886. URL: https://doi.org/10.1016/j.compchemeng.2020.106886.

[47] Paula Fraga-Lamas et al. "A Review on Industrial Augmented Reality Systems for the Industry 4.0 Shipyard". In: *IEEE Access* 6 (2018), pp. 13358–13375. ISSN: 21693536. DOI: 10.1109/ACCESS.2018.2808326.

[48] Volodymyr Mnih et al. "Human-level control through deep reinforcement learning". In: *nature* 518.7540 (2015), pp. 529–533.

[49] Hong-Wei Ng et al. "Deep learning for emotion recognition on small datasets using transfer learning". In: *Proceedings of the 2015 ACM on international conference on multimodal interaction*. 2015, pp. 443–449.

[50] Matthew E Taylor, Halit Bener Suay, and Sonia Chernova. "Integrating reinforcement learning with human demonstrations of varying ability". In: *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*. Citeseer. 2011, pp. 617–624.

[51] Todd Hester et al. "Learning from demonstrations for real world reinforcement learning". In: (2017).

[52] Jie Chen et al. "Towards transferring skills to flexible surgical robots with programming by demonstration and reinforcement learning". In: *Proceedings of the 8th International Conference on Advanced Computational Intelligence, ICACI 2016* (2016), pp. 378–384. DOI: 10.1109/ICACI.2016.7449855.

[53] Aravind Rajeswaran et al. "Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations". In: *arXiv* (2017). ISSN: 23318422. DOI: 10.15607/rss.2018.xiv.049. arXiv: 1709.10087.

[54] "A Machine Learning Approach for Collaborative Robot Smart Manufacturing Inspection for Quality Control Systems". In: *Procedia Manufacturing* 51 (2020), pp. 11–18. ISSN: 23519789. URL: https://www.sciencedirect.com/science/article/pii/S2351978920318588.

[55] Alexander Schmidt, Florian Schellroth, and Oliver Riedel. "Control architecture for embedding reinforcement learning frameworks on industrial control hardware". In: *ACM International Conference Proceeding Series* (2020), pp. 1–5. DOI: 10.1145/3378184.3378198.

[56] Fahad Alaieri and André Vellino. "Ethical Decision Making in Robots: Autonomy, Trust and Responsibility: Autonomy Trust and Responsibility". In: *Social Robotics: 8th International Conference, ICSR 2016, Kansas City, MO, USA, November 1-3, 2016 Proceedings 8*. Springer. 2016, pp. 159–168.

[57] Tengteng Zhang and Hongwei Mo. "Reinforcement learning for robot research: A comprehensive review and open issues". In: *International Journal of Advanced Robotic Systems* 18.3 (2021), p. 17298814211007305.

[58] Matteo Leonetti, Luca Iocchi, and Peter Stone. "A synthesis of automated planning and reinforcement learning for efficient, robust decision-making". In: *Artificial Intelligence* 241 (2016), pp. 103–130.

[59] Michael Everett, Yu Fan Chen, and Jonathan P How. "Motion planning among dynamic, decision-making agents with deep reinforcement learning". In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 3052–3059.

[60] Mai Xu et al. "Predicting head movement in panoramic video: A deep reinforcement learning approach". In: *IEEE transactions on pattern analysis and machine intelligence* 41.11 (2018), pp. 2693–2708.

[61] Artemij Amiranashvili et al. "Motion perception in reinforcement learning with dynamic objects". In: *Conference on Robot Learning*. PMLR. 2018, pp. 156–168.

[62] Sean Chen et al. "ASHA: Assistive teleoperation via human-in-the-loop reinforcement learning". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 7505–7512.

[63] Wenshuai Zhao, Jorge Peña Queralta, and Tomi Westerlund. "Sim-to-real transfer in deep reinforcement learning for robotics: a survey". In: *2020 IEEE symposium series on computational intelligence (SSCI)*. IEEE. 2020, pp. 737–744.

[64] Erica Salvato et al. "Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning". In: *IEEE Access* 9 (2021), pp. 153171–153187.

[65] Jinna Li et al. "Nonzero-Sum Game Reinforcement Learning for Performance Optimization in Large-Scale Industrial Processes". In: *IEEE Transactions on Cybernetics* 50.9 (2020), pp. 4132–4145. ISSN: 21682275. DOI: 10.1109/TCYB.2019.2950262.

[66] Gabriella Rossi and Paul Nicholas. "Haptic Learning Towards Neural-Network-based adaptive Cobot Path-Planning for unstructured spaces". In: (2020), pp. 201–210. DOI: 10.5151/proceedings-ecaadesigradi2019_280.

[67] Lingwei Zhu, Yunduan Cui, and Takamitsu Matsubara. “Dynamic Actor-Advisor Programming for Scalable Safe Reinforcement Learning”. In: *Proceedings - IEEE International Conference on Robotics and Automation* (2020), pp. 10681–10687. ISSN: 10504729. DOI: `10.1109/ICRA40945.2020.9197200`.

[68] Yi Jiang et al. “Data-Driven Flotation Industrial Process Operational Optimal Control Based on Reinforcement Learning”. In: *IEEE Transactions on Industrial Informatics* 14.5 (2018), pp. 1974–1989. ISSN: 15513203. DOI: `10.1109/TII.2017.2761852`.

[69] Nourma Khader and Sang Won Yoon. “Online control of stencil printing parameters using reinforcement learning approach”. In: *Procedia Manufacturing* 17 (2018), pp. 94–101. ISSN: 23519789. DOI: `10.1016/j.promfg.2018.10.018`. URL: `https://doi.org/10.1016/j.promfg.2018.10.018`.

[70] Erwin Coumans and Yunfei Bai. *PyBullet, a Python module for physics simulation for games, robotics and machine learning.* `http://pybullet.org`. 2016.

[71] G. Flandin, F. Chaumette, and E. Marchand. “Eye-in-hand/eye-to-hand cooperation for visual servoing”. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065).* Vol. 3. 2000, 2741–2746 vol.3. DOI: `10.1109/ROBOT.2000.846442`.

[72] Josh Tobin et al. “Domain randomization for transferring deep neural networks from simulation to the real world”. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2017, pp. 23–30.

[73] Mircea Cimpoi et al. “Describing textures in the wild”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2014, pp. 3606–3613.

[74] Wenzhen Yuan, Siyuan Dong, and Edward H Adelson. “Gelsight: High-resolution robot tactile sensors for estimating geometry and force”. In: *Sensors* 17.12 (2017), p. 2762.

[75] Andrea Cirillo et al. “A distributed tactile sensor for intuitive human-robot interfacing”. In: *Journal of Sensors* 2017 (2017).

[76] Qing Shi et al. “Design and implementation of an omnidirectional vision system for robot perception”. In: *Mechatronics* 41 (2017), pp. 58–66.

[77] Hong Qiao, Jiahao Chen, and Xiao Huang. “A Survey of Brain-Inspired Intelligent Robots: Integration of Vision, Decision, Motion Control, and Musculoskeletal Systems”. In: *IEEE Transactions on Cybernetics* 52.10 (2022), pp. 11267–11280.

[78] Runqing Miao, Qingxuan Jia, and Fuchun Sun. “Long-term robot manipulation task planning with scene graph and semantic knowledge”. In: *Robotic Intelligence and Automation* 43.1 (2023), pp. 12–22.

[79] Shifeng Lin and Ning Wang. “Cloud robotic grasping of Gaussian mixture model based on point cloud projection under occlusion”. In: *Assembly Automation* 41.3 (2021), pp. 312–323.

[80] Guoyang Wan et al. “A novel robotic 6DOF pose measurement strategy for large-size casts based on stereo vision”. In: *Assembly Automation* 42.4 (2022), pp. 458–473.

[81] Chao Zeng et al. “Robot learning human stiffness regulation for hybrid manufacture”. In: *Assembly Automation* 38.5 (2018), pp. 539–547.

[82] Yu Qiu et al. “Concurrent-learning-based visual servo tracking and scene identification of mobile robots”. In: *Assembly Automation* (2019).

[83] Richard Bloss. “Automation meets logistics at the Promat Show and demonstrates faster packing and order filling”. In: *Assembly Automation* 31.4 (2011), pp. 315–318.

[84] Weibang Bai et al. “Dual-arm Coordinated Manipulation for Object Twisting with Human Intelligence”. In: *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. 2021, pp. 902–908.

[85] David G Lowe. “Distinctive image features from scale-invariant keypoints”. In: *International Journal of Computer Vision* 60 (2004), pp. 91–110.

[86] Duy-Nguyen Ta et al. “Surftrac: Efficient tracking and continuous object recognition using local feature descriptors”. In: *2009 IEEE conference on computer vision and pattern recognition*. IEEE. 2009, pp. 2937–2944.

[87] Deepak Geetha Viswanathan. “Features from accelerated segment test (fast)”. In: *Proceedings of the 10th workshop on image analysis for multimedia interactive services, London, UK*. 2009, pp. 6–8.

[88] Ethan Rublee et al. “ORB: An efficient alternative to SIFT or SURF”. In: *2011 International conference on computer vision*. Ieee. 2011, pp. 2564–2571.

[89] Reagan L Galvez et al. “Object detection using convolutional neural networks”. In: *TENCON 2018-2018 IEEE Region 10 Conference*. IEEE. 2018, pp. 2023–2027.

[90] Kai Kang et al. “Object detection from video tubelets with convolutional neural networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 817–825.

[91] Ming Liang and Xiaolin Hu. "Recurrent convolutional neural network for object recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 3367–3375.

[92] Courtney J Spoerer, Patrick McClure, and Nikolaus Kriegeskorte. "Recurrent convolutional neural networks: a better model of biological object recognition". In: *Frontiers in psychology* 8 (2017), p. 1551.

[93] Naiyan Wang and Dit-Yan Yeung. "Learning a deep compact image representation for visual tracking". In: *Advances in neural information processing systems* 26 (2013).

[94] Guanghan Ning et al. "Spatially supervised recurrent convolutional neural networks for visual object tracking". In: *2017 IEEE international symposium on circuits and systems (ISCAS).* IEEE. 2017, pp. 1–4.

[95] Bo Gao and Michael W Spratling. "Robust template matching via hierarchical convolutional features from a shape biased CNN". In: *The International Conference on Image, Vision and Intelligent Systems (ICIVIS 2021).* Springer. 2022, pp. 333–344.

[96] Jordi Pages et al. "An approach to visual servoing based on coded light". In: *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE. 2006, pp. 4118–4123.

[97] Alessandro De Luca, Giuseppe Oriolo, and Paolo Robuffo Giordano. "Feature depth observation for image-based visual servoing: Theory and experiments". In: *The International Journal of Robotics Research* 27.10 (2008), pp. 1093–1116.

[98] Tao Xue et al. "A New Delayless Adaptive Oscillator for Gait Assistance". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* 2020, pp. 3459–3464. DOI: `10.1109/IROS45743.2020.9341375`.

[99] Yanan Li et al. "A review on interaction control for contact robots through intent detection". In: *Progress in Biomedical Engineering* 4.3 (2022), p. 032004.

[100] Zhang Chen et al. "Virtual-joint based motion similarity criteria for human–robot kinematics mapping". In: *Robotics and Autonomous Systems* 125 (2020), p. 103412. ISSN: 0921-8890. DOI: `https://doi.org/10.1016/j.robot.2019.103412`.

[101] Darong Huang et al. "Cooperative Manipulation of Deformable Objects by Single-Leader–Dual-Follower Teleoperation". In: *IEEE Transactions on Industrial Electronics* 69.12 (2022), pp. 13162–13170. DOI: `10.1109/TIE.2021.3139228`.

[102] Sarah Chams Bacha et al. “Deep Reinforcement Learning-Based Control Framework for Multilateral Telesurgery”. In: *IEEE Transactions on Medical Robotics and Bionics* 4.2 (2022), pp. 352–355. DOI: 10.1109/TMRB.2022.3170786.

[103] Ziwei Wang et al. “Finite-time output-feedback control for teleoperation systems subject to mismatched term and state constraints”. In: *Journal of the Franklin Institute* 357.16 (2020), pp. 11421–11447.

[104] Darong Huang et al. “Motion Regulation Solutions to Holding & Moving an Object for Single-Leader-Dual-Follower Teleoperation”. In: *IEEE Transactions on Industrial Informatics* (2023), pp. 1–12. DOI: 10.1109/TII.2022.3229149.

[105] Ziwei Wang et al. “Learning to Assist Bimanual Teleoperation using Interval Type-2 Polynomial Fuzzy Inference”. In: *IEEE Transactions on Cognitive and Developmental Systems* (2023), pp. 1–1. DOI: 10.1109/TCDS.2023.3272730.

[106] Zebin Huang et al. “A novel training and collaboration integrated framework for human–agent teleoperation”. In: *Sensors* 21.24 (2021), p. 8341.

[107] Yanan Li et al. “A review on interaction control for contact robots through intent detection”. In: *Progress in Biomedical Engineering* 4.3 (2022), p. 032004.

[108] Sarah Chams Bacha et al. “Deep Reinforcement Learning-Based Control Framework for Multilateral Telesurgery”. In: *IEEE Transactions on Medical Robotics and Bionics* 4.2 (2022), pp. 352–355. DOI: 10.1109/TMRB.2022.3170786.

[109] Weibang Bai et al. “Anthropomorphic Dual-Arm Coordinated Control for a Single-Port Surgical Robot Based on Dual-Step Optimization”. In: *IEEE Transactions on Medical Robotics and Bionics* 4.1 (2022), pp. 72–84. DOI: 10.1109/TMRB.2022.3145673.

[110] Zhaoyang Jacopo Hu et al. “Towards Human-Robot Collaborative Surgery: Trajectory and Strategy Learning in Bimanual Peg Transfer”. In: *IEEE Robotics and Automation Letters* 8.8 (2023), pp. 4553–4560. DOI: 10.1109/LRA.2023.3285478.

[111] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2016, pp. 779–788.

[112] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2015, pp. 3431–3440.

[113] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. “Learning deconvolution network for semantic segmentation”. In: *Proceedings of the IEEE international conference on computer vision.* 2015, pp. 1520–1528.

[114] Andres Milioto, Philipp Lottes, and Cyrill Stachniss. “Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs”. In: *2018 IEEE international conference on robotics and automation (ICRA).* IEEE. 2018, pp. 2229–2235.

[115] Mulham Fawakherji et al. “Crop and weeds classification for precision agriculture using context-independent pixel-wise segmentation”. In: *2019 Third IEEE International Conference on Robotic Computing (IRC).* IEEE. 2019, pp. 146–152.

[116] Di Feng et al. “Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges”. In: *IEEE Transactions on Intelligent Transportation Systems* 22.3 (2020), pp. 1341–1360.

[117] Mennatullah Siam et al. “Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges”. In: *2017 IEEE 20th international conference on intelligent transportation systems (ITSC).* IEEE. 2017, pp. 1–8.

[118] Wonsuk Kim and Junhee Seok. “Indoor semantic segmentation for robot navigating on mobile”. In: *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN).* IEEE. 2018, pp. 22–25.

[119] Daniel Seichter et al. “Efficient rgb-d semantic segmentation for indoor scene analysis”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA).* IEEE. 2021, pp. 13525–13531.

[120] Bowen Pan et al. “Cross-view semantic segmentation for sensing surroundings”. In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 4867–4873.

[121] Stefan Ainetter and Friedrich Fraundorfer. “End-to-end trainable deep neural network for robotic grasp detection and semantic segmentation from rgb”. In: *2021 IEEE International Conference on Robotics and Automation (ICRA).* IEEE. 2021, pp. 13452–13458.

[122] Alwaseela Abdalla et al. “Fine-tuning convolutional neural network with transfer learning for semantic segmentation of ground-level oilseed rape images in a field with high weed pressure”. In: *Computers and electronics in agriculture* 167 (2019), p. 105091.

[123] Yi Zhu et al. “Improving semantic segmentation via efficient self-training”. In: *IEEE transactions on pattern analysis and machine intelligence* (2021).

[124] Pedro O Pinheiro et al. “Learning to refine object segments”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, pp. 75–91.

[125] Aljoša Ošep et al. “Track, then decide: Category-agnostic vision-based multi-object tracking”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 3494–3501.

[126] Michael Danielczuk et al. “Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 7283–7290.

[127] Chi Zhang et al. “Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5217–5226.

[128] Fereshteh Sadeghi et al. “Sim2real view invariant visual servoing by recurrent control”. In: *arXiv preprint arXiv:1712.07642* (2017).

[129] Giacomo Palmieri et al. “A comparison between position-based and image-based dynamic visual servoings in the control of a translating parallel manipulator”. In: *Journal of Robotics* 2012 (2012).

[130] Gangqi Dong and ZH Zhu. “Position-based visual servo control of autonomous robotic manipulators”. In: *Acta Astronautica* 115 (2015), pp. 291–302.

[131] Dongliang Zheng et al. “Image-based visual servoing of a quadrotor using virtual camera approach”. In: *IEEE/ASME Transactions on Mechatronics* 22.2 (2016), pp. 972–982.

[132] Min Xia et al. “Intelligent process monitoring of laser-induced graphene production with deep transfer learning”. In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022), pp. 1–9.

[133] Tao Xue et al. “Fixed-time constrained acceleration reconstruction scheme for robotic exoskeleton via neural networks”. In: *Frontiers of Information Technology & Electronic Engineering* 21.5 (2020), pp. 705–722.

[134] Sarah Chams Bacha et al. “Deep Reinforcement Learning-Based Control Framework for Multilateral Telesurgery”. In: *IEEE Transactions on Medical Robotics and Bionics* 4.2 (2022), pp. 352–355.

[135] Ziling Wu, Xiaofeng Wu, and Yunhui Zhu. “Structured illumination-based phase retrieval via Generative Adversarial Network”. In: *Quantitative Phase Imaging VI*. Vol. 11249. SPIE. 2020, pp. 14–22.

[136] Bo Xiao et al. “Optimization for Interval Type-2 Polynomial Fuzzy Systems: A Deep Reinforcement Learning Approach”. In: *IEEE Transactions on Artificial Intelligence* (2022), pp. 1–12.

[137] Yuval Litvak, Armin Biess, and Aharon Bar-Hillel. “Learning pose estimation for high-precision robotic assembly using simulated depth images”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 3521–3527.

[138] Andy Zeng et al. “Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching”. In: *The International Journal of Robotics Research* 41.7 (2022), pp. 690–705.

[139] Suren Kumar, Pankaj Singhal, and Venkat N Krovi. “Computer-vision-based decision support in surgical robotics”. In: *IEEE Design & Test* 32.5 (2015), pp. 89–97.

[140] Bixiao Wu, Junpei Zhong, and Chenguang Yang. “A visual-based gesture prediction framework applied in social robots”. In: *IEEE/CAA Journal of Automatica Sinica* 9.3 (2021), pp. 510–519.

[141] Kiru Park et al. “Multi-task template matching for object detection, segmentation and pose estimation using depth images”. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 7207–7213.

[142] Xucheng Wang, Chenning Tao, and Zhenrong Zheng. “Occlusion-aware light field depth estimation with view attention”. In: *Optics and Lasers in Engineering* 160 (2023), p. 107299.

[143] Pyojin Kim, Hyon Lim, and H Jin Kim. “Robust visual odometry to irregular illumination changes with RGB-D camera”. In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2015, pp. 3688–3694.

[144] Jay M Wong et al. “Segicp: Integrated deep semantic segmentation and pose estimation”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 5784–5789.

[145] Ross Girshick. “Fast r-cnn”. In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[146] Kaiming He et al. “Mask R-CNN”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2961–2969.

[147] Tsung-Yi Lin et al. “Focal loss for dense object detection”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2980–2988.

[148] Wadim Kehl et al. “SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1521–1529.

[149] Peter Florence, Lucas Manuelli, and Russ Tedrake. "Self-supervised correspondence in visuomotor policy learning". In: *IEEE Robotics and Automation Letters* 5.2 (2019), pp. 492–499.

[150] Sergey Levine et al. "End-to-end training of deep visuomotor policies". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 1334–1373.

[151] Sergey Levine et al. "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection". In: *The International journal of robotics research* 37.4-5 (2018), pp. 421–436.

[152] Lerrel Pinto and Abhinav Gupta. "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours". In: *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2016, pp. 3406–3413.

[153] Lerrel Pinto et al. "Asymmetric actor critic for image-based robot learning". In: *arXiv* (2017). ISSN: 23318422. DOI: `10.15607/rss.2018.xiv.008`. arXiv: `1710.06542`.

[154] Yevgen Chebotar et al. "Path integral guided policy search". In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 3381–3388.

[155] Bo Xiao et al. "Sampled-data control through model-free reinforcement learning with effective experience replay". In: *Journal of Automation and Intelligence* 2.1 (2023), pp. 20–30. ISSN: 2949-8554. DOI: `https://doi.org/10.1016/j.jai.2023.100018`. URL: `https://www.sciencedirect.com/science/article/pii/S2949855423000011`.

[156] Guillaume Lample and Devendra Singh Chaplot. "Playing FPS games with deep reinforcement learning". In: *Thirty-First AAAI Conference on Artificial Intelligence*. 2017.

[157] Jan Matas, Stephen James, and Andrew J Davison. "Sim-to-real reinforcement learning for deformable object manipulation". In: *Conference on Robot Learning*. PMLR. 2018, pp. 734–743.

[158] Andrei A Rusu et al. "Sim-to-real robot learning from pixels with progressive nets". In: *Conference on Robot Learning*. PMLR. 2017, pp. 262–270.

[159] Florian Golemo et al. "Sim-to-real transfer with neural-augmented robot simulation". In: *Conference on Robot Learning*. PMLR. 2018, pp. 817–828.

[160] Michael A Goodrich and Alan C Schultz. *Human-robot interaction: a survey.* Now Publishers Inc, 2008.

[161] Zhaoyang Jacopo Hu et al. "Towards Human-Robot Collaborative Surgery: Trajectory and Strategy Learning in Bimanual Peg Transfer". In: *IEEE Robotics and Automation Letters* 8.8 (2023), pp. 4553–4560. DOI: 10.1109/LRA.2023.3285478.

[162] Thomas B Sheridan. "Human–robot interaction: status and challenges". In: *Human factors* 58.4 (2016), pp. 525–532.

[163] Amedeo Cesta et al. "Towards a planning-based framework for symbiotic human-robot collaboration". In: *2016 IEEE 21st international conference on emerging technologies and factory automation (ETFA)*. IEEE. 2016, pp. 1–8.

[164] Wen Qi et al. "Multi-sensor guided hand gesture recognition for a teleoperated robot using a recurrent neural network". In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 6039–6045.

[165] Shiyuan Qiu et al. "Brain–machine interface and visual compressive sensing-based teleoperation control of an exoskeleton robot". In: *IEEE Transactions on Fuzzy Systems* 25.1 (2016), pp. 58–69.

[166] Rajni V Patel, S Farokh Atashzar, and Mahdi Tavakoli. "Haptic feedback and force-based teleoperation in surgical robotics". In: *Proceedings of the IEEE* 110.7 (2022), pp. 1012–1027.

[167] Weibang Bai et al. "Anthropomorphic Dual-Arm Coordinated Control for a Single-Port Surgical Robot Based on Dual-Step Optimization". In: *IEEE Transactions on Medical Robotics and Bionics* 4.1 (2022), pp. 72–84. DOI: 10.1109/TMRB.2022.3145673.

[168] Yijun Cheng et al. "Foot gestures to control the grasping of a surgical robot". In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 6844–6850. DOI: 10.1109/ICRA48891.2023.10160368.

[169] Gongfa Li et al. "Hand gesture recognition based on convolution neural network". In: *Cluster Computing* 22 (2019), pp. 2719–2729.

[170] Pavlo Molchanov et al. "Hand gesture recognition with 3D convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2015, pp. 1–7.

[171] Nurettin Çağrı Kılıboz and Uğur Güdükbay. "A hand gesture recognition technique for human–computer interaction". In: *Journal of Visual Communication and Image Representation* 28 (2015), pp. 97–104.

[172] Christian Zimmermann and Thomas Brox. "Learning to estimate 3d hand pose from single rgb images". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 4903–4911.

[173] Liuhao Ge et al. "3d hand shape and pose estimation from a single rgb image". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 10833–10842.

[174] Tomas Simon et al. "Hand keypoint detection in single images using multiview bootstrapping". In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2017, pp. 1145–1153.

[175] Osama Mazhar et al. "A real-time human-robot interaction framework with robust background invariant hand gesture detection". In: *Robotics and Computer-Integrated Manufacturing* 60 (2019), pp. 34–48.

[176] Ankur Handa et al. "Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system". In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 9164–9170.

[177] Yuzhe Qin et al. "Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system". In: *arXiv preprint arXiv:2307.04577* (2023).

[178] Michael E Walker, Hooman Hedayati, and Daniel Szafir. "Robot teleoperation with augmented reality virtual surrogates". In: *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE. 2019, pp. 202–210.

[179] Patrick Stotko et al. "A VR system for immersive teleoperation and live exploration with a mobile robot". In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 3630–3637.

[180] George Michalos et al. "Augmented reality (AR) applications for supporting human-robot interactive cooperation". In: *Procedia CIRP* 41 (2016), pp. 370–375.

[181] Jared A Frank, Matthew Moorhead, and Vikram Kapila. "Realizing mixed-reality environments with tablets for intuitive human-robot collaboration for object manipulation tasks". In: *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2016, pp. 302–307.

[182] Rebecca M Pierce, Elizabeth A Fedalei, and Katherine J Kuchenbecker. "A wearable device for controlling a robot gripper with fingertip contact, pressure, vibrotactile, and grip force feedback". In: *2014 IEEE Haptics Symposium (HAPTICS)*. IEEE. 2014, pp. 19–25.

[183] Tommaso Proietti et al. "Sensing and control of a multi-joint soft wearable robot for upper-limb assistance and rehabilitation". In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 2381–2388.

[184] Bin Fang et al. "A novel data glove using inertial and magnetic sensors for motion capture and robotic arm-hand teleoperation". In: *Industrial Robot: An International Journal* 44.2 (2017), pp. 155–165.

[185] Hooman Hedayati, Michael Walker, and Daniel Szafir. "Improving collocated robot teleoperation with augmented reality". In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. 2018, pp. 78–86.

[186] Xin Wang, Dharmaraj Veeramani, and Zhenhua Zhu. "Wearable Sensors-Based Hand Gesture Recognition for Human–Robot Collaboration in Construction". In: *IEEE Sensors Journal* 23.1 (2022), pp. 495–505.

[187] Sharlene N Flesher et al. "A brain-computer interface that evokes tactile sensations improves robotic arm control". In: *Science* 372.6544 (2021), pp. 831–836.

[188] Rihab Bousseta et al. "EEG based brain computer interface for controlling a robot arm movement through thought". In: *Irbm* 39.2 (2018), pp. 129–135.

[189] Heike Brock and Randy Gomez. "Personalization of human-robot gestural communication through voice interaction grounding". In: *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2021, pp. 846–853.

[190] Marc Peral, Alberto Sanfeliu, and Anaıés Garrell. "Efficient hand gesture recognition for human-robot interaction". In: *IEEE Robotics and Automation Letters* 7.4 (2022), pp. 10272–10279.

[191] Ziwei Wang et al. "Learning to Assist Bimanual Teleoperation using Interval Type-2 Polynomial Fuzzy Inference". In: *IEEE Transactions on Cognitive and Developmental Systems* (2023), pp. 1–1. DOI: 10.1109/TCDS.2023.3272730.

[192] Julieta Martinez, Michael J Black, and Javier Romero. "On human motion prediction using recurrent neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2891–2900.

[193] Muhammad Awais and Dominik Henrich. "Proactive premature intention estimation for intuitive human-robot collaboration". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 4098–4103.

[194] Loris Roveda et al. "Assisting operators in heavy industrial tasks: On the design of an optimized cooperative impedance fuzzy-controller with embedded safety rules". In: *Frontiers in Robotics and AI* 6 (2019), p. 463524.

[195] Andrey Rudenko et al. "Human motion trajectory prediction: A survey". In: *The International Journal of Robotics Research* 39.8 (2020), pp. 895–935.

[196] Stefanos Nikolaidis et al. “Efficient model learning from joint-action demonstrations for human-robot collaborative tasks”. In: *Proceedings of the tenth annual ACM/IEEE international conference on human-robot interaction*. 2015, pp. 189–196.

[197] Thibaut Munzer, Marc Toussaint, and Manuel Lopes. “Efficient behavior learning in human–robot collaboration”. In: *Autonomous Robots* 42 (2018), pp. 1103–1115.

[198] Samuele Vinanzi, Angelo Cangelosi, and Christian Goerick. “The role of social cues for goal disambiguation in human-robot cooperation”. In: *2020 29th IEEE international conference on robot and human interactive communication (RO-MAN)*. IEEE. 2020, pp. 971–977.

[199] Przemyslaw A Lasota and Julie A Shah. “A multiple-predictor approach to human motion prediction”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 2300–2307.

[200] Mark Ison and Panagiotis Artemiadis. “Proportional myoelectric control of robots: muscle synergy development drives performance enhancement, retainment, and generalization”. In: *IEEE Transactions on Robotics* 31.2 (2015), pp. 259–268.

[201] Hang Su et al. “Deep neural network approach in EMG-based force estimation for human–robot interaction”. In: *IEEE Transactions on Artificial Intelligence* 2.5 (2021), pp. 404–412.

[202] Markus Nowak and Claudio Castellini. “The LET procedure for prosthetic myocontrol: towards multi-DOF control using single-DOF activations”. In: *PLoS One* 11.9 (2016), e0161678.

[203] Chenguang Yang et al. “Development of a robotic teaching interface for human to human skill transfer”. In: *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 710–716.

[204] Achim Buerkle et al. “EEG based arm movement intention recognition towards enhanced safety in symbiotic Human-Robot Collaboration”. In: *Robotics and Computer-Integrated Manufacturing* 70 (2021), p. 102137.

[205] Taekyoung Kim et al. “Heterogeneous sensing in a multifunctional soft sensor for human-robot interfaces”. In: *Science robotics* 5.49 (2020), eabc6878.

[206] Nadav D Kahanowich and Avishai Sintov. “Robust classification of grasped objects in intuitive human-robot collaboration using a wearable force-myography device”. In: *IEEE Robotics and Automation Letters* 6.2 (2021), pp. 1192–1199.

[207] Yijun Cheng et al. “Foot gestures to control the grasping of a surgical robot”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023, pp. 6844–6850. DOI: 10.1109/ICRA48891.2023.10160368.

[208] Su Kyoung Kim et al. “Intrinsic interactive reinforcement learning–Using error-related potentials for real world human-robot interaction”. In: *Scientific reports* 7.1 (2017), pp. 1–16.

[209] Mikkel KNUDSEN and Jari KAİVO-OJA. “Collaborative Robots: Frontiers of Current Literature”. In: *Journal of Intelligent Systems: Theory and Applications* 3.2 (2020), pp. 13–20. DOI: 10.38016/jista.682479.

[210] Xin Ma et al. “Digital twin enhanced human-machine interaction in product lifecycle”. In: *Procedia CIRP* 83 (2019), pp. 789–793. ISSN: 22128271. DOI: 10.1016/j.procir.2019.04.330. URL: https://doi.org/10.1016/j.procir.2019.04.330.

[211] Danica Kragic et al. “Interactive, collaborative robots: Challenges and opportunities”. In: *IJCAI International Joint Conference on Artificial Intelligence* 2018-July (2018), pp. 18–25. ISSN: 10450823. DOI: 10.24963/ijcai.2018/3.

[212] Atle Aalerud and Geir Hovland. “Evaluation of Perception Latencies in a Human-Robot Collaborative Environment”. In: *Proceedings - IEEE International Conference on Robotics and Automation* 978 (2020), pp. 5018–5023. ISSN: 10504729. DOI: 10.1109/ICRA40945.2020.9197067.

[213] Bruno Maric, Alan Mutka, and Matko Orsag. “Collaborative Human-Robot Framework for Delicate Sanding of Complex Shape Surfaces”. In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 2848–2855. ISSN: 23773766. DOI: 10.1109/LRA.2020.2969951.

[214] Ye Gu, Anand Thobbi, and Weihua Sheng. “Human-robot collaborative manipulation through imitation and reinforcement learning”. In: *2011 IEEE International Conference on Information and Automation, ICIA 2011* June (2011), pp. 151–156. DOI: 10.1109/ICINFA.2011.5948979.

[215] Ye Gu et al. “Automated assembly skill acquisition and implementation through human demonstration”. In: *Robotics and Autonomous Systems* 99 (2018), pp. 1–16. ISSN: 09218890. DOI: 10.1016/j.robot.2017.10.002.

[216] Sergey Levine et al. “Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection”. In: *International Journal of Robotics Research* 37.4-5 (2018), pp. 421–436. ISSN: 17413176. DOI: 10.1177/0278364917710318. arXiv: 1603.02199.

[217] Leonel Rozo et al. "Learning and sequencing of object-centric manipulation skills for industrial tasks". In: *arXiv* ii (2020), pp. 9072–9079. ISSN: 23318422. DOI: 10.1109/iros45743.2020.9341570. arXiv: 2008.10471.

[218] Akira Kanazawa, Jun Kinugawa, and Kazuhiro Kosuge. "Adaptive Motion Planning for a Collaborative Robot Based on Prediction Uncertainty to Enhance Human Safety and Work Efficiency". In: *IEEE Transactions on Robotics* 35.4 (2019), pp. 817–832. ISSN: 19410468. DOI: 10.1109/TRO.2019.2911800.

[219] Marco Faber, Alexander Mertens, and Christopher M. Schlick. "Cognition-enhanced assembly sequence planning for ergonomic and productive human–robot collaboration in self-optimizing assembly cells". In: *Production Engineering* 11.2 (2017), pp. 145–154. ISSN: 18637353. DOI: 10.1007/s11740-017-0732-9.

[220] Timo Bänziger, Andreas Kunz, and Konrad Wegener. "Optimizing human–robot task allocation using a simulation tool based on standardized work descriptions". In: *Journal of Intelligent Manufacturing* 31.7 (2020), pp. 1635–1648. ISSN: 15728145. DOI: 10.1007/s10845-018-1411-1. URL: https://doi.org/10.1007/s10845-018-1411-1.

[221] Liang-Chieh Chen et al. "Encoder-decoder with atrous separable convolution for semantic image segmentation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 801–818.

[222] Jonghee Kim et al. "Robust template matching using scale-adaptive deep convolutional features". In: *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE. 2017, pp. 708–711.

[223] Jae-Chern Yoo and Tae Hee Han. "Fast normalized cross-correlation". In: *Circuits, Systems and Signal Processing* 28 (2009), pp. 819–843.

[224] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[225] Tuomas Haarnoja et al. "Soft actor-critic algorithms and applications". In: *arXiv preprint arXiv:1812.05905* (2018).

[226] Abien Fred Agarap. "Deep learning using rectified linear units (relu)". In: *arXiv preprint arXiv:1803.08375* (2018).

[227] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[228] Tuomas Haarnoja et al. “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor”. In: *International conference on machine learning*. PMLR. 2018, pp. 1861–1870.

[229] Tuomas Haarnoja et al. “Reinforcement learning with deep energy-based policies”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1352–1361.

[230] Marcin Andrychowicz et al. “Hindsight experience replay”. In: *Advances in neural information processing systems* 30 (2017).

[231] Pauline C Ng and Steven Henikoff. “SIFT: Predicting amino acid changes that affect protein function”. In: *Nucleic acids research* 31.13 (2003), pp. 3812–3814.

[232] Joseph Redmon and Ali Farhadi. “Yolov3: An incremental improvement”. In: *arXiv preprint arXiv:1804.02767* (2018).

[233] Berk Calli et al. “The ycb object and model set: Towards common benchmarks for manipulation research”. In: *2015 international conference on advanced robotics (ICAR)*. IEEE. 2015, pp. 510–517.

[234] Benjamin Ellenberger. *PyBullet Gymperium*. `https://github.com/benelot/pybullet-gym`. 2018.

[235] Mircea Cimpoi et al. “Describing textures in the wild”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 3606–3613.

[236] Timothy P Lillicrap et al. “Continuous control with deep reinforcement learning”. In: *arXiv preprint arXiv:1509.02971* (2015).

[237] Zhaojie Ju et al. “An integrative framework of human hand gesture segmentation for human–robot interaction”. In: *IEEE Systems Journal* 11.3 (2015), pp. 1326–1336.

[238] Pedro Neto et al. “Gesture-based human-robot interaction for human assistance in manufacturing”. In: *The International Journal of Advanced Manufacturing Technology* 101 (2019), pp. 119–135.

[239] Cristina Nuzzi et al. “HANDS: an RGB-D dataset of static hand-gestures for human-robot interaction”. In: *Data in Brief* 35 (2021), p. 106791.

[240] Dan Xu et al. “Online dynamic gesture recognition for human robot interaction”. In: *Journal of Intelligent & Robotic Systems* 77.3-4 (2015), pp. 583–596.

[241] Jonathan Eden et al. “Principles of human movement augmentation and the challenges in making it a reality”. In: *Nature Communications* 13.1 (2022), p. 1345.

[242] A. Zhang. *Speech Recognition*. Version 3.8 [Software]. Available from `https://github.com/Uberi/speech_recognition`. 2017.

[243] Camillo Lugaresi et al. "Mediapipe: A framework for building perception pipelines". In: *arXiv preprint arXiv:1906.08172* (2019).

[244] Yao Guo et al. "Eye-tracking for performance evaluation and workload estimation in space telerobotic training". In: *IEEE Transactions on Human-Machine Systems* 52.1 (2021), pp. 1–11.

[245] Declan Shanahan, Ziwei Wang, and Allahyar Montazeri. "Robotics and Artificial Intelligence in the Nuclear Industry: From Teleoperation to Cyber Physical Systems". In: *Artificial Intelligence for Robotics and Autonomous Systems Applications*. Springer, 2023, pp. 123–166.

[246] Ziwei Wang et al. "Multiple-Pilot Collaboration for Advanced Remote Intervention using Reinforcement Learning". In: *IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society*. 2021, pp. 1–6. DOI: `10.1109/IECON48115.2021.9589570`.

[247] Changwei Luo et al. "Locating facial landmarks using probabilistic random forest". In: *IEEE Signal Processing Letters* 22.12 (2015), pp. 2324–2328.

[248] Hira Ansar et al. "Hand gesture recognition based on auto-landmark localization and reweighted genetic algorithm for healthcare muscle activities". In: *Sustainability* 13.5 (2021), p. 2961.