Augmented Reality-Driven Robotic Arm Control for Industrial Automation: Integrating Real-Time Gesture Recognition and Image Segmentation

Yuhang Jiang

Advisor: Dr. Vincenzo Cutrona

Department of Innovative Technology (DTI)

University of Applied Science and Arts of Southern Switzerland (SUPSI)

September 2024

***Abstract*--This project integrates Augmented Reality (AR) technology with industrial automation to provide an efficient, intuitive solution for controlling a robotic arm during part-picking tasks. By leveraging Microsoft HoloLens 2 for hand gesture recognition and using advanced models like FastSAM and YOLOv8, the system allows users to visually interact with an AR interface or use simple gestures to pick specific parts in real-time. The project successfully demonstrated high segmentation accuracy and real-time response, enhancing the operational efficiency of industrial environments. However, it also encountered challenges, such as hand occlusion during segmentation, which were mitigated using a time-sharing processing strategy. The project's flexibility, combined with its potential for expansion into multi-device collaboration and additional gesture controls, positions it as a promising solution for future smart manufacturing scenarios.**

***Keywords:* Augmented Reality (AR), Industrial Automation, Real-Time Gesture Recognition, Image Segmentation, Industry 5.0**

1. Introduction

1.1 Project Background and Motivation

With the increasing digitalization of global manufacturing, intelligent manufacturing and automation technologies are gradually replacing traditional manual labor. One of the core concepts of Industry 4.0 is leveraging technologies such as the Internet of

Things (IoT), cloud computing, and artificial intelligence (AI) to enhance production efficiency, improve quality, and reduce costs. Against this backdrop, automated production lines, intelligent robots, and data-driven decision-making systems have become integral parts of modern manufacturing.

However, numerous challenges persist in the field of industrial automation, particularly in tasks that are complex and varied. In the processes of part picking and assembly, for example, uncertainties regarding part shapes, sizes, and positions often necessitate precise manual operations, which in turn reduce overall production efficiency. Additionally, manual operations are susceptible to fatigue, emotional factors, and other subjective influences, leading to potential errors and safety risks.

Augmented reality (AR) technology offers a viable solution to these challenges. By overlaying virtual information onto the real world, AR helps operators access visualized environmental data in real time, enabling them to make accurate decisions. This technology has already been applied in various fields such as training, maintenance, design, and production. The integration of AR with industrial automation holds the potential to significantly improve the efficiency of completing complex tasks, reduce human errors, and lower operational complexity.

The motivation for this project is to explore how AR technology can be integrated with industrial robots to simplify part-picking tasks through visual and interactive means. We chose Microsoft HoloLens 2 as the core AR device, leveraging its robust spatial recognition and fingertip tracking capabilities. Users can interact directly with virtual interfaces and control a robotic arm for part picking. By reducing the complexity of manual operations and providing real-time monitoring of task progress, the goal of this project is to offer a more efficient and intuitive solution for automated part-picking in industrial production.

1.2 Interaction Framework Design Motivation and Objectives

The design motivation for the project is twofold:

1) In industrial manufacturing, where a wide variety of products are produced, traditional segmentation methods often require separate model training for each product or part. Applying these methods to complex industrial processes is challenging, as it is not feasible to collect data and train specific models for each operational process, which would be resource-intensive. **Can this project achieve its task without using any datasets specifically tailored to the scene?**

2) Industrial environments often involve complex interaction tasks. **Can an AR-based system provide a more intuitive, straightforward, and "what-you-see-is-what-you-get" interactive experience?**

To achieve the desired project outcomes, the design of the AR interaction interface must meet several key goals and requirements:

1) **Simple and Intuitive User Interface:** The AR interface, serving as a bridge

between the user and the robotic arm, must be easy to understand and operate. Given the diversity of industrial settings, the interface needs to be highly flexible to accommodate different task requirements. By wearing the HoloLens 2, users can clearly see the task operation interface in the augmented reality environment. All control functions, part locations, and the robotic arm's status will be intuitively displayed in the user’s field of view. This design eliminates the need for complex manual inputs, allowing users to control the robotic arm with simple gestures or taps for precise operations.

2) **Efficient Part-Picking Identification and Mapping:** The project employs the Fast Segment Anything Model (FastSAM) for image segmentation to automatically identify the tray containing parts on the workstation. This model can quickly and accurately segment areas of the tray with black parts and map them precisely to the robotic arm’s movement. Users only need to point to a specific part’s area with their finger, and HoloLens 2 will automatically capture the spatial coordinates of the finger and map them to the robotic arm’s preset picking path. This design is both intuitive and efficient, as it does not require additional dataset collection for model training, thereby greatly reducing resource consumption.

3) **Multi-Mode Operation:** To meet the diverse demands of industrial environments, the AR interface offers two modes of operation, enhancing the system's flexibility and scalability. First, users can directly indicate to the robotic arm which specific part to pick through hand gestures. Second, the AR interface provides a control panel where users can select the corresponding part area and initiate the picking operation. These two methods complement each other, enabling the system to adapt to industrial tasks ranging from simple to complex.

4) **Scalability and Future Applications:** The AR system not only applies to the current part-picking task but also possesses good scalability. The framework uses YOLOv8 for gesture-based command execution, where each gesture corresponds to a specific command. In this project, only one gesture is used to trigger FastSAM. In the future, the system could be expanded to include additional features, such as using different gestures to drive different task sequences or introducing multi-device collaboration to handle more complex production tasks. Moreover, as industrial environments evolve, the AR interface design can easily adapt to new task requirements, offering more flexible operational support.

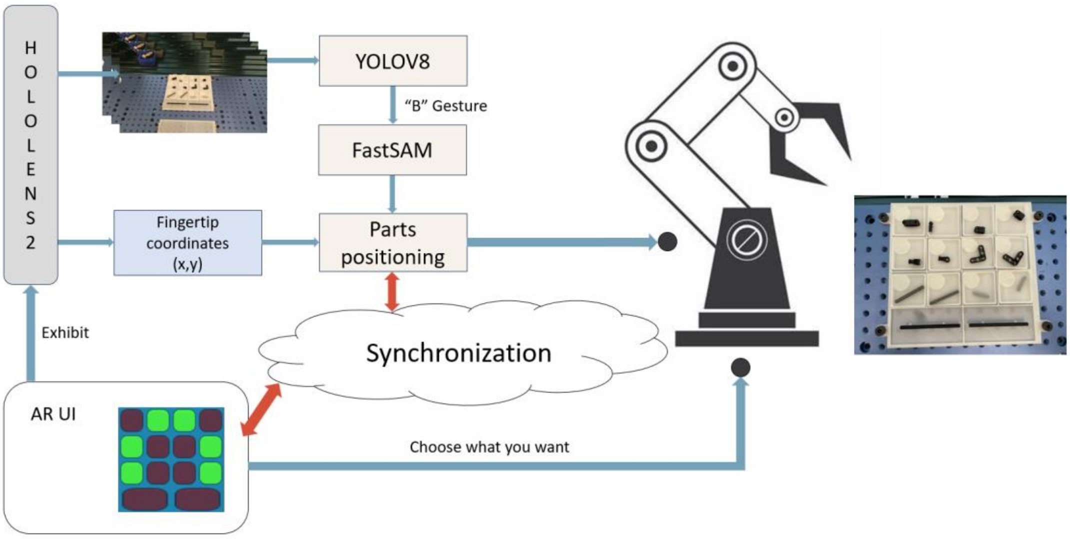


Figure 1: System Architecture

As shown in Figure 1, the AR interaction interface design in this project aims to provide an efficient and intuitive part-picking solution for industrial automation scenarios through a simple and intuitive user interface, efficient recognition and mapping, diverse operation modes, and real-time feedback. This design not only significantly enhances operational efficiency but also reduces human errors, offering new ideas and technical support for the development of industrial automation.

2. Literature Review

Extended Reality (XR) technology, which includes Virtual Reality (VR), Augmented Reality (AR), and Mixed Reality (MR), is playing an increasingly important role in the industrial sector. As the industry transitions from automation-centric Industry 4.0 to a more human-centered Industry 5.0, XR technology, as an emerging collaborative tool, is driving productivity improvements, process optimization, and enhanced worker training and safety in manufacturing.

The key to Industry 5.0 lies in achieving a deep integration between humans and technology, leveraging human creativity and innovation to collaborate with advanced technologies, promoting personalized manufacturing and sustainable development. XR technology has unique advantages in this transformation, not only enhancing operational efficiency but also improving worker engagement and safety through human-machine interaction and immersive experiences.

2.1 Core Concepts of XR Technology

XR technology encompasses multiple levels of immersive experiences and interaction methods, which are primarily divided into the following three categories:

Virtual Reality (VR): Creates a fully immersive digital environment, isolating the user from the real world. VR typically requires a head-mounted display (HMD) and often includes hand controllers for interaction. Popular applications range from gaming and entertainment (e.g., Oculus Rift, HTC Vive) to training simulations and virtual tours [1].

Augmented Reality (AR): Enhances the real world with digital overlays. AR applications use cameras and sensors to superimpose virtual objects onto the physical environment, providing contextual information and interactive experiences. Common examples include mobile apps like Pokémon GO and tools used in industrial maintenance and training [2].

Mixed Reality (MR): Integrates real and virtual worlds to produce new environments where physical and digital objects coexist and interact in real-time. MR technologies, such as Microsoft HoloLens, enable more complex interactions by mapping the physical space and responding to real-world elements.

The development of Extended Reality (XR) technologies relies heavily on advancements in hardware, software, and connectivity, each contributing significantly to the overall user experience and application capabilities. Below is an overview of the key hardware components in XR technologies and their specific roles:

Head Mounted Visualization Device (HMVD):

HMVDs are core devices in Virtual Reality (VR) and Mixed Reality (MR) applications, providing immersive visual and auditory experiences. These devices include see-through displays (e.g., HoloLens) and fully enclosed headsets (e.g., Oculus Rift), which use high-resolution screens and 3D audio to enhance user immersion.

Hand-Held Visualization Device (HHVD):

HHVDs are typically used in Augmented Reality (AR) applications, primarily leveraging the cameras, GPS, and accelerometers in mobile devices. Smartphones and tablets serve as HHVDs, enabling AR experiences through these sensors. Additionally, AR glasses (e.g., Microsoft HoloLens, Google Glass) represent an advanced form of HHVDs, combining see-through displays with powerful sensor arrays to deliver enhanced AR functionality.

Sensors and Cameras:

Sensors and cameras are crucial in MR and advanced AR applications, enabling devices to track user movements and map the surrounding environment. Key sensors include:

1) Depth Sensors (e.g., Kinect): Capture user movements and perform real-time 3D modeling.
2) Eye-Tracking Sensors: Enhance user experience by optimizing rendering and interaction based on eye movements.
3) Gesture Recognition Sensors: Capture hand movements for more natural

interaction.

4) LIDAR (Light Detection and Ranging) Sensors: Precisely map the environment, enhancing the integration of virtual content with the real world.

Haptic Feedback Devices:

Haptic feedback devices provide tactile feedback, making virtual interactions feel more realistic. These devices range from simple vibration feedback to more sophisticated force feedback and full-body suits:

1) Force Feedback Gloves: Simulate the weight, hardness, and texture of objects.

2) Full-Body Haptic Suits: Deliver full-body tactile feedback, enhancing the sense of presence and interaction in virtual environments.

Audio Systems:

Sound is a critical factor in enhancing immersion. 3D audio engines and spatial audio systems deliver realistic sound experiences by accurately positioning sound sources, allowing users to perceive sound coming from various directions, further enhancing immersion.

Networking and Connectivity:

Low-latency networking is essential for real-time data transmission in XR applications, particularly in multi-user collaborative environments. Key technologies include:

1) 5G Networks: Support high bandwidth and low latency communication.

2) Edge Computing: Brings computational resources closer to the data source, reducing latency and improving response times.

Wearable Devices:

Other wearable devices, such as biometric sensors and brain-computer interfaces (BCI), further enhance interaction depth and personalization in XR environments. These devices can monitor users' physiological states and dynamically adjust the content in XR applications, providing a more personalized and immersive experience.

Hardware devices are diverse, with HMVD devices being the most widely used. Table 1 lists commonly used HMVD devices, excluding those that were once popular but have since been discontinued or whose product lines are no longer updated.

Table 1: Commonly used HMVD equipment

Company	Device	Release Year	Type	Configuration	Applications
Microsoft	HoloLens 2	2019	AR	Mixed reality headset with eye tracking, iris recognition, and voice commands. Uses Qualcomm Snapdragon 850 Compute Platform. Display: 2k 3:2 light engines, Resolution: 1440x936, MEMS display. Weight: 566g.	[3-7]

	HoloLens 1	2016	AR	Mixed reality headset with gesture and voice recognition. Uses Intel 32-bit architecture. Display: 2 HD 16:9 light engines. Weight: 579g.	[8-11]
HTC	Vive XR Elite	2023	MR	Mixed reality headset with full-color passthrough, combining VR and AR capabilities. High-resolution display with 6DOF tracking. Compatible with SteamVR and Viveport platforms.	[12]
	Vive Pro 2	2021	VR	High-end VR headset with 5K resolution (2448 x 2448 pixels per eye), 120Hz refresh rate, 120-degree FOV. Requires connection to a high-performance PC. Compatible with SteamVR and Viveport.	——
	Vive Pro 1	2018	VR	VR headset with 2880 x 1600 combined resolution (1440 x 1600 pixels per eye), 90Hz refresh rate, 110-degree FOV. Compatible with SteamVR and Viveport platforms.	[13-14]
Lenovo	ThinkReality VRX	2022	VR	VR headset with full-color passthrough and 6DOF mixed reality capabilities. Powered by Qualcomm Snapdragon XR2 processor, designed for enterprise training, virtual meetings, and immersive simulations. Supports Lenovo's ThinkReality software platform for remote management and content distribution.	——
	ThinkReality A3	2021	AR	AR smart glasses supporting up to 5 virtual displays, equipped with dual 8MP RGB cameras for video recording and streaming. Suitable for enterprise applications such as remote collaboration, 3D visualization, and virtual monitor expansion. Compatible with Windows PCs and select Motorola smartphones. Focused on enterprise applications.	——
	Mirage VR S3	2020	VR	VR headset developed in collaboration with Pico, focused on enterprise-level immersive learning and training. Features a 4K display for clear visual experiences. Standalone device, designed for enterprise customers, does not require external devices.	——
Meta	Oculus Quest 3	2023	VR	Standalone VR headset with Qualcomm Snapdragon XR2 Gen 2 platform, offering higher resolution displays (2064 x 2208 pixels per eye), 120Hz refresh rate, and mixed reality capabilities with full-color passthrough. Improved comfort and slimmer design compared to Quest 2.	——
	Oculus Quest 2	2020	VR	Standalone VR headset with Snapdragon XR2 platform, 6GB RAM, and up to 256GB storage. Resolution: 1832 x 1920 pixels per eye. Refresh Rate: 120 Hz.	[15-16]
Apple	Apple Vision Pro	2024	AR	AR headset with dual 4K micro-OLED displays, M2 chip, 12 cameras for spatial awareness, eye tracking, hand tracking.	——

2.2 XR Development Tools

Software platforms and development environments (see Table 2) are vital for creating XR applications.

1) Game Engines: Unity3D [17] and Unreal Engine [18] are popular choices for XR development due to their robust tools and cross-platform support.

2) AR Development Kits: ARKit (Apple) [19] and ARCore (Google) [20] offer frameworks for building AR applications on mobile devices.

3) MR Platforms: Windows Mixed Reality [21] and HoloLens SDK provide tools for developing MR experiences.

Table 2: Common platforms for developing XR tools

Software Framework	Features/Functions Description
Unity3D	Unity is a cross-platform game engine widely used for XR development, popular due to its robust tools and cross-platform support. It supports AR, VR, and MR development, suitable for various devices.
Unreal Engine	Unreal Engine is another popular cross-platform game engine, widely used for high-end VR and MR experience development. It is known for its realistic graphics and powerful physics engine, suitable for applications requiring high-quality visual effects.
ARKit (Apple)	ARKit is an AR development toolkit provided by Apple for iOS devices, supporting AR applications on iPhones and iPads. It offers deep integration with device cameras, sensors, and CPU/GPU for efficient AR experiences.
ARCore (Google)	ARCore is an AR development platform provided by Google, used for building augmented reality experiences on Android devices. It enables motion tracking, environmental understanding, and light estimation, providing stable AR experiences and supporting cross-platform development.
Windows Mixed Reality	Windows Mixed Reality is a platform provided by Microsoft, supporting the development of MR experiences (such as HoloLens). It offers a range of development tools and SDKs to enable immersive MR experiences.
HoloLens SDK	The HoloLens SDK is a development toolkit provided by Microsoft for its HoloLens devices, supporting the creation of MR applications. It includes features like spatial mapping, gesture recognition, and eye tracking, suitable for MR applications in industrial, medical, and other fields.

2.3 Specific Applications of XR Technology in Manufacturing

In the manufacturing sector, XR technology has already been applied in several areas, such as worker training, equipment maintenance, and design prototyping. The integration of XR technology offers new solutions for complex manufacturing tasks, particularly in reducing cognitive load, improving collaboration efficiency, and enhancing safety.

2.3.1 Training and Education

In the era of Industry 5.0, the demand for highly skilled workers in manufacturing has significantly increased. Extended Reality (XR) technology has been widely applied in manufacturing training and education, aiming to enhance workers' skills, ensure safety, and optimize production processes. Firstly, XR can simulate real working environments, allowing workers to train in a safe, risk-free setting, thereby reducing potential occupational hazards. For example, VR is widely used in the initial and learning stages of training, creating virtual layers that ensure workers' safety and help avoid risks associated with actual operations. Related applications are shown in Table 3.

Table 3: Application of XR in Training and Education

References	Personalization	Benefits	Potential Risks
Assessment of virtual reality-based manufacturing assembly training system [22]	Utilizing VR for the training of assembly operations within manufacturing processes. It incorporates various VR tools, including visual and haptic feedback, to enhance training realism and effectiveness.	VR simulation of assembly processes reduces errors and training time, enhancing understanding in a risk-free environment.	High initial setup costs and potential resistance to change due to new technology adoption.
Designing a Technological Pathway to Empower Vocational Education and Training in the Circular Wood and Furniture Sector through Extended Reality [23]	Integrates extended reality (XR) in vocational training for the wood and furniture sector, employing interactive simulations to engage learners.	Enhanced skill acquisition, increased engagement, and adaptability to various learning styles.	Requires significant investment in technology and ongoing maintenance.
Development of an Extended Reality-Based Collaborative Platform for Engineering Education Operator 5.0 [24]	Develops a collaborative platform using XR to simulate engineering environments, focusing on teamwork and real-time problem solving among students.	Promotes remote education by supporting interaction with complex machinery and fostering a skilled digital workforce.	Technical challenges in implementation and potential over-reliance on virtual scenarios for practical skills training.
Gamified Virtual Reality Training Environment for the Manufacturing Industry [25]	Utilizes gamification in VR training to enhance motivation and learning outcomes in manufacturing settings. Includes real-time feedback and progress tracking.	Increases motivation and learning outcomes through engaging, game-like training environments with higher retention rates.	Risk of cognitive overload and distraction from core learning objectives.
Virtual Reality-Based Engineering Education to Enhance Manufacturing Sustainability in Industry 4.0 [26]	Applies VR in engineering education to simulate sustainable manufacturing processes, with a focus on minimizing environmental impact.	Educates and instills a responsibility towards sustainable practices through virtual scenarios of manufacturing impacts.	Potential high costs and the complexity of integrating VR with existing educational frameworks.

2.3.2 Maintenance and Remote Support

In manufacturing, Extended Reality (XR) technology has seen significant advancements in maintenance and remote support applications. By integrating Augmented Reality (AR) and Virtual Reality (VR), companies can improve the efficiency and accuracy of equipment maintenance, particularly for complex or hard-to-reach components. XR enables technicians to receive real-time guidance and visual overlays, enhancing their ability to perform precise repairs. Additionally, remote experts can provide support, reducing downtime and ensuring maintenance tasks are completed accurately, ultimately boosting equipment reliability and operational efficiency. Related applications are shown in Table 4.

Table 4: Application of XR in Maintenance and Remote Support

Reference	Personalization	Benefits	Potential Risks

Creating an Open-Source Augmented Reality Remote Support Tool for Industry: Challenges and Learnings [27]	Development of an open-source AR framework for remote maintenance and collaboration, enhancing efficiency.	Facilitates customization and adaptation to specific maintenance needs, reducing dependence on proprietary solutions.	Potential technical challenges in maintaining and updating the open-source framework, ensuring compatibility and security.
Real-Time Remote Maintenance Support Based on Augmented Reality (AR) [28]	Uses AR for real-time remote maintenance, providing interactive communication channels between technicians and engineers.	Improves immediacy and effectiveness of maintenance operations, reducing downtime and enhancing operational efficiency.	Challenges related to network dependency, real-time data transmission, and ensuring stable and secure communication channels.
Remote Video Collaboration During COVID-19 [29]	Implements Remote Video Collaboration (RVC) with AR to facilitate equipment installation, maintenance, and training amid travel restrictions.	Ensures continuity of production activities despite logistical challenges, utilizing AR for enhanced interactive support.	Intellectual property concerns with remote access to sensitive information, plus potential resistance to adopting new remote operation models.
Supporting Remote Maintenance in Industry 4.0 through Augmented Reality [30]	Employs off-the-shelf mobile and AR technologies for effective remote maintenance by connecting skilled operators with onsite personnel.	Allows for effective and efficient remote guidance and troubleshooting, reducing the need for expert physical presence.	Dependence on the reliability of mobile and AR technologies, which may face operational challenges in industrial environments.

2.3.3 Design and Prototype

In manufacturing, Extended Reality (XR) technologies, particularly Virtual Reality (VR) and Augmented Reality (AR), have significantly enhanced flexibility and efficiency in design and prototyping. Several case studies from various papers demonstrate how these technologies are practically applied to improve product design and testing processes. By integrating XR, companies can rapidly iterate on designs, visualize prototypes in real-time, and conduct virtual testing, leading to faster development cycles and more innovative products, ultimately optimizing the overall design workflow. Related applications are shown in Table 5.

Table 5: Application of XR in Design and Prototype

Reference	Personalization	Benefits	Potential Risks
Semi-Immersive Virtual Turbine Engine Simulation System [31]	Utilizes a semi-immersive VR system for aircraft turbine engine assembly verification. Features stereoscopic visuals, surround sound, and haptic feedback, along with a special software architecture for VR, including collision detection for assembly interference checks.	Enhances interaction with the model, providing a realistic experience that aids in detailed verification of assembly processes and part design, improving planning operations.	May require high computational capabilities and can be resource-intensive to simulate complex products, possibly limiting field of view and realism.
Exploring the Benefits of Virtual Reality Technologies for Assembly Retrieval Applications [32]	The system supports intuitive interaction through gestures and voice commands, emphasizing its role in efficiently conveying complex assembly similarities and reducing the need for physical prototypes,	Speeds up the validation process of assembly procedures, especially beneficial in complex and detailed assemblies where traditional methods are cumbersome and error-prone.	Dependent on the accuracy and quality of VR systems; limitations in technology may lead to less effective training or planning.

	thereby speeding up design modifications and cost savings.		

2.3.4 Production Process

In the manufacturing industry, the use of extended reality (XR) technology is reshaping the production process, especially through the application of augmented reality (AR) and virtual reality (VR) technology, providing unprecedented visual support and interaction methods for production lines.

These studies (see Table 6) show that by introducing XR technology, the production process of the manufacturing industry can achieve more efficient operations, more precise quality control, and lower error rates. With the continuous development and application of these technologies, it is expected that the production efficiency and product quality of the manufacturing industry will be further improved in the future.

Table 6: Application of XR in Production Process

Reference	Personalization	Benefits	Potential Risks
Quality Assurance and Process Control in Virtual Reality [33]	Focuses on VR-based systems for process control, employing adaptive algorithms that tailor the VR environment according to real-time data from the production line.	Enhances monitoring accuracy and operational efficiency, allowing for quicker response times and adjustments in production processes.	Relies heavily on the accurate synchronization of virtual and real environments, risking delays or errors in response to real-world changes.
The Production Quality Control Process, Enhanced with Augmented Reality Glasses [34]	Utilizes AR glasses to overlay digital information directly onto production components, customized to display relevant data based on the user's tasks and location within the factory.	Improves the accuracy of quality control inspections and speeds up the training process for new employees by providing contextual information and guidance.	Dependence on AR device reliability and the potential for decreased effectiveness if visual overlays are inaccurate or misaligned.
Using Augmented Reality for Industrial Quality Assurance [35]	Employs AR to assist operators on the shop floor by overlaying step-by-step instructions and quality checkpoints directly onto work pieces, with dynamic adjustment based on the task at hand.	Significantly reduces errors and operational time by providing real-time, situational feedback and instructions, improving overall quality assurance.	If AR guidance is incorrect or if there are technical failures, it could lead to significant operational disruptions and quality control issues.
Dialogue Enhanced Extended Reality Interactive System for the Operator 4.0 [36]	Integrates voice interaction into AR systems, allowing operators to customize and interact with the AR environment using natural language processing to adjust instructions or get additional information as needed.	Enhances user engagement and allows for more flexible, intuitive interaction with quality control processes, potentially increasing adherence to standards.	Complexity of maintaining accurate voice recognition in noisy industrial environments, which could lead to misunderstandings or incorrect data display.

2.4 Human Interaction

In the manufacturing industry, Extended Reality (XR) technology demonstrates how to effectively integrate human operators with complex machine systems for human-machine interaction. Table 8 summarizes specific application cases, illustrating how XR technology plays a role in the design of human-machine collaboration sites, machine operation, and human-machine collaboration. Human-centric design is the focus of Industry 5.0, and in the next section, we will focus on the key design considerations related to human factors in XR.

Table 7: Application of XR in Human Interaction

Reference	Personalization	Benefits	Potential Risks
An Approach Based on VR to Design Industrial Human-Robot Collaborative Workstations [37]	Explores the use of VR in designing workstations for human-robot collaboration, enabling risk-free testing and optimization of human-machine interactions before real-world implementation.	Facilitates safe and efficient design of collaborative spaces, minimizing risks and allowing for optimization before actual deployment.	Depends heavily on the fidelity of VR simulations to actual working conditions; discrepancies can lead to ineffective or unsafe designs.
Extended Reality Application Framework for a Digital-Twin-Based Smart Crane [38]	Develops an XR framework using AR and VR to enhance operator interaction with a smart crane. Integrates real-time viewing and operation of complex mechanical components, utilizing digital twin technology for precise and efficient operations.	Improves operational precision and efficiency by allowing operators to interact more intuitively with the crane components.	High reliance on the accuracy of the digital twin model and the XR system's performance, which could affect operational safety if inaccurate.
Using Virtual Manufacturing to Design Human-Centric Factories [39]	Utilizes VR to simulate human-machine interaction scenarios to optimize workstation design and task planning. Focuses on ergonomics to enhance operator comfort and productivity, using tools like Unity 3D, HTC VIVE, Xsens for tracking, and Leap Motion for gesture recognition.	Enhances workstation design for better ergonomics, reducing inefficiencies and improving operator comfort and factory productivity.	Initial high costs for VR setup and potential issues with integrating VR tools into existing manufacturing systems.

2.4.1 User Interface Design

In the realm of Extended Reality (XR) technology, user experience and interface design emerge as pivotal components within industrial applications. Throughout user engagement, the User Interface (UI) represents the most utilized and intimately interacted facet. A meticulously crafted UI can markedly diminish the cognitive burden on users, augment operational efficiency, and elevate overall user satisfaction. Table 8 delineates a variety of adverse consequences stemming from suboptimally designed elements.

Table 8: Adverse consequences stemming from suboptimally designed elements

Poorly designed aspects	Detail	Impact
Poor Integration of Learning Modalities	XR interfaces that poorly integrate different learning modalities—such as auditory, visual, and kinesthetic—can increase cognitive load. This is particularly	This can lead to misunderstandings, errors in task execution, and increased cognitive

	problematic when the interface forces users to split their attention between different types of content, such as text and images, which is not effectively integrated.	load, reducing overall learning outcomes.
Inadequate User Feedback and Interaction	If the XR interface does not provide adequate feedback or allows for intuitive interaction, users may find the system unresponsive or difficult to manipulate. This is often due to poor tactile feedback or unintuitive motion controls.	Poor user experience and increased frustration can result, which diminishes the learning efficiency and can lead to poor retention of information.
Overwhelming Information Presentation	Interfaces that overload users with information without prioritizing content or allowing customization to user preferences can overwhelm users, particularly in educational settings where paced learning is crucial.	This can overload the cognitive capacity of users, leading to reduced information processing efficiency and lower overall engagement with the learning material.

Aiming to address these negative issues, the design of XR User Interfaces (UI) has become a popular research direction. From a modality perspective, this involves the integration of multimodalities. Zimmerer et al. [40] explores how the use of multimodal interfaces can decrease the cognitive load in digital tabletop gaming. The results indicate that integrating visual, auditory, and tactile feedback significantly reduces cognitive stress during complex game tasks, thereby enhancing the player's gaming experience and performance. From an interface design perspective, Xia et al. [41] compared the effects of color in real-world and virtual reality environments on cognitive performance. The findings suggest that color configurations have a significant impact on users' cognitive and emotional states in virtual environments, providing crucial guidance for visual design in VR applications. The impact of interface layout design on learning efficiency on mobile learning platforms was explored. Studies have shown that vertical and horizontal layouts have significantly different impacts on user learning efficiency, where vertical layouts contribute to faster information processing speed and higher learning satisfaction. Zhang et al. [42] explored the impact of interface layout design on learning efficiency on mobile learning platforms. Research indicates that vertical and horizontal layouts significantly affect user learning efficiency differently, with vertical layouts aiding in the enhancement of information processing speed and learning satisfaction.

From the perspective of user psychological cognition, Zhou et al. [43] studied how personality traits affect user trust in human-machine collaboration under conditions of uncertainty and cognitive load. Experiments involving 42 participants found that the display of uncertainty can increase trust under low cognitive load conditions, but reduces trust under high cognitive load conditions. Different personality traits have varying impacts on trust, providing valuable insights for human-machine interaction design. Bellman et al. [44] explored the impact of feedback mechanisms on task performance and user preferences for interfaces. The study assessed, through experimental methods, how different types of feedback (positive and negative) influence task execution. Results show that appropriate feedback can significantly enhance task performance, and users' interface preferences vary according to their personal characteristics and task requirements. Han et al. [45] studied cognitive load issues in highly immersive virtual reality environments. Using experimental methods, the study explored how different virtual reality tasks demand and impact users' cognitive resources. It was found that a moderate cognitive load can enhance user experience and performance in highly immersive environments, but excessive load may lead to cognitive overload.

2.4.2 Ergonomics and Health

When exploring the ergonomics and health impacts of XR technology, we must focus on the physical comfort and health risks for users during prolonged use of these devices. This section will specifically discuss the ergonomic principles that should be considered in the design of XR devices, as well as how to mitigate the negative health effects on users through design improvements.

Key Areas of Improvement in XR Ergonomics

Weight Distribution:

Many XR devices are front-heavy, which can cause neck strain. Research often explores ways to distribute the weight more evenly, such as by using lighter materials or by adjusting the design to balance the weight across the head.

Fit and Adjustability:

Adjustable straps and customizable padding are essential to ensure that XR devices fit a wide range of users comfortably. This also includes the design of nose bridges and facial interfaces to minimize pressure points.

Heat Management:

XR devices generate heat during use, which can be uncomfortable for the wearer. Studies often look into materials and ventilation systems that improve airflow and reduce heat buildup.

Optical Comfort:

The position of lenses and the quality of the display can affect eye strain. Improvements in optics, such as adjustable interpupillary distance (IPD) and reducing the screen-door effect, are common areas of focus.

User Interaction:

The design of controllers and hand tracking systems can also affect ergonomics. Natural hand positioning, button placement, and the use of haptic feedback are all areas where research is often conducted to improve comfort.

The discomfort symptoms that may be caused by wearing VR/AR are as follows:

Table 9: Discomfort symptoms caused by wearing VR/AR.

Symptom	Description
Cybersickness and Visual Discomfort	**Cybersickness** is primarily caused by sensory conflicts between visual and vestibular systems, leading to symptoms like nausea, dizziness, and visual discomfort. [46]

	Visual strain due to VR exposure can lead to eye fatigue and headaches, often exacerbated by the mismatch between accommodation and vergence in stereoscopic displays. [47]
Physical Discomfort	**Weight of VR headsets** contributes significantly to discomfort, particularly in the neck and shoulders, due to increased muscle strain and pressure on the head. [48]
	Design and fit of VR headsets have a direct impact on physical comfort, with heavier headsets generally leading to shorter periods of tolerable use. [49]
Visual Strain	**Prolonged exposure to digital screens**, including VR, can lead to visual strain, dry eyes, and discomfort. This is often due to a combination of factors including screen flicker, poor resolution, and the close proximity of screens to the eyes. [50]
	3D stereoscopic content in VR can exacerbate eye strain and visual discomfort due to vergence-accommodation conflicts, which require constant eye adjustment to maintain a clear image. [51]
	Brain wave studies have shown that VR, especially when viewed on larger screens, can cause significant visual fatigue, which is reflected in changes in brain wave activity and increased visual load. [52]
Spatial Disorientation and Balance Issues	**Virtual reality environments** can cause spatial disorientation and balance issues due to mismatches between visual inputs from the VR system and the body's vestibular and proprioceptive senses. This can lead to increased postural instability and balance problems, sometimes persisting after the VR session has ended. [53]
	Full-immersion VR games, especially those with changing backgrounds, have been found to significantly affect static balance, leading to dizziness and other adverse effects such as eye fatigue. [54]
Mental Fatigue	**Cognitive load** in VR environments can lead to mental fatigue, particularly when users engage in complex tasks or face information overload. Studies have shown that VR environments can increase cognitive load, leading to reduced performance and increased mental strain. [55]
	Prolonged use of VR can induce mental fatigue, which impairs cognitive functions such as attention and task performance. This has been observed in both simulated and real-world environments. [56-57

The XRSISE [14] system enhances XR training ergonomics through advanced technologies. It uses Digital Human Models (DHM) to simulate tasks and assess ergonomic risks, allowing for preemptive adjustments to reduce physical strain. Its Biophysical Assessment Module analyzes biomechanics such as posture and muscle strain, enabling ergonomic optimization of virtual environments. The system personalizes training scenarios to individual ergonomic needs via virtual profiles, ensuring active engagement in safe, optimized settings. Moreover, it features real-time ergonomic feedback tools that facilitate immediate adjustments during training sessions, enhancing both safety and comfort. These integrated elements underscore XRSISE's commitment to embedding ergonomic principles deeply within industrial XR training systems, aiming to elevate safety, comfort, and efficiency.

Zhang et al. [58] investigated the impact of significant anatomical contractions on user comfort in virtual reality (VR) and augmented reality (AR) environments. By using surface electromyography (EMG) sensors, the authors measured and modeled the substantial strain levels experienced by users when using head-mounted displays (HMDs). The study proposed a biophysically-based computational model capable of predicting users' maximum contraction levels (MCL) during and before head movements, thereby forecasting potential discomfort. Through user experiments and exploratory studies, the accuracy of the model was evaluated, and its potential to optimize visual target layouts to reduce bottlenecks was demonstrated. The model's success shows that prior to designing and deploying VR/AR applications, computer-based predictions can help reduce user discomfort, providing a new direction for future health-focused VR/AR design.

2.4.3 Ethical and Privacy Considerations

As Extended Reality (XR) technology advances, especially in manufacturing, ethical and privacy concerns have become critical discussion points. The integration of XR in industrial settings involves the collection of significant personal and operational data, raising issues around data security, user consent, and potential misuse. The immersive nature of XR further complicates these concerns, making it essential to address them to ensure responsible and ethical deployment of the technology. This section will explore these ethical and privacy challenges and discuss strategies to mitigate them in the context of XR's application in manufacturing.

Holderman et al. [59] outline the essential privacy, safety, and wellbeing considerations necessary for AR and VR technologies. They emphasize privacy-centric technology practices across hardware and application layers, ethical and transparent information collection, especially regarding indigenous and minority cultures, and the need for privacy by design in open-source XR frameworks. The paper serves as a crucial reminder of the societal impacts these technologies may have, pushing for developers and companies to adopt responsible design practices

Maurice et al. [60] discuss the ethical and social considerations for the introduction of human-centered technologies in workplaces, particularly focusing on collaborative robots, exoskeletons, and wearable sensors. The study addresses both the potential and the challenges of these technologies in improving working conditions, particularly in reducing the prevalence of musculoskeletal disorders among workers. However, it underscores that while these technologies can improve physical safety, their successful deployment hinges on addressing ethical and social aspects to ensure technology acceptance.

Indiparambil [61] delves into the complex socio-ethical issues associated with on-the-job surveillance, emphasizing the clash between the potential benefits of surveillance, such as increased profitability and productivity, and its ethical pitfalls, particularly the pervasive invasion of privacy. The paper critically evaluates both the technological frameworks that facilitate surveillance and the justifications often cited for its use, such as workplace safety and risk management. Indiparambil argues that the moral and social implications of surveillance often transcend the immediate benefits touted by organizations, urging a reevaluation of ethical practices in workplace monitoring. The study highlights the dual nature of surveillance as both a method of coercive control and a form of care, exploring how these dimensions influence organizational behavior and employee perceptions. Indiparambil proposes an ethics of workplace surveillance founded on trust and transparency, suggesting modifications to traditional surveillance practices to better align them with ethical principles and legal compliance.

3. Application Report

This project aims to integrate AR technology with industrial scenarios to help users intuitively and efficiently control a robotic arm for part-picking tasks. In industrial automation, precise and efficient part-picking is crucial for the continuity and efficiency of production lines. The design of the AR interaction interface is intended to reduce the complexity of traditional manual inputs, allowing users to control the robotic arm through natural gestures or an intuitive graphical interface, thereby significantly improving operational smoothness and flexibility.

The AR interaction interface is based on two primary modes of operation:

1) The user performs the corresponding gesture within their field of view, triggering the FastSAM model to segment the white tray area in the image. Subsequently, Hough transform and geometric prior information of the tray are combined to extract individual part areas. By pointing directly at the part to be picked, the system matches the fingertip spatial coordinates from the HoloLens 2 with the extracted part areas and maps them to the robotic arm's picking position, enabling the successful retrieval of the desired part.

2) In the AR control interface, the user can select the part area by clicking on the corresponding region on the panel, driving the robotic arm to pick the part. This operation is more intuitive and faster, serving as a complement to Operation Mode 1.

The core function of the AR interaction interface is to assist users in precisely controlling the industrial robotic arm through a visualized interface and gesture control. The interface architecture includes:

1) **Gesture Recognition Module**: The YOLOv8 model is used to recognize the corresponding command gestures, which then trigger the FastSAM process. This avoids the need for FastSAM to run continuously in real-time, preventing unnecessary computational resource usage and maintaining system stability.

2) **Tray Area Segmentation Module**: This module combines FastSAM, Hough transform, and the prior knowledge of part placement in the tray to quickly segment the part areas in the tray, enabling mapping to the corresponding robotic arm operations.

3) **AR Control Panel**: Through the control panel, users can directly select parts and drive the robotic arm for part-picking. This design provides a simpler alternative operation mode, suitable for different task requirements.

3.1 Models Description

In this project, we use the YOLOv8 and FastSAM models to accomplish object detection and image segmentation tasks. Below is a detailed explanation of these two models:

3.1.1 YOLOv8 Model

YOLOv8 (You Only Look Once, version 8) [62] is an advanced object detection model widely used in real-time detection tasks due to its excellent speed and accuracy. In this project, YOLOv8 is employed to recognize the user's current gestures in real-time. Considering both accuracy and speed, I chose the YOLOv8s model for fine-tuning. Its structure is shown in Figure 2.

YOLOv8 processes image data through three main stages: Backbone, Neck, and Head. Each stage performs specific tasks to progressively refine the image data and produce high-quality object detection predictions.

Backbone:

1) The backbone is responsible for extracting feature representations from the input image. YOLOv8 uses CSPDarknet, a variant of the Darknet architecture, as its backbone.
2) Image data is initially processed through several convolutional layers in the backbone, which downsample the image at different scales. This helps the model capture both low-level and high-level features across various receptive fields.
3) Stages: As shown in the diagram, the backbone consists of multiple stages (P1 to P5) where each stage performs downsampling using convolutions and pooling. This process reduces the spatial resolution of the image while increasing the depth of feature maps, allowing the model to focus on important details such as edges and textures.

Neck:

1) After feature extraction, the Neck module aggregates and refines the feature maps from different stages of the backbone. The YOLOv8PAFPN (Path Aggregation Feature Pyramid Network) is employed in this step.
2) The Neck performs upsampling and concatenation of feature maps from different layers to create a more detailed, multi-scale representation of the image.
3) This allows the model to detect objects of varying sizes by considering features from different resolutions. The diagram shows how these features are combined and passed forward using CSPLayer 2Conv blocks, ensuring both efficiency and richness of information.

Head:

1) In the final stage, the **Head** is responsible for producing the detection results. YOLOv8 uses **Decoupled Head** modules, which separate the tasks of **classification** (predicting object categories) and **bounding box regression** (predicting the location of objects).

2) The **Head** processes the refined features from the Neck and generates object predictions at multiple scales. Each scale is specialized for detecting objects of specific sizes.

3) The output includes the bounding boxes, objectness scores, and class probabilities for each detected object. These are passed through non-maximum suppression (NMS) to filter out overlapping boxes and provide the final predictions.

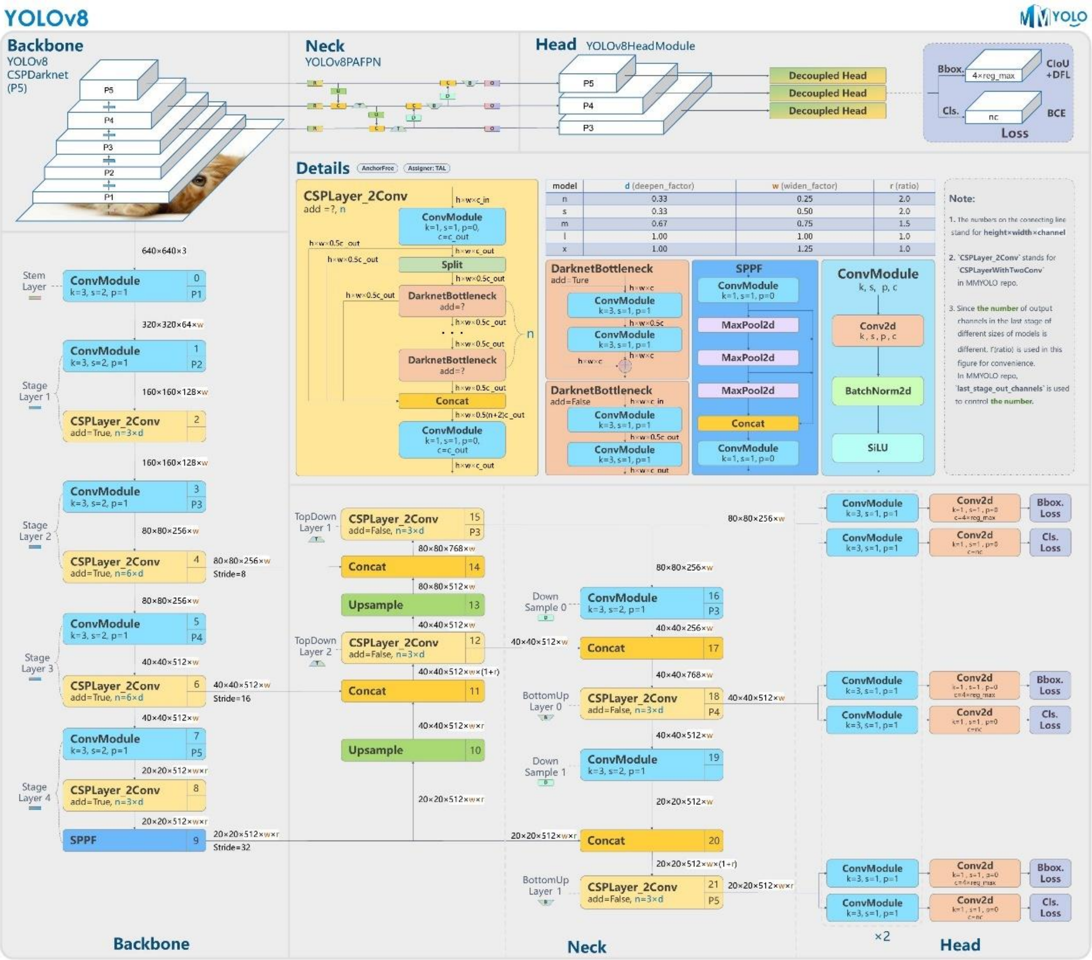


Figure 2: YOLOv8 Model Architecture

3.1.2 FastSAM Model

FastSAM [63] is a lightweight segmentation model based on YOLOv8. It segments all instances in an image and then selects specific objects for segmentation based on prompts. FastSAM is an optimized version of Meta AI's SAM (Segment Anything Model) [64], which excels in performing almost any type of segmentation task, while

FastSAM is specifically optimized for speed. This feature is particularly useful for our project because, with traditional deep learning image segmentation models, we would need to collect a large number of images from the operational scene and spend considerable effort manually annotating the white trays containing black parts, which is impractical. FastSAM helps us achieve this efficiently.

FastSAM divides the task of "segmenting any object" into two steps: first, **All-instance Segmentation**, and second, **Prompt-guided Selection**. In my project, FastSAM uses these two steps to quickly and accurately segment the black parts on the tray and select parts for picking based on user gestures or text input. FastSAM supports various types of prompts, including **Point Prompt**, **Box Prompt**, and **Text Prompt**. The text prompt relies heavily on the CLIP model, which converts natural language descriptions into corresponding image features to select the target object based on the description. In this project, the point prompt allows users to point to the part they want to pick with their finger, while the text prompt enables users to specify the target part through a description. FastSAM uses CLIP to implement text prompts, allowing direct input of text such as "black part," which then automatically selects and segments the corresponding part from the tray for picking. The workflow is shown in Figure 3.

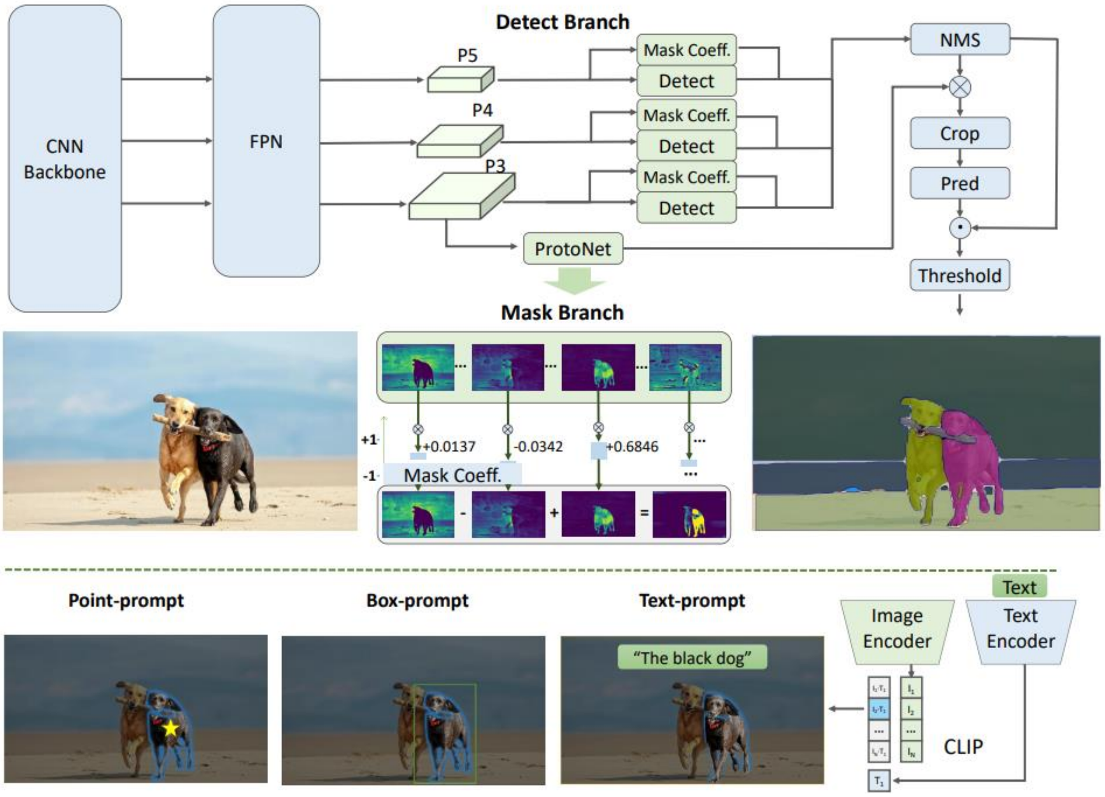


Figure 3: FastSAM Model Workflow and Prompt Types

3.1.3 CLIP Model and Its Text Prompt

The CLIP (Contrastive Language-Image Pre-training) [65] model is a vision-and-language joint training model that maps images and text into the same embedding space through contrastive learning. What sets CLIP apart is its ability to learn from a large number of image-text pairs, enabling it to support zero-shot tasks—performing tasks without additional training on a specific dataset. Through CLIP, users can select objects directly via textual descriptions, providing foundational support for text prompts in this project. The workflow is shown in Figure 4.

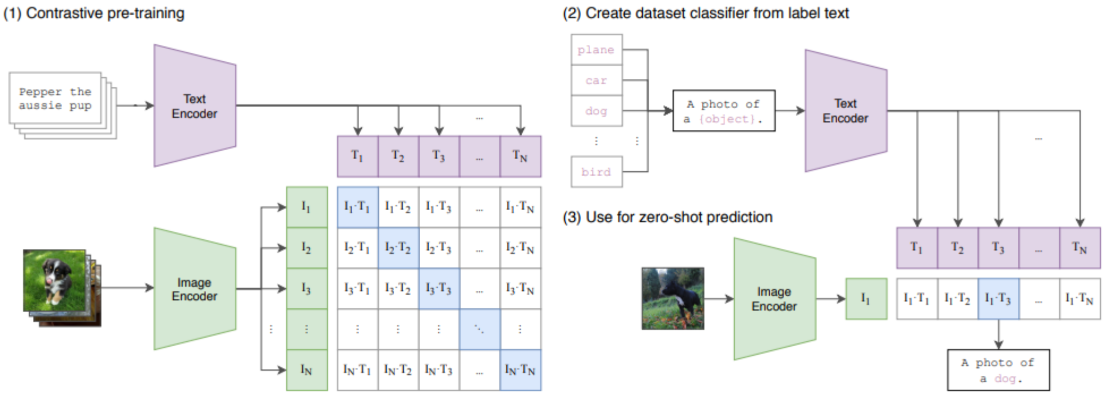

(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
T_1 T_2 T_3 ... T_N
I_1 I_2 I_3 ... I_N
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
$I_2 \cdot T_1$ $I_2 \cdot T_2$ $I_2 \cdot T_3$ ... $I_2 \cdot T_N$
$I_3 \cdot T_1$ $I_3 \cdot T_2$ $I_3 \cdot T_3$ ... $I_3 \cdot T_N$
$I_N \cdot T_1$ $I_N \cdot T_2$ $I_N \cdot T_3$ ... $I_N \cdot T_N$
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
I_1
$I_1 \cdot T_1$ $I_1 \cdot T_2$ $I_1 \cdot T_3$ ... $I_1 \cdot T_N$
A photo of a dog.

Figure 4: CLIP Model Workflow

The working principle of CLIP involves training an image encoder and a text encoder simultaneously, and object recognition is achieved by calculating the cosine similarity between the image embeddings and text embeddings. In my project, CLIP is used to extract embedding features from natural language descriptions such as "black part" or "next part," and these features are matched with the segmented objects, helping the robotic arm accurately select and pick the target object.

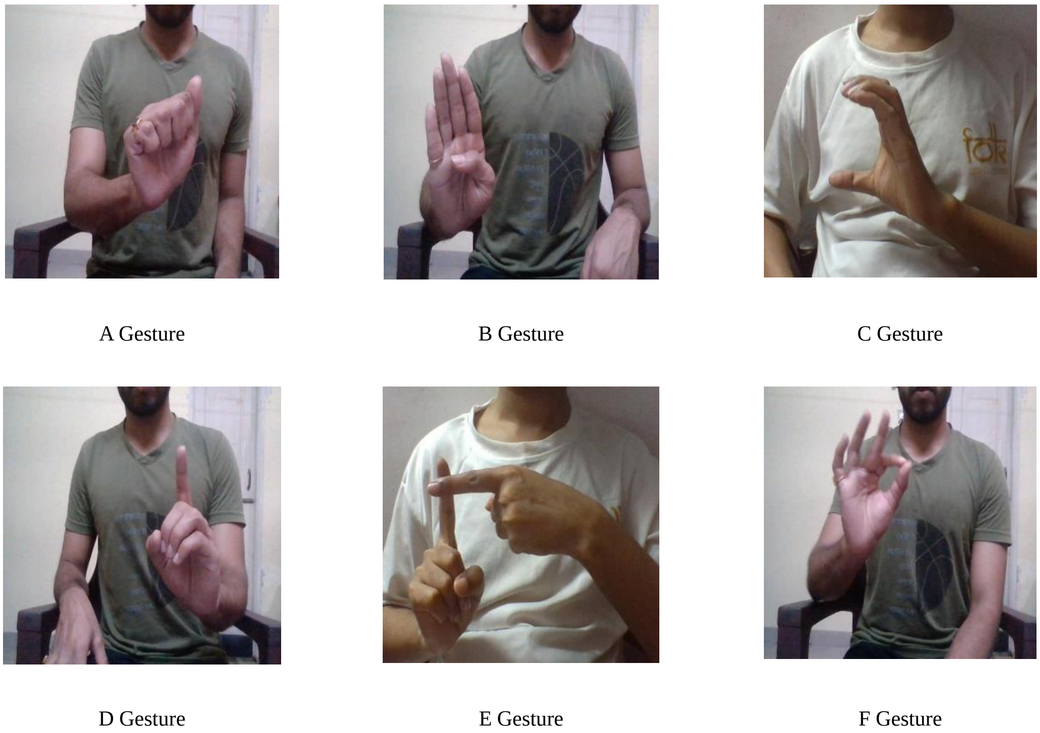
3.2 Overall Framework Design

3.2.1 YOLOV8s Fine-tuning

By combining FastSAM and YOLOv8, the project achieves efficient processing of image segmentation and object detection, providing stable data input for the precise picking operation of the robotic arm.

Relevant Experiments and Test Results

In this project, the YOLOV8 model was fine-tuned using a pre-trained YOLOV8s model from the COCO dataset, and further fine-tuned on a multi-gesture language detection dataset [66]. This dataset contains a total of 276 images, with 193 images in the training set, 55 in the validation set, and 28 in the test set. The dataset includes six types of gestures, as detailed below:



A Gesture

B Gesture

C Gesture

D Gesture

E Gesture

F Gesture

Figure 5: The Sample Category of Gesture Dataset

Since the project currently only requires one gesture as a command, this command should rarely occur in manual operations to avoid false detections by the model. Based on these considerations, we chose gesture type B as the command. Other gestures can be used when additional functionalities are needed in the future.

The hardware configuration for fine-tuning is as follows:

Table 10: Hardware Configuration for YOLOv8 Fine-tuning.

Component	Specification
CPU	Intel(R) Xeon(R) Platinum 8255C
GPU	NVIDIA RTX 2080 Ti (12 GB)
Memory	43 GB
Operating System	Ubuntu 22.04
Python	3.12
Pytorch	torch2.3+cu121
ultralytics	8.2.87

Training Results for YOLOv8 Gesture Detection

Precision-Recall Curve:

The Precision-Recall (PR) curve shows excellent performance across all gesture classes, with a mean average precision (mAP@0.5) of 0.980. The curves for each gesture class (A, B, C, D, E, F) demonstrate near-perfect precision and recall values, particularly for classes A, B, C, E, and F, all achieving a precision of 0.995. Class D shows slightly lower performance, with a recall around 0.906.

Confusion Matrix:

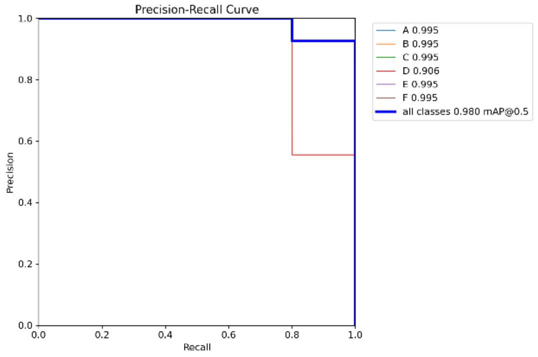
The normalized confusion matrix highlights the model’s classification accuracy across all gesture classes. Classes A, B, E, and F are perfectly classified with no misclassifications. However, some confusion exists between classes C and D, where 20% of class C gestures are misclassified as class D, and 80% of class D gestures are correctly classified. This indicates an area for further improvement in distinguishing between these two gestures.

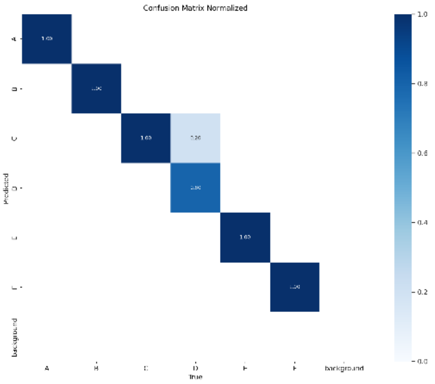
F1-Confidence Curve:

The F1-confidence curve presents the F1 scores of all classes as a function of confidence. The model achieves an overall F1 score of 0.97 at a confidence threshold of 0.766. Most gesture classes show high F1 scores, confirming the model's robust classification ability, though class D again shows slightly lower performance.

Precision-Confidence Curve:

The Precision-confidence curve indicates the precision of all classes as confidence increases. The model achieves an overall precision of 1.00 at a confidence threshold of 0.892. The majority of gesture classes show high precision, with class D exhibiting lower precision, which is consistent with the results observed in the PR and F1-confidence curves.





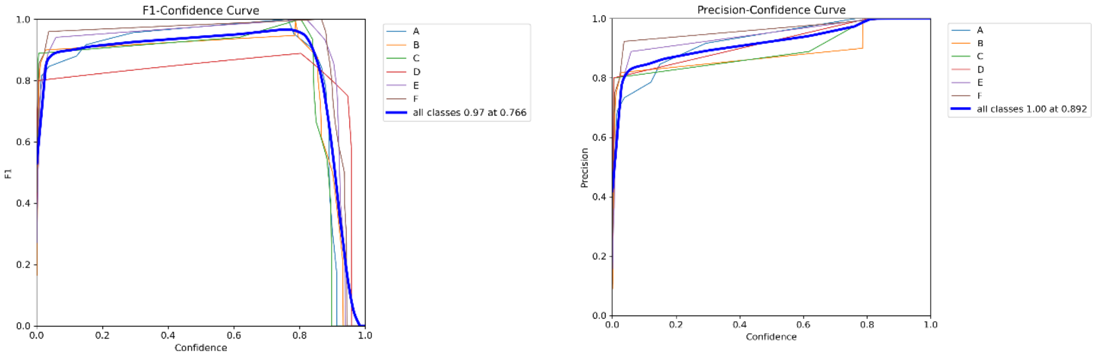
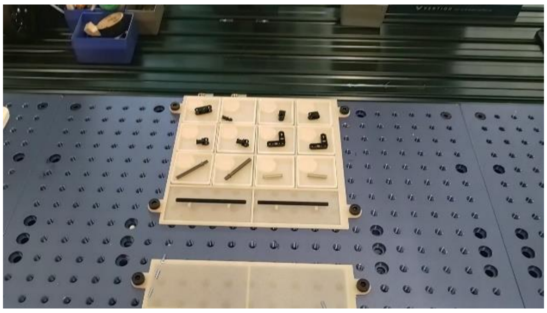


Figure 6: Model Performance Evaluation for Gesture Recognition

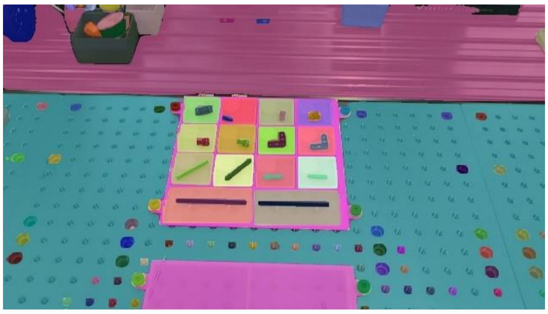
3.3.2 Design and Implementation of FastSAM Segmentation Scenario

During testing, FastSAM demonstrated good performance in terms of segmentation accuracy and real-time processing. However, it struggled when parts of the tray were occluded by a human hand, leading to segmentation errors. This occurred because we did not fine-tune FastSAM for this specific scenario and instead relied solely on prompts to drive the segmentation, which proved to be somewhat unstable. This is also why we designed an additional control panel for the user. Nevertheless, we can employ a delay strategy to capture images at different moments, thereby avoiding interference. Below are the detailed implementation steps.

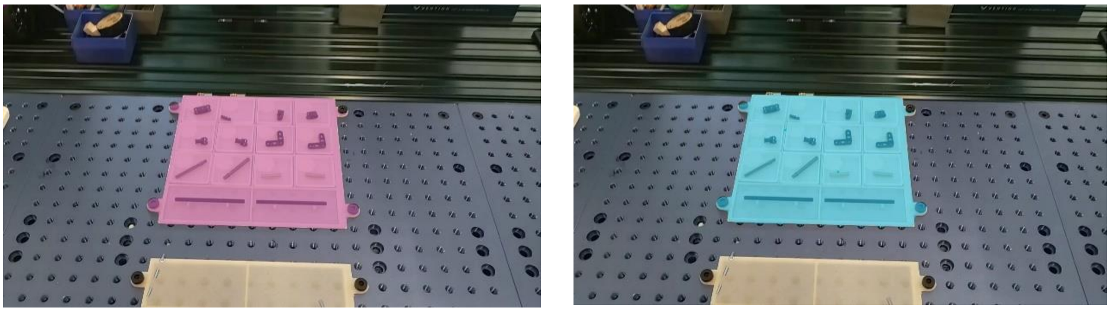
Starting with FastSAM's segmentation process, it first performs all-instance segmentation on an image (1920x1080 resolution), followed by selecting the required part based on the user-provided prompt. Therefore, the first step is to test whether FastSAM can accurately segment the tray area. The specific results are shown in Figure 7. As can be seen, in the "Segment Everything" mode, FastSAM not only segments the tray itself but also accurately segments the parts within the tray. This precise segmentation capability is exactly what we need. Building on this, both text prompts and point prompts can further segment the required tray area. However, the points used in the point prompt are fixed, which is not suitable for our scenario. Hence, we proceed with further exploration using the text prompt.



Raw Image



Segment Everything



Text Prompt ("white tray holding black mechanical parts")

Point Prompt (Two Points)

Figure 7: FastSAM Segmentation and Prompt Results

The core reason for segmenting the tray area is that it possesses distinct geometric features. Once this area is segmented, we can find the four corner points of the tray, connect them, and then divide the area evenly to obtain a grid that can be mapped to the part regions. Therefore, the first step is to locate the corner points. For this, we use the Hough transform to extract the lines. To facilitate extraction, we need to use a color that contrasts sharply with the surrounding environment to increase the difference. Here, I used orange as the tray's marker color.

At the same time, to accurately extract the lines while avoiding interference from other objects, I restricted the calculation area to the vicinity of the four edges of the mask. There are two methods to achieve this. The first method involves obtaining the upper-left corner coordinates of the mask's bounding box and using this as the starting point to divide four rectangles, each containing an edge. The second method calculates the center of the mask and uses it as the starting point to divide horizontal and vertical rectangles as the calculation area. This project implemented the first method while reserving an interface for the second method. It should be noted that the scale parameter is not fixed and any suitable value can be selected.

The process for calculating the part's area is shown in Figure 8, and it produced excellent results on the image, as shown in Figure 9 and 10.

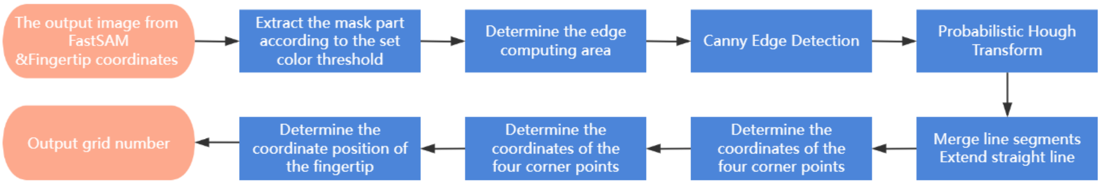

The output image from FastSAM &Fingertip coordinates
Extract the mask part according to the set color threshold
Determine the edge computing area
Canny Edge Detection
Probabilistic Hough Transform
Merge line segments Extend straight line
Determine the coordinates of the four corner points
Determine the coordinates of the four corner points
Determine the coordinate position of the fingertip
Output grid number

Figure 8: Hough Transform-Based Tray Segmentation

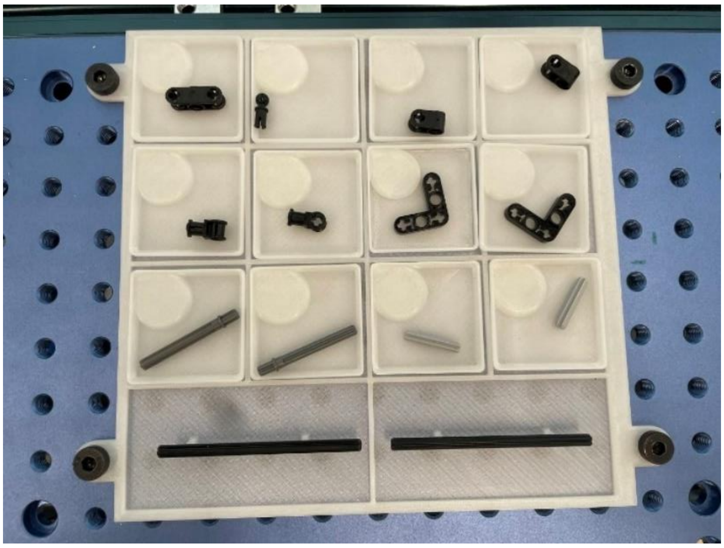


Figure 9: Initial Layout of Parts on the Tray

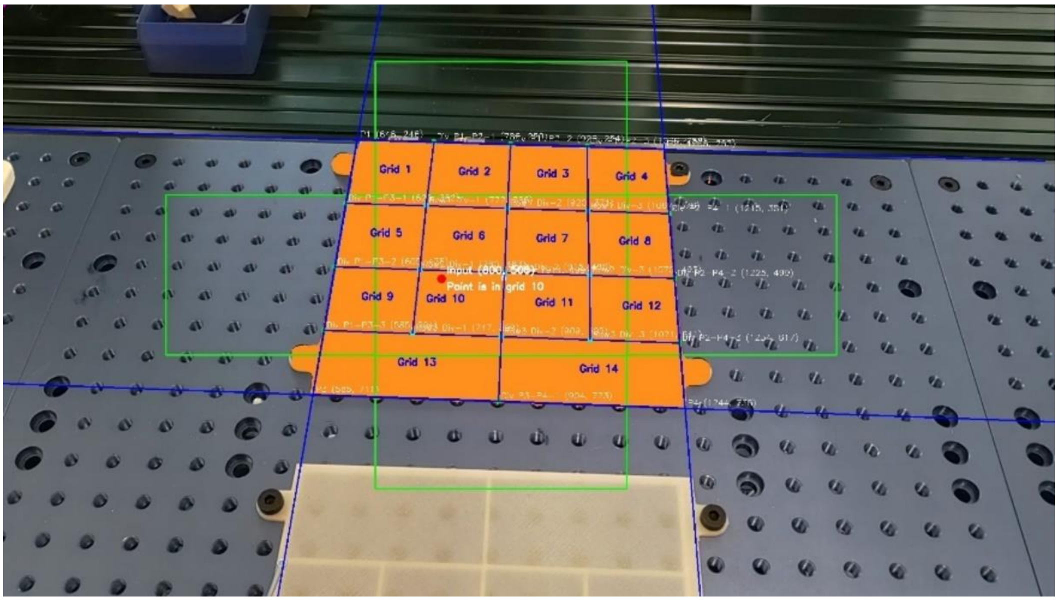


Figure 10: Gridded Segmentation of Parts

However, in actual testing, it was found that when the user's hand occludes the tray, FastSAM is unable to segment the tray, making it impossible to locate the exact position of the parts. Therefore, this project adopts a time-sharing processing strategy, where the model processes data from different time intervals separately and then merges the results to make a final decision. This approach completely avoids interference caused by the user's hand. The detailed process is shown in Figure 11.

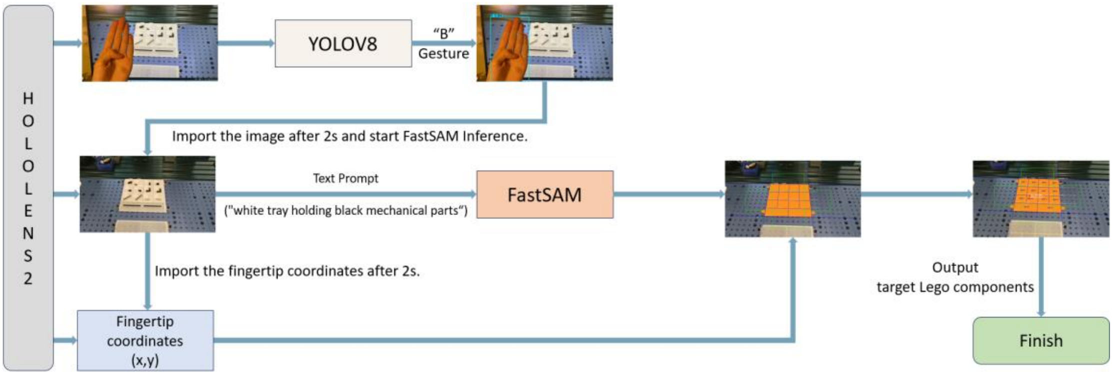

HOLOLENS2
YOLOV8
"B" Gesture
Import the image after 2s and start FastSAM Inference.
Text Prompt
("white tray holding black mechanical parts")
FastSAM
Import the fingertip coordinates after 2s.
Fingertip coordinates (x,y)
Output target Lego components
Finish

Figure 11: Part-Picking System Overview

During the project's runtime, YOLOv8 acts as a trigger for FastSAM. When the predetermined gesture ("B" gesture) is detected, the image captured two seconds later is passed to FastSAM to locate the area of each part. Two seconds after that, the fingertip coordinates are compared with the previously segmented part areas to determine the target part's location. During this period, it is crucial that the change in the image range between frames is minimized. This method achieves excellent results without the need for significant human and material resources to collect a specialized dataset for training the corresponding model.

This part of the development was carried out on a laptop, with the runtime environment based on the official open-source code of FastSAM. The configuration is as follows:

Table 11: Environment Configuration for FastSAM

Component	Specification
CPU	Intel(R) Core(TM) i7-14700HX @2.1GHZ
GPU	NVIDIA GeForce RTX 4070 Laptop GPU
Memory	32GB
Operating System	Windows 11 Home
Python	3.9.19
Pytorch	2.4.1+cu124
ultralytics	8.2.99

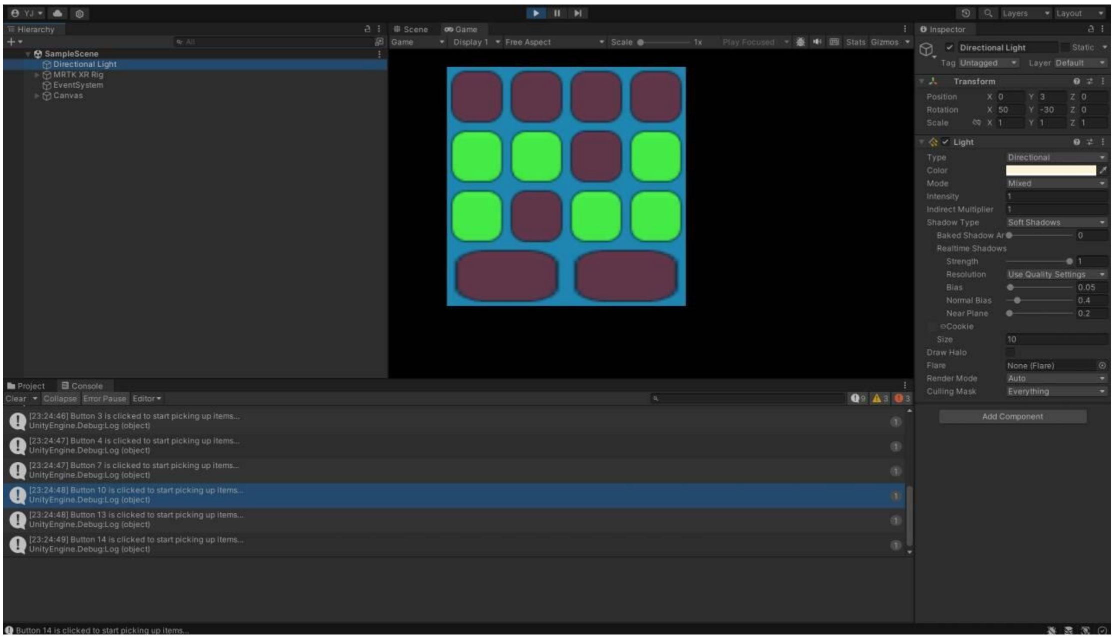
3.3.3 AR user interface

In Unity, I designed and implemented a user interface based on a Canvas(Fig.10), which maps the individual parts on the parts tray. Each part corresponds to a button, and upon clicking the button, the robotic arm will execute the picking operation according to the corresponding part number. Each button in the UI is mapped one-to-one with the actual part location, and the mapping logic has been successfully implemented.

The specific implementation is as follows:

1) **UI Design**: A button grid was designed in Unity's Canvas to represent the parts on the tray. Buttons are color-coded to indicate their status. When a button is clicked, the robotic arm picks up the corresponding part.

2) **Mapping Functionality**: The click event of each button is mapped to the robotic arm's control logic, ensuring that when a button is clicked, the robotic arm can pick up the designated part. This functionality has been tested, with corresponding logs output in Unity's debug console, confirming the success of the button clicks and operations.



```
[23:24:46] Button 3 is clicked to start picking up items...
UnityEngine.Debug:Log (object)
[23:24:47] Button 4 is clicked to start picking up items...
UnityEngine.Debug:Log (object)
[23:24:47] Button 7 is clicked to start picking up items...
UnityEngine.Debug:Log (object)
[23:24:48] Button 10 is clicked to start picking up items...
UnityEngine.Debug:Log (object)
[23:24:48] Button 13 is clicked to start picking up items...
UnityEngine.Debug:Log (object)
[23:24:49] Button 14 is clicked to start picking up items...
UnityEngine.Debug:Log (object)
Button 14 is clicked to start picking up items...
```

Figure 12: Unity User Interface for Part Selection

3.3.4 Operation information synchronization

In this project, the synchronization of operation information is crucial, as it ensures the correctness and consistency of the system across multiple modes. First, the synchronization of information between the UI interface and the gesture recognition system can prevent the same part from being picked up multiple times. For each part, regardless of whether it is triggered through button clicks or gesture recognition, it can only be picked up once. If the operation information is not synchronized, the system may attempt to pick up a part that has already been selected through the UI after recognizing a gesture, leading to duplicate operations and resource waste.

WebSocket [67] is a communication protocol that provides full-duplex, bi-directional communication channels over a single, long-lived TCP connection. It is particularly well-suited for applications requiring real-time data exchange between a client (e.g., a browser, a device like HoloLens 2) and a server. The information synchronization operation in this project is completed through WebSocket, which is mainly implemented through three code files.

HoloLens Camera Integration (HololensCamera.cs)

The HoloLens camera is used to capture real-time images and track the user's hand gestures to guide the robotic arm in picking parts. The key steps are outlined below:

1) **Camera Initialization**: The camera is initialized using the Unity PhotoCapture API, which captures images in 1920x1080 resolution. The captured images are processed to identify the user’s right-hand index fingertip coordinates, crucial for mapping the part-picking operation.

2) **Photo Capture and Processing**: The TakePhoto method is called asynchronously to capture an image, which is then processed in the OnCapturedPhotoToMemory method. The image data is uploaded to a texture, converted to a JPG format, and packaged along with fingertip coordinates.

3) **Hand Tracking**: To detect the right-hand index fingertip's position, the GetRightIndexFingerTipPosition method is used. The method utilizes HandJointUtils from the Microsoft Mixed Reality Toolkit (MRTK) to obtain the fingertip coordinates.

4) **WebSocket Communication**: Once the image and fingertip data are packaged, they are sent to a WebSocket server (ws://localhost:8765) using the SendDataOverWebSocket method. This ensures that the image data and the detected fingertip position are transferred to a server for further processing, such as robotic arm control.

5) **Continuous Operation**: The camera continues to capture images in a loop, allowing for continuous monitoring of the hand and ensuring real-time updates to the robotic arm based on the user’s hand movements.

Data Parsing on Server (data_parser.py)

The data parser on the server side is designed to handle incoming image and coordinate data sent by the HoloLens camera via WebSocket. The key functionality includes:

1) **WebSocket Listener**: The server listens for incoming data on the WebSocket connection, and once the data is received, it processes the image and coordinates.

2) **Data Separation**: The incoming data consists of both image data and fingertip position information. The parser first separates the two parts by extracting the length of the image and position data, which are sent together in a packaged format.

3) **Robotic Arm Command Generation**: Once the fingertip position is processed, the system generates commands for the robotic arm, instructing it to move to the

specified coordinates and pick the corresponding part.

Server-Side Handling (server.py)

On the server side, this script manages the WebSocket connection with the HoloLens device, handles the received data, and communicates with the robotic arm. The process involves:

1) **WebSocket Initialization**: A WebSocket server is set up to listen for incoming connections from the HoloLens. Once a connection is established, the server continuously receives data packets containing images and fingertip coordinates.
2) **Image and Coordinate Handling**: After receiving the data, the server processes the image using FastSAM and identifies the exact part of the tray where the fingertip is pointing. The fingertip's x and y coordinates are crucial for locating the correct part.
3) **Part Picking Synchronization**: The server is responsible for ensuring that each part is picked only once. If a part is selected via hand gesture or the user interface, it is marked as picked, and the system synchronizes this information to avoid duplicate picks.

It should be emphasized that in this project, the robot arm is purely an object to be acted upon, and there is no feedback mechanism, so there is no setting and synchronization operation of the robot arm state.

4. Conclusions and Critical Analysis

This project aimed to integrate Augmented Reality (AR) technology with industrial automation to create a more intuitive and efficient robotic arm control system for part-picking tasks. By leveraging Microsoft HoloLens 2, FastSAM, and YOLOv8, we designed a system where users could control the robotic arm through hand gestures or a simple user interface. The AR interface allowed for efficient part recognition and seamless interaction, which are critical for improving production line continuity and reducing the complexity of manual operations in industrial environments.

4.1 Project Summary

The project's success lies in the efficient integration of AR, gesture recognition, and image segmentation technologies. The use of FastSAM for part segmentation, combined with YOLOv8 for gesture detection, allowed for precise control of the robotic arm. The system achieved its goal of enabling the user to intuitively control the robotic arm in an industrial setting, significantly reducing the need for complex manual inputs. The multi-mode interaction, offering both gesture-based control and a UI panel, added flexibility to the system, making it adaptable to different industrial needs.

4.2 Limitations

Despite its success, the project has certain limitations:

1) **Occlusion Issues**: FastSAM's segmentation accuracy decreased when the user's hand occluded parts of the tray. Although this was mitigated by implementing a time-based processing strategy, where images were captured at different intervals, this workaround might not always be feasible in real-time scenarios.

2) **Limited Gesture Set**: The system currently uses only one gesture to trigger actions. While this reduces the risk of misrecognition, it limits the potential for more complex operations that could be performed using additional gestures.

3) **No Feedback from the Robotic Arm**: The system only provides one-way communication from the user to the robotic arm. In future iterations, a feedback loop from the robotic arm could enhance operational safety and ensure more accurate part-picking.

4.3 Potential for Expansion

1) **Gesture Set Expansion:** The system can be expanded by incorporating more gestures to trigger different commands for the robotic arm. Future iterations could also incorporate voice commands or other input modalities to further enhance user control.

2) **Improved Object Segmentation:** Fine-tuning FastSAM for specific industrial scenarios, such as when objects are occluded or partially hidden, could improve segmentation accuracy. Using a combination of data-driven models and domain-specific heuristics could further enhance this capability.

3) **Multi-Device Collaboration:** The system could be extended to support collaboration between multiple AR devices or robotic arms, enabling more complex industrial tasks and team-based workflows. This would align with the increasing trend of Industry 5.0, focusing on human-machine collaboration.

4.4 Comparison to Existing Literature

Compared to existing solutions in the field of AR-driven industrial automation, this project provides a unique approach by integrating a lightweight segmentation model (FastSAM) and a real-time gesture recognition model (YOLOv8) into one system. The combination of AR for real-time control with advanced AI models for part recognition and gesture detection is still relatively novel. Existing literature often focuses on one of these aspects, either improving AR interfaces or refining gesture recognition, but seldom integrates these into a cohesive industrial workflow as demonstrated in this project.

4.5 Practical Relevance and Potential

The system has significant potential to address real-world industrial challenges. In manufacturing environments where the variety of parts is large and manual part-picking is prone to error, this system offers a practical solution that reduces human error and increases efficiency. By removing the need for manual data collection and custom model training, the system reduces the burden on operators while maintaining flexibility in dynamic production settings.

In conclusion, this project successfully demonstrated the potential for AR technology, combined with AI models, to solve real-world industrial automation problems. With improvements in occlusion handling, expanded gesture sets, and multi-device collaboration, this system could be a valuable tool in the future of smart manufacturing.

REFERENCES

[1] Milgram, Paul, et al. "Augmented reality: A class of displays on the reality-virtuality continuum." Telemanipulator and telepresence technologies. Vol. 2351. Spie, 1995.

[2] Cipresso, Pietro, et al. "The past, present, and future of virtual and augmented reality research: a network and cluster analysis of the literature." Frontiers in psychology 9 (2018): 2086.

[3] Seeliger, Arne, Raphael P. Weibel, and Stefan Feuerriegel. "Context-adaptive visual cues for safe navigation in augmented reality using machine learning." International Journal of Human–Computer Interaction 40.3 (2024): 761-781.

[4] Lu, Xueshi, et al. "itext: Hands-free text entry on an imaginary keyboard for augmented reality systems." The 34th Annual ACM Symposium on User Interface Software and Technology. 2021.

[5] Wysopal, Abby, et al. "Level-of-detail ar: Dynamically adjusting augmented reality level of detail based on visual angle." 2023 IEEE Conference Virtual Reality and 3D User Interfaces (VR). IEEE, 2023.

[6] Seeliger, Arne, Long Cheng, and Torbjørn Netland. "Augmented reality for industrial quality inspection: An experiment assessing task performance and human factors." Computers in Industry 151 (2023): 103985.

[7] Tu, Xinyi, et al. "TwinXR: Method for using digital twin descriptions in industrial eXtended reality applications." Frontiers in Virtual Reality 4 (2023): 1019080.

[8] Morales Mojica, Cristina M., et al. "A holographic augmented reality interface for visualizing of MRI data and planning of neurosurgical procedures." Journal of Digital Imaging 34 (2021): 1014-1025.

[9] Pfeuffer, Ken, et al. "ARtention: A design space for gaze-adaptive user interfaces in augmented reality." Computers & Graphics 95 (2021): 1-12.

[10] Lei, Xin, Yueh-Lin Tsai, and Pei-Luen Patrick Rau. "Harnessing the visual salience effect with augmented reality to enhance relevant information and to impair distracting information." International Journal of Human–Computer Interaction 39.6 (2023): 1280-1293.

[11] Kolla, Sri Sudha Vijay Keshav, Andre Sanchez, and Peter Plapper. "Comparing software frameworks of augmented reality solutions for manufacturing." Procedia Manufacturing 55 (2021): 312-318.

[12] Filippidis, Achilleas, et al. "VR Isle Academy: A VR Digital Twin Approach for Robotic Surgical Skill Development." arXiv preprint arXiv:2406.00002 (2024).

[13] Šarić, Matko, et al. "Extended reality telemedicine collaboration system using patient avatar based on 3D body pose estimation." Sensors 24.1 (2023): 27.

[14] Pavlou, Michail, et al. "XRSISE: An XR training system for interactive simulation and ergonomics assessment." Frontiers in Virtual Reality 2 (2021): 646415.

[15] Sardar, Suman Kalyan, et al. "Ergonomic risk assessment of manufacturing works in virtual reality context." International Journal of Human–Computer Interaction 40.14 (2024): 3856-3872.

[16] Chang, Eunhee, Yongjae Lee, and Byounghyun Yoo. "A user study on the comparison of view interfaces for VR-AR communication in XR remote collaboration." International Journal of Human–Computer Interaction (2023): 1-16.

[17] Unity Technologies. (2022). Unity Manual. Retrieved from https://docs.unity3d.com/Manual/index.html

[18] Epic Games. (2022). Unreal Engine Documentation. Retrieved from https://docs.unrealengine.com/en-US/index.html

[19] Apple Inc. (2022). ARKit. Retrieved from Augmented Reality - Apple Developer

[20] Google LLC. (2022). ARCore. Retrieved from https://developers.google.com/ar

[21] Microsoft Corporation. (2020). Windows Mixed Reality. Retrieved from https://docs.microsoft.com/en-us/windows/mixed-reality/

[22] Abidi, Mustufa Haider, et al. "Assessment of virtual reality-based manufacturing assembly training system." The International Journal of Advanced Manufacturing Technology 105.9 (2019): 3743-3759.

[23] Ortega-Gras, Juan-José, et al. "Designing a technological pathway to empower vocational education and training in the circular wood and furniture sector through extended reality." Electronics 12.10 (2023): 2328.

[24] Mourtzis, Dimitris, and John Angelopoulos. "Development of an Extended Reality-Based Collaborative Platform for Engineering Education: Operator 5.0." Electronics 12.17 (2023): 3663.

[25] Ulmer, Jessica, et al. "Gamified virtual reality training environment for the manufacturing industry." 2020 19th international conference on mechatronics-mechatronika (ME). IEEE, 2020.

[26] Virtual Reality-Based Engineering Education to Enhance Manufacturing Sustainability in Industry 4.0

[27] Aschauer, Andrea, Irene Reisner-Kollmann, and Josef Wolfartsberger. "Creating an open-source augmented reality remote support tool for industry: challenges and learnings." Procedia Computer Science 180 (2021): 269-279.

[28] Mourtzis, Dimitris, Vasileios Siatras, and John Angelopoulos. "Real-time remote maintenance support based on augmented reality (AR)." Applied Sciences 10.5 (2020): 1855.

[29] Cavaleri, Joey, et al. "Remote video collaboration during COVID-19." 2021 32nd Annual SEMI Advanced Semiconductor Manufacturing Conference (ASMC). IEEE, 2021.

[30] Masoni, Riccardo, et al. "Supporting remote maintenance in industry 4.0 through augmented reality." Procedia manufacturing 11 (2017): 1296-1302.

[31] Abidi, Mustufa H., et al. "Semi-immersive virtual turbine engine simulation system." International Journal of Turbo & Jet-Engines 35.2 (2018): 149-160.

[32] Lupinetti, Katia, et al. "Exploring the benefits of the virtual reality technologies for assembly retrieval applications." International conference on augmented reality, virtual reality and computer graphics. Cham: Springer International Publishing, 2019.

[33] Ratava, J., et al. "Quality assurance and process control in virtual reality." Procedia Manufacturing 38 (2019): 497-504.

[34] Szajna, Andrzej, et al. "The production quality control process, enhanced with augmented reality glasses and the new generation computing support system." Procedia computer science 176 (2020): 3618-3625.

[35] Alves, Joao Bernardo, et al. "Using augmented reality for industrial quality assurance: a shop floor user study." The International Journal of Advanced Manufacturing Technology 115.1 (2021): 105-116.

[36] Serras, Manex, et al. "Dialogue enhanced extended reality: Interactive system for the operator 4.0." Applied Sciences 10.11 (2020): 3960.

[37] Prati, Elisa, et al. "An approach based on VR to design industrial human-robot collaborative workstations." Applied Sciences 11.24 (2021): 11773.

[38] Yang, Chao, et al. "Extended reality application framework for a digital-twin-based smart crane." Applied Sciences 12.12 (2022): 6030.

[39] Peruzzini, Margherita, et al. "Using virtual manufacturing to design human-centric factories: an industrial case." The international journal of advanced manufacturing technology 115.3 (2021): 873-887.

[40] Zimmerer, Chris, et al. "Reducing the cognitive load of playing a digital tabletop game with a multimodal interface." Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems. 2022.

[41] Xia, Guobin, et al. "A comparative study of colour effects on cognitive performance in real-world and VR environments." Brain sciences 12.1 (2021): 31.

[42] Zhang, Mingming, Guanhua Hou, and Yeh-Cheng Chen. "Effects of interface

layout design on mobile learning efficiency: a comparison of interface layouts for mobile learning platform." Library hi tech 41.5 (2023): 1420-1435.

[43] Zhou, Jianlong, Simon Luo, and Fang Chen. "Effects of personality traits on user trust in human–machine collaborations." Journal on Multimodal User Interfaces 14.4 (2020): 387-400.

[44] Bellman, Steven, and Kyle B. Murray. "Feedback, task performance, and interface preferences." European Journal of Information Systems 27.6 (2018): 654-669.

[45] Han, Jining, Qiyu Zheng, and Yunan Ding. "Lost in virtual reality? Cognitive load in high immersive VR environments." Journal of Advances in Information Technology 12.4 (2021).

[46] Caserman, Polona, et al. "Cybersickness in current-generation virtual reality head-mounted displays: systematic review and outlook." Virtual Reality 25.4 (2021): 1153-1170.

[47] Carnegie, Kieran, and Taehyun Rhee. "Reducing visual discomfort with HMDs using dynamic depth of field." IEEE computer graphics and applications 35.5 (2015): 34-41.

[48] Kim, Eunjee, and Gwanseob Shin. "User discomfort while using a virtual reality headset as a personal viewing system for text-intensive office tasks." Ergonomics 64.7 (2021): 891-899.

[49] Sarig-Bahat, Hilla, Patrice L. Tamar Weiss, and Yocheved Laufer. "Neck pain assessment in a virtual environment." Spine 35.4 (2010): E105-E112.

[50] Mehra, Divy, and Anat Galor. "Digital screen use and dry eye: a review." Asia-Pacific journal of ophthalmology 9.6 (2020): 491-497.

[51] Shibata, Takashi, et al. "The zone of comfort: Predicting visual discomfort with stereo displays." Journal of vision 11.8 (2011): 11-11.

[52] Lee, Chun-Chia, Hsiu-Sen Chiang, and Meng-Hsing Hsiao. "Effects of screen size and visual presentation on visual fatigue based on regional brain wave activity." The journal of supercomputing 77 (2021): 4831-4851.

[53] Nishiike, Suetaka, et al. "The effect of visual-vestibulosomatosensory conflict induced by virtual reality on postural stability in humans." The Journal of Medical Investigation 60.3.4 (2013): 236-239.

[54] Park, SoHu, and GyuChang Lee. "Full-immersion virtual reality: Adverse effects related to static balance." Neuroscience letters 733 (2020): 134974.

[55] Li, Chunping, et al. "Cognitive load measurement in the impact of VR intervention in learning." 2022 International Conference on Advanced Learning Technologies (ICALT). IEEE, 2022.

[56] Xiageyiqi Mun, Sungchul, Eun-Soo Kim, and Min-Chul Park. "Effect of mental fatigue caused by mobile 3D viewing on selective attention: An ERP study." International Journal of Psychophysiology 94.3 (2014): 373-381.

[57] Zeuwts, Linus HRH, et al. "Mental fatigue delays visual search behaviour in young cyclists when negotiating complex traffic situations: A study in virtual reality." Accident Analysis & Prevention 161 (2021): 106387.

[58] Zhang, Yunxiang, Kenneth Chen, and Qi Sun. "Toward Optimized VR/AR Ergonomics: Modeling and Predicting User Neck Muscle Contraction." ACM SIGGRAPH 2023 Conference Proceedings. 2023.

[59] Callie Holderman, Eakta Jain, Michael Running Wolf, and Liv Erickson. 2022. Privacy, Safety and Wellbeing: Solutions for the Future of AR and VR. In ACM SIGGRAPH 2022 Panels (SIGGRAPH '22). Association for Computing Machinery, New York, NY, USA, Article 1, 1–2. https://doi.org/10.1145/3532718.3535620

[60] Maurice, Pauline, et al. "Ethical and social considerations for the introduction of human-centered technologies at work." 2018 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO). IEEE, 2018.

[61] Indiparambil, Jijo James. "Privacy and beyond: Socio-ethical concerns of ‘on-the-job’surveillance." Asian Journal of Business Ethics 8.1 (2019): 73-105.

[62] Jocher, Glenn, et al. "YOLOv8: Real-Time Object Detection." GitHub, 2023, https://github.com/ultralytics/ultralytics.

[63] Zhao, Xu, et al. "Fast segment anything." arXiv preprint arXiv:2306.12156 (2023).

[64] Kirillov, Alexander, et al. "Segment anything." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023.

[65] Radford, Alec, et al. "Learning transferable visual models from natural language supervision." International conference on machine learning. PMLR, 2021.

[66] Christ University. "Multiple-Sign-Language-Detection Dataset." Roboflow Universe, Roboflow, July 2024, https://universe.roboflow.com/christ-university-ilp52/multiple-sign-language-detection. Accessed 15 Sept. 2024.

[67] McGee, Aymeric Augustin. "websockets: A Python WebSocket Library." GitHub, https://github.com/aaugustin/websockets.