Master Thesis
In Master´s Program
International Entrepreneurship, Digitalization and Sustainability (IEDS)
at the Neu-Ulm University of Applied Sciences

AI driven Augmented Reality Assistance for Manual Pre-Assembly

First Proofreader: Prof. Dr. Tobias Engel

Second Proofreader: Prof. Dr. Jürgen Grinninger

Supervisor: Jonathan Kreßel

Author: Jan Diebold (Matriculation-Nr.: 375425)

Topic received: 13.12.2025

Work submitted: 24.02.2026

Contents

Abstract

Manual assembly of complex technical products presents recurring challenges for manufacturing organizations: assembly errors caused by paper-based work instructions, high onboarding effort for new operators and limited digital traceability of production processes. This thesis investigates how artificial intelligence in combination with augmented reality smart glasses can enhance manual pre-assembly in a smart factory environment from a technical perspective. Following a Design Science Research methodology, the work develops and evaluates a proof of concept that integrates AI-driven object detection with AR-based pick-by-vision guidance on a commercially available Meta Quest 3 headset.

The artifact addresses two design goals defined in collaboration with the Technology Transfer Centre Leipheim. The first goal is to replace a conventional pick-by-light assistance with spatially anchored AR work instructions that guide the operator hands-free through an 11-step drone assembly workflow. With the inclusion of integrating wireframe bin highlights, three-dimensional component previews and a multi-modal instruction panel. The second goal introduces an AI-based controlling mechanism through a custom-trained object detection model that verifies each pick action via the headset camera and prevents workflow advancement until the correct component has been positively identified. The system architecture is built on MQTT, an ISO-standardized industrial messaging protocol, ensuring compatibility with existing manufacturing infrastructure. A self-proposed modular three-layer conception separates the AR frontend, communication layer and AI backend, enabling independent scaling, platform transferability and future integration with enterprise systems.

The contribution of this work is architectural and technical. It demonstrates the feasibility of combining spatial worker guidance and AI-driven quality verification into a single application on cost-effective commercial hardware. A combination that existing literature has predominantly investigated only as separate systems on separate devices. The operational benefits described in the thesis represent design properties of the architecture. Empirical validation through controlled user studies is identified as the highest-priority direction for future research, alongside eight follow-up proposals addressing, like wearable edge inference, smart glasses adaptation and enterprise system integration.

List of Abbreviations

Abbreviation	Full Term
AI	Artificial Intelligence
AR	Augmented Reality
ARAGT	AR Assembly Guidance Tool
ARSG	Augmented Reality Smart Glasses
ASF	Adaptive Smart Factory
AV	Augmented Virtuality
BOM	Bill of Materials
BTFs	Bidirectional Texture Functions
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
CV	Computer Vision
DoF	Degrees of Freedom
DSR	Design Science Research
DT	Digital Twin
EEG	Electroencephalography
EPM	Extent of Presence Metaphor
EWM	Enterprise Warehouse Management
EXAI	Explainable Artificial Intelligence
fps	Frames Per Second
GANs	Generative Adversarial Networks
HDT	Human Digital Twin
HMDs	Head-Mounted Displays
HNU	Hochschule Neu-Ulm
HRC	Human-Robot Collaboration
IDE	Integrated Development Environment
IMU	Inertial Measurement Units
JSON	JavaScript Object Notation
LRMs	Large Reconstruction Models
mAP	Mean Average Precision
ML	Machine Learning
MLP	Multi Level Perspective
MQTT	Message Queuing Telemetry Transport
MR	Mixed Reality
NeRF	Neural Radiance Fields
ODT	Operator Digital Twin
OVD	Open-Vocabulary Detectors
PSO	Particle Swarm Optimization
R-CNN	Region-based Convolutional Neural Network
RF	Reproduction Fidelity
RV	Reality-Virtuality
SLR	Systematic Literature Review
SPP	Spatial Pyramid Pooling
SSL	Secure Sockets Layer
TLS	Transport Layer Security

TTZ	Technologie-Transfer-Zentrum
UAVs	Unmanned Aerial Vehicles
VR	Virtual Reality
XR	Extended Reality
YOLO	You Only Look Once

List of Illustrations

List of Tables

Appendix Overview

Appendix Number	File Name	Contents
Appendix A	Market Screen Device Evaluation Table	Outline of Holistic comparison of VR Headsets and Smart Glasses ranging from low budget commercial to high-end industrial devices.
Appendix B	Overview AI Tree	Categorization of AI Technologies in the area of object detection
Appendix C	Initial Work Instruction Drone Pre-Assembly	Showing the current paperwork instruction at the workstation in a digital text file.
Appendix D	LowLevel_Architecture	Showing a low-level architecture version of the created conceptual artifact
Appendix E	MidLevel_Architecture	Showing a mid-level architecture version of the created conceptual artifact
Appendix F	Holistic_IT_Architecture	Showing a high-level detailed architecture version of the created conceptual artifact
Appendix G	Appendix G Artifact Doc without Unity files	Prototype C#, Python, Documentation Scripts without the Unity AR Project
Appendix H	DemoVideo_SplitView	Demonstration Video with Operator and Object Detection View
Appendix I	DemoVideo_OperatorView	Demonstration Video with Operator View

AI Use

Applying the Guidelines for the Use of AI Tools in Master Thesis/Seminar Paper of the faculty information management.

Purpose

AI-supported tools were used selectively while writing this master's thesis. They were used exclusively as supporting aids in clearly defined areas, in the structural organization of technical concepts and in the development and debugging of software artifacts that were created in the practical part of the thesis. The tools were used to increase efficiency in recurring auxiliary tasks and to ensure the quality of the developed scripts, but not to replace own scientific analysis or argumentation. The AI was used as a technical critical assessment partner for testing scripts and reviewing scripts. Academic independence was ensured through several measures. Every AI-generated suggestion was critically reviewed against the author's own understanding of the system and the research context. No AI output was incorporated without modification and verification. The conceptual framework, research design, system architecture, implementation and all evaluative conclusions were developed independently by the author.

Tools used

Tool	Model	Provider	Usage
Claude Code	Claude-sonnet-4.6	Anthropic	Prototyping enhancement and support for complex technical software development kits and developer samples including debugging and technical understanding.

Table 1 AI tool usage during the thesis

Reflection

The use of AI tools has supported the work in several ways. In the area of software development, the use of Claude Code enabled faster identification of threading issues and API incompatibilities that would otherwise have had to be resolved through time-consuming manual searches of documentation. One challenge was that AI-generated answers in the technical field could not be accepted uncritically, particularly in the case of SDK-specific questions. The answers were sometimes outdated or incomplete and always had to be verified against official documentation.

No confidential or sensitive data, including proprietary information from the Technology Transfer Centre Leipheim, personal data or unpublished research findings from collaborating partners, were entered into the AI system.

1 Introduction

Since many tasks in discrete industries are still either hard to automate with conventional industrial robots, also not economically feasible to automate or not yet adaptable for humanoid robots, real humans have to do specific tasks. To support the human operator, several different assistance approaches appear to guide the worker between the working steps. Classical methods like paper instructions are stated as old fashioned and ineffective compared to methods like three-dimensional visualization and dynamic explanations to each specific tasks (Sahu et al., 2021, p. 4909).

In this thesis a use case at a manual workstation of a drone pre-assembly has been used with which the general goal is to technically explore which technical opportunities there are to possibly improve the operators' work regarding efficiency and quality with new technologies and approaches. Two broad technologies have been selected, to be applied, by the stakeholders of the Technology-Transfer-Centre Leipheim, Augmented Reality and Artificial Intelligence. Additionally, a Meta Quest 3 device was provided, to explore technical feasibilities on the device´s hardware and software.

The following provides an overview of the current industry situation regarding augmented reality in manufacturing and augmented reality devices on the current market. Afterwards the goal setting of the thesis is set and the overview structure is presented, before the literature deep dive begins.

1.1 Current Industry Situation

While many use cases have been developed in laboratory environments, the use of Smart Glasses and VR Headsets have not had its breakthrough in the industry. While approximately only 32% of manufacturing companies in the discrete and process industry actually implemented XR somehow, the remaining 68% somehow indicate some level of consideration (ABI, 2024). The reasons for not implementing XR Technologies are diverse. They range from technical implementations to high procurement costs to discomfort for the user. Several papers criticize the usability of smart glasses (Rejeb et al., 2021, pp. 3760–3762).

Even though the authors use the term smart glasses, they actually use headsets, like the HoloLens2 or the Meta Quest 3. In current literature, these headsets are not considered smart glasses or Augmented Reality Smart Glasses (ARSG) anymore. "The goal should be that ARSG should be "like ordinary glasses" so that they can be used for long periods." (Danielsson et al., 2021, p. 104909). This leads to a wording conflict between literature in the past and current literature.

For Syberfeldt et al. there are three main pain points, that make the headsets not wearable for full shifts every day on the shopfloor (Syberfeldt et al., 2017, p. 9129). These three points can be found in other papers too (Rejeb et al., 2021, p. 3762; Windhausen et al., 2024, p. 9). First is the weight, with around 400g (Appendix A), the devices lead to neck pain and marks on the face (Danielsson et al., 2021, p. 104909; Rejeb et al., 2021, p. 3762;

Windhausen et al., 2024, p. 4). Second, many devices need a cable to run properly, this hinders the operator to move freely (Syberfeldt et al., 2017, p. 9129). The headsets are not useable for people wearing regular glasses. They can´t be combined and vision impairment is not corrected by the headset itself (Syberfeldt et al., 2017, p. 9129). In addition the users avoided longer use due to the battery life of 2-3h (Windhausen et al., 2024, p. 4) but also due to possible side effects like fatigue, lack of concentration or eyestrain (Rejeb et al., 2021, p. 3762; Windhausen et al., 2024, p. 9). Figure 1 shows a general overview of challenges implementing Smart Glasses.

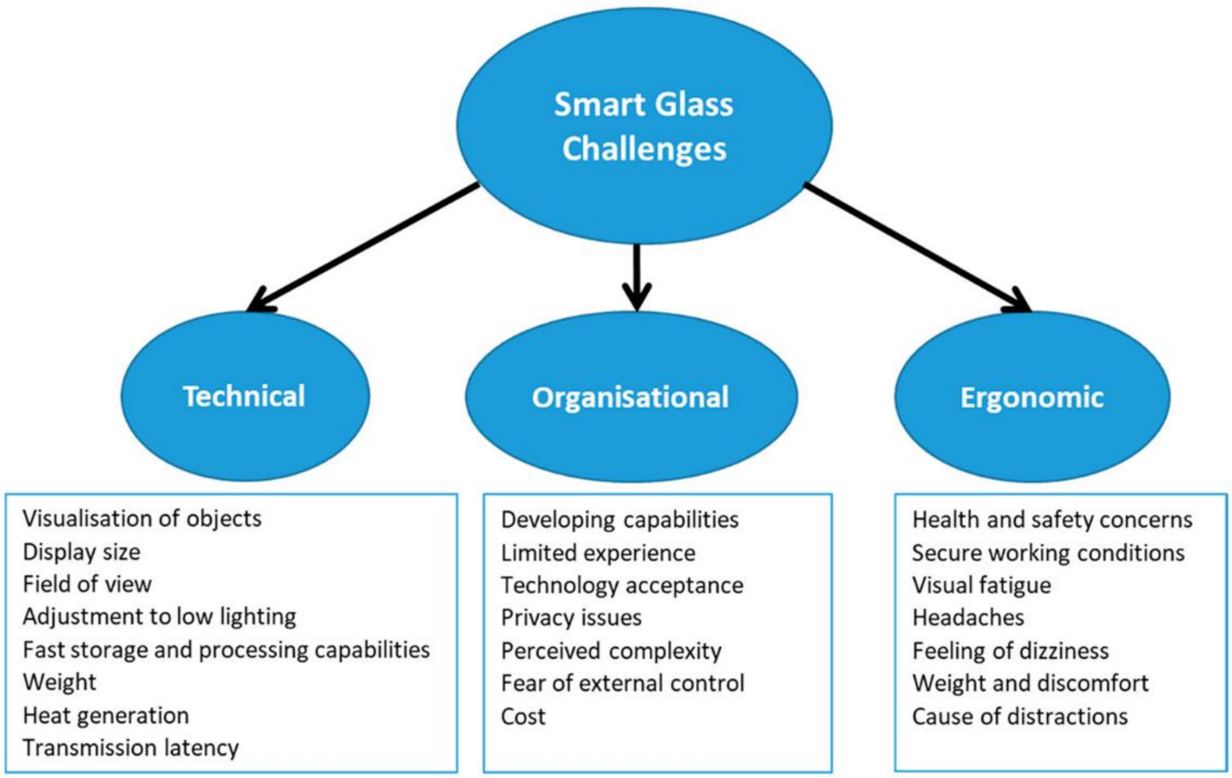


Figure 1 technical challenges using AR in manufacturing industry (Rejeb et al., 2021, p. 3760)

As smart glasses become more commercial on the mainstream market and more companies penetrate the market segment, more opportunities for using light weight models appear. Appendix A presents an overview of current devices, which range from headsets exhibiting the previously discussed limitations to products that have resolved these issues. These devices are the smart glasses, with lightweight design, see-through glasses and overall glasses instead of bulky headset look. One example would be the Glasses from the company XReal which are lightweight and can be ordered with ground lenses (*XREAL Project Aura*, 2025). During the research it has been remarkable that many companies published developer devices over the last two years, in order to gain feedback and experience on the hardware and software. At the end of the year 2025, many developer devices are not produced anymore while different models from XReal (*XREAL*

Project Aura, 2025) or Samsung (Chip.de, 2025) will be published somewhen in 2026, which month or quarter they will be released, has not been published due the date of research.

1.2 Target Setting Research Gap

Facing this situation, the Technology Transfer Centre Leipheim seeks to explore the practical applicability of current AR and AI technologies for manual assembly through targeted proof-of-concept development. The original project plan envisioned deployment on an XREAL smart glasses device, a lightweight, optical see-through glasses design well suited to industrial wear. However, due to cost and time constraints, this device was replaced with the Meta Quest 3, shifting the hardware platform from lightweight smart glasses to a significantly heavier mixed reality headset. While this change affects the ergonomic profile of the system, the underlying research approach, research question and methodology remain unchanged. The work continues to pursue the same investigative goals on a different device, with the understanding that the architectural and conceptual findings are intended to be transferable across platforms and devices. Keeping in mind that it should be applicable on lightweight device designs.

To operationalize this investigation, the following research question has been formulated: "How can AI in combination with augmented reality smart glasses enhance manual pre-assembly applications in a smart factory?" In collaboration with the Technology Transfer Centre Leipheim, two concrete improvement goals were defined to guide the project. The first goal is to provide the operator with a pick-by-vision system that replaces the pick-by-light assistance, currently deployed at the workstation, delivering spatially anchored visual guidance, directly within the operator's field of view. The second goal is to implement a controlling mechanism based on computer vision that verifies whether the operator has retrieved the correct component before the assembly workflow is permitted to advance. Both goals are to be pursued on the basis of the current technological state of the art.

A three-step methodology was defined to reach these goals. The first step consists of a structured exploration of the current technological landscape, combining a systematic literature review using the methods described in chapter 2 with a market analysis of available hardware platforms (Appendix A). Building on the knowledge gained from this research, the second step develops an architectural concept for the practical application of the identified technologies on the real pre-assembly workstation. The third step, the implementation of this concept as a functioning proof of concept, was declared optional from the outset due to time, licensing and hardware cost constraints. The resulting proof of concept is intended to be evaluated by stakeholders external to the thesis and depending on its assessed effectiveness, further developed beyond the scope of the present work.

The research gap that this work addresses lies in the combination of these two methods on a single commercial device. Existing literature has predominantly investigated AR-based worker guidance and computer vision-based component verification as separate applications, typically deployed on different hardware. Industrial AR headsets such as the

Microsoft HoloLens 2 for spatial instruction delivery and external camera systems with connected displays for object recognition. The integration of both capabilities into a single application running on commercially available, cost-effective hardware, such as the Meta Quest 3 or comparable devices, has not been demonstrated in the reviewed literature. This separation reflects a broader challenge recognized by manufacturing practitioners. Neither on the hardware side nor on the software side does a standardized, integrated solution currently exist for operational use. Organizations instead face a fragmented landscape of partial approaches, each addressing only one dimension of the assembly assistance problem.

The present work targets this gap by exploring the current state of the art in both domains and applying the findings to an artifact that combines pick-by-vision guidance with AI-driven object detection on a single headset platform. The contribution of this thesis is architectural and technical in nature. It demonstrates the feasibility of building an end-to-end AR-AI assembly assistance system on real, commercially available hardware and provides a documented, reproducible system design that integrates both capabilities within a modular, standards-based architecture. The contribution does not extend to empirical proof of productivity gains, this validation is explicitly deferred to subsequent evaluation by external stakeholders and the follow-up research projects proposed in chapter 9.

1.3 Thesis Structure

The Thesis is structured in chapters, which range from the already stated problem definition to a concrete recommendation for action in form of a concept and an artifact in form of a proof of concept that can be used and further developed. Following provides a summary for each chapter.

Chapter 1 introduces the problem statement, research question and general industry situation
Chapter 2 explains the research designs which has been used through the whole thesis
Chapter 3 shows the current state of literature, including current technological levels of software, hardware and use cases.
Chapter 4 presents the industry scenario of the TTZ Leipheim
Chapter 5 presents a comprehensive concept in a theoretical and practical approach different design opportunities
Chapter 6 presents the execution and its value proposition of the concept in form of an artifact used in laboratory environment
Chapter 7 verifies the design result regarding stated research question and design goals
Chapter 8 presents limitations of the current design - including hardware and software aspects
Chapter 9 prospects follow-up projects to overcome current limitations
Chapter 10 summary with conclusion

Table 2 Chapter overview with short descriptions

2 Research Design Methodology

The research design, as already stated in the introduction, is separated into three phases of the thesis. The first phase handles the problem statement with the method of systematic literature review. Here, typical pain points of the industry are analyzed and compared to solutions that the current state of literature provides. In here also current limits of literature and possible new research fields are presented. The second phase uses the Design Science method to present a concept on how the current research gap can be filled. Proposing a conceptual solution on how the research question of the problem statement in phase one can be answered. The last phase proposes an artifact of the conceptualized solution of phase two, a prototype that has been developed and released in a laboratory environment. Table 1 visualizes the methodology of the used research designs.

RESEARCH PHASE	RESEARCH QUESTION	METHODOLOGICAL APPROACH	EXPECTED RESULTS	TIMELINE
PROBLEM STATEMENT Current industry situation	How can AI in combination with Augmented Reality enhance manual Pre-Assembly Applications in a Smart Factory?	Systematic Literature Review	• Identification of current XR/AI assistance in smart factories • Analysis of limitations in existing solutions regarding hardware and software	November - December
DESIGN AND DEVELOPEMENT Concept creation	How could a concept of current technical assistance look like?	Design Science	• Conceptual model for AI-enhanced AR • Architecture draft (hardware, software, data flow) • Prototype design specifications	January
DEMONSTRATION/ PROTOTYPING Concept execution Outlook	How can the concept be implemented and fulfil technical requirements in a real assembly station?	Design Science	• Presenting a functional prototype • Validation of concept effectiveness Technical evaluation	February

Table 3 Research Design

The following chapters explain the methodology of the used research methods. For the research phase of the problem statement, a systematic literature review has been used, based on the methodology of Webster and Watson (Webster & Watson, 2002). Design and Development, same as the Prototyping Phase, are based on the Design science methodology of Hevner et al (Hevner et al., 2004).

2.1 Systematic Literature Review

This thesis follows the systematic literature review (SLR) methodology proposed by Webster and Watson (Webster & Watson, 2002), which emphasizes conceptual rigor, transparency and replicability in building a theoretical foundation. In chapter 3 the process began with a comprehensive multiple times improved keyword-based search across major scientific databases provided by the University of Applied Sciences of Neu-Ulm. Due to the restriction of this precise topic, the keywords have been applied only to the title of the paper. Including the keywords into the abstract search or the text search led to hundreds of results, that did not even hit the research topic correctly. Search terms were derived from three conceptual domains, Extended Reality (XR) technologies coupled with artificial intelligence (AI) techniques and manufacturing. The whole and finalized search query is presented in the following. To ensure the inclusion of relevant and high-quality sources, mainly peer-reviewed journal articles and conference papers published between 2020 and 2025 were considered. Following Webster and Watson's iterative approach, a forward and backward search was performed. Backward searching involved reviewing the references of key papers, while forward searching identified newer studies that cited those key papers. The collected literature was then coded and analyzed conceptually to identify recurring themes, theoretical models, technologies and application domains. This structured synthesis enabled the identification of research clusters, gaps and emerging trends in the integration of AI and XR within smart manufacturing environments. The rigorous adherence to Webster and Watson's methodology ensures both comprehensiveness and theoretical depth, providing a replicable foundation for the subsequent design science research phase. (Webster & Watson, 2002)

Due to the dynamic development of software applications and software providers in this area, a PRISMA method is not applicable. Too many papers and technical documentations, based on the same goals or even technology, use different terms for the same things, which lead to a brought research. As already mentioned in the introduction, one example are the "Smart Glasses" of the past years, which actually have been bulky VR headsets and only in the last two years real smart glasses entered the market. A basic search query for the basis of the literature research, that has been used, is provided in the following.

Search Query:

TITLE (

("augmented reality" OR "virtual reality" OR "mixed reality" OR "extended reality" OR XR OR MR OR AR OR VR OR "smart glasses" OR "data glasses" OR "industrial glasses" OR "head-mounted display" OR HMD OR wearable*)

AND

(assembly OR "manual assembly" OR "manufacturing" OR "production line" OR "operator support" OR "worker assistance")

AND

("artificial intelligence" OR "machine learning" OR "deep learning" OR "reinforcement learning" OR "computer vision" OR "digital twin" OR "knowledge graph" OR "large language model" OR "generative AI" OR "AI agent" OR "multi-agent system" OR "natural language processing" OR "cyber-physical system" OR "predictive analytics" OR "smart factory" OR "smart manufacturing" OR "industry 4.0" OR "industry 5.0"))

2.2 Design Science

The design phase of this research follows the Design Science Research (DSR) methodology established by Hevner et al. (Hevner et al., 2004), which emphasizes the creation and evaluation of innovative artifacts to address relevant organizational and technological problems. Within this paradigm, knowledge is advanced through the building and application of artifacts. Including architectures, models, methods and instantiations, that provide purposeful solutions to identified challenges. In alignment with Hevner's framework, the research process is guided by the relevance, rigor and design cycles. The relevance cycle connects the study to the practical problem context of smart manufacturing, ensuring that the artifact, a concept and prototype for AI-enhanced Augmented Reality (AR) in manual assembly, is developed to meet industrial needs. The rigor cycle ensures that existing scientific foundations, kernel theories and methodological standards from prior literature are conducted by the prior systematic literature review and applied in the practical work to guarantee validity and reproducibility. The design cycle constitutes the iterative process of building, demonstrating and evaluating the artifact through systematic refinement. Consistent with Hevner's seven design guidelines, the research (1) produces a viable artifact, (2) addresses a relevant industrial problem, (3) evaluates artifact utility and performance, (4) contributes new design knowledge, (5) applies rigorous methods, (6) employs a structured search process for optimal solutions and (7) communicates the outcomes effectively to both academic and practitioner audiences. This methodological approach ensures that the developed AI–XR concept not only demonstrates technical feasibility and utility but also contributes to the theoretical knowledge base of design-oriented research in smart manufacturing. (Hevner et al., 2004)

To visualize the process, figure 2 shows the development, evaluation loop of Marino et al. which is based on Hevner et al. and has been applied through the project.

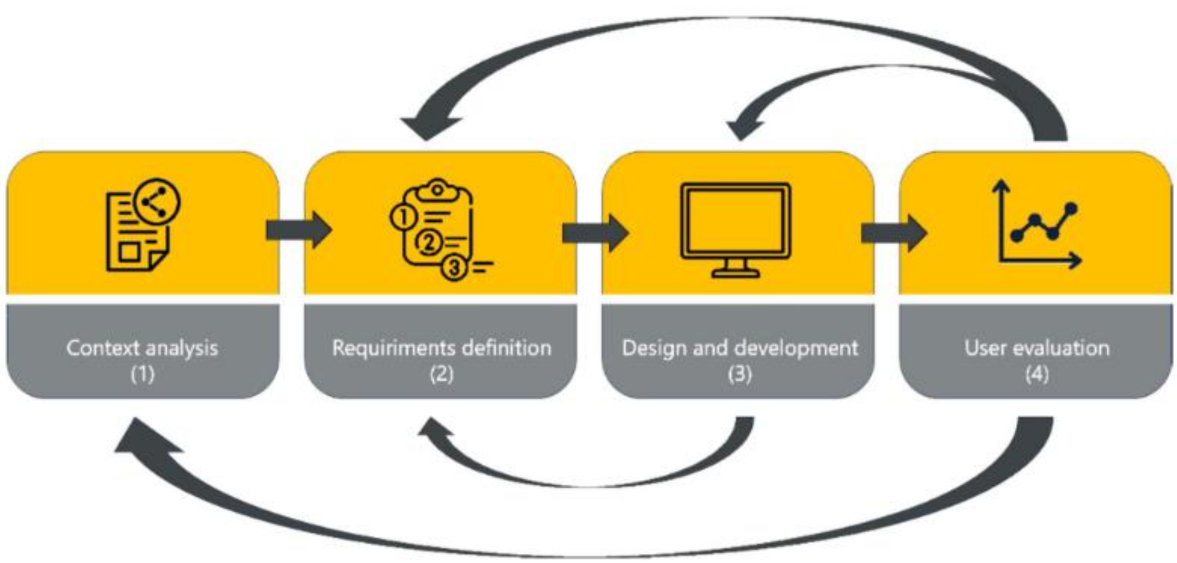


Figure 2 User centered design approach adopted for developing the proposed AR assembly guidance tool (Marino et al., 2024, p. 4944)

3 Literature Review

The literature review is structured from macro to micro. Starting with general methods like industry 5.0 and ending with specific technological applications on the real shop floor. This hierarchical structure allows a contextual understanding of how overarching industrial trends inform specific technological choices in pre-assembly assistance systems and related applications towards technical architectures of current technologies.

3.1 Industry 5.0

"[...]Industry 5.0 is not replacing the Industry 4.0 paradigm chronologically. Instead, it should be regarded as the logical continuation of the existing digital industrial transformation, which can systematically address Industry 4.0 shortcomings in empowering societal and ecological values. Although Industry 5.0 is centered around socio-environmental values, it builds on many features of Industry 4.0, including the idea of merging the physical and virtual worlds through innovative technologies, integrating humans and machines, pushing technological development and contributing to a more competitive industry. Industry 5.0 covers many technologies, principles and components associated with Industry 4.0 while shifting from a technological perspective to a human-centered one"(Ghobakhloo et al., 2023, p. 440)

Industry 4.0 turns from a "technocentric" to a "value centric" paradigm in Industry 5.0. The focus shifts from economic / shareholder profit towards a mindset of environmental sustainability and the wellbeing of industry workers (Enang et al., 2023, p. 1). That should support an 'win-win' interaction between industry and society. Industry 5.0 presented by the European Commission in 2021 (Breque et al., 2021) is linked to 'society 5.0 which was first presented by the Japanese government in 2016 (Huang et al., 2022).

The shift is driven by macro-level external forces and shifting societal expectations, analyzed through the Multiple Level Perspective (MLP) transition theory as a reconfiguration pattern. Landscape factors, which are exogenous and beyond direct control, include large-scale pressures like the COVID-19 pandemic or political conflicts, which underscored the need for resilience and adaptability and Climate Change, which intensified the focus on environmental limits and resource efficiency. Simultaneously, Regime factors including evolving policy, stakeholder expectations and perception to demand changes, such as the rising expectation for mass customization and greater accountability regarding trust, privacy and data security. The transition is fundamentally a socially pulled and technologically pushed digital transformation phenomenon. (Enang et al., 2023, pp. 7881–7883)

The architecture of Industry 5.0 is defined by foundational Techno-Functional Principles and integrated components that enable its value-centric goals. These principles include, decentralization, ensuring more agile, flexible and autonomous production but also data transparency, availability and decision processes. Interoperability, ensuring meaningful communication between all system elements like humans, machines and tools across the value network. Modularity, allowing the decomposition of complex systems into reconfigurable sub-systems to boost flexibility and resilience. Virtualization, primarily relying on working with Digital Twins (DT) and more (Ghobakhloo et al., 2023, pp. 437–438).A holistic overview of the design is visualized in figure 3. The ecosystem's major components span the entire value chain, including the Adaptive Smart Factory (ASF) and integrating Smart Stakeholders through the value and supply chain to mitigate the negative implications of digitalization and ensure the fair distribution of benefits (Ghobakhloo et al., 2023, pp. 438–439).

Industry 5.0 relies on technologies that facilitate Human-Machine Symbiosis and deliver human-centric values. DT technology is extended to the worker through the Human-DT (HDT) or Operator-DT (ODT), allowing for the real-time monitoring of physiological and cognitive states, such as physical exertion, cognitive load and fatigue, to enhance well-being and productivity (Villani et al., 2025). Extended Reality (XR) encompassing Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR) are critical for visual guidance assembly, training and maintenance, offering immersive experiences that improve worker efficiency, safety. (Marino et al., 2024). Advanced systems must be complemented by Explainable AI (EXAI), providing transparency and interpretability to automated decisions, thereby enhancing user comprehension, confidence and trust in the computational systems If that is ensured they are reliable, safe and trustworthy (Trivedi et al., 2024).

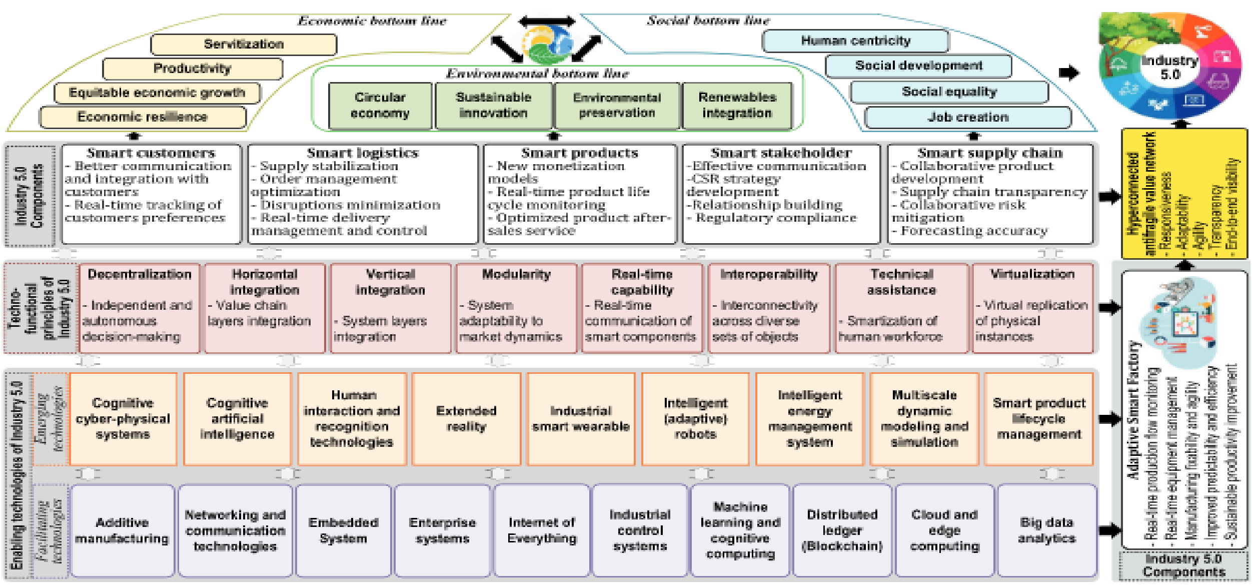


Figure 3 The architectural design of Industry 5.0 (Ghobakhloo et al., 2023, p. 441).

The focus lays on the enabling technologies of industry 5.0, to be more specific the emerging technologies. In this thesis the frame of cyber physical systems, artificial intelligence, human interaction and recognition technologies, extended reality and smart wearables are specially in the focus.

3.2 Smart Factory

Smart manufacturing denotes a data-driven, highly connected and flexible production ecosystem that integrates advanced digital technologies with traditional manufacturing processes to optimize performance across the entire value chain. By combining cyber-physical systems, the Industrial Internet of Things, cloud and edge computing, artificial intelligence and advanced analytics, smart manufacturing environments enable real-time monitoring, decentralized decision-making and continuous self-optimization of machines, lines and factories in real time (Brunetti et al., 2022, pp. 10–11). This results in more resilient and adaptable production systems capable of handling high product variability, shorter innovation cycles and volatile market demands, while maintaining or improving productivity, quality and cost efficiency. At the same time, smart manufacturing fosters end-to-end traceability and transparency from suppliers to customers, in real time (Brunetti et al., 2022, p. 13).

Through high interoperability and interacting systems, methods like predictive maintenance, mass customization, energy and resource efficiency are individually steerable. These capabilities are becoming increasingly essential under regulatory, sustainability and supply-chain robustness requirements due to tense supply chain (Brunetti et al., 2022, p. 25).

Smart factory environments rely on standardized communication protocols to enable data exchange between machines, sensors, controllers and enterprise systems. Figure 4 presents an overview based on the automation pyramid of a factory. Two widely adopted industrial transmission protocols in this domain are OPC UA (Open Platform Communications Unified Architecture (OPCUA) and Message Queuing Telemetry Transport (MQTT). (Sai Guptha et al., 2025, p. 1)

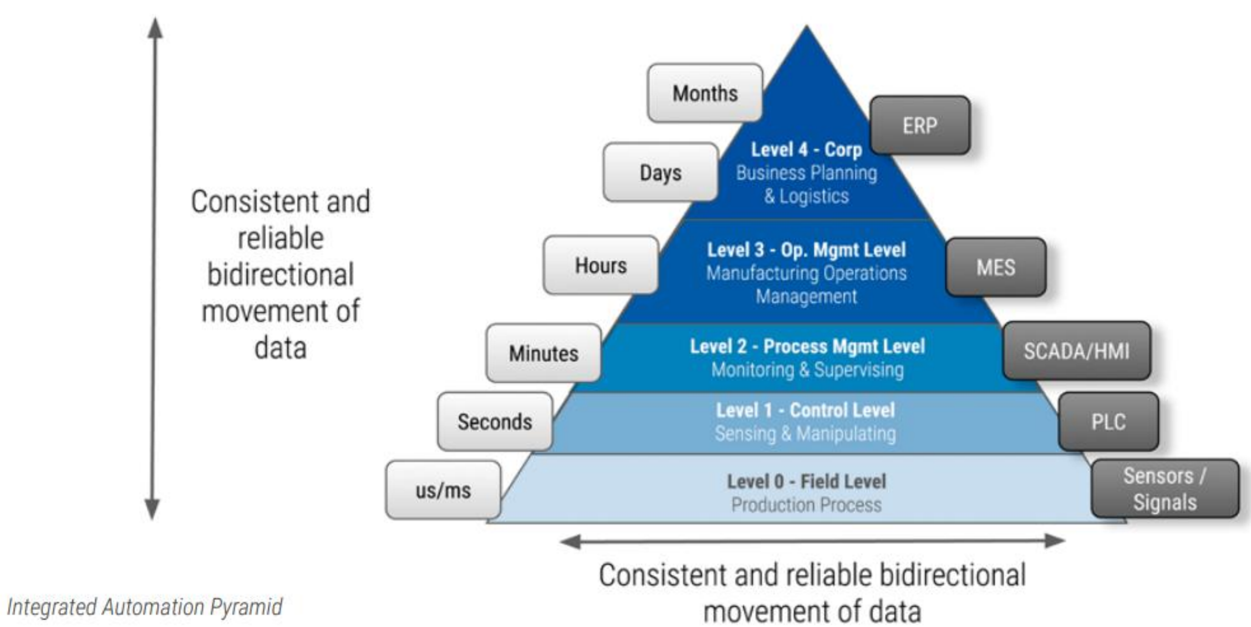


Figure 4 Levels of Integration to Achieve Smart Manufacturing (Modernizing the Smart Manufacturing Industry with MQTT, 2020, p. 4)

OPC UA is a platform-independent, service-oriented communication architecture designed to enable interoperability of industrial automation devices, systems and software applications from different vendors. It organizes information in hierarchical data models and supports both a client-server and a publish-subscribe communication model, enabling rich semantic data exchange between PLCs, SCADA systems, MES and ERP platforms. (Ji & Xu, 2025, p. 9)

MQTT, by contrast, is a lightweight publish-subscribe messaging protocol that has become a standard for connecting IoT devices and is increasingly adopted within smart manufacturing and Industry 4.0 environments. That is also standardized by ISO certification (*ISO/IEC 20922*, 2016). Its publish-subscribe architecture decouples communication between clients, allowing different vendors and systems to implement MQTT independently, all clients push data to a central broker that forwards messages to subscribing clients. (Bender et al., 2021, pp. 1–2; *Modernizing the Smart Manufacturing Industry with MQTT*, 2020, pp. 7–6)

For applications demanding high-speed, synchronized data exchange such as test benches or controller-to-controller communication both protocols reach “low time-to completion times”(D. Silva et al., 2021, p. 28)

3.3 Augmented Reality

Augmented Reality (AR) is a technology that enhances the human´s perception of the real world by overlaying it with computer-generated virtual content. Unlike Virtual Reality, which completely immerses users in a synthetic environment, AR allows to see the real world with virtual objects superimposed upon it supplementing reality rather than replacing it. The concept dates back to the late 1960s when Ivan Sutherland created the first AR system. Today, researchers widely use the definition proposed by Ronald Azuma in 1997, which identifies three key characteristics of AR systems:

- Combination of reality and virtuality: Real and virtual elements are perceived together as a coherent whole
- Real-time interactivity: Users can interact with virtual content instantly
- 3D registration: Virtual objects are anchored to specific locations in physical space, behaving like real objects at those positions

This 3D registration is particularly important. When you change your viewing angle, the virtual object maintains its spatial relationship to the real environment, just as a physical object would. Today this is separated into 3DoF (Three Degrees of Freedom) and 6DoF (Six Degrees of Freedom). 3DoF lets you look around, while 6DoF lets you move around within the digital environment (*3DoF vs 6DoF*, 2025). While AR is most commonly associated with visual augmentation, what we see through smartphone screens or smart glasses can theoretically extend to any sensory experience, including sound, touch and smell. The key distinction from VR is that AR never completely replaces sensory impressions; instead, real and virtual perceptions are layered together. AR is often discussed alongside Mixed Reality (MR), a broader term representing the entire spectrum between complete reality and complete virtuality. Within this reality-virtuality continuum, AR occupies the space where the real world is enhanced with virtual elements. (Doerner et al., 2022, pp. 18–19)

3.3.1 Reality Virtuality Continuum

The studies about mixed reality go back to the year 1994, in which Paul Milgram and Fumio Kishino published a paper with the name "A Taxonomy of Mixed Reality Visual Displays". There they introduced the reality-virtuality (RV) continuum and the term "mixed reality" (MR) (Milgram et al., 1994). The authors back in these days used displays to realize their first ideas and proposed their RV continuum in figure 5.

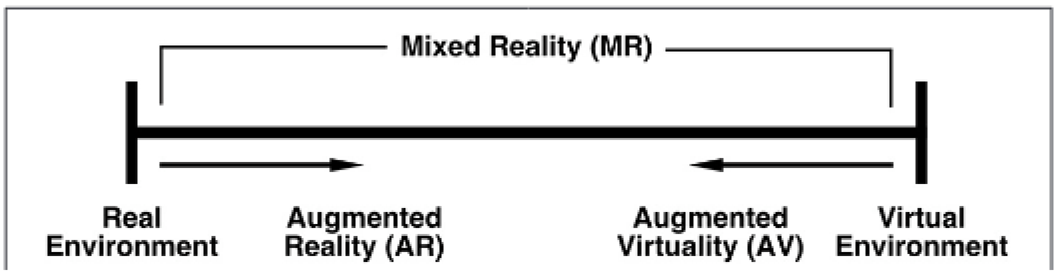


Figure 5 Milgram and Kishino's reality-virtuality continuum (Milgram et al., 1994, p. 283)

Milgram and Kishino's original framework proposed a spectrum ranging from completely real to completely virtual environments, with mixed reality encompassing the middle ground where real and virtual objects blend together. Within this framework, augmented reality refers to real-world environments enhanced with virtual content, while augmented virtuality (AV) describes predominantly virtual environments incorporating some real-world elements (Milgram et al., 1994, p. 283). Figure 6 gives an visual example of how the different realities and virtualities are categorized.

For classifying different MR systems, they proposed three axes for MR classification. First one is the Extent of World Knowledge, it measures how well the system "knows about the environment and its objects. The more accurate the world model is, the better the systems can align to integrate real and virtual elements. Second is the Reproduction Fidelity (RF), it measures the visual and sensory quality of the displayed content (Milgram et al., 1994, pp. 288–289). The higher the RF the more seamless real and virtual elements immerse. Third one is the Extent of Presence Metaphor (EPM), it captures the degree if immersion, whether the user feels inside or outside the mixed environment (Milgram et al., 1994, p. 290).

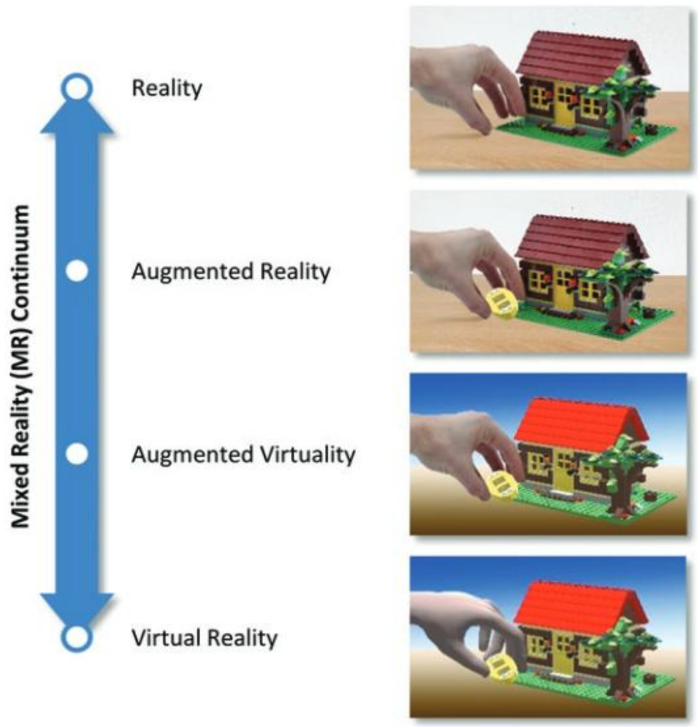


Figure 6 Reality–Virtuality Continuum (Doerner et al., 2022, p. 20)

Skarbez et al. challenge this continuum by making three key arguments. First, they contend that the RV continuum is inherently discontinuous because the "virtual reality" endpoint is unreachable, meaning that all technology mediated reality is fundamentally mixed reality. Second, they propose refining Milgram and Kishino's definition of mixed reality by broadening the requirement from combining real and virtual objects within a single display to combining them within a single percept. Third, they present a new taxonomy for categorizing users' mixed reality experiences, drawing inspiration from the original framework while addressing its limitations. (Skarbez et al., 2021, pp. 1–2)

Central to their argument is the distinction between exteroceptive and interoceptive senses. Modern virtual reality systems have achieved remarkable sophistication in manipulating exteroceptive senses regarding sight, hearing, touch, smell and taste. (Skarbez et al., 2021, p. 2)

However, they argue that even Sutherland's hypothetical "Ultimate Display" (Sutherland, 1965) a system with complete control over all exteroceptive senses, would remain within the realm of mixed reality. The critical limitation lies in interoceptive senses, which monitor the body's internal state, including vestibular and proprioceptive systems. They illustrate this with the example of a virtual spacewalk, regardless of visual fidelity, users would still perceive gravitational orientation and feel their feet on the floor, creating inherent sensory conflicts that characterize conventional virtual reality systems. (Skarbez et al., 2021, p. 3)

Skarbez et al. conclude that external virtual environments, which stimulate exteroceptive senses while leaving interoceptive senses unaltered, cannot achieve true virtuality. They argue that only systems capable of direct brain stimulation—like the fictional technology depicted in the Matrix films, which disconnects sensory organs from the brain to control both exteroceptive and interoceptive inputs—could exist outside the mixed reality spectrum. All other systems, including the Ultimate Display, present mixed and potentially conflicting sensory stimuli to users, making them fundamentally forms of mixed reality rather than pure virtual reality. Visualized in a not reachable addition called the "Matrix-like" virtual environment, presented in figure 7.

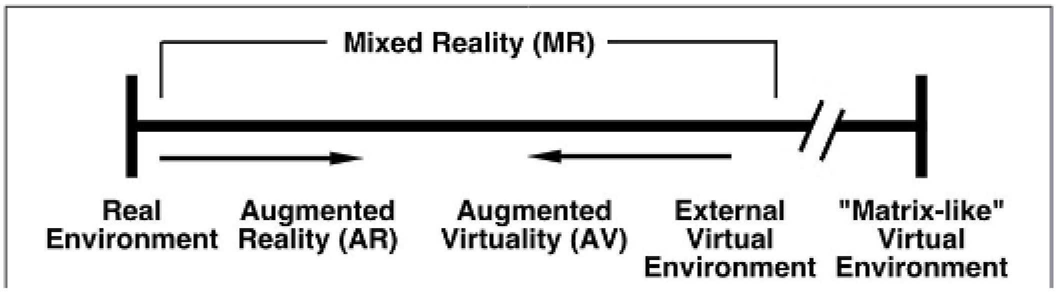


Figure 7 revised reality-virtuality continuum by Skarbez et al. "that the External Virtual Environment (traditionally called "Virtual Reality") is still part of MR" (Skarbez et al., 2021, p. 3)

3.3.2 AR Device Comparison

Head-Mounted Displays

This thesis focuses on Head-Mounted Displays (HMDs), often referred to as smart glasses, they are the most frequently adopted devices in the studies analyzed, particularly for assembly, maintenance and order picking tasks. Their primary strength is their hands-free nature, which allows operators to maintain eye contact with their work and significantly reduces error rates compared to traditional methods. However, they come with notable weaknesses. Studies found out that using HMDs often increased the time required to complete tasks due to operator unfamiliarity. Furthermore, operators reported negative feedback regarding ergonomics, specifically citing that the weight and bulk of the devices caused headaches and dizziness after prolonged use (Belsito et al., 2024, pp. 16–19). This is the case because in these studies, bulky and heavy AR Glasses have been used, comparable to the products that are stated in Appendix A, having a weight over 200g.

AR Projectors

Projectors are considered a valid and highly effective alternative to wearable technology because they project instructions directly onto the workspace or object. They are particularly strong in terms of speed; for example, pick by light systems were shown to reduce pick-up times by more than 70% while also eliminating the physical discomfort associated with wearing headsets (Belsito et al., 2024, p. 22). Consequently, they are especially preferred for employee training. Their main limitation, implied by their nature, is that they are fixed to a specific location, unlike mobile solutions (Belsito et al., 2024, p. 16).

Handheld Devices

Handheld devices, like tablets and smartphones, rely on screens to show digital data onto the real world. Tablets are praised for being inexpensive and easy to implement. Additionally, they are appreciated by senior operators who are already familiar with the technology and perceive them less intrusive than headsets. However, their major weakness is that they are not hands-free; operators must occupy one or both hands to hold the device, which limits their ability to perform manual tasks. Smartphones share these issues but are further hindered by small screens that cannot display wide views of industrial data, making them the least preferred choice for complex applications. (Belsito et al., 2024, p. 16)

Monitors

Monitors represent the least used solution in modern industrial AR contexts, showing a decreasing trend in adoption over recent years. While they can be used for quality control or assembly, their design is fundamentally flawed for manual work. They force the operator to look away from the activity to check the display, causing distractions and breaking the workflow. Because of this ergonomic disadvantage, they are rarely the first choice compared to the other technologies listed (Belsito et al., 2024, p. 16). Figure 8 shows that not all participants could be improved with bulky AR devices (Belsito et al., 2024, p. 19)

		Metrics						
		Objective		Subjective				
		Task completion time	Error rate	Perceived workload	Usability	Learning rate	Qualitative evaluation	
AR device	HMD	↘	↗	↗	↗	→	↘	Traditional method
	Projector	↗	↗	→	↗	–	↗	
	Tablet	↗	↗	→	↗	–	↗	
	Monitor	↗	↗	↗	–	–	–	
	Smartphone	↗	→	–	–	–	–	

*Legend: ↗: improved performance, ↘: worsened performance, →: no significant correlation

Figure 8 comparison of different AR devices (Belsito et al., 2024, p. 19)

In general, there are many more wearable technologies that monitor and enhance the operator. Examples are " [...] intelligent safety helmets equipped with Inertial Measurement Units (IMU) and electroencephalography (EEG) sensors, which can analyze data using artificial intelligence algorithms [...]" (Nguyen et al., 2023, p. 227) or " [...] sensor-equipped safety vests, smart eyewear and smartwatches used for mining operations [...]" (Nguyen et al., 2023, p. 227).

In this context the focus is on see through wearables which are head mounted on the user. These are glasses in different forms. The following section provides an overview of current smart glasses models that can be used in manufacturing applications. The guidelines for reviewing the current market are based on (Syberfeldt et al., 2017)

3.3.3 Current Wearable Smart Glasses Market

A comprehensive overview of current market products is available in Appendix A. Its origin of criteria is built on (Syberfeldt et al., 2017) and (Danielsson et al., 2024) and has been further developed due to the use case presented in chapter 4. Due to the use case it is essential, that the glasses fulfil ideally all of the five criteria:

1. The glasses have a camera included
2. The integrated camera is able to do red green blue detected images
3. The controls of the system use hand gestures without any controller devices
4. An avaliable open development platform to implement machine integration interfaces
5. The glasses weigh below 100g and look like normal glasses so that an operator might wear for a full shift – bulky classic VR headsets are excluded

Analyzing the product overview in Appendix A you can see that most current products are not able to fulfil these criteria. There are market prospects that announced future releases, who might fulfil these criteria, but to the current execution of the thesis, they are not available on the market yet and their full technical details have not been published yet (Hutchinson, 2025; *XREAL Project Aura*, 2025). This lead to the criteria based winner on place number one, which has not been used due to time and cost constraints in this thesis to create the artifact.

Implementing AR solutions and enhancing them with AI or other technologies like digital twins or data lake systems like machine data, can be challenging. Issues that companies are facing no matter what strategy they follow, are current hardware and software limitations of the glasses. Already in 2017 (Syberfeldt et al., 2017) created a comprehensive overview of smart glasses which lead to the conclusion that there have been sophisticated products on the market back in these days. They are very heterogenous because they implement different technologies and different designs, which lead to strengths and weaknesses of the products (Syberfeldt et al., 2017, p. 9129). Recreating this review in November 2025, a completely new variety of products can be examined. ARSG is currently turning from an industrial niche market into a consumer-based lifestyle product. Many smart glasses target end consumers in enhancing their daily life with integrated displays, sound speakers and Gen AI gadgets. The capabilities of these glasses are rather restricted and often only perform in the manufacturer´s ecosystem but therefore the producers sell it for a lucrative price. Reviewing the current market, it turned out that the year 2025 is rather a year of many pilot projects for AR smart glasses. Many producers tried out a variety of developments and are preparing to present next year’s future projects. Through this research, a new perception arises. Assessing technology and design can also lead to individual and subjective strengths and weaknesses of the product, due to different human favors regarding the ergonomics and comfort of wearing these glasses.

The selection of appropriate augmented reality smart glasses (ARSG) for industrial applications requires a structured, multi-step evaluation approach to ensure both economic and operational feasibility. The proposed process comprises three main stages. First is the initial budget and availability identification, second is the preliminary screening and third, a comprehensive evaluation (Syberfeldt et al., 2017).

Budget and availability identification

Smart glasses that are not available due to limited production or its development stage are not valid to evaluate if they can´t be used. In addition, the costs have to be justified, not only the purchasing price but also the total cost of ownership must be defined.

Eliminating unsuitable products

The devices have to be used by the operators, that means they have to able but also have to be comfortable with using the devices. For assessing this, multiple factors play is role depending on the use case. Main parameters are the (1) powering, (2) weight, (3) field of view, (4) battery life and (5) optics.

Comprehensive technical evaluation

The technical evaluation goes beyond the use of the operator. Controls, processors and connectivity of the devices have to be evaluated. Smart glasses that can´t be connected to the needed systems or can be connected but can´t provide the needed processing power, slow down the use or make the use completely obsolete.

The third part is the most time-consuming due to technical capabilities regarding different interfaces. No matter how good the research on technical capabilities is, the software might be changing rapidly and so the evaluation might be outdated within several months. In addition, the evaluation is only as good as the public available documentation is. To get the best and most reliable evaluation it is important to stick to best practices and service developing providers, that reference to successful implementations.

3.4 Artificial Intelligence In Augmented Reality Applications

In AR applications there is way more AI than at first hand might be expected. Augmented reality has evolved from simple overlay systems to sophisticated platforms that understand, interpret and enhance physical surroundings in real-time. This transformation is driven by artificial intelligence, which provides AR systems with the capability to perceive depth, recognize objects, track movements, improve computational power and generate contextually appropriate digital content. The integration of AI into AR follows similar patterns to VR development, moving from basic immersion to encompass intelligentization, interconnection and continuous iteration. (L. Wang et al., 2025, pp. 1–2)

The foundational algorithms of augmented reality, like simultaneous localization and mapping, environment meshing, hand and eye tracking, spatial anchoring and content rendering, already incorporate AI-driven techniques at their core and are available as platform-level capabilities on current commercial headsets. The technological basis for AR-assisted industrial applications therefore exists and still continuously is being improved. The challenge for manufacturing use cases is not to develop these algorithms from scratch but to build upon them and leveraging the existing AI-powered perception and tracking infrastructure as a foundation and extending it with domain-specific capabilities to address the particular requirements of the industrial context.

3.4.1 General Object Detection

Computer Vision (CV) represents a specialized domain within Artificial Intelligence (AI) and computer science, focused on enabling computational systems to understand and process visual information. Using algorithms, statistical models and Machine Learning (ML) approaches, CV systems can analyze digital images and video content to perform diverse functions. These include object recognition, facial expression analysis, image segmentation and human action recognition. These methods are significantly improving all kinds of VR, AR, XR, MR towards better recognition and understanding of visual information. (Lopes & Lopes, 2024, p. 13) figure 9 shows an example of the different steps between classification, localization and object detection, while figure 10 shows the difference between Object detection and Instance Segmentation.

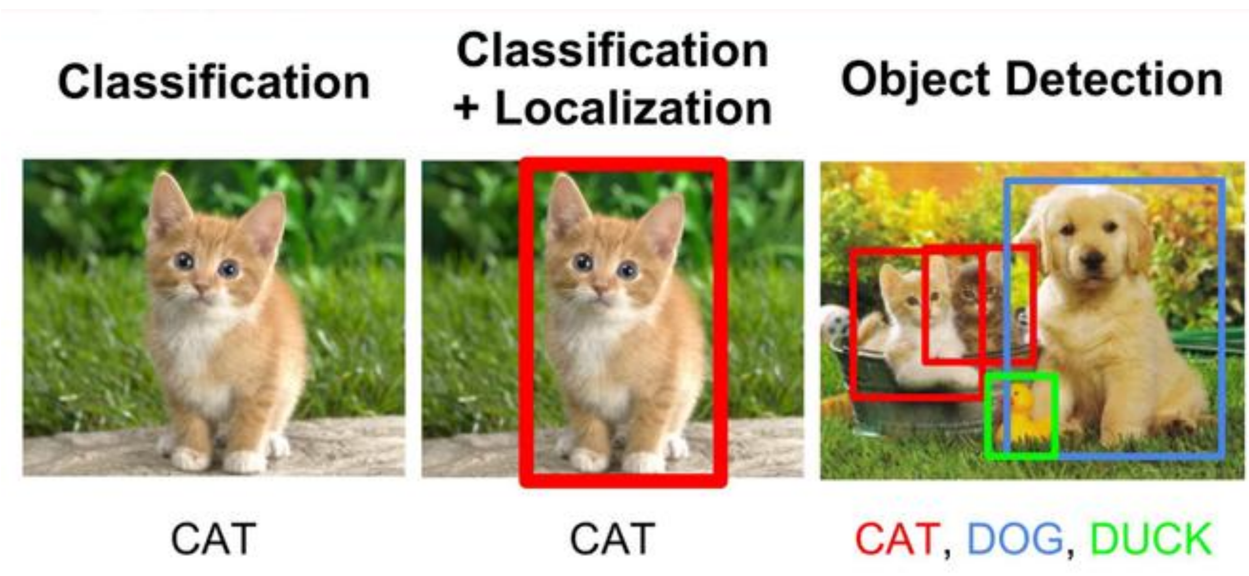


Figure 9 Difference between classification. Localization and Detection (Object Detection vs Object Recognition vs Image Segmentation, 2025)

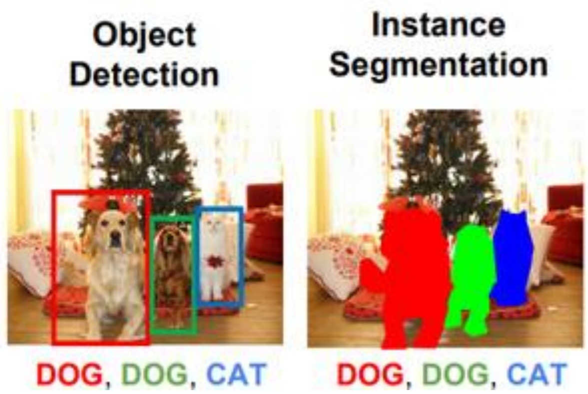


Figure 10 Object Detection vs Segmentation (Object Detection vs Object Recognition vs Image Segmentation, 2025)

The focus is on object detection, because in a manual assembly station, the AR application should detect what object there is and guide the user according to whether it is the correct or the wrong component. For object recognition there are many different models, which reach from very precise and computational intense models to so called nano models which are created for low resolution and low computational power (DigitalOcean, 2025). Depending on the hardware, the recommended software should be used.

For object detection, several different technologies can be used. There are Classical detections and Deep Learning based detections. The classical have been used before the deep learning methods. Within the Deep Learning based detection, there are Convolutional Neural Network (CNN)-Based Detectors, Transformer-Based Detectors. A holistic overview in form of a tree diagram about fundamental methods is listed in Appendix B. In the following a few examples of the Deep Learning Two Staged Detectors but also the One Staged detectors will be explained.

Two Stage algorithm examples

In 2014 the Region-based Convolutional Neural Network (R-CNN) algorithm was introduced by Ross Girshick and his team. Achieving a mean average precision of 53.3% on the PASCAL VOC 2010-12 dataset. The algorithm operates through three distinct stages. First, it identifies potential regions of interest by generating bounding boxes across the image. Second, a deep neural network extracts features from each identified region. Third, these features are used to classify the detected objects. The algorithm employs Selective Search to create bounding boxes, which groups adjacent pixels based on similarities in color and intensity, producing 2000 region proposals of varying sizes per image. For feature extraction, the original implementation utilized a modified version of the AlexNet architecture. The extracted feature maps are then processed by Support Vector Machine classifiers to determine object categories for each candidate region. (Tsirtsakis et al., 2025, p. 302)

Girshick introduced Fast R-CNN in 2015, merging key elements from R-CNN and Spatial Pyramid Pooling (SPP) networks. The architecture builds upon a pre-trained VGG-16 network with modifications. The final subsampling layer is substituted with a Region of

Interest layer and two distinct output branches are incorporated after the fully connected layers. The system processes both complete images and region proposals generated by Selective Search. The VGG-16 backbone first creates a feature map from the input image. Subsequently, the ROI layer produces fixed-length feature vectors for each proposed region, functioning similarly to spatial pyramid pooling. These vectors then feed into dual output branches—one performing classification via SoftMax activation, while the other refines the bounding box coordinates through regression. (Girshick, 2015)

The algorithms discussed previously all rely on Selective Search to locate potential object regions, a computationally expensive process. To address this bottleneck, Ren Shaoqing and colleagues introduced Faster R-CNN in 2016, which retains the Fast R-CNN architecture while eliminating Selective Search entirely. The key innovation is the Region Proposal Network, a component that generates candidate regions directly from the convolutional feature maps produced by the CNN backbone. Through internal convolutions, the RPN outputs approximately 300 bounding box proposals along with their corresponding confidence scores. The detection pipeline mirrors Fast R-CNN's approach, an input image of arbitrary dimensions passes through the VGG network to produce feature maps, which the RPN then analyzes to propose regions. These proposals proceed to the ROI layer for further processing identical to Fast R-CNN. This architecture achieves significantly faster performance than its predecessors, processing images in under 0.2 seconds with a mean average precision of 73.2%, making it suitable for real-time object detection applications. (Ren et al., 2016)

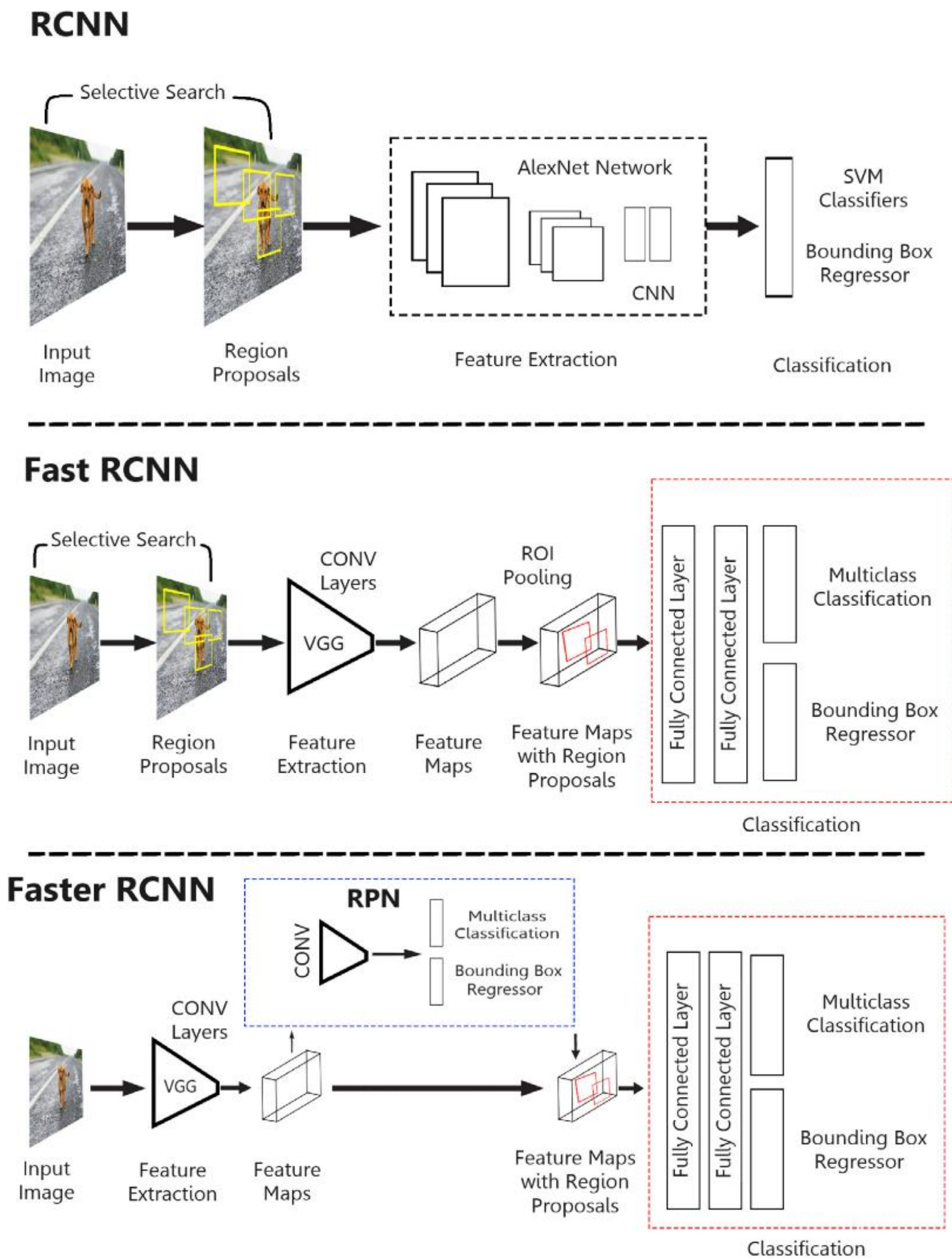


Figure 11 Illustration of RCNN family detection algorithms (Tsirtsakis et al., 2025, p. 303)

One-stage algorithm examples

The OverFeat algorithm, released in 2013, pioneered the concept of using a unified CNN to handle detection, recognition and classification simultaneously. The central premise was that joint training across multiple tasks could enhance overall performance through a multi-scale classification approach, where images are analyzed at various dimensions through at least five passes. Built upon the AlexNet (Tsirtsakis et al., 2025, p. 301) architecture, OverFeat comes in two variants, a fast version and an accurate version. These variants differ primarily in their convolutional filter sizes and stride values. The fast version employs larger parameters, producing smaller feature maps and reduced subsampling layer dimensions compared to the accurate version. For object detection, the architecture departs from AlexNet by replacing the SoftMax classification layer with two fully connected layers that output bounding box coordinates and detection results. While OverFeat offered superior training speed, it achieved lower accuracy than R-CNN, its contemporary competitor. The algorithm secured first place in the localization challenge at the ILSVRC 2013 competition. (Tsirtsakis et al., 2025, pp. 303–304)

The You Only Look Once (YOLO) section represents the most widely adopted object detection algorithms originally developed by Joseph Redmon and his team in 2016 and continuously refined since (Redmon et al., 2016; Ultralytics, 2026c). YOLO frames object detection as a regression problem using a CNN architecture, distinguishing itself through exceptional speed, high frame rate support and strong accuracy. These characteristics enable real-time detection applications. The initial YOLO version divides input images into uniform grid cells. For each cell, the algorithm generates bounding box predictions accompanied by confidence scores indicating object presence. Each prediction encompasses the object's center coordinates relative to the cell, bounding box dimensions and class probabilities. The detection responsibility is assigned exclusively to the cell, containing an object's center point. Built on the GoogLeNet architecture, YOLO v1 achieved 63.4% mean average precision at 45 frames per second and 52.7% mAP at 155 FPS on the PASCAL VOC 2007 dataset. Despite these achievements, limitations emerged, prompting the development of subsequent YOLO iterations. (Tsirtsakis et al., 2025, p. 304)

Each Yolo model has its strengths, while continuous development has not stopped yet. The most current Yolo version is YOLOv12. Additionally to the normal model versions, there are size versions like tiny, small, medium or large which are specified to the use case, based on the computational power and the desired output. (soumyadip, 2023)

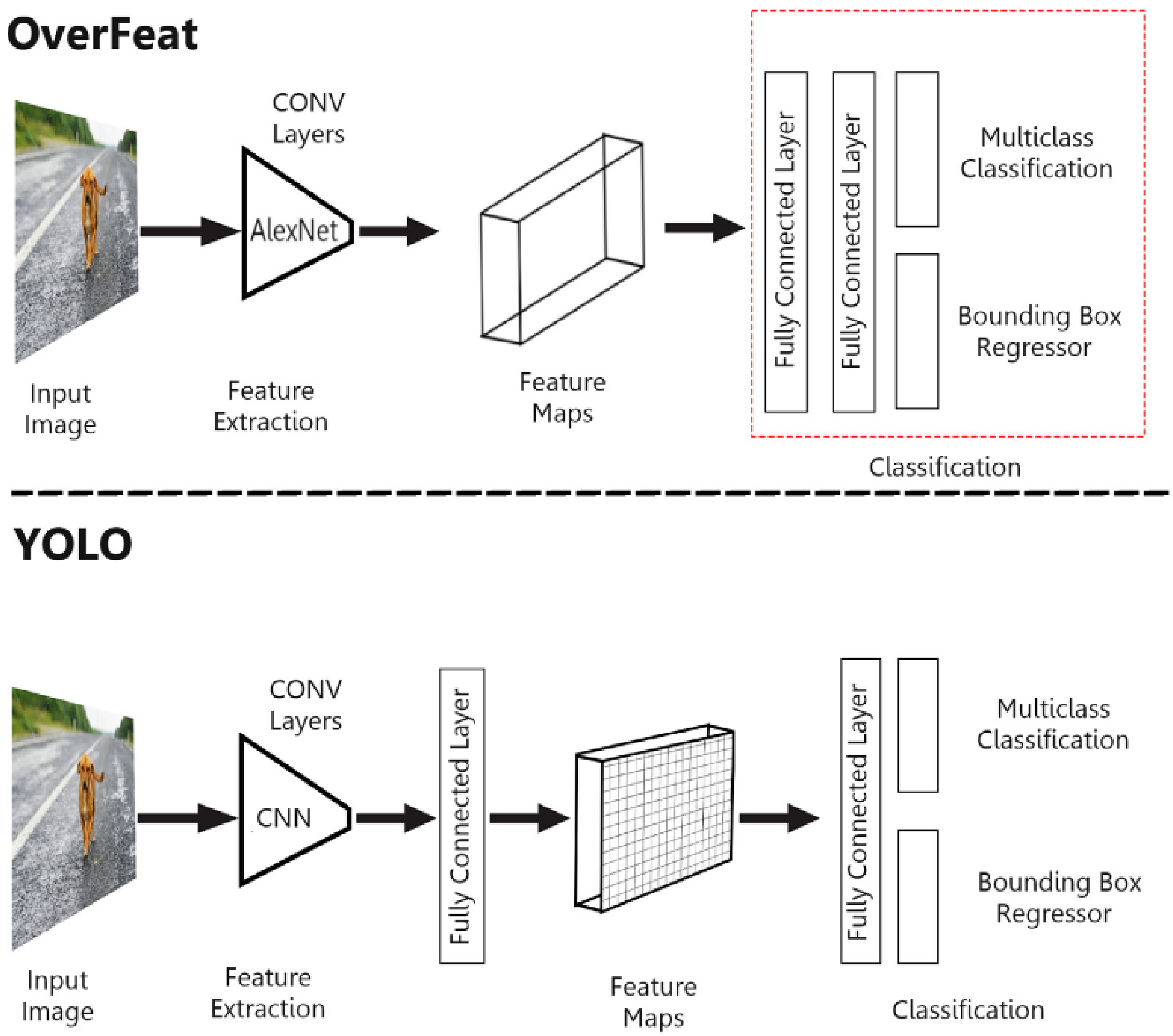


Figure 12 Illustration of OverFeat and YOLO detection algorithms (Tsirtsakis et al., 2025, p. 305)

3.4.2 AI-Generated Content Representation

To improve the quality and the speed of what the user through the AR/VR glasses sees, the computational power of the glasses is usually not sufficient, that's why improvements like in the following sections are presented. The basis of the following methods is that visual content, which the human sees through his eyes, is represented and further used through AI algorithms to provide an as realistic as possible image to the user. Therefore, two approaches are briefly explained, the Neural Radiance Fields (NeRF) as an implicit approach and 3D Gaussians as an explicit approach. (L. Wang et al., 2025, p. 8)

These two methods are used in passthrough devices, which have to replicate the surrounding area of the user. The device overs the field of view with a head mounted display which shows the environment with the latency of the inference from the device camera, hardware computation and display screening. See through devices don´t have to replicate the environment because the user can see them regularly through the glasses, which solves that problem by itself. Still the approaches are needed for the used Meta Quest 3.

The NeRF algorithm represents a 3D scene not as a geometric object, but as a continuous mathematical function with a five-dimensional input and a volume density + color as output. Three inputs are spatial coordinates and two inputs are viewing directions. The density defines how transparent the pixel is and the color what color the pixel has (L. Wang et al., 2025, pp. 8–9). Traditional implementation of NeRF require very long training durations, around 20h are typical (L. Wang et al., 2025, p. 9).

To improve the training time one approach is Point-NeRF. It employs feature point clouds as an intermediary representation in the volume rendering pipeline. This method utilizes a pre-trained 3D convolutional neural network to generate depth maps and surface probabilities from cost volumes constructed from multiple training views, thereby producing dense point clouds that encode geometric information. By representing the volumetric radiance field through this neural point cloud structure, Point-NeRF facilitates highly efficient scene reconstruction with 30x faster training time, representing a big improvement over the original NeRF's requirement of over 20 hours. (L. Wang et al., 2025, p. 9; Xu et al., 2023, p. 1)

An alternative paradigm is the Plenoxels representation, which models scenes as sparse three-dimensional voxel grids. In this framework, each voxel stores density values and spherical harmonic coefficients that encode view-dependent appearance. A key innovation of this approach is that the representation can be optimized directly on the voxel grid from calibrated images through gradient descent and regularization, entirely circumventing the need for MLP training. The optimization procedure follows a coarse-to-fine strategy, beginning with a lower-resolution dense grid that is progressively refined through the elimination of unnecessary voxels and subsequent up sampling of the remaining structure. This methodology of NeRF leads to a training time more than 100x speedup. (L. Wang et al., 2025, p. 9; Yu et al., 2021, p. 1)

Addressing the dual objectives of rapid training and minimal storage requirements, Chen et al. proposed TensoRF, which tackles the inefficient utilization of voxel grids observed in previous methods. TensoRF models the scene's radiance field as a four-dimensional tensor, representing the three-dimensional voxel grid with multi-channel features at each voxel location. The method then applies conventional tensor decomposition algorithms during the radiance field modeling process, factorizing the full tensor into multiple compact, low-rank tensor components. Through dimensionality reduction and data compression afforded by tensor decomposition, TensoRF achieves with the vector matrix decomposition, reconstruction speeds of less than 10 minutes while simultaneously reducing spatial memory requirements during the modeling phase. (Chen et al., 2022, p. 1; L. Wang et al., 2025, p. 9)

The current state-of-the-art in training efficiency is exemplified by Instant-NGP, proposed by Müller et al., which achieves training times of mere seconds. This method introduces a learnable, parametric multi-resolution hash encoding scheme that is optimized concurrently with NeRF's MLP components. By combining this parametric encoding approach with sophisticated ray marching techniques. Including exponential stepping strategies, mechanisms for skipping empty spatial regions and sample compression methods, instant-NGP represents a paradigm shift in NeRF training efficiency. This method is reducing the temporal requirements from hours to seconds while maintaining high-quality reconstruction fidelity. (Müller et al., 2022, pp. 1–2; L. Wang et al., 2025, p. 9)

The 3D Gaussian algorithm represents a scene explicitly as a collection of millions of 3D ellipsoids, imaginably shaped like 3D ovals. Each ellipsoid has a learnable property, like its center, transparency, shape and color. The algorithm automatically clones or splits ellipsoids to add detail where needed or prune them where they are transparent or unnecessary. Unlike the NeRF, the 3D Gaussian algorithm is trained in real time which outperforms the NeRF algorithm. (L. Wang et al., 2025, pp. 10–11)

The 3D Gaussians representation offers a fundamental advantage over Neural Radiance Fields. the integral of a 3D Gaussian along a specific axis yields a 2D Gaussian, requiring only one sample per 3D Gaussian and enabling superior performance. However, depth sorting for each pixel during rasterization creates a significant computational bottleneck. To address this challenge, the tile-based parallel computation method that partitions images into 16×16-pixel tiles has been proposed by Kerbl et al. to achieve a real time training of "[...] real-time (>= 30 fps) novel-view synthesis at 1080p resolution." (Kerbl et al., 2023, p. 1). The algorithm identifies tile-Gaussian intersections, duplicates Gaussians spanning multiple tiles and assigns tile identifiers. These IDs are combined with depth values to generate a sortable byte list, where high-order bits encode tile IDs and low-order bits represent depth. This sorted list directly facilitates alpha synthesis, with acceleration achieved through independent tile and pixel rendering, shared memory access and uniform read ordering. Despite tile-based acceleration, large-scale and dynamic scenes requiring millions of Gaussians still demand substantial construction time. Current efforts to improve efficiency fall into three principal directions (L. Wang et al., 2025, p. 11).

One approach to improvement is to reduce the number of 3D Gaussians through pruning. Lu et al. introduced Scaffold-GS, which leverages scene structure to guide removal of overextended Gaussians. The method constructs a sparse anchor point mesh, attaches learnable Gaussians to each anchor and predicts properties dynamically. A gradient-guided strategy governs anchor point addition, while volumetric regularization minimizes Gaussian overlap (L. Wang et al., 2025, p. 11). This approach can render at fraction of the storage size, with mostly higher FPS, compared to usual 3D Gaussians (Lu et al., 2023, p. 6).

Another approach to improvement employs hierarchical modeling. Kerbl et al. proposed Hier3DGaussian, which recursively partitions Gaussians into axis-aligned bounding boxes using a divide-and-conquer strategy. Each leaf node contains a single Gaussian, while intermediate nodes form a multi-layered binary tree. Gaussian properties for intermediate nodes are computed bottom-up from leaf nodes, establishing a hierarchical representation that enables precision-level selection based on quality requirements (L. Wang et al., 2025, pp. 10–11). This method gives the opportunity to render large scale scenes, which are not even able to be rendered by past approaches. It is designed to handle these massive scenes by only rendering a subset of Gaussians based on viewing distance and required detail. Therefore the author evaluated the method by chunks and compared it to the classical 3D Gaussian in the chunk comparison, which always reached the highest PSNR and FPS, depending on the fine tuning of the model (Kerbl et al., 2024, pp. 10–12).

3.4.3 Content Rendering Models

Material representation and operations

To make virtual components in an Augmented Reality environment as realistic as possible, a lot of computational power and big data sets are needed. The challenge is that real world material surfaces are complex and data derived from scanning real objects often result in Bidirectional Texture Functions (BTFs). These Functions contain 6-dimensional datasets gaining massive amounts of information. Storing and processing this data is difficult and traditional methods often fail to capture complex effects like depth within a surface or light scattering under the surface. The solution is to compress these massive material datasets into small and efficient models. Therefore, an AI model learns the “essence” of a material and can reproduce it efficiently. (L. Wang et al., 2025, pp. 20–22)

Light Transport Computation

The standard method to calculate realistic light is the Monte Carlo ray tracing method, with which millions of light rays are traced randomly. This high amount of rays is needed to pretend getting a noisy or grainy image. With Neural Path Guiding an AI is used to “guide” the rays instead of shooting rays randomly. Through analyzing the scene the neural network calculates the probability of from where the light is likely to come from and focuses on these areas. This produces a cleaner image much faster. (L. Wang et al., 2025, pp. 22–23)

Rendering Post-Process

Because the computer generates the image quickly using fewer light samples, the raw result has random noise, that must be cleaned. Denoising is the mathematical process of smoothing this out. Therefore, the slower Offline Denoising for high quality and the faster Real-Time Denoising for VR can be applied. Offline Denoising is used for movies or still images. The AI looks at the grainy image and predicts what the smooth version should look like. It uses CNNs, to filter out the noise while keeping sharp edges, so the image doesn't look blurry. Real-time Denoising, is used in games where speed is critical and uses

Temporal Coherence. This means the AI looks at the previous frame to help fix the current frame. If a pixel was blue in the last millisecond, it's likely still blue in the next millisecond. By blending information from the past and present, it can instantly smooth out the noise. (L. Wang et al., 2025, pp. 24–25)

The user wearing smart glasses still should be able to see high quality images. This is possible due to Super Resolution which aims to increase the spatial resolution of images or video frames. Therefore, the images are not just stretched, which would make them blurry, but due to Deep Learning, that predicts the missing details. It uses additional information like motion vectors across multiple frames and creates a sharp, high-resolution image that looks almost identical to one rendered natively. (L. Wang et al., 2025, p. 25)

To decrease motion sickness in VR Frame Interpolation is used. The computer renders frame 1 and frame 2 and generates additional fake frames between them and inserts them in the middle to increase the frames per second rate. The AI analyzes movement of pixels in between the two real frames and creates a mathematical map of the movements. Based on that calculation it constructs these new images. The user sees smoother motion without the computer having to do the heavy physics calculations, from the traditional material representation and light transport calculation for each pixel, of that middle frame. (L. Wang et al., 2025, p. 25)

3.4.4 Generative Models In AR Content Creation

The creation of 3D content, ranging from individual objects to complex scenes, is foundational to AR applications. Traditionally a labor-intensive task for artists, creating virtual models, this process has been improved by deep generative models that create 3D assets conditioned on text or 2D images. The field relies heavily on two main architectures, Generative Adversarial Networks (GANs) and Diffusion Models. GANs utilize a generator-discriminator adversarial training loop to produce 3D representations like point clouds, voxels and implicit surfaces, they struggle with highly complex, high-dimensional distributions(L. Wang et al., 2025, p. 26). Consequently, diffusion models have gained prominence because they "[...] can outperform GANs after being trained on a large scale of data."(L. Wang et al., 2025, p. 26) Through a series of noise-driven steps called the forward process, a original data distribution is transformed into a simpler distribution and the model learns how to reverse this process to generate new samples. These samples are similar to the original distribution (L. Wang et al., 2025, p. 26)

3D Object Generation Techniques

Object generation involves creating detailed 3D models from scratch or limited inputs, requiring a deep understanding of geometry and texture. Current approaches generally fall into three categories. First, 2D generative models, that leverage powerful 2D image generators to turn these images into 3D shapes. This method often suffers from multi-view inconsistency (L. Wang et al., 2025, pp. 26–27). Second, methods that impose multi-view images aiming to fix these consistency issues by training models to generate consistent views or using view-aware mechanisms. Still this method sacrifices geometric detail (L. Wang et al., 2025, p. 27). Third, 3D native generation models trained directly on 3D datasets. This includes Large Reconstruction Models (LRMs) that reconstruct implicit representations from sparse views and explicit models that generate point clouds or meshes directly (L. Wang et al., 2025, pp. 27–28).

Implicit and Explicit Scene Synthesis Approaches

Beyond individual objects, scene synthesis generates complex environments with logical layouts, to understand the objects in the scene. This field is evolving from GAN-based methods toward LLM and diffusion-based techniques, categorized into implicit and explicit synthesis (L. Wang et al., 2025, pp. 28–31). Implicit scene synthesis uses continuous functional representations to generate organic, fluid scenes often controlled by layout, text or image priors. For instance, some models use Bird's-Eye-View representations or text-driven diffusion priors to create navigable 3D worlds (L. Wang et al., 2025, pp. 29–30). Conversely, explicit scene synthesis constructs scenes from distinct components like objects and layouts, offering greater control and interpretability. Recent innovations even utilize Large Language Models to plan scene layouts based on textual descriptions. While implicit methods excel at organic generation, explicit methods provide precise control. Future prospects are likely to combine both for immersive VR and AR scene creation (L. Wang et al., 2025, p. 30).

3.4.5 User Tracking And Recognition

Augmented Reality and Virtual Reality interactions involve the relationship between the user, virtual elements and the user's subjective perception. AI enhances these interactions through three primary pillars, which are:

- Behavior Recognition: Identifying user actions accurately.
- Interaction Optimization: Improving user actions within the virtual space.
- Perception Analysis: Analyzing user perception to increase immersion and satisfaction.

Focusing on Behavior Recognition, the systems must recognize complex intentional behaviors through hands, eyes and face. The most important thing for the use case of a manual assembly station is hand gesture behavior, because with the hands the operator should be guided but also tracked to pick up the correct component from the correct box.

Hand tracking systems face several significant obstacles that can disrupt the user experience. Occlusion presents the most substantial physical barrier. When hands overlap or when users form a fist, fingers become hidden from the camera's view. Without visual access to all digits, the system loses its ability to accurately map the hand's skeletal structure, resulting in tracking failures. Another persistent challenge stems from depth ambiguity. Standard RGB cameras capture only two-dimensional images, making it difficult to distinguish whether a hand appears small due to distance from the sensor or simply because of the user's natural hand size. This ambiguity can lead to inconsistent tracking accuracy and scaling issues. Finally, field of view limitations impose practical constraints on interaction design. Whenever users turn their attention away from their hands, such as when reaching behind themselves to retrieve an item from a virtual backpack, the headset's cameras lose sight of the hands entirely. This breaks the tracking chain and interrupts the interaction. (L. Wang et al., 2025, pp. 43–49)

3.5 Enhancing A Smart Factory with AR

Throughout the product lifecycle, AR demonstrates multifaceted utility across operational domains. During manufacturing phases, AR applications support assembly procedures, quality assurance protocols and collaborative safety frameworks (Sahu et al., 2021, pp. 4903–4904). In the commercialization phase, AR enables prospective consumers to visualize product integration within their operational or domestic area. Post-purchase, AR functions as a diagnostic instrument for troubleshooting and maintenance interventions. Applicable to products, but also machinery that has been installed in the factory. At end-of-life, AR can be redeployed in manufacturing contexts to facilitate product disassembly and material recovery operations (Sahu et al., 2021, p. 4909).

A distinguishing characteristic of ARSG is its minimal hardware requirements, necessitating only the head-mounted display unit without auxiliary equipment. The integration of hand gesture recognition, eye-tracking and voice command interfaces enables intuitive human-computer interaction. Through system interoperability, ARSG can be interfaced with heterogeneous data infrastructures, facilitating continuous bidirectional information exchange between operators and databases while maintaining uninterrupted workflow attention. This cognitive support mechanism has demonstrated effectiveness across three primary industrial applications, assembly operations, maintenance procedures and workforce training (Nitol et al., 2025, pp. 13–15).

The seamless presentation of procedural information through AR interfaces is important for minimizing operator cognitive load, which directly correlates with reduced operational expenditure, enhanced process efficiency and increased productivity metrics. In AR-assisted assembly and maintenance operations, practitioners maintain visual and cognitive focus on manual tasks without attentional reallocation. This capability enables operators to execute procedures involving unfamiliar components, as AR systems eliminate reliance on memorized operational sequences. Consequently, product customization can be implemented with minimal process complexity. Empirical studies have documented error rate reductions of up to 82% in AR-assisted assembly operations. While information provisioning constitutes the primary functional objective, several secondary benefits contribute substantively to productivity enhancement. These include mitigation of cognitive disruption associated with traditional information retrieval methods e.g., computer terminals or printed documentation, collaborative functionality within AR frameworks, system interoperability with manufacturing execution systems and adaptive capabilities that respond to both facility-specific parameters and real-time operator status. AR systems can dynamically retrieve production schedules and present component inventories and assembly protocols directly at workstations. The technology has been successfully deployed across diverse maintenance activities, including inspection, metrology, servicing and assembly tasks. Bidirectional context-aware content authoring further augments overall productivity. Empirical evaluations indicate that AR-mediated training for complex manufacturing procedures achieves equivalence with traditional face-to-face instruction methodologies. Additionally, AR enables virtual simulation and process development prior to physical implementation, supporting first-time-right execution strategies. Collaborative AR applications facilitate simultaneous access to distributed expert knowledge across geographically dispersed locations, enabling multi-stakeholder problem-solving and decision-making processes. (Sahu et al., 2021, pp. 4909–4911)

As already presented, AR technology fundamentally reduces cognitive load by providing contextually appropriate information through optimized temporal and modal delivery mechanisms, thereby preserving operator attentional focus and task engagement. However, implementation challenges exist within industrial environments. Elevated ambient noise levels in manufacturing facilities preclude reliable audio-based or voice-command AR interactions without implementing alternative user interface modalities or robust acoustic noise suppression algorithms. Furthermore, mandatory personal protective equipment compliance constrains the deployment of haptic feedback modalities, necessitating alternative interaction paradigms for effective human-AR system integration.(Sahu et al., 2021, p. 4904)

3.5.1 Overview

The operational scope and functional capabilities of ARSG are contingent upon hardware specifications and enterprise system integration, enabling operator support both within and beyond the production floor. Extensive research has documented diverse use cases and industrial applications, which are systematically presented in this section. Table 4 provides a comprehensive taxonomy of primary manufacturing domains where ARSG implementation has demonstrated significant operational impact. Analyzing the Terms in a row, it has to be clarified, that terms of one column can be combined with terms of other columns from other rows too, they are not limited to its own row. An example would be that the primary support function of remote expert supports, can not only enhance equipment optimization, but also workforce training, assembly operations and maintenance procedures.

Application Domain	Primary Support Function	Operational Impact
Warehouse Logistics	Hands-free Operation	Enhanced mobility and concurrent task execution
Assembly Operations	Real-time Information Access	Immediate procedural guidance and specification retrieval
Production Scheduling	Integrated Tool Consolidation	Unified interface for multiple planning systems
Maintenance Procedures	Contextual and Adaptive Support	Situation-aware technical assistance and diagnostics
Quality Control	Documentation and Compliance	Automated inspection protocols and regulatory adherence
Workforce Training	Multilingual and Universal Communication	Standardized instruction delivery across linguistic barriers
Occupational Safety	Scalability and Hazard Mitigation	Extensible safety protocols and risk awareness
Equipment Optimization	Remote Expert Support	Geographically distributed technical consultation
Operator Monitoring	Performance and Ergonomic Tracking	Workflow analysis and occupational health assessment

Table 4 Primary Application Domains and Support Functions of ARSG in Smart Manufacturing Environments

The subsequent sections provide detailed analysis of each primary support function, examining specific use cases, implementation methodologies and empirical evidence of operational effectiveness within contemporary manufacturing environments. The following section will focus on the operational advantages using ARSG, excluding current disadvantages and challenges of using ARSG which have been stated in chapter 1.1.

3.5.2 Warehouse Logistics

With Augmented Reality there are four main methods to improve the logistic area of a smart factory. The first is improved visualization, with which transportation routes and processes can be virtualized and evaluated completely virtual or added in an augmented version of the real location of the shop floor(Danielsson et al., 2024, pp. 3756–3758). Not only transportation vehicles, sequences and paths can be planned faster, but also picking the correct components can be improved with pick by vision (Jumahat et al., 2023, pp. 5–7). In addition to visualization, the interaction with cyber physical systems is enhanced by continuous feedback to the worker from the system (Rejeb et al., 2021, p. 3758). Decreasing the risk of potential safety issues or executing wrong processes. Because smart glasses don´t have to be held in the hand, like a tablet or piece of paper, the convenience can be improved by hands free operation (Rejeb et al., 2021, p. 3578). Lastly ARSG can improve the navigation of manned (Rejeb et al., 2021, p. 3579) and the planning of unmanned transportation vehicles (Mourtzis et al., 2024). Manned transportation can be coupled with an interacting environment in which other operators can be seen through the virtual environment if in the real environment it is hard to see. But also, to improve routing in a virtual environment, which can´t be automated yet. Real use cases of the applications are described in the following.

improved Visualization

The shortening of product and process development cycles necessitates efficient methods, tools and techniques for planning complex production systems and facilitating logistics operations. The virtualization of organizational activities creates a need for improved visualization of actual processes. Visualization enables better workspace comprehension and rapid identification of solutions to logistical challenges. Augmented reality virtualization through ARSG represents a particularly promising opportunity with significant potential for business process enhancement. ARSG supported by virtualized workstations, offer speed and flexibility improvements conceptualized and tested in a virtual environment. With applying virtual elements into real applications, the application reduces unnecessary motion, reducing time wastage and minimizing operator confusion through lean information visualization. During order picking operations, information displayed directly in operators' field of view reduces task completion times by eliminating superfluous head and body movements. This technique is also called pick by vision and showed productivity and operational efficiency improvements in many studies (Jumahat et al., 2023, pp. 5–7).The big advantage is, that the information is directly displayed in front of the operator and can automatically be adjusted, just by looking at it. With studies demonstrating reductions in job completion time ranging from 6% to 20% compared to traditional paper-based methods, there is a big benefit in reducing time. This performance improvement stems from enhanced flexibility to execute very different tasks to hands free responsiveness to workers' requirements. ARSG can similarly display shelf content during item location tasks, with data collection and projection on a screen within the user's field of vision. Facilitating visualization of subsequent manufacturing steps and providing essential inspection information on the goods stored in the warehouse. These devices can be equipped with egocentric cameras that naturally follow users' attention, documenting all actions while providing superior information. Specifically, such devices can track users' gaze orientation, thereby augmenting human perception, enabling continuous monitoring and enhancing reasoning and decision-making capabilities. Due to these capabilities ARSG significantly enhances situational awareness, enabling safer and more efficient task execution. (Rejeb et al., 2021, pp. 3756–3758)

Combining these benefits of ARSG, significant differences in picking components time have been measured. (Maio et al., 2023) proposed an AR tool, that supports the training of logistic operators on industrial shopfloors. They identified the operators difficulties and challenges (Maio et al., 2023, p. 5) and improved them with the application of three different conditions, with which participant have been tested (Maio et al., 2023, pp. 6–

10). An extract of (Maio et al., 2023) presents the result of the study in figure 13, showing a comparison of using the three different conditions, paper manual, head-mounted display and handheld device. Head mounted display in this case is comparable to ARSG and " [...] was preferred by all participants and considered more useful and efficient in supporting the operators activities on the shop floor." (Maio et al., 2023, p. 1)

Condition	Average time per component (seconds)	
	Sequential method	Non-sequential method
Paper Manual	36	28
Head-Mounted Display	11	13
Handheld Device	25	17

Figure 13 Average picking time per component using sequential and non-sequential order in three conditions (Maio et al., 2023, p. 14)

Improved interaction

In traditional industries, operation management systems are based on standalone solutions that do not communicate with the surrounding environment, potentially resulting in cost inefficiencies and process delays, due to lack of control over the correctness of execution. With ARSG the smart factory is able to create a production and warehouse environment capable of interoperating and communicating with humans, machines and products. Research has established that modern interactive systems should incorporate design features and capabilities alongside inputs from end users. A fundamental aspect of augmented reality is the ability to interact with objects in real-time. ARSG essentially enables operators to capture photos and videos through interaction via virtual touchscreen or voice control. To lead the operator step by step through his tasks and not only have the capability to check whether the tasks have been done or how good they have been done, but also to monitor the operators 'physical condition. That means, dangerous situations and safety risks can be identified and mitigated with ARSG. Drowsiness and fatigue-detection systems based on wearable ARSG have been developed to enhance road safety, investigating vehicle driver status in real-time through miniature infrared sensor transceivers mounted on the glasses that continuously monitor eyelid voltage signals to alert drivers when tiredness is detected. Such systems help operators avoid dangerous situations by obtaining detailed real-time information regarding drowsiness risk. That leads to the opportunity that ARSG can enable improvements in operator safety, allowing preventive measures to be applied to specific operational tasks and equipment. (Rejeb et al., 2021, p. 3758)

Improved user convenience

User convenience, most frequently denoted as ease of use in academic literature, is considered one of the most frequently employed constructs for evaluating technology adoption. It refers to an individual's perception of the effort required to use a system. The advantages of ARSG in logistics are linked to the perception of their simplicity and convenience. They are perceived to be powerful, efficient and comfortable to wear, which complements their hands-free interface. Unlike tablets and smartphones that can distract workers' attention, ARSG allow users to view information in a hands-free manner while simultaneously performing other tasks. Virtual and real objects can be manipulated using both hands while displayed in the line of sight. Using that technology they can avoid carrying handheld scanners or written documents such as paper lists for order picking. This functionality makes ARSG more convenient and versatile than traditional mobile devices and portable displays. ARSG also advanced the application of Lean Manufacturing Execution Systems by enabling effortless operations and providing intuitive user interfaces for the factory floor. The opportunity to operate and interact hands-free using gestures, touch and voice recognition is particularly attractive for industrial applications, as operators can move and maneuver while easily accessing task-relevant information in a head-up device using either one or two eyes. (Rejeb et al., 2021, pp. 3578–3579)

Improved navigation

ARSG facilitate navigation in diverse logistics settings by constantly providing workers with workspace information, enabling easy site navigation while optimizing paths to required locations. Indoor navigation systems based on ARSG can track pedestrian position using built-in inertial measurement unit sensors, delivering for short-range walks. This position tracking and location determination capability makes ARSG well-suited for logistics environments, enabling operators to navigate even in low visibility conditions by displaying position and routes to targets. ARSG can be applied to order picking processes, allowing efficient item retrieval even in large warehouse settings and factories. The technology provides logistics operators with real-time information during wayfinding, with role-based and context-specific views enabling operators to access the right information at the right time and easily locate physical targets. (Rejeb et al., 2021, p. 3579)

Improved Navigation can not only be applied to workers that are using forklifts or rout trains on the shopfloor. One rising application is driverless transport systems, that are not driving on the ground but flying in the air. Unmanned Aerial Vehicles (UAVs) in industrial environments provides new opportunities for inspection, monitoring, data collection, logistics and maintenance tasks (Mourtzis et al., 2024, p. 3366). Manufacturing facilities often consist of confined spaces with numerous obstacles such as machines, structural elements and storage systems. Additionally, the presence of workers introduces stringent safety requirements. Because GPS signals are unavailable indoors, UAVs cannot rely on conventional localization methods, thereby requiring alternative approaches for safe and efficient operation (Mourtzis et al., 2024, pp. 3362–3363).

To address these limitations, the Augmented Reality–Supported Guidance system provides a comprehensive workflow that leverages AR to improve situational awareness, precision and efficiency in indoor UAV mission planning. First, AR enables engineers to visualize a digital twin of the machine shop, combining real-world environments with virtual overlays such as obstacles, flight paths and waypoints. This immersive visualization offers intuitive spatial understanding and supports safe mission planning without requiring physical presence in the workspace. A key component of the system is accurate environment mapping. UAV-captured images are processed via photogrammetry to generate a 3D point cloud, which is subsequently converted into a voxelized representation that distinguishes free from occupied space. This voxel grid forms the basis for safe navigation and obstacle avoidance. For path planning, the system employs Particle Swarm Optimization (PSO), which computes an optimal flight path by balancing multiple objectives, including shortest distance, data collection requirements, avoidance of occupied voxels and compliance with predefined safety zones. Unlike traditional algorithms such as A*, PSO offers greater flexibility for multi-objective optimization in complex environments. (Mourtzis et al., 2024, pp. 3369–3372).

Through AR interfaces, engineers can define or adjust waypoints either interactively or by entering coordinates. The AR environment visualizes the resulting drone path, waypoint sequence and obstacle distribution, simplifying mission setup and reducing user errors. Before execution, the system simulates the UAV’s flight path within the AR environment, enabling engineers to identify potential issues, such as collision risks or unreachable areas, prior to transmitting the mission plan. Once validated, the final waypoint list is uploaded wirelessly to the UAV controller, reducing the need for manual control and increasing operational safety (Mourtzis et al., 2024, p. 3376). Overall, the ARSG approach enhances safety by incorporating voxel-based safety buffers around both obstacles and UAV operating areas, ensuring that unsafe paths cannot be generated. In addition, empirical results demonstrate that AR-assisted navigation improves mission planning efficiency by approximately 37% compared to manual methods (Mourtzis et al., 2024, p. 3376).

3.5.3 Assembly

XR technologies have been integrated across the product assembly lifecycle. The lifecycle can be categorized into three distinct chronological stages like pre-assembly, on-site execution and post-assembly maintenance. In the pre-assembly phase, XR facilitates process design and simulation. Enabling the virtual optimization of workstation layouts, ergonomic evaluations and collision detection, prior to physical implementation, while simultaneously offering immersive, risk-free training environments that accelerate learning and replace traditional documentation. During on-site assembly, XR augments manual operations through holographic visual guidance and enhances Human-Robot Collaboration (HRC) by visualizing safety zones and robot paths to mitigate accidents and promoting trust. Furthermore, this stage leverages IoT interoperability to display real-time measurement data and supports on site quality control by examining inappropriate geometrics and identifying component errors. Finally, in the post-assembly phase, XR streamlines maintenance execution through visualized disassembly sequences and fault diagnosis, while enabling remote guidance solutions that allow off-site experts to provide real-time, annotated assistance to on-site technicians. (B. Wang et al., 2024, pp. 781–791)

Figure 14 gives a holistic overview of the assembly process including different approaches of applications but also different devices that can be used for AR enhanced assembly.

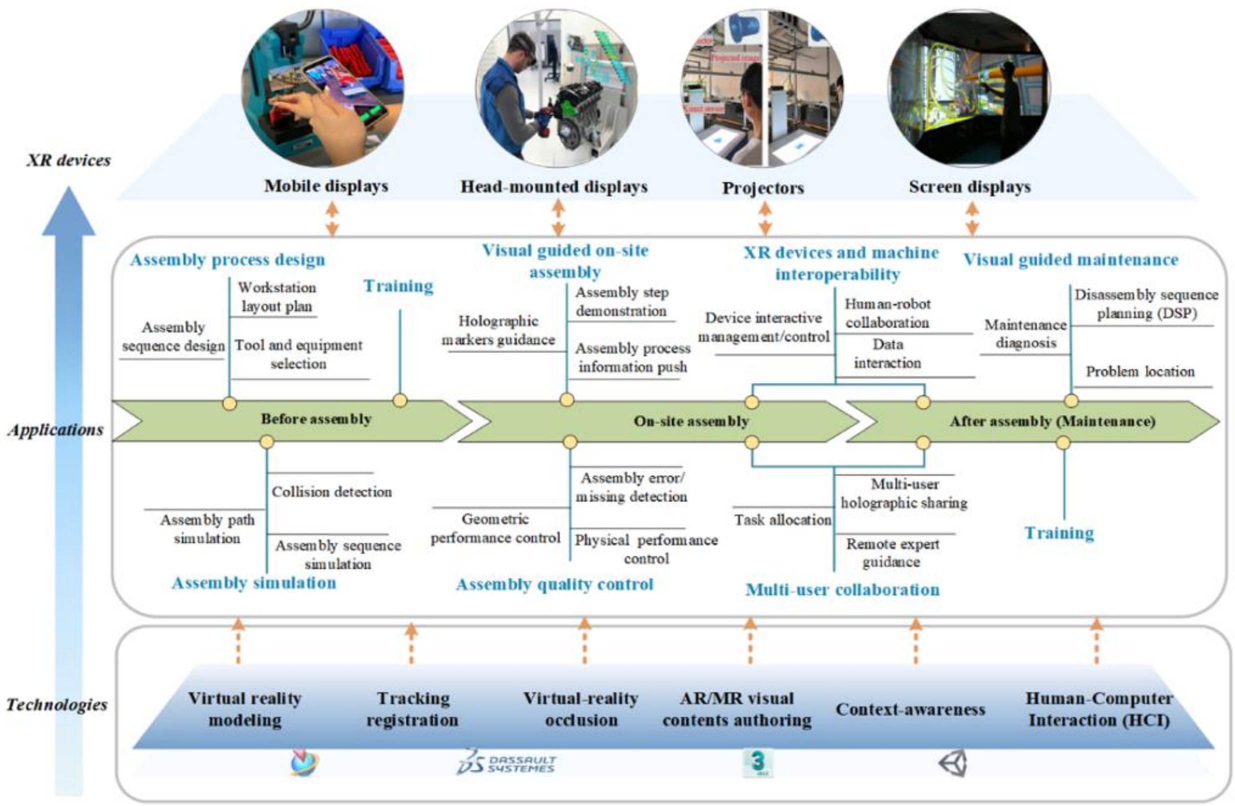


Figure 14 Chronological stages of product assembly (B. Wang et al., 2024, p. 779)

Holistic guidance level approach

(Fang et al., 2025) approached the AR assembly guidance on two certain levels, the factory level and the workstation level. Dividing the guidance into two levels leads to hierarchical guidance which can eliminate the search time between workstation AR models. The concept is guiding employees via optimal paths to the workstations of the value chain on the shopfloor, by optical character recognition, which automatically search the correct guidance for each workstation (Fang et al., 2025, pp. 6–10). In this case they used mobile phones and tablets for the AR application, but the method is transferable to smart glasses. They did not use industrial glasses like the HoloLens 2 because they are "[…] still heavy for long-term usage and expensive for large-scale deployments." (Fang et al., 2025, p. 4)

Training of assembly operators

To enhance the training of new assembly operators, it has been proven that visualizations help to understand the assembly process in a more efficient way. Comparing different test groups in a first assembly study, the test groups using filmed or animated instructions always have been faster then the assembly group using paper instructions (Dalle Mura & Dini, 2021, p. 9; Hořejší et al., 2020, pp. 94334–94335).

The creation of training material for assembly purposes can be enhanced by process mining. Experts who know the procedure in detail, showing how to execute the procedure correctly, this will be analyzed step by step with process mining methods, to automatically label each step and automatically create instructions for untrained workers. (Roldán et al., 2019)

Customization of work instructions

(Marino et al., 2024) proposes the AR assembly guidance tool (ARAGT), which primary object is to streamline product assembly by integrating virtual information into the physical workspace. This approach is a typical AR application to enhance the assembly. The innovative approach is to customize the virtual information from AR to the worker. The worker needs and expectations should be met by corresponding 3D models, custom interaction through different steps, custom graphic design of the work instructions and the opportunity to create virtual annotations for issues that encountered, which can be shared with colleagues (Marino et al., 2024, p. 4942).

Connecting Assembly Guidance with Computer Vision Models

Additional to the customization of the worker AR guidance, there is the opportunity to enhance the guidance by automizing the component control and dynamic AR overlays with computer vision models. (Raj et al., 2025) compared different computer vision models for manual assembly AR assistance and increased the speed of assembly tasks by 14%, by laying the instructions onto the workspace directly on the used components (Raj et al., 2025, pp. 9–10). Looking at separate 2D or 3D models or video guidance or written instructions isn´t needed. Color-coded holograms provide real time feedback which allows deviations during the assembly process to be detected and corrected at an early stage. That leads to a dynamic tracking of assembly and verification of correct execution. To optimize the accuracy and efficiency of hologram registration several detection models have been applied and tested (Raj et al., 2025, p. 3).

3.5.4 Production Scheduling

In complex manufacturing environments, there is a need for tight coordination between machines and human workers. Delays or errors by workers can disrupt pre-optimized scheduling plans. Traditional methods of instruction like paper or screens lack real-time context and integration with the machine's actual status. Therefore (Firme et al., 2024) proposed a custom industrial scheduling environment, whose goal it is to bridge the gap between physical execution and digital scheduling. They are enhancing their workers ‘capabilities with real-time digital information from machine controls and scheduling data bases (Firme et al., 2024, pp. 9–10). Using this system there are different pages available. Each worker owns his specific worker page with its own schedule and assigns tasks for the day. The database page provides step-by-step instructions for manual tasks, which are checked and updated in the central schedule in real time. The machines page overlays real-time sensor data if an AR marker is hit by the physical instance or the virtual instance of the machine (Firme et al., 2024, pp. 10–11).

3.5.5 Summary of Additional Use Cases

Augmented Reality has emerged as a transformative technology across industrial and operational contexts, demonstrating significant potential in multiple application domains. In maintenance operations, AR systems provide technicians with real-time visual guidance and contextual information overlaid onto physical equipment, thereby reducing error rates and improving repair efficiency (Ariansyah et al., 2022; Esen et al., 2025; Sara et al., 2022; Wei et al., 2022; Yazdi, 2024)The technology has proven particularly valuable for training purposes, where immersive AR environments enable workers to acquire complex procedural knowledge and develop technical skills in safe, controlled settings without disrupting actual production processes (Doolani et al., 2020; Kittithanon et al., 2025; Konstantinidis et al., 2023; Lavric et al., 2022; Perno et al., 2025; Roldán et al., 2019; Shkarupylo et al., 2024; J. Silva et al., 2023; Zigart et al., 2023). Furthermore, AR contributes substantially to workplace safety by highlighting hazardous zones, providing real-time alerts and facilitating risk assessment (Dodoo et al., 2025; Esen et al., 2025; Suh et al., 2024, 2024). Beyond human-centric applications, AR supports machine optimization through enhanced visualization of operational parameters and performance metrics (Emporio et al., 2024; Zhang et al., 2025), while simultaneously fostering improved human-machine collaboration by creating intuitive interfaces that bridge the gap between operators and automated systems (Chan et al., 2022; Choi et al., 2025; Das et al., 2025; Enang et al., 2023; Geng et al., 2022; Li et al., 2024; Liao & Cai, 2024). Additionally, AR addresses operator ergonomics by optimizing workstation layouts and reducing physical strain through intelligent task guidance (Mao et al., 2024) (Hijry et al., 2024; Hou et al., 2025; Wen et al., 2023; Windhausen et al., 2024) while monitoring applications leverage AR capabilities to provide operators with comprehensive, real-time oversight of complex industrial processes (Ashok et al., 2023; Schmitt et al., 2025; Shkarupylo et al., 2024). Collectively, these diverse use cases underscore AR's versatility as an enabling technology that enhances operational efficiency, worker capability and system performance across the industrial landscape.

4 Industry Scenario and Technical Context

The Technologie-Transfer-Zentrum (TTZ) Leipheim, operated by Hochschule Neu-Ulm (HNU) in Bavaria, serves as a vital bridge between academic research and industrial practice, with a particular focus on supporting small and medium-sized enterprises in the region. The center's core mission encompasses applied research in automation and Industry 4.0 technologies, providing shared research infrastructure and testing facilities that enable collaboration between university researchers and industrial partners while offering students practical training opportunities on real-world projects. TTZ Leipheim specializes in warehouse automation and logistics optimization, automated storage and retrieval systems, sensor integration and IoT applications, process digitalization, smart manufacturing and data analytics for industrial processes. By granting small and medium-sized enterprises access to advanced research capabilities and specialized equipment that would otherwise be financially prohibitive, while simultaneously providing students with hands-on experience addressing genuine industrial challenges, the center exemplifies effective knowledge transfer from academia to practice and strengthens the competitive position of regional businesses in an increasingly digitalized industrial landscape.

4.1 Pre-Assembly Workstation and Drone Assembly Process

As already outlined in the introduction, the industry use case is to enhance a manual pre-assembly station, with AR and AI. The specific use cases are replacing the pick by light system with a pick by vision system and replacing external cameras with the VR Headset to run an object detection model. The real pre-assembly station is located at the Technology-Transfer-Zentrum in Leipheim, it is illustrated in figure 15. The production covers a manual drone assembly, where most of the components are put on a workpiece carrier at the pre-assembly station, which is the focused location of this thesis.

Figure 15 shows the location of application, an educational laboratory research facility. The system features multi-tiered shelving structures constructed from aluminum extrusion profiles, housing numerous blue plastic storage bins arranged across four distinct levels. Main area is the horizontal conveyer track, which includes the rolling workpiece carrier. Along the track different storage bins are equipped with Kanban cards on their front face. Each bin contains a custom component varying in the components itself but also in its customization, like color. The bins are also equipped with control buttons and LED lights, these indicate the pick by vision system.

Figure 15 pre-assembly station at the TTZ (self-made)

This pick by vision systems works as the following. The Operator can either read the paper instruction on top of the bins or he can look at the screen where the needed component is shown as a picture. The pick by light system triggers the LED at the box where the correct component is in. The Operator has to pick the component out of the box and has to click the LED which is simultaneously also a button, to tell the system the correct box has been used.

On top of the conveyer track is an external camera that detects whether the correct component has been picked. There might be a chance that the wrong component is in the box. The detection runs via a Yolo object detection model, that has been designed customized to the assembly. The camera is positioned vertically upside down from a height of approximately two meters, that leads to the constraint that the component has purposely held below the camera to be detected. In addition, if the workpiece carrier is below the camera and the desired detected component, the detection model struggles. This appears probably because of the number of objects the system detects.

The screen appears green and shows the next component when the correct one has been captured. The LED at each box is not connected to the detection system and must be pressed manually. To combine both systems into one device, the design science method of chapter 2.2 applies in chapter 5.

On top of the workstation a paper instruction is printed out and hanging above head size of the operator. The physical outline is in figure 16 while the virtual version is a multiple page paper, it is saved in Appendix C as a separate file.

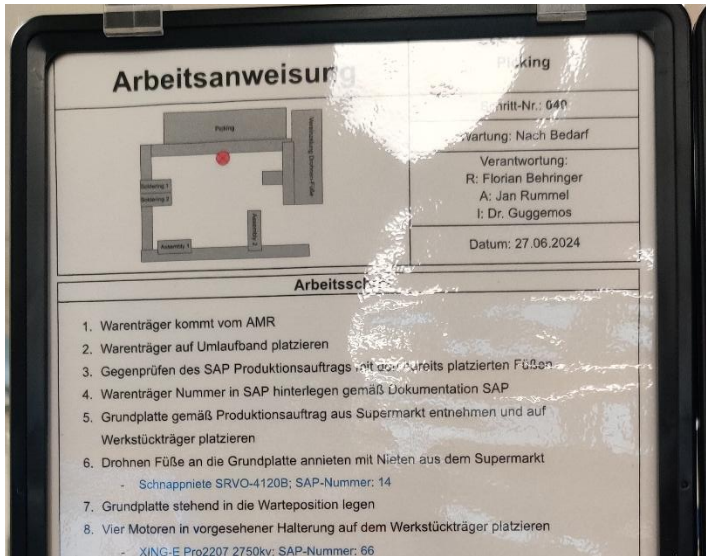


Figure 16 paper work instruction at the pre-assembly station (self-made)

4.2 Technical Challenges of AR Integration

Operators that work for years in exactly this configuration are unfamiliar with AR devices and must readjust themselves to accept new technologies and new working methods. The better the new working method the faster the adaptation by the operator is. Therefore, it must be faster and easier to use, but also fast and easy to learn how to use it. In addition, new technologies like smart glasses might face unanticipated downtimes, which must be recovered fast. This situation might need several test cycles to have a resilient tool that enhances the worker and is also accepted by the worker. These test cycles should be beyond short laboratory uses. Technical challenges that were faced during the artifact development are presented in the limitations, chapter 8.

In 2021 Sahu et al. criticized the current technological situation via hardware and software side of different AR applications. The key aspects are performance and robustness issues, inflexibility and manual dependencies and limited intelligence and data loss. An overview of lack of capabilities is stated in the following citation:

“Current AR applications exhibit several critical limitations:

1) computational algorithms demonstrate excessive resource demands and insufficient robustness across heterogeneous environmental conditions
2) existing AR implementation strategies require substantial manual configuration and are constrained to predetermined environments, one example is camera calibration protocols that have to be specialized
3) object detection and tracking modules necessitate specialized and heterogeneous feature engineering

4) pose estimation techniques lack robustness, consequently introducing variable latency and runtime instability in AR systems
5) data storage architectures exhibit rigidity and limited adaptability
6) virtual object representations are statically embedded within system architecture rather than dynamically instantiated
7) registration methodologies predominantly employ open-loop approaches with feedback mechanisms to the user but not to connected systems
8) current rendering architectures implement oversimplified algorithmic rules for context-aware functionality and utilize information-lossy methods for visualization optimisation, thereby degrading data integrity.

These multifaceted deficiencies underscore the necessity for comprehensive improvements across all architectural and algorithmic components of procedure-based AR systems.” (Sahu et al., 2021, pp. 4904–4905)

Since this statement was proposed in 2021, there has been rapid development in all of the aspects addressed. Still, it is a challenge to evaluate which specific architecture is the preferable for this use case, keeping in mind these challenges have to be solved, to create a working prototype. Several technologies from the literature review in chapter 3 are analyzed and shaped the concept in chapter 6. Iterative design improvements improved the concept and the artifact to handle the challenges stated by Sahu et al.

4.3 Hardware Platform Selection

The original project plan, defined in collaboration with the Technology Transfer Centre Leipheim, envisioned deployment on an XREAL smart glasses device. The XREAL platform offers an optical see-through form factor with significantly lower weight than mixed reality headsets, making it a candidate well suited to extended industrial wear. However, due to cost constraints, delivery time and limited development time, this plan was revised during the early stages of the project. The Meta Quest 3 was selected as the replacement platform.

This substitution represents a deliberate trade-off. The Meta Quest 3 is heavier and more encumbering than the XREAL smart glasses, making it less suitable for full-shift industrial deployment from an ergonomic perspective. However, it offers several advantages that were decisive for a research proof of concept under the given constraints. The device provides a high-resolution color passthrough camera suitable for real-time image capture and transmission to the AI backend, a prerequisite for the object detection pipeline central to this work. Its integrated hand tracking enables spatial interaction with AR content without additional input devices. The Spatial Anchor API allows persistent calibration across application restarts. The Unity-based development environment is well documented, actively maintained and supported by a mature XR interaction framework. Critically, the device is commercially available at a fraction of the cost of industrial AR headsets such as the Microsoft HoloLens or other devices, aligning with the project's objective of demonstrating feasibility on cost-effective, accessible hardware. (Appendix A)

The shift from smart glasses to a mixed reality headset does not alter the research question or the two improvement goals defined for this work. The investigation of how AI-driven AR assistance can enhance manual pre-assembly remains valid regardless of the specific device, as do the objectives of implementing pick-by-vision guidance and AI-based component verification. The ergonomic trade-offs introduced by the heavier form factor are acknowledged as limitations in chapter 8 and the adaptation of the system to lightweight smart glasses is proposed as a dedicated follow-up project in chapter 9.3.

5 AI-Driven AR Pre-Assembly Concept

Prior to developing the system concept, the existing process landscape at the pre-assembly workstation required clarification. At the outset of the project, multiple process definitions coexisted for the same assembly workflow. The established production process relied on a pick-by-light system combined with paper-based work instructions, as described in chapter 4. In parallel, an object detection approach using image-based instructions had been proposed. Notably, the two instruction sets were not identical in content and they differed in their integration with upstream systems. The pick-by-light process triggered a booking in the ERP system each time the operator confirmed a pick by pressing the LED indicator, whereas the detection-based approach included no such booking mechanism. Additionally, an MQTT message architecture is proposed by prior projects that defined dedicated messages for requesting, picking and placing each component. However, this message structure had not been connected to either of the two existing systems and its validity had been questioned during preliminary discussions. No preferred instruction format or system architecture was specified by stakeholders, leaving the conceptual design open to be developed freely within the scope of this work.

The system concept developed here aims to transition from static paper-based instruction delivery to an interactive, context-aware digital format. Paper-based work instructions require the operator to manually reference printed documents during pre-assembly, an approach that presents several well-known limitations, Instructions cannot be updated in real time, physical documents are subject to misplacement and version control difficulties and the operator must repeatedly shift attention between the documentation and the workpiece.

A heads-up display delivered through the AR headset addresses these limitations by the opportunity of presenting work instructions directly within the operator's field of view. The operator receives the current version of each instruction without leaving the workstation. The need to shift attention between a separate document and the assembly task is eliminated. The digital format further enables customization based on operator preferences, language, text size, visuals and layout can all be adjusted within the AR development environment, supporting diverse workforce requirements. While the potential relevance of such customization for operators with limited literacy might be a consideration that may apply in certain manufacturing contexts in developing regions it is not actively addressed in the present concept.

The concept spans multiple software domains, from a real-time three-dimensional development platform for AR content creation to machine learning models that require dedicated training on component-specific datasets. All components demand sufficient computational power not only to execute their core functions but also to ensure a fluid and comfortable operator experience. This constraint directly informs the architectural decisions regarding processing distribution discussed in the following sections.

The subsequent chapters detail the concept across several areas. The general data flow architecture will be visualized, the visualization of virtual information within the operator's field of view, intuitive transitions between assembly steps and the integration of machine interfaces and computer vision systems will be presented. The AR content, work instructions, spatial highlights and three-dimensional component models, is developed in Unity, a cross-platform development engine widely adopted in both commercial and academic contexts for creating interactive three-dimensional, virtual reality and augmented reality applications (*unity.com*, 2026). The communication architecture is based on MQTT, selected both for its suitability as an industrial messaging protocol, as discussed in chapter 3 and for the practical reason that an MQTT broker was already available at the Technology Transfer Centre for other testing purposes.

5.1 Device Independent Concept Basis

Independent on which device is used to create such an AR assistance, three basic layers have been established during the concept creation work. The three layers are explained in the following sections and can be applied to any device that is following this or similar use case. Creating a concept for a different use case, changes the application of the following three layers.

5.1.1 Hardware Layer

Digital work instructions are structured as modular, sequential steps with associated multimedia content. Based on the operator´s preferences or expertise available with different versions, which can be customized for each operator. The focus of customization lays in the areas of text, images, 3D models, videos, animations and chatbots. Chatbots are added because in training status or any status of misunderstanding, the work instruction could be enhanced by a Large Language Model, explaining the different steps, so no additional person is needed. Especially in training purposes it can be used to explain different situations, like nonconformance, to mediate the experience gained of previous operators to new operators by the trained AI.

No matter what is included in the instructions, the visualization of the instruction itself differ from the Degree of Freedom (DoF). DoF plays here an important role for the spatial in the room. Either 3-DoF or 6-DoF are applied, which is restrained to the hardware that is used. In 3-DoF the instruction follows the user because no transactional movement can be recognized, the headset doesn´t understand the location in the room and adjusts the visualization always to the rotational movement of the user. With 6-DoF the visualization can be put into a specified position and stays there when moving in the room. To use multiple room locations simultaneously, 6DoF is absolutely preferred in this concept.

6DoF creates the opportunity to build a digital twin of the work station which stays anchored in the room where the real work station might stand. If 3DoF is applied, the work station twin would be useless, because the twin follows the user and stays always in the same distance. It would not be possible to actively show the user from which bin to pick from. That's leading to non-fulfillment of the pick by vision system.

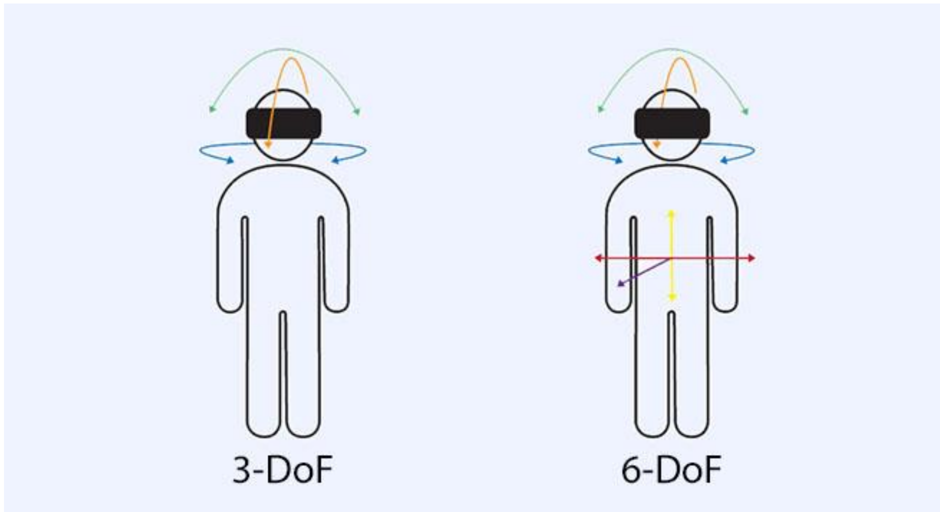


Figure 17 The difference between 3-DoF (rotational movement) and 6-DoF (rotational and translational movement) with a VR headset (Degrees of Freedom (DoF), 2025)

5.1.2 AR Engine Layer

In this case Unity as an AR engine layer is responsible for rendering instructions, managing user interaction, tracking work progress and spatial registration. The focus on this layer is to give the user smooth usage of the instructions, no matter if it is on how many frames per seconds are used or how the transition between contents works but also how the instruction box is spatial in the room.

To visually direct an operator's attention to the correct storage bin, various highlighting techniques can be applied within an AR environment. Possible approaches include colored overlays or transparent fill effects that tint the bin area in a specific color, pulsing light effects that catch the eye through rhythmic brightness changes, projected arrows or directional indicators pointing toward the target location. Spatial labels and icons floating above the bin can also be implemented. Each method offers different trade-offs between visibility, occlusion of the bin's actual contents and cognitive load.

The transition of and between instruction contents can be automated but should also remain manually controllable to accommodate varying operational contexts and individual user preferences. The opportunity to redo mistakes by the automated system should also be available with a manual system. A hybrid interaction mode that balances autonomous progression with user-initiated control provides optimal flexibility across diverse manufacturing scenarios and operator skill levels. Depending on the control, a different asset must be applied in unity. Each asset contains different software and is continuously developing by company developers and open-source coders. Conceptual manual transition controls are the following.

Virtual Button Interface

Spatially anchored virtual buttons rendered within the AR environment allow operators to explicitly navigate between instruction steps. These buttons can be activated through gaze-and-dwell selection. Operators fixate their gaze on a button for a configurable duration, triggering the action without physical contact and without using their hands but only their direction of view. Also common are air-tap gestures, hand tracking enables operators to perform pointing gestures toward virtual buttons and execute a tapping motion.

Gesture-Based Navigation

Hand gesture recognition enables intuitive, contactless control like horizontal swiping gestures. Swiping left and right with a certain gesture to navigate forward and backward through instruction sequences. Vertical swipe gestures like up and down to expand or collapse detailed information panels. Pinch-and-zoom gestures scale 3D models or technical diagrams for closer inspection. Palm-up/palm-down orientations toggle between instruction overview and detail views. The number of opportunities on how to exactly work with instructions are very high.

Gesture control proves especially valuable when operators' hands are visible but occupied with light tasks, allowing navigation without releasing tools or components. However, gesture recognition requires adequate lighting conditions and unobstructed camera views, limiting applicability in certain industrial environments.

Voice Command Interface

Speech recognition provides the most hands-free interaction option, with simple command vocabulary like "Next," "Previous," "Repeat," "Pause," "Help," "Complete" and Natural language queries like "Show me the torque specification," "Where does this component go," "What tools do I need". The opportunities and specialized use cases are endless.

Voice interaction excels in scenarios where operators' hands and field of view are fully occupied, such as during assembly operations requiring bimanual manipulation. Challenges include background noise in industrial settings, pronunciation variations across diverse workforces and cultural preferences regarding speaking aloud in shared workspaces. Noise-canceling microphones and speaker-dependent voice model training can mitigate these limitations but must be approved in the specific setting.

5.1.3 Data Integration Layer

Backend connectivity to enterprise systems like Manufacturing Execution System (MES), Enterprise Resource Planning (ERP), Feedback Systems etc.) for instruction retrieval, versioning, completion logging and general data storage + data exchange regarding from the shopfloor. The focus on this layer is to integrate the worker into the dynamic capabilities of planning and documenting systems.

Connecting the work instructions to a MES allows the production scheduling to dynamically adapt to real-time shopfloor conditions. As operators complete instruction steps through the smart glasses interface, the MES receives immediate feedback on task progress, enabling automatic updates to production schedules, resource allocation and downstream work center planning. This bidirectional communication creates a closed-loop system where scheduled work orders automatically trigger the delivery of appropriate work instructions to operators, while completion data flows back to update order status, inventory consumption and capacity utilization metrics.

Connection to Enterprise Resource Planning systems provides the work instruction platform with access to master data including bill of materials (BOM), product variants, quality specifications and customer-specific requirements. It enables complete traceability by automatic documentation. Automated logging of operator actions, components used, tools employed and timestamp data creates a comprehensive digital thread linking each produced unit to its manufacturing history. This data satisfies regulatory compliance requirements and enables root cause analysis in case of quality issues. Additionally, it can be connected to Enterprise Warehouse Management (EWM) systems to track current storage loads at the operator´s station and automatically order new material from the supermarket. With that a digital and automated Kanban system can be used while freeing the operator up from the communication task of requesting new material.

Connection to Feedback Systems allows the operator to give feedback on certain guided steps. The idea is to give the operator the opportunity to be involved in the continuous improvement process during his work shift. To achieve that, the operator should either be able to mark instruction steps or to directly give feedback to the steps. These marked ones should be deployed in a feedback loop with engineering and industrialization departments to capture frontline knowledge and continuously improve the production process. Additionally, the time tracking of each step should be automated to not only wait for feedback if a step is not working correctly but also to recognize bottlenecks of the value stream automatically to enhance process efficiency.

5.2 Pick by Vision Virtual Scene Alignment Options

The system aims to transition from traditional paper-based instruction delivery to an interactive, context-aware digital format that enhances operator efficiency, reduces errors and enables real-time data integration. Work instructions in this case exist as static paper documents that operators must reference manually during pre-assembly procedures. As already discussed in Chapter 3.5 this approach presents limitations including lack of real-time updates, potential for misplacement, difficulty in version control and requirement for operators to shift attention between documentation and workpiece.

To fulfill the pick by vision system the operator has to be guided visually from where to pick the components. In this case all types of components are stored in separate bins. These bins should virtually light up or be marked whenever they are the desired one but also stay in the correct position, virtually and in real life, to not create any confusion at the work station. In that case the 6DoF technology is applied.

The current AR system architecture transmits camera images from a Meta Quest 3 headset via MQTT to a Python-based object recognition pipeline (Chapter 5.3 & 5.4). To ensure accurate spatial alignment of virtual AR content with the physical workspace, a marker-based alignment solution is required. Three distinct approaches have been evaluated based on technical feasibility, integration complexity and operational reliability.

5.2.1 ArUco Marker Detection via Python Pipeline

This approach leverages the existing system architecture by integrating ArUco marker detection into the current Python-based processing pipeline. ArUco markers are square fiducial markers developed for camera pose estimation, with built-in support in OpenCV. The implementation workflow involves affixing a printed ArUco marker to the workspace, detecting this marker within the Python script that already processes incoming camera images, computing the 6DoF (chapter 5.1.1) relative to the camera, transmitting this pose data via MQTT to the Quest headset and adjusting all AR content in the Unity AR application on the headset environment accordingly. (*OpenCV: Detection of ArUco Markers*, 2026)

The primary advantage of this approach is seamless integration with the existing MQTT-based architecture, requiring no fundamental structural changes. OpenCV's ArUco detection algorithms are computationally efficient and robust under varying lighting conditions and viewing angles. The pose estimation provides complete spatial information including translation and rotation in three-dimensional space, enabling precise alignment. However, this approach introduces additional network latency since pose data must traverse the MQTT communication path. The accuracy depends on camera calibration quality and marker size relative to viewing distance. (*OpenCV: Detection of ArUco Markers*, 2026)

5.2.2 QR Code Recognition Directly on Quest Headset

The second option processes marker recognition entirely on the Quest headset using the Passthrough Camera Access API and a C# barcode detection library such as ZXing.Net, for which samples are provided by the Meta development department. The QR code would be detected directly from the camera feed within Unity, eliminating the need for external processing. Recognition results would be immediately available to the AR application without network communication. (developers.meta.com, 2026)

This approach offers significantly reduced latency since no external communication is required for marker detection. The implementation remains self-contained within the Unity application, simplifying the overall system architecture. However, standard QR code detection libraries provide only two-dimensional position information within the image frame. Extracting three-dimensional pose information including depth and orientation requires additional calibration procedures or assumptions about marker size and camera parameters. The computational load shifts to the Quest hardware, which may impact performance if combined with other intensive AR operations. Furthermore, ZXing and similar libraries are optimized for barcode scanning rather than continuous pose tracking, potentially limiting tracking stability during head movements. (developers.meta.com, 2026; *Oculus-Samples/Unity-MRUtilityKitSample*, 2025)

5.2.3 Meta Spatial Anchors

Meta's spatial anchor system enables the Quest headset to establish persistent spatial reference points within a physical environment. The system uses the headset's simultaneous localization and mapping capabilities to recognize previously scanned spaces and restore spatial anchors across different sessions. Once configured, virtual content automatically aligns with the physical environment without requiring visible markers. (*Spatial Anchors*, 2025)

The elimination of physical markers represents a significant operational advantage, avoiding issues related to marker placement, occlusion or damage. Spatial anchors persist across sessions, maintaining alignment without repeated calibration. The system operates independently of external processing, reducing infrastructure dependencies. However, spatial anchors require initial room scanning and anchor placement for each deployment location. Recognition reliability depends on the environment containing sufficient visual features and remaining relatively unchanged. Performance may degrade in feature-poor environments or when significant physical changes occur. This approach also introduces dependency on Meta's proprietary spatial computing platform. (*Spatial Anchors*, 2025)

5.2.4 Comparative Analysis

Evaluating these options across key criteria reveals distinct trade-offs. Regarding integration complexity, Option 1 requires minimal architectural changes since it extends the existing Python pipeline, whereas Option 2 necessitates new Unity-side implementation and Option 3 involves configuring Meta's spatial computing features. Latency considerations favor Options 2 and 3, which process locally on the headset, over Option 1's network-dependent approach. For spatial accuracy, Option 1 provides complete six-degree-of-freedom pose information, Option 3 offers comparable accuracy after proper calibration, while Option 2 delivers limited spatial information without additional processing.

Infrastructure requirements differ substantially. Option 1 utilizes the already-deployed MQTT communication system, requiring only software extensions. Option 2 operates entirely within the Unity environment with no external dependencies. Option 3 requires no additional infrastructure but depends on Meta's cloud services and spatial mapping capabilities. Maintenance and reliability also vary, option 1 requires ensuring marker visibility and condition, option 2 faces similar marker maintenance challenges, while option 3 eliminates marker-related issues but requires periodic room rescanning if the environment changes significantly.

During Implementation the following issue occurred which made the usage not sufficiently good. When the Meta Quest 3 detects an ArUco marker, it estimates the marker's position and orientation in 3D space based on the camera's current viewpoint. The problem is that this estimation isn't perfectly consistent, each time the marker is tracked from a different angle or distance, the calculated pose of position and rotation varies slightly. These small deviations accumulate or shift depending on the user's viewing angle, lighting conditions and camera distortion at the edges of the passthrough image.

In practice, this means that if you look at the marker from the left, the virtual overlay might appear to sit a few centimeters in one direction and when you move to view it from the right, it shifts noticeably. The coordinate frame that the headset derives from the marker isn't stable, is switches between observations that differ from each other. Depending on the detection and update rate, an uncomfortable feeling appears due to the automatic change of the environment.

For rough spatial alignment, e.g., placing a general information panel near a machine, this level of accuracy is acceptable. But for a workstation scenario, where virtual work instructions need to align precisely with specific physical components, tools or assembly positions, even a few millimeters of inconsistency make the overlay unreliable and potentially confusing for the worker.

Before concept execution Option 1 emerges as the most suitable approach for this specific use case. The existing MQTT-based architecture already handles image transmission and processing, making ArUco marker integration a natural extension rather than a fundamental redesign. OpenCV's mature ArUco implementation provides reliable pose estimation with minimal additional computational overhead. The system can be deployed with straightforward modifications to the existing codebase and a single printed marker affixed to the workspace.

Still manual adjustment is necessary due to the imprecise alignment of the marker. Depending on the viewing angle the anchors are not set precisely set in the operators' environment leading to a not sufficient solution in practice even though it was in conceptional theory.

5.3 Object Detection Deployment Options

In this chapter the focus lays on the one hand on the location where the object recognition model is running and on the other hand the type of dataset which the model has been trained with. For the location of the Object Recognition Model, four opportunities have been evaluated based on the circumstances of the use case and its factory.

5.3.1 Headset Deployment

In this scenario, the object recognition model runs entirely on the headset itself. All processing steps, including image acquisition and inference, are performed locally on the device.

Characteristics & Advantages

- Fully edge-based processing, no continuous network connection required.
- Low latency since no channels to external systems are needed.

Limitations

- Power and thermal constraints: The available computing power is limited by the headset's battery and thermal design. Long-running, compute-intensive models can lead to throttling or reduced battery life.

- Hardware fragility: The headset is a single point of failure. If the device is damaged or malfunctions, the entire recognition pipeline is affected.
- Limited model complexity and scalability: The device cannot efficiently host large or multiple advanced models, making scaling to more complex use cases difficult.
- Main implication: Due to these constraints, this option is primarily suitable for prototyping, demos and testing, but not for scalable production environments.

5.3.2 Not Tethered Hardware Deployment

MQTT Data Stream

In this architecture, the headset performs local decoding or preprocessing, then transmits relevant information via an MQTT-based messaging architecture to the VM, where the object recognition model is executed. The exact technical explanation is in the subsections of chapter 5.4.

Characteristics & Advantages

- Modular architecture: The system is logically divided into components (headset, message broker, processing services on the VM), which can be developed and scaled independently.
- Customizable and extensible: The MQTT-based setup allows easy integration of additional services like logging, analytics, dashboarding or integration into MES/ERP systems.
- Scalable: Multiple headsets and models can be handled by scaling the VM resources and/or deploying containerized services.

This option fits well with typical manufacturing requirements, where robustness, scalability and integration into existing IT/OT landscapes are crucial. Due to its modular and scalable nature, this architecture forms the main focus for productive manufacturing use cases and is favored in this project by the project stakeholders.

Video-Livestream Data Stream

In this variant, the headset streams its camera feed to a virtual machine via the horizon meta link application (*Meta Horizon Link App*, 2026). The VM decodes the livestream frames and performs the object recognition inference. The decoding part is exactly the same as from the MQTT architecture but is fully executed locally on the windows machine.

Characteristics & Advantages

- Processing is offloaded to the VM, allowing the use of more powerful hardware and larger models.
- The headset acts mainly as a sensor and display device.

Limitations

- High bandwidth requirements: Continuous livestreaming of image data requires a stable and high-throughput network connection. Network issues directly impact recognition performance.

- Dependency on vendor ecosystem: The solution relies on Meta/Headset-specific streaming apps and APIs, introducing dependencies on proprietary components and their update cycles.
- Architectural complexity: The overall system architecture of streaming, decoding, processing and feedback is relatively complex, increasing integration and maintenance effort due to the missing access to the livestream data.

This approach is mainly interesting if the livestream itself provides additional value like remote assistance, recording or monitoring. For pure object recognition use cases, the overhead and strong coupling to streaming infrastructure of the headset provider itself can be risky due expiring support (*Microsoft Ends Support for HoloLens*, 2024) or very high and specific hardware requirements (*Requirements for Meta Horizon Link*, 2026).

OPCUA Data Stream

In this architecture, the headset communicates with the backend via OPC UA instead of MQTT. The object recognition model runs on a VM or on-premises server and the headset exposes or consumes data through OPC UA endpoints.

Characteristics & Advantages

- Industrial standard protocol: OPC UA is a widely adopted standard in industrial automation and is already present in many manufacturing environments like PLC, SCADA, MES.
- Rich information modelling: OPC UA provides a structured information model including nodes, types and methods, allowing semantic descriptions of objects, states and events instead of just raw messages.
- Security built-in: The protocol includes standardized mechanisms for encryption, authentication and authorization, fitting certain OT security requirements.
- Integration into existing shopfloor landscape: Since many machines and systems already speak OPC UA, the headset-based solution can be integrated into existing OT/IT systems without introducing a parallel messaging infrastructure.

Limitations

- Higher integration effort on the headset side: Direct OPC UA support on a mixed reality headset is usually limited. Often an additional gateway or service is needed that translates between the headset's APIs and OPC UA, increasing implementation effort.
- Less lightweight than MQTT: For simple event-driven messaging, OPC UA is typically more heavyweight and complex to configure and maintain than MQTT, especially for cloud-based or highly distributed scenarios.
- Scalability and flexibility: While OPC UA can scale, dynamic scaling like containerized cloud environments and multi-tenant scenarios are often simpler with MQTT-style broker architectures.
- Developer ecosystem: The tooling and libraries around OPC UA are more focused on classical industrial automation than on fast-moving backend/cloud stacks, which can slow down development compared to an MQTT-based microservice architecture.

OPC UA is particularly suitable when the primary goal is tight integration with existing machines and OT systems and when an OPC UA infrastructure is already in place. For flexible, cloud-native or rapidly scalable architectures, MQTT often offers a leaner and more modular approach. A hybrid approach is useful in practice with internally, services communicate via MQTT or similar messaging, while a dedicated gateway exposes relevant data and commands as OPC UA to existing shopfloor systems.

Lastly it can not be ruled out to combine both technologies of MQTT and OPCUA. Depending on further interoperability and the connection of machines to retrieve their data it can make sense to combine the strengths of both protocols. (Sai Guptha et al., 2025)

5.3.3 Tethered On-Premise Hardware Deployment

In this scenario, dedicated on-prem hardware, e.g., an industrial PC or edge server is directly connected to the headset. The object recognition runs on this hardware, not in the cloud.

Characteristics & Advantages

- Local processing with potentially powerful hardware, without sending data to the cloud.
- Potentially reduced latency compared to remote cloud deployments.

Limitations

- Restricted mobility: The physical connection or proximity requirement between headset and hardware can limit the user's freedom of movement.
- Hardware constraints: The solution depends on specific hardware components, which may be difficult to replace, upgrade or replicate across sites.
- Costly scalability: Scaling the solution typically requires additional dedicated hardware at each location or workstation, leading to higher investment and maintenance costs.
- Limited capabilities in practice: Especially when suitable hardware is not yet available or standardized, this option is often limited in applicability and flexibility.

Local and powerful but limited by motion and hardware restrictions, with high costs for scaling; practical use is constrained where suitable hardware is not broadly available and might change quickly due to dynamic software developments.

5.3.4 Pre-Trained Object Detection Model

In the first stage of the project, a pretrained YOLO model (Redmon et al., 2016) is deployed to evaluate object detection performance under controlled conditions, using publicly available weights provided by the Ultralytics framework. The model runs within a Python-based integrated development environment (IDE) on a virtualized Windows machine, which offers a reproducible and isolated setup for experimentation and integration with the previously described MQTT-based image streaming pipeline.

The virtual Windows machine hosts a Python IDE and provides all necessary dependencies in form of public available libraries for computer vision and deep learning.

These libraries can be installed locally into the python project and enable a quick setup with only a few lines in terminal and main script. They include the Ultralytics YOLO package for loading and executing the YOLO11n model, additionally opencv-python for image and video handling, numpy for numerical operations (van der Walt et al., 2011) and, if available, GPU-accelerated computation via Compute Unified Device Architecture (CUDA)-enabled PyTorch (Paszke et al., 2019). CUDA is a Nvidia specific GPU driver toolkit that can only be used on NVIDIA GPUS at a certain current version (*CUDA - NVIDIA*, 2025). It is used for development purposes and usually to enhance heavy computational usage. The library environment is typically managed via a virtual environment or Conda environment to ensure that package versions remain consistent and the experimental setup is reproducible. (Mukherjee, 2026)

The YOLO11n model itself is used in its pretrained form, that means with weights that have been trained by Ultralytics on a large, generic object detection dataset. This pretrained model is loaded directly by referencing the corresponding weight file, here for example yolo11n.pt, through the Ultralytics API. The model architecture follows the standard YOLO design , which consisting of feature extraction, feature aggregation at multiple scales and predicting bounding boxes, objectness scores and class probabilities. YOLO11n is one of the smallest “nano” variant, optimized for real-time performance and reduced resource usage, making it suitable for deployment on modest hardware while still providing competitive detection accuracy (Ultralytics, 2026b).

For inference, the virtual machine receives input images either from static files, like. test datasets on the local file system or via the MQTT-based streaming pipeline from the Meta Quest 3 headset. In both cases, the images are processed using OpenCV to ensure the correct color format and resolution required by YOLO11n. The images are passed to the model’s inference function, which returns detection results in the form of bounding boxes, class labels and confidence scores. These outputs are subsequently post-processed and visualized by drawing bounding boxes and labels onto the original images. The annotated images can be displayed in real time in a GUI window or stored for later analysis and relevant metadata such as inference time, number of detections per frame and confidence thresholds are logged for evaluation. Following the concept line, it makes sense to test the object recognition with a local dataset and an external cable camera first, to ensure the computational capability of the virtual machine. After ensuring solid execution, the MQTT pipeline with the encoded image stream should be applied so that the external camera is replaced by the Meta Quest camera.

An important aspect of this architecture is the separation of concerns between data acquisition and model execution. The virtual Windows environment is solely responsible for model inference and analysis, while the Meta Quest 3 and MQTT broker handle frame capture and transmission. This modular structure allows the pretrained YOLO11n model to be evaluated independently of the sensing hardware, using both live camera streams and offline datasets. It also provides a clear baseline for comparing future experiments, such as retraining the model on domain-specific data or replacing YOLO11n with alternative architectures. By starting with a pretrained model, the project can focus initially on validating the end-to-end pipeline, measuring latency and detection quality and establishing a reference performance level before moving on to more complex tasks such as individual dataset creation and training.

5.3.5 Creating Individual-Trained Object Detection Model

In the next stage, the focus shifts from evaluating a pretrained YOLO11n model to training a customized model on a domain-specific dataset. The objective is to adapt the generic capabilities of YOLO11n to the specific objects and environmental conditions present on the shopfloor. This step enables the model to recognize classes that are not covered in the pre-trained set. To cover these, they have to be added in a specific way, which will be explained in this chapter.

The overall software and hardware environment remains consistent with the setup used for the pretrained model. Training is conducted on the same virtualized Windows machine that hosts the Python IDE and the previously installed public libraries for computer vision and deep learning. This includes the Ultralytics YOLO package, opencv-python, numpy and if available, CUDA-enabled PyTorch (Paszke et al., 2019) for GPU acceleration. Maintaining the same environment and dependency configuration ensures reproducibility and allows direct comparison between the performance of the pretrained and the custom-trained models. As before, a virtual environment or Conda environment is used to manage library versions and encapsulate the experimental setup. Using this structure both datasets can be stored on the virtual machine and individually combined or separated in usage. Switching between both only needs a restart of the script and the edit of the script line stating the name of the dataset. (Mukherjee, 2026)

The preparation of a custom dataset involves several steps. First, images must be collected that realistically represent the target domain, such as photographs or video frames from the shopfloor, including the objects of interest under different lighting conditions, viewpoints and occlusions. This means the assembling components have to be photographed for couple of hundred times with different backgrounds. Essential backgrounds would be holding it in the hand or being in the workpiece carrier where the operator has to put it in. These images can originate from the external cable camera used in the initial tests, from the Meta Quest 3 camera via the MQTT pipeline or from other relevant sources within the production environment. Because of the given resolution that comes from the Meta Quest camera and the MQTT transfer, it is preferred to feed the model with pictures of that origin. Once the raw image data is collected, it must be annotated according to the YOLO format, which requires a text file per image containing the bounding boxes and class labels for all visible objects. Annotation can be performed using publicly available labeling tools that support YOLO-compatible exports. During this process, a clear class definition and naming convention must be established to ensure that the model can distinguish between different object categories in a meaningful and consistent way. This is a manual step that has to be done carefully and correctly to ensure that the objects are labeled but also detected correctly. Depending on the picture and the number of components this step takes time. To execute this task Label Studio has been applied. (*Label Studio*, 2026; Ultralytics, 2026a)

After annotation, the dataset is organized into a directory structure that follows the conventions expected by the Ultralytics YOLO training pipeline. Typically, the data is split into separate subsets for training, validation and optionally testing. This split is crucial for evaluating the generalization performance of the model and for monitoring overfitting during training. Additionally, a configuration file in YAML format is created to describe the dataset to the YOLO training script. This configuration file contains paths to the train and validation images, the list of class names and the number of classes. By centralizing this information, the training process can be parameterized and repeated with different datasets or modified splits without changing the training code itself.(Ultralytics, 2026a)

The actual training procedure is orchestrated through the Ultralytics YOLO API in Python. In contrast to the inference-only workflow described previously, the model is now initialized in training mode, either starting from scratch or by fine-tuning the existing pretrained weights such as yolo11n.pt. Fine-tuning is often preferred, as it allows the model to retain general features learned from large-scale datasets while adapting to the specific characteristics of the new domain. Training hyperparameters such as learning rate, batch size, number of epochs, image resolution and augmentation settings are defined in the training script or in an additional configuration file. Data augmentation can include operations such as random cropping, flipping, scaling and color adjustments, which help the model to become more robust to variations in the input data and to avoid overfitting to the limited set of training images.(GeeksforGeeks, 2025)

Throughout training, the Ultralytics framework provides metrics and intermediate results that are essential for scientific evaluation. These include loss values for classification, localization and objectness, as well as performance metrics such as mean Average Precision (mAP) on the validation set (Padilla et al., 2020, p. 239). The training process can be monitored in real time in the console or via generated log files and training curves. By analyzing these metrics, it is possible to assess whether the model is converging, whether it is overfitting or underfitting and how changes in hyperparameters or dataset composition affect performance. Checkpoints of the model weights are saved at regular intervals or when improvements on the validation metrics are observed, enabling the selection of the best-performing model at the end of training. One approach therefore is the dropout technique, that applies during training (Srivastava et al., 2014).

From an architectural standpoint, the training phase complements the previously described inference setup while following the same modular principles. The virtual Windows environment remains the central location for computation, but now it is responsible not only for model inference and analysis, but can also be used for learning and optimization. The device camera plays a supportive role during dataset creation by providing realistic image data but is not directly involved in the training loop. This separation ensures that the training procedure can be repeated, extended or modified independently of the live streaming infrastructure. Once a custom-trained YOLO model has been obtained and evaluated using validation and test data, it can simply replace the pretrained model in the existing inference pipeline by loading the new weight file instead of the original yolo11n.pt that has been used for testing purposes.

5.4 Desired Data Exchange Architecture

Since there could have been different machine protocols and data communication channels, the most useful for this use case is the not tethered MQTT communication channel due to the provision of a cloud broker, which can directly be used without any further configuration. This option has been chosen because the infrastructure of this option is already given and no further costs would be created, therefore was favored to be tested by stakeholders of the TTZ.

5.4.1 General MQTT Message Communication

This architecture is used to create a channel for sending messages between the headset and the connected server, which is for example the corresponding script or MES system guiding the steps in a certain order.

The proposed architecture establishes a secure, real-time communication framework, which uses the MQTT architecture. The MQTT architecture is next to OPCUA, a current machine interface that is used in today's industry. MQTT always needs a client and a broker. The client can publish and subscribe to a given topic and to a configured broker to receive the topic´s messages. This design enables bidirectional data exchange, allowing the client to both send telemetry data and receive commands, content updates and synchronization signals from remote systems or other connected devices. The configuration of the broker consists of the port, the host name and in manufacturing also in encryption. The encryption of this tunnel contains, username and password and valid certificates to trust the broker and the client.

At its core, the architecture consists of three primary components working in concert. The AR headset serves as the client device, running a Unity application that integrates an MQTT client library written in C#. This client maintains a persistent connection to the HiveMQ broker, which acts as the central message routing hub. The HiveMQ broker is a cloud-based broker that is available from anywhere, also outside the shop floor environment. Therefore, the broker itself is configured with Secure Sockets Layer (SSL)/Transport Layer Security (TLS) encryption to ensure all communication remains secure and protected from interception. Beyond these core elements, the architecture can accommodate additional backend services, analytics platforms, content management systems or even other AR devices that connect to the same broker, creating a networked ecosystem of interconnected applications. In this scenario we considered the opportunity of connecting an Erp, MES or a Dashboard application. A general visualization of the message direction can be viewed in figure 18.

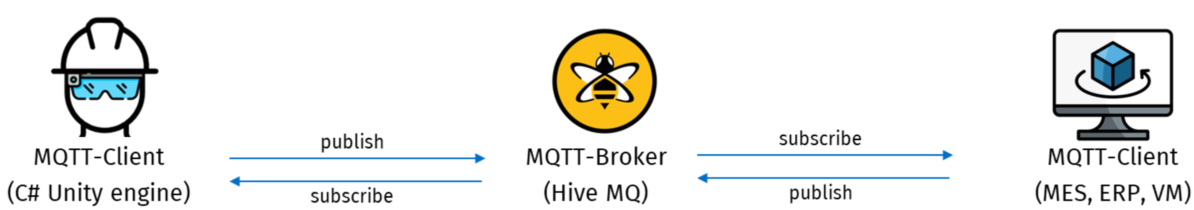


Figure 18 Basic one bidirectional MQTT Channel Architecture example (self-made)

The focus for future work on this use case is to establish a holistic MQTT message architecture that is coherent with the BOM and the work instruction of the product. The whole message architecture is available in Appendix F. The headset publishes a message when starting the instruction step and publishes again one when the step is finished. The documenting system, for example the MES subscribes the messages and publishes messages when a step is done. The headset subscribes to this to start the next step as soon as it got the message from the MES. Both sites have to be programmed to be triggered to publish a message on a certain topic when they receive a message of a certain topic.

The MQTT architecture for Unity-based AR headsets with HiveMQ provides a robust, secure and scalable foundation for building connected AR experiences. By leveraging industry-standard protocols, implementing comprehensive security measures and carefully addressing Unity's specific requirements, this design enables real-time communication between AR devices and cloud services while maintaining the performance necessary for immersive experiences. The architecture's flexibility supports evolution from simple single-device applications to complex multi-user collaborative systems without requiring fundamental redesign, making it an excellent choice for both current development and future expansion.

5.4.2 Decoded MQTT Image Communication

The proposed system implements a lightweight, MQTT-based pipeline for transmitting camera frames from a Meta Quest 3 headset to a remote object detection service running YOLO11n on a Windows machine. The primary objective is to enable near real-time object recognition at an initial target frame rate of approximately 5 frames per second (fps), using an architecture that is sufficiently flexible and scalable for further experimentation and optimization.

At the sensing side, the Meta Quest 3 is used as a mobile camera platform. A Unity application running on the headset accesses the device’s camera via a Unity Camera object and renders each frame into a Render Texture. This render target is then copied into a Texture2D, which allows the frame to be encoded into a compressed image format. For efficient transmission over the network, the image is encoded as a JPEG with a configurable quality factor and the resolution is deliberately kept modest, for example at 320×240 pixels in the initial setup. This balance between resolution, compression quality and frame rate is chosen to keep bandwidth consumption low while still providing sufficient visual information for the YOLO11n object detection model. High bandwidth consumption leads to slow data exchange and possible collapse of the connection.

The Meta Quest 3 acts as the MQTT publisher. It uses an MQTT client library within the Unity environment, which is developed in the Unity C# project script, to establish a connection to a cloud-based MQTT broker. The broker host, port and authentication credentials are configurable to match the specific MQTT service and therefore applicable for multiple users on the shopfloor but also different broker and encryption types. After connecting, the Unity application periodically captures frames at the desired rate conceptual with ten fps, corresponding to a 100ms interval, within the game loop, encodes them to JPEG and transforms the resulting binary data into a Base64-encoded string. This

Base64 string is then embedded into a JavaScript Object Notation (JSON) payload that also contains metadata such as a Unix timestamp in milliseconds and a unique frame identifier. A typical payload structure thus consists of three fields, timestamp, frame id and image, where image holds the Base64-encoded JPEG data. The JSON string is finally published as the payload of an MQTT message to a dedicated topic.

On the processing side, a Windows-based client acts as the MQTT subscriber and object detection engine. This client is typically implemented in Python, leveraging the paho-mqtt library (*Paho-Mqtt*, 2026) for MQTT communication, OpenCV (*Opencv-Python*, 2026) for image decoding and visualization, numpy (*Numpy*, 2026) for numerical array handling and the Ultralytics YOLO package for loading and executing the YOLO11n model as the used object recognition model. The subscriber connects to the same cloud MQTT broker with corresponding credentials and subscribes to the frame topic used by the Meta Quest. Upon receiving a message, the client decodes the payload from UTF-8, parses the JSON and extracts the Base64-encoded image and metadata. The Base64 string is decoded back to raw JPEG bytes, which are then converted into an OpenCV image using standard decoding functions. This reconstructed frame is fed directly into the YOLO11n model, which performs object detection and returns bounding boxes, class labels and confidence scores. The results can be visualized by rendering bounding boxes and labels onto the frame, which is displayed in a window of the windows machine to provide live feedback of the detection process. Additionally, the subscriber logs relevant information, such as the timestamp, frame identifier and number of detected objects per frame, allowing basic performance and correctness checks.

A central design decision in this architecture is the use of Base64-encoded payloads within JSON, instead of transmitting raw binary data over MQTT. From a technical perspective, MQTT supports arbitrary binary payloads and sending the JPEG byte array directly would be more efficient in terms of bandwidth and CPU usage, as Base64 encoding introduces approximately 33% overhead in payload size (*Web Development*, 2024) and additional encoding/decoding computation on both ends. However, Base64 within JSON significantly simplifies interoperability, debugging and extensibility. JSON is human-readable and can easily carry additional metadata alongside the image. Timestamps, frame identifiers, camera parameters are considered as metadata in a structured way, which is highly convenient in early prototyping stages and for scientific documentation. The trade-off is consciously accepted because the editable configuration results in a manageable data rate on the order of a few hundred kilobytes per second, which is well within the capacity of typical cloud MQTT brokers and network links in a proof-of-concept scenario.

From a scalability and future-proofing perspective, the architecture is designed to be incrementally optimizable. Once a first successful implementation is achieved and validated, several improvement steps are possible. Increasing the image resolution to enhance detection accuracy, adjusting JPEG quality to fine-tune the trade-off between visual fidelity and bandwidth or incrementally increasing the frame rate beyond 5 fps are possible. As long as the broker, network and processing pipeline can handle the load a fluent work is scured. If bandwidth or CPU usage becomes critical, the system can be refactored to transmit raw binary payloads instead of Base64, thereby reducing both data size and encoding overhead. In that case, metadata could either be transported via a separate JSON-based topic or encoded in a custom binary header in the same payload. Additionally, QoS levels could be revisited if stronger delivery guarantees are required for specific use cases, at the cost of additional overhead.

5.5 Device Dependent Final Concept Architecture Overview

Figure 19 shows the AR-guided assembly system built around three interconnected layers, that have been described in the prior chapter. While Figure 19 shows a shorter version, the holistic version is available in Appendix F. On the left is the Meta Quest 3 headset as the hardware layer, which hosts four functional modules. A Calibration Module that supports hand placement, controller input and ArUco marker detection and persists its state across restarts; a Camera Module that continuously captures passthrough footage and streams it to the backend; a Guided Workflow Display that renders wireframe box highlights, 3D models, step-by-step instructions and detection overlays across assembly steps via grabbable and resizable panels; and a Telemetry Module that records step completions, timestamps, events and errors.

All communication between the headset and the backend runs through a MQTT broker hosted on HiveMQ Cloud using TLS 1.2. Four distinct data channels pass through this layer. Camera images flow from the headset to the AI backend, detection commands + calibration data flow back from the AI to the headset and production telemetry is forwarded outward toward analytics systems.

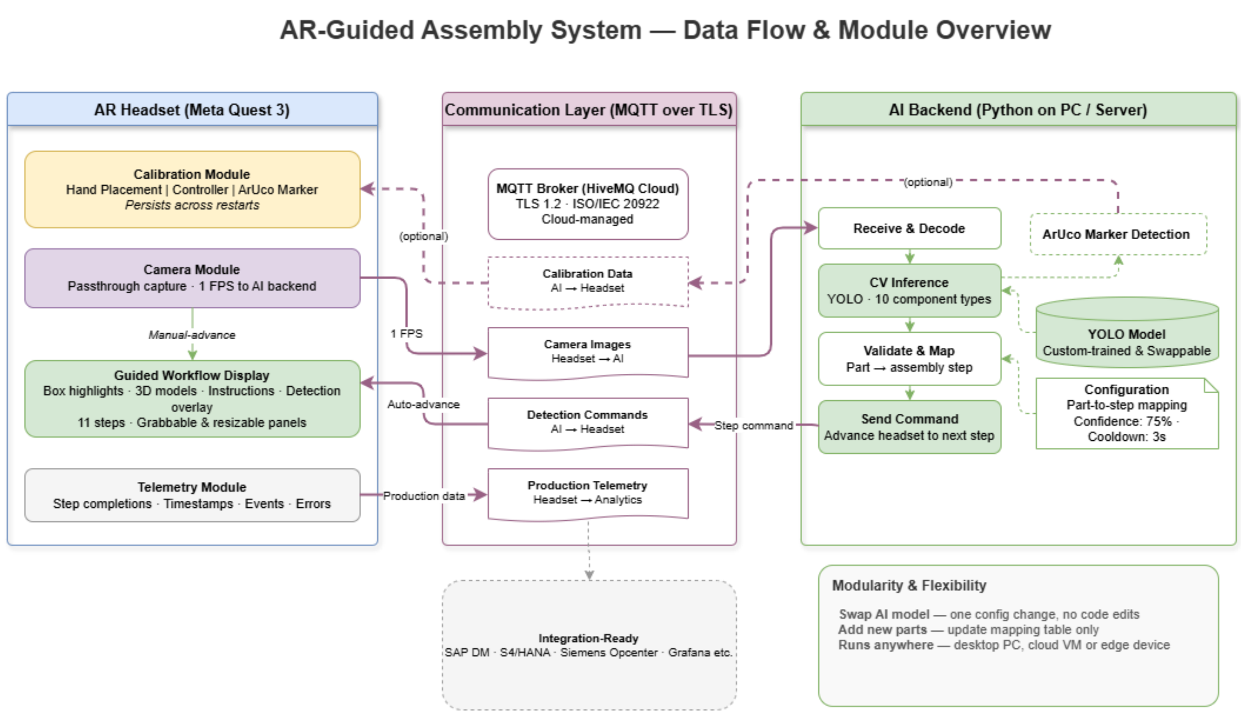


Figure 19 short version overview of the dataflow of the concept (self-made)

The AI backend, running Python on a PC or server, processes incoming images through a four-stage pipeline. Receiving & decode, CV inference via a custom-trained and swappable object detection model, matching detected parts to assembly steps using a configurable part-to-step mapping and cooldown time and finally sending a step-advance command back to the headset. Additionally, ArUco marker detection runs as a parallel sub-process within the inference stage and can optionally feed calibration corrections back to the headset via a dedicated channel.

The diagram closes with two callout areas highlighting design philosophy. First, communication is architected to feed into external systems such as SAP DM, S/4HANA, Siemens Opcenter or Grafana, which are example systems who support a MQTT machine interface. Second, the overall system is designed for modularity, swapping the AI model requires only a single config change, adding new parts only requires updating the mapping table and the backend can run on a desktop PC, a cloud VM or an edge device.

Since this chapter summarizes the concept architecture in a visual way, the following chapter presents the new capabilities from an operational point of view. Demonstrating what core functionalities have been established with the concept, to provide a proof of the concept.

6 Value Proposition of the Artifact

The following sections present the developed artifact´s core capabilities from an operational perspective, structured around one auxiliary setting, the spatial calibration of the scene and multiple following functional areas. The descriptions emphasize practical benefits and integration potential relevant to manufacturing stakeholders, while detailed technical implementation is documented in chapter 5 and Appendix F. The drone pre-assembly workflow used as the demonstration scenario throughout this chapter is introduced in Appendix C.

Since the scope of this work focused on system design and implementation, the operational benefits described in the following sections are presented as design properties of the architecture rather than empirically validated findings. Formal evaluation of these properties, including comparative studies of error rates, onboarding time and cycle time improvements, are identified as a priority for future work in chapter 9.1.

6.1 Pick by Vision Guided Assembly

A central aspect of the research question, how AI-driven AR assistance can enhance manual pre-assembly, concerns the manner in which work instructions are delivered to the operator. The system developed in this work implements a pick-by-vision approach in which context-aware, spatially anchored guidance is presented directly within the operator's field of view via the Meta Quest 3 headset. This replaces conventional instruction media such as paper manuals, printed checklists or stationary monitors and allows the operator to keep both hands free for assembly tasks, at all times.

6.1.1 Spatial Calibration

For AR-based work instructions to be operationally useful, the virtual content must be accurately aligned with the physical workstation. The presented system provides three calibration methods of varying complexity to accommodate different deployment environments.

The primary method uses the headset's built-in hand tracking. The operator pinches thumb and index finger to grab all virtual content as a single group and physically moves it into alignment with the real storage locations. Upon release, the position is automatically saved using Meta's Spatial Anchor technology, meaning the calibration persists across application restarts and does not need to be repeated at the beginning of each shift. In practice, this process can be completed in under 10 seconds and requires no additional equipment or technical expertise.

For environments where hand tracking is unreliable, such as workstations with poor lighting or situations requiring thick protective gloves, a controller-based method provides precise adjustment through thumb stick input at centimeter and degree resolution. This option is automatically activated if the primary method is used with controllers in the hands. If the controllers are not visible for the headset, this option is not applicable.

A third, fully automated method uses a printed ArUco fiducial marker placed on the workstation. The marker is detected by the headset camera and processed by the system's Python backend, which computes the spatial relationship between the marker and the headset and transmits the resulting calibration to the AR application. This method is particularly relevant for standardized, multi-shift deployments where consistent calibration accuracy must be maintained independently of the individual operator. This method can be expanded for multiple new use cases and doesn´t have to be used only for spatial calibration.

All three methods produce the same result, a single spatial reference point from which all AR content, like instruction panels, component highlights, three-dimensional models and detection overlays, is positioned relative to the physical workstation.

6.1.2 Spatial Bin Location Guidance

To provide the operator with unambiguous spatial direction at each assembly step, the system includes virtual representations of all storage bins present on the physical workstation. These virtual bins are modelled within the Unity AR development environment as spatial twins of their real counterparts, positioned through the calibration methods described in section 5.2. Each virtual bin is linked to one assembly step and a visual trigger is activated automatically whenever the instruction panel advances to the corresponding step, no manual activation of the bin highlighting is required.

The trigger takes the form of a pulsing green wireframe overlay that appears at the position of the relevant storage bin. The wireframe representation was chosen deliberately. Unlike a solid overlay, it allows the operator to see through the highlight and into the bin, maintaining visibility of the actual components inside. At any given step, only the bin associated with the current component is highlighted, directing the operator's hand to the correct location without requiring them to interpret written instructions or memorize bin assignments. When the step advances, whether through automated object detection or manual input, the wireframe transitions seamlessly to the next relevant bin.

To reinforce the spatial relationship between the instruction content and the physical storage location, the system renders a directional line from the instruction panel to the currently highlighted bin. This visual connection serves as an additional guidance cue, particularly in situations where the highlighted bin is outside the operator's immediate field of view or where multiple bins are positioned in close proximity.

The line originates from the instruction panel and terminates at the wireframe-highlighted storage bin, adopting the same green color as the bin highlight to maintain visual consistency. When a step transition occurs, the line updates simultaneously with the wireframe and instruction content, redirecting to the newly active bin. This ensures that the operator always has a clear, unambiguous visual path from the information being displayed to the physical location where the next action is required.

It should be noted that the system does not dynamically detect the physical positions of the storage bins at runtime. The bin locations are predefined within the Unity scene. The accuracy of both the wireframe highlights and the directional line therefore depend entirely on the quality of the initial calibration. If the physical workstation layout changes, for instance, if bins are rearranged or replaced, the virtual bin positions must be updated in the scene configuration accordingly.

6.1.3 Augmented Work Instruction Panel

The central interface element of the system is a floating instruction panel that presents all step-relevant information within the operator's field of view. The panel supports full spatial interaction e.g. it can be grabbed and repositioned freely using a pinch gesture, resized to any desired ratio by adjusting its corners and rotated in all directions. Once placed, the panel remains at its spatial position due to six-degree-of-freedom anchoring, it stays wherever the operator finds it most comfortable or least obstructive and does not drift or follow the operator's head movement. The panel is structured into four functional layers, each serving a distinct informational purpose, presented in figure 20 and briefly explained in table 4. The following paragraphs explain the layers more in detail.

The first layer displays a three-dimensional replica of the component to be picked in the current step. The replica rotates continuously at a tilted angle to ensure visibility from multiple perspectives. Giving the operator an unambiguous geometric reference of what to look for. The rotation speed, tilt angle, model size, color and position are configurable. In the current implementation, the component replica is anchored to the top of the panel and adapts proportionally when the panel is resized.

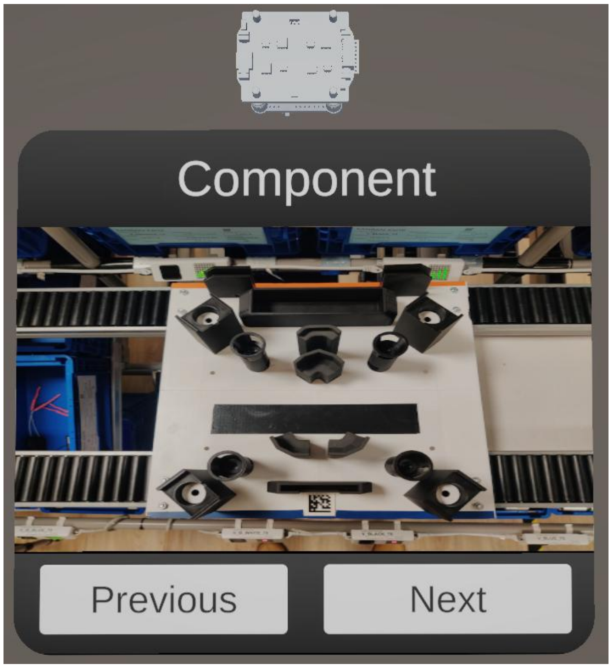


Figure 20 Work Instruction Panel Screenshot from Unity (self-made)

Layer 1	drone component 3D replica visualization
Layer 2	component name
Layer 3	workpiece image on the conveyer belt including component placement locations
Layer 4	manual step buttons

Table 5 Work Instruction Panel Layer Overview

The second layer displays the component name as a text label. At the initial step, step zero, the label reads "New Drone" and the first layer shows the complete assembled drone as a reference. Once the operator manually initiates the workflow by pressing the next button, the text updates to name the specific component required at each subsequent step, changing automatically with each step transition.

The third layer presents a reference image of the workpiece carrier, showing the operator where the current component must be placed. This image updates at each step to reflect the evolving assembly state. For example, the first assembly step shows the white base plate positioned vertically at the rear of the carrier, while the second step shows the yellow cover placed horizontally in the center. A limitation of the current implementation is that image resolution decreases when rendered within the AR application. A potential improvement, identified for future iterations, would be to replace the static reference images with a three-dimensional digital twin of the workpiece carrier, which would both improve visual clarity and create the opportunity to combine the carrier model with the individual component replicas already present in the first layer.

The fourth layer contains two navigation buttons, previous and next. These allow the operator to manually control step transitions at any time. These buttons serve as a complement to the automated progression driven by the object detection system described in section 6.2. For assembly steps one through ten, which correspond to individual components, the step transition is automated upon successful detection. Step zero, which presents the complete drone overview, remains manual, as it serves as the workflow entry point rather than a component pick action.

6.2 Automated Object Detection

The next core functionality is the automated component check, which operates through an AI backend running on a separate device outside the headset. This outsourced processing architecture, detailed in section 5.3, allows the system to perform computationally intensive object detection without burdening the headset's processor. The integration of automated detection elevates the AR headset from a passive instruction display to an active verification tool that independently confirms each pick action through computer vision.

When the operator presents a component to the headset's camera, the AI backend processes the captured frame and returns a detection result. The operator receives two simultaneous visual indicators confirming a correct pick. First, the background of the component name layer on the instruction panel turns green, providing immediate feedback within the instruction interface itself. Second, a bounding box overlay appears at the position of the detected component in the operator's real-world view, visually confirming what the system has recognized and where, figure 21 examples a correct detection.

Upon a confirmed match, the instruction panel advances automatically to the next step, the component name updates, the reference image changes and the spatial bin highlight transitions to the next storage location. No manual interaction is required for this progression.

Figure 21 operator view object detection overlay – detecting the correct component of the current step (self-made)

Figure 22 operator view object detection overlay – detecting a component of not the current step (self-made)

Critically, the detection is bound to the current assembly step. Once a step has been completed and the workflow has advanced, the same component will no longer be accepted as a valid detection for the new step. This behavior is intentional and serves as a safeguard against workflow errors. If the operator continues to hold or present the previously detected component after the step has already advanced, the system will indicate that the component does not match the current step, even though the backend correctly identifies the object. The detection result is displayed with an orange bounding box rather than a green one, signaling the operator that while the component has been recognized, it is not the one required at the current step. Figure 22 gives an example of that case.

This step-bound validation logic ensures that the workflow can only progress through the correct sequence and that no step can be inadvertently skipped through repeated detection of a previously validated component.

6.3 System Interoperability

The previous sections describe the system's capabilities from the operator's perspective, while the present section addresses the underlying architectural decisions that determine whether the system can be integrated into existing manufacturing infrastructure. A consideration that is critical for the research question, since an AR assistance system that operates in isolation from the factory's data ecosystem offers limited value beyond the individual workstation.

6.3.1 MQTT-Based Communication

All communication between the AR headset and the AI backend uses MQTT, an ISO-standardized messaging protocol widely adopted in industrial IoT and Industry 4.0 environments for machine-to-machine communication, sensor data collection and supervisory control systems. Section 3.2

This protocol choice carries strategic significance for manufacturing stakeholders. In this production environment, MQTT infrastructure is already in place, meaning the AR assembly system can connect to the existing message broker without requiring dedicated communication infrastructure. The system becomes a participant in the factory's existing data ecosystem rather than an isolated addition.

The protocol publish-subscribe architecture ensures that the headset and the AI backend are fully decoupled, neither component has direct knowledge of the other's implementation, network location or availability. This decoupling has practical implications for long-term maintainability. Either component can be updated, replaced or scaled independently. Deployment topology is equally flexible. The current implementation uses a cloud-hosted broker, allowing the headset and backend to operate on separate networks. Alternatively, an on-premises broker can be deployed for environments that require local operation or the entire backend can run on an edge computing device directly on the factory floor to minimize latency. All communication is encrypted using TLS and per-device credentials support access control and auditability.

Beyond the core detection and instruction pipeline, the system publishes structured telemetry events covering step completions, operator interactions, instruction lifecycle data and the opportunity for error conditions. These events use standardized JSON payloads with ISO 8601 (*ISO - ISO 8601, Date and Time Format*, 2017)timestamps, making them directly consumable by data platforms commonly used in manufacturing operations. Applicable, also for time-series databases, for performance monitoring, visualization tools for real-time dashboards or enterprise resource planning and manufacturing execution systems for production tracking and shift planning.

This telemetry capability is designed to transform the AR system from a standalone operator tool into a data source for continuous improvement. Cycle time analysis, bottleneck identification, error frequency monitoring and compliance documentation can all be derived from the telemetry stream without additional instrumentation of the assembly process.

6.3.2 Cross-Platform AI Backend

The implemented AI backend is using widely adopted, platform-independent technologies, ensuring that it is not constrained to a specific hardware configuration or deployment environment. The same backend can run on a standard workstation with a dedicated GPU for maximum inference speed, on a CPU-only machine where GPU hardware is unavailable, on a cloud virtual machine for centralized management or on an industrial edge device for latency-sensitive or network-restricted deployments. The detection model itself is loaded as a self-contained file, meaning that adapting the system to a different product line requires only replacing the model and updating the component-to-step mapping. The detection pipeline, communication layer and validation logic remain unchanged. This interchangeability is relevant for manufacturing environments where product variants change frequently or where multiple product lines must be supported in parallel.

The system's topic-based communication structure further supports multi-station scalability. Multiple headsets can publish to a shared AI backend instance or dedicated instances can be deployed per workstation, both configurations are supported without architectural changes, allowing the deployment to scale incrementally as operational needs evolve. It is advisable to create a topic structure system that clearly defines which headset and which station works with which product.

The combination of standardized communication, structured telemetry and modular backend design creates a foundation for integration pathways that extend beyond the current implementation. Enterprise system integration through telemetry subscription, multi-station deployment through parameterized communication channels, migration between edge and cloud processing, analytics dashboard connectivity and coordination with possible robotic systems through shared communication infrastructure are all enabled by the architectural decisions described above, without requiring re-engineering of the core system.

For manufacturing decision-makers, this architectural openness reduces adoption risk. The system does not introduce proprietary protocols or closed data formats that would create vendor dependency. Instead, it builds on the same industrial standards that the broader factory infrastructure already uses, ensuring that the investment in AR-assisted assembly remains compatible with the organization's evolving technology landscape.

7 Design Verification

The Design Science Research methodology requires a formal assessment of the developed artifact against the objectives defined at the outset of the work. This section provides that assessment, referencing the detailed evidence documented in the preceding chapters rather than restating it.

The research question asks how AI in combination with augmented reality smart glasses can enhance manual pre-assembly applications in a smart factory environment. The artifact answers this question through a functioning proof of concept that integrates AI-based object detection with AR-guided pick-by-vision assembly on a commercially available Meta Quest 3 headset, connected to an external AI backend via an MQTT communication pipeline built on open industrial standards. The system simultaneously guides the operator and verifies each pick action, demonstrating that the combination is technically feasible on cost-effective, commercially available hardware. The answer is architectural and technical. Empirical validation of productivity gains is explicitly deferred to the follow-up work proposed in chapter 9.

The three-step methodology was fulfilled as defined. The literature review (chapter 3) and market analysis (Appendix A) informed the selection of current state-of-the-art technologies, using YOLO for object detection, Meta's XR SDK in Unity and Spatial Anchor API for AR development and MQTT with TLS encryption for industrial communication. The concept creation phase in chapter 5 produced a modular three-layer architecture with clearly defined responsibilities and a closed-loop data flow from camera capture through detection to step advancement. The proof of concept, declared optional due to time and cost constraints, was nevertheless delivered as a functioning end-to-end implementation in chapter 6.

The first design goal, replacing pick-by-light with pick-by-vision guidance, is handed over to the TTZ stakeholders to further testing and development cycles. The system delivers spatially anchored bin highlighting, three-dimensional component previews and a multi-modal instruction panel directly within the operator's field of view, documented in section 6.1. The second design goal, implementing an AI-based controlling mechanism that prevents the assembly of incorrect components, is fulfilled. The closed-loop detection pipeline documented in section 6.2 functions as an active quality gate that blocks workflow progression until the correct component has been positively identified. Whether these architectural properties produce measurable error rate reductions under operational conditions remains a question for subsequent evaluation, consistent with the separation between feasibility demonstration and operational validation established in the target setting.

The research gap, the absence of a combined pick-by-vision and AI-driven detection application on a single commercial device, is addressed. The artifact integrates both capabilities on the Quest 3, with the detection pipeline executed on an external backend for the architectural reasons documented in section 5.3. This distributed approach is transferable to comparable hardware platforms and the path toward a self-contained solution is defined in the future research proposals of chapter 9.

8 Artifact Limitations

8.1 Network Latency Limitations

A latency tracker class included in the MQTT subscription script of the python backend as a diagnostic component, designed to collect, aggregate and report timing data across the real-time image processing pipeline, has been created. The pipeline spans two physically separate devices, as separate viewed processes via separate timestamps. First, a Meta Quest headset for image capture + encoding and second, a Python server including the MQTT reception, image decoding and YOLO object detection inference. The tracker addresses one of the fundamental challenges in distributed systems profiling, accurately measuring latency when timestamps originate from different, potentially unsynchronized clocks.

The technical validation of the latency tracking included has to be challenged by appropriate methods and appropriate measurements from experts. To the current state of the artifact there is no heavy latency recognizable if sufficient hardware is used for the object detection inference. An overview of the calculation is presented in table 6 and 7, while figure 23 shows a real evaluation of the current measures included.

The measurement of end-to-end latency is captured by the following time stamp capture:

- Total latency: T7 - T1
- Network latency: T5 - T3
- Processing time: T7 - T6
- Encoding time: T2 - T1

ID	Variable Name	Device	Description
T1	t_capture	Quest	Moment camera image is captured
T2	t_encoded	Quest	JPEG + Base64 + JSON encoding complete
T3	≈ T2	Quest	MQTT publish (negligible delta from T2)
T4		Quest / Internet	Network Transit time – not measured due to different system clocks
T5	t_received	Python	MQTT message received by subscriber
T6	t_yolo_start	Python	YOLO inference begins
T7	t_yolo_done	Python	YOLO inference complete

Table 6 Latency calculation via timestamps on all devices at all processes

Intervals where both timestamps originate from the same device clock are inherently accurate and require no correction. The tracker computes the following same-machine intervals:

Interval	Span	Description
T1 → T2	Quest	Image encoding time (JPEG compression, Base64, JSON serialization)
T5 → T6	Python	Image decoding time (Base64 decode, JPEG decompress)
T6 → T7	Python	YOLO inference duration
T5 → T7	Python	Total Python-side processing (decode + inference)

Table 7 Latency calculation on different devices separated with processes on the same device

This calculation is done by the subscriber python script on the windows machine every time the script disconnects successfully from the MQTT broker, by stopping the script itself or closing the object detection user interface. One example result show the limitation, that an accurate measurement has not been possible, due to wrong timestamp synchronization of the different devices, which lead to negative latencies in the evaluation. Figure 23 shows an example report, with average, minimum, maximum and P95 as percentile numbers. P95 indicates that 95% of all measured values were below this value, meaning that the latency was worse in only 5% of the frames.

```
======================================================================
  LATENCY REPORT  (1678 frames)
======================================================================

  ACCURATE METRICS (same-machine timestamps)
  Stage                                Avg       Min       Max       P95
  ----------------------------------------------------------------
  T1->T2  Encoding (Quest)          13.8ms     7.9ms    33.6ms    19.1ms
  T5      Image Decode               2.0ms     1.0ms     5.3ms     3.8ms
  T6->T7  YOLO Inference            54.3ms    51.9ms   596.9ms    56.0ms
  T5->T7  Python Processing         56.3ms    53.4ms   602.5ms    58.9ms

  CROSS-MACHINE METRICS (clocks synchronized)
  ----------------------------------------------------------------
  T2->T5  Network + Broker         -54.3ms   -88.8ms   501.7ms    -1.7ms
  T1->T7  Total Pipeline            15.8ms   -19.8ms   650.2ms    69.5ms
```

Figure 23 latency report one way calculated by AI backend(self-made)

8.2 Operator Controlling Object Detection

The object detection model training is based on a few hundred pictures. The current configuration leads to the fact that sometimes incorrect objects are considered to be correct, which leads to an acceptance of the work step by the system while the operator did not actively do the work step. This can also happen when the bin is rotated and the opening is shown towards the operator while the model detects a part that has not been picked up by the operator. The Operator doesn´t intentionally look at the component but the object detection model detects everything that the headset camera captures, no matter if it is intended at that moment or not.

8.3 Battery Life-Time Limitation

Due to the heavy computing of the unity application the headset´s battery suffers a lot. A capacity percentage from 95% to 20% has been reached multiple times within 30 until 35 minutes while continuously using pre-assembly assistance. A value that doesn't make it feasible to use in serial production on the shop floor, especially not because there is no hot-swap battery. The battery of the Meta Quest 3 is integrated into the headset and can´t be removed. Therefore, the only solution is either to swap the whole device or the usage of a USB-C cable which is plugged into the headset. Latter leads to constraints in the area of movement.

8.4 Calibration and Spatial Accuracy

While the system provides three calibration methods as described in section 5.2, the spatial accuracy of each method has not been quantitatively evaluated against ground truth. Several questions remain unanswered, like the positional error of the hand placement method in millimeters, the rate at which spatial anchor calibration degrades over time, the comparative accuracy of the ArUco method relative to the manual methods and the minimum calibration accuracy at which operators perceive the AR overlays as correctly aligned with the physical workstation.

Future research should develop a quantitative calibration evaluation protocol using precisely measured reference points to determine the absolute positional and rotational error of each calibration method. Longitudinal studies measuring spatial anchor drift over days and weeks, particularly in environments where the Quest 3's environmental mesh may change due to rearranged equipment or varying lighting, would inform decisions about recalibration intervals. Additionally, investigating the perceptual alignment threshold, the minimum spatial accuracy at which operators perceive AR content as correctly registered, would establish a concrete accuracy requirement against which the calibration methods can be evaluated.

The ArUco-based calibration currently uses an estimated camera intrinsic matrix, with focal length approximated from image dimensions rather than derived from a calibrated camera model. This introduces systematic pose estimation errors, particularly at oblique viewing angles and larger distances between the camera and the marker. The coordinate system conversion from the backends' right-handed convention to Unity's left-handed convention further involves assumptions about the marker's physical orientation relative to the workstation that may not hold in all deployment configurations.

Future work should pursue obtaining or computing the precise intrinsic parameters of the Quest 3's passthrough cameras, including lens distortion coefficients, to improve pose estimation accuracy. A longer-term direction is marker-free spatial registration using the Quest 3's scene understanding capabilities, plane detection and room mesh reconstruction, to recognize the workstation's geometry directly from the environmental mesh, eliminating the need for physical markers entirely.

8.5 Human Factors and Ergonomics

A significant methodological limitation of the current work is the absence of controlled user studies. The system has been developed and evaluated from a technical perspective, but its actual impact on operator performance, assembly time, error rates, cognitive load and user satisfaction, has not been empirically measured. Without user studies, the operational benefits described in section 3.5 remain design-level properties rather than validated findings.

A controlled experimental design comparing assembly performance across instruction modalities, paper-based, tablet-based, AR-guided without AI detection and the full AI-assisted AR system, would isolate the contribution of each system component. Standardized instruments such as the NASA Task Load Index for cognitive load and the System Usability Scale for usability assessment should be employed to enable comparison with results reported in the existing literature. Longitudinal adoption studies investigating how operator behavior changes over extended use periods would address additional questions, including whether operators develop over trust in the AI system, whether they cease to visually verify their own picks and whether initial performance measurements are inflated by novelty effects.

The Meta Quest 3 still imposes ergonomic constraints for extended industrial use. Prolonged wear over full shifts may cause neck fatigue, pressure-point discomfort or headaches, though the magnitude of these effects has not been evaluated for the present system. The headset's passthrough field of view, while wide, remains narrower than natural human vision, which may affect spatial awareness and safety in dynamic manufacturing environments. In multi-shift operations, shared headset use raises hygiene concerns for the face interface and head straps. Continuous AR overlay rendering at close focal distances may also contribute to visual fatigue over extended periods.

Future research should include longitudinal ergonomic assessments of full-shift headset use, measuring neck muscle fatigue, subjective comfort and visual accommodation patterns over four-to-eight-hour periods. Investigating minimal-UI design approaches, for example, displaying only the wireframe highlight and omitting the instruction panel once an operator has gained familiarity with the assembly sequence, could reduce visual complexity and fatigue without degrading assembly accuracy. Evaluating whether lighter AR form factors such as smart glasses with limited display capabilities could deliver sufficient guidance for experienced operators, while reserving the full headset experience for training and complex assemblies, represents a further direction.

8.6 Scalability and Generalizability

8.6.1 Long Development Cycles and Short Functional Guarantee

The creation of AR projects is very time consuming. The main issue the developing process is facing is that the game development engine Unity has to build the whole instance and move it to the headset, where it has to build up and run. Building up the instance in unity regularly takes several minutes. For each iteration a new scene and new application for the headset must be created and loaded onto the headset. Running the project live on the headset while editing it is only possible with the meta horizon link application, which has very high computational requirements. These requirements could not be met with the university´s or private´s hardware equipment. That led to the fact that only minor developments took up to multiple days to implement them as good as required to use.

The concept has been created as modular and as reconstructible as possible. Still the main building blocks of the AR application are based on the Meta Software Development Kits because a Meta Quest 3 has been used. The scenes can´t be applied on different smart glasses or headsets because they build on their own producers Software Development Kits. Basic Unity buttons or visuals and functions remain the same, but the main configurations and most of the self-created scripts build on permissions and functionalities of the Meta Software Development Kit. In addition, there has been many development Kit changes, where packages have been deprecated or will be deprecated in the future, furthermore camera access of the meta quest is still in experimental stage. For both areas it is not clear how dynamic the changes of the producer will be in future, so that a long-term use cannot be guaranteed.

8.6.2 Product-Specific Deployment Overhead

The current system is configured for a specific drone assembly with ten known component types. Deploying the system for a different product requires collecting training images of the new components, training or fine-tuning the detection model, updating the component-to-step mapping, creating new three-dimensional models and instruction content and repositioning the spatial highlights and recalibrating the workstation. This deployment overhead may limit adoption in high-mix, low-volume manufacturing environments where product variants change frequently.

Future research should systematically evaluate transfer learning efficiency, specifically, how much training data is required to achieve reliable detection for new component types when fine-tuning from the existing model. Establishing minimum viable dataset sizes per component class would enable realistic deployment planning. Investigating whether instruction content and three-dimensional models can be automatically generated from CAD files or product lifecycle management systems would reduce the manual content creation effort. The zero-shot detection approaches discussed in section 9.4 are also directly relevant here, as they could enable rapid deployment for new products without the data collection and training overhead that currently represents the primary bottleneck.

8.6.3 Single Station Architecture

The current system supports one operator at one workstation. Scaling to a production line with multiple stations and operators introduces challenges including topic management for multiple headsets on shared MQTT infrastructure, load balancing for backend instances processing images from concurrent headsets and coordination of workflow state across stations that form a sequential assembly line.

Future research should address multi-station orchestration through a station-aware architecture in which the communication structure, workflow state and AI backend are parameterized per station, enabling centralized monitoring while maintaining decentralized operation. Extending the system to model inter-station dependencies, passing quality data downstream and integrating with manufacturing execution systems for end-to-end production tracking, would position the system for productive deployment in line-based manufacturing environments.

9 Follow-Up Research and Development Proposals

The system developed in this work establishes a functional foundation for AI-driven AR-assisted manual assembly. However, as the limitations discussed in chapter 8, the current prototype represents an early stage in a broader research and development exploration. The following proposals outline follow-up projects that are built directly on the architecture, infrastructure and findings of the present work. They are ordered by personal assessment of added value to the current system and the TTZ Leipheim. Beginning with projects that address immediate evaluation and architectural gaps as short term implementations before progressing toward more ambitious integration and scaling scenarios, that require long term planning.

9.1 Formative Usability Study and Operator-Driven Platform Development

The limitations chapter identifies the absence of user studies as a significant methodological gap in the current work. Due to the prototype's current maturity level, particularly regarding latency performance, AR visualization quality and headset weight, a comparative performance study measuring assembly time and error rate improvements would be premature. The system is not yet at a stage where operators would adopt it as a preferred working tool and measuring performance against established methods would produce results that reflect prototype maturity rather than the viability of the underlying concept.

A more productive and methodologically appropriate approach is a formative usability study designed not to validate performance gains but to identify and prioritize the technical and experiential barriers that must be resolved before productive deployment. Using structured interviews, think-aloud protocols during guided assembly sessions and standardized instruments such as the NASA Task Load Index (*TLX @ NASA Ames*, 2020; Zigart et al., 2023, p. 168) and the System Usability Scale (Klug, 2017), this study would produce a validated requirements roadmap for subsequent development iterations. Within the Design Science Research methodology that frames this thesis, such an evaluation constitutes an enormous feedback cycle that directly influences the next design iteration, with the view of several different users.

A central finding that can already be anticipated is the need for greater flexibility in how operators interact with the interface and being able to customize the AR interface. Accordingly, this project should extend beyond pure evaluation into the development of a platform-oriented configuration layer, an interface through which individual operators, production engineers or deployment teams can adjust visual design preferences, panel layouts, instruction formats, color schemes and interaction modes without requiring custom software development. The goal could be to transform the current system from a developer-configured prototype into a self-service platform where the user experience is as modular and individually adaptable as the underlying technical architecture already permits. The contribution of this project could be twofold. An empirically grounded prioritization of improvement areas and the introduction of a platform architecture that enables non-technical stakeholders to shape the AR experience to their specific operational and ergonomic needs.

9.2 Wearable Edge Inference Architecture

The current system's dependency on Wi-Fi-based image transmission, to an external backend, creates latency, network fragility and data protection constraints that were identified during the use of the artifact. Pure on-device inference on the Quest 3's mobile processor offers insufficient computational power for reliable real-time detection, while a stationary edge server reintroduces a cable tether to the workstation that constrains operator movement.

This project proposes a body-worn edge computing architecture in which a compact, high-performance inference device, such as an NVIDIA Jetson Orin Nano (Conrad.de, 2026) or a comparable mobile GPU platform, is carried directly on the operator's body, for instance mounted on a belt, integrated into a vest or attached to the headset's rear strap as a counterweight that simultaneously improves ergonomic weight distribution. The device would connect to the headset via cable, receiving camera frames and returning detection results with single-digit millisecond latency. All image data remains physically on the operator's person, eliminating both the network dependency and any data security concerns associated with wireless image transmission.

The research contribution lies in the design and evaluation of a complete body-worn AR inference pipeline, including hardware selection and form factor optimization for industrial wearability, power management for full-shift operation, thermal considerations for body-proximate GPU computing and the interface architecture between headset and edge device. The resulting system should be fully self-contained and mobile. The operator could move between workstations, work in areas without Wi-Fi coverage or operate in air-gapped environments without any loss of AI-assisted detection capability. This project could build directly on the MQTT-based communication architecture of the current system. With the same message protocol serving as the internal communication layer between headset and body-worn device via cable instead of the internet, preserving the modular design principles established in the current architecture.

9.3 Augmented Reality Smart Glasses Adaptation

The Meta Quest 3's weight and form factor were identified in section3.3.3 as ergonomic constraints for extended industrial use. Optical see-through smart glasses, which project virtual content directly onto transparent lenses rather than rendering a full passthrough video feed, offer a significantly lighter and less encumbering form factor that may be more suitable for full-shift industrial deployment.

This project proposes adapting the modular system architecture developed in the present work to an optical see-through smart glasses platform. The key research challenge lies not in porting the backend infrastructure, but in redesigning the AR visualization layer for the fundamentally different display characteristics of smart glasses. Optical see-through displays offer narrower fields of view, lower brightness, limited color rendering against bright backgrounds and no ability to occlude the real world. The wireframe highlights, three-dimensional model previews and instruction panels designed for the Quest 3's high-resolution passthrough display must be reconceived for these constraints.

The contribution of this project would be a comparative evaluation of the pick-by-vision concept across display modalities, establishing a playbook for future applications on different devices. Including which guidance elements translate effectively to smart glasses, which require redesign and whether the reduced visual fidelity is offset by the ergonomic advantages of a lighter device. The findings would inform a tiered deployment strategy. The proposed strategy might be that, smart glasses serve experienced operators who require only spatial highlighting and minimal instruction cues, while the full headset experience is reserved for training scenarios and complex assemblies where multi-modal guidance is essential.

9.4 Zero-Shot and Few-Shot Component Detection

Section 5.3.5 identifies the product-specific training requirement as a significant barrier to deployment scalability. Each new product line currently requires collecting training images, labelling objects, training or fine-tuning a YOLO model and validating detection accuracy. A process that may take days to weeks and requires machine learning expertise that might not be available in all manufacturing organizations.

This project proposes replacing the product-specific YOLO training pipeline with vision-language models capable of zero-shot or few-shot component detection. Architectures such as CLIP-based open-vocabulary detectors (OVD) (Xi et al., 2025, p. 25455) or vision language models (Cheng et al., 2024), that could enable deployment for a new product within hours using only reference photographs or CAD renders, with no model training required. The MQTT-based architecture of the current system is well suited for this transition, as only the inference component of the backend requires replacement, the communication layer, step validation logic and AR visualization pipeline remain unchanged.

The research challenge is complex. Vision-language models might lag behind task-specific detectors in fine-grained recognition accuracy, particularly for visually similar industrial components where subtle geometric or color differences determine correctness. The project would need to systematically evaluate detection reliability across component categories of varying visual distinctiveness, establish the conditions under which zero-shot detection provides acceptable accuracy and identify failure cases where minimal fine-tuning with a small number of reference images is required. The contribution would be an evidence-based assessment of whether and under what conditions training-free deployment is viable for industrial AR-assisted assembly, along with a practical deployment protocol that manufacturing organizations can follow without machine learning expertise.

9.5 Multi-Modal Quality Gate

The current system verifies only one aspect of the assembly process, whether the correct component has been picked or not. It does not verify whether the component was correctly placed, properly oriented or assembled with the required torque, force or alignment. This verification gap means that the system addresses pick errors but not placement errors.

This project proposes extending the verification scope by integrating additional sensor modalities into the existing MQTT infrastructure. Torque wrenches publishing tightening values, weight sensors confirming component placement on designated surfaces and barcode or RFID scanners verifying component serial numbers could all publish their data to dedicated MQTT topics. A complementary backend would then evaluate a multi-modal evidence set for each assembly step. The current visual confirmation of the correct component via object detection would be combined with sensor confirmation of correct assembly action, before advancing the workflow.

The contribution is the design and evaluation of a multi-modal quality gate architecture that closes the verification loop from pick to confirmed placement, with additional use cases then the pick by vision system. The MQTT publish-subscribe infrastructure established in the present work provides a natural integration point for additional sensor streams without requiring architectural changes. The research challenge lies in defining the fusion logic on how to combine evidence from heterogeneous sources with different reliability characteristics, latency profiles and potential failure modes into a unified step validation decision. The resulting system would offer a level of assembly verification that approaches automated quality inspection assistance, while the operator retains full manual control over the physical assembly process.

9.6 ERP and MES Integration

Section 5.1.3 describes the system's structured telemetry events, step completions, operator interactions, instruction lifecycle data and error conditions, as a data foundation for enterprise integration. Due to time constraints and uncertainties in the long-term use of certain systems, this integration has not been implemented or validated in the current work. The telemetry data is published to MQTT topics but is not consumed by any downstream system.

This project proposes connecting the existing telemetry streams to an enterprise resource planning or manufacturing execution system, such as SAP Digital Manufacturing or Siemens Opcenter, to demonstrate end-to-end data flow from the AR-guided assembly process to the enterprise data layer. The objective is to achieve real-time production KPI reporting, automatic quality documentation generated from step-level telemetry and compliance-ready traceability records, all derived directly from the operator's AR-guided assembly activities. Automated and reliable through the whole architecture without requiring manual data entries.

The research contribution is twofold. First, the project would validate whether the telemetry data structure and granularity designed in the current system is sufficient for meaningful enterprise integration or whether additional data points, formats or aggregation logic are required. Second, it would produce a reference architecture for AR-to-ERP data pipelines that other researchers and practitioners can adapt to their specific enterprise system landscape. The MQTT-to-enterprise integration pattern, subscribing to telemetry topics and transforming JSON events into system-specific formats, is generalizable beyond the specific AR system developed here and could serve as a template for integrating other shop floor data sources into manufacturing execution workflows. Since AR modules are device specific due to the provider SDK´s, the focus could lay on the environment and the interoperability beyond the headset or glasses to create a product that can be applied to almost any data integration layer system independent from the used device on the shop floor.

9.7 Multi-Station Production Line Deployment

The current system supports one operator at one workstation. Section 8.6 identifies the challenges involved in scaling to a production line with multiple stations, including MQTT topic management, backend load balancing and inter-station workflow coordination.

This project proposes designing and deploying a station-aware multi-headset architecture in which the communication structure, workflow state and AI backend are parameterized per station. Each workstation would operate its own detection pipeline, either through dedicated backend instances or through a shared backend with station-specific topic routing. While a centralized orchestration layer monitors progress across the line, production scheduling, KPI creation and more opportunities can be used for line level planning and performance analysis.

The key research questions concern the boundary between centralized and decentralized control. Which functions benefit from centralization, model management, telemetry aggregation, workflow sequencing and which must remain local to the station for latency and reliability reasons? How should the system handle situations where one station falls behind or produces a quality exception that affects downstream stations? The contribution would be a validated multi-station AR assembly architecture that demonstrates the scalability of the MQTT-based design principles established in the single-station prototype, along with empirical data on the operational characteristics, latency, throughput, failure resilience, of the scaled deployment.

9.8 Digital Twin Synchronization

The final proposal extends the system's data output beyond telemetry events toward a continuous spatial representation of the assembly process. By feeding real-time assembly progress and detection events into a digital twin platform, a live three-dimensional replica of the workstation could be maintained. Reflecting the current assembly state, the operator's progress through the workflow and the most recent detection results, would be mirrored in the digital representation. Enabling remote experts to observe the assembly process in real time for support, training supervision or process optimization purposes.

The research challenge lies in achieving sufficient synchronization fidelity, both temporal and spatial, for the digital twin to provide actionable remote insight. The latency between a physical assembly action and its reflection in the digital twin must be low enough for real-time support scenarios and the spatial accuracy of the replicated workstation must be sufficient for remote experts to understand the operator's context. The contribution would be a reference implementation demonstrating how the telemetry and detection data already generated by the AR assembly system can be repurposed as the input layer for a holistic digital twin. Going along with an evaluation of the synchronization fidelity achieved and its adequacy for the target use cases of remote support and process analysis.

10 Conclusion

The present work sets out to investigate how AI-driven augmented reality assistance can enhance manual pre-assembly in a smart factory context. The answer, grounded in a functioning proof of concept rather than theoretical argumentation, is that the combination of AR-based spatial guidance and AI-powered object detection can simultaneously address two persistent challenges in manual assembly, reliable instruction delivery and automated quality verification, both realized on commercially available hardware without proprietary infrastructure.

The developed system demonstrates that a pick-by-vision approach, in which spatially anchored work instructions are presented directly within the operator's field of view, replacing conventional instruction media while keeping both hands free for assembly work. Three calibration methods of varying complexity ensure that the spatial registration between virtual content and the physical workstation can be established across diverse deployment environments, from controlled laboratory settings to industrial shop floors. The integration of a custom-trained YOLO object detection model extends the system beyond passive guidance into active quality assurance. Transforming the AR headset from an instruction display into a verification tool that prevents incorrect picks by design rather than by operator diligence alone.

The architectural decisions underlying this concept execution, a cloud-mediated MQTT communication pipeline, a modular separation between AR frontend and AI backend, structured telemetry publishing on standardized topics, were driven not only by technical considerations but by the practical realities of manufacturing integration. By building on an ISO-standardized industrial messaging protocol, the system is designed to participate in existing factory data ecosystems rather than operate in isolation. The configurability of both the detection model and the instruction content ensures that the same architecture can serve different products and assembly sequences without re-engineering, a property that is essential for long-term viability in manufacturing environments where product variants and workforce composition change regularly.

At the same time, this work is transparent about what it does not yet demonstrate. The operational benefits described throughout the thesis, like reduced pick errors, faster onboarding and hands-free efficiency, are design properties of the architecture, not empirically validated findings. The absence of controlled user studies, baseline comparisons against established instruction methods and field deployment in productive manufacturing environments represents the most significant methodological limitation. The prototype's current maturity level, particularly regarding inference latency, headset ergonomics and AR visualization refinement, places it at a stage where the concept is proven but the experience is not yet ready for operator adoption. These limitations are not incidental observations but the foundation from which the eight follow-up research proposals presented in this thesis were derived, each one traceable to a specific gap identified in the critical evaluation.

The contribution of this work is therefore twofold. For researchers, it provides a documented, reproducible system architecture that integrates object detection with AR-guided assembly on commercial hardware, accompanied by a critical evaluation that ranks what has been achieved from what remains to be validated. For manufacturing stakeholders, it demonstrates that the technological foundation for AI-assisted AR assembly exists today on cost-effective, commercially available platforms and that the remaining barriers to productive deployment are matters of engineering maturation and empirical validation rather than fundamental feasibility. The question is no longer whether AI-driven AR can enhance manual pre-assembly, but under what conditions and at what level of refinement will it do so reliably and comfortably enough for the users to justify operational adoption? This thesis provides both the architectural foundation and the research agenda to pursue that answer

Bibliography

3DoF vs 6DoF: Key Differences Explained for VR, Gaming & AR. (2025). ARwall. https://arwall.co/blogs/arwall-blogs/3dof-vs-6dof

ABI. (2024). *ABI Research*. https://www.abiresearch.com/press/survey-finds-32-of-discrete-and-process-manufacturers-are-implementing-xr-technologies-into-their-operations

Ariansyah, D., Erkoyuncu, J. A., Eimontaite, I., Johnson, T., Oostveen, A.-M., Fletcher, S., & Sharples, S. (2022). A head mounted augmented reality design practice for maintenance assembly: Toward meeting perceptual and cognitive needs of AR users. *Applied Ergonomics*, *98*, 103597. https://doi.org/10.1016/j.apergo.2021.103597

Ashok, J., Kumar, N. A., Raj, D. W. P., Ashok, J., Bhushan, A. V., & Edem, S. (2023). Design and Implementation of Smart Manufacturing Systems Through AR for Data-Driven Digital Twin System. *SN Computer Science*, *4*(5), 580. https://doi.org/10.1007/s42979-023-01956-1

Belsito, C., Franciosi, C., & Di Pasquale, V. (2024). The Impact of Augmented Reality Devices on Operator Performance in Manufacturing Contexts. *Management and Production Engineering Review*, *15*(1). https://doi.org/10.24425/mper.2024.149987

Bender, M., Kirdan, E., Pahl, M.-O., & Carle, G. (2021). *Open-Source MQTT Evaluation*. 1–4. https://doi.org/10.1109/CCNC49032.2021.9369499

Breque, M., Cotta, J., De Nul, L., & Petridis, A. (2021). *Industry 5.0: Towards a sustainable, human centric and resilient European industry*. Publications Office of the European Union. https://data.europa.eu/doi/10.2777/308407

Brunetti, D., Gena, C., & Vernero, F. (2022). Smart Interactive Technologies in the Human-Centric Factory 5.0: A Survey. *Applied Sciences*, *12*(16), 7965. https://doi.org/10.3390/app12167965

Chan, W. P., Hanks, G., Sakr, M., Zhang, H., Zuo, T., Van der Loos, H. F. M., & Croft, E. (2022). Design and Evaluation of an Augmented Reality Head-mounted Display Interface for Human Robot Teams Collaborating in Physically Shared Manufacturing Tasks. *Acm Transactions on Human-Robot Interaction*, *11*(3), 31. https://doi.org/10.1145/3524082

Chen, A., Xu, Z., Geiger, A., Yu, J., & Su, H. (2022). *TensoRF: Tensorial Radiance Fields* (arXiv:2203.09517). arXiv. https://doi.org/10.48550/arXiv.2203.09517

Cheng, T., Song, L., Ge, Y., Liu, W., Wang, X., & Shan, Y. (2024). *YOLO-World: Real-Time Open-Vocabulary Object Detection* (arXiv:2401.17270). arXiv. https://doi.org/10.48550/arXiv.2401.17270

Chip.de. (2025, November 20). *Samsung Galaxy Smart Glasses schon 2026? Erste Details zur Brille sickern durch*. https://www.chip.de/news/handy/samsung-galaxy-smart-glasses-schon-2026-erste-details-zur-brille-sickern-durch_c35a83cd-3bcd-49ae-adf5-aa50bd2b081c.html

Choi, S. H., Kim, M., & Lee, J. Y. (2025). Smart and user-centric manufacturing information recommendation using multimodal learning to support human-robot collaboration in mixed reality environments. *Robotics and Computer-Integrated Manufacturing*, *91*, 102836. https://doi.org/10.1016/j.rcim.2024.102836

Conrad.de. (2026). *Nvidia Super Developer Kit Jetson Orin Nano 8 GB 6 x 1.5 GHz kaufen*. https://www.conrad.de/de/p/nvidia-super-developer-kit-jetson-orin-nano-8-gb-6-x-1-5-ghz-2998506.html

CUDA - NVIDIA. (2025). https://docs.nvidia.com/cuda/cuda-programming-guide/

Dalle Mura, M., & Dini, G. (2021). Augmented Reality in Assembly Systems: State of the Art and Future Perspectives. In S. Ratchev (Ed.), *Smart Technologies for Precision Assembly* (pp. 3–22). Springer International Publishing. https://doi.org/10.1007/978-3-030-72632-4_1

Danielsson, O., Ettehad, M., & Syberfeldt, A. (2024). *Augmented Reality Smart Glasses for Industry: How to Choose the Right Glasses*. https://doi.org/10.3233/ATDE240173

Danielsson, O., Holm, M., & SYBERFELDT, A. (2021). Evaluation Framework for Augmented Reality Smart Glasses as Assembly Operator Support: Case Study of Tool Implementation. *ResearchGate*. https://doi.org/10.1109/ACCESS.2021.3096855

Das, A., Rahman, A., Hossain, S. T., Ahmed, R., & Ara, M. (2025). Mixed Reality for Human–Robot Teaming to Enhance Work Health and Safety in Manufacturing Industries: A Systematic Literature Review. *Augmented Human Research*, *10*(1), 11. https://doi.org/10.1007/s41133-025-00085-z

Degrees of Freedom (DoF): 3-DoF vs 6-DoF for VR Headset Selection. (2025, April 10). VirtualSpeech. https://virtualspeech.com/blog/degrees-of-freedom-vr

developers.meta.com. (2026). *Track QR codes in MR Utility Kit*. https://developers.meta.com/horizon/documentation/unity/unity-mr-utility-kit-qrcode-detection/

DigitalOcean. (2025). *Top Object Detection Models for Your Projects in 2025 | DigitalOcean*. https://www.digitalocean.com/community/tutorials/best-object-detection-models-guide

Dodoo, J. E., Al-Samarraie, H., Alzahrani, A. I., & Tang, T. (2025). XR and Workers' safety in High-Risk Industries: A comprehensive review. *Safety Science*, *185*, 106804. https://doi.org/10.1016/j.ssci.2025.106804

Doerner, R., Broll, W., Grimm, P., & Jung, B. (Eds.). (2022). *Virtual and Augmented Reality (VR/AR): Foundations and Methods of Extended Realities (XR)*. Springer International Publishing. https://doi.org/10.1007/978-3-030-79062-2

Doolani, S., Wessels, C., Kanal, V., Sevastopoulos, C., Jaiswal, A., Nambiappan, H., & Makedon, F. (2020). A Review of Extended Reality (XR) Technologies for Manufacturing Training. *Technologies*, *8*(4), 77. https://doi.org/10.3390/technologies8040077

Emporio, M., Caputo, A., Pintani, D., Cheng, D. S., De Marchi, T., Forte, G., Fummi, F., & Giachetti, A. (2024). Integration of Extended Reality with a Cyber-Physical Factory Environment and its Digital Twins. *Proceedings of the Acm on Human Computer Interaction*, *8*(EICS), 250. https://doi.org/10.1145/3660246

Enang, E., Bashiri, M., & Jarvis, D. (2023). Exploring the transition from techno centric industry 4.0 towards value centric industry 5.0: A systematic literature review. *International Journal of Production Research*, *61*(22), 7866–7902. (172441299). https://doi.org/10.1080/00207543.2023.2221344

Esen, Y. E., Kaan Çetincan, B., Yayan, K., Güray Gürlek, H., & Yayan, U. (2025). XR-Driven Robotic System Training for Occupational Health, Safety, and Maintenance. *IEEE Access*, *13*, 61708–61727. https://doi.org/10.1109/ACCESS.2025.3556699

Fang, W., Zhang, T., Chen, L., Cao, Y., Hu, H., & Bi, J. (2025). HiMAR: A hierarchical context-aware mobile AR assembly for large-scale cable products. *International*

Journal of Production Research, 1–22. (188513545). https://doi.org/10.1080/00207543.2025.2568740

Firme, B., Rodriguesb, J., Restivob, M. T., Vieiraa, S., & Sousaa, J. M. C. (2024). *Augmented reality in intelligent automation: An application to support industrial scheduling—ABI/INFORM Global—ProQuest*. https://www.proquest.com/abiglobal/docview/3142548182/46A0576E56454F41PQ/3?accountid=15918&sourcetype=Scholarly%20Journals

GeeksforGeeks. (2025). *How to Avoid Overfitting in Machine Learning?* GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/how-to-avoid-overfitting-in-machine-learning/

Geng, R., Li, M., Hu, Z., Han, Z., & Zheng, R. (2022). Digital Twin in smart manufacturing: Remote control and virtual machining using VR and AR technologies. *Structural and Multidisciplinary Optimization*, *65*(11), 321. https://doi.org/10.1007/s00158-022-03426-3

Ghobakhloo, M., Iranmanesh, M., Tseng, M.-L., Grybauskas, A., Stefanini, A., & Amran, A. (2023). Behind the definition of Industry 5.0: A systematic review of technologies, principles, components, and values. *Journal of Industrial and Production Engineering*, *40*(6), 432–447. https://doi.org/10.1080/21681015.2023.2216701

Girshick, R. (2015). *Fast R-CNN* (arXiv:1504.08083). arXiv. https://doi.org/10.48550/arXiv.1504.08083

Hevner, A., R, A., March, S., T, S., Park, Park, J., Ram, & Sudha. (2004). Design Science in Information Systems Research. *Management Information Systems Quarterly*, *28*, 75.

Hijry, H., Meesam Raza Naqvi, S., Javed, K., Albalawi, O. H., Olawoyin, R., Varnier, C., & Zerhouni, N. (2024). Real Time Worker Stress Prediction in a Smart Factory Assembly Line. *IEEE Access*, *12*, 116238–116249. https://doi.org/10.1109/ACCESS.2024.3446875

Hořejší, P., Novikov, K., & Šimon, M. (2020). A Smart Factory in a Smart City: Virtual and Augmented Reality in a Smart Assembly Line. *IEEE Access*, *8*, 94330–94340. https://doi.org/10.1109/ACCESS.2020.2994650

Hou, Y., Xie, Q., Zhang, N., & Lv, J. (2025). Cognitive load classification of mixed reality human computer interaction tasks based on multimodal sensor signals. *Scientific Reports*, *15*(1), 13732. https://doi.org/10.1038/s41598-025-98891-3

Huang, S., Wang, B., Li, X., Zheng, P., Mourtzis, D., & Wang, L. (2022). Industry 5.0 and Society 5.0—Comparison, complementation and co-evolution. *Journal of Manufacturing Systems*, *64*, 424–428. https://doi.org/10.1016/j.jmsy.2022.07.010

Hutchinson, R. (2025, October 14). Samsung Smart Glasses. *Geeky Gadgets*. https://www.geeky-gadgets.com/samsung-smart-glasses/

ISO - ISO 8601—Date and time format. (2017, February 21). ISO. https://www.iso.org/iso-8601-date-and-time-format.html

ISO/IEC 20922:2016. (2016). ISO. https://www.iso.org/standard/69466.html

Ji, T., & Xu, X. (2025). Exploring the Integration of cloud manufacturing and cyber-physical systems in the era of industry 4.0 – An OPC UA approach. *Robotics and Computer-Integrated Manufacturing*, *93*, 102927. https://doi.org/10.1016/j.rcim.2024.102927

Jumahat, S., Sidhu, M. S., & Shah, S. M. (2023). A review on the positive implications of augmented reality pick-by-vision in warehouse management systems. *Acta Logistica*, *10*(1), 1–10. https://doi.org/10.22306/al.v10i1.337

Kerbl, B., Kopanas, G., Leimkühler, T., & Drettakis, G. (2023). *3D Gaussian Splatting for Real-Time Radiance Field Rendering* (arXiv:2308.04079). arXiv. https://doi.org/10.48550/arXiv.2308.04079

Kerbl, B., Meuleman, A., Kopanas, G., Wimmer, M., Lanvin, A., & Drettakis, G. (2024). *A Hierarchical 3D Gaussian Representation for Real-Time Rendering of Very Large Datasets* (arXiv:2406.12080). arXiv. https://doi.org/10.48550/arXiv.2406.12080

Kittithanon, S., Wicha, S., Tongpaeng, Y., & Kantawong, K. (2025). Developing a Mixed Reality (MR) Technology System for Training Intelligent Logistics Personnel, a Learning Course on Vehicle Condition Inspection. In P. Sureephong, C. Danjou, & A. Bouras (Eds.), *Product Lifecycle Management. Leveraging AI, Digital Twins, and Smart Technologies* (pp. 354–364). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-93323-3_27

Klug, B. (2017). An Overview of the System Usability Scale in Library Website and System Usability Testing. *Weave: Journal of Library User Experience*, *1*(6). https://doi.org/10.3998/weave.12535642.0001.602

Konstantinidis, F. K., Myrillas, N., Tsintotas, K. A., Mouroutsos, S. G., & Gasteratos, A. (2023). A technology maturity assessment framework for Industry 5.0 machine vision systems based on systematic literature review in automotive manufacturing. *International Journal of Production Research*, 1–37. (173008719). https://doi.org/10.1080/00207543.2023.2270588

Label Studio. (2026). https://labelstud.io/templates/

Lavric, T., Bricard, E., Preda, M., & Zaharia, T. (2022). A Low-Cost AR Training System for Manual Assembly Operations. *Computer Science and Information Systems*, *19*(2), 1047–1073. https://doi.org/10.2298/CSIS211123013L

Li, S., You, Y., Zheng, P., Wang, X. V., & Wang, L. (2024). Mutual-cognition for proactive human–robot collaboration: A mixed reality-enabled visual reasoning-based method. *IISE Transactions*, *56*(10), 1099–1111. https://doi.org/10.1080/24725854.2024.2313647

Liao, Z., & Cai, Y. (2024). AR-enhanced digital twin for human–robot interaction in manufacturing systems. *Energy, Ecology and Environment*, *9*(5), 530–548. https://doi.org/10.1007/s40974-024-00327-7

Lopes, J. C., & Lopes, R. P. (2024). Computer Vision in Augmented, Virtual, Mixed and Extended Reality environments—A bibliometric review. *Visual Informatics*, *8*(4), 13–22. https://doi.org/10.1016/j.visinf.2024.11.002

Lu, T., Yu, M., Xu, L., Xiangli, Y., Wang, L., Lin, D., & Dai, B. (2023). *Scaffold-GS: Structured 3D Gaussians for View-Adaptive Rendering* (arXiv:2312.00109). arXiv. https://doi.org/10.48550/arXiv.2312.00109

Maio, R., Santos, A., Marques, B., Ferreira, C., Almeida, D., Ramalho, P., Batista, J., Dias, P., & Santos, B. S. (2023). Pervasive Augmented Reality to support logistics operators in industrial scenarios: A shop floor user study on kit assembly. *International Journal of Advanced Manufacturing Technology*, *127*(3–4), 1631–1649. https://doi.org/10.1007/s00170-023-11289-1

Marino, E., Barbieri, L., Sollazzo, C., & Bruno, F. (2024). An Augmented Reality tool to support Industry 5.0 operators in product assembly activities. *The International*

Journal of Advanced Manufacturing Technology, *135*(9), 4941–4961. https://doi.org/10.1007/s00170-024-14777-0

Meta Horizon Link App. (2026). https://www.meta.com/de-de/help/quest/1517439565442928/

Microsoft Ends Support for HoloLens: What This Means for Enterprise XR. (2024). https://hololight.com/news/microsoft-ends-support-for-hololens-what-this-means-for-enterprise-xr

Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. (1994). Augmented reality: A class of displays on the reality-virtuality continuum. *Telemanipulator and Telepresence Technologies*, *2351*. https://doi.org/10.1117/12.197321

Modernizing the Smart Manufacturing Industry with MQTT. (2020). https://www.hivemq.com/resources/modernizing-the-manufacturing-industry/

Mourtzis, D., Angelopoulos, J., & Panopoulos, N. (2024). Unmanned Aerial Vehicle (UAV) path planning and control assisted by Augmented Reality (AR): The case of indoor drones. *International Journal of Production Research*, *62*(9), 3361–3382. (176294441). https://doi.org/10.1080/00207543.2023.2232470

Mukherjee, S. (2026). *DigitalOcean*. https://www.digitalocean.com/community/tutorials/yolov8-for-gpu-accelerate-object-detection

Müller, T., Evans, A., Schied, C., & Keller, A. (2022). Instant Neural Graphics Primitives with a Multiresolution Hash Encoding. *ACM Transactions on Graphics*, *41*(4), 1–15. https://doi.org/10.1145/3528223.3530127

Nguyen, T., Tran, K. D., Raza, A., Nguyen, Q.-T., Bui, H. M., & Tran, K. P. (2023). Wearable Technology for Smart Manufacturing in Industry 5.0. In K. P. Tran (Ed.), *Artificial Intelligence for Smart Manufacturing: Methods, Applications, and Challenges* (pp. 225–254). Springer International Publishing. https://doi.org/10.1007/978-3-031-30510-8_11

Nitol, S., Gadow, V., & Ramy, H. (2025). Emerging Technologies in Augmented Reality (AR) and Virtual Reality (VR) for Manufacturing Applications: A Comprehensive Review. *Journal of Manufacturing and Materials Processing*, *9*(9). https://doi.org/10.3390/jmmp9090297

Numpy 2.4.2 (Version 2.4.2). (2026). [C, Python; MacOS, Microsoft :: Windows, POSIX, Unix].

Object Detection vs Object Recognition vs Image Segmentation. (2025, July 12). GeeksforGeeks. https://www.geeksforgeeks.org/machine-learning/object-detection-vs-object-recognition-vs-image-segmentation/

OpenCV: Detection of ArUco Markers. (2026). https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html

Opencv-python (Version 4.13.0.92). (2026). [C++, Python; MacOS, Microsoft :: Windows, POSIX, Unix]. https://github.com/opencv/opencv-python

Padilla, R., Netto, S. L., & da Silva, E. A. B. (2020). A Survey on Performance Metrics for Object-Detection Algorithms. *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, 237–242. https://doi.org/10.1109/IWSSIP48289.2020.9145130

Paho-mqtt 2.1.0 (Version 2.1.0). (2026). [Python; MacOS :: MacOS X, Microsoft :: Windows, POSIX]. http://eclipse.org/paho

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). *PyTorch: An Imperative Style, High-Performance Deep Learning Library* (arXiv:1912.01703). arXiv. https://doi.org/10.48550/arXiv.1912.01703

Perno, M., Hvam, L., & Haug, A. (2025). Uses and challenges of digital twins-based augmented reality in operator training and data visualization in process manufacturing lines. *Flexible Services and Manufacturing Journal*. https://doi.org/10.1007/s10696-025-09605-w

Raj, S., Karmakar, B., Kumar, G., Mukhopadhyay, A., Chandrahas, R., & Biswas, P. (2025). Comparing computer vision models for low resource dataset to develop a mixed reality based manual assembly assistant. *Discover Robotics*, *1*(1), 5. https://doi.org/10.1007/s44430-025-00005-1

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788. https://doi.org/10.1109/CVPR.2016.91

Rejeb, A., Keogh, J. G., & Leong, G. K. (2021). *Potentials and challenges of augmented reality smart glasses in logistics and supply chain management: A systematic literature review.—Datensatzdetails—EBSCOhost Research Databases*. https://research-ebsco-com.ezproxy.hnu.de/c/6h4qqv/search/details/xbtxijacqv?db=bsh&limiters=FT%3AY%2CDT1%3A2020-10-21%2F2025-10-21&q=%28%0A%20%20%22Augmented%20Reality%22%20OR%20AR%20OR%0A%20%20%22Smart%20Glasses%22%20OR%20SG%20OR%0A%20%20%22Head%20Mounted%20Display%22%20OR%20HMD%20OR%0A%20%20Wearable%2A%20OR%0A%20%20%22Extended%20Reality%22%20OR%20XR%20OR

%0A%20%20%22Mixed%20Reality%22%20OR%20MR%20OR%0A%20%20%22Data%20Glasses%22%0A%29%0AAND%0A%28%0A%20%20Manufacturing%20OR%20Production%20OR%0A%20%20Assembly%20OR%0A%20%20Maintenance%20OR%20Repair%20OR%0A%20%20Logistics%20OR%0A%20%20%22Industrial%20Process%2A%22%20OR%0A%20%20%22Smart%20Factory%22%20OR%0A%20%20%22Order%20Picking%22%0A%29%0AAND%0A%28%0A%20%20%22Artificial%20Intelligence%22%20OR%20AI%20OR%0A%20%20%22Machine%20Learning%22%20OR%20ML%20OR%0A%20%20%22Digital%20Twin%22%20OR%20DT%20OR%0A%20%20%22Industry%204.0%22%20OR%20%22Industry%205.0%22%20OR%0A%20%20Augmentation%20OR%0A%20%20Usability%20OR%20Acceptance%20OR%0A%20%20%22Human%20Factor%2A%22%20OR%20%22Cognitive%20Load%22%20OR%0A%20%20%22Risk%20Management%22%20OR%20Safety%20OR%0A%20%20Productivity%20OR%20Efficiency%20OR%0A%20%20Training%20OR%20%22Glasses%20Technology%22%0A%29&searchMode=boolean

Ren, S., He, K., Girshick, R., & Sun, J. (2016). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks* (arXiv:1506.01497). arXiv. https://doi.org/10.48550/arXiv.1506.01497

Requirements for Meta Horizon Link. (2026). https://www.meta.com/de-de/help/quest/140991407990979/

Roldán, J. J., Crespo, E., Martín-Barrio, A., Peña-Tapia, E., & Barrientos, A. (2019). A training system for Industry 4.0 operators in complex assemblies based on virtual reality and process mining. *Robotics & Computer-Integrated Manufacturing*, *59*, 305–316. (136728749). https://doi.org/10.1016/j.rcim.2019.05.004

Sahu, C. K., Young, C., & Rai, R. (2021). Artificial intelligence (AI) in augmented reality (AR)-assisted manufacturing applications: A review. *International Journal of Production Research*, *59*(16), 4903–4959. (151932914). https://doi.org/10.1080/00207543.2020.1859636

Sai Guptha, V. D. P., Anamalamudi, S., Madupuri, R. P., & Sivarajan, S. (2025). Interoperability Between OPC-UA and MQTT in Hybrid Industrial IoT(IIoT) Environments. *2025 17th International Conference on COMmunication Systems and NETworks (COMSNETS)*, 1189–1194. https://doi.org/10.1109/COMSNETS63942.2025.10885710

Sara, G., Todde, G., & Caria, M. (2022). Assessment of video see-through smart glasses for augmented reality to support technicians during milking machine maintenance. *Scientific Reports*, *12*(1), 15729. https://doi.org/10.1038/s41598-022-20154-2

Schmitt, T., Viklund, P., Sjölander, M., Hanson, L., Urenda Moris, M., & Amouzgar, K. (2025). Augmented Reality for Machine Monitoring in Industrial Manufacturing: A Media Comparison in Terms of Efficiency, Effectiveness, and Satisfaction. *IEEE Access*, *13*, 82129–82143. https://doi.org/10.1109/ACCESS.2025.3566442

Shkarupylo, V., Alsayaydeh, J. A. J., Yusof, M. F. B., Oliinyk, A., Artemchuk, V., & Herawan, S. G. (2024). Exploring the Potential Network Vulnerabilities in the Smart Manufacturing Process of Industry 5.0 via the Use of Machine Learning Methods. *IEEE Access*, *12*, 152262–152276. https://doi.org/10.1109/ACCESS.2024.3474861

Silva, D., Carvalho, L. I., Soares, J., & Sofia, R. C. (2021). A Performance Analysis of Internet of Things Networking Protocols: Evaluating MQTT, CoAP, OPC UA. *Applied Sciences*, *11*(11), 4879. https://doi.org/10.3390/app11114879

Silva, J., Nogueira, P., Martins, P., Vaz, P., & Abrantes, J. (2023). Exploring the Potential of Mixed Reality as a Support Tool for Industrial Training and Production Processes: A Case Study Utilizing Microsoft HoloLens. In D. H. de la Iglesia, J. F. de Paz Santana, & A. J. López Rivero (Eds.), *New Trends in Disruptive Technologies, Tech Ethics and Artificial Intelligence* (pp. 187–196). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-38344-1_18

Skarbez, R., Smith, M., & Whitton, M. C. (2021). Revisiting Milgram and Kishino's Reality-Virtuality Continuum. *Frontiers in Virtual Reality*, *2*. https://doi.org/10.3389/frvir.2021.647997

soumyadip. (2023, December 26). *Mastering All YOLO Models from YOLOv1 to YOLOv12*. https://learnopencv.com/mastering-all-yolo-models/

Spatial anchors. (2025, August). https://developers.meta.com/horizon/design/mr-design-spatial-anchors/

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958.

Suh, S., Rey, V. F., & Lukowicz, P. (2024). Wearable Sensor-Based Human Activity Recognition for Worker Safety in Manufacturing Line. In J. Soldatos (Ed.), *Artificial Intelligence in Manufacturing: Enabling Intelligent, Flexible and Cost-Effective Production Through AI* (pp. 303–317). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-46452-2_17

Sutherland, I. (1965). The Ultimate Display. *Proceedings of the IFIPS Congress 65(2):506-508. New York: IFIP*, *2*.

Syberfeldt, A., Danielsson, O., & Gustavsson, P. (2017). Augmented Reality Smart Glasses in the Smart Factory: Product Evaluation Guidelines and Review of Available Products. *IEEE Access*, *5*, 9118–9130. https://doi.org/10.1109/ACCESS.2017.2703952

TLX @ NASA Ames. (2020). https://humansystems.arc.nasa.gov/groups/tlx/

Trivedi, C., Bhattacharya, P., Prasad, V. K., Patel, V., Singh, A., Tanwar, S., Sharma, R., Aluvala, S., Pau, G., & Sharma, G. (2024). Explainable AI for Industry 5.0: Vision, Architecture, and Potential Directions. *IEEE Open Journal of Industry Applications*, *5*, 177–208. https://doi.org/10.1109/OJIA.2024.3399057

Tsirtsakis, P., Zacharis, G., Maraslidis, G. S., & Fragulis, G. F. (2025). Deep learning for object recognition: A comprehensive review of models and algorithms. *International Journal of Cognitive Computing in Engineering*, *6*, 298–312. https://doi.org/10.1016/j.ijcce.2025.01.004

Ultralytics. (2026a). *Advanced Customization*. https://docs.ultralytics.com/usage/engine/

Ultralytics. (2026b). *Ultralytics YOLO11*. https://docs.ultralytics.com/models/yolo11/

Ultralytics. (2026c). *Yolo: A Brief History*. https://docs.ultralytics.com/

Unity.com. (2026). Unity. https://unity.com

Unity-MRUtilityKitSample. (2025). GitHub. https://github.com/oculus-samples/Unity-MRUtilityKitSample

van der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering*, *13*(2), 22–30. https://doi.org/10.1109/MCSE.2011.37

Villani, V., Picone, M., Mamei, M., & Sabattini, L. (2025). A Digital Twin Driven Human-Centric Ecosystem for Industry 5.0. *IEEE Transactions on Automation Science and Engineering*, *22*, 11291–11303. https://doi.org/10.1109/TASE.2024.3410703

Wang, B., Zheng, L., Wang, Y., Fang, W., & Wang, L. (2024). Towards the industry 5.0 frontier: Review and prospect of XR in product assembly. *Journal of Manufacturing Systems*, *74*, 777–811. https://doi.org/10.1016/j.jmsy.2024.05.002

Wang, L., Xu, W., Liu, Y., Wang, M., Wang, B., Yang, X., Xu, L., Tan, Z., Fan, R., Wang, Z., Wang, C., Zhang, H., Wen, Y., Yang, H., Wu, J., Fan, J., Wang, H., Zhang, Q., Wang, G., … Zhao, Q. (2025). Artificial intelligence for virtual reality: A review. *Science China Information Sciences*, *69*(1), 111101. https://doi.org/10.1007/s11432-024-4541-9

Web Development. (2024, August 31). https://inventivehq.com/blog/why-does-base64-increase-file-size

Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. *MIS Quarterly*, *26*(2), xiii–xxiii.

Wei, W., Liu, L., Yang, M., Li, J., & Wu, F. (2022). Predictive Maintenance System for Production Line Equipment Based on Digital Twin and Augmented Reality. In Y. Wang, K. Martinsen, T. Yu, & K. Wang (Eds.), *Advanced Manufacturing and Automation XI* (pp. 479–486). Springer. https://doi.org/10.1007/978-981-19-0572-8_61

Wen, D., Yuan, J., Li, J., Sun, Y., Wang, X., Shi, R., Wan, X., Zhou, Y., Song, H., Dong, X., Xu, F., Lan, X., & Jung, T.-P. (2023). Design and Test of Spatial Cognitive Training and Evaluation System Based on Virtual Reality Head-Mounted Display With EEG

Recording. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *31*, 2705–2714. https://doi.org/10.1109/TNSRE.2023.3283328

Windhausen, A., Heller, J., Hilken, T., Mahr, D., Di Palma, R., & Quintens, L. (2024). Exploring the impact of augmented reality smart glasses on worker well-being in warehouse order picking. *Computers in Human Behavior*, *155*, 108153. https://doi.org/10.1016/j.chb.2024.108153

Xi, X., Huang, Y., Luo, R., & Qiu, Y. (2025). OW-OVD: Unified Open World and Open Vocabulary Object Detection. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 25454–25464. https://doi.org/10.1109/CVPR52734.2025.02370

XREAL project aura. (2025). https://www.xreal.com/aura

Xu, Q., Xu, Z., Philip, J., Bi, S., Shu, Z., Sunkavalli, K., & Neumann, U. (2023). *Point-NeRF: Point-based Neural Radiance Fields* (arXiv:2201.08845). arXiv. https://doi.org/10.48550/arXiv.2201.08845

Yazdi, M. (2024). Augmented Reality (AR) and Virtual Reality (VR) in Maintenance Training. In M. Yazdi (Ed.), *Advances in Computational Mathematics for Industrial System Reliability and Maintainability* (pp. 169–183). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-53514-7_10

Yu, A., Fridovich-Keil, S., Tancik, M., Chen, Q., Recht, B., & Kanazawa, A. (2021). *Plenoxels: Radiance Fields without Neural Networks* (arXiv:2112.05131). arXiv. https://doi.org/10.48550/arXiv.2112.05131

Zhang, X., Wenlei, S., Lixin, W., Hua, Y., & Ke, C. (2025). Virtual reality integration and credible traceability management method of transformer coil manufacturing process based on industrial internet and digital twin. *International Journal of*

Computer Integrated Manufacturing, 1–28. (182359985). https://doi.org/10.1080/0951192x.2025.2452615

Zigart, T., Stürzl, F., Niedermayr, D., Wolfartsberger, J., & Sorko, S. R. (2023). Comparative Evaluation of Virtual Reality and In-Person Onboarding for Assembly Trainings in Manufacturing. *2023 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, 167–174. https://doi.org/10.1109/ISMAR-Adjunct60411.2023.00041

Appendix A - Market Screen Device Evaluation Table

Also available in a separate Excel file in the appendix folder of the submission

Ranking	1	2	3	4	5	6	7	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Device	**XREAL Air 2 Ultra**	Pico 4 Ultra	HTC Vive XR Elite	Meta Quest 3	Magic Leap 2	ThirdEye Gen X2	Apple Vision Pro	RealWear Navigator 520	Vuzix M400	Meta Ray-Ban gen 2	DigiLens ARGO	VITURE Luma Ultra XR/AR Glasses	Rokid AR max 2	RayNeo Air 3S	Lenovo ThinkReality A3	XREAL One Pro	Snap Spectacles 5	Vuzix Shield	Microsoft HoloLens 2	INMO Air 2	Brilliant Labs Frame
Theoretical Knockout Criteria	-	no smart glass	no smart glass	no smart glass	price	price	price	price + interactivity	price + interactivity	interactivity	price + avaliability	no camera +interactivity	no camera +interactivity	no camera +interactivity	avaliability + interactivity	interactivity + camera add on	avaliability	avaliability	avaliability	avaliability + interactivity	avaliability + interactivity
Picture																					
avaliable	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	no	yes	no	no	no	no	preorder
Price (Germany)	891,57 €	€ 399	~€699	€ 549	~€4,298.99	$ 2,450	from €3.699	~€3,000	~$1,799.99	€ 499	-	$599	~€599 - €749	€ 288,15	~€1.400 - €1.600	€ 689	-	-	-	€ 879	preorder
spatial-computing	See-Through	Video pass-through	Video pass-through	Video pass-through	See-Through	See-Through	Video pass-through	See-Through	See-Through	See-Through	See-Through	See-Through	See-Through	See-Through	See-Through	See-Through	Optical see-through	Optical see-through	Video pass-through	Optical see-through	Optical see-through
Weight	83g	580g	625g (Full), 273g (Glasses mode)	515g	260g	~168g	750-800g (+353g Battery)	274g	180g	49g	~200g	83g	49g	~76g	130g	87 g	226g	~160g	566g	~99g	39g
Interactivity	Hand Tracking, 6DoF Head, Phone Screen	Controllers, Hands	Controllers, Hand Tracking	Controllers, Hand Tracking	Hand/Eye Tracking, Controller, Voice	Gaze, Hand Gestures	Eyes, Hands, Voice	Voice	Voice, Buttons, Touchpad	Voice, Touch, Wrist	Voice, Gaze, Hands, Scroll Wheel	Device dependent (Phone/PC)	Touchpad (Station), Head Motion	Phone/External Device	Gaze, Voice, Phone	phone screen, Buttons	Hand Tracking, Voice	Hand Tracking, Voice	Full Hand Tracking, Eye	Ring, Touch, Voice	Voice, Head Tilt
Development Platforms	NRSDK,Android XR, WebXR	Pico SDK, Android XR, Open XR	VIVE Wave, OpenXR	OpenXR, Oculus, Android XR, Meta SDK	WebXR, OpenXR, AndroidXR	Android XR	Xcode, RealityKit, OpenXR	WearHF	Android XR	Meta Spark / SDK (closed, for meta only)	Snapdragon Spaces, DigiOS SDK	Viture SDK	YodaOS SDK	N/A	Snapdragon Spaces	NRSDK,Android XR, WebXR	N/A	N/A	deprecated support	N/A	N/A
Development Programs	Unity, Unreal	Unity, Unreal	Unity, Unreal	Unity, Unreal	Unity, Unreal, Vuforia	Unity, Android Studio	Unity	N/A	Vuzix App	N/A	Unity	SpaceWalker, Unity	N/A	N/A	N/A	Unity, Unreal	Lens Studio	Unity, Unreal	Unity, Unreal	Android Studio	N/A
Framerate	90 Hz (3D mode) / 120 Hz (2D)	90 Hz	90 Hz	72/90/120 Hz	120 Hz	60Hz	90/96/100/120 Hz	60 Hz	60 Hz	N/A (No Display)	60 Hz	120 Hz	90/120 Hz	120 Hz	60 Hz	120 Hz	60Hz	60Hz	60Hz	60Hz	N/A (Static/Low)
Weight distribution	Nose-heavy (Glasses style)	Balanced (Battery in rear)	Balanced (Battery in back)	Front-heavy	Light Headset (Compute Pack on hip)	Front Heavy	Front-heavy	Balanced (Boom arm)	Side-mounted (Requires frames)	Excellent	Balanced (Rugged Frames)	Nose-heavy (Glasses style)	Glasses (Light), Station (Handheld)	Nose-heavy	Front-heavy	Front-heavy (Standard glasses form factor)	Balanced (Thick temples)	Front Heavy	Balanced (Halo Strap)	Balanced	Excellent
Powering	Tethered (USB-C)	Internal Battery	Hot-swappable Battery Cradle	Internal Battery	Tethered Compute Pack	Standalone	External Battery (Tethered)	Hot-swappable Battery	Hot-swappable Battery	Standalone (Case)	Internal + USB-C (Hot-swap capable)	Tethered (USB-C)	Station 2 Battery (5000mAh)	Tethered (USB-C)	Tethered (USB-C)	Tethered (USB-C)	Standalone	Standalone	Standalone	Standalone	Standalone (batteries)
Field of view	52°	105°	110°	110° H / 96° V	70°	42° (Diagonal)	~100°	24°	16.8°	N/A	30°	46°	50°	46°	~45-50°	57° (Diagonal)	46° (Diagonal)	30° (Diagonal)	52° (Diagonal)	26° (Diagonal)	20° (Diagonal)
Battery Life	None (Host dependent)	~2.5 Hours	~2 Hours	~2.2 Hours	up to 3.5 Hours	~2 Hours	~2 Hours	6-8 Hours	2-12 Hours (Battery dependent)	~4 Hours	~4-6 Hours (Usage dependent)	None (Host dependent)	~5 Hours (Video)	None (Host dependent)	None (Host dependent)	None (Host dependent)	~45 Mins (Active)	~3-4 Hours	~2-3 Hours	~3-4 Hours	All Day (Assisted)
Optics	Sony Micro-OLED 1920x1080p/eye	Pancake, 2160x2160/eye	Pancake, 1920x1920/eye	Pancake, Dual LCD (2064x2208/eye)	LCoS Waveguide (1440x1760/eye)	Binocular OLED (720p)	Micro-OLED (23M Pixels)	24-bit color LCD (HD 1280x720 resolution)	OLED nHD	None (Cameras only)	Crystal30 Waveguide	Micro-OLED 1200p	Micro-OLED 1920x1200	SeeYa Micro-OLED	1080p/eye	Micro-OLED (1920x1080 per eye)	LCoS Waveguide	Binocular MicroLED	Laser Beam Scanning	Binocular Waveguide	Monocular Prism (Red)
Built-in Camera Specs	Dual 3D Sensors (Grayscale/Tracking)	Dual 32 MP	16 MP (RGB Passthrough)	Dual RGB (Approx. 4MP, Fixed Focus)	12 MP (RGB Auto-Focus)	13MP Auto-Focus	6.5 MP (Stereo/3D Main)	48 MP (High Res, Auto-Focus)	12.8 MP (Auto-Focus)	12MP Ultra-Wide	48 MP (High Res, Auto-Focus)	None	None	None	8 MP RGB + Dual Fisheye	as add on: 12 MP (Requires Eye Module)	4x CV Cameras	2x 13MP Stereo	8MP + 4x Tracking	8MP + AI Cam	720p AI Cam
Native CV Capability (Internal Cam)	Limited Access allowed, but sensors are low-detail grayscale (bad for QC).	Open / Native (Enterprise SDK) Raw camera access is allowed for developers.	Blocked Privacy locked. No raw pixel access.	Blocked Privacy locked. No raw pixel access.	Restricted Requires Enterprise API access.	Yes (Snapdragon xR1)	Restricted Blocked by default. Requires Enterprise Entitlement.	Open / Native Standard Android access. Best for detailed QC.	Open / Native Standard Android access. Easy to implement.	No (Cloud/Phone)	Open / Native Full raw access. Excellent for industrial QC.	Impossible No camera hardware.	Impossible No camera hardware.	Impossible No camera hardware.	Good (Tethered) CV processed by the connected PC or Motorola phone.	Feasible (Tethered) Video sent to phone; processed by phone app.	Yes (Custom Snap OS)	Yes (8-Core CPU)	Yes (HPU 2.0)	Yes (Android)	No (Cloud AI)
External Camera Capability	Difficult USB-C port is occupied by the phone connection. Needs a hub.	Yes (Direct USB)Supports standard UVC webcams via the USB-C port.	Yes (Direct USB) Supports UVC webcams via USB-C port.	Yes (Direct USB) Supports UVC webcams via Android USB-C.	Difficult USB-C port exists, but mounting/drivers are complex.	Yes (USB-C)	No / Very Difficult No data port available without Developer Strap; driver support poor.	Yes (Direct USB) Supports standard USB cameras via side port.	Yes (Direct USB) Supports standard USB cameras via USB-C.	No	Yes (Direct USB) Supports standard USB cameras via USB-C.	Via Host Device Must plug camera into the phone/PC, not the glasses.	Via Host Device Must plug camera into the phone/PC, not the glasses.	Via Host Device Must plug camera into the phone/PC, not the glasses.	Via Host Device Easy to add camera to the PC running the glasses.	Via Host Device Must plug camera into the phone/Beam, not the glasses.	No	No	Yes (USB)	USB-C (Limited)	No
Open API	Yes (NRSDK for Developers)	Yes	Yes (OpenXR)	Yes (OpenXR)	Yes	yes (Android)	Restricted (Apple Ecosystem)	Enterprise SDK	Yes (Android)	Low (Restricted)	Yes (Android AOSP / OpenXR)	VITURE Unity XR SDK	Yes	N/A	Yes (Snapdragon Spaces)	Yes (via XREAL NRSDK for Android/Unity)	yes (Snap Only)	yes (Android)	yes (Windows/OpenXR)	High (Android)	Very High (Open)
Audio	Directional Stereo Speakers, 2 Mics	Stereo Spatial Audio	Integrated Speakers/Mics	Integrated Spatial Audio	Spatial Audio, 4 Mics	Stereo Speakers	Spatial Audio Pods	Integrated Speaker/Mic (Noise Cancellation)	Speaker, Noise Cancelling Mic	Open Ear Speakers	Stereo Spatial Speakers, 5 Mics	Harman Speakers	Integrated Speakers	3D Surround	3 Mics, Stereo Speakers	Open-ear speakers Dual Microphones	Spatial Audio	Stereo Speakers	Spatial Audio	Stereo Speakers	No (Mic only)
Sensors	2x 3D Env.[2][3][4] Sensors, IMU, Depth	4 Env Cams, 2 RGB, iToF	4 Tracking Cams, Depth Sensor	4 Tracking Cams, Depth, IMU	Eye, Hand, Depth, IMU, Magnetometer	SLAM, Depth, IMU	LiDAR, TrueDepth, IR, Eye Tracking	IMU, GPS	GPS, IMU	Capsense, IMU	6DoF Tracking Cams, GPS, IMU	IMU (3DoF Beta)	IMU (Glasses), Station Sensors	Accelerometer, Gyro, Proximity	6DoF Tracking Cams	IMU	6DoF, Depth, IMU	6DoF, Depth	Depth (ToF), IMU, Eye	IMU, Touch	IMU
Controls	Hands, Head, Host Phone/Beam Pro	Controllers, Hands	Controllers, Hands	Touch Plus Controllers, Hands	6DoF Controller, Gaze, Hand, Voice	Head, Voice, Controller	Eyes, Hands, Voice	Voice Commands (Deutsch)	Voice, Touchpad, Buttons	Voice, Tap, Neural	Voice, Scroll Wheel (Temple), Hands	Host Device, Buttons on frame, Hands	Touch (Station), Head	Phone/External Device	Gaze, Phone	On-frame buttons	Hand, Voice	Touch, Voice, Gesture	Hand, Eye, Voice	Ring, Touch, Voice	Voice, Head Gesture
Processors	None (Display only) / Host dependent	Snapdragon XR2 Gen 2	Snapdragon XR2	Snapdragon XR2 Gen 2	AMD Quad-core Zen 2	Snapdragon XR1	Apple M5 + R1	Snapdragon 662	Snapdragon XR1	Snapdragon AR1 Gen 1	Snapdragon XR2 Gen 1	None (Display only)	Snapdragon Platform (Station 2)	None (Display only)	Snapdragon XR1 SmartViewer	XREAL X1	Dual Snapdragon	8-Core ARM	Snapdragon 850 + HPU	Quad Core (UNISOC)	Nordic MCU + FPGA
Storage	None	256 GB	128 GB	512 GB	256 GB	32GB	256GB / 512GB / 1TB	64 GB	64 GB	32GB	128 GB	None (Host)	128 GB (Station 2)	None	None (Host)	None (Host)	N/A (Cloud/Stream)	64GB	64GB	32GB	N/A
RAM	None	12 GB	12 GB	8 GB	16 GB	4GB	16 GB	4 GB	6 GB	N/A	12 GB	None	8 GB (Station 2)	None	None (Host)	None	N/A	6GB	4GB	2GB	N/A
Connectivity	USB-C (DP Alt Mode)	Wi-Fi 7, BT 5.3	Wi-Fi 6E, USB-C, BT 5.2	Wi-Fi 6E, BT 5.2	Wi-Fi 6, BT 5.1	Wi-Fi, BT	Wi-Fi 6, BT 5.3	Wi-Fi 5, BT 5.1	Wi-Fi 5, BT 5.0, USB-C	Wi-Fi 6, BT 5.3	Wi-Fi 6E, BT 5.2, GPS, USB-C	USB-C (DP Alt Mode)	Wi-Fi 6, BT 5.2, USB-C	USB-C (DP Alt Mode)	USB-C	USB-C (DisplayPort Alt Mode)	Wi-Fi 6, BT	Wi-Fi, BT	Wi-Fi 5, BT 5.0	Wi-Fi, BT	BT 5.0
Operating System	N/A (Nebula OS runs on Host)	Pico OS (Android 14)	VIVE Platform (Android based)	Meta Horizon OS (Android)	Magic Leap OS (Android based)	Android 8/9	visionOS	Android 11	Android 13	Meta OS	DigiOS (Android 12 AOSP)	N/A (Host dependent)	YodaOS-Master (Android)	N/A (Host dependent)	Host dependent	N/A (Nebula OS runs on Host)	Snap OS	Android 10/11	Windows Holographic	IMOS (Android 10)	Lua OS
Durable against dust	Not Rated	Not Rated	Not Rated	Not Rated	IP20 (Indoor)	IP54	Not Rated	IP66	IP67	IPX4	IP65	Not Rated	Not Rated	Not Rated	IP5X	Not Rated	Not Rated	IP54	Not Rated	Not Rated	Not Rated
Durable against water	Not Rated	Not Rated	Not Rated	Not Rated	Not Rated	IP54	Not Rated	IP66	IP67	IPX4	IP65	Not Rated	Not Rated	Not Rated	IPX4 (Splash resistant)	Not Rated	Not Rated	IP54	Not Rated	Not Rated	Not Rated
Appearance	Sunglasses (Titanium Frame)	VR Visor	Convertible Visor	VR Visor (Slimmer)	Goggles / Visor	Utility	Curved Glass Visor	Monocular Boom	Monocular Attachment	Normal Glasses	Thick Industrial Safety Glasses	Sunglasses	Sunglasses	Sunglasses	Bulky Glasses	Sunglass-style	Futuristic	Industrial	Utility	Chunky Glasses	Retro / Hipster
Aesthetic look	Modern Tech / Premium Consumer	Consumer White/Grey	Sleek Matte Black	Consumer Electronics (White)	Futuristic Tech	Enterprise / Industrial	Premium Consumer	Industrial Rugged	Functional Industrial	Fashion-First	Rugged Enterprise / Industrial	Stylish / Casual	Modern Consumer	Consumer	Enterprise / Industrial	Modern	Modern	Consumer	Enterprise / Industrial	Tech-Forward	Minimalist/Geek
Compability	Android, Mac, Windows, iPhone 15+	Standalone, PCVR	Standalone, PCVR	Standalone, PCVR	Standalone (with Puck)	Android	Standalone	Standalone	Standalone	Android / iOS	Standalone	PC, Mac, Mobile, Consoles	Standalone (via Station)	Phone, PC, Console	Motorola Phones, PC	Phone	Standalone	Android / iOS	PC / Standalone	Android / iOS	Android / iOS
Compactness	Foldable, Compact	Bulky Visor	Modular / Foldable (Glasses mode)	Bulky Visor	Rigid Headband (Bulky)	Medium (Foldable)	Bulky Visor	Foldable boom, bulky body	Compact Unit (excl. frames)	High	Non-foldable (Rigid Frame)	Foldable, Compact	Glasses fold, Station is small	Foldable	Foldable but thick	Foldable	Medium (Foldable)	Medium (Foldable)	Low (Rigid Headset)	Medium	High
Data security (DE/EU)	Host dependent (Nebula App requires Login)	Consumer (ByteDance)	Consumer / Enterprise	Consumer (Meta Account)	High (GDPR/ISO 27001)	High (Enterprise)	High (Optic ID, Secure Enclave)	High (GDPR Compliant)	High (GDPR Compliant)	Low (US Cloud/Meta)	High (Enterprise MDM / AOSP)	Host dependent	Consumer	Host dependent	High (Enterprise)	High (Data resides on the connected host device).	High (US/EU GDPR)	High (Enterprise)	High (Enterprise)	Low (China Cloud)	High (Open Source)
Design frame	Titanium Ring, 3-position nose pads	Rigid Strap, Plastic	Plastic	Plastic, Fabric Strap	Plastic/Metal, Rigid	Plastic	Aluminum, Glass, Fabric	Ruggedized, PPE Compatible	Mounts to Safety Frames	Plastic/Acetate	Rugged, ANSI Z87.1 Safety Rated	Aluminum, Titanium hinge	Plastic/Metal	Plastic	Safety Glass (ANSI Z87.1)	Plastic/Metal	Magnesium/Plastic	Safety Rated Polycarb	Carbon Fiber / Plastic	Plastic	Nylon/Plastic
Durability	Consumer Grade	Consumer Grade	Consumer Grade	Consumer Grade	Moderate	High (Drop tested)	Low (Glass front)	High (2m Drop Tested)	High (2m Drop Tested)	High	MIL-STD-810H (Shock/Drop/Temp)	Consumer Grade	Consumer Grade	Consumer Grade	Moderate (IP54)	Moderate	Medium	High (ANSI Z87.1)	Medium	Medium	Delicate
Ergonomics	3-Level Temple adjustment, Nose pads	Balanced Design	Adjustable IPD, Diopter	Y-Strap (Adjustable)	Adjustable Headband, Nose pads	OK (Strap recommended)	Solo Knit / Dual Loop Bands	Adjustable Boom, Head strap	Left/Right Eye Switchable	Excellent	Interchangeable nose pads, Wheel interface	Myopia adjust, Electrochromic	Myopia adjust, Nose pads	Adjustable Nose Pads	Nose pads, Ear horns	Adjustable IPD, Adjustable nose pads and temple arms.	Good (Thick arms)	Good (Nose pads)	Excellent (Halo)	Good	Good
Tracking	6DoF (Inside-out)	6DoF	6DoF Inside-Out	6DoF (Inside-out)	6DoF (SLAM)	6DoF (VisionEye)	6DoF, Eye Tracking	None (Assisted Reality)	3DoF (Head Tracking)	N/A	6DoF (Inside-out)	6DoF	3DoF	3DoF	6DoF	3DoF + 6DoF avaliable	6DoF	6DoF	6DoF	3DoF (SLAM Lite)	N/A
last viewed date	29.11.2025	29.11.2025	29.11.2025	29.11.2025	29.11.2025	30.11.2025	29.11.2025	29.11.2025	29.11.2025	30.11.2025	29.11.2025	29.11.2025	29.11.2025	29.11.2025	29.11.2025	29.11.2025	30.11.2025	30.11.2025	30.11.2025	30.11.2025	30.11.2025
Datasheet	XREAL	PICO4 Ultra	VIVE XR Elite Specs	Meta Quest 3	3,37343E+13	ThirdEye	Apple Vision Pro	RealWear Navigator 520	Vuzix M400	Ray-Ban-Meta-Brochure.pdf	digilens.com	VITURE Luma Ultra XR/AR	Rokid Max 2	RayNeo Air 3s Pro Smart	ThinkReality A3 - Datasheet	XREAL One Pro	Lens Studio	-	HoloLens_Generic_DataSheet	INMO Air2 Documentation	Welcome | Brilliant Documentation
Shop	XREAL Air 2 Ultra	MediaMarkt	VIVE XR Elite	Meta Quest 3	Magic Leap	X2 Mixed Reality Headset	Apple Vision Pro	RealWear Navigator® 520	Vuzix M400	Ray-Ban	DigiLens.com/visualize	VITURE Luma Ultra XR/AR	Rokid Max 2	RayNeo Air 3s Pro Smart	ThinkReality A3 PC Edition	XREAL One Pro – XREAL EU shop	Spectacles	vuzix.com	-	inmoxr.com	Brilliant Labs

Appendix B – Overview AI Tree

Also available in a separate text file in the appendix folder of the submission

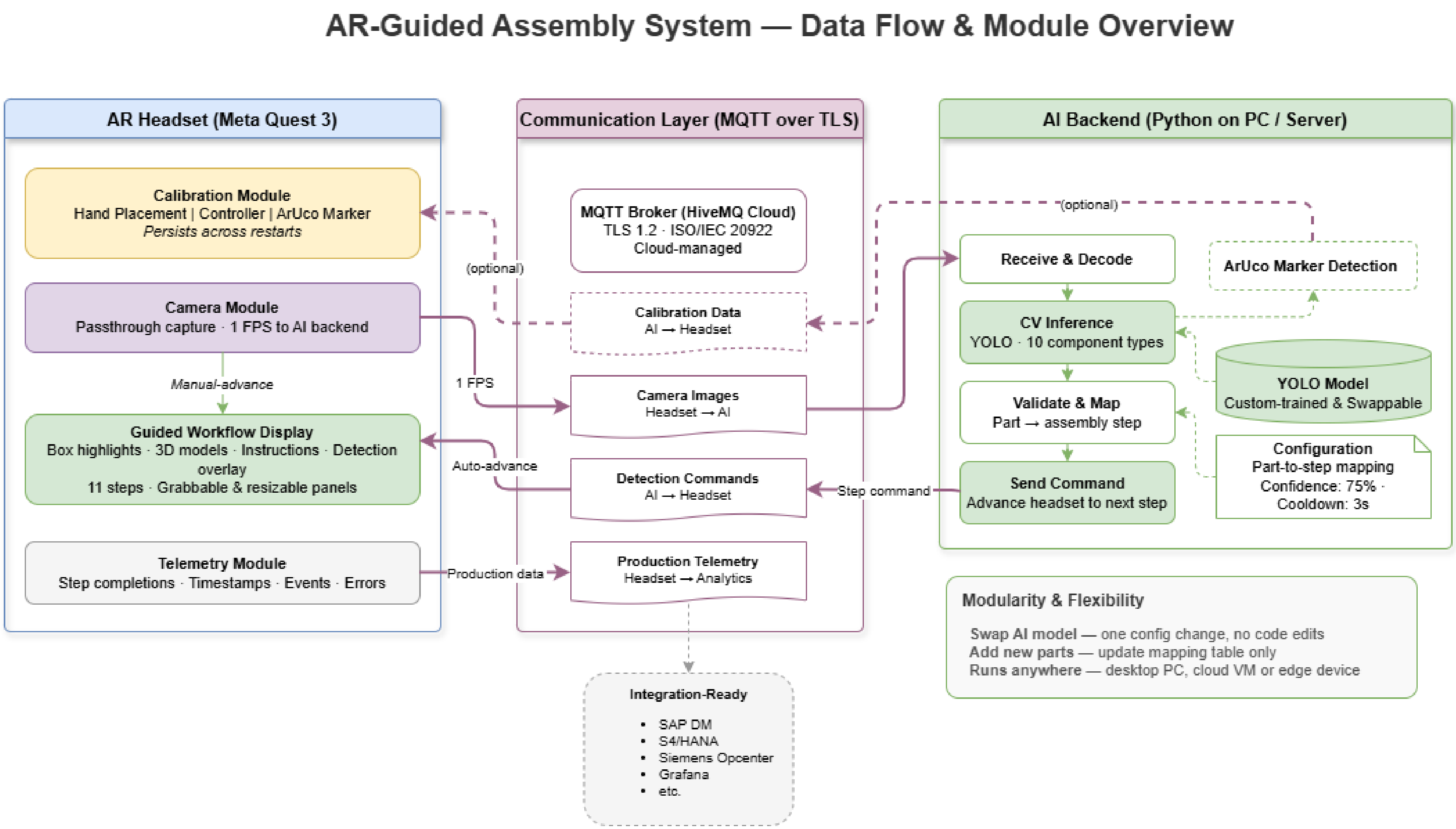
```
Artificial Intelligence (AI)
    │
    └── Machine Learning (ML)
        │
        ├── Supervised Learning
        │   ├── Regression
        │   ├── Classification
        │   │   ├── Support Vector Machines (SVM)
        │   │   ├── k-Nearest Neighbors (k-NN)
        │   │   ├── Decision Trees
        │   │   └── Random Forests
        │
        ├── Unsupervised Learning
        │   ├── Clustering
        │   │   ├── k-Means
        │   │   ├── Hierarchical Clustering
        │   │   └── DBSCAN
        │   └── Dimensionality Reduction
        │       ├── Principal Component Analysis (PCA)
        │       └── Autoencoders
        │
        ├── Semi-Supervised Learning
        │
        ├── Reinforcement Learning (RL)
        │   ├── Q-Learning
        │   ├── Policy Gradient Methods
        │   └── Deep Reinforcement Learning
        │
        └── Deep Learning (DL)
            │
            └── Computer Vision (CV)
                │
                └── Object Detection
                    │
                    ├── Classical Detection (Pre-Deep Learning)
                    │   │
                    │   ├── Sliding Window Methods
                    │   │
                    │   ├── Handcrafted Feature-Based Detection
                    │   │   ├── Haar Features
                    │   │   ├── Histogram of Oriented Gradients (HOG)
                    │   │   └── Scale-Invariant Feature Transform (SIFT)
                    │   │
                    │   └── Traditional Classifiers
                    │       ├── Support Vector Machines (SVM)
                    │       └── Boosting Methods
                    │
                    └── Deep Learning-Based Detection
                        │
                        ├── CNN-Based Detectors
                        │   │
                        │   ├── Two-Stage Detectors
                        │   │   ├── R-CNN
                        │   │   ├── Fast R-CNN
                        │   │   ├── Faster R-CNN
                        │   │   └── Cascade R-CNN
                        │   │
                        │   ├── One-Stage Detectors
                        │   │   ├── SSD
                        │   │   ├── YOLO
                        │   │   └── RetinaNet
                        │   │
                        │   ├── Anchor-Based Detectors
                        │   │   ├── Faster R-CNN
                        │   │   ├── SSD
                        │   │   └── RetinaNet
                        │   │
                        │   └── Anchor-Free Detectors
                        │       ├── FCOS
                        │       ├── CenterNet
                        │       └── CornerNet
                        │
                        └── Transformer-Based Detectors
                            ├── DETR
                            ├── Deformable DETR
                            └── Conditional DETR
```

Appendix C – Drone Paper Work Instruction

Available in a separate Word file in the appendix folder of the submission

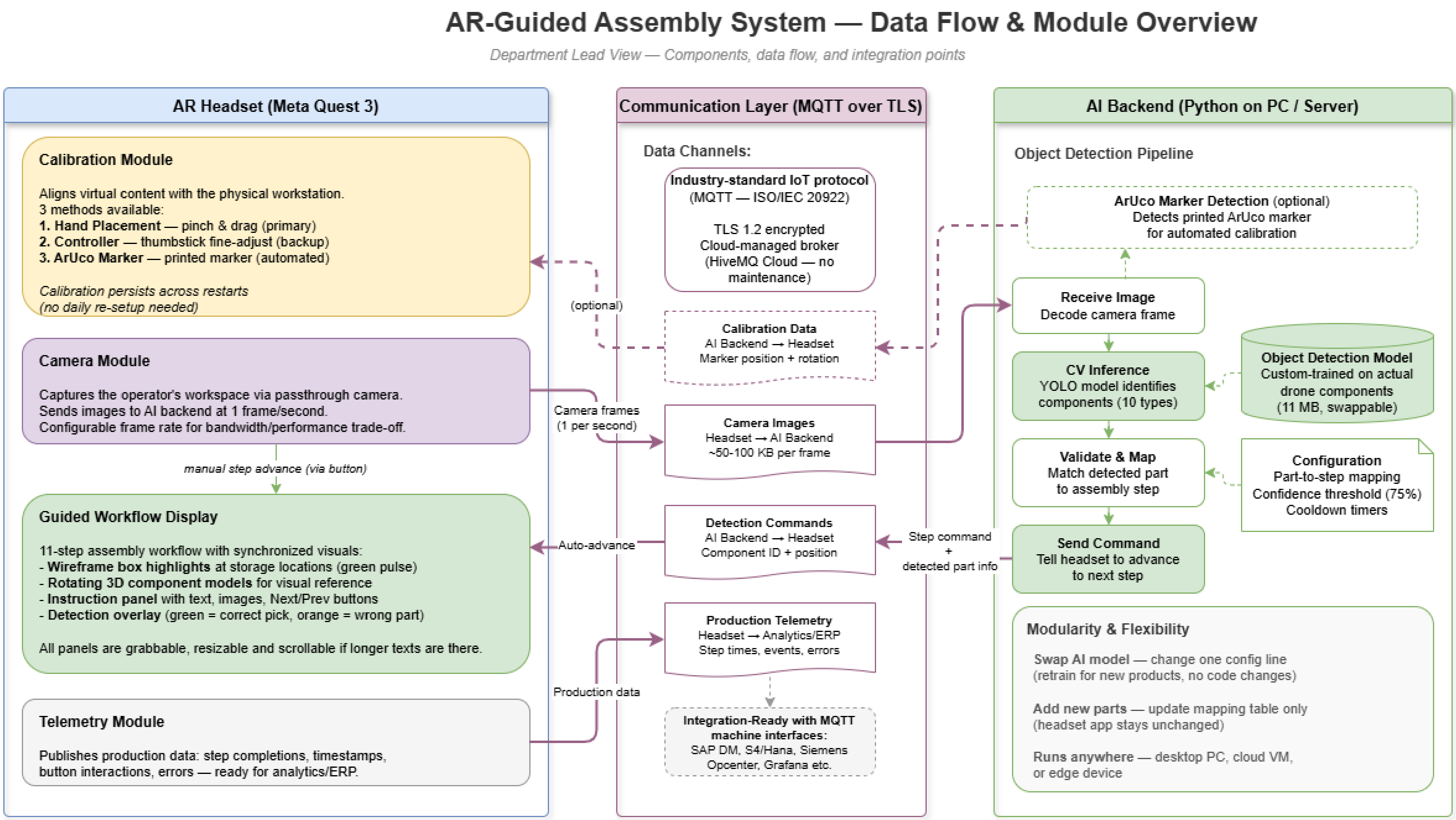
Appendix D – Low Level Architecture

Also available in a separate .drawio file in the appendix folder of the submission



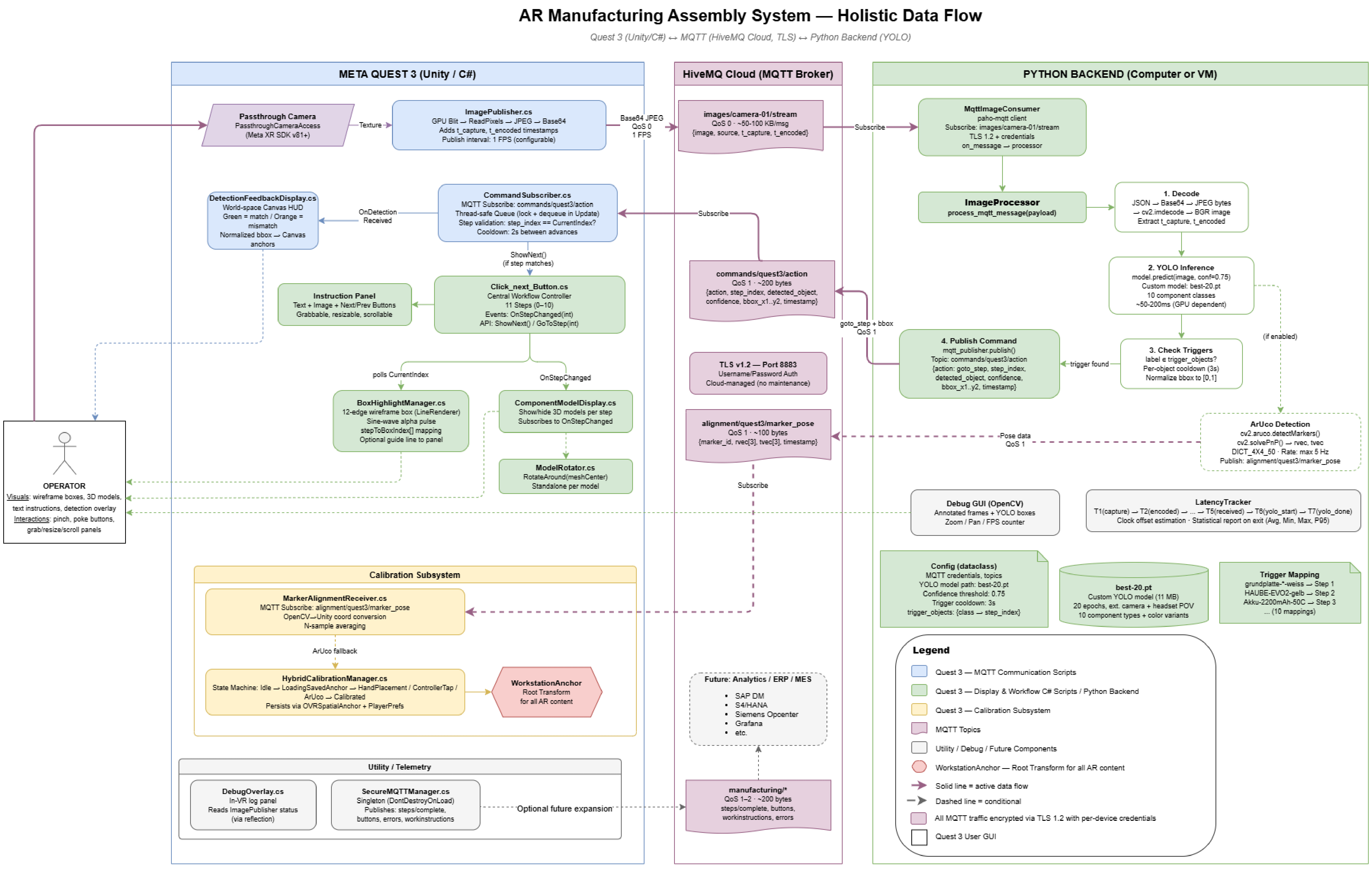
Appendix E – Mid Level Architecture

Also available in a separate .drawio file in the appendix folder of the submission



Appendix F – High Level Architecture

Also available in a separate .drawio file in the appendix folder of the submission



Appendix G – Artifact Doc without Unity Files

Available in a separate folder in the appendix folder of the submission

Appendix H - DemoVideo_SplitView

Available in a separate folder in the appendix folder of the submission

Appendix I - DemoVideo_OperatorView

Available in a separate folder in the appendix folder of the submission