**Ain Shams University**

**Faculty of Computer and Information Sciences**

**DIGITAL MULTIMEDIA PROGRAM**

# Transforming Learning with AR-Enabled Intelligence (The Learning Lens)

By

**Mahmoud Hany Mazroa**
**Ali Haitham Youssef**
**Karen Walid Magdy**
**Mohamed Shokry Mostafa**
**Mahmoud Sayed Abdein**

Under Supervision of
**Prof./Dr. Sherin Rady**
Professor in Information Systems Department,
Faculty of Computer and Information Sciences,
Ain Shams University

**TA. Yomna Ahmed**
Assistant Lecturer in Computer Science Department
Faculty of Computer and Information Sciences,
Ain Shams University

# Acknowledgement

We first and foremost express our gratitude to the Merciful Allah, who has endowed all with all that is needed to prepare and present this work. He gave me strength in my weakness, hope in my despair, and patience in our affliction.

We are very grateful to our parents and families for always being supportive and encouraging us throughout our school period. They have been the cause of our success through their sacrifices and love, and we pray to be able to return their kindness and effort one day.

We are most thankful to our esteemed supervisor, Dr. Sherin Rady, whose support and guidance were unabating. Her guidance, mentorship, and tireless effort were the nucleus of surmounting the challenges we encountered. Without her scholarly guidance and motivation, completion of this project would have been immensely difficult.

Special gratitude should be extended to T.A. Yomna Ahmed, whose precious assistance and unbreakable encouragement significantly facilitated the enrichment of our project's creation.

We also thank all those who, regardless of how little or significant a part, played a role in making our project successful. Last but not least, we also acknowledge the significant input of each member of our team whose collective effort and determination was priceless to the success of the project. We, as a team, have accomplished a milestone that we can be proud of, and for it, we are modestly grateful.

# Abstract

The project responds to the need for new training solutions within the automobile industry, particularly for beginner mechanics. Fusing Virtual Reality (VR) and Augmented Reality (AR) technologies, we designed an interactive learning platform with the aim of enhancing students' assembly skills and critical thinking abilities. Focusing on motorcycle assembly, the app enables users to have an immersive and interactive experience for both children and adults.

***Main Features:***

- Personal Account System: Allows the users to track their progress and tailor their learning experience.
- Engaging Learning Modules: Offers many courses and lessons specifically written to effectively instruct assembly procedures.
- Dynamic Model Retrieval: Enables the backend to deliver 3D models, thus maintaining up-to-date and precise representations.

The system created not only educates the user on assembly skills but also evaluates their performance and provides them with constructive feedback on how to optimize learning outcomes. This aligns with contemporary trends in immersive learning, as seen in recent studies on the subject.

# Table of Contents

# List of Figures

# List of Abbreviations

| | |
|---|---|
| **AR** | Augmented Reality |
| **VR** | Virtual Reality |
| **UI** | User Interface |
| **XR** | Extended Reality |
| **MR** | Mixed Reality |
| **SDK** | Software Development Kit |
| **REST** | Representational State Transfer |
| **API** | Application Program Interface |
| **ViT** | Vision Transformer |
| **IoU** | Intersection over union |
| **RLE** | Run Length Encoded |
| **LPIPS** | Learned Perceptual Image Patch Similarity |
| **SSIM** | Structural Similarity Index Measure |
| **ASP.NET** | Active Server Pages .NET |
| **DI** | Dependency Injection |

| | |
|---|---|
| **IoC** | Inversion of Control |
| **DTO** | Data Transfer Object |
| **SQL** | Structured Query Language |
| **EF Core** | Entity Framework Core |
| **MQDH** | Meta Quest Developer Hub |
| **FPS** | Frames Per Second |
| **GPU** | Graphics Processing Unit |
| **OVR** | Oculus Virtual Reality (prefix used in Meta SDK components, e.g., OVRCameraRig) |
| **TConv** | Transpose Convolution |
| **HTTP** | HyperText Transfer Protocol |
| **HTTPS** | HyperText Transfer Protocol Secure |
| **SSL/TLS** | Secure Sockets Layer / Transport Layer Security |
| **AI** | Artificial Intelligence |
| **SUS** | System Usability Scale |
| **UEQ** | User Experience Questionnaire |

# Chapter 1
# Introduction

## 1.1 Motivation

With the modern context of large-scale production and rapidly evolving industrial frameworks, the demand for effective and versatile personnel is accelerating at an unprecedented rate. Conventional methods of teaching, depending as they do to a large degree on physical equipment and demonstrations, are no longer sustainable.

Although these methods may have been effective in the past, they require huge financial investments, dedicated time slots for training, and supervision. Noticeably, such conditions tend to subject learners to risky situations before they have developed the necessary skills, heightening the chances of accidents and human error. In fast-paced conditions such as factories, workshops, or assembly lines, even a slight error could result in damage to equipment, delay in production, or severe injury.

Furthermore, manual training is ineffective because it cannot be quantified with accuracy. Defects in training periods lead to

material wastage, equipment damage, and even injury. It is no longer possible to rely totally on physical training.

The motivation for our study comes from the apparent disconnect between growing needs for specialized skills and the constraints of existing training approaches. We observed that while there is ongoing development in industries, the process by which new employees are trained remains antiquated, expensive, and dangerous.

Using AR and VR, we can replicate real-world environments where practitioners can interact with complex tasks while offering a safe and cost-effective space. Regardless of the level of training, no training can completely eliminate human error, but by providing this arena, we can bridge the gap.

## 1.2 Problem Definition

With the current dynamic environment of skill acquisition and experiential learning, initiating students to hands-on, on-the-job work continues to be an uphill battle. Within the classroom

environment, the workshop, or technical training at the introductory level, conventional methods of learning—i.e., training using live equipment under direct supervision—have intrinsic deficiencies. Such techniques not only waste time and funds but even risk danger for beginners who are still learning elementary facts to perform things like that with confidence.

Among numerous issues, the following are the most important issues:

- **Risks to Safety:** Students are often presented with new or complicated equipment before they are adequately ready, raising the risk of injury, breakage, or frustration, particularly in contact settings.
- **Limited Flexibility:** These conventional learning processes are less flexible in terms of adjusting to fit the individual learner's learning style or pace. They also don't provide personalized feedback and progress monitoring, thereby dampening the level of learner improvement and confidence.

- **Slower Skill Acquisition:** Rigid and one-size-fits-all approaches will slow down onboarding and lead to uneven skill acquisition. This can translate to increased errors, slower learning outcomes, and decreased engagement.
- **Scalability Issues:** As learning contexts increase—whether in classrooms, organizations, or training programs—conventional approaches can't keep pace. Increasing access to experiential learning generally involves replicating resources, and that is not viable in the long term.

Whereas most current AR/VR platforms try to solve some of the above challenges, they don't quite work in practice. Existing solutions are either too specialized, minimally interactive, or too rigid for industrial use cases, making them less useful in general-purpose skill-acquisition settings. Additionally, their non-realism and rigidity diminish their usefulness for that subset of students who gain from contextual, immersive practice.

Our product, The Learning Lens, shatters these limitations with a mixed reality platform that simulates experiential learning in a

virtual environment that is secure, interactive, and scalable. Leveraging interactive assembly-based activities as a foundation model, the system enables students to interact with realistic simulations, receive immediate instruction, and monitor individual progress regardless of age and aptitude. It is not focused on factory simulation, but on creating a flexible and adaptive environment that will make it easier, more quantifiable, and safer to learn practical tasks. By doing this, The Learning Lens aims to shatter the confines of conventional training and unleash the possibilities for more modern, inclusive, and effective skills learning.

## 1.3 Objective

The ultimate objective of this project is to transform hands-on learning by developing an adaptable and immersive mixed reality platform for simulating real-world tasks in a safe, interactive environment. While the system is currently focused on assembly-based tasks because of their simplicity and ease of visualization, The Learning Lens is conceived to have general

applicability to skill development use cases—from academic and introductory technical training to task simulation of a more general-purpose nature. The framework aims to enhance the efficacy, safety, and scalability of experience-based learning activities in diverse settings.

The project has the following specific objectives:

- Creating and designing an interactive training system with augmented reality (AR) through Lego-based simulation and virtual reality (VR) with advanced 3D models to deliver differentiated and dynamic learning experiences.
- To provide an interesting user interface whereby students can create profiles, select lessons of varying degrees of difficulty, receive immediate feedback, and monitor their own progress over time.
- To include embedded assessment functions that measure skill acquisition in training, monitoring accuracy, productivity, and task completion to give short, measurable performance information.
- To create general-purpose flexibility by building a modular system that would be extensible after assembly to support a wide range of domains, students, and training requirements.
- To minimize risk, cost, and reliance on physical hardware by transferring early-stage learning into the virtual

environment, thus enhancing safety and accessibility without any loss of realism.

- To create a platform for education that is both future-proof and scalable, which can expand to support new forms of content, sophisticated feedback mechanisms, and growing user bases to address changing education and technology requirements.

With these goals, The Learning Lens seeks to bridge the gap between theoretical training and real-world application—a smarter, safer, more interactive approach to skill acquisition in the era of accelerating change.

## 1.4 Document Organization

**Chapter 2 – Literature Review –** provides a comprehensive overview of the technological and academic background relevant to the project. It begins by highlighting the rapid growth and cross-disciplinary impact of VR and AR technologies, particularly in educational and industrial training contexts. The chapter then

reviews several related studies that support the viability of using immersive technologies for skill acquisition and procedural training, especially in the automotive industry. A focus is placed on how AR and VR enhance engagement, enable real-time feedback, and support adaptive learning environments. Furthermore, the chapter discusses evaluation methods used to assess built 3D structures, including voxel-based 3D reconstruction and image similarity techniques, and justifies the selection of specific models and metrics (e.g., IoU, LPIPS) for assessing user performance in the AR/MR environment.

**Chapter 3 – Analysis and Design –** outlines the overall system architecture and design of the VR/AR training application. It presents a three-tier architecture consisting of a Unity-based presentation layer, an ASP.NET backend, and a SQL Server database. The chapter details the system's functional and nonfunctional requirements, target users, and design artifacts including use case, class, sequence, and database diagrams. It also explains how various modules collaborate to deliver interactive

training, performance tracking, and personalized feedback for both VR and AR learning modes.

**Chapter 4 – Implementation** – describes the core functions, techniques, and algorithms implemented in both the Unity-based frontend and ASP.NET Core backend of the VR/AR training system. It covers the development of immersive interfaces, interaction mechanics, progress tracking, and AR-specific features like image capture and build assessment. On the backend, it explains the modular architecture, RESTful APIs, authentication, and the use of Entity Framework. Finally, the chapter evaluates two AR assessment approaches—3D reconstruction using Pix2Vox variants (which failed to meet reliability needs) and image similarity using LPIPS, which proved effective for detecting perceptual differences and was ultimately adopted.

**Chapter 5 – User Manual** – provides a step-by-step installation and operation guide for the VR and AR applications of the educational system on Meta Quest devices. It covers how to sideload the APKs using MQDH or SideQuest, outlines the

minimum hardware requirements, and explains how to navigate and interact within each app. Users learn how to launch lessons, assemble objects in VR using controllers, and capture images in AR for automated assessment using gestures or buttons.

**Chapter 6 – Conclusions and Future work –** presents the final conclusions of the project and outlines potential areas for future development. It summarizes the key achievements of the system—such as immersive training, real-time assessment, and modular architecture—and reflects on the results obtained through the implementation. The chapter also proposes enhancements like introducing a trainer role, creating a web portal, developing a Unity SDK for content creation, improving lesson discoverability, and adopting more advanced AR assessment techniques to further elevate the system's impact and usability.

# Chapter 2
# Literature Review

## 2.1 Field of the Project/General Overview

VR and AR markets have been growing rapidly ,and noteworthy research and changes are being made in both technologies all over the world. Thus VR and AR has made a great impact in various fields ranging from educational fields up to Industrial and medical fields. Moreover, those technologies have shown strong capabilities where VR offers a strong immersive environment and on the other hand, AR which overlays virtual elements into the real world.

VR has enhanced engagement for kids in the educational sector, providing a fully fun and interactive experience. In addition, it has provided a mesmeric experience in both industrial and medical fields, boosting focus on the task at hand.

On the other side, AR has offered real-time, real-life training on machines accompanied by different assistance methods through panels like written instructions, voice assistance, and tutorial videos.

Similarly, in the automotive industry which we have targeted in our project. VR/AR technologies have allowed practitioners to perform assembly, repair, and diagnostic tasks in a safer environment and a more efficient approach. As a result, it has reduced training time, resources and cost.

Unity engine is one of the most popular engines that are used in creating those VR/AR experiences. We have used it to develop a software that focuses on performance enhancement and evaluation to those who are interested in the automotive industry to learn in a fun, interactive, educational environment and improve their skills.

## 2.2 Related Studies

### 2.2.1 VR/AR Studies

#### *1. Mobile devices within manufacturing environments: a BMW applicability study (2012)*

The paper examines BMW's application of Augmented Reality (AR) technology in training for car assembly. The paper identifies advantages of AR in training, specifically that it has the capability of projecting digital instructions onto physical parts so that employees are able to engage with the assembly process in real time. By delivering hands-free, on-the-job instructions, AR enhances the efficiency and accuracy of assembly operations, decreases training time, and enhances the learning experience. This method has a direct correlation to our project, which also makes use of AR to offer interactive assembly instructions for motorcycles. The AR system in this project is intended to lead novice mechanics through every stage of the assembly process, giving feedback in real time and deepening their comprehension

of component relationships. The outcomes of BMW's adaptation of AR in assembly training are a significant source of informing our system so that it's not only feasible but also viable to augment the learning process.

### *2. Research on the Application of "AR/VR+" Traditional Cultural Education Based on Artificial Intelligence (2021)*

This paper discusses the application of VR and AR in cultural education in the way it improves the engagement of students with traditional art and cultural heritage. It explains how it is essential to have immersive learning experiences in order to facilitate users' better comprehension of difficult topics through interaction.

### *3. Curriculum System of Preschool Education under the Background of AR Intelligence (2021)*

This study is about using AR in preschool education to make learning engaging for students. The article outlines how AR technologies, if coupled with AI, create adaptive learning environments that are able to react to the needs of individual

students. The ability of AR to provide context-aware information is in line with the goals of our project, which is to implement AR in providing personalized learning experiences in motorcycle assembly training.

### *4. Augmented Reality and Digital Twin for Mineral in Industry (2023)*

This paper discusses the convergence of AR and digital twin technology in the mineral industry. More specifically, it describes how AR systems provide real-time, interactive training through overlaying digital information onto real-world objects. This process allows users to interact with the environment as they learn about complex systems. Similarly, in the automotive world, AR can be used to overlay assembly instructions on motorcycle components, guiding mechanics and providing instant feedback.

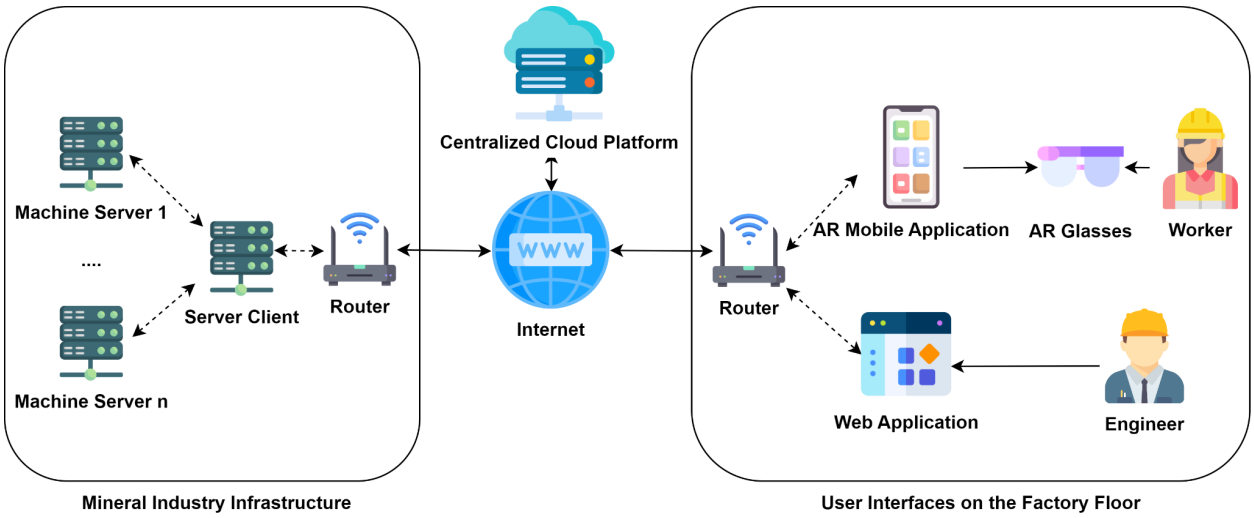



Figure 2.1 – General Architecture

The architecture disclosed herein is a multi-component system to deliver real-time monitoring and control over industrial processes. The architecture includes several layers of communication among physical devices (machines, servers) and computer systems to enable error-free data transfer from the factory floor to the cloud platform. The most essential component of this arrangement is the central position of a centralized cloud platform. The system is composed of machine servers which collect data and send it to a server client through routers. The data

is sent via the internet to the centralized cloud platform, where it is available for various user interfaces. These include AR mobile applications and AR glasses operated by workers, and web applications which can be operated by engineers for additional analysis and monitoring.

This architecture shows how real-time feedback is achieved through the use of augmented reality (AR), which is a key requirement for the digital twin system. In our project, the same principle is used, but with a different simplified architecture where we depended only on our app on the AR glasses which is used to overlay assembly instructions over physical components to enable more interactive and interesting learning. Our motorcycle assembly training system can adapt to this model, where data will be utilized to train trainees and their progress tracked in real time.

### *5.An Interactive Augmented Reality Framework to Enhance the User Experience and Operational Skills in Electronics Laboratories (2024)*

The paper explains the employment of Augmented Reality (AR) in schools to enhance user experience and procedural proficiency in electronic labs. The research indicates how AR-based systems can provide interactive learning environments, enhanced understanding of sophisticated tools, and procedures. This comes into focus in relation to our project, which employs AR to make the users learn how to assemble motorbikes. Results of this study validate our method of incorporating AR to offer real-time, contextual learning advice for students such that they enhance their hands-on ability and efficiency of operations in the assembly process. We plan to use the same AR method to offer an equally interactive and efficient training alternative for new motorcycle technicians using AR to make learning easier, intuitive, and hands-on without the use of actual parts.

### *6. The Significance of each paper for our project*

| Paper Title | Relevance to our project | Key contributions to our project |
|---|---|---|
| Mobile devices within manufacturing environments: a BMW applicability study | BMW explored using mobile devices in manufacturing especially for assembly training. In addition, they evaluated Ultra Mobile PCs (UMPC) and Augmented Reality (AR) for hands-free operation in assembly and found out that mobile devices improve communication and efficiency in manufacturing. | Hand-free training using mobile devices is applicable to our project of augmented reality for assembly instructions for motorcycles. Applying augmented reality to train hands-free aligns with our goals of skill and efficiency development in our virtual and augmented reality motorcycle assembly system. |
| Research on the Application of "AR/VR+" Traditional Cultural Education Based on Artificial Intelligence | Highlights how immersive AR/VR environments enhance engagement and make complex or abstract learning more tangible and interactive. | Supported our approach of using VR to simulate assembly tasks in a risk-free environment, providing learners with a virtual space to explore, practice, and receive guidance without needing physical resources. |
| Curriculum System of Preschool Education under the Background of AR Intelligence | Shows how AR can adapt content to user interaction, especially in educational contexts, promoting personalized learning experiences. | Inspired our use of AR to adapt the training flow based on user progress. Offering a more engaging, learner-centered experience in motorcycle |

| | | |
|---|---|---|
| | | assembly. |
| **Paper Title** | **Relevance to our project** | **Key contributions to our project** |
| Augmented Reality and Digital Twin for Mineral Industry | Demonstrates the use of AR in overlaying digital instructions on physical equipment for real-time, interactive training in industrial settings. | Adopted the concept of real-time feedback and instructional overlays for motorcycle assembly, and included performance monitoring through AR |
| An Interactive Augmented Reality Framework to Enhance the User Experience and Operational Skills in Electronics Laboratories | Discusses the impact of AR in enhancing learning in electronics laboratories, relevant to our project as we aim to provide an interactive environment for AR training in automotive assembly. | Inspired the use of AR for interactive learning in mechanical tasks, aiming to improve operational skills in motorcycle assembly. |

| Paper Title | Scores/Metrics |
| --- | --- |
| Mobile devices within manufacturing environments: a BMW applicability study | **Not Applicable**: This study primarily investigates the applicability of mobile devices in manufacturing environments and does not present empirical data or specific quantitative usability/user experience metrics such as SUS or UEQ scores. |
| Research on the Application of "AR/VR+" Traditional Cultural Education Based on Artificial Intelligence | **Not Applicable**: This paper is a literature review focusing on the theoretical applications of AR/VR in traditional cultural education and does not report empirical study findings with user experience or usability metrics like SUS or UEQ scores. |
| Curriculum System of Preschool Education under the Background of AR Intelligence | **Student Satisfaction Data**: (Scores on an unspecified scale; higher values indicate greater satisfaction.)<br>- VR training methods: 84, 82, 80, 79, 83, 86, 84, 87, 82, 81<br>- Traditional training methods: 78, 77, 75, 76, 80, 80, 81, 81, 77, 78<br>**Note**: While student satisfaction data is provided, the paper does not explicitly report scores from standardized |

| | |
|---|---|
| | usability or user experience questionnaires such as SUS or UEQ. |
| Augmented Reality and Digital Twin for Mineral Industry | **System Usability Scale (SUS)**:<br>- Average Score: 80.68 (categorized as "Excellent")<br>- Standard Deviation: 11.03<br>- Range: 50 (lowest) to 90 (highest)<br>**Task Completion Ease**: Evaluated as "very easy to perform" by most participants.<br>**User Interest in AR Technology Implementation**: 50% "Moderately Interested", 50% "Very Interested".<br>**Willingness to Use Prototype in Work**: 83.3% of industrial participants expressed willingness to use the solution. |
| An Interactive Augmented Reality Framework to Enhance the User Experience and Operational Skills in Electronics Laboratories | **System Usability Scale (SUS)**: 80.9 (categorized as "Good")<br>**User Experience Questionnaire (UEQ)**:<br>- Attractiveness: Mean 1.61, SD 1.15 (Good)<br>- Perspicuity: Mean 1.94, SD 0.71 (Good)<br>- Efficiency: Mean 1.46, SD 1.08 (Above Average)<br>- Dependability: Mean 1.74, SD |

| | |
|---|---|
| | 0.63 (Excellent)<br>- Stimulation: Mean 1.97, SD 0.68 (Excellent)<br>- Novelty: Mean 1.60, SD 1.00 (Good) |

### 2.2.2 Evaluation methods studies

***3D models/built structures evaluation Related Studies:***

One of the proposed methods for the evaluation of a physical 3D structure was to perform 3D reconstruction on the built structure, and then compare it to a reference 3D model via IoU score which is the same method used for evaluating a 3D reconstruction deep learning model. This is relevant to our project, mainly in the evaluation of the structures built by the user in the AR/MR mode. The following are some of the studies conducted on the topic of 3D reconstruction -especially multi-view 3D reconstruction- since the nature of our project requires an accurate representation of the built structure, to precisely evaluate it, without depending on the "guessing the unseen parts" of the single-view 3D reconstruction models. The idea can't be achieved without a

model that has a perfectIoU score, overwise the reconstructed model won't accurately represent the structure built by the user. Additionally, Since inference time was crucial in our project, only studies that had a reasonable inference time were considered in our project.

### *1. Pix2Vox: Context-aware 3D Reconstruction from Single and Multi-view Images (2019)*

This study proposed a deep learning model for multi-view 3D reconstruction (see Figure 1) that could, according to the authors, reconstruct unseen categories that it had not been trained on, which made it a good fit for our project. It takes up to 20 2D images captured from different angles of an object as input. It then sequentially passes them through a backbone network -VGG16 in this case- that acts as an **Encoder** that outputs a set of high-level feature maps per image. These features are then sequentially passed through a **Decoder** that reshapes and decodes the feature maps of each input image into an independent 3D model, so it acts as a single-view reconstructor until this stage.

The output set of 3D models are then fused together using a context-aware fusion module named a **Merger** and this is where the multi-view reconstruction step takes place. The merged 3D volumes are finally passed to a module called the **Refiner** that, according to Xie et al (2019), "corrects the wrongly recovered parts of the fused 3D volumes" obtained from the **Merger** module.

Figure 2 shows the full architecture of Pix2Vox.

This study is important because, first, 3D reconstruction itself is needed in the AR/MR mode. second, because it showed how to effectively compute the IoU score for a binarized voxel (see Eq. 1), third, the study is a bit outdated (published in 2019) and the model had a mean IoU score of (0.706)so there was a room for improving the outcomes using more recent backbones and other methods that'll be shown later. Finally, the model had a very reasonable inference time (~55.5 ms for 8-views reconstruction).

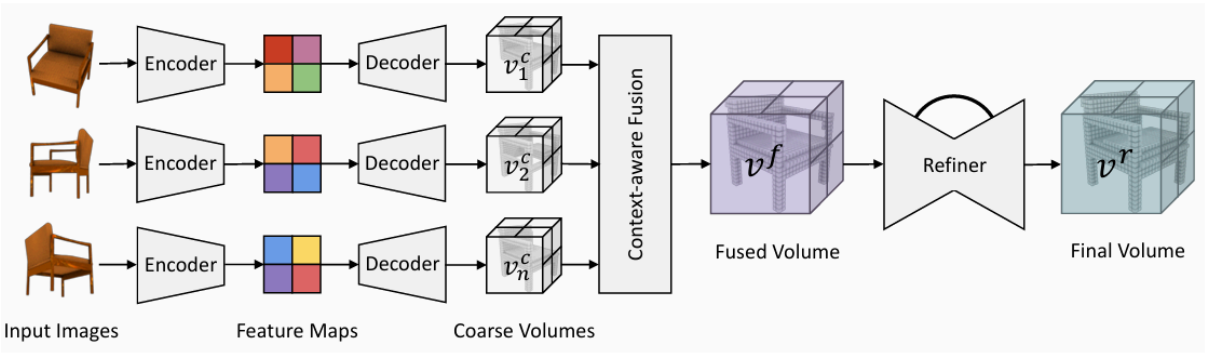



Figure 2.2 – Pix2Vox overview

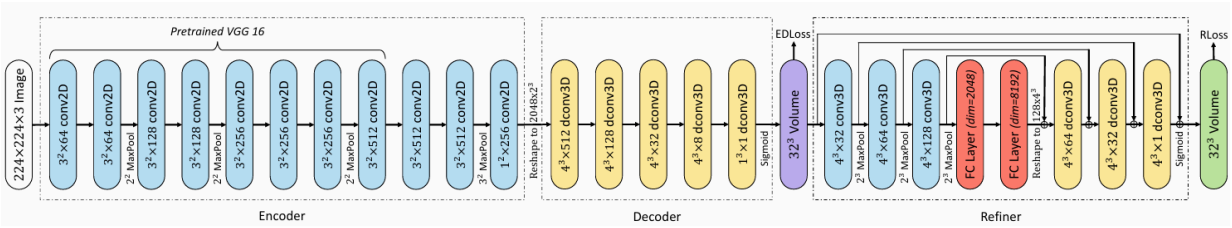



Figure 2.3 – Pix2Vox Architecture

$$IoU = \frac{\sum_{i,j,k} I(p_{(i,j,k)} > t) \cdot I(gt_{(i,j,k)})}{\sum_{i,j,k} I[I(p_{(i,j,k)} > t) + I(gt_{(i,j,k)})]}$$

Equation 2.1 – t is a binarizing threshold, I is an indicator function, $p_{(i,jk)}$ is the probability of having a voxel in position $i, j, k$ , and gt is the ground truth

## 2. *Pix2Vox++: Multi-scale Context-aware 3D Object Reconstruction from Single and Multiple Images (2020)*

Pix2Vox++ introduced a significant performance improvement over the original Pix2Vox model. The first significant difference was the use of Resnet 18/50 as a backbone; it also introduced some changes to the mechanism of the context-aware fusion module. These changes contributed to a better overall IoU score (0.843 for 8-views reconstruction on shapenet). Figure 2.4 shows the Pix2Vox++ model architecture.

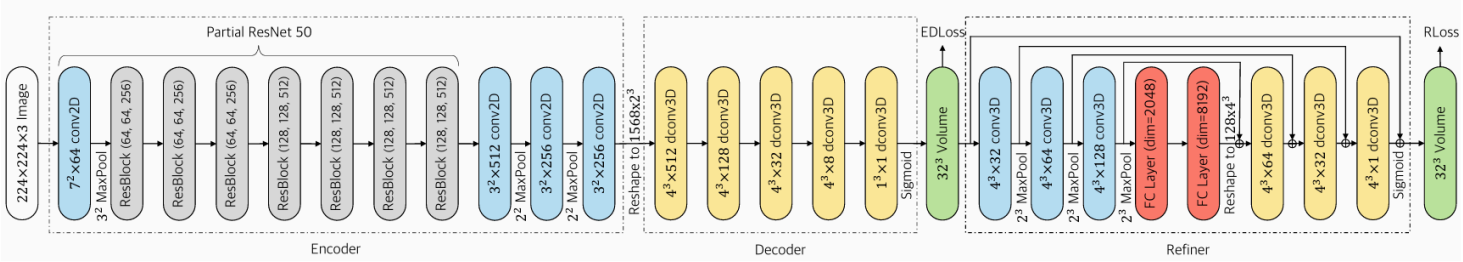


Figure 2.4 – Pix2Vox++ architecture

### *3. SnakeVoxFormer: Transformer-based Single Image Voxel Reconstruction with Run Length Encoding*

This study introduced a transformer-based architecture (Check Figure 4) for reconstructing 3D voxel models from a single or multiple 2D images. The key novelty of the model lies in how it compresses the voxel space using Run Length Encoding (RLE), which linearly traverses the voxel grid in a “snake-like” pattern and encodes it into a 1D sequence. These compressed sequences are then tokenized and used as input to a transformer decoder that reconstructs the 3D voxel structure.

The encoder used a Vision Transformer (ViT) pretrained on ImageNet to extract high-level features from the input images. These features were then decoded into tokens that map to a dictionary of Run Length Encoded sequences, which are then decompressed back into 3D voxels. This method allowed the

authors to reduce the input data size to about 1% of its original volume, while still achieving an IoU score of 0.933 on 20-views reconstruction, which made it the State-Of-The-Art of its time.

Despite its relevance to our project—especially due to its voxel compression efficiency and high IoU score—this study was basically a dead end. As the authors claimed that they would release the source code, but it was never published, which made it impossible to build upon, especially since that it was unclear on how to exactly implement the RLE technique.

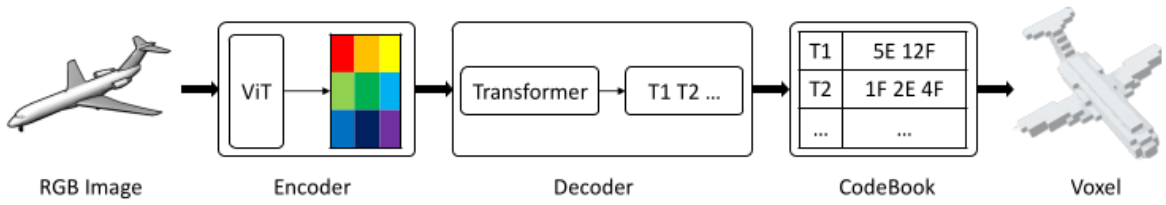



Figure 2.5 – SnakeVoxFormer Model Architecture

***Image Similarity based evaluation Studies:***

An alternative method to evaluate the structures built by the user is to capture images from different angles of the built structures, and compare it via image similarity metrics to reference images of the corresponding angles.

The following are studies related to image similarity techniques that were tested for our project.

### *4. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric (LPIPS)*

LPIPS is a perceptual similarity metric that measures how close two images *feel* to humans by comparing deep features from a pretrained network (e.g., VGG or AlexNet). Instead of relying on pixel-wise differences, it uses high-level features calibrated on a massive human-labeled dataset to match human perception. The advantage LPIPS has over SSIM is that slight shifts or transitions between the images won't destroy the similarity score — if the two images are actually similar, LPIPS will say so.

### *5. SSIM: Structural Similarity Index (2004)*

The Structural Similarity Index (SSIM) is a classic image similarity metric widely used to measure perceptual quality. It compares two images based on local luminance, contrast, and structure patterns. Despite its popularity, SSIM has a critical flaw for our project: it fails badly if there is even a slight spatial shift or misalignment between the compared images. Since in our AR/MR task, the object being evaluated might be slightly shifted compared to the reference, SSIM would report poor similarity even when the structures are visually identical. Therefore, SSIM is unsuitable as a reliable evaluation metric for our application unless the 2 images identically match each other, which is almost impossible to achieve.

***The Significance of each paper for our project***

| Study | Significance to our project |
|---|---|
| Pix2Vox (2019) | - Used for 3D reconstruction in AR/MR mode to evaluate built structures.<br>- Demonstrated how to compute IoU for binarized voxels (see Eq. 1).<br>- Despite being outdated, it provides a |

| | |
|---|---|
| | foundation and benchmark (IoU = 0.706) that can be improved with newer methods.<br>- Offered reasonable inference time (~55.5 ms for 8-view reconstruction), making it practical for real-time use. |
| Pix2Vox++ (2020) | - Improved over Pix2Vox using ResNet backbone and enhanced fusion module.<br>- Delivered significantly better IoU score (0.843), making it more accurate for evaluating AR-built structures.<br>- A strong candidate for replacing Pix2Vox due to better accuracy and architecture. |
| SnakeVoxFormer | - Achieved state-of-the-art IoU (0.933) and excellent compression using Run Length Encoding (ideal for performance-critical tasks like ours).<br>- Had great potential for fast and accurate reconstruction.<br>- Ultimately not usable due to unavailable source code and unclear implementation details. |
| LPIPS | - Provides image similarity evaluation that aligns with human perception.<br>- Tolerant to slight shifts between user-built and reference images, making it ideal for AR evaluation tasks.<br>- More reliable than SSIM when perfect alignment between images is not guaranteed. |
| SSIM | - Considered as an image similarity method but found unsuitable. |

| | |
|---|---|
| | - Fails under slight spatial shifts, which are common in AR/MR tasks.<br>- Rejected due to its inability to handle minor misalignments between user and reference images. |

# Chapter 3
# Analysis and Design

# 3.1 System Overview

## 3.1.1 System Architecture

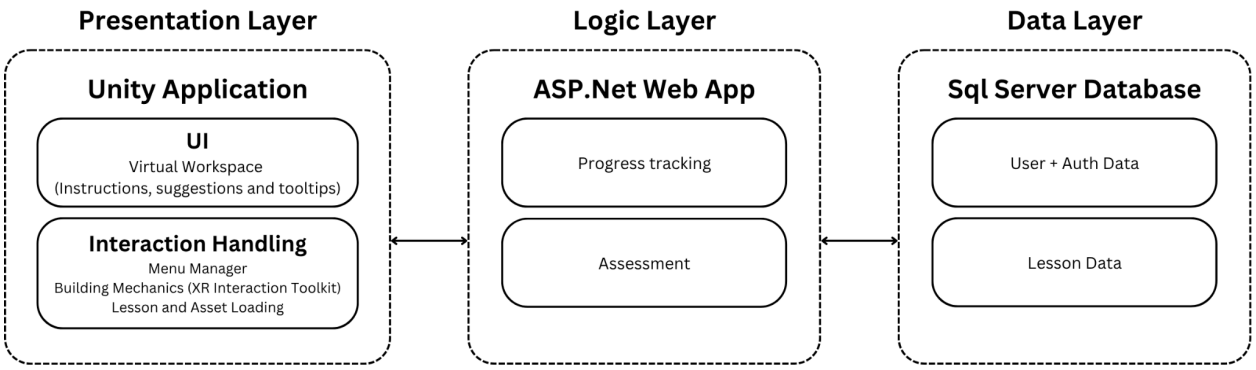



Figure 3.1 – System Architecture

Shown above, is the three-tier architecture of the system. Where the presentation layer is the Unity application that the user will install and train through. There are two Unity applications: one for accessing VR lessons, and another for accessing AR lessons. This layer consists of the User Interface (UI), everything the user sees, as well as the logic used to handle interaction with the UI. Such as their virtual workspace the user works in when training in VR app, the instructions and information shown to a user during training. The system handles the user's input using different

modules such as a menu manager, Unity's XR Interaction toolkit, lesson loader, and an asset loader. The presentation layer interacts with the logic layer through a RESTful API.

The logic layer is an ASP.NET web application that manages users, tracks their progress, and assesses AR Lessons. The presentation layer sends requests to the logic layer either to load data, record data, or to assess an AR Lesson. Available to the logic layer is a python script that can assess the work done in an AR lesson. The logic layer is directly connected to the data layer which holds important data and assets for both users and lessons.

Ideally, the presentation layer would run on a client device that the user has access to, communicating with a server where the logic layer and the data layer would run.

### 3.1.2 Functional Requirements

***A. User Account and Access***

- The system must allow a user to sign up as a new user.
- The system must allow the user to log in with their credentials.
- The system must allow the user to continue as a guest.

***B. VR Lesson Management and Interaction***

- The system must allow the user to browse the lessons available in VR.
- The system should allow the user to download a VR lesson locally to their device.
- The system must allow the user to start an available VR lesson in either “Learning mode” or “Test mode”.
- The system must provide the user with a reference model in the virtual building space.
- The system must allow the user to attach the pieces together to assemble the model.

- In the learning mode of a VR lesson, the system must show the user a list of the pieces that are yet to be assembled.
- In the learning mode of a VR lesson, the system must show the user information about the piece in his hand.
- In the learning mode of a VR lesson, the system must show the user an indicator of the progress made so far in terms of percentage.
- In the learning mode of a VR lesson, the system should provide the user with negative feedback when they attach a piece incorrectly.

***C. AR Lesson Management and Interaction***

- The system must allow the user to browse the lessons available in AR.
- The system should inform the user of the building set/materials required to assemble or build the final item.
- The system must allow the user to start an AR lesson in either “Learning mode” or “Test mode”.
- In the learning mode of an AR Lesson, the system must provide the user with guidance (based on the original input,

this point was incomplete, so I've completed it to reflect a common function in AR learning).

***D. Performance Tracking***

1. The system should keep record of the user's performance across all attempts of each lesson (VR or AR), and allow the user access to an overview of his performance across those attempts.

### 3.1.3 Nonfunctional Requirements

***A. Performance***

1. **Response Time:** The system should be able to deliver real-time response with a lag of not more than 1-2 seconds between user input and system reaction. This is very important for interactive learning systems.
2. **Frame Rate:** The VR experience should be a minimum of 60 frames per second (FPS) to enable seamless and immersive interaction. Anything less would cause the user discomfort and dizziness.
3. **System Scalability:** The system must be scalable to accommodate an increasing number of users and assembly

tasks, in such a manner that it can process more than one session at a time without degradation in performance.

### *B. Usability*

1. **User Interface:** The user interface of the system must be user-friendly and easy to navigate for both novice and experienced users. It must possess intuitive visual cues and easy navigation among AR/VR environments.

2. **User Experience:** The AR/VR interactions must be intuitive to learn and must not need previous experience with the technology. Usability testing must demonstrate a 95% user satisfaction rate with the interface and interaction techniques.

3. **Multilingual Support:** The system may be multilingual, with English and Arabic being the first two languages supported, in order to meet the requirements of users from diverse locales.

### *C. Reliability and Availability*

1. **System Availability:** The system should be available 99.5% of the time, apart from planned maintenance. This guarantees little downtime and availability for training sessions.

2. **Fault Tolerance:** Upon failure, the system should give comprehensive error messages and permit users to continue training without loss of data. Critical errors should be logged for future investigation.

3. **Data Integrity:** The system must not lose any user data (progress, results, feedback) during training sessions. All data must be backed up at regular intervals and kept securely.

### *D. Security*

1. **Data Privacy:** Personal user data, such as training progress and user credentials, must be encrypted and adhere to applicable data privacy laws (e.g., GDPR, CCPA).

2. **Authentication:** Authenticate users using a secure login mechanism, with role-based access for admins and trainees. The system should prevent unauthorized users from accessing confidential information.

3. **Secure Communication:** Any communication between front-end (VR/AR) and back-end systems should be encrypted using industry-standard encryption methods (e.g., HTTPS, SSL/TLS).

### *E. Maintainability*

1. **Code Modularity:** The system must be based on modular code that is easy to update and to which new features can be easily added (e.g., new training modules or task simulations).

2. **Documentation:** The system must be well-documented, such as a developer's guide, API documentation, and user manual. This makes maintenance and updates easier.

3. **Bug Fixes:** The system must have a mechanism for updating and fixing bugs on an ongoing basis, with a

turnaround of not more than 2 weeks for high-priority defects.

### *F. Compatibility*

1. **Device Compatibility:** The system must be compatible with well-known VR headsets (e.g., Meta Quest, HTC Vive), along with AR glasses. It must also have mobile AR (iOS/Android) support for users who want smartphone-based interactions.

### *G. Accessibility*

1. **Accessibility Features:** The application may be accessible for individuals with disabilities. It should have visual or auditory aid for disabled individuals, such as dynamic font sizes, screen readers, and text-to-speech capability.

2. **VR Comfort:** The VR experience must reduce motion sickness or discomfort, with adjustable movement speed settings, comfort settings, and user comfort preference settings.

### *H. Legal and Regulatory Compliance*

1. **Data Protection Compliance:** The system will need to adhere to data protection laws like GDPR for users in the EU or other applicable laws for users in specific countries.

2. **Content Licensing:** Any digital content (3D models, AR/VR assets) utilized in the training system shall be duly licensed or qualify for fair use.

## 3.1.4 System Users

### *A. Intended Users*

1. **Trainees (Beginners/Novices):**

   **Purpose**: The system is designed for users who lack experience in the assembly process and want to train in a safe, virtual environment. These are students in different industries, ranging from automotive to schools. The system supports practice of assembly tasks such as **motorcycle parts**, although the platform's modularity makes it easily adaptable to training in any other field as well, e.g., **Lego**

**construction** for children or more complex industrial assembly operations.

**Usage**: Trainees will use the system with VR headsets or AR glasses, whichever is suitable for the environment, and will be displayed step-by-step guides on how to assemble parts. They will also be able to see their progress in real time and get feedback on their work.

2. **Experienced Workers/Professionals:**

   **Purpose**: The system is also aimed at professional workers who might need to rehearse more complicated tasks or enhance their assembly method in various industries. While the project presently encompasses **motorcycle assembly**, the platform might be modified to offer training in other areas, for example, **engineering, robotics, or toy assembly**.

   **Use**: They will use the platform to learn advanced assembly methods or new tools and methods in their profession. The real-time feedback will allow professionals to refine their skills and validate their expertise in various situations.

3. **Instructors/Trainers:**

**Purpose**: The system will be utilized by trainers to instruct and monitor trainees as they progress through assembly exercises. Instructors are able to modify the training material for various domains, ranging from educational play (e.g., kids' Lego assembly) to technical domains like car manufacturing.

**Usage**: Trainers will check trainee performance, monitor progress, and offer further support according to data gathered from the system. They will also be able to introduce new modules or activities for training.

**4. System Administrators/Developers:**

**Purpose**: The backend will be handled by administrators and developers who will make sure the system functions properly. They will have the duty of user accounts, storage of data, and permission to access.

**Use**: The admins and developers will be working behind the scenes to keep the system running and updated. They will also include updates on training modules, the system's performance optimization, and bug fixing.

### *B. User Characteristics*

1. **Trainees (Beginners/Novices):**

   **Experience/Skills**: No prior experience working with assembly procedures or AR/VR systems is necessary. General computer proficiency and experience working with technology will be helpful. Trainees will have the ability to pick up from user-friendly tutorials and interactive elements of the system without needing a lot of previous experience.

   **Technology Comfort**: The system will be user-friendly, with a simple interface to navigate. Trainees might or might not be familiar with VR/AR technologies, but the system will be built to support users with no familiarity with these technologies.

   **Learning Attitude**: Candidates must possess a desire to learn new concepts and be open to getting their hands dirty through practice. Whether building a motorcycle, a Lego model, or any other assembly activities, an **attitude to attempt and play around with the platform** will be necessary.

2. **Experienced Workers/Professionals:**

   **Experience/Skills**: These users must have background experience in their chosen fields, whether that is automotive assembly,

engineering, or another area. They must be knowledgeable about assembly processes and possibly related tools and equipment.

**Comfort with Technology**: They are not necessarily VR/AR experts, but they must be willing to use new technologies and be prepared to adopt new learning techniques that can enhance their assembly skills.

**Learning Attitude**: The professionals need to possess a continuous learning attitude and an open mindset for embracing new technologies since they will utilize the system to enhance their current skills and keep abreast of new practices and tools.

3. **Instructors/Trainers:**

**Experience/Skills**: Trainers need lots of experience in teaching assembly skills and leading students through technical procedures. Instructors must also be experienced in evaluating the performance of end-users and giving feedback for remediation.

**Technology Comfort**: Trainers must be comfortable with using online platforms to track and evaluate user performance. Familiarity with AR/VR platforms is not necessary but will facilitate the development and implementation of successful training.

**Learning Attitude**: Trainers must be patient and versatile, with the skills to customize training modules to various learning styles and modify content to various fields (ranging from educational play for children to technical skills for adults).

4. **System Administrators/Developers:**

   **Experience/Skills**: Developers and administrators both need to have a solid background in **backend development**, **database management**, and **AR/VR technologies**. They need to be experts in **ASP.NET Core**, **Unity 3D**, and **cloud platforms** to ensure the performance and functionality of the system.

   **Technology Comfort**: They need to be extremely comfortable with the application of backend frameworks and AR/VR technologies, along with combining different system components seamlessly.

**Learning Attitude**: The admins and developers must have a learning attitude in order to continuously enhance the system according to user feedback and technological upgrades.

# 3.2 System Analysis & Design

## 3.2.1 Use Case Diagram

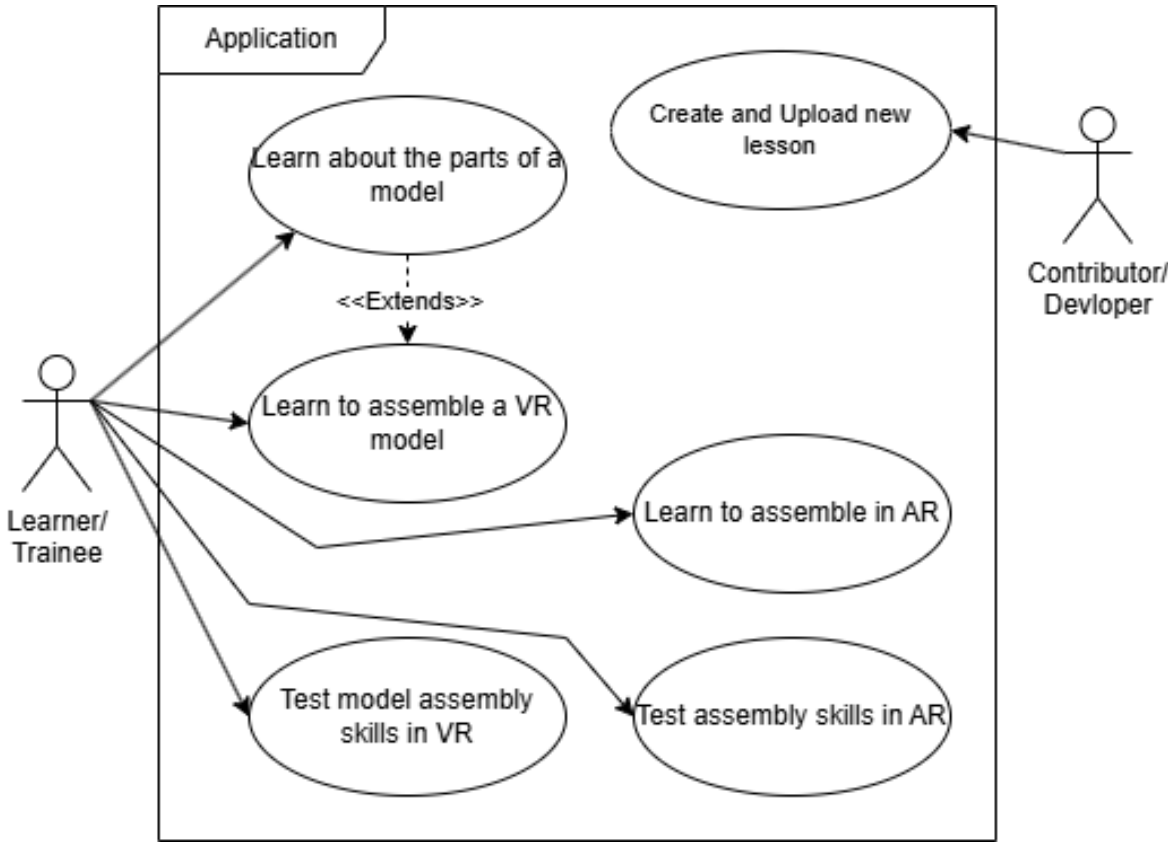



Figure 3.2 – Use Case Diagram

Use cases in our application differ depending on what the user's stage is in learning. The user would start out going through lessons in "learning mode" in order to either gain information about the assembly pieces or to just learn how to assemble them, or both. At some point the user would need to assess his ability to build the model without the hints and instructions provided by the "learning mode" and that's when he would use "test mode", to test his assembly skills. The same applies for AR, where the user can access the lesson in "learning mode" or "test mode".

Usecases would mostly differ based on the lesson the user chooses to take, and practice. Lessons can range from carpentry models, to mechanical vehicular models, to simple toy models.

## 3.2.2 Class Diagram

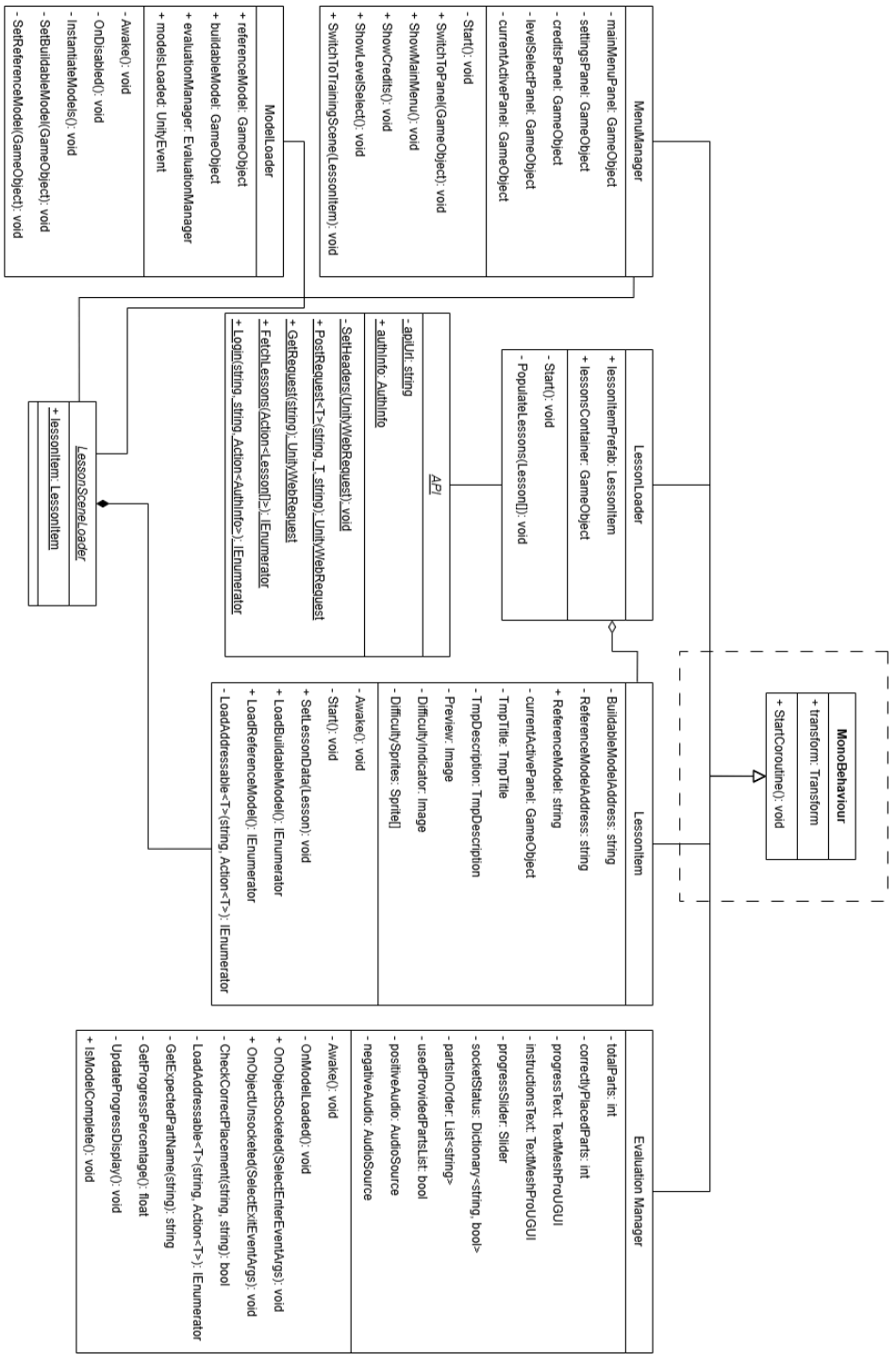



Figure 3.3 – UML class diagram

### 3.2.3 System Sequence Diagram

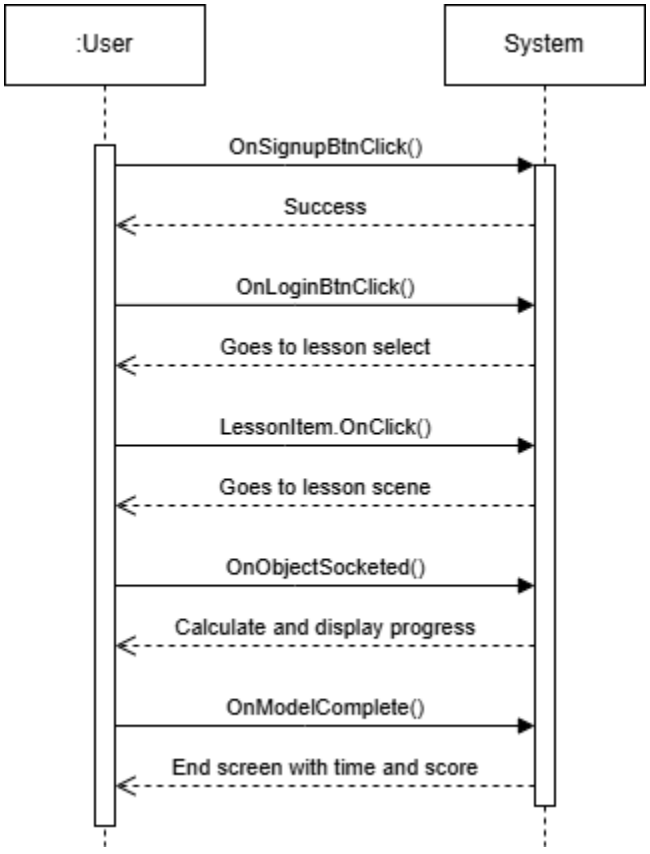



Figure 3.4 – System Sequence Diagram in VR App

The system sequence diagram in Figure 3.4 illustrates the typical interaction between a user and the VR application during a standard session, starting from account creation to completing a

lesson. It outlines the chronological flow of events as triggered by the user and handled by the system.

1. ***OnSignupBtnClick()***

The interaction begins when a new user initiates account creation by clicking the sign-up button. The system processes the request and returns a success response upon successful registration.

2. ***OnLoginBtnClick()***

After registration, the user proceeds to log in. Upon a successful login, the system navigates the user to the lesson selection interface.

3. ***LessonItem.OnClick()***

The user selects a lesson from the available list. The system responds by loading and transitioning to the corresponding VR lesson scene.

***4. OnObjectSocketed()***

During the lesson, the user performs actions—such as assembling components—triggering an event when an object is successfully "socketed" into place. The system then calculates progress based on the user's actions.

***5. OnModelComplete()***

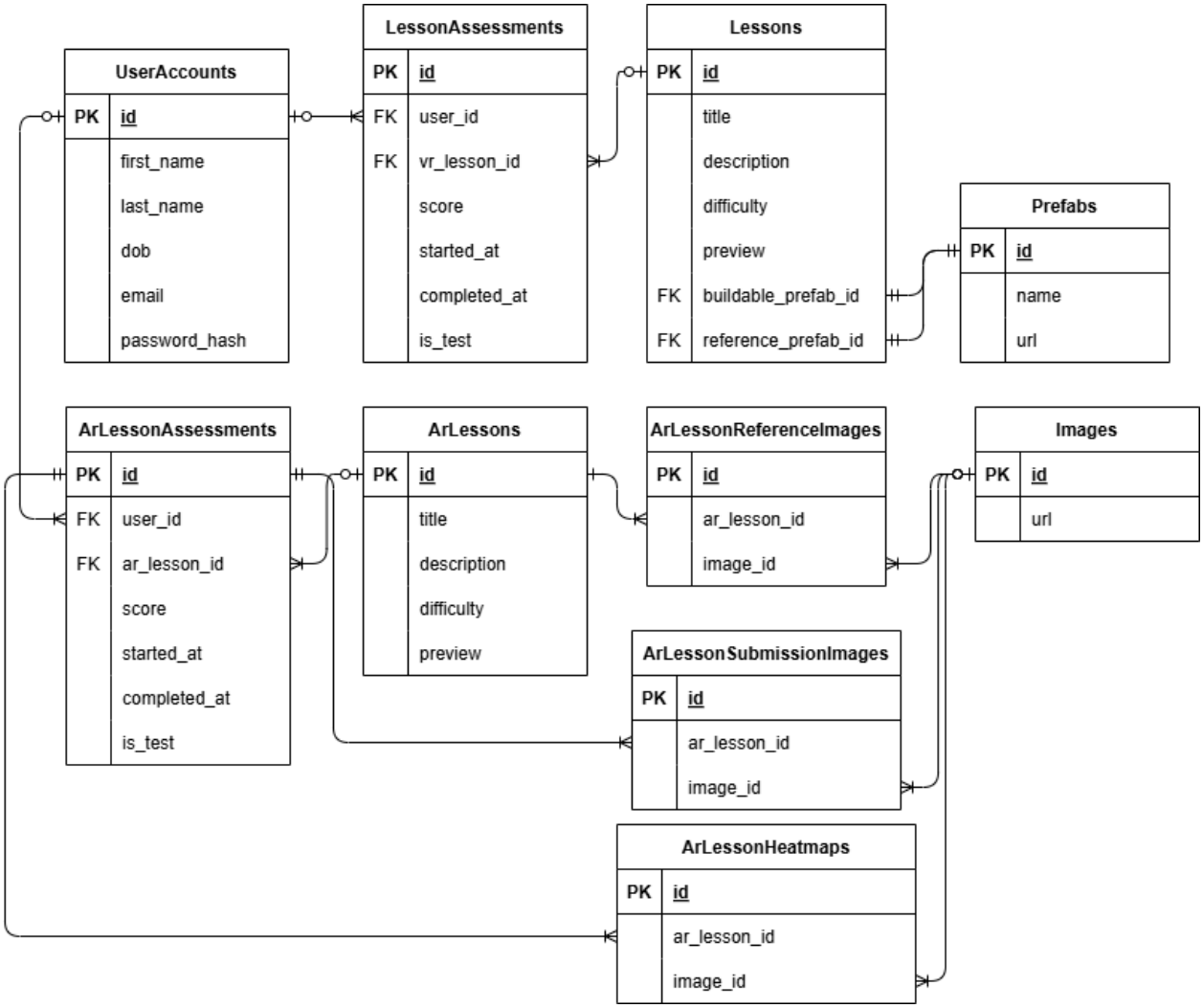
Once the user successfully completes the model or task, the system finalizes the session.

***6. End screen with time and score***

The system displays a summary screen showing the user's performance, including the total time taken and score achieved.

This diagram demonstrates the user-centered flow of the VR application, focusing on interactivity, progress tracking, and performance feedback.

## 3.2.4 Database Diagram




Figures 3.5 – Database Scheme

### *A. Overview*

- The **UserAccounts** table holds records of important user information. The email field holds their email address, used for mailing as well as login. The password_hash field holds the hash of their password to check against during login. Other fields are for a better user experience down the line, such as showing the player's name for other users later on.
- Each lesson (VR or AR) has some general information about the lesson, so the user can get a quick idea about the lesson without having to visit it. The preview field holds the url to an image that will be shown to the user while they browse through lessons.

***B. Tables for VR application***

- The **Lessons** table stores information about VR lessons that are available through the system, and the ArLessons table stores information about the VR lessons that are available.

- The **Prefab** table stores the url and name of a Unity Asset Bundle that has a GameObject of the model that the user will be building in the application.
- Every time a user completes a VR lesson in either learning mode or test mode, a record is added to the **LessonAssessment** table, storing the user's score, the time at which they started, and the time at which they finished. The start time and end time will allow the system to derive the time taken to complete the lesson. The table has two foreign keys: user_id referencing the user who has completed the lesson, and lesson_id referencing the lesson that the user completed. The is_test field indicates the mode in which the user has completed the lesson.

***C. Tables for AR application***

- The **ArLessonReferenceImages** table is a junction table between the ArLessons table and the Images table. An AR lesson needs to have reference images that define what the result should look like.

- Each record in **ArLessonAssessments** has a user_id and an ar_lesson_id pointing to the user who completed the AR lesson and the AR lesson that has been completed respectively. In addition, there are submission images submitted by the user for each assessment.
- **ArLessonSubmissionImages** is a junction table that links between the images submitted by the user and the ArLessonAsessment record.
- **ArLessonHeatmaps** is a junction table that links between the images produced by the image similarity script and the ArLessonAssessment record.

# Chapter 4
# Implementation

# Functions, Techniques, and Algorithms Implemented

## 4.1 Development of the Interactive Frontend in Unity

***Function description:***

The frontend of the VR/AR educational system was developed using Unity and is responsible for rendering immersive environments, managing user interactions, and delivering instructional content. It allows users to navigate virtual workshops, interact with 3D objects to build realistic models such as motorcycles or furniture in VR, and utilize mixed reality tools—such as instructional videos or reference images in AR—to assist with real-world construction tasks. The frontend also features responsive UI elements designed for VR and AR input

systems, including hand tracking and controller input allowing the user to be fully immersed in the task at hand.

***Key functions include:***


Figure 4.1 – Lesson Select Menu

- **Main Menu System**: Allows users to choose between training modules (e.g., mechanic, carpenter), as shown in Figure 4.1.

Figure 4.2 – Preview of the bike lesson scene

- **Scene Management**: Loads different environments and lesson scenarios, such as the bike lesson scene shown in Figure 4.2

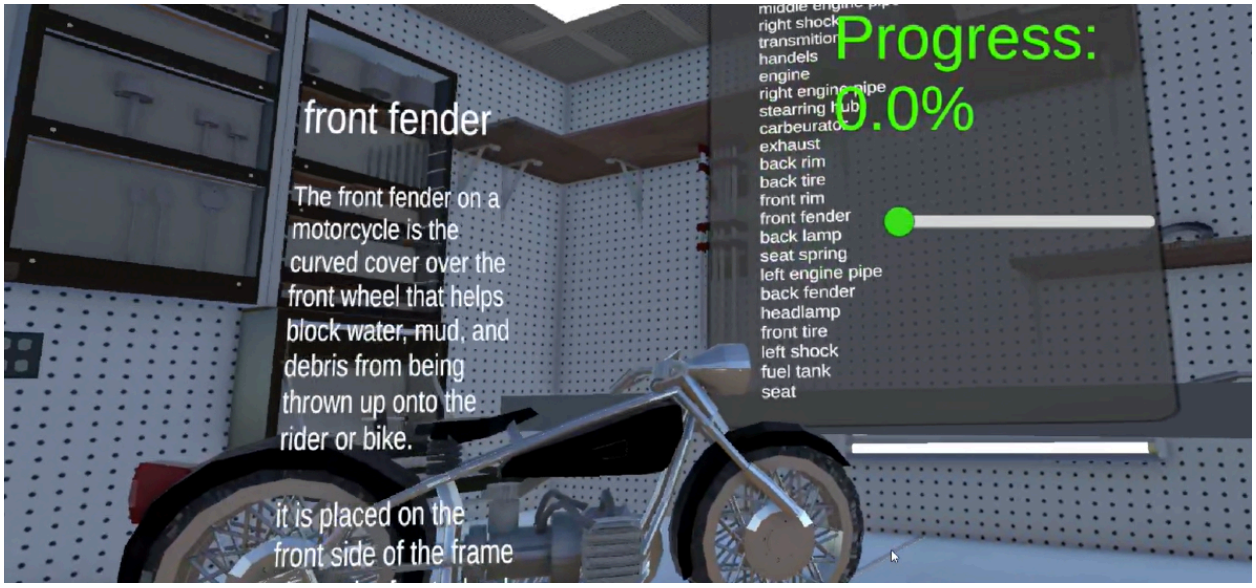



Figure 4.3 – User guidance within the application + reference model

- **Interactive Tutorials**: Guides the user step-by-step through job tasks.

Figure 4.4 – Wardrobe assembly lesson in test mode, no reference model is included

- **Assessment Interface**: Presents users with practical tasks or quiz-based assessments, records performance, and provides feedback based on task completion accuracy and time.

- **AR Companion View**: Displays supplementary content on AR-capable devices, including video guides, and instructional overlays positioned in real-world space.

- **AR Image Capture and Build Tracking**: Allows users to take photos of their real-world progress using the AR interface to document and compare their physical build with the virtual model. These images are reviewed as part of assessment and submitted to our AR evaluation model.

- **Object Interaction and Snap Systems**: Enables users to pick up, move, and connect parts (e.g., tools, screws, furniture pieces) using natural interactions like hand gestures or VR controllers, with snapping and alignment assistance to mimic real-world accuracy.

- **Progress Tracking and User Profiles**: Saves user data such as completed lessons, test scores, and captured builds, allowing learners to track improvement over time.

- **Immersive Instructional Feedback**: Uses voice and visual cues (e.g., highlights) to correct user mistakes or provide next-step guidance during interactive tutorials.

***Techniques and Algorithms:***

Several Unity-specific techniques and algorithms were applied to develop the frontend of the application:

- **Scene Management**: Unity's `SceneManager` was used to load and unload training modules dynamically, reducing memory usage.
- **Asset Loading:** Unity's `Addressables` package provides great utility for loading all sorts of remote assets including entire game objects with their scripts.
- **XR Interaction Toolkit**: Used to provide universal interaction components that work across both VR and AR. This included ray-based interactivity for VR and touch interaction for AR.
- **Custom UI System**: The user interface was built using Unity's **World Space Canvas** to ensure menus and feedback were anchored in 3D space and accessible via XR interactions.

- **Task Flow System**: A state machine was implemented to control tutorial progression, guiding the user through a sequence of instructional steps.
- **Data Persistence**: Scriptable Objects and PlayerPrefs were used to store session data, such as completed lessons and test results.
- **Gaze and Hand Gesture Tracking**: Implemented for hands-free interaction using hand recognition when supported with Meta Quest hand tracking and xr interaction tool kit hand tracking.
  **Raycasting Algorithms**: Used to detect user input, highlight interactable objects, and register selection via triggers or button presses.
- **VR evaluation:** We developed a custom script to evaluate the user's build by leveraging Unity's object naming system. The evaluation runs in two stages:

1. Socket Discovery (O(n)): The system first searches through the scene to identify all available connection sockets, which scales linearly with the number of sockets present.

2. Placement Verification (O(1)): Once sockets are registered, the script performs real-time evaluations whenever two objects are connected. It directly compares the names of connected parts using predefined name pairs, allowing for immediate (constant-time) correctness checks during interaction.

This structure allows for fast, responsive feedback to the user during assembly while keeping the initial setup phase lightweight and scalable.

***New Technologies Used***

Several new and emerging technologies were integrated into the front-end system to ensure immersive and intuitive user experiences:

- **Unity XR Interaction Toolkit (XRIT):** Provided a cross-platform framework for building interactive VR/AR applications, simplifying controller and hand input handling.

- **Oculus/Meta SDK and OpenXR:** Enabled support for multiple devices including Meta Quest 2/3 and PC VR through OpenXR, ensuring device flexibility.

And most importantly for our AR project integration.

- **Meta XR Core SDK**

  The Meta XR Core SDK (formerly known as the Oculus Integration SDK) enables developers to access the Meta Quest 3's camera and build mixed reality experiences. This SDK includes key components such as the Passthrough API,

which allows developers to render the headset's camera feed directly in Unity, creating immersive mixed reality environments. Additional tools like the Scene API provide scene understanding capabilities, enabling the detection of real-world surfaces and objects. For integrating camera-based features such as capturing real-world images, developers can use components like `OVRCameraRig`, `OVRPassthroughLayer`, and `OVRSceneManager`. These tools allow the passthrough feed to be rendered onto textures in Unity, which can then be captured and saved as images using Unity's `RenderTexture` and `Texture2D.ReadPixels()` methods. This functionality allowed us to access the cameras of the meta quest 3 to capture images of the users building journey allowing us to assist in the progress and assess the users build at the end of his AR experience with our application.

## 4.2 Development of the Backend Web Application:

The backend of the system was developed using ASP.NET Core, using a modern and modular architecture that separates concerns across data access, business logic, and HTTP interface layers. Key technologies include Entity Framework Core for data persistence and ASP.NET Identity for authentication and authorization. A SQL Server database serves as the underlying storage engine.

### *API Design*

The system exposes a (Representational State Transfer) REST interface built with controller-based APIs. Each controller is responsible for handling HTTP requests related to a specific domain, such as user management, lesson retrieval, or content submission. The structure follows clear separation of concerns, with minimal logic in the controllers themselves. Business rules and data access are delegated to services.

### *Dependency Injection and Inversion of Control*

The backend architecture leverages ASP.NET Core's built-in Dependency Injection (DI) container to manage object creation and enforce Inversion of Control (IoC) across the application. Services, repositories, and other application components are registered in the Program.cs file and injected wherever needed via constructor injection. This promotes loose coupling, improves testability, and aligns with SOLID design principles—particularly the Dependency Inversion Principle. Interfaces are used to abstract service contracts, making it easy to substitute mock implementations during unit testing. Scoped, singleton, and transient lifetimes are appropriately configured based on the use case of each service. This design pattern simplifies the flow of dependencies throughout the system while keeping the codebase modular and maintainable.

***Authentication and Authorization***

ASP.NET Core Identity was used to implement authentication and role-based access control. Users are assigned roles (e.g., Trainee, Admin), and endpoints are protected accordingly. Custom policies

and role checks are enforced via attributes and middleware. Token-based authentication is employed to secure API communication.

***User Context Middleware***

To facilitate easy access to the data of the authenticated user across the application (especially in services where HttpContext is not directly accessible), a custom middleware was implemented. This middleware extracts and caches user-related information from the request context and makes it available via a scoped service throughout the request lifecycle. This abstraction simplifies user-specific logic across the backend and improves maintainability.

***Data Access and Entity Framework Core***

Entity Framework Core was used for object-relational mapping, allowing for a clean and expressive way to interact with the SQL Server database. The schema includes entities for users, lessons, prefabs, interactions, and related metadata. Relationships (e.g.,

one-to-many, many-to-many) are modeled using EF Core conventions and Fluent API configurations where needed.

Migrations were used to evolve the database schema over time, with changes tracked and applied through `dotnet ef` commands.

***DTOs and AutoMapper***

To decouple internal models from external-facing APIs, Data Transfer Objects (DTOs) were defined for input and output operations. This ensures security, validation, and clarity in API contracts. AutoMapper was configured to handle the mapping between domain models and DTOs, reducing boilerplate and enhancing code readability.

***Validation***

Input validation was handled primarily using .NET Data Annotations, such as [Required], [MaxLength], [Range], and custom validation attributes where needed. These were applied to DTOs to ensure clean and consistent input before business logic

execution. Invalid requests return standardized error responses using ASP.NET Core's built-in model validation mechanism.

***Calling AR Assessment/Evaluation:***

AR Evaluation is used as a python script that takes a folder of submitted images, and another folder of reference images as input then outputs heatmaps to a specified directory. This seemed like the best way to handle the use of the python script, without increasing delays much further.

## 4.3 AR Evaluation Methods:

### *Approach 1 / 3D Reconstruction:*

The first approach we tried was to capture multiple images of the constructed object from different angles, then passing them through a Multi-View 3D reconstruction model. The reconstructed model would then be compared –via IoU– to a reference model. The user would pass the lesson if he passes a predefined threshold, otherwise, the parts that do not match with the reference model would be highlighted, and the user would be required to fix them, or provide clearer images for a better and more precise reconstruction. Initially, the original Pix2Vox and Pix2Vox models were tested, however the reconstruction results were unsatisfactory. Both models showed results that can't be relied on as shown in Figure 4.1.

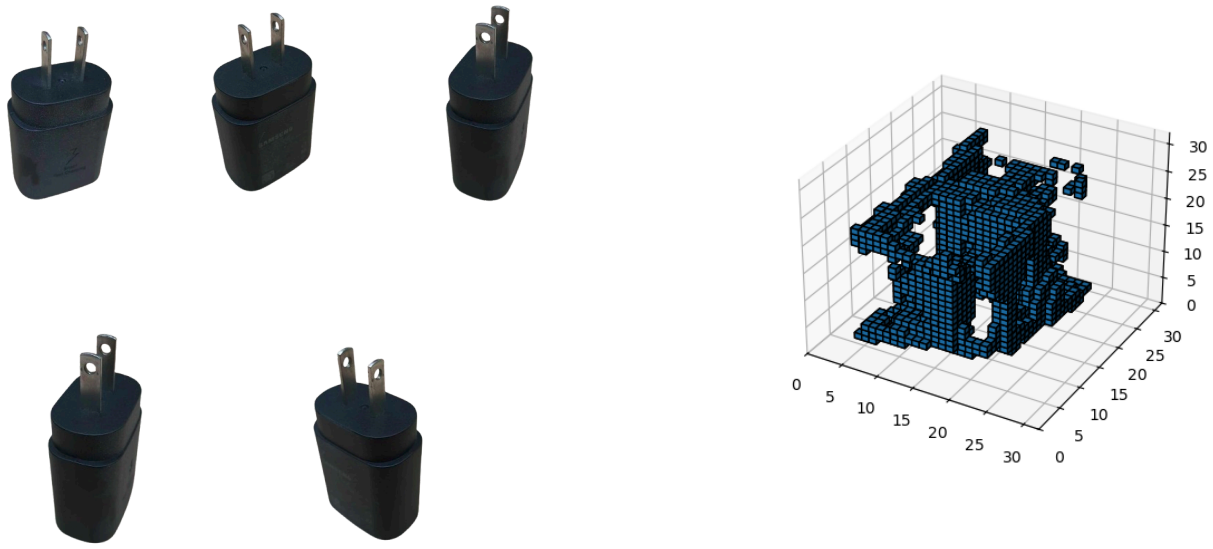



Figure 4.5 – Pix2Vox failed to reconstruct charger's head

We modified Pix2Vox's architecture as an attempt to improve its IoU score and make it more reliable in our project. Table N shows some of the methods/changes we applied on the architecture.

More recent backbone networks such as ViT and ConvNext were used, to improve the quality of the features extracted from the input images, as well as versions of backbone networks that were pre-trained on ImageNet22k, as it has more diverse classes which should –at least in theory– result in a better generalization, additionally skip connections between the encoder and the decoder were added, which wasn't present in the original architecture. Finally, in one of the attempts, we changed the decoder's upsampling block to match ConvNext's block (see Figure 4.2) but with Transpose Convolutions rather than Convolutions.

| | Ours | Pix2Vox++ |
|---|---|---|
| Backbone Network | ResNet/ConvNext/Swin/ViT | ResNet50 |
| Backbone's Dataset | ImageNet22k/ImageNet1k | ImageNet1k |
| Encoder-Decoder Skip Connections | YES | NO |

| | | |
|---|---|---|
| Activation Functions | Leaky ReLU / GeLU | Leaky ReLU |
| Upsampling Block | Simple TConv/ConvNext style | Simple TConv |

Table N. Changes done to the architecture

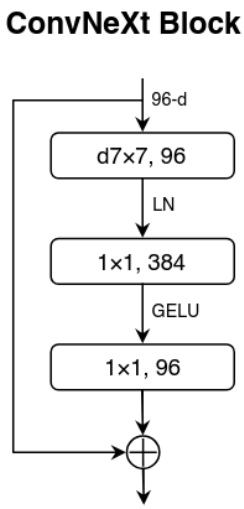



Figure 4.6 – ConvNext Block

The following table (Table N) shows the results of different architectures that we built.

| Changes made to Architecture | IoU@30 |
|---|---|
| - Backbone: ViT+ResNet[N]<br>- Encoder-Decoder Skip-connections<br>- Changed some layers parameters | 52% |
| - Backbone: ConvNext<br>- ImageNet22k<br>- Changed some layers parameters | 54% |
| - Backbone: SwinTransformer<br>- ImageNet22k<br>- GeLU activation function [rather than LReLU]<br>- Changed some layers parameters | <55% |
| - Backbone: ConvNext<br>- Upsampling Block: ConvNext style<br>- ImageNet22k<br>- GeLU activation function | 54% |

In conclusion, all modifications done to the architecture failed to even reproduce the original network's performance. And due to the lack of research done on multi-view reconstruction as well as

the given time constraints, we deduced that the evaluation via 3D reconstruction is an overly optimistic approach and that maybe this technology is probably not ripe enough yet to be used in such applications. so we worked on an alternative approach.

***Approach 2 / Image Similarity comparison:***

The alternative approach that we considered, to evaluate the structures built by the end-user was to prompt the user to capture images of the object from specific angles, and then compare the pictures captured by the user to reference images of corresponding angles. The output of this method is a heatmap showing where the 2 images most differ, and where they match. To achieve this, we needed comparison methods that would satisfy the 2 following conditions:

1. should be invariant to **small** shifts or angle differences
2. should be able to highlight the misplaced parts of the structure, rather than giving a single similarity score value for the 2 pictures in general.

The second condition can be met by dividing both images into patches, and comparing each two corresponding patches together via some image similarity method – again, rather than comparing the 2 whole images–.

The image similarity methods that we looked into were **SSIM** and **LPIPS**.

**SSIM** is a method that compares between 2 images in terms of their Luminance, Structure, and Contrast (see Eq. 2)

Whereas **LPIPS** measures perceptual similarity between image patches using deep features extracted from trained neural networks. The features from multiple layers are normalized, weighted, and compared, reflecting human perceptual judgments better than traditional metrics (see Eq. 3).

$$SSIM(x,\ y)\ =\ [l(x,\ y)\ \cdot\ c(x,\ y)\ \cdot\ s(x,\ y)]$$

Equation 4.1 – $l$ is the luminance between x and y, $c$ is the contrast and $s$ is the structure

$$LPIPS(x,\ y)\ =\ \sum_{l} w_{l} \cdot \left|\left| \hat{\phi}_{l}(x)\ -\ \hat{\phi}_{l}(y)\ \right|\right|_{2}^{2}$$

Equation 4.2 – $w_l$ are weights given for each layer, that are learned using human judgement $\hat{\phi}_l$ is the activation feature map for layer $l$

After several experimentations, it turned out that SSIM is extremely sensitive to even the slightest shifts or changes between images. As an attempt to address this issue, we took a large corpus of reference images each with extremely small translations and rotations –instead of a single reference image– and then increased this number of the reference images by applying augmentation over them. After comparing the user's image with each image in the corpus, the scores of all patches for each image would be summed, and the reference image with the highest sum would be chosen as the closest evaluator, however this method also failed to account for SSIM's extreme sensitivity (see figure 4.3).

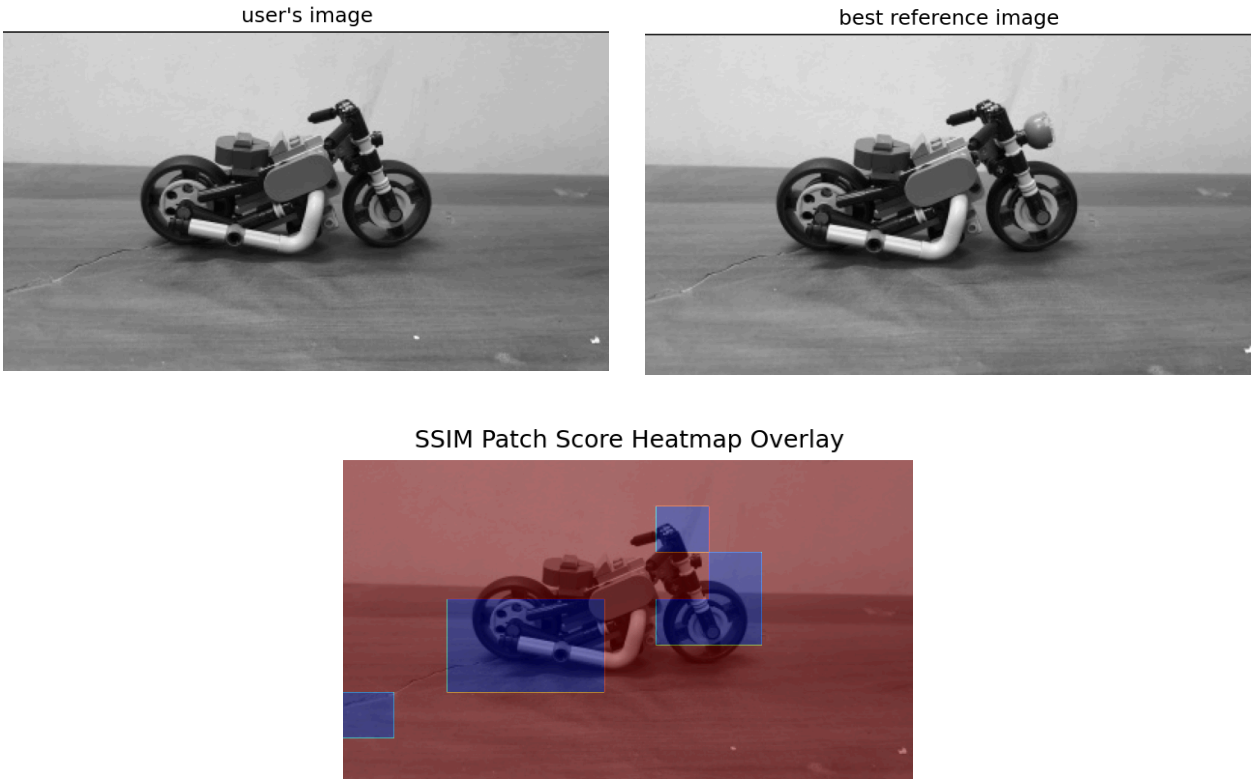



Figure 4.7 – SSIM managed to capture the difference between the 2 images -the missing headlight- but it also has lots of false positives as it marked a lot of parts as different while they are clearly not different

This problem however, was not present in LPIPS, as it showed more tolerance towards slight changes and shifts, while accurately

capturing the clear differences between the user and reference images (see figures N and M).

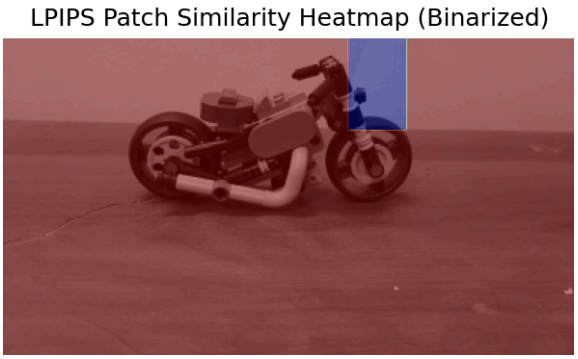



Figure 4.8 LPIPS managed to capture the difference without getting influenced by any noise or slight variations.

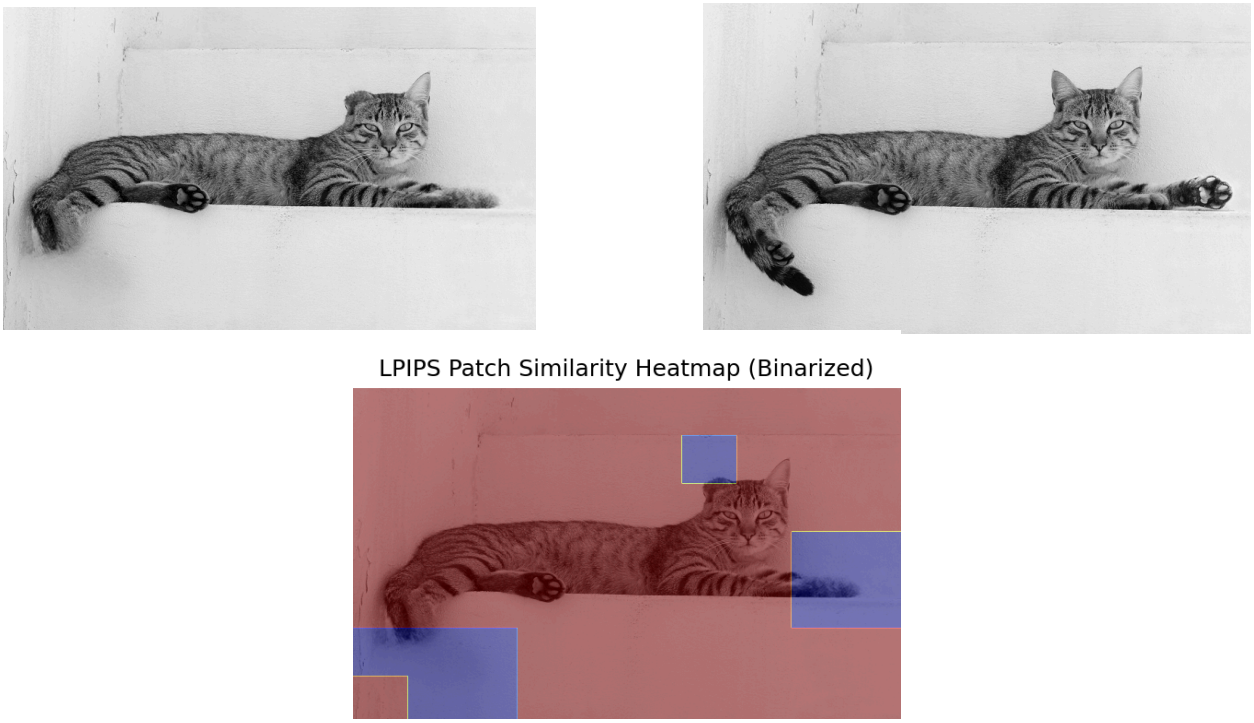



Figure 4.9 Another test case showing LPIPS ability to accurately capture only the noticeable differences

In conclusion, LPIPS proved to be a robust method to capture the perceptual differences between the user's work and the reference

work, which is why we settled on this method to evaluate the AR builds.

# Chapter 5
# User Manual

## 5.1 Installation GuideTo install and use the VR/AR Educational System on your Meta Quest  follow the instructions below to sideload the application.

***Step 1: Download the APK***

Visit our GitHub repository to download the latest build of the application:

- https://github.com/Shokryy/the-learning-lens-ar (AR APP)
- https://github.com/ali-hy/the-learning-lens (VR APP)

Download the `.apk` file from the "Releases" section or a provided direct link.

***Step 2: Set Up Your Quest 3 for Sideloading***

1. Enable Developer Mode:
    - Open the Meta Quest mobile app.
    - Go to Devices > Developer Mode and toggle it ON.

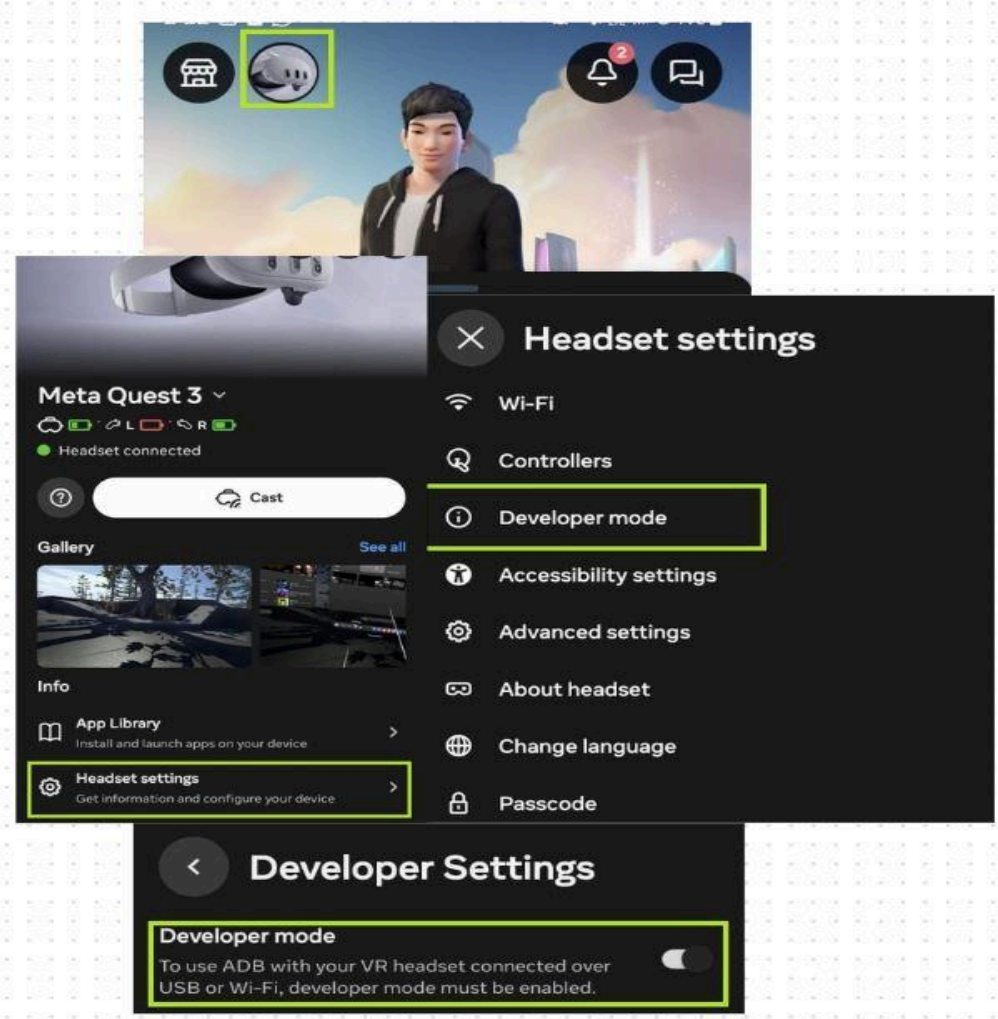



Figure 5.1 –Enable Developer Mode

2. Connect your Quest 3 via USB to your PC and approve any authorization prompts.

***Step 3: Install the APK Using SideQuest or MQDH***

Option A: Using Meta Quest Developer Hub (MQDH)

- Download MQDH from https://developer.oculus.com/downloads/.
- Launch MQDH and select your connected Quest 3.
- Drag and drop the `.apk` file into the hub or click "Install APK".

Option B: Using SideQuest

- Download SideQuest from https://sidequestvr.com/.
- Launch SideQuest with your headset connected.
- Use the "Install APK file from folder" option and select your `.apk`.

***Step 4: Launch the App on Your Headset***

1. Put on your Meta Quest 3.
2. Open the Apps menu.
3. Switch the filter to Unknown Sources.
4. Locate and launch your application.

***System Requirements:***

Meta quest 3 is recommended for both vr and ar apps

Minimum for vr app is meta quest 2

Minimum for ar app is meta quest 3s

## 5.2 Operating Instructions

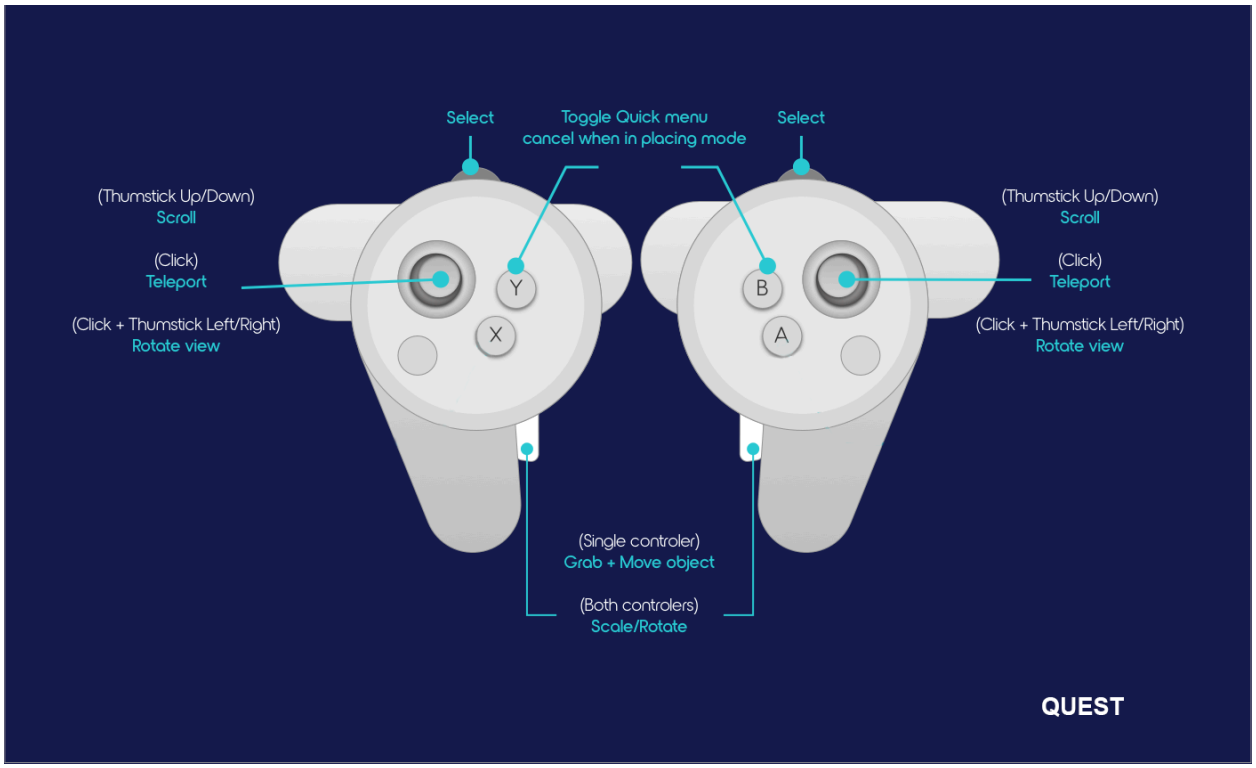



Figure 5.2 – Meta Quest Controllers

### *VR Application*

After launching the VR application, you will be greeted by the **Main Menu Room**. Use the **analog sticks** on your controllers to move around, and use the **trigger buttons** to interact with menu panels.

You can choose to **log in or sign up** to access the full feature set, or **continue as a guest** for a quick overview of the system. Once logged in, select the **training category** (e.g., mechanic, carpenter), then choose your desired **lesson** or **test**.

You will be transitioned into the selected training scene, where you can navigate freely and begin assembling the chosen object (e.g., a motorcycle or piece of furniture). Use the **grab buttons** on your controllers to pick up, move, and place virtual components. Throughout the process, various in-app tools and guides will assist you.

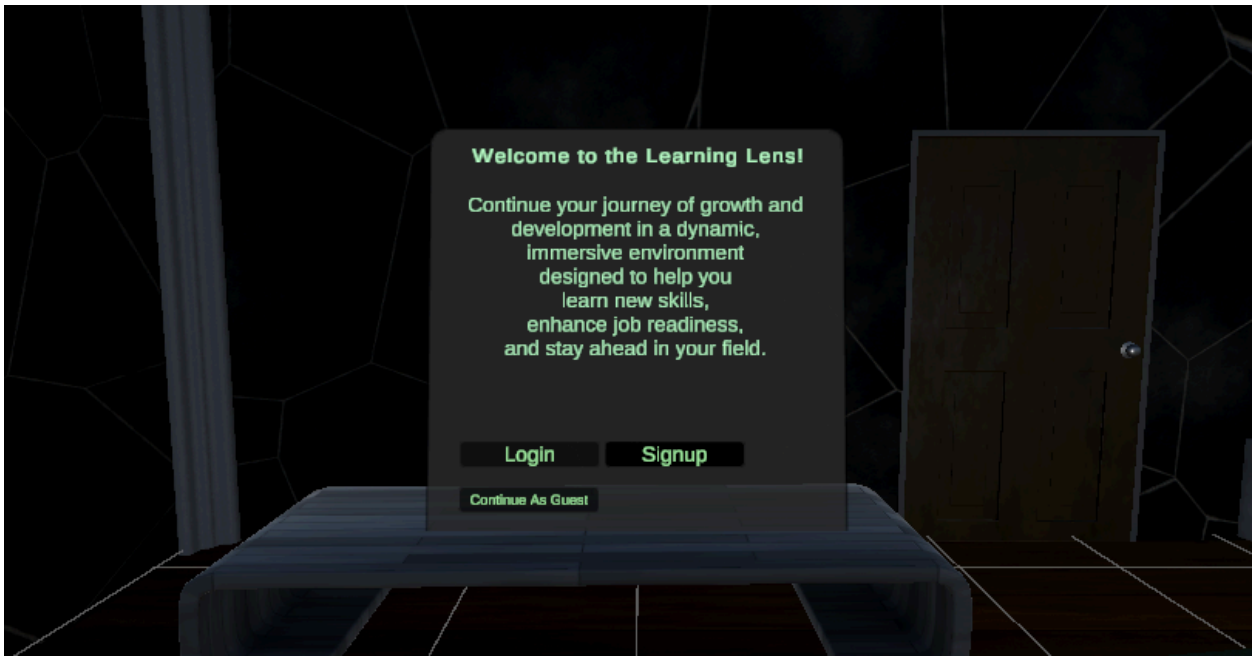


Figure 5.3 –Main Menu Panel

***AR Application***

When you launch the AR application, you will see your **real-world surroundings** through the passthrough view. Use your **hands** to interact with the on-screen UI panels.

Choose an object from the available collection to **build in AR**, either in **guided lesson mode** or **test mode**. During tests, you will be required to **capture photos** of your completed build from four sides and a top-down view.

To take a photo, you can:

- **Pinch your right index finger and thumb** while looking directly at the object, or

- Press the **A button** on your right-hand controller.

These photos are used by the system to evaluate your build and generate your final **evaluation score**.

# Chapter 6
# Conclusions and Future Work

## 6.1 Conclusions

This project introduced an immersive VR/AR educational system designed to teach and assess users in practical, hands-on assembly tasks such as mechanical and carpentry models. Through a carefully designed three-tier architecture, the system combined a Unity-based interactive frontend with a modular ASP.NET Core backend to deliver real-time, user-centric training experiences.

The VR component enabled users to engage in simulated environments with responsive object interaction and guided tutorials, while the AR component allowed learners to replicate real-world assemblies and receive performance feedback through advanced image similarity evaluation. The system supported account-based progress tracking, role-based access, and used modern techniques such as hand tracking, gaze input, and heatmap-based assessments.

While the evaluation of AR builds faced technical challenges, especially with 3D reconstruction, the project successfully implemented a robust and scalable solution using perceptual similarity metrics like LPIPS. Overall, the application demonstrates the potential of XR technologies to reshape how practical skills are taught and evaluated, laying a strong foundation for further development and future integrations.

## 6.2 Future Work

***Trainer Role***

The system overall can be improved by introducing a trainer role, and making use of these roles to authorize different actions. A trainer would have access to his trainees' data to measure improvement over time, allowing better group work. Trainers could assign tasks / lessons to trainees.

### *Web application portal*

People don't always have access to a VR headset, so it would be useful to have a web application portal that can run in the browser. This is useful for both trainers and trainees, but I can imagine trainers benefiting from it even more than the trainees.

### *Unity SDK for building models*

The training scene in our unity application works perfectly on any model made with many sockets, allowing developers or even technically capable users to create new models and upload them to the system. The only problem is the lack of an SDK to guide the process, and constrain the user from making unsuitable models for the application. A Unity SDK would make it significantly easier for anyone to ensure their models comply with our system, thus allowing for a community-driven system for anyone who is interested.

### *Filtering of lessons*

With more lessons it will be difficult to navigate and browse through them all. The user will need ways to search for lessons either by title, description, or even category. The introduction of categories would be useful for most users for finding the model(s) they want to learn about.

### *AR Assessment*

As multi-view 3D reconstruction methods become more reliable, our AR system can benefit from more precise and accurate assessments of structures built by the user. This will enhance user feedback, reduce reliance on exact viewpoints – as was the case with image similarity based methods – and support a more intuitive learning experience.

# References

[1] Xie, H., Yao, H., Sun, X., Zhou, S. and Zhang, S., 2019. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 2690-2698).

[2] Xie, H., Yao, H., Zhang, S., Zhou, S. and Sun, W., 2020. Pix2Vox++: Multi-scale context-aware 3D object reconstruction

from single and multiple images. International Journal of Computer Vision, 128(12), pp.2919-2935.

[3] Lee, J.J. and Benes, B., 2023. SnakeVoxFormer: Transformer-based Single Image\\Voxel Reconstruction with Run Length Encoding. arXiv preprint arXiv:2303.16293.

[4] Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P., 2004. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing, 13(4), pp.600-612.

[5] Zhang, R., Isola, P., Efros, A.A., Shechtman, E. and Wang, O., 2018. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 586-595).

[6] Chen, J., Lu, Y., Yu, Q., Luo, X., Adeli, E., Wang, Y., Lu, L., Yuille, A.L. and Zhou, Y., 2021. Transunet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306.

[7] Morkos, B., Taiber, J., Summers, J., Mears, L., Fadel, G. and Rilka, T. (2012) 'Mobile devices within manufacturing environments: a BMW applicability study', International Journal of Interactive Design and Manufacturing, 6(2), pp. 101–111. DOI: 10.1007/s12008-012-0148-x.

[8] Chen, H. and Liu, X. (2021) 'Research on the Application of "AR/VR+" Traditional Cultural Education Based on Artificial Intelligence', in 2021 2nd International Conference on Information Science and Education (ICISE-IE). DOI: 10.1109/ICISE-IE53922.2021.00370.

[9] Ye, Z. and Sitthiworachart, J. (2021) 'Curriculum System of Preschool Education under the Background of AR Intelligence', in 2021 International Conference on High Performance Big Data and

Intelligent Systems (HPBD&IS). DOI: 10.1109/HPBDIS53214.2021.9658441.

[10] Francisco, D., Gonçalves, A., Cruz, A., Rodrigues, N. and Ribeiro, R. (2023) 'Augmented Reality and Digital Twin for Mineral Industry', in 2023 International Conference on Graphics and Interaction (ICGI). DOI: 10.1109/ICGI60907.2023.10452719.

[11] Singh, G. and Ahmad, F. (2024) 'An interactive augmented reality framework to enhance the user experience and operational skills in electronics laboratories', Smart Learning Environments, 11(5). DOI: 10.1186/s40561-023-00287-1.