A VLM-Driven Workflow for Generating Context-Specific Fire Hazards for VR Construction Safety Training

Kexin Liu[1], Mohamed Sabek[1], Gaang Lee[1], Max Kinateder[2], Vicente A. Gonzalez[3]

[1]Infrastructure Human Tech (IHT) Lab, Department of Civil and Environmental Engineering, Faculty of Engineering, University of Alberta, Edmonton, Alberta, Canada
[2]Construction Research Centre, National Research Council Canada, Ottawa, Ontario, Canada
[3]Infrastructure and Human Tech (IHT) Lab, Strategic Projects Insight Centre in Engineering (SPICE), Department of Civil and Environmental Engineering, University of Alberta, Canada
kexin10@ualberta.ca, sabek@ualberta.ca, gaang@ualberta.ca, Max.Kinateder@nrc-cnrc.gc.ca, vagonzal@ualberta.ca

Abstract –

While Virtual Reality (VR) provides a promising platform for construction safety training, generating virtual environments (VEs) that are tailored to specific project and task contexts remains a bottleneck. Current workflows, ranging from manual modeling to semi-automated BIM-to-VR pipelines, are often labor-intensive and result in static, idealized VEs that fail to capture the dynamic reality of active construction sites. This paper addresses these limitations by proposing a cyclical framework driven by Vision-Language Models (VLMs), designed to both generate procedural fire hazard scenarios and systematically capture expert knowledge for continuous model improvement. By focusing on fire safety, a domain requiring a deep understanding of spatial relationships between objects and activities, the workflow integrates VLMs to extract tacit risk context that traditional object detection misses. Human-in-the-loop corrections then serve as ground-truth data for future model fine-tuning. As an initial proof of concept, we developed a semi-automated Unity prototype that connects unstructured site imagery with 3D asset generation through a hybrid generative pipeline. The prototype illustrates the potential of the proposed workflow to transform raw construction imagery into privacy-preserving, context-specific digital twin training environments. By reducing the manual effort required for VR content creation, this approach lays the foundation for scalable, self-evolving AI-driven safety training systems.

Keywords –

Construction safety, Fire hazards, Virtual Reality, Vision-Language Models

1 Introduction

Construction fire safety remains a persistent challenge, driven by the complex, dynamic nature of job sites and the interactions among workers, equipment, and changing site conditions [1]. VR is a promising solution by providing controlled environments for behavioural research and immersive platforms for safety training [2]. However, the creation and maintenance of VEs for training remains a major bottleneck, limiting both research validity and the scalability of VR-based training systems in practice [3].

This bottleneck is driven by two intersecting challenges. First, high-quality VR training scenarios still rely heavily on the tacit knowledge from senior safety personnel [4]. Existing literature, standards, and reported incidents provide important general principles, but they rarely describe hazards at the level of detail needed to build convincing, site- and task-specific VR scenarios. Additionally, each construction project is unique, requiring flexible, context-specific judgement and decision-making. As a result, expert input remains essential at two crucial stages: (1) to accurately identify complex, context-dependent hazards from raw site data [5]; and (2) to validate the realism of the resulting virtual simulation [4]. Despite this importance, this tacit knowledge of experts is rarely formalized or reused in a structured way [6]. At the same time, the construction industry faces a worldwide aging workforce and accelerating labor shortages [7-9]. As experienced personnel retire, this specialized domain expertise is at risk of disappearing. Without a mechanism to capture and digitize this visual reasoning, the industry risks losing the capacity to generate and verify realistic safety scenarios.

Second, even with expert knowledge available, the technical workflows for turning this hazard-relevant knowledge into VR content remain inefficient. Current

digital twins incrementally update elements or zones based on scans, sensor data, or manual inputs, which is effective for progress monitoring and quality control [10]. However, for safety training, it is still time-consuming and labour-intensive to propagate these updates into VR environments [11]. Most importantly, within the large volume of new data, only a subset of the updated information is needed for safety training. What matters are fire hazard–relevant changes, such as new ignition sources or flammable materials introduced by new construction activities, rather than a perfect geometric copy of every site detail. This suggests an alternative strategy in which, instead of fully twinning the whole site, we focus on twinning hazard-relevant information while still reflecting the local context where workers operate.

Building on these challenges, this paper presents a semi-automated, VLM-driven workflow that twins hazard-specific information into VR while capturing and storing expert knowledge. As a proof-of-concept, we developed an initial prototype implementation within the popular game engine Unity to illustrate the practical feasibility of the proposed framework.

2 Related Work

2.1 Construction VR Scene Generation

VR is increasingly used in the architecture, engineering, and construction (AEC) sector for design reviews, project planning, and safety training, providing more transparent communication and improved learning outcomes [12]. However, achieving high ecological validity in VR scenes, meaning they accurately reflect task-relevant conditions in specific construction environments, remains challenging due to their complexity [3]. These environments contain many objects, changing site conditions, and heterogeneous data sources such as BIM, point clouds, and GIS, all of which require significant cleaning and optimization before becoming VR-ready [13].

Currently, there are two primary workflows to generate virtual environments. Semi-automated BIM-to-VR pipelines are the most common but frequently suffer from loss of BIM semantics during export and require aggressive geometric optimisation to meet the rendering constraints of standalone VR headsets [14]. Manual scene authoring supports highly customized simulations but is slow and requires advanced modelling and game-engine expertise [15]. Crucially, neither workflow includes context-specific hazards, such as oily rags left near heaters or other flammable materials. Standard BIM models focus on permanent, “as-designed” elements [1] and rarely capture short-lived site conditions like waste, temporary storage, or poor housekeeping. In scan- or sensor-based workflows, these transient objects are often filtered out as noise [10]. However, such temporary combustibles and unsafe arrangements are a common source of fire risk on active sites and are therefore essential components of realistic safety training scenarios.

Recent advances in generative AI offer a promising alternative. Text-to-3D and image-to-3D models, such as Hunyuan3D 2.0, Microsoft TRELLIS [16], and Meta SAM 3D [17], can synthesize high-quality 3D objects directly from natural-language descriptions or site imagery, reducing reliance on fragmented asset libraries and bypassing many manual modelling steps. However, despite these advances, the AEC domain still lacks a practical workflow that applies generative models to VR scene generation, especially for safety-critical contexts such as hazard recognition research and training.

2.2 Potential of VLMs for Construction Hazard Recognition Training

The integration of advanced computational models into construction safety management is an active and expanding area of research. Traditional computer vision (CV) techniques, often relying on convolutional neural networks (CNNs), have demonstrated success in object detection tasks, such as identifying personnel, machinery, and personal protective equipment [18]. However, these methods often struggle to interpret context and relationships among objects, which is paramount for safety analysis. A ladder, for example, is not inherently a hazard; it becomes one when it is unsecured, placed on an unstable surface, or used near an open electrical panel.

This is the gap that VLMs are beginning to fill. VLMs such as LLaVA (Large Language and Vision Assistant) [19] and GEMMA 3 [20], and those based on the GPT-4 architecture [21], combine the perceptual capabilities of CV with the sophisticated reasoning of large language models [22]. This multimodal approach allows VLMs to move beyond mere object identification to complex scene contextual processing and complex pattern recognition. Recent studies have explored using VLMs to analyze static site images for safety-rule violations by processing both the visual data and textual prompts containing regulatory guidelines [18,23]. This allows them to identify not only the presence of a potential hazard, but the reason it violates a safety protocol.

Despite these advances, current studies have largely focused on demonstrating the hazard detection capabilities of VLMs [22], while the translation of VLM outputs into reusable, context-rich VR training assets remains underexplored. In particular, limited work has examined how contextual information embedded in construction hazard scenes can be structured into a format that can be directly interpreted and rendered in a virtual environment. Additionally, the use of VLMs in safety-critical settings requires caution. VLMs may generate plausible but incorrect hazard descriptions or

safety explanations, commonly referred to as "hallucinations." They may also struggle with fine-grained spatial reasoning, such as estimating 3D proximity or depth from 2D imagery [27]. Therefore, VLM-generated outputs may require further structuring and validation by human experts before being integrated into VR-based safety training environments.

3 Research Method

We propose a five-stage, human-in-the-loop workflow designed to transform unstructured construction site imagery into structured, context-aware fire hazard libraries for VR. As illustrated in Figure 1, the framework operates iteratively. It not only generates hazard scenarios but also uses expert validation to continuously fine-tune the underlying VLM. This helps the system adapt to specific project constraints and reduces the rate of hallucinations that can contribute to unnecessary information or misleading hazard detections over time. The following sections detail each stage.

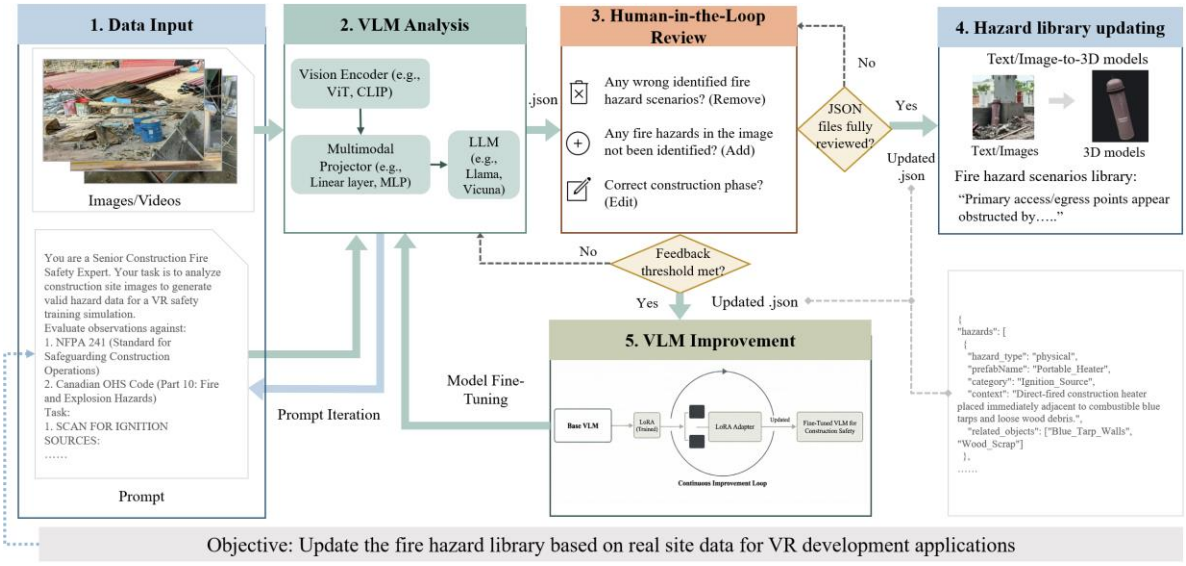

Figure 1. The proposed VLM-driven workflow for virtual fire hazard library update

3.1 Data Input

The workflow begins with the ingestion of raw visual data (images or video frames) captured from active construction sites. To ensure practical scalability and minimize additional hardware costs, the workflow can integrate with existing site CCTV surveillance systems.

To accurately translate these 2D images into valid 3D VR components, we employed a goal-oriented prompt engineering strategy [24]. The prompt design was reverse engineered from the specific data requirements of the VR Scene Library. Specifically, the system requires not only the identification of physical objects (e.g., oil drums or welding equipment) and their spatial context, but also the recognition of abstract violations. Abstract hazards, such as blocked emergency exits or missing fire extinguishers, are critical for safety training but lack a single physical object to detect. Therefore, the prompt is structured to extract both explicit physical assets and implicit procedural non-compliance.

After defining the expected output, we implemented a multi-stage prompting architecture to achieve reliable extraction. First, we applied Role-Prompting by instructing the model to adopt the persona of a "Senior Construction Fire Safety Expert"[25]. This constrains the model's focus to regulatory compliance rather than generic image description. Second, we used Chain-of-Thought (CoT) [26] to ensure the model follows a logical checklist and to mitigate hallucinations. This step-by-step reasoning is critical for visual analysis [22], as it prevents the model from guessing and ensures that every identified hazard is based on clear visual evidence. Finally, we required the output in a strict data format (JSON). This structured format ensures the data is machine-readable, allowing for direct parsing into the user interface and facilitating the automated construction of fine-tuning datasets.

3.2 VLM-based Hazard Analysis

In this stage, the VLM processes the visual input alongside the prompt to generate a structured description of potential hazards. The model identifies specific hazard instances and extracts critical metadata required for VR instantiation. This includes:

- Hazard classification, distinguishing between

physical hazards that must be represented as 3D entities in the virtual environment (e.g., heaters and wood debris) and abstract hazards that represent procedural or regulatory violations (e.g., blocked egress routes, improper material storage).

- Prefab associations, where each physical hazard is mapped to a corresponding VR asset ("prefab") to enable automatic placement within the simulation.
- Contextual metadata, which captures the hazard's relevance, associated risks, and the environmental cues driving the classification (e.g., proximity of a heater to combustible traps, presence of fuel containers near ignition sources).
- Relational linkages, documenting interactions between hazards, such as ignition sources located within NFPA 241's minimum clearance distance from Class A or B fuels, to support scenario logic and rule-based triggers in the simulation.
- Compliance attributes, representing inferred violations of NFPA 241 or the Canadian OHS Code (Part 10), enabling automated generation of safety feedback and assessment scoring.

3.3 Human-in-the-Loop Review

Given the safety-critical nature of this domain, relying on VLM output alone is insufficient due to the potential for semantic errors. We therefore introduce a Human-in-the-Loop review process [27]. A domain expert reviews the JSON output against the original image to verify the presence and classification of the hazard. The expert can correct the output by removing false positives or amending inaccurate context descriptions and can augment it by adding hazards that the VLM has missed. The review is iterative, and the JSON file is edited until the expert confirms that all images in the batch have their hazards fully marked and correctly represented. Only cases where the expert modifies the VLM output are counted as feedback and stored in a feedback repository. The final, validated JSON is stored in an updated JSON file and used both to refine prompts, hazard templates, and configuration rules for the VLM and to populate a feedback repository. VLM updates are triggered once the number of such corrections exceeds a predefined feedback threshold [27]. The specific threshold is configurable depending on project requirements and update cadence; for example, it could correspond to the completion of a scheduled image batch, such as a weekly data collection cycle. In this way, this process acts as a form of "human coaching", helping future hazard descriptions and scene suggestions better align with expert judgement.

3.4 Hazard Library Update

The validated data from Section 3.3 is used to support two parallel processes: generating 3D models (corresponding to the physical field in the VLM output) and populating the Abstract Scene Library (corresponding to the abstract field). This library serves as the semantic back-end for the VR environment. This eliminates the need to repeatedly search for site photographs, locate appropriate 3D objects, or request additional expert validation, thereby streamlining the construction of accurate, context-aware VR fire-safety training scenes.

3.5 Continuous VLM Improvement

A critical component of the proposed workflow is the establishment of a "data flywheel" (Stage 5), which ensures the VLM's accuracy and domain specificity improve over time. This stage operationalizes the feedback gathered during the Human-in-the-Loop Review (Stage 3). The "Updated.json" file generated by the human expert is not only the finalized data for the VR library but also serves as a high-quality, expert-validated ground truth dataset. This curated dataset becomes the training data for iteratively refining the VLM. By comparing the VLM's initial JSON output with the expert-corrected JSON, we can identify systematic errors in the model's reasoning, such as misclassifying construction phases, overlooking "abstract" hazards, or providing incomplete contextual descriptions.

As depicted in the workflow (Figure 1), this refinement process can be efficiently implemented using parameter-efficient fine-tuning (PEFT) techniques, such as Low-Rank Adaptation (LoRA). LoRA allows fine-tuning the VLM on this new, domain-specific dataset by updating only a small subset of the model's parameters. This approach is computationally much more efficient than fully retraining the entire model, making it feasible to run periodic improvements as more curated data is collected. The ultimate goal of this continuous improvement loop is to enhance VLM's performance in construction-specific hazard analysis, thereby reducing the manual correction burden on human experts in future iterations and increasing overall pipeline automation.

4 Implementation

This section presents a proof-of-concept prototype developed as the primary artifact to illustrate the proposed workflow concept. Accordingly, systematic parameterization, full module integration, and empirical evaluation across a larger image corpus will be addressed in future work. Unity (2022.3.46f1) was selected for its mature VR development capabilities and its ability to integrate external AI modules through standard APIs. Although the workflow can be implemented in other engines, Unity provides a stable and practical environment for rapid prototyping. Within Unity, we

implemented a custom user interface (UI) that facilitates user interaction, displays intermediate results, and coordinates communication with external system components. As shown in Figure 2, Unity serves as the central layer, linking data processing, model inference, and VR scene assembly into a cohesive pipeline.

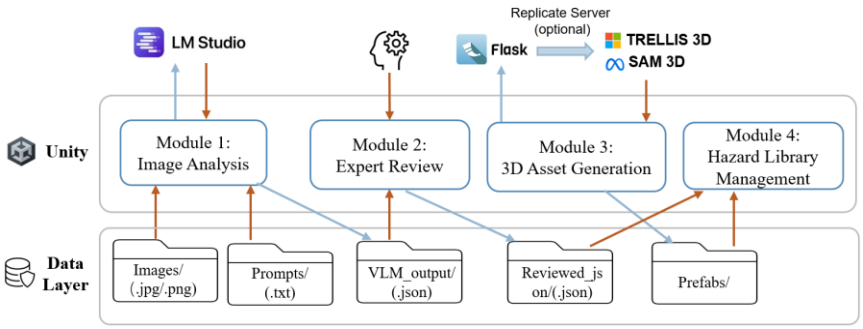

Figure 2. Schematic overview of the Unity plug-in implementing the proposed workflow

4.1 Image Analysis Module

The Image Analysis Module serves as the primary entry point for the fire hazard analysis and hazard library generation workflow. The implementation is compatible with widely used local LLM servers, LM Studio (www.lmstudio.ai/). We selected LM Studio to facilitate the efficient local deployment of quantized models via a standardized REST API, minimizing integration complexity. This enables the use of various open-source VLMs, such as LLaVA or GEMMA 3, which are well-suited for this type of detailed visual analysis (as shown in Figure 3). It is designed with local VLM backends, ensuring data privacy and control.

For a proof-of-concept demonstration, the prototype was tested using a small set of publicly available construction-site images. These images were used for feasibility illustration rather than as a formal evaluation dataset. As shown in Figure 4, in the configuration tab, the user selects the VLM model, target image, and domain-specific prompt (from Section 3.1) from the local file system. In our prototype, it is put under the Unity project asset file for easy management. Upon clicking "Analyze", the plug-in converts the image to a base64 string and serializes it into a single API request. This request is sent via an HTTP POST request to the local VLM's API endpoint (e.g., LM Studio's default server at http://127.0.0.1:1234). The plug-in receives the raw JSON text response from the VLM and displays it in the "Results Preview" field. This JSON data is then passed to the "Expert Review" module (Section 4.2), which deserializes the JSON and uses its contents to populate the interactive fields for human-in-the-loop curation. This setup provides a self-contained, integrated toolset for a VR developer to manage the entire data-to-asset pipeline without leaving the Unity editor.

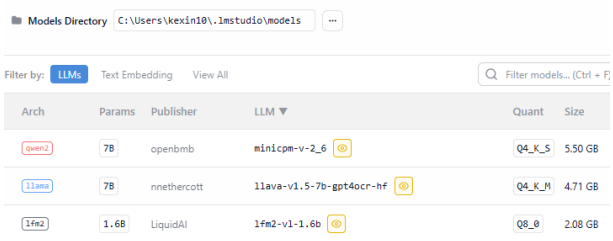

Figure 3. Locally hosted VLM models in LM Studio

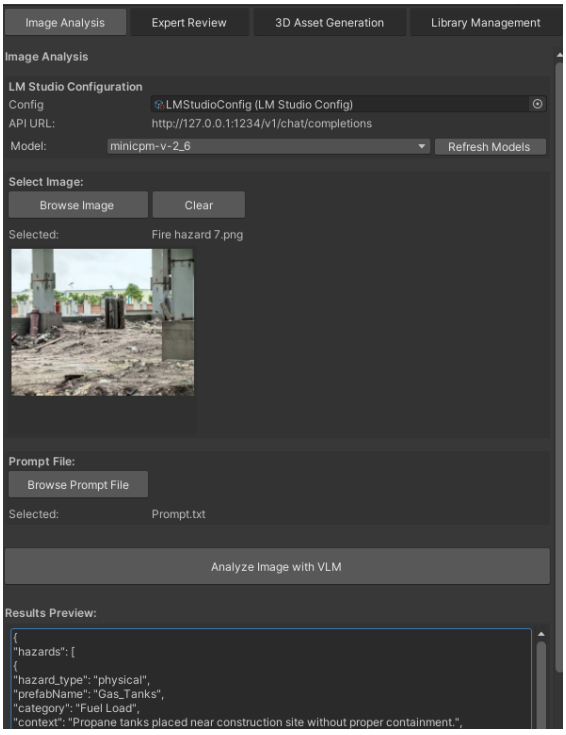

Figure 4. Image Analysis Module UI

4.2 Expert Review Module

This module streamlines the Human-in-the-Loop review stage described in Section 3.3 by providing an interface for experts to inspect and modify all identified fire-hazard scenarios. As shown in Figure 5, it starts with loading the JSON file generated by the Image Analysis module. It presents the extracted fields, such as prefab name, category, and related contextual objects, in a structured and readable format. Experts can edit any field by selecting "Edit," which switches the interface into an editable state, allowing them to correct or update the information. If a scenario has been misidentified, the expert can remove it using the "Remove" option. Conversely, if the VLM-based detection missed a relevant fire hazard, the expert can add a new one using the "Add New Hazard" function. After completing their review, experts can confirm the revisions by "Confirm All Changes." The system then generates an updated JSON file as described in Section 3.3, ensuring that all corrections and additions are incorporated into the downstream modules.

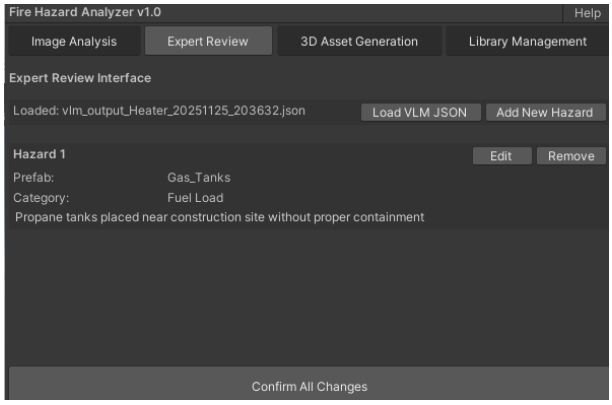

Figure 5. Expert Review Module UI

4.3 3D Asset Generation Module

Module 3 employs a hybrid asset generation pipeline. The system first queries the internal Hazard Library (Module 4) for pre-validated prefabs matching the detected hazard scenarios. If no match is found, the system triggers an on-demand generative workflow, as shown in Figure 6.

The generative process supports both text-to-3D and image-to-3D pathways by choosing different models. In this prototype, we tested the image-to-3D route using TRELLIS [16] and SAM-3D [17]. We explored two deployment strategies to integrate these generative models into the Unity environment. In the local deployment configuration, the models are hosted on the workstation (Intel Core i9-13900KF, 64 GB RAM, NVIDIA GeForce RTX 4090 24GB), and Unity communicates with them through a lightweight Python microservice built with Flask (https://flask.palletsprojects.com), which passes Unity's generation requests to the locally running 3D model. This option avoids per-generation costs but requires a high-performance GPU and sufficient memory.

Alternatively, the cloud-based configuration uses Replicate's model-hosting service (https://replicate.com), where Unity sends requests directly to the cloud and receives generated assets without dedicated hardware. This approach is significantly more accessible and can run on a standard laptop, although each generation incurs a small cost (typically $0.02–$0.07 depending on the model and version). Depending on the available computing resources and budget constraints, either configuration can be used to support text-based or image-based 3D asset generation within the pipeline.

This module is not yet fully developed due to the complexity of integrating Image-to-3D models into the plug-in, but this will be completed in future work. To validate the workflow concept, Figure 7 shows the 3D models generated by SAM 3D through raw construction site images.

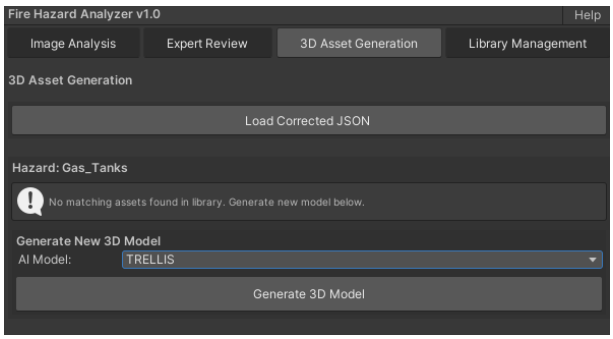

Figure 6. 3D Asset Generation Module UI

Figure 7. Example of a 3D construction object generation from raw site images

4.4 Library Management Module

This module aims to display all existing prefabs within the Unity project alongside the fire-hazard scenarios generated or revised in the updated JSON file. As shown in Figure 8, it functions as a central repository, allowing users to review previously identified hazards and track how scenarios have evolved over time. By maintaining a structured record of hazards and their associated assets, the module supports efficient development of VR environments that better reflect real site conditions. Due to current limitations in the 3D asset-generation model, this component is not yet fully implemented. Its underlying functions will be further developed and validated in future work.

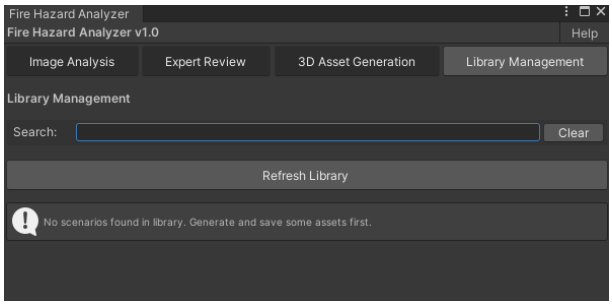

Figure 8. Library Management Module UI

5 Conclusion and Future Work

This study makes three main contributions. First, it introduces a hazard-focused twinning concept that updates only fire hazard–relevant elements in VR, rather than mirroring the entire construction environment. This provides an alternative to full digital twins for training use cases where task-relevant ecological validity is more important than exhaustive geometric fidelity. Second, it proposes a novel, semi-automated pipeline connecting real construction data, VLM reasoning, and procedural 3D objects generation. By integrating VLM analysis with a human-in-the-loop verification step, the framework not only automates the creation of context-aware fire hazard scenarios but also systematically digitizes the tacit knowledge of senior safety experts. Third, it presents an initial Unity-based prototype that illustrates how the workflow can be implemented in practice and used to generate fire hazard-relevant models from real construction imagery.

Despite these contributions, several limitations remain. The system's performance is sensitive to the quality and perspective of input imagery. Occluded objects or poor lighting can hinder VLM semantic extraction. Additionally, while the framework extracts spatial context from 2D images, it currently lacks precise depth estimation. In some cases, objects may appear closer or farther away than they actually are in the original scene. Furthermore, while the feedback loop for model fine-tuning is theoretically defined, the long-term impact of expert corrections on model performance has not yet been empirically validated in this study. Finally, the seamless integration of different models within Unity still requires additional development efforts, and systematic quantitative evaluation of hazard identification performance (e.g., precision and recall metrics) is beyond the scope of this proof-of-concept study and is identified as a priority for future empirical validation.

Future research includes strengthening the core components of the proposed workflow. First, improving depth reasoning from 2D inputs, whether through lightweight monocular depth models [28] or multi-view image capture, will help reduce spatial ambiguities during VLM interpretation. Additional effort is also needed to formally evaluate the feedback loop by measuring how expert corrections affect model accuracy across successive fine-tuning cycles. Such an assessment would clarify whether the system can reliably accumulate and operationalize tacit safety knowledge over time. Finally, the framework's scope could be broadened beyond fire safety to address other critical challenges, such as fall protection and visual clutter analysis, thereby validating the generalizability of VLM-driven simulation across the construction safety domain.

Acknowledgement

This research has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) via the Discovery Grant program (Grant No. RGPIN-2023-04239). The authors gratefully acknowledge NSERC funds, which have facilitated the development of this research.

References

[1] S. Zhang, K. Sulankivi, M. Kiviniemi, I. Romo, C. M. Eastman, and J. Teizer. BIM-based fall hazard identification and prevention in construction safety planning. *Safety Science*. 72:31–45, 2015.

[2] S. S. Man, H. Wen, and B. C. L. So. Are virtual reality applications effective for construction safety training and education? A systematic review and meta-analysis. *Journal of Safety Research*. 88:230–243, 2024.

[3] X. Li, W. Yi, H.-L. Chi, X. Wang, and A. P. C. Chan. A critical review of virtual and augmented reality (VR/AR) applications in construction safety. *Automation in Construction*. 86:150–162, 2018.

[4] V. Getuli, P. Capone, A. Bruttini, and S. Isaac. BIM-based immersive Virtual Reality for construction workspace planning: A safety-oriented approach. *Automation in Construction*. 114:103160, 2020.

[5] M. Namian, A. Albert, and J. Feng. Effect of Distraction on Hazard Recognition and Safety Risk Perception. *Journal of Construction Engineering and Management*. 144(4)2018.

[6] I. S. Abotaleb *et al.* A framework to integrate virtual reality into international standard safety trainings. *Engineering, Construction and Architectural Management*. 32(4):2320–2341, 2025.

[7] Statistics Canada. Labour shortage trends in Canada. Accessed: Jan. 01, 2026. Online: https://www.statcan.gc.ca/en/subjects-start/labour_/labour-shortage-trends-canada

[8] Claire McAnaw Gallagher. The Construction Industry: Characteristics of the Employed, 2003–2020., U.S. Bureau of Labor Statistics. Accessed: Jan. 01, 2026. Online: https://www.bls.gov/spotlight/2022/the-construction-industry-labor-force-2003-to-2020/home.htm

[9] European Labour Authority. Labour shortages and surpluses in Europe 2024. Accessed: Jan. 01, 2026. Online: https://www.ela.europa.eu/en/publications/labour-shortages-and-surpluses-europe-2024

[10] V. V. Tuhaise, J. H. M. Tah, and F. H. Abanda. Technologies for digital twin applications in construction. *Automation in Construction*. 152:104931, 2023.

[11] N. Akindele, R. Taiwo, H. Sarvari, B. I. Oluleye, I. A. Awodele, and T. O. Olaniran. A state-of-the-art analysis of virtual reality applications in construction health and safety. *Results in Engineering*. 23:102382, 2024.

[12] Y. Zhang, H. Liu, S.-C. Kang, and M. Al-Hussein. Virtual reality applications for the built environment: Research trends and opportunities. *Automation in Construction*. 118:103311, 2020.

[13] M. Johansson and M. Roupé. Real-world applications of BIM and immersive VR in construction. *Automation in Construction*. 158:105233, 2024.

[14] S. Alizadehsalehi, A. Hadavi, and J. C. Huang. From BIM to extended reality in AEC industry. *Automation in Construction*. 116:103254, 2020.

[15] V. Getuli, P. Capone, A. Bruttini, and S. Isaac. BIM-based immersive Virtual Reality for construction workspace planning: A safety-oriented approach. *Automation in Construction*. 114:103160, 2020.

[16] J. Xiang *et al.* Structured 3D Latents for Scalable and Versatile 3D Generation. 2025.

[17] SAM 3D Team *et al.* SAM 3D: 3Dfy Anything in Images. 2025.

[18] M. Adil, G. Lee, V. A. Gonzalez, and Q. Mei. Using Vision Language Models for Safety Hazard Identification in Construction., 2025.

[19] H. L. C. W. Q. L. Y. J. Liu. Improved Baselines with Visual Instruction Tuning., in *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, :26296–26306.

[20] Gemma Team *et al.* Gemma 3 Technical Report. 2025. Online: http://arxiv.org/abs/2503.19786

[21] OpenAI *et al.* GPT-4 Technical Report. 2024. Online: http://arxiv.org/abs/2303.08774

[22] X. Chen and Z. Zou. Are Large Pre-trained Vision Language Models Effective Construction Safety Inspectors? 2025. Online: http://arxiv.org/abs/2508.11011

[23] Y. Wang, H. Luo, and W. Fang. An integrated approach for automatic safety inspection in construction: Domain knowledge with multimodal large language model. *Advanced Engineering Informatics*. 652025.

[24] Q. Ma, W. Peng, C. Yang, H. Shen, K. Koedinger, and T. Wu. What Should We Engineer in Prompts? Training Humans in Requirement-Driven LLM Use. *ACM Transactions on Computer-Human Interaction*. 32(4):1–27, 2025.

[25] E. Saravia. Prompt Engineering Guide. *https://github.com/dair-ai/Prompt-Engineering-Guide*. 2022.

[26] J. Wei *et al.* Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. 2023. Online: http://arxiv.org/abs/2201.11903

[27] E. Mosqueira-Rey, E. Hernández-Pereira, D. Alonso-Ríos, J. Bobes-Bascarán, and Á. Fernández-Leal. Human-in-the-loop machine learning: a state of the art. *Artificial Intelligence Review*. 56(4):3005–3054, 2023.

[28] D. Meng *et al.* Conditional DETR for Fast Training Convergence. in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Oct. 2021, :3631–3640.