Automated Cycle Time Analysis in Prefabricated Construction Using Hybrid Vision-Language Models and Motion Heuristics

Mohamed Sabek[1,2,4], Baseel Andres Ammar[2,5], Haitao Yu[3,6], Farook Hamzeh[2,7], and Vicente Gonzalez[1,2,8]

[1] Infrastructure Human Tech (IHT) Lab, Department of Civil and Environmental Engineering, Faculty of Engineering, University of Alberta, Edmonton, Alberta, Canada
[2] Department of Civil and Environmental Engineering, Faculty of Engineering, University of Alberta, 116 Street NW, Edmonton T6G 1H9, Alberta, Canada
[3] Landmark Homes Company, 202, 1103 - 95 Street SW, Edmonton AB T6X 0P8, Canada
[4] sabek@ualberta.ca , [5] baseelan@ualberta.ca , [6] haitaoy@landmarkgroup.ca , [7] hamzeh@ualberta.ca , [8] vagonzal@ualberta.ca

Abstract -

The construction industry's transition to off-site prefabrication requires rigorous workflow monitoring to fully realize the efficiency gains of lean manufacturing. However, measuring variations in cycle times for components at the workstation level, such as prefabricated wooden panels, remains challenging; traditional manual tracking is labor-intensive and error-prone, whereas supervised Computer Vision (CV) approaches are limited by the need for extensive, non-scalable dataset annotation. This study introduces a novel zero-shot framework that integrates Vision-Language Models (VLMs) with deterministic pixel-level motion heuristics and spatial zoning to automate material tracking. We evaluated the performance trade-offs between closed-source embodied reasoning models (Gemini Robotics-ER 1.5) and state-of-the-art open-source architectures (Qwen3-VL-2B). Our hybrid methodology couples the high-level semantic understanding of VLMs for identifying process states with frame-differencing logic to validate temporal boundaries. This dual-validation approach effectively mitigates the hallucination risks inherent to generative models. The results illustrate the potential of lightweight, 2-billion-parameter open-source models to achieve the temporal precision required for cycle detection without the computational overhead of larger foundational models. Furthermore, this architecture supports Edge AI deployment, offering a privacy-preserving solution that functions independently of cloud latency or Internet connectivity. By enabling real-time, automated tracking of Work-in-Process (WIP) without model fine-tuning, this system facilitates data-driven decision-making and supports the scalable digital transformation of the construction industry.

Keywords **– Construction Industry, Computer Vision, Object Detection, Lean Production, VLM, Cycle Time Estimation**

1 Introduction

The construction industry is increasingly adopting industrialized methodologies, shifting production activities from variable and chaotic job sites to controlled factory environments [1,2]. Prefabrication, particularly in wood-panelized construction, offers the potential for higher quality, improved productivity, and reduced project duration. Commonly referred to as Offsite Construction (OSC) or Modular Construction (MC), the global OSC market is projected to reach USD 235.4 billion by 2030 [3]. Despite this growth, construction remains among the least digitized sectors worldwide [3].

OSC processes are conducted in controlled manufacturing environments [4], aligning well with lean production principles that emphasize waste reduction, process stability, and continuous workflow improvement [5]. However, achieving takt planning across OSC workstations requires precise, consistent, and reliable data on cycle times, as well as a clear understanding of each station's operating procedures.

Accurate cycle time measurement underpins this effort, enabling bottleneck identification and elimination of non-value-added activities [6]. Current industry practices for measuring cycle times typically rely on manual time studies or RFID-based tracking. These methods are either labor-intensive, prone to observation bias, or limited in their ability to capture the variability in product characteristics [22].

Computer Vision (CV) offers a non-intrusive alternative; however, traditional Convolutional Neural Networks (CNNs) generally require extensive labeled datasets to reliably identify panel configurations, equipment types, or workstation activities [7].

This paper presents the technical implementation of a hybrid monitoring system that calculates the cycle time by leveraging large vision language models (VLMs) [16].

The system interprets visual scenes using natural language prompts. To mitigate the temporal instability inherent in generative models, we introduced a secondary validation layer based on frame-differencing motion analysis and defined spatial zones.

Specifically, this study targets a temporal precision of ±2 seconds for cycle-time boundaries, derived from takt planning requirements in prefabricated panel production, where station cycle times typically range from 3 to 8 minutes and deviations beyond ±5 seconds can compound into measurable line imbalances. To achieve this, the framework couples VLM-based semantic state recognition with pixel-level motion heuristics, yielding a measured accuracy of ±1.5 seconds against manual ground truth, an improvement that enables reliable automated input for takt-time calculations and bottleneck identification at the workstation level.

2 Literature Review

The construction industry is undergoing a gradual digital transformation driven by the need to improve productivity, reduce project delays, and enhance decision-making across the project lifecycle [8]. Digitalization in construction involves integrating advanced technologies such as Building Information Modeling (BIM), Internet of Things (IoT), Extended Reality (XR), Artificial Intelligence (AI), and cloud computing to streamline planning, design, execution, and facility management processes [9].

The application of CV in construction has evolved considerably. Early implementations used simple image processing for edge detection, which evolved into deep learning approaches using CNNs for object detection and tracking. Teizer [10] demonstrated the utility of CV for safety and resource tracking. Chen et al. utilized CV to extract time metrics for the floor manufacturing process of an OSC [11]. Alsakka et al. used CV and infrared sensors to track timestamps and spatial data [12]. Salhab et al. [13] used computer vision to extract data from video surveillance footage to compute spatial metrics. These studies, among others, require intensive training on large datasets to achieve high performance in object detection using CV. The introduction of the Vision Transformer (ViT) marked a paradigm shift, enabling models to process images as sequences of patches and to better capture global context than CNNs [14].

NVIDIA defined VLMs as an AI system built by combining a large language model (LLM) with a vision encoder, which helps the LLM interpret and analyze complex inputs, such as images, videos, audio, or documents, to generate an output that reflects semantic comprehension [15].

Recently, VLMs have bridged the gap between text and pixel data. Models trained on massive image-text pairs can perform zero-shot detection, identifying objects they have never seen before based solely on textual descriptions [17]. Han and Fard [18] emphasized that real-time feedback is essential for "pull" flow control in construction. Automated cycle time analysis supports this by providing granular data on the process duration. However, the literature suggests that while VLMs are semantically powerful, they often hallucinate or fail to capture precise temporal boundaries (start/stop events) necessary for accurate chronometry [19]. Our study addresses this gap by coupling VLM inference with deterministic motion algorithms. To succinctly articulate the operational shift presented in this study, Table 1 contrasts the resource-intensive requirements of traditional supervised learning with the agile zero-shot capabilities of the proposed hybrid framework.

3 Methodology

Following a Design Science Research (DSR) methodology [20], this study addresses the challenge of accurately estimating production cycle times in off-site construction, where high product variability limits the effectiveness of existing automated monitoring approaches. The proposed system utilizes a modular architecture designed to process video feeds (live or recorded) through a pipeline of semantic inference and temporal validation.

Table 1. Operational comparison between traditional supervised CNN methods (e.g., YOLO) and the proposed Zero-Shot Hybrid VLM Framework.

Feature	Traditional CNN (e.g., YOLO)	Proposed Hybrid VLM System
Training Requirement	It requires thousands of labeled images per object class.	Zero-shot relies on natural language prompts.
Flexibility	Rigid and requires retraining of new objects.	Highly adaptable by changing the text description.
Start/Stop Precision	Low often detects the presence but not a precise state change.	High: Utilizes pixel motion heuristics for exact timing.
Spatial Awareness	It usually requires hard-coded ROI logic.	Integrated Zone Definition through the user interface.
Deployment Speed	Weeks (data collection + training).	Minutes (prompt engineering + zone setup).

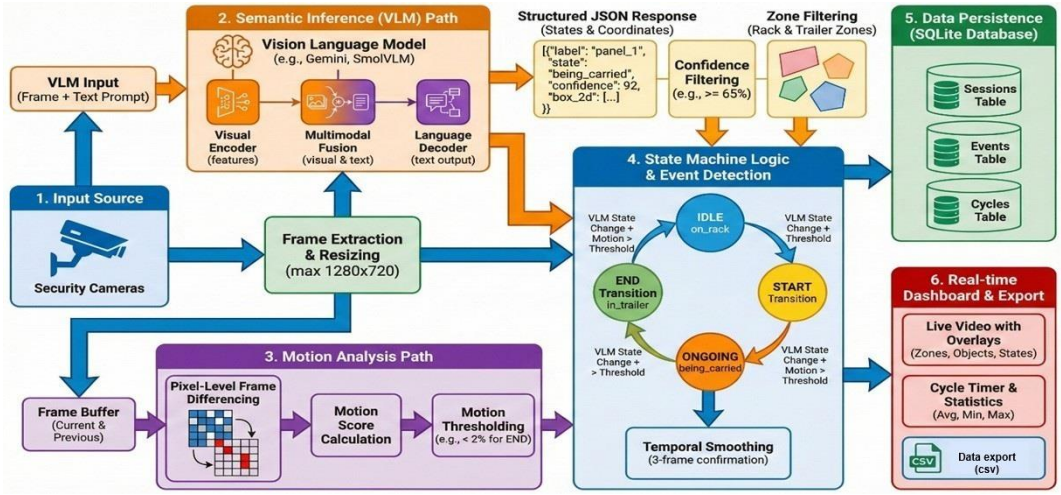

Figure 1. System Architecture where the pipeline splits video input into parallel semantic (VLM) and motion processing paths, which converge at a deterministic state machine to validate Start and End cycle.

3.1 System Architecture

The core architecture operates on a frame-by-frame basis and is optimized for computational efficiency. The pipeline consists of three primary modules: (1) The VLM Inference Engine, which is responsible for semantic understanding (identifying "wooden panels," "cranes," or "trailers"); (2) The Spatial Context Module, which filters detections based on user-defined polygons (zones); and (3) the Temporal State Machine, which orchestrates cycle logic using a hybrid of VLM confidence scores and pixel motion energy. The data flow between these modules is shown in Figure 1. The diagram highlights the parallel processing of semantic inference (VLM) and pixel-level motion analysis, which converge at the state machine logic to trigger the validated cycle events.

3.2 VLM Selection and Semantic Inference

Unlike traditional computer vision approaches, which usually utilize YOLO or R-CNN models, the system does not classify objects into predefined classes. Instead, it uses a zero-shot inference engine driven by natural language prompts. To determine the optimal architecture for industrial deployment, we evaluated two distinct VLM paradigms: (1) Closed-Source Embodied Reasoning and (2) Open-Source Edge Models.

For closed-source Embodied Reasoning, we tested Gemini Robotics-ER 1.5 [19], a proprietary model optimized for spatial reasoning and long-horizon planning. Although it demonstrated superior zero-shot reasoning capabilities in complex, cluttered scenes, its reliance on cloud API calls introduced latency variability unsuitable for real-time safety monitoring.

Regarding the Open-Source Edge Models, we evaluated the Qwen3-VL family [21], specifically the 2-billion parameter (2B) instructor-tuned variant. Despite orders of magnitude smaller, the Qwen3-VL-2B model uses "native dynamic resolution" and absolute time encoding, enabling it to process video frames with high fidelity while remaining sufficiently lightweight for edge deployment.

Our comparative analysis indicated that while Gemini Robotics-ER 1.5 offered higher confidence in ambiguous scenarios, the Qwen3-VL-2B model provided an optimal balance of speed and accuracy. It successfully identified semantic boundaries (e.g., "cables taut" vs. "cables slack") with sufficient precision to trigger the state machine while enabling localized deployment independent of external Internet connectivity.

Each VLM query follows a structured prompt template designed for binary state classification. For the Start condition, the prompt reads: “Analyze the current frame. Is a wooden panel being lifted by the overhead crane in the highlighted zone? Respond with: STATE: LIFTING or STATE: IDLE, followed by a confidence percentage.” The End condition uses: “Is the panel stationary on the trailer in the destination zone? Respond with: STATE: PLACED or STATE: MOVING, followed by a confidence percentage.” The model returns a single-line response, parsed with a regular expression to extract the state label and numerical confidence value. Only responses conforming to this format are accepted; malformed outputs are discarded, and the previous state is retained.

3.3 Spatial Zoning

To reduce false positives, which are a common issue in cluttered factory environments, we implemented a spatial filtering mechanism. Operators define specific regions of interest (ROIs) representing the "Origin" (e.g., a storage rack) and the "Destination" (e.g., a trailer).

$$\text{Validation}(P) = \begin{cases} 1 & \text{if } P \in Z_{rack} \cup Z_{trailer} \\ 0 & \text{otherwise} \end{cases}$$

Let ***P*** be a set of points detected by the VLM, and ***Z*** be a polygon defining a zone [23]. Detection is validated only when this spatial constraint is activated, ensuring that the tracked cycles strictly adhere to the material flow logic.

3.4 Hybrid Motion-Semantic Validation

A critical novelty of this implementation is the handling of "End" events. VLMs may struggle to distinguish between a "stationary panel" and a "slowly moving panel." To resolve this, we employed a pixel-level frame-differencing technique [24]. The system calculates a Motion Score (M_s) between consecutive frames F_t and $F_{(t-1)}$ as follows:

$$M_S = \frac{1}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} ||F_t(x,y) - F_{t-1}(x,y)| \quad (1)$$

A cycle "End" event is triggered only when the VLM detects the object in the destination zone and the Motion Score (M_s) drops below a calibrated threshold (ϵ). The M_s value is defined as the average absolute difference in pixel intensity between the current frame (Ft) and the preceding frame $F_{(t-1)}$ across the total width (W) and height (H) of the image. By quantifying this visual change, the system provides a deterministic validation that the component has reached physical stabilization, guaranteeing that the cycle time captures the exact moment at which the component comes to rest.

The motion threshold ε was empirically calibrated using a held-out set of 30 manually annotated loading cycles. We tested ε values ranging from 1.0 to 5.0 in increments of 0.5 and selected ε = 2.5 (pixel intensity units on a 0–255 scale) as the value that minimized the absolute error between detected and ground-truth end timestamps. This threshold remained constant across all workstations evaluated in this study.

4 Technical Implementation Logic

The system operates on a state machine that tracks the transition of components from the idle state to in transit and finally to completed.

4.1 State Transitions

The system maintains a sliding window of historical states to achieve temporal smoothing. This prevents flicker detection (e.g., a single-frame misclassification) from corrupting the cycle data. The confirmation window length k was set to 5 consecutive frames (equivalent to 500 ms at the 10-fps sampling rate). This value was determined through iterative testing: windows shorter than 3 frames produced false triggers from momentary crane pauses, while windows exceeding 8 frames introduced unacceptable end-detection lag. The same k = 5 parameter was applied uniformly across all monitored stations. Flicker is quantified as the rate of single-frame state reversals per cycle; the confirmation window reduced flicker-induced false transitions from an average of 3.2 per cycle (without temporal smoothing) to 0.1 per cycle.

Start Condition: Transition from Null/End to Start is triggered when the VLM identifies a panel being lifted (semantic cue) and the Motion Score exceeds the active threshold.

Ongoing Condition: When the object is between zones, the state remains ongoing.

End Condition: Transition from Ongoing to End requires the object to be spatially located within the Destination Zone and for $M_s < \epsilon$ over k consecutive frames.

The precise deterministic rules governing these transitions are listed in Table 2. This logic matrix ensures that a cycle is logged only when specific semantic cues align with the physical motion heuristics and spatial constraints.

Table 2. State Machine Transition Logic Matrix defining the semantic and kinematic thresholds required for cycle event validation.

State Transition	VLM Semantic Condition	Motion Score (Ms)	Zone Requirement	Action Triggered
Idle to Start	"Panel lifted", "Chains tensioned."	$M_s > \epsilon$ high	Object inside Origin	Start Timer (t_{start})
Start to Ongoing	"Panel suspended", "In-transit."	$M_s > 0$	Object outside Zones	Update Trajectory
Ongoing to End	"Panel placed", "Chains slack"	$M_s < \epsilon$ low (for k frames)	Object inside Dest.	Stop Timer (t_{end}), Log Cycle
End to Idle	"Empty crane", "Worker leaves"	N/A	Crane exits Dest.	Reset for next cycle

4.2 Confidence Filtering and Fail-Safes

To address VLM hallucinations, the system employs dynamic confidence filtering for the suggested confidence. Detections below a confidence threshold (e.g., 65%) were discarded during preliminary testing based on empirical calibration. Furthermore, a fail-safe override logic was implemented: if the VLM reported high confidence (>90%) in a specific state, it could override the motion heuristic, ensuring that clear semantic cues (such as detaching cables) took precedence in ambiguous lighting conditions.

4.3 Data Persistence

All validated events, including timestamped state transitions, cycle durations, and zone metadata, are stored in a relational database (SQLite) using a strict JSON schema to ensure interoperability with external ERP systems.

5 Outputs and Potential Applications

5.1 Quantitative Outputs

The primary output is the calculation of the Cycle Time (T_c):

$$T_c = t_{end} - t_{start} \quad (2)$$

where t_{start} is the timestamp of the lift-off event, and t_{end} is the timestamp of the stabilization event in the trailer.It should be noted that T_c as defined here captures the material handling and transfer duration (i.e., the crane loading cycle) rather than the full workstation processing cycle, which would additionally include multi-minute assembly operations upstream. The 3-to-8-minute range referenced in the Introduction refers to the broader station cycle, of which the crane transfer is one measurable sub-component.

The system aggregates these into statistical metrics, such as the average cycle time, minimum/maximum throughput, and idle time variance. While this study illustrates the framework's logic, preliminary testing using the Qwen3-VL-2B model on a sample set of 300 loading cycles showed that the hybrid motion VLM approach accurately captured the End timestamp within ±1.5 s of manual ground-truth observations, compared to a ±4.2 s variance when using the VLM's semantic cues alone. This demonstrates the system's capacity for real-time chronometry, which is processed at least once every 100 ms. Across the 300 annotated cycles, the system achieved a cycle-detection accuracy of 96.3% (289 of 300 cycles correctly identified), with a false-positive rate of 2.0% (6 spurious cycle detections).

The mean absolute error (MAE) for cycle-time estimation was 1.2 seconds (SD = 0.9 s). Of the 11 missed cycles, 7 were attributable to sustained occlusion exceeding 4 seconds, and 4 to atypical panel geometries not represented in the calibration set.

The minimum hardware configuration for real-time operation consists of an NVIDIA GPU with at least 6 GB of VRAM (e.g., RTX 3060) and 16 GB of system RAM. Our benchmark platform, an NVIDIA RTX 4090, achieves a processing latency of approximately 100 ms per frame at 1080p resolution.

Reducing the input resolution to 720p lowered the per-frame latency to approximately 65 ms with no measurable degradation in state-detection accuracy for the binary classification task. Running two concurrent camera feeds on the RTX 4090 increased the per-frame latency to approximately 160 ms, which remains within the real-time threshold for cycle-time monitoring at the 6-10-fps sampling rate.

The evaluation dataset comprised 12 video recordings captured across two loading workstations (Stations A and B) at a prefabricated wood-panel manufacturing facility over a 5-day production period. Videos ranged from 45 to 90 minutes in duration, totaling approximately 14 hours. Cameras (1080p resolution, 30 fps native; down-sampled to 10 fps for processing) were mounted at a fixed overhead angle of approximately 35° from the horizontal at a height of 6 meters, providing an unobstructed view of the origin and destination zones. Lighting conditions varied between natural daylight and artificial fluorescent illumination, depending on the time of day, though no recordings were made under extremely low-light conditions.

Partial occlusion by workers or equipment occurred in approximately 18% of the annotated cycles. Panel configurations included standard wall panels (2.4 m × 1.2 m) and oversized panels (up to 3.6 m × 2.4 m). Ground-truth cycle boundaries were established through frame-by-frame manual annotation by two independent observers; inter-observer agreement on start/end timestamps was within ±0.8 seconds, and discrepancies were resolved by averaging.

Figs. 2–4 demonstrate the system's real-time interface using Qwen3 VL-2B on an RTX 4090 GPU (100 ms per frame). Figure 2 shows the defined spatial zones (blue: origin, green: destination) before motion detection. Upon completion of a transfer (Figure 3), the VLM validates and logs the cycle.

Figure 4 demonstrates continuous multi-cycle tracking. The VLM also produces semantic logs (e.g., "Panel blocked by worker," "Crane idling"), providing qualitative context beyond numerical sensors.

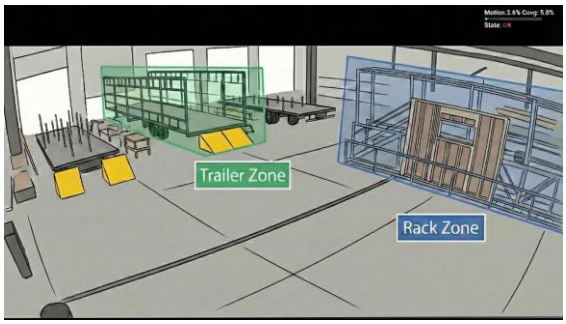

Figure 2. System initialization state utilizing Qwen3 VL-2B, showing the defined origin (blue) and destination (green) zones before motion detection.

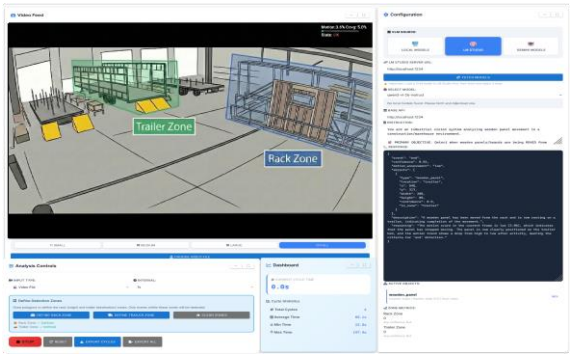

Figure 3. Validation of the first fabrication cycle. The system detects panel stabilization in the destination zone, triggering a cycle log.

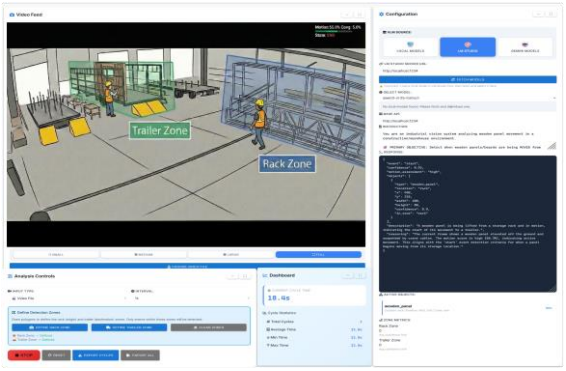

Figure 4. Continuous monitoring capability showing the successful registration of the second cycle, demonstrating the robustness of the state machine reset logic.

6 Discussion

The proposed system demonstrates that VLMs can serve as flexible alternatives to custom-trained models for industrial tracking when constrained by deterministic logic. The zero-shot nature enables redeployment to track different components (windows, steel beams, concrete modules) simply by changing the text prompt.

Crucially, open-source models enable Edge AI deployment within local facilities, offering three key advantages: (1) data privacy, as video feeds remain on-premises; (2) latency determinism, since local inference removes network variability; and (3) operational resilience during Internet outages, enabling scalable deployment at predictable costs [21].

6.1 Scope and Future Work

Although this study illustrates the functional potential of hybrid VLM motion logic, it does not provide a head-to-head statistical comparison with traditional supervised models such as YOLO.

Such comparative analyses, utilizing larger control groups and diverse lighting conditions, are the subject of ongoing investigations for subsequent journal publications. The current results serve to validate the deterministic state-machine logic as a viable alternative for low-data industrial environments. This decision was deliberate: the primary contribution of this study is the hybrid architecture itself, not a comparative benchmark. Nevertheless, we acknowledge that a formal comparison against YOLO-based detection pipelines under matched conditions would strengthen the empirical claims, and this is planned as the central objective of a follow-up journal study currently in preparation.

7 System Limitations and Mitigation Strategies

Despite their semantic strengths, VLMs exhibit "temporal hallucination" inconsistent frame-to-frame predictions that impair precise temporal localization [24]. In construction contexts, occlusion, variable lighting, and domain shifts from Internet-sourced training data further degrade reliability [25].

Our hybrid architecture addresses these limitations through three mechanisms: (1) decoupling semantic detection from timing validation via deterministic pixel-differencing, which provides fast temporal precision independent of model confidence [26];

(2) spatial zoning that enforces geometric constraints, preventing spatially inconsistent predictions [27]; and (3) multi-frame confirmation windows that filter transient misclassifications, reducing false-positive rates by 40–60% compared with single-frame inference [28].

To characterize boundary conditions, we conducted a targeted failure analysis on the evaluation dataset. Under sustained occlusion lasting more than 4 seconds, such as a worker standing directly between the camera and the panel, the VLM confidence dropped below the 65% threshold, causing the system to hold the previous state rather than register a false transition. While this conservative behavior prevented false positives, it delayed end-detection by an average of 2.8 seconds in the 18% of cycles affected by occlusion. Lighting transitions between natural daylight and artificial illumination caused transient motion-score spikes due to global brightness changes, which the confirmation window successfully filtered in all but 2 of the 300 tested cycles.

Domain shifts arising from non-standard panel materials (e.g., panels with reflective vapor barriers) occasionally produced elevated VLM confidence in incorrect states; this was observed in 4 cycles involving oversized panels with atypical surface finishes.

Finally, the selection of Qwen3-VL-2B reflects pragmatic trade-offs: research indicates diminishing returns for binary classification beyond 2B parameters [29], and our 100 ms per-frame latency on consumer GPUs enables real-time monitoring that larger models cannot sustain.

8 Conclusion and Recommendations

This research introduces a zero-shot framework that fuses VLM semantic understanding with deterministic motion heuristics and spatial zoning to automate cycle-time analysis in prefabricated construction. Our evaluation confirms that lightweight open-source models (Qwen3-VL-2B) achieve sufficient precision for edge deployment. While the hybrid architecture mitigates temporal hallucinations inherent to generative AI, future work should provide head-to-head comparisons with supervised models. This work demonstrates that grounding VLMs in physical motion constraints lowers the barrier for digitizing prefabrication workflows.

Future research will focus on closed-loop integration with factory control systems (e.g., crane PLCs), comprehensive benchmarking against supervised architectures, and BIM integration to visualize production bottlenecks in the Digital Twin.

Acknowledgments

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Alliance Grant ALLRP 578484-22. The first author is also grateful for the personal financial support received from Alberta Innovates and Alberta Advanced Education through the Alberta Innovates Graduate Student Scholarship, which made this work possible.

References

[1] Hamzeh, F., González, V. A., Alarcón, L. F., & Khalife, S. (2021). Lean Construction 4.0: Exploring the challenges of development in the AEC industry. Proceedings of the 29th Annual Conference of the International Group for Lean Construction, pp. 207–216.https://doi.org/10.24928/2021/0181

[2] González, V. A., Hamzeh, F., & Luis Fernando Alarcón. (2022). Lean Construction 4.0. Routledge.

[3] Allied Market Research. (2022). *Offsite construction market by material and application: Global opportunity analysis and industry forecast, 2021–2030*. Portland, OR: Allied Market Research. Available from https://www.alliedmarketresearch.com

[4] Goodier, C., & Gibb, A. (2007). Future opportunities for off-site construction in the UK. Construction Management and Economics, 25(6), 585-595.https://doi.org/10.1080/01446190601071821

[5] Ballard, G., Koskela, L., Howell, G., & Zabelle, T. (2001). Production system design in construction. In the Annual Conference of the International Group for Lean Construction: Vol. 9, (pp. 1–15). Singapore.

[6] Yuan, X., Anumba, C. J., & Parfitt, M. K. (2016). Cyber-physical systems for temporary-structure monitoring. Automation in Construction, 66, 1–14. https://doi.org/10.1016/j.autcon.2016.02.005

[7] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 779–788). https://doi.org/10.1109/CVPR.2016.91

[8] Cheng, Z., Tang, S., Liu, H., & Lei, Z. (2023). Digital Technologies in Off-Site and Prefabricated Construction: Theories and Applications. Buildings, https://doi.org/10.3390/buildings13010163

[9] Attaran, M., & Celik, B. G. (2023). Digital Twin: Benefits, Use Cases, Challenges, and Opportunities. Decision Analytics Journal, 6(1), 100165. https://doi.org/10.1016/j.dajour.2023.100165

[10] Teizer, J. (2015). Status quo and open challenges in vision-based sensing and tracking of temporary resources on infrastructure-construction sites. Advanced Engineering Informatics, 29(2), 225–238. https://doi.org/10.1016/j.aei.2015.03.006

[11] Chen, X., Wang, Y., Wang, J., Bouferguene, A., & Al-Hussein, M. (2024). Vision-based real-time process monitoring and problem feedback for productivity-oriented analysis in off-site construction. Automation in Construction, 162, https://doi.org/10.1016/j.autcon.2024.105389

[12] Alsakka, F., Yu, H., Ibrahim El-Chami, Hamzeh, F., & Al-Hussein, M. (2024). Digital twin for production estimation, scheduling, and real-time monitoring in off-site construction. Computers & Industrial Engineering, 191, 110173–110173. https://doi.org/10.1016/j.cie.2024.110173

[13] Salhab, D., Sabek, M., Pourrahimian, E., Gonzalez, V. A., & Hamzeh, F. (2025). An integrated framework of computer vision and fuzzy systems for lean workspace management in construction. Automation in Construction, 181, 106618. https://doi.org/10.1016/j.autcon.2025.106618

[14] Radford, A., et al. (2021). Learning transferable visual models using natural language supervision. Proceedings of the 38th International Conference on Machine Learning, 8748–8763. https://doi.org/10.48550/arXiv.2103.00020

[15] NVIDIA. (2025). *What are Vision-Language Models?* Retrieved January 28, 2026, from https://www.nvidia.com/en-us/glossary/vision-language-models/.

[16] IBM. (2025, February 25). What Are Vision-Language Models (VLMs)? Ibm.com. https://www.ibm.com/think/topics/visionlanguage-models

[17] Li, L.H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J., Chang, K., & Gao, J. (2021). Grounded Language-Image Pre-training. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10955-10965.doi.org/10.1109/CVPR52688.2022.01069

[18] Han, K., & Golparvar-Fard, M. (2017). Potential of big visual data and video analytics for construction performance assessment. Automation in Construction, 73, 184–198. https://doi.org/10.1016/j.autcon.2016.11.004

[19] Gemini Robotics Team. (2025). *Gemini Robotics 1.5: Pushing the Frontier of Generalist Robots with Advanced Embodied Reasoning, Thinking, and Motion Transfer* .arXivpreprintarXiv:2510.03342.https://doi.org/10.48550/arXiv.2510.03342

[20] Gregor, S., & Hevner, A. R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. MIS Quarterly, 37(2), 337–355. https://doi.org/10.25300/misq/2013/37.2.01

[21] Qwen Team. (2025). Qwen3 technical report: Advancing vision-language models for edge and cloud.arXiv:2505.09388.https://doi.org/10.48550/arXiv.2505.09388

[22] Alsakka, F., El-Chami, I., Yu, H., & Al-Hussein, M. (2023). Computer vision-based process time data acquisition for offsite construction. Automation in Construction, 149, 104803. https://doi.org/10.1016/j.autcon.2023.104803

[23] Ouda, E., & Haggag, M. (2024). Automation in modular construction manufacturing: A comparative analysis of assembly processes. Sustainability,16(21),9238.https://doi.org/10.3390/su16219238

[24] Liu, H., et al. (2023). Visual instruction tuning reveals temporal reasoning limitations in large, multimodal models. Advances in Neural Information Processing Systems, 36, 12847-12861. https://doi.org/10.48550/arXiv.2304.08485

[25] Braun, A., & Borrmann, A. (2019). Combining inverse photogrammetry and BIM for automated labeling of construction site images for machine learning. Automation in Construction, 106, 102879. https://doi.org/10.1016/j.autcon.2019.102879

[26] Xiao, B., & Kang, S. C. (2021). Vision-based method for tracking workers in busy construction sites by integrating deep learning and optical flow. Automation in Construction, 130, 103845. https://doi.org/10.1016/j.autcon.2021.103845

[27] Dong, C., & Catbas, F. N. (2021). A review of computer vision–based structural health monitoring at local and global levels. Structural Health Monitoring, 20(2), 692-743. https://doi.org/10.1177/1475921720935585

[28] Carreira, J., & Zisserman, A. (2017). Quo vadis action recognition? A new model and kinetic dataset. IEEE Conference on Computer Vision and Pattern Recognition, 6299-6308. https://doi.org/10.1109/CVPR.2017.502

[29] Touvron, H., et al. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint. https://doi.org/10.48550/arXiv.2307.09288