Interpretable Real-Time Construction Safety Monitoring via Video-Text Retrieval

Sejun Park[1] Jaehoo Kim[2] Dongmin Lee[3]

[123]School of Civil, Environmental, and Architectural Engineering, Korea University, Republic of Korea
sejun0909@korea.ac.kr, wogn0505@korea.ac.kr, leedm@korea.ac.kr

Abstract

Recent construction sites increasingly employ real-time safety monitoring systems that apply rule-based logic to frame-level object detections. However, because such approaches rely on isolated detections and manually defined rules, they provide limited evidence for how workers, PPE, actions, locations, and hazards jointly constitute safety risk, and their predictions often fluctuate over time, leading to false alarms and missed detections. To address these limitations, we propose an interpretable framework for streaming construction safety monitoring based on video–text retrieval, which captures spatiotemporal context and provides explicit textual explanations. The proposed method evaluates short video segments by retrieving relevant evidence from a structured safety text bank and estimating risk from the retrieved results, while also presenting representative evidence sentences for operator verification. Experiments on representative scaffold safety scenarios demonstrated strong risk classification performance, with accuracy of up to 97.6%, while the retrieved evidence consistently outperformed captioning baselines in explanatory quality, achieving top-1 BERTScore-F1 values of 0.925–0.943 and top-1 SPICE values of 0.618–0.727. The system also achieved a mean end-to-end latency of 210.8 ms on a single GPU, supporting real-time operation. These findings indicate that video–text retrieval offers a practical and interpretable approach to spatiotemporally informed construction safety monitoring without site-specific retraining.

Keywords

construction safety, real-time monitoring, interpretability, video–text retrieval

1 Introduction

Computer-vision (CV)-based real-time safety monitoring has become a central approach for detecting hazardous situations in construction sites, where visual data are continuously analysed to identify workers, equipment, and site conditions. However, construction sites remain highly dynamic environments: construction phases, spatial layouts, and equipment placements change frequently, while viewpoint and illumination changes, occlusions, and motion blur caused by worker movements continuously alter the visual conditions under which CV models operate.

Consequently, the visibility and appearance of target entities can vary substantially from frame to frame, causing detection outputs to fluctuate even within the same ongoing situation. This temporal instability in detections often results in intermittent misses and spurious detections over time, i.e., temporal inconsistency. Because frame-wise decision-making treats each frame independently, it cannot effectively use adjacent temporal evidence to stabilize predictions, which in turn increases false positives and false negatives and undermines the reliability of hazard detection[1].

Moreover, construction-site hazards are not determined solely by the presence of a particular object in an isolated frame, but by how multiple risk cues, such as workers, personal protective equipment (PPE) compliance, actions, objects, and locations, interact and evolve in spatiotemporal context[2]. Consequently, reliance on frame-level detections alone can either miss hazardous events or incorrectly classify normal operations as dangerous[3]. These observations motivate moving beyond frame-wise detection toward video-based safety monitoring that incorporates temporal context and supports more reliable interpretation of how risk cues jointly indicate hazardous situations.

In response to these challenges, prior studies have explored several video-based approaches for safety monitoring. Temporal modelling methods, such as RNN/LSTM-based approaches, aggregate information across consecutive frames to reduce prediction fluctuations and improve temporal consistency [4], while video anomaly detection methods learn normal patterns from videos and assign anomaly scores to clips that deviate from them[5]. However, both approaches mainly produce scores, labels, or detections, providing limited explicit evidence about why a situation is risky or how specific risk cues contribute to the assessment. Video

captioning methods improve interpretability by describing scenes in natural language[6]. However, generated descriptions can vary across similar scenes and may omit or overstate critical evidence, while repeated clip-wise decoding also increases computational cost and end-to-end latency in real-time deployment.

Thus, existing approaches remain limited in two practical respects for construction safety monitoring: explicit interpretability and real-time suitability. To address these limitations, we propose a framework for real-time construction safety monitoring that constructs a structured text bank encoding relationships among risk cues from safety regulations and real-world site data, and applies video–text retrieval to provide interpretable evidence for streaming inputs.

2 Related Work

2.1 Computer Vision-based construction safety monitoring

Construction safety monitoring has evolved toward detecting workers, equipment, and site conditions from vision data (e.g., CCTV/UAV videos) and issuing alarms by rule-based assessment of PPE compliance or hazardous states. Recently, pipelines have been proposed that first detect objects using real-time single-stage detectors and then integrate knowledge-based (ontology-driven) reasoning as post-processing to determine violations or hazardous situations[7]. In addition, PPE monitoring has progressed beyond a binary "worn/not worn" decision, with attempts to combine instance segmentation and tracking to associate PPE with individual workers and to refine compliance states at a finer granularity[8].

Nevertheless, frame-level, detection-based risk judgment is highly susceptible to occlusions, viewpoint/illumination changes, and motion blur, causing predictions to fluctuate and making errors accumulate as false positives and false negatives over time. To mitigate this, prior work has proposed aggregating decisions over a temporal window to reduce false alarms, or enhancing spatial reasoning with depth information [1][9]. However, such temporal and spatial augmentations primarily aim to stabilize alarm outputs, and they remain limited in explaining why a scene is judged as hazardous—i.e., which relationships among PPE, actions, locations, and hazard factors constitute the basis for the decision.

2.2 Unsupervised/Self-supervised Video Anomaly Detection (VAD)

To overcome the limitations of frame-based decision-making, a growing body of work has explored applying unsupervised/self-supervised video anomaly detection (VAD) to safety monitoring. In general, VAD learns normal patterns from videos and then assigns anomaly scores or detects abnormal segments by identifying sequences that deviate from the learned normal distribution. In construction safety monitoring, a study combined unsupervised GAN-based learning with pseudo-anomaly synthesis and introduced the WeTeam22 dataset to detect unsafe worker behaviors[10].

However, because VAD typically outputs anomaly scores or abnormal intervals, it is difficult to directly translate the results into concrete risk cues that operators can readily understand and act upon. Recently, benchmarks have been proposed that encourage models to go beyond mere anomaly detection and explain what happened, why it occurred, and what evidence supports the risk judgment[11]. This trend suggests that producing such explanations generally requires additional annotations and/or a separate language reasoning module. Moreover, construction sites exhibit substantial distribution shifts due to changes in phases, spatial configurations, and camera conditions, which induces drift in the normal-data distribution and imposes considerable burdens in collecting, curating, maintaining, and re-adapting normal data for unsupervised learning.

2.3 Video Captioning-based construction safety monitoring

Video captioning aims to complement situation understanding by describing video scenes in natural language. Vision-language models (VLMs), such as Qwen2.5, SmolVLM, and InternVL, process video or multi-frame inputs to perform clip-level scene understanding and generate textual descriptions accordingly[12][13][14]. In another high-risk domain, transportation, research has continued toward explainable video understanding—for example, domain-specific captioning approaches such as TrafficVLM, which describe incidents in a stepwise manner and incorporate output control mechanisms[15].

In contrast, within the construction domain, captioning has been used primarily for recording and documentation, while it has received limited attention as a real-time interpretation mechanism for safety monitoring[16]. This is partly because scene distributions vary substantially across sites, making it difficult to build training data, and because accurately reflecting regulatory semantics—such as PPE compliance or proximity to hazardous zones—often requires vocabulary standardization. Moreover, when deployed in streaming settings, clip-wise text decoding must be repeatedly executed, which can increase computational cost and end-to-end latency.

Overall, prior approaches have advanced either

toward stabilizing alarms or toward providing natural-language descriptions, but mechanisms that consistently present explanations of streaming scenes while accounting for temporal context remain limited. To address this gap, we propose a low-latency framework that combines a structured text bank encoding relationships among key risk factors in construction site with video–text retrieval, thereby achieving both temporal consistency and interpretability in real-time monitoring.

3 Method

This study proposes a real-time monitoring framework that interprets construction-site videos via similarity-based retrieval against a text bank and simultaneously provides a risk score and evidence-grounded explanations. After aligning text and video in a shared embedding space, the framework retrieves text entries most similar to each segment embedding to form an evidence set, and then aggregates unsafe evidence to compute a risk score. Figure 1 illustrates the overall workflow of the proposed framework. The workflow begins with constructing a structured text bank from safety knowledge and indexing its text embeddings in the joint embedding space. During real-time inference, the input video stream is segmented, each segment is encoded into a video embedding, and top-K similar text entries are retrieved by similarity search. The retrieved texts are then used to compute the risk score RS(t), and the results are rendered as real-time overlay outputs.

3.1 Text Bank Construction in a Joint Embedding Space

To provide evidence for risk assessment in terms of relationships among key risk cues, we construct a text bank that describes the work subject, PPE state, action, location, and hazard factor. Safety risk is not determined solely by the presence of a single object; rather, it emerges from combinations of who (SUBJECT) performs what action (ACTION) under which protective condition (SUBJECT_STATE), at which location (LOCATION), and with which associated hazard factor (RISK). This formulation is aligned with semantic scene representations used in computer vision and multimodal scene understanding, while explicitly extending them to construction-safety attributes such as PPE condition and hazard relevance[17]. In construction safety, PPE compliance and work location are particularly decisive determinants of risk, and prior studies have emphasized the importance of entity interactions and contextual information[18]. Accordingly, texts structured around subject, PPE state, action, location, and risk provide an appropriate basis for composing evidence cues.

Each text entry c_i is encoded by the text encoder $f_T(\cdot)$ of a pretrained video–text dual encoder, producing a text embedding $e_T(i) \in \mathbb{R}^d$, which is stored in the bank. The collection of these embeddings forms a joint embedding space $\mathbb{R}^d$ that enables similarity-based interpretation of streaming scenes. To enable low-latency retrieval, all text embeddings are pre-indexed in an offline stage, allowing efficient text search under real-time monitoring constraints.

3.2 Streaming Video Segmentation

In our framework, the input stream is processed by grouping frames into fixed-length segments. Because a segment comprises consecutive frames over a short time interval rather than a single frame, it is less sensitive to instantaneous noise and can capture both motion dynamics and scene cues[1].

Concretely, we partition the input stream into segments V_t of duration t seconds and apply a sliding window with stride S seconds to generate temporally

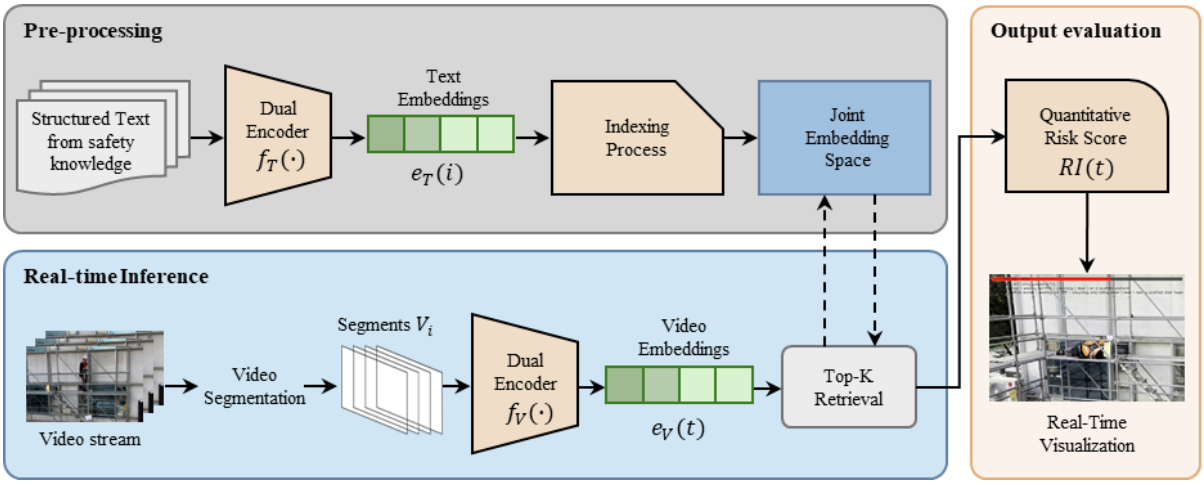

Figure 1. Research framework

overlapping segments along the timeline. All subsequent steps—embedding extraction, evidence retrieval, risk score computation, and visualization—are performed at the segment level.

3.3 Video Embedding Extraction

Each segment V_t is encoded into a video embedding $e_V(t) \in \mathbb{R}^d$ using the video encoder $f_V(\cdot)$ of the same dual-encoder model. Given the consecutive frames that constitute the segment, the video encoder summarizes the scene information over the corresponding time interval into a single fixed-length vector. The resulting video embedding is represented in the same joint embedding space $\mathbb{R}^d$ as the text embeddings and is subsequently used to select evidence texts via similarity-based retrieval.

3.4 Similarity-Based Retrieval of Evidence Texts

Given the video embedding $e_V(t)$ at time t, we retrieve the top-K texts from the text-bank embedding set $\{e_T(i)\}$ with the highest cosine similarity to form an evidence set $E_K(t)$. The video–text matching score is defined as the cosine similarity $\mathrm{sim}(\cdot)$, computed as the inner product after applying ℓ_2-normalization to all embeddings:

$$E_K(t) = \mathrm{TopK}_i(\mathrm{sim}(e_V(t), e_T(i))). \quad (1)$$

Here, $E_K(t)$ denotes the candidate set of evidence texts that describe the scene observed at time t.

3.5 Evidence-Grounded Risk Score

Based on the retrieved evidence set $E_K(t)$, we compute a segment-level Risk Score $RS(t)$. Our formulation relies on two assumptions: (i) in hazardous scenes, the proportion of unsafe evidence among the retrieved texts increases, and (ii) evidence with higher retrieval scores (similarities) provides more direct explanatory power for risk assessment.

Each text c is assigned a binary label $u(c) \in \{0,1\}$ indicating whether it constitutes unsafe evidence. In this study, texts that contain explicit worker-centered risk cues—such as missing PPE and proximity to missing guardrails/openings—are labeled as $u(c) = 1$ (unsafe), whereas texts describing normal working conditions are labeled as $u(c) = 0$ (safe). These labels are assigned consistently using a rule-based procedure grounded in the OSHA guide A Guide to Scaffold Use in the Construction Industry and the AI-Hub real-time video dataset for work-at-height scenarios [19]. For each evidence text $c \in E_K(t)$, its weight $w(c)$ is set to the cosine similarity returned by the retrieval step.

With these definitions, the risk score is computed as the weighted proportion of unsafe evidence among the retrieved top-K texts and normalized to the range $[0, 100]$:

$$RS(t) = 100 \cdot \frac{\sum_{c \in E_K(t)} w(c)\, u(c)}{\sum_{c \in E_K(t)} w(c)} \quad (2)$$

Accordingly, $RS(t)$ increases when a larger number of highly similar evidence texts are labeled as unsafe.

3.6 Real-Time Visualization with Evidence Captions

The proposed framework outputs a segment-level risk score $RS(t)$ sequentially over time for streaming inputs, while simultaneously presenting a subset of the retrieved evidence texts (top-N) as evidence captions. This enables operators to inspect both the current risk level and the textual evidence underlying the risk assessment. For real-time monitoring, the final outputs are continuously rendered as an overlay on the video frames.

4 Experiments and Results

4.1 Safety Text Bank Construction and Indexing

In this study, we built a text bank to interpret real-time monitoring scenes in terms of relationships among key risk factors. Each text is structured into five fields—SUBJECT / SUBJECT_STATE / ACTION / LOCATION / RISK—and the vocabulary for each field was defined by extracting and organizing terms from the OSHA guide and the AI-Hub dataset. The numbers of tokens per field are 3 for SUBJECT, 6 for SUBJECT_STATE, 44 for ACTION, 37 for LOCATION, and 19 for RISK, yielding a total of 151,296 texts through combinatorial generation. In addition, each text entry c_i is assigned a binary label $u(c_i) \in \{0,1\}$ indicating whether it constitutes unsafe evidence.

To compare the text bank and video segments within the same embedding space, we employ a pretrained video–text dual encoder, Perception Encoder (PE-Core). PE-Core maps video and text representations into a shared embedding space via contrastive vision–language alignment learned from large-scale data. In our experiments, the embedding dimension is $d = 1024$[20]. For real-time top-K retrieval over the large-scale text bank, we pre-index the text embeddings offline using a FAISS IVF-PQ (IndexIVFPQ) approximate nearest neighbor (ANN) index.

4.2 Streaming Inference Settings

We partition the real-time streaming video into segments of length $t = 2.0$s and apply a sliding window

with stride $S = 0.5$ s to continuously update the risk signal. For each segment, we uniformly sample 16 frames as input to the video encoder to produce a segment embedding $e_V(t)$. In the online stage, we retrieve $K = 10$ texts to construct $E_K(t)$, and in the visualization stage, we select the top $N = 2$ entries as evidence captions and render them as an overlay on the video stream. All streaming experiments were conducted on a single NVIDIA GeForce RTX 5070 Ti (16GB) GPU with Driver 591.44 and CUDA 13.1.

4.3 Dataset and Ground Truth Annotation

To assess field deployability, we recorded three staged video clips in a scaffold environment at Building 208, Chung-Ang University, Seoul, Republic of Korea. The clips correspond to three hazard scenarios: Vertical (1 min 50 s), referring to vertically overlapping work; Mobile (1 min 11 s), referring to mobile-scaffold movement with a worker riding on the scaffold; and Fall (48 s), referring to falling events. For evaluation, we manually assigned segment-level binary ground-truth labels (unsafe = 1, safe = 0) for all clips. Notably, the unsafe label was defined to cover not only the hazardous moment itself, but also the precursor interval immediately before the event and the recovery interval afterward when applicable. In addition, we manually wrote detailed descriptive captions for each segment, which were used to evaluate the interpretability of the retrieved evidence texts.

For baseline comparison, the same segmented clips were additionally evaluated using three open-source vision-language models: Qwen2.5-VL, SmolVLM2-2.2B, and InternVL2.5-2B. Each model was prompted to generate a one-sentence neutral description of the visible construction scene and to classify the scene as safe or unsafe based only on visible evidence. To ensure a consistent output format across models, we used a fixed instruction requiring the response to contain exactly two fields, caption: and scene_safety:.

4.4 Interpretability of Evidence Captions

Table 1 presents a scenario-wise comparison (Vertical, Mobile, and Fall) of how well the descriptions generated or retrieved by each model support the ground-truth (GT) explanatory captions for each segment. The evaluation was conducted using BERTScore-F_1, which measures sentence-level semantic similarity, and SPICE, which reflects structural semantic alignment in terms of objects, actions, and relations. The baseline models (Intern, Qwen, and Smol) generate a single final caption for each segment; therefore, their performance was evaluated using the top-1 F_1 score of that single output. In contrast, the proposed method (Ours) retrieves a set of evidence texts ranked by similarity, and thus reports not only top-1 performance but also the maximum score within the top- K retrieved set (max F_1@5 and max F_1@10).

Table 1. BERTScore Between Evidence Captions and Ground-Truth Text

Scenario	Model	Metric	Top-1 F_1	Max F_1@5	Max F_1@10
Vertical	Intern	BERT	0.888	-	-
Vertical	Qwen	BERT	0.886	-	-
Vertical	Smol	BERT	0.876	-	-
Vertical	Ours	BERT	0.925	0.938	0.941
Mobile	Intern	BERT	0.888	-	-
Mobile	Qwen	BERT	0.891	-	-
Mobile	Smol	BERT	0.879	-	-
Mobile	Ours	BERT	0.925	0.933	0.934
Fall	Intern	BERT	0.893	-	-
Fall	Qwen	BERT	0.891	-	-
Fall	Smol	BERT	0.890	-	-
Fall	Ours	BERT	0.943	0.948	0.950

Overall, the proposed method achieved the highest caption alignment across all scenarios. In terms of BERTScore-F_1, the top-1 F_1 of Ours was 0.925 for Vertical, 0.925 for Mobile, and 0.943 for Fall, while the corresponding max F_1@5 scores increased to 0.938, 0.933, and 0.948, and the max F_1@10 scores further increased to 0.941, 0.934, and 0.950. In contrast, the top-1 BERTScore-F_1 values of the baseline models remained within 0.876-0.888 for Vertical, 0.879-0.891 for Mobile, and 0.890-0.893 for Fall. This indicates that the evidence retrieved by the proposed method maintains a higher degree of semantic consistency with the GT descriptions.

In Table 2 the difference becomes even more pronounced in SPICE. The top-1 SPICE of Ours was 0.618 for Vertical, 0.647 for Mobile, and 0.727 for Fall, while the corresponding max F_1@5 scores increased to 0.679, 0.720, and 0.767, and the max F_1@10 scores further improved to 0.715, 0.734, and 0.779. In contrast, the top-1 SPICE scores of the baseline models were limited to 0.119-0.193 for Vertical, 0.183-0.225 for Mobile, and 0.144-0.190 for Fall. These results suggest that the baseline models fail to sufficiently capture the objects, actions, and spatial relations that constitute the actual scene.

Table 2. SPICE score Between Evidence Captions and Ground-Truth Text

Scenario	Model	Metric	Top-1 F_1	Max F_1@5	Max F_1@10
Vertical	Intern	SPICE	0.193		
Vertical	Qwen	SPICE	0.128		
Vertical	Smol	SPICE	0.119		
Vertical	Ours	SPICE	0.618	0.679	0.715
Mobile	Intern	SPICE	0.225		
Mobile	Qwen	SPICE	0.183		
Mobile	Smol	SPICE	0.208		
Mobile	Ours	SPICE	0.647	0.720	0.734
Fall	Intern	SPICE	0.190		
Fall	Qwen	SPICE	0.144		
Fall	Smol	SPICE	0.178		
Fall	Ours	SPICE	0.727	0.767	0.779

In other words, although the captions produced by existing generative models may appear linguistically natural, they may still fail to accurately describe the key evidence underlying hazardous situations. Therefore, such single generated captions are difficult to rely on directly as interpretable outputs or as a basis for safety judgment. In contrast, the proposed method not only shows high semantic similarity to the GT descriptions but also achieves a substantial advantage in SPICE, confirming that it provides evidence that more accurately reflects the structural meaning of the scene.

In addition, the proposed method consistently maintained strong performance across all three scenarios: Vertical, Mobile, and Fall. This suggests that its descriptive capability is not limited to a particular scenario type, but can robustly reflect both semantic and structural information across diverse work situations and risk contexts. The substantial margin observed in SPICE across all scenarios further indicates that the proposed method more robustly captures scene-level objects, actions, and spatial relations, and therefore can provide highly interpretable evidence under a wider range of site conditions.

4.5 Risk Score Performance

Risk-score performance was evaluated by applying multiple thresholds (0.1-0.9) to the proposed Risk Score, classifying each segment as unsafe if its score exceeded the given threshold and as safe otherwise, and then comparing the resulting predictions with the segment-level ground-truth labels. Table 3 summarizes the segment-level classification performance under each threshold. As the threshold increased from 0.1 to 0.7, Accuracy steadily improved from 88.5% to 97.6%, while Recall remained at 1.00 throughout this range. F1 also increased consistently from 0.936 to 0.985. In contrast, when the threshold was further increased to 0.8 and 0.9, Accuracy remained high at 97.5% and 96.1%, respectively, but Recall decreased to 0.983 and 0.957, and F_1 was 0.985 and 0.977.

Table 3. Risk Score Performance Across Thresholds (Acc. = Accuracy)

Threshold	Acc(%)	Recall	F_1
0.1	88.5	1	0.936
0.2	88.7	1	0.937
0.3	88.7	1	0.937
0.4	90.0	1	0.944
0.5	92.8	1	0.959
0.6	94.9	1	0.970
0.7	97.6	1	0.985
0.8	97.5	0.983	0.985
0.9	96.1	0.957	0.977

These results indicate that gradually increasing the threshold improves classification reliability, but excessively high thresholds may cause some hazardous segments to be missed. In particular, a threshold of 0.7 can be regarded as the most balanced operating point overall, as it achieved the highest Accuracy (97.6%) while simultaneously maintaining Recall of 1.00 and the highest F_1 score (0.985). Therefore, in this study, 0.7 was selected as the representative threshold for the subsequent analysis of temporal variation patterns.

Figure 2 shows the temporal evolution of $RS(t)$ under this representative threshold in the streaming setting ($t = 2.0\text{s}$ $S = 0.5\text{s}$) across three hazard scenarios. In the graph, the risk score rises distinctly during the annotated hazardous intervals and is generally well aligned with the shaded GT-unsafe regions. This suggests that the proposed Risk Score appropriately reflects the timing of hazardous events across different hazard situations, including Vertical, Mobile, and Fall.

Overall, the proposed Risk Score enabled the identification of a stable operating point through threshold analysis, and under this threshold, it showed high segment-level classification performance and favorable temporal alignment across all three hazard scenarios. These results support that, even in a streaming-based safety monitoring environment, the proposed method can achieve both strong detection performance and stable operational behavior.

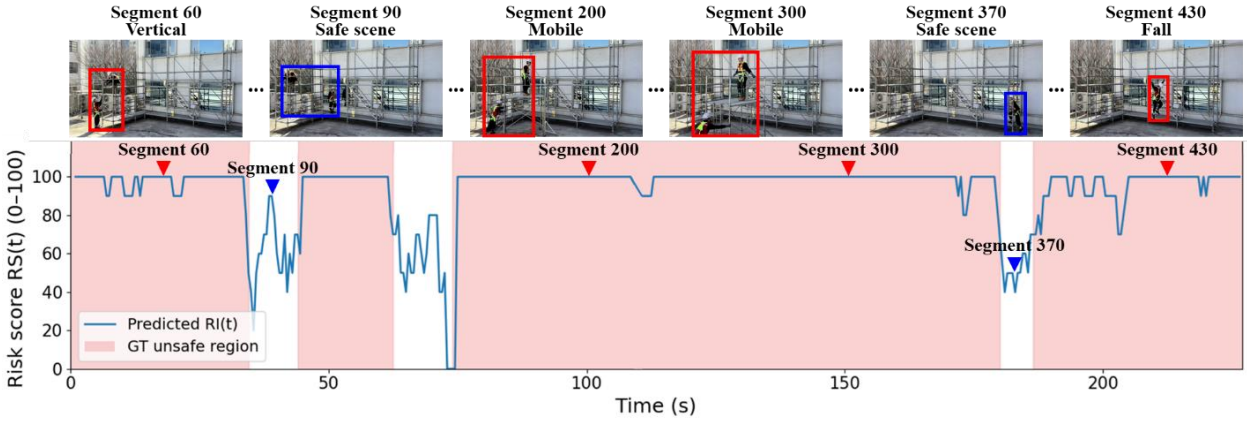

Figure 2. Risk Score Align with Unsafe Events in Streaming Video

4.6 End-to-End Streaming Feasibility

Table 4 reports the end-to-end per-segment processing latency of each model under the streaming stride setting ($S = 0.5$s) to verify whether the system satisfies the real-time condition $L \leq S$.

Table 4. Comparison for Real-Time Operation

Model	Stride	Latency	Realtime
Intern	500.0 ms	1274.3ms	✗
Qwen	500.0 ms	1663.8ms	✗
Smol	500.0 ms	3748.6ms	✗
Ours	500.0 ms	210.8 ms	✓

In this study, real-time capability is defined in terms of timeliness, that is, whether the system can update its output without delay as an event unfolds. Because the streaming pipeline updates its output every 0.5s, real-time operation is considered feasible only when the mean per-segment processing latency L does not exceed the stride S. As shown in Table 3, the proposed method achieved a mean latency of 210.8 ms, which is substantially smaller than the stride of 500.0 ms and therefore satisfies the real-time condition. In contrast, the baseline models, InternVL2.5-2B, Qwen2.5-VL, and SmolVLM2-2.2B, showed mean latencies of 1274.3 ms, 1663.8 ms, and 3748.6 ms, respectively, all of which exceed the stride. This indicates that real-time online operation is difficult for these models under the same setting. These results show that only the proposed method can provide timely segment-level updates within the required streaming interval. Therefore, the proposed framework not only demonstrates effective hazard interpretation performance, but also secures a processing speed that is practically applicable to real-time safety monitoring even in a single-GPU environment.

5 Conclusion

This study proposed a real-time construction safety monitoring framework that interprets streaming site videos through text-bank-based retrieval and provides both a segment-level risk score $RS(t)$, and supporting evidence texts. The framework constructs a structured text bank, aligns video and text in a shared embedding space, and retrieves top-K evidence texts for each segment to compute and visualize risk without site-specific retraining.

Experimental results showed that the proposed Risk Score achieved stable classification performance across threshold settings, with 0.7 identified as the most balanced operating point (Accuracy = 97.6%, Recall = 1.00, and F_1 = 0.985). Under this threshold, the temporal response of the risk score was generally well aligned with the GT-unsafe intervals across the three hazard scenarios: Vertical, Mobile, and Fall. In addition, the proposed method consistently outperformed the generative baseline models in interpretability, showing stronger semantic and structural alignment with the GT explanatory captions.

The proposed framework also demonstrated practical real-time capability. Under the streaming setting of $S = 0.5$ s, the mean end-to-end latency was 210.8 ms, satisfying the condition $L \leq S$, whereas the baseline VLMs exceeded the allowable interval. These results suggest that the proposed method can support low-latency safety monitoring in streaming environments.

This study has several limitations. Validation was conducted on a limited set of hazard scenarios and a single experimental setting, so generalization under more diverse conditions remains to be verified. Future work will extend the framework through broader evaluation, expansion of the hazard taxonomy and text bank, and adaptive alarm strategies based on temporal accumulation, smoothing, and dynamic thresholds.

Acknowledgement

This research was supported by a grant(RS-2022-00143493) from Digital-Based Building Construction and Safety Supervision Technology Research Program funded by Ministry of Land, Infrastructure and Transport of Korean Government.

This research was supported by the Young Researcher Program (RS-2026-25529211) through the Korea Agency for Infrastructure Technology Advancement (KAIA), funded by the Ministry of Land, Infrastructure and Transport of the Republic of Korea.

References

[1] S. F. A. Zaidi, J. Yang, M. S. Abbas, R. Hussain, D. Lee, and C. Park. Vision-based construction safety monitoring utilizing temporal analysis to reduce false alarms. Buildings, 14(6):1878, 2024.

[2] Y. Li, H. Wei, Z. Han, N. Jiang, W. Wang, and J. Huang. Computer vision-based hazard identification of construction site using visual relationship detection and ontology. Buildings, 12(6):857, 2022.

[3] W. Fang, L. Ding, P. E. Love, H. Luo, H. Li, F. Peña-Mora, et al. Computer vision applications in construction safety assurance. Automation in Construction, 110:103013, 2020.

[4] X. Li, T. Hao, F. Li, L. Zhao, and Z. Wang. Faster R-CNN-LSTM construction site unsafe behavior recognition model. Applied Sciences, 13(19):10700, 2023.

[5] W. Sultani, C. Chen, and M. Shah. Real-world anomaly detection in surveillance videos. In Proceedings of the IEEE CVPR, pages 6479–6488, 2018.

[6] X. Zhou, A. Arnab, S. Buch, S. Yan, A. Myers, X. Xiong, et al. Streaming dense video captioning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 18243–18252, 2024.

[7] S. K. Lee and J. H. Yu. Ontological inference process using AI-based object recognition for hazard awareness in construction sites. Automation in Construction, 153:104961, 2023.

[8] Y. R. Lee, S. H. Jung, K. S. Kang, H. C. Ryu, and H. G. Ryu. Deep learning-based framework for monitoring wearing personal protective equipment on construction sites. Journal of Computational Design and Engineering, 10(2):905–917, 2023.

[9] M. S. Abbas, R. Hussain, S. F. A. Zaidi, D. Lee, and C. Park. Computer vision-based safety monitoring of mobile scaffolding integrating depth sensors. Buildings, 15(13):2147, 2025.

[10] C. Ding, Q. Liu, X. Guo, T. Xue, and Z. Wang. Identifying unsafe behaviors of construction workers through an unsupervised multi-anomaly GAN approach. Automation in Construction, 165:105509, 2024.

[11] H. Du, S. Zhang, B. Xie, G. Nan, J. Zhang, J. Xu, et al. Uncovering what why and how: A comprehensive benchmark for causation understanding of video anomaly. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 18793–18803, 2024.

[12] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, et al. Qwen2.5-VL technical report. arXiv preprint arXiv:2502.13923, 2025.

[13] A. Marafioti, O. Zohar, M. Farré, M. Noyan, E. Bakouch, P. Cuenca, et al. SmolVLM: Redefining small and efficient multimodal models. In Conference on Language Modeling (COLM), 2025.

[14] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 24185–24198, 2024.

[15] Q. M. Dinh, M. K. Ho, A. Q. Dang, and H. P. Tran. Trafficvlm: A controllable visual language model for traffic video captioning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 7134–7143, 2024.

[16] B. Xiao, Y. Wang, Y. Zhang, C. Chen, and A. Darko. Automated daily report generation from construction videos using ChatGPT and computer vision. Automation in Construction, 168:105874, 2024.

[17] D. Yang, M. Kim, S. Mac Kim, B. W. Kwak, M. Park, J. Hong, et al. Llm meets scene graph: Can large language models understand and generate scene graphs? A benchmark and empirical study. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers, pages 21335–21360, 2025.

[18] S. M. H. Abidi, S. M. Raza, and S. Y. Shin. SafeVision: Vision-language reasoning for context-aware safety monitoring. Neurocomputing, 669:132479, 2026.

[19] Occupational Safety and Health Administration (OSHA). A Guide to Scaffold Use in the Construction Industry. OSHA Publication 3150 (Revised 2002), 2002.

[20] D. Bolya, P.-Y. Huang, P. Sun, J. H. Cho, A. Madotto, C. Wei, T. Ma, J. Zhi, J. Rajasegaran, H. A. Rasheed, J. Wang, M. Monteiro, H. Xu, S. Dong, N. Ravi, S.-W. Li, P. Dollár, and C. Feichtenhofer. Perception Encoder: The best visual embeddings are not at the output of the network. NeurIPS, 2025