AN INTEGRATED YOLO AND VLM SYSTEM FOR FIRE DETECTION IN ENCLOSED ENVIRONMENTS

Jongeun Kim,* Yejin Lee,* Dongsik Yoon*, Chansung Jung & Gunhee Lee
HDC LABS
Seoul, Korea
{JongeunKim, yejin.lee, kevinds1106}@hdc-labs.com

ABSTRACT

While YOLO models show promise in car fire detection, they remain insufficient for real-world deployment in underground and indoor car parks due to dataset limitations, evaluation gaps, and deployment constraints. We first fine-tune YOLO on fire/smoke-augmented dataset, but analysis reveals that it struggles with ambiguous fire-smoke boundaries, leading to false predictions. To address this, we propose a real-time end-to-end framework integrating YOLOv8s with Florence2 VLM, combining object detection with contextual reasoning. While YOLOv8s with VLM improves detection reliability, challenges are still ongoing. Our findings highlight YOLO's limitations in fire detection and the need for a more adaptive, environment-aware approach.

1 INTRODUCTION

A recent study (Meraner, 2023) highlights the growing risks of vehicle fires in underground/indoor car parks, emphasizing how the reduced spacing between adjacent cars -resulting from larger vehicle sizes and limited parking availability- leads to substantial property and environmental damage. Furthermore, Brzezinska & Bryant (2022) provides an in-depth review of the increasing fire incidents involving electric vehicles (EVs) powered by high-energy batteries, posing greater risks as the battery pack continues to burn even after water extinguishment, spreading to nearby vehicles within 5 minutes. Despite this fact, existing fire detection research (Moradi et al., 2024; Seydi et al., 2022; Shen et al., 2018; Gupta et al., 2021; Namozov & Im Cho, 2018; Son et al., 2018) primarily depends on multi-sensor information or focuses on detection in environments such as outdoor settings of wildland management to urban infrastructure (e.g., forest fires, building fires), or close-up fire imagery, leaving a critical gap in addressing vehicles fire detection in confine car parks. As a result, early fire suppression is crucial and only a few studies explored car fire detection in surveillance videos. Zhang et al. (2022) used a handmade dataset with a modified YOLOv4, while Dilshad et al. (2023) proposed an end-to-end system inspired by VGG16. However, both studies rely on custom datasets focused on already-developed flames, neglecting early smoke detection and indoor car park scenarios, where the risk of damage is highest.

First, we tune YOLOv8s capabilities by training it on our proposed dataset, handling the issue of data scarcity in vehicle fire detection within car park zones. Building on AI-generated data augmentation strategies (Rombach et al., 2022; Yang et al., 2023; He et al., 2022; Fang et al., 2024; Dunlap et al., 2023; Tian et al., 2024), we use Stable Diffusion (Rombach et al., 2022) to generate synthetic yet realistic images of vehicle fires set against car park backgrounds. This method pursues to address the lack of annotated fire and smoke images in these environments and provide a robust and diverse training dataset to effectively handle various real-world fire scenarios. However, through extensive analysis, we notice this is insufficient for immediate on-the-scene application.

To tackle the drawbacks of previous studies and handle car fire detection in confined parking zones, we further propose a novel End-to-End **(E2E)** fire detection framework specialized for challenging indoor and underground settings, by incorporating **VLM** (Vision-Language Model) as an auxiliary model to enhance the **YOLO** (You Only Look Once) (Redmon, 2016) object detection model. This unified approach seeks to adopt YOLO's real-time detection abilities and VLM's overall contextual scene understanding qualities.

In the end, we apply our framework to real-time CCTV footage of actual underground car parks. This comprehensive system processes live surveillance feeds, promptly detects fire or smoke with

*Equal contribution.

minimal latency, and enables swift emergency responses to enhance safety and reduce potential property damage in high-risk parking facilities.

In summary, contributions of this paper are as follows:

- We fine-tune YOLO model on custom smoke and fire detection dataset specifically created for indoor/underground car park and conduct analysis on why it fails on real-world.
- We incorporate the VLM with the YOLO model and suggest a new metric as a unified framework to overcome drawbacks of practical implications.
- Finally, we introduce an end-to-end framework operating real-time on CCTV footage, targeting prompt response to fire incidents of the actual underground parking zones.

2 Customary Approach: Finetuning on Specific Domain

In this section, we first finetune the YOLO model on fire/smoke augemented dataset and provide analysis of fundamental issues to facilitate real-world applications of previous approaches.

2.1 Fire/Smoke Dataset Augmentation

To address the lack of annotated fire and smoke data, synthetic yet realistic images are generated using Stable Diffusion (Rombach et al., 2022) on our custom-obtained underground car park CCTV footage. These augmented images simulate fire and smoke in underground and indoor car park environments, providing diverse and representative data for training the detection model. An elaborate flowchart of how we create fire and smoke in these images is explained further in the Appendix A.1. In total, we train on 11.5K images of various indoor/outdoor scenarios targeting car park fire as well as original flames and forest fires, where synthetic data comprises each 21% for training and both 12% for validation and test. Detailed information of the datatset can be found in Appendix A.2

2.2 YOLO Finetuning

The YOLO object detection model is trained on both augmented and acquired datasets to improve its detection ability, including specific circumstances of the enclosed parking zone. This finetuning process equips the model to handle car park environments' unique and challenging conditions.

2.3 Analysis

(a) (b)

Figure 1: Qualitative results of YOLO model, where YOLO prediction on top and GT on bottom.

We conduct a thorough analysis and explain why simply fine-tuning the YOLO model on benchmark and synthetic datasets is ineffective in real-world application. Qualitative results on Figure 1 focus on failure cases in our test set, with success cases in Appendix D. In Subfigure 1a we focus on shortcomings of dataset annotations, which struggle with ambiguous smoke and fire boundaries, leading to false predictions. In particular, since fire and smoke are part of the same combustion process, it is difficult to pinpoint where fire ends and smoke begins, leading to inaccurate annotations, false negatives, mislabeling, and low confidence scores.

While the traditional object detection metric is essential for evaluating object detection models, its strict per-instance thresholds for localization and confidence often fail to reflect practical outcomes in fire and smoke detection applications. When considering fire detection in real-world scenarios, the primary goal should be to obtain at least one reliable detection rather than striving for perfect bounding box alignment. Subfigure 1b demonstrates such cases, where detected bounding boxes closely align with the ground truth but may still be penalized under instance-based evaluation criteria. To manage this, we introduce a per-image binary detection evaluation metric in Section 3.2,

which combines classification with spatial consideration, ensuring reliable detection system through high-confidence detections with partial overlaps.

3 END-TO-END FIRE DETECTION FRAMEWORK

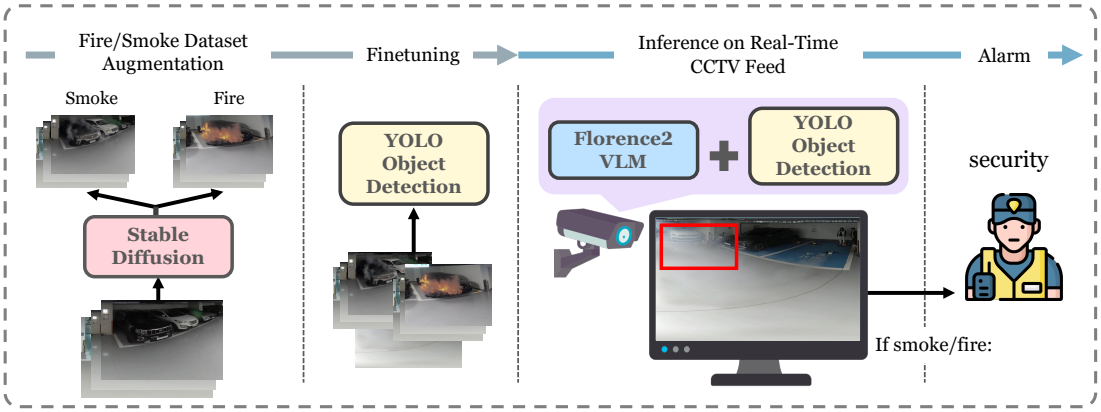

Figure 2: An End-to-End fire detection framework designed for detecting car fire/smoke events in real-time CCTV footage of confined environments. The framework consists of four main stages: **data augmentation, training, real-time inference, and alerting security.**

3.1 REAL-TIME INFERENCE WITH YOLO AND FLORENCE2 INTEGRATION

$$\mathcal{T}_{\text{pred}} = \begin{cases} \mathcal{T}_{\text{mod}}, & \text{if VLM detects smoke or fire,} \\ \mathcal{T}_{\text{init}}, & \text{otherwise.} \end{cases} \tag{eq1}$$

To develop an effective real-world fire detection system, we propose a novel end-to-end framework, as illustrated in Figure 2. Beyond the previous domain adaptation stages, the trained YOLO model collaborates with the Florence2 VLM during inference, analyzing CCTV frames to respond to the contextual prompt, *"Is there smoke or fire?"*. If the VLM detects smoke or fire as shown in eq1, $\mathcal{T}_{\text{pred}}$, the confidence threshold of the YOLO model, is dynamically lowered to $\mathcal{T}_{\text{mod}}$ to enhance detection sensitivity. Conversely, if the VLM does not identify smoke or fire, the YOLO model maintains its initial confidence threshold, $\mathcal{T}_{\text{init}}$, to reduce false positives. This unified approach combines to take advantage of the strengths of both models: real-time object detection of YOLO and language-based contextual understanding of VLM. We choose Florecne2 as our main VLM due to its strong overall features among other VLMs, where performance comparison are given in Appendix C.1.

3.2 PER-IMAGE DETECTION SUCCESS/FAILURE(BINARY CLASSIFICATION)

In the context of E2E fire detection, the priority should be securing at least one reliable detection rather than striving for precise bounding box alignment. Therefore, we propose a modified evaluation metric that prioritizes detecting the presence of fire or smoke in each frame.

For each image I_i, if there exists at least one predicted box whose IoU with the ground truth box is $\geq \mathcal{T}_{\text{iou}}$, we regard this image as having a "successful detection" ($d_i = 1$). Otherwise, we say the detection failed ($d_i = 0$). Formally:

$$d_i = \mathbf{1}\Big(\max_{b_{ij} \in \mathcal{B}_i^{\mathcal{T}}} \text{IoU}(b_{ij}, b_i^*) \;\geq\; \mathcal{T}_{\text{iou}}\Big), \tag{eq2}$$

where $\mathbf{1}(\cdot)$ is the indicator function, returning 1 if the condition is true, and 0 otherwise.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} \;= \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad \text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

Using eq2, we define precision and recall, where precision represents the proportion of correctly predicted "object-present" instances, and recall measures the percentage of actual positives (images containing the object) that are accurately detected. Refer to B.2 for detailed definitions.

3.3 ALERTING SECURITY

When the YOLO model predicts the presence of fire or smoke, simultaneously validated by the VLM, an alert is immediately triggered to notify security. This facilitates a prompt response to potential fire accidents, thereby reinforcing safety measures and minimizing property damage. The algorithm outlining the process for the E2E framework is provided in Appendix C.4.

4 EXPERIMENTS

This section breaks down the comprehensive experiments of our proposed framework, where in each table VLM indicates Florence2 Model and best scores are indicated in bold.

model	precision		recall		F1 score	model	precision		recall		F1 score
	fire	smoke	fire	smoke			fire	smoke	fire	smoke	
YOLOv6s	**.861**	.915	.853	.797	.855	+VLM	**.85**	.901	.885	.847	.8707
YOLOv6m	.86	.911	.872	.849	.873	+VLM	.837	**.904**	.894	.879	.8784
YOLOv8s	.843	**.944**	.835	.833	.863	**+VLM**	.838	**.931**	.904	.885	**.8895**
YOLOv8m	.843	.936	.839	.838	.863	+VLM	.839	.918	**.908**	**.89**	.8886

Table 1: Automatic evaluation of YOLO models with and without the VLM integration on test set. A complete comparison of different YOLO model versions can be found in Appendix C.3.

4.1 QUANTITATIVE RESULTS

To design our end-to-end system, we assess the models by using our per-image binary detection metric, where its necessity mentioned above in Section 2.3. Since metrics for individual bounding boxes are secondary to the system's ability, we leave traditional evaluation in Appendix C.2 and emphasize on accurately detecting the presence of the fire and smoke. As shown in Table 1, the YOLO model with VLM, adjusting the confidence threshold from 0.5 to 0.3, outperforms all and each standalone YOLO versions, despite the precision and recall trade-off.

We identify ***YOLOv8s*** as the most suitable model for our proposed system due to its lightweight design and superior F1 score across metrics. YOLOv8s achieves the highest F1-score, 0.8895, slightly outperforming YOLOv8m, 0.8886, offering a more balanced trade-off between precision and recall. Its slightly higher precision makes it the more reliable option when reducing false positives.

4.2 ABLATION STUDY

Model Settings			precision		recall		F1 score	Δ
	Training Dataset	τ_{mod}	fire	smoke	fire	smoke		
YOLOv8s	*	0.5	**0.843**	**0.944**	0.835	0.833	0.863	-
+VLM	*	0.4	0.839	0.935	0.885	0.863	0.8805	0.0175
+VLM	*	0.3	0.838	0.931	**0.904**	**0.885**	**0.8895**	0.0265
YOLOv8s	w/o syn data	0.5	**0.843**	0.897	0.843	0.702	0.8183	-0.045

Table 2: Ablation study on the impact of confidence threshold adjustments and the exclusion of synthetic data.

To evaluate our proposed framework, we conduct ablation studies shown in Table 2. We first test confidence thresholds $\mathcal{T}_{\text{mod}}$ of 0.5, 0.4, 0.3 to determine the optimal value for performance. The results indicate that as the threshold decreases, recall for both fire and smoke detection improves significantly, reaching the highest values of 0.904 and 0.885, respectively, at a threshold of 0.3. However, precision for both categories shows a slight decline, reflecting a trade-off between precision and recall. The integration of VLM also consistently enhances overall performance, as evidenced by the increase in the F1 score from 0.863 to 0.8895.

Moreover, we provide evaluations on the effect of synthetic fire and smoke data by training the model with and without synthetic data. For fire, which had relatively fewer samples, the impact was minimal, and in some cases, recall even increased. However, for smoke, which relied heavily on synthetic data, both precision and recall dropped significantly. The model trained with synthetic data achieved an F1 score that was 0.045 higher than the one trained without it.

5 DISCUSSION

Despite our novel approach to unify Florence 2 with YOLO as an end-to-end framework, some minor considerations remain. Despite utilizing Using Stable Diffusion to synthesize specific fire patterns remains challenging, as flames and smoke, despite their upward-rising characteristics, are restricted to the car's bounding box, limiting realism. Another attribute arises during the transition from offline to online deployment, particularly in addressing site-specific environmental configurations. Notably, reading frames from RTSP streams and uploading them to the AWS server takes approximately 1.6 times longer than offline processing, introducing additional latency. Consequently, to maintain real-time inference, VLM is limited to operating at 4 FPS as a complementary module to YOLO, rather than running at full capacity or independently. In the Appendix E, we outline the potential strategies to further enhance both the real-world applicability and performance of our approach.

REFERENCES

Dorota Brzezinska and Paul Bryant. Performance-based analysis in evaluation of safety in car parks under electric vehicle fire conditions. *Energies*, 15(2):649, 2022.

Naqqash Dilshad, Taimoor Khan, and Jaeseung Song. Efficient deep learning framework for fire detection in complex surveillance environment. *Comput. Syst. Sci. Eng.*, 46(1):749–764, 2023.

Lisa Dunlap, Alyssa Umino, Han Zhang, Jiezhi Yang, Joseph E Gonzalez, and Trevor Darrell. Diversify your vision datasets with automatic diffusion-based augmentation. *Advances in neural information processing systems*, 36:79024–79034, 2023.

Haoyang Fang, Boran Han, Shuai Zhang, Su Zhou, Cuixiong Hu, and Wen-Ming Ye. Data augmentation for object detection via controllable diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1257–1266, 2024.

Nisarg Gupta, Prachi Deshpande, Jefferson Diaz, Siddharth Jangam, and Archana Shirke. F-alert: early fire detection using machine learning techniques. *Int. J. of Electronics Engineering and Applications*, 9(3):34–43, 2021.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022.

Christoph Meraner. Car park fires: A review of fire incidents, progress in research and future challenges. In *Seventh International Conference on Fires in Vehicles, Stavanger, Norway, April 24-25, 2023*, pp. 7, 2023.

Sina Moradi, Mohadeseh Hafezi, and Aras Sheikhi. Early wildfire detection using different machine learning algorithms. *Remote Sensing Applications: Society and Environment*, 36:101346, 2024.

Abdulaziz Namozov and Young Im Cho. An efficient deep learning algorithm for fire and smoke detection with limited data. *Advances in Electrical and Computer Engineering*, 18(4):121–128, 2018.

J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Seyd Teymoor Seydi, Vahideh Saeidi, Bahareh Kalantar, Naonori Ueda, and Alfian Abdul Halin. Fire-net: A deep learning framework for active forest fire detection. *Journal of Sensors*, 2022(1): 8044390, 2022.

Dongqing Shen, Xin Chen, Minh Nguyen, and Wei Qi Yan. Flame detection using deep learning. In *2018 4th International conference on control, automation and robotics (ICCAR)*, pp. 416–420. IEEE, 2018.

GeumYoung Son, Jang-Sik Park, Byung-Woo Yoon, and Jong-Gwan Song. Video based smoke and flame detection using convolutional neural network. In *2018 14th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pp. 365–368. IEEE, 2018.

Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. Stablerep: Synthetic images from text-to-image models make strong visual representation learners. *Advances in Neural Information Processing Systems*, 36, 2024.

Zuhao Yang, Fangneng Zhan, Kunhao Liu, Muyu Xu, and Shijian Lu. Ai-generated images as data source: The dawn of synthetic era. *arXiv preprint arXiv:2310.01830*, 2023.

Shiyu Zhang, Qing Yang, Yuchen Gao, and Dexin Gao. Real-time fire detection method for electric vehicle charging stations based on machine vision. *World electric vehicle journal*, 13(2):23, 2022.

Broader Impact Statement

This work highlights the challenges and limitations of applying deep learning to fire and smoke detection in confined environments, such as indoor and underground parking facilities. While our framework demonstrates significant improvements in synthetic-real data alignment and early detection capabilities, it also exposes the practical hurdles of domain adaptation, gap between real and synthetic datasets. By addressing these gaps, we aim to bridge the divide between research advancements and real-world deployment, fostering discussions on the reliability and robustness of AI systems in high-stakes scenarios. This work emphasizes the need for transparent evaluation and practical solutions to ensure the effective application of deep learning where it matters most.

A Data Exploration

A.1 Synthetic Image Generation via Stable Diffusion

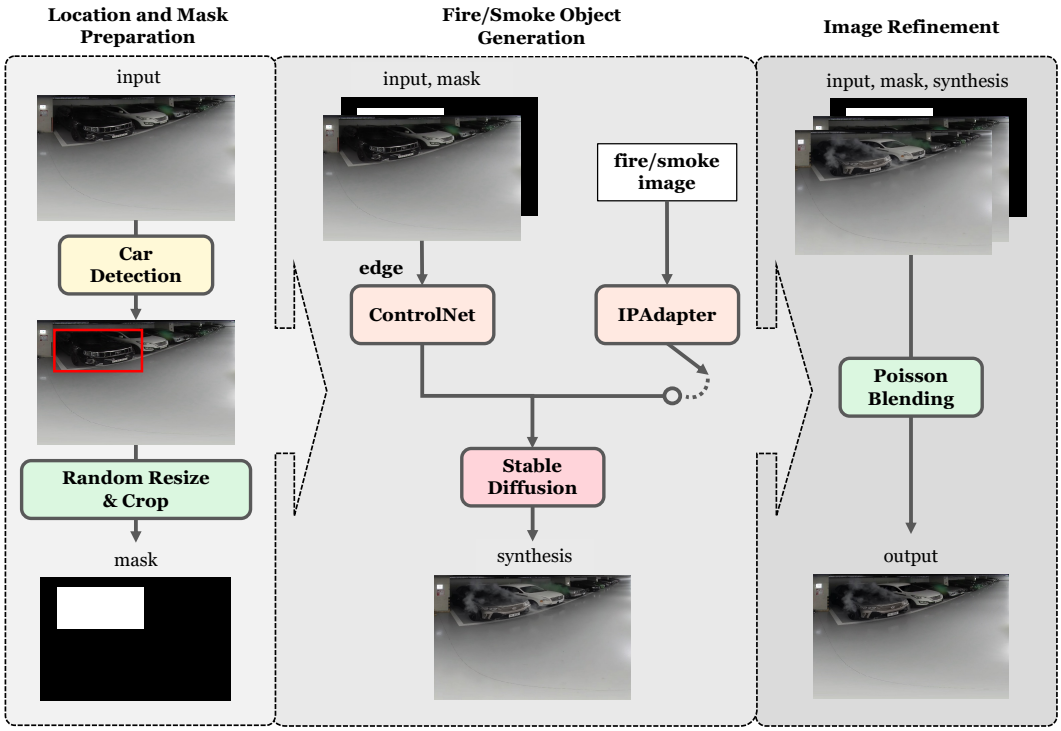

Figure 3: An End-to-End fire detection framework designed for detecting fire/smoke events in real-time CCTV footage of confined environments. The framework consists of four main stages: **data augmentation, training, real-time inference, and alerting security.**

Figure 3 illustrates our fire/smoke object dataset generation workflow, specifically designed to generate realistic synthetic datasets for fire scenarios in underground parking zones. The pipeline consists of three distinct stages: (1) identifying vehicle objects within the parking zones and determining the location of the calamity, (2) synthesizing images of the target vehicle engulfed in smoke or fire using a controllable diffusion model, and (3) applying blending-based post-processing to enhance visual fidelity.

In the first stage, an input image without fire or smoke serves as the base. A pretrained YOLOv5 model is utlized to detect objects of interest, such as vehicles. Subsequently, random resizing and padding operations are applied to generate a mask that specifies irregular regions for synthesizing fire or smoke effects.

In the second stage, synthetic fire or smoke images are generated using a controllable diffusion model. Specifically, Canny edges(**?**) are extracted from the input image and fed into ControlNet , ensuring structural alignment and producing realistic fire- or smoke-engulfed scenes. The edge condition is applied only during the initial two-thirds of the diffusion process to preserve the vehicle's structure, while the remaining steps allow greater flexibility in shaping the fire and smoke. For cases requiring specific fire styles or flame patterns, an optional IPAdapter is incorporated, enabling to customize the visual characteristics of the synthesized effects. The synthesis is performed using inpainting Stable Diffusion (SDInpaint) to generate the fire or smoke effects within the designated mask.

Finally, in the Image Refinement stage, Poisson blending is applied to seamlessly integrate the synthetic images with the original image. This refinement step effectively removes artifacts and ensures the final output is visually consistent with real-world scenes.

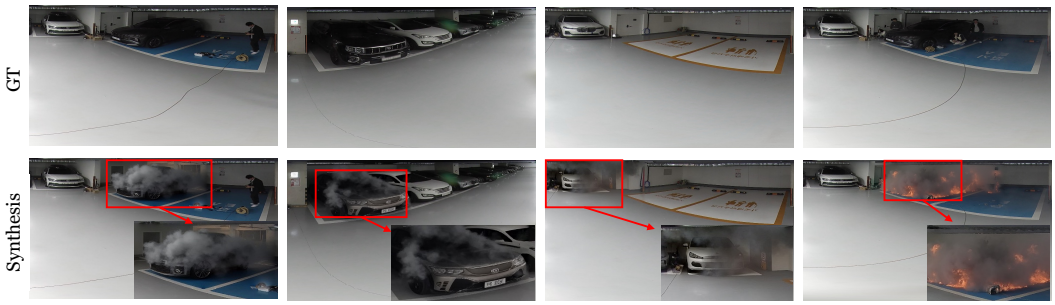

Figure 4: Success case of fire/smoke image generation.

As shown in Figure 4, our pipeline successfully synthesizes realistic fire- and smoke-engulfed scenes. However, due to the instability of SDInpaint, failure cases are discarded before entering the training phase. As is well known, diffusion models sometimes fail to accurately reflect text prompts, even when provided with the proper conditions. Therefore, at the final stage of the pipeline, users sometimes manually remove unrealistic(failure) samples.

A.2 Dataset Details

Dataset Name	Train	Val	Test	Total	Scenario	Target	Label
Tau_house_40	1782	200	200	2182	indoor	car	fire,smoke
CCTV-fire_50	1418	166	168	1752	indoor/outdoor	diverse	fire,smoke
fire and smoke	1607	159	159	1925	outdoor	big fires	fire,smoke
firecops	222	21	9	252	outdoor	car	fire
Synthetic Data - smoke	2354	130	130	2614	indoor	car	smoke
cctv-pano_50	4071	407	407	4885	indoor	car	
Synthetic Data - fire	100	9	8	117	indoor	car	fire
Total	**11.5K**	**1K**	**1K**				

Table 3: Details of the datasets used for our experiment, including the total number of samples, scenarios, targets, and labels for each dataset.

The dataset comprises a diverse collection of images designed to facilitate fire and smoke detection tasks across varying environments and scenarios. It includes seven sub-datasets, totaling 13,727 images, with 11,554 for training, 1,092 for validation, and 1,081 for testing. These datasets span multiple domains, including indoor, outdoor, and indoor/outdoor mixed environments, targeting specific objects such as cars or diverse items, and are annotated for fire, smoke, or both.
Real-world datasets, such as *Tau_house_40* and *cctv-fire_50*, provide annotated images for challenging indoor and mixed environments, whereas *fire and smoke* and *firecops* focus on outdoor settings with big fires and car fires, respectively. Additionally, synthetic datasets, including *Synthetic Data - smoke* and *Synthetic Data - fire*, simulate realistic indoor fire and smoke scenarios to address the scarcity of annotated data in such confined environments. Notably, *CCTV-pano_50*, the largest subset with 4,885 images, offers extensive data for indoor scenarios involving cars. This dataset ensures a comprehensive representation of real-world and synthetic conditions, enabling robust model training for early fire and smoke detection across various domains, particularly in confined spaces like indoor and underground parking facilities.

B Evaluation Metrics

B.1 Basic Definitions

Let $\mathcal{D} = \{I_1, I_2, \ldots, I_N\}$ be the set of all images to be evaluated, and let $N = |\mathcal{D}|$. If the target object exists in image I_i, then $G_i = 1$. Otherwise, $G_i = 0$. Let $\mathcal{B}_i = \{b_{i1}, b_{i2}, \ldots, b_{iM_i}\}$ be the set of bounding boxes predicted by the model for image I_i. Here, M_i is the number of predicted

boxes for I_i. Each predicted bounding box b_{ij} is associated with a confidence score s_{ij}. Here, s_{ij} represents the confidence score of the bounding box b_{ij}, and only boxes with scores exceeding a predefined confidence threshold τ_{pred} are considered:

$$\mathcal{B}_i^\tau = \{b_{ij} \mid s_{ij} \geq \tau_{pred}\}.$$

This ensures that only bounding boxes with sufficient confidence are used for evaluation. The IoU between a predicted bounding box b_{ij} and the ground truth bounding box b_i^* (in image I_i) is defined as:

$$\text{IoU}(b_{ij}, b_i^*) = \frac{|\, b_{ij} \cap b_i^* \,|}{|\, b_{ij} \cup b_i^* \,|}.$$

Let τ_{iou} be the minimum IoU threshold above which a predicted bounding box is considered a valid detection (i.e., "matched" with the ground truth). If IoU is not considered at all, detections in completely different locations may still be recognized as correct answers despite being false positives.

B.2 Per-Image Detection Success/Failure(Binary Classification)

For each image I_i, if there exists at least one predicted box whose IoU with the ground truth box is $\geq \tau_{iou}$, we regard this image as having a "successful detection" ($d_i = 1$). Otherwise, we say the detection failed ($d_i = 0$). Formally:

$$d_i = \mathbf{1}\Big(\max_{b_{ij} \in \mathcal{B}_i^\tau} \text{IoU}(b_{ij}, b_i^*) \;\geq\; \tau_{iou}\Big),$$

where $\mathbf{1}(\cdot)$ is the indicator function, returning 1 if the condition is true, and 0 otherwise.

- $d_i = 1$ means "the model claims there is at least one instance of the object in image I_i."
- $d_i = 0$ means "the model claims no object is found in image I_i."

We can interpret (G_i, d_i) as a binary classification scenario. Thus, the standard definitions of TP, FP, FN, and TN apply:

$$\text{TP} = \sum_{i=1}^{N} [\, G_i \cdot d_i \,], \quad \textbf{True Positive (TP):} \text{ The total count of images in which the object exists } (G_i = 1) \text{ and the model detects it } (d_i = 1). \tag{1}$$

$$\text{FP} = \sum_{i=1}^{N} [(1 - G_i) \cdot d_i], \quad \textbf{False Positive (FP):} \text{ The total count of images in which the object does not exist } (G_i = 0) \text{ but the model claims detection } (d_i = 1). \tag{2}$$

$$\text{FN} = \sum_{i=1}^{N} [G_i \cdot (1 - d_i)], \quad \textbf{False Negative (FN)} \text{ The total count of images in which the object exists } (G_i = 1), \text{ but the model fails to detect it } (d_i = 0). \tag{3}$$

$$\text{TN} = \sum_{i=1}^{N} [(1 - G_i) \cdot (1 - d_i)], \quad \textbf{True Negative (TN)} \text{ The total count of images in which the object does not exist } (G_i = 0), \text{ and the model also does not detect it } (d_i = 0). \tag{4}$$

Using the TP, FP, and FN values defined above, we can compute Precision and Recall for the entire dataset:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} \quad = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

Precision indicates the fraction of "object-present" predictions that are correct, while recall indicates the fraction of actual positives (images with the object) that are correctly identified.

In summary, if an image I_i contains at least one instance of the target object, then $G_i = 1$; otherwise, $G_i = 0$. If there is at least one predicted box with IoU $\geq \tau_{iou}$ against the ground truth box, then $d_i = 1$; otherwise, $d_i = 0$. After computing d_i for each image $I_i \in \mathcal{D}$, we sum up to get TP, FP, FN, and TN. Finally, we calculate the Precision and Recall values using the definitions above.

B.3 Per-Image Average Precision Calculation with Confidence Thresholds

To evaluate model performance across different confidence levels, we compute Precision-Recall (PR) curves by varying the confidence threshold τ_{pred} from 0 to 1 with a step size of 0.01. The Average Precision (AP) is then computed as the area under the PR curve. For each confidence threshold τ_{pred} , we compute the precision and recall using the previously defined formulas. The Precision-Recall Curve is constructed by plotting Precision against Recall at different confidence levels τ_{pred}. The AP is then computed as the area under this curve:

$$\text{AP} = \int_0^1 \text{Precision}(\text{Recall})\, d(\text{Recall}).$$

In practice, we approximate this integral using discrete summation:

$$\text{AP} \approx \sum_{k=1}^{K} (R_k - R_{k-1}) P_k.$$

whereP_k and R_k are precision and recall at different confidence thresholds and K is the total number of evaluated thresholds.

C Model Comparison

C.1 Vision-Language-Model Comparison

Model	Initial Detection Time	Detection	Params	Latency	Memory
BLIP - IC - base	17.58 s	smoke/fire	253M	0.164 s	3 GB
BLIP - IC - large	8.00 s	smoke/firee	580M	0.211 s	3.8 GB
BLIP2 - OTP - COCO	-	-	2.7 B	-	17 GB
BLIP2 - FLAN T5	8.87 s	fire	3B	0.43 s	17.5GB
LLAVA 7B (fp16)	8.1 s	fire	7B	0.85 s	16.3 GB
Florence 2 - base (fp16)	unclear	smoke/fire	0.23B	0.18s	1.2GB
Florence 2 - large (fp16)	**3.03s**	smoke/fire	0.77B	0.28s	2.3 GB

Table 4: Comparison of Vision-Language Models (VLMs) based on initial detection time, detected labels, number of parameters, frame latency, and GPU memory usage in a real underground cark park fire CCTV footage.

Table 4 presents a comparison of Vision-Language Models (VLMs) for detecting smoke and fire on a real underground car park fire CCTV footage, evaluating them based on initial detection time, detected labels, number of parameters, frame latency, and GPU memory usage. Across the models, **Florence 2 - large (fp16)** stands out with the best overall performance, featuring the fastest initial detection time of 3.03 seconds, the ability to detect both fire and smoke, a moderate latency of 0.28 seconds, and efficient GPU memory usage of 2.3GB, making it highly suitable for real-time application.
In contrast, *BLIP - IC base* exhibits the slowest detection time at 17.58 seconds, while *LLAVA 7B* consumes the most GPU memory (16.3GB) and has the highest latency (0.85 seconds), indicating limitations for deployment in low-resource environments and real-time. Although *Florence 2 - base (fp16)* offers the smallest parameter size (0.23B) and the lowest memory usage (1.2GB), its unclear detection capability makes it less reliable for this specific task. Similarly, models like *BLIP2 - FLAN T5* and *LLAVA 7B* is only able to identify fire detection, reducing their versatility.

Overall, this analysis highlights the trade-offs between detection speed, computational requirements, and model versatility across different VLMs, providing evidence as to why we chose *Florence 2* as our main VLM when merging with the YOLO model.

model_name	Precision	Recall	mAP50	mAP50:95
YOLOv5s	0.651	0.62	0.641	0.377
YOLOv5m	0.664	0.634	0.65	0.377
YOLOv6s	0.645	**0.662**	0.645	0.37
YOLOv6m	0.679	0.654	**0.669**	0.377
YOLOv8s	**0.681**	0.634	0.649	0.377
YOLOv8m	0.634	0.639	0.643	**0.38**
YOLOv10s	0.676	0.62	0.637	0.368
YOLOv10m	0.647	0.617	0.625	0.356

Table 5: Performance comparison of YOLO models with and without VLM integration.

C.2 YOLO Model Comparison

Table 5 presents a performance comparison of various YOLO models with and without VLM integration, using standard object detection evaluation metrics including precision, recall, mAP50, and mAP50:95. Among the models, *YOLOv8s* achieves the highest precision (0.681), highlighting its accuracy in correctly identifying objects. *YOLOv6m* stands out as the most balanced model, achieving the highest mAP50 (0.669) and AP per-image (0.9227), showcasing its strong object detection and classification capabilities. *YOLOv8m* outperforms all models in mAP50:95 (0.38), making it the most robust under stricter IoU thresholds. *YOLOv6s* leads in recall (0.662), demonstrating its effectiveness in minimizing missed detections. *YOLOv6m* stands out in classification and detection accuracy, while *YOLOv8m* performs well under stricter IoU conditions. Considering general performance, **YOLOv6m** and **YOLOv8s** exhibit competitive results in all metrics, making them versatile choices for general-purpose tasks when the object detection model is used independently.

C.3 Full YOLO model comparison with and without VLM

Model	precision		recall		F1	Model	precision		recall		F1
	fire	smoke	fire	smoke	Score		fire	smoke	fire	smoke	Score
yolov5s	0.835	0.894	0.766	0.811	0.8248	5s+VLM	0.838	0.883	0.881	0.885	0.8716
yolov5m	0.842	0.919	0.784	0.811	0.8369	5m+VLM	0.839	0.905	0.885	**0.89**	0.8797
yolov6s	**0.861**	0.915	0.853	0.797	0.8553	6s+VLM	0.85	0.901	0.885	0.847	0.8707
yolov6m	0.86	0.911	0.872	0.849	0.8728	6m+VLM	0.837	0.904	0.894	0.879	0.8784
yolov8s	0.843	**0.944**	0.835	0.833	0.8627	**8s+VLM**	0.838	0.931	0.904	0.885	**0.8895**
yolov8m	0.843	0.936	0.839	0.838	0.8632	8m+VLM	0.839	0.918	**0.908**	**0.89**	0.8886
yolov10s	0.851	0.934	0.789	0.814	0.8446	10s+VLM	0.85	0.912	0.858	0.882	0.8755
yolov10m	0.841	0.943	0.803	0.822	0.8504	10m+VLM	0.842	0.919	0.881	0.866	0.8770

C.4 Algorithm of Inference on Real-Time CCTV Feed

Algorithm 1 Real-Time Inference with YOLO and Florence2 Integration

Input: Trained YOLO model, Florence2 VLM, CCTV feed, initial threshold τ_{init}, modified threshold τ_{mod}

Output: Alert trigger for fire/smoke detection

Initialize $\tau_{pred} \leftarrow \tau_{init}$

for each frame in CCTV feed **do**

  Pass frame to Florence2 VLM with prompt: "Is there smoke or fire?"

  if VLM detects smoke or fire **then**

    $\tau_{pred} \leftarrow \tau_{mod}$ {Lower threshold for enhanced sensitivity}

  else

    $\tau_{pred} \leftarrow \tau_{init}$ {Maintain initial threshold to reduce false positives}

  end if

  Perform object detection using YOLO model with confidence threshold τ_{pred}

  if YOLO predicts fire or smoke **then**

    if Validated by VLM **then**

      Trigger alert to notify security

    end if

  end if

end for

D QUALITATIVE RESULTS

Figure 5: Qualitative results on our constructed test set. The first row presents the YOLOv8s predictions, while the second row displays the corresponding ground truth annotations. We present examples of successful cases, where the predicted bounding boxes closely match the ground truth, achieving an IoU exceeding the threshold $\mathcal{T}_{iou}$ and demonstrating high confidence scores.

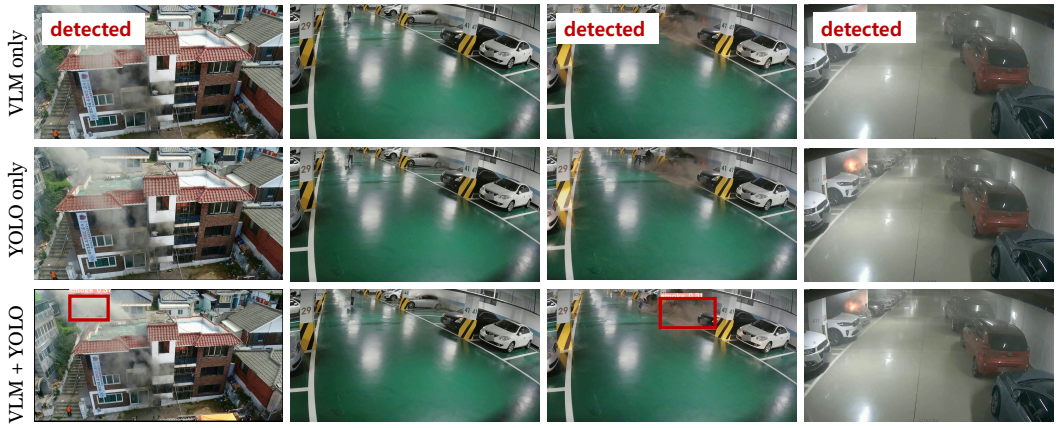

Figure 6: Qualitative results on real-world fire comparing on VLM only, YOLO only, and our proposed approach of VLM and YOLO integration.

As shown in 6, the VLM-only model detects three out of four smoke or fire incidents, whereas YOLO alone fails in all cases. However, when integrated with the VLM at a threshold of 0.3, YOLO successfully detects smoke and fire in two instances. Despite the higher detection performance offered by the VLM, it also leads to an increased number of false alarms.

E FUTURE WORKS

While the unified end-to-end fire detection framework combining YOLOv8s and Florence2 VLM improves performance and reliability of previous approaches, challenges remain, including ambiguous annotations, the synthetic-real data gap, and real-time deployment issues. Future work will focus on enhancing dataset quality, bridging the synthetic-real data gap, and optimizing the framework for seamless real-time deployment to further improve its practical performance.