Research Article

LangGas: Introducing Language in Selective Zero-Shot Background Subtraction for Semi-Transparent Gas Leak Detection with a New Dataset

Wenqi Guo[1,2], Yiyang Du[2,3], Shan Du[1]

1. Department of CMPS, University of British Columbia, Canada; 2. Group of Methane Emission Observation & Warning (MEOW), Weathon Software, Canada; 3. Department of Computational Linguistics, University of British Columbia, Canada

Gas leakage poses a significant hazard that requires prevention. Traditionally, human inspection has been used for detection, a slow and labour-intensive process. Recent research has applied machine learning techniques to this problem, yet there remains a shortage of high-quality, publicly available datasets. This paper introduces a synthetic dataset featuring diverse backgrounds, interfering foreground objects, diverse leak locations, and precise segmentation ground truth. We propose a zero-shot method that combines background subtraction, zero-shot object detection, filtering, and segmentation to leverage this dataset. Experimental results indicate that our approach significantly outperforms baseline methods based solely on background subtraction and zero-shot object detection with segmentation, reaching an IoU of 69% overall. We also present an analysis of various prompt configurations and threshold settings to provide deeper insights into the performance of our method. The dataset is available at https://forms.gle/aPHPfnM4Lwaz9FKB8.

Corresponding author: Shan Du, shan.du@ubc.ca

1. Introduction

Various organic gases are extensively used today across industry, both as fuels (for example, natural gas, methane) and starting materials for synthesis. However, methane emissions and other gases, particularly unintended leaks, are harmful and should be prevented. Methane has a significantly greater greenhouse impact than carbon dioxide (CO_2), exhibiting a heat-trapping potential over 28 times that of CO_2[1].

Additionally, many hydrocarbon gases can be toxic to humans[2]. In enclosed areas, leaked gases may pose a risk of hypoxia, endangering personnel, or causing fire and explosion hazards. For additional background, detailed justification, and the need for effective leak detection, readers may refer to several previously published papers[3][4][5].

There are many works focusing on gas leak detection using computer vision; however, despite many computer vision algorithms being data-intensive, public datasets are very scarce. Three major datasets in this field are GasVid[3], Gas-DB[6], and the Industry Invisible Gas Dataset (IIG)[5].

GasVid[3] is a dataset featuring controlled methane gas releases against a clear sky background, making it nearly ideal for foreground (leak) segmentation via background subtraction. Yet, this scenario rarely reflects real-world conditions. Moreover, GasVid lacks segmentation ground truth, providing only quantification classification that limits segmentation assessment to visual inspection. Gas-DB[6] includes segmentation but consists of still images instead of full-motion video. Although the images are sequential, they exhibit low continuity and short duration, making many video-based methods inapplicable. Both GasVid and Gas-DB rely on human-controlled releases, where leaks originate at the end of a releasing device such as a pipe. In comparison, IIG[5] contains videos of real gas leaks, but it has only bounding box annotations and is captured on handheld cameras, which introduces substantial camera motion.

We attempted to label pixel-level segmentation ground truth for GasVid[3], but the semi-transparent, blurry boundaries of gas leaks made annotation difficult. Consequently, our proposed method of Priori Ground Truth—knowing the ground truth before generating input data—can address these challenges by establishing accurate annotations from the outset.

Datasets in all domains exhibit inherent biases. GasVid and Gas-DB, for example, have spatial biases related to specific releasing devices, and all three datasets can be influenced by factors such as camera type, location, and lighting. Compiling large-scale gas leak detection data is inherently difficult, resulting in relatively small datasets that risk propagating any underlying biases to trained models. Recently, zero-shot techniques have gained attention for their low implementation cost and independence from training data.

To advance this area of research, we propose a novel computer-synthetic dataset offering diverse leakage points, perfectly accurate segmentation ground truth, and stable video recordings containing multiple moving objects. We produce high-quality data that avoids human labelling and retains precise

segmentations by overlaying realistically rendered leaks and interfering foreground elements onto varied backgrounds. In addition, we proposed a zero-shot method using a vision language model to avoid the model bias trained on this dataset. Our experiment shows that this method can achieve promising performance on this dataset.

Our contributions can be summarized as follows:

- We construct a diverse video-based computer-rendered dataset with complex backgrounds, interfering moving objects, and accurate ground truth.
- We propose a new baseline algorithm that combines background subtraction and zero-shot object detection to segment gas leakage accurately.

2. Related Work

2.1. Gas Leak Detection and Datasets

There are three main public datasets in the gas leak detection field. GasVid was proposed with GasNet[3] and VideoGasNet[7]. It contains a video dataset of controlled gas release. Most videos include the sky as a background, with gas released from a chimney-like structure. However, it was originally used as a classification dataset to determine if there is leakage and the amount of leakage without localization or segmentation information. Segmentation is not only important for precise localization but also required for better quantification of the gas release[4]. Most of its videos also do not have interference from other moving objects such as humans or cars, makinRGBund-subtraction method able to segment out the foreground (leak) with promising performances. In GasNet[3] and VideoGasNet[7], authors used background subtraction to remove non-moving parts in the video and kept a “soft” subtraction (without thresholding) Then, these subtracted frames (still frames in GasNet[3] and sequence of frames in VideoGasNet[7]) were sent into a CNN or ConvLSTM[8] to classify if there is a leak in the frames and the amount of leak.

Gas-DB[6], on the other hand, is a segmentation dataset. It contains over 1000 RGB-T images (images with 3 RGB channels and one thermal channel) with carefully labelled segmentation masks in different environments with other moving objects. The RGB channels provided more textual information than the thermal-only images. However, it was designed for image segmentation tasks without considering temporal information, which makes it hard to distinguish other similar-looking objects with leaks or

hard to detect faint leakage. Although the images are collected in temporal sequence and can be connected as videos, the lengths are usually very short, and the frame continuity is very low. Their model uses cross-modality attention to leverage the information in both RGB channels and thermal channels, achieving 56.52% IoU results. However, they split the dataset into training and validation sets using frame level instead of video level splits, meaning frames in the same video, which could have similar environments, can end up in training and validation sets. This means when the model is applied to unseen environments, the performance could potentially drop. This is similar to the situation for background subtraction for seen scenes vs. unseen scenes (Section 2.2). Additionally, both of these datasets have gas "leaked" from the end of pipe- or chimney-like structures. Therefore, if we train a model on these raw images (i.e. not the background-subtracted images in GasNet[3] and VideoGasNet[7]), the model could be biased toward these structures such that it will tend to relate these structures to the leakage, which is not necessary the case in real scenarios.

The Industrial Invisible Gas (IIG) dataset[5] is another recent IR-camera-captured dataset designed for object detection in real-world industrial environments. Unlike artificially simulated gas leaks, this dataset represents actual scenarios, avoiding biases associated with predefined leak locations, such as the ends of pipes. It consists of 5,569 images and includes five distinct scenes: pump oil seals, oil tank vents, gas stations, industrial chimneys, and other industrial settings. It was also captured as videos with high continuity, but the videos were captured using a handheld camera, which introduced camera jitter motion, making some video-based methods (such as background subtraction) hard to apply.

Furthermore, both GasVid[3] and GasDB[6] utilize real-world simulations with controlled gas releases. Although these methods aim to replicate real-world scenarios closely, they pose significant fire and explosion hazards (with the risk being lower in GasVid due to its open-air setting but higher in GasDB, where some experiments take place in partially enclosed spaces) and contribute to the release of greenhouse gases into the environment.

According to the GasVid paper[3], at least 12 kg of methane was released for the dataset used in the study —excluding emissions from testing and failed attempts (such as tank releases). While this quantity may be negligible globally, it still represents an avoidable environmental impact from a single experimental dataset. In contrast, a computer-generated dataset can achieve the same objectives without contributing to greenhouse gas emissions.

Humans' labelling of segmentation masks is inefficient and inaccurate. This is due to the gas plume's blurry boundary and transparent nature.

Synthetic datasets have been used in many areas. Guo et al.[9] demonstrated that for molecular model captioning, using a rendered dataset provides a straightforward method to obtain large amounts of data without human involvement, significantly boosting the downstream performance on real datasets. Wang et al.[10] used Unreal Engine 5 to simulate forest fire. Mao et al.[11] used 3D software to render forest fire smoke images and used CycleGAN[12] to generate more images. Gu et al.[13] also used the rendered dataset to simulate gas leakage to avoid manual labelling segmentation data.

2.2. Background Subtraction

Background subtractions (BGS)[14][15][16][17][18][19] have been used to detect moving parts in a video.[1] Non-deep-learning methods, such as MOG[14], MOG2[15][16], and kNN[16], do not require prior training or masks labelling for frames at the beginning of the video.

Over an extended period, one class of supervised background subtraction methods is video or video-group-optimized deep learning methods[20][21]. These methods require some frames from the target video or target video group to be labelled (i.e., segmentation by humans of what objects are in the foreground) to perform well. When applied to unseen videos, their performance drops dramatically[17]. For example, the original F-score reported by FgSegNet v2[20] using a video-optimized method was nearly perfect (0.9789), but when trained using the video-agnostic method by[17], the f-score is only 0.3715.

BSUV-Net[17] is one of the first deep-learning-based methods that could be applied to unseen videos with good results. BSUV-Net 2.0[18] improved it by using stronger data augmentation. Other deep learning methods for unseen BGS include[22][23][24][25][26].

Zero-shot background subtraction (ZBS)[19] introduced open vocabulary detection and segmentation for BGS in zero-shot settings. It uses an open vocabulary object detector to detect all objects in the image and use their movement across the video to determine if the object is foreground or background. While ZBS demonstrates effectiveness in handling illumination changes and pixel noise, it presents a fundamental limitation: it requires objects to be detectable before background subtraction can occur. This requirement creates a significant problem for applications such as leak detection. In such scenarios,

background subtraction is used as a preliminary step for object detection by extracting the moving objects—not the other way around. This creates a dependency loop for these applications.

2.3. Vision Language Models

Vision language models (VLMs)[27][28][29][30][31][32][33][34][35][36][37][38] have been advanced dramatically recently and show promising results in tasks combining vision and language, especially in zero-shot settings, such as language guided classification, segmentation, object detection, and vision grounding. CLIP (Contrastive Language Image Pretraining)[27] is one of the earliest and foundational works in this field. It uses contrastive learning pre-task where matched image-text pairs are used. This allows for zero-shot image classification by giving the model a list of candidate classes in natural language and calculating the similarity between the image and text features. Other VLMs expended CLIP-like zero-shot capabilities to localization (including grounding and detection)[37][39][40][36][35][30]. By combining these localization models with Segmente Anything Model (SAM)[41], language-guided instance level segmentation could be achieved, such as in Grounding SAM[42] and APOVIS[43]. More details of these models can be found in the supplemental material.

3. Dataset

We created the dataset by overlaying interfering foreground objects and gas leakage simulation footage onto background scenes. The foreground elements were sourced from two IR datasets, BU-TIV[44] and CAMEL[45][46], which include objects such as bats, cars, and humans. To extract objects of interest from these IR videos, we segmented the objects of interest using either thresholding or the SAM 2 model[47] with box annotations from the original dataset. Background footage was selected from GasVid[3] from non-leak portions or generated using DALL-E-2[48]. GasVid backgrounds are used to ensure our dataset includes sensotr noise, real-life lighting change, etc.. DALL-E-2 generated backgrounds are used to diversify the background senses. Gas leakage simulations were rendered in Blender using smoke simulation and force field. Some foreground objects, leakage simulations, and background scenes were reused in different combinations. For ground truth for segmentation, we used the generated “smoke” footage at the same position as in the overlay. A detailed comparison of our dataset with GasVid and GasDB is provided in the Table 1.

	Gas-DB[6]	GasVid[3]	IIG[5]	Ours
Format	Image Sequence	Video	Video	Video
Video Clip Length	Short (∼60 frames)	Long (∼21K frames)	Medium (∼500 frames)	Short-medium (∼300 frames)
Collecting Method	Controlled Release	Controlled Release	Real Emission	Computer Simulation
Spatial Bias	Yes	Yes	No	No
Background	Complex	Simple	Complex	Complex
Ground Truth	Manually labeled Masks	Quantification Classes	Bounding Boxes	Priori Ground Truth
Scene Variation	8	1	5	9
Other Moving Objects	Yes	Rarely	Some	Yes
Frame Continuity	Low	High	High	High
Total Number of Frames	1.2k	700k	5k	12k
Inter-Video Similarity	Low	High	Low	Medium
Availability	Public	Public	Upon Request	Public

Table 1. Comparison of Different Datasets

Similar to Gas-Vid[3] and Gas-DB[6], we removed some videos (26, 27, and 28) because they exhibit a strong, highly localized wind that causes smoke to disperse significantly. We believed that such extreme, erratic wind behaviour is rare in real-life scenarios. Therefore, we decided to exclude these videos, as they do not represent typical conditions and introduce unrealistic challenges. We also removed video 24 because of a misalignment between the video and the ground truth. We have retained these videos in the published dataset so that readers can review them and assess our decision.

4. Methods

A pipeline of our method can be found in Figure 1.

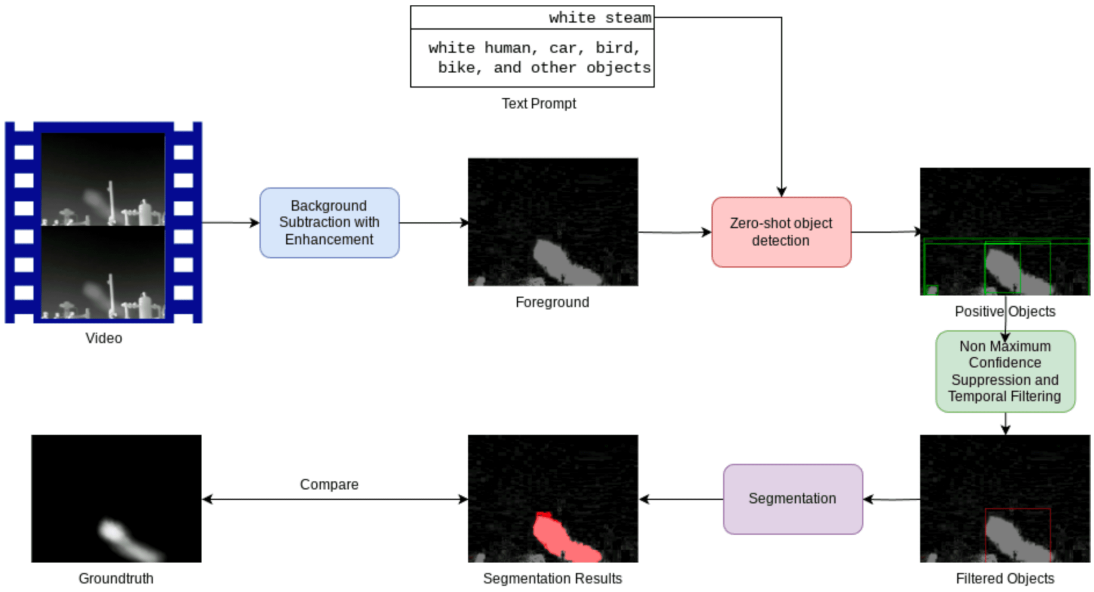

Figure 1. Method Overview: Our method for gas leak detection involves background subtraction, zero-shot object detection, non-maximum suppression (NMS), temporal filtering, and segmentation. First, background subtraction is used to identify the moving parts in the video. Then, two text prompts (positive and negative prompts) are employed to guide a zero-shot object detector in detecting leaks. We use the prompt ”white steam” because it is more commonly recognized than phrases explicitly mentioning gas leaks. NMS and temporal filtering are then applied to remove extra boxes and fix false positives or negatives based on past temporal information. Finally, a segmentation model—such as the Segment Anything Model 2 (SAM 2)—is used to convert the bounding boxes into segmentation masks.

4.1. Background Subtraction and Enhancement

A sequence of simulated IR-compared frames is processed with background subtraction to extract the moving part of the video. We used a short history (30) to avoid false positives from slow-moving objects such as clouds, which is the same approach used in GasNet[3].

Instead of relying on built-in mask generation for different background subtraction (BGS) methods, we extracted the background image from the algorithm and then computed the absolute difference between

the current frame and the background image: $I_i' = |I_{bg} - I_i|$ where I_i' is the difference, I_{bg} is the background image obtained from the BGS algorithm, and I_i is the current frame.

Since the difference could be subtle, we enhanced the image by a factor α and clipped the values between 0 and 255: $I_i'' = \min(\max(\alpha I_i', 0), 255)$.

Because the intensity of the difference may vary across different scenarios, we used an adaptive enhancement factor, as shown in Equation 1. We set the default factor to 15; however, this value could sometimes be too high when the intensity of the difference is large, leading to clipping and loss of image details. To mitigate this issue, we ensured that $\mu_{I_i'} + \sigma_{I_i'}$ (one standard deviation above the mean) does not exceed 255 by selecting a lower α, as shown in Equation 1.

$$\alpha = \min\left(\frac{255}{\mu_{I_i'} + \sigma_{I_i'}}, 15\right) \tag{1}$$

4.2. VLM Filtering

After background subtraction, all moving objects, including non-leak objects like humans, cars, birds, etc, are also extracted. To select leaks, we leverage the zero-shot object detection capability of a Vision-Language Model (VLM) Owlv2[36] to filter interested objects (leak), with the VLM Threshold (τ_{VLM}) as a hyperparameter that determines the threshold for a positive box output. However, using a prompt of “gas leak” or something similar might not be ideal as these models are usually trained on RGB modality images, in which gas is usually non-visible. Nevertheless, gas leaks in IR images resemble steam or smoke in RGB images. Thus, we chose to use “white steam” as the prompt for object detection. Detailed experiments with different prompts are shown in Section 5.4. We also used one negative prompt (*white human, car, bird, bike, and other objects*), such that when the objects were similar to “white steam” but more similar to the negative prompt, the VLM could correctly avoid it to reduce false positives. Finally, we applied a non-maximum suppression based on the confidence of each bounding box and the IoU between them to reduce overlapping boxes.

4.3. Temporal Filtering

Since the VLM only considers the current enhanced difference frame (Ii'') as input, it lacks temporal information. This limitation can lead to transient false positives or false negatives, causing issues with poor segmentation and false alarms.

To address this problem, we assume that a leak does not appear or disappear suddenly. We implement a temporal filtering mechanism that ensures a detected box is considered valid only if, within the past k_1 frames, at least n_1 boxes have an IoU greater than τ_{tIoU_1} or absolute shift greater than τ_{tShift} with the current box. This prevents transient false positives caused by noise or non-leak objects. We used $k_1 = 10, n_1 = 1, \tau_{tIoU_1} = 0.3$, and $\tau_{tShift} = 40$ in our experiment.

Similarly, we assume that a leak will not vanish suddenly. Therefore, if no leak is detected in the current frame, we look over the past k_2 frames and compare all detected boxes across these frames. If two boxes in different frames have an IoU greater than τ_{tIoU_2}, we infer that the leak is still present and add the corresponding box to the current frame. We used $k_2 = 3$ and $\tau_{tIoU_2} = 0.3$ in our experement.

These hyperparameters were not tuned extensively to avoid overfitting to a certain dataset.

These two filters provide a simple method to balance response time against false positive and false negative rates, leveraging the assumption that leaks are generally continuous. A detailed pseudocode implementation is provided in Algorithm 1 in the supplemental material, and we demonstrate its effectiveness in the experiment section.

4.4. Segmentation

After temporal filtering, the boxes are passed to SAM 2[47] to generate segmentation masks. SAM 2 produces a mask for each given box, which is combined using an OR operator. We utilize SAM 2 because it is less susceptible to noise and can effectively disregard non-primary objects, ensuring more accurate segmentation of the target subject.

5. Experiments and Results

5.1. Settings

In this paper, for each method tested, we performed a hyper-h method tested on key parameters such as morphological kernel size and threshold. To reduce computational intensity, ensure the approach aligns with real-world frame rate limitations—where hardware performance is constrained despite the need for real-time monitoring—and acknowledge that closely spaced frames are often similar, making individual evaluation unnecessary, we process every frame using the BGS algorithm but only perform VLM filtering and subsequent stages only every 5 frames. To ensure consistent comparison for the BGS-only baseline, we also performed BGS for every frame but only evaluated the result every 5 frames. The Owl-V2 model is

loaded from Huggingface using 4bit quantization and float16 computation type. The SAM 2 model is 2.1 Hiera Small using an official repository. We reported four metrics for comparison: IoU (I), precision (P), recall (R), and frame-level accuracy (FLA). Frame-level accuracy is the accuracy of frame-level classification. If any pixels in the frame are segmented, it is considered positive.

5.2. BGS-Only Baseline

In GasNet[3] and VideoGasNet[7], using only the background subtraction method yields a very clean and clear leakage (foreground) segmentation due to the static background and the absence of interfering moving objects such as cars and people. Therefore, we aim to establish a baseline using only BGS on our dataset. We experimented with different BGS methods, as shown in Table 2. All methods are run by using a history of 30, getting the background image from the background model and subtracting it from the current image; this difference image is then multiplied by 15 and thresholded by 40. These hyperparameters are hand-turned on MOG and broadcast to all methods, as tunning all hyperparameters for all settings is unrealistic.

BGS Method	**Refinement**	**Stationary Foreground I/P/R/FLA**	**Moving Foreground I/P/R/FLA**	**Overall I/P/R/FLA**
Median	Morph	0.50/0.63/0.74/**0.85**	0.30/0.52/0.44/0.68	0.43/0.59/0.63/0.79
Median	-	0.41/0.67/0.53/**0.85**	0.25/0.56/0.33/0.68	0.35/0.63/0.45/**0.79**
MOG2[15][16]	Morph	**0.56**/0.67/**0.8**/**0.85**	**0.38**/0.56/**0.57**/**0.69**	**0.5**/0.63/0.7/0.79
MOG2[15][16]	-	0.51/**0.68**/0.68/**0.85**	0.35/**0.6**/0.47/**0.69**	0.45/**0.65**/0.6/**0.79**
k-NN[16]	Morph	0.23/0.36/0.46/0.78	0.16/0.27/0.29/0.63	0.21/0.32/0.41/0.72
k-NN[16]	-	0.16/0.32/0.26/0.78	0.11/0.24/0.18/0.63	0.14/0.29/0.23/0.72

Table 2. BGS Only Baseline on Our Dataset We tested different background subtraction (BGS) methods with different refinement settings. We reported the intersection of union (IoU, I), precession (P), recall (R), and frame level accuracy (FLA). Frame level accuracy is the method's accuracy in classifying each frame to leak (with positive pixels) and no leak (all pixels are negative).

To refine the masks generated by the BGS method, we also evaluated the performance of morphological operations, specifically using opening to reduce salt noise and closing to connect separated segmentations (due to the weak appearance of the leakage). We tested various closing kernel sizes ranging from 10 to 50, with the best results presented in Table 2^2. Regardless of the morphological settings, the opening operation is applied to all runs, but the closing operation is only applied on the ones with morph checked. Results are reported separately for Stationary Foregrounds (without interfering objects) and scenes containing moving objects. Additionally, the performance of each method across different closing kernel sizes is illustrated in Figure 3.

From Figure 3 and Table 2, we can observe that MOG2 performs the best and larger morphological size tends to yield better results.

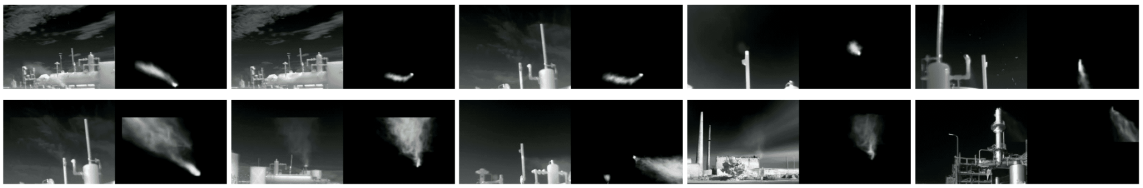

Figure 2. Preview of Our Dataset. These images are selected from 10 different videos. For each side-by-side subplot, the left one is the input frame, and the right one is the thresholded ground truth. Some of these videos use GasVid[3] as background, while others use DALL-E-2[48] generated background.

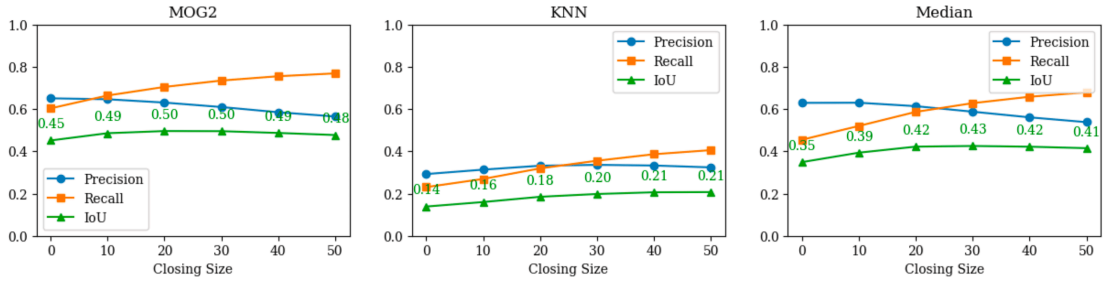

Figure 3. BGS Only Baseline on Our Dataset With Different Morphological Closing Operation Sizes

doi.org/10.32388/1HECOo

5.3. Ablation Study

Our method consists of four main components: background subtraction, VLM filtering, temporal filtering, and SAM 2[47] for segmentation. To systematically investigate the contribution and effectiveness of each individual component in our pipeline, we conducted an ablation study as summarized in Table 3. Specifically, we compared five experimental conditions:

1. **BGS-Only Baseline:** This row represents the best-performing result from the background-subtraction-only experiments described in Section 5.2. This could be considered as an improved version of the first step of GasNet[3] and VideoGasNet[7].
2. **BGS + VLM Filtering + SAM 2 (optimal threshold, without temporal filtering):** In this condition, we integrated visual-language model (VLM) filtering into the best-performing background subtraction system and employed Segment Anything Model 2 (SAM 2)[47] for converting bounding boxes generated by the VLM into segmentation masks. Noticeably, the only distinction from our complete proposed method is the absence of temporal filtering. By comparing this condition with our complete method, we analyzed the incremental value provided by the temporal filtering component. Note that the optimal threshold for the VLM filtering differs between this setting and our proposed method; thus, we conducted separate threshold sweeps for both versions. The corresponding results are visualized in Figure 4. From this figure, we also observed that our complete method has a broader effective threshold range, indicating higher robustness against threshold variations.
3. **Proposed Method without Background Subtraction (VLM + temporal filtering + SAM 2, no BGS):** In this configuration, background subtraction was omitted. The hypothesis was that not eliminating stationary objects would lead to difficulties in both correctly identifying leaks and avoiding false positives from non-leak objects. To achieve the best possible performance under this configuration (since its modality is different from our configuration), a grid search was conducted on the enhancement factor and VLM threshold to determine the optimal settings. Details of the grid search are provided in the supplementary material.
4. **Proposed Method with Traditional Segmentation (BGS + VLM filtering + temporal filtering + Otsu[49]):** We replaced the powerful SAM 2 segmentation method with the conventional segmentation technique proposed by Otsu[49], combined with simple morphological operations. Given that our test scenario consists primarily of a clear white leakage region against a dark background, it might be possible to achieve reasonable segmentation results using simpler

methods. Thus, we evaluated this setting to establish the necessity and advantage of employing the more advanced SAM 2 segmentation algorithm.

5. **Complete Proposed Method (BGS + VLM filtering + temporal filtering + SAM 2):** This condition represents our complete proposed approach, integrating all the discussed components to achieve robust leakage detection and segmentation.

BGS	VLM Filtering	Temporal Filtering	Seg.	τ_{VLM}	Stationary Foreground I/P/R/FLA	Moving Foreground I/P/R/FLA	Overall I/P/R/FLA
✓			None	-	0.56/0.64/0.83/0.85	0.38/0.53/0.58/0.69	0.5/0.61/0.73/0.79
✓	✓		SAM 2	0.09	0.67/0.81/0.79/**0.88**	0.54/0.79/0.65/0.83	0.62/0.80/0.74/0.86
	✓	✓	SAM 2	0.19	0.22/0.39/0.28/0.57	0.46/0.65/0.59/0.74	0.31/0.49/0.4/0.63
✓	✓	✓	Trad.	0.12	0.57/**0.85**/0.65/0.83	0.35/**0.88**/0.37/0.72	0.49/**0.86**/0.55/0.79
✓	✓	✓	SAM 2	0.12	**0.70**/0.83/**0.82**/0.87	**0.69**/0.79/**0.84**/**0.92**	**0.69**/0.82/**0.82**/**0.89**

Table 3. Ablation study of different components with IoU (I), Precision (P), Recall (R), and Frame Level Accuracy (FLA). In the segmentation column (Seg.), traditional (Trad.) means Otsu[49] combined with morphological transformations. This analysis corresponds to our ablation study, detailed in Section 5.3.

Prompt	Stationary Foreground I/P/R/FLA	Moving Foreground I/P/R/FLA	Overall I/P/R/FLA
white gas	0.59/0.82/0.70/0.88	0.55/0.74/0.66/0.80	0.57/0.80/0.67/0.83
white plume	0.35/0.55/0.52/0.67	0.31/0.48/0.45/0.63	0.34/0.52/0.50/0.66
white steam	**0.71**/0.82/**0.84**/0.90	**0.70**/**0.83**/**0.82**/**0.91**	**0.69**/**0.83**/0.81/0.88
white methane leak...	0.70/0.82/0.83/0.89	0.62/0.75/0.77/0.89	0.67/0.79/0.81/0.89
methane gas leak	0.62/0.79/0.75/0.82	0.63/0.77/0.77/0.87	0.62/0.75/0.79/0.86
gas leak	0.62/0.79/0.76/0.83	0.57/0.75/0.70/0.86	0.60/0.77/0.74/0.84
white smoke	**0.71**/**0.83**/**0.84**/**0.91**	0.65/0.79/0.79/**0.91**	0.68/0.81/**0.82**/**0.91**

Table 4. Performance of Different Prompts on Stationary, Moving and Overall Backgrounds. The complete form of "The white methane leak..." is "white methane leak on black background in the infrared image." The prompts containing "white smoke" and "white steam" yielded the highest performance. In terms of overall performance, as measured by Intersection over Union (IoU), the prompt with "white steam" demonstrated a slight advantage over the prompt with "white smoke".

The ablation study shows that using only background subtraction (case 1) provides reasonable results, especially in cases where there is no moving foreground. However, when VLM filtering and Segmented Energy Model 2 are added (case 2) for segmentation, performance improves significantly, with an overall IOU increase of more than 10%. This enhancement allows for better filtering of non-leak objects.

In contrast, removing background subtraction (case 3) to detect moving objects leads to the worst performance, even lower than the baseline of case 1. Without background subtraction, the model struggles to identify leaks and correctly classify non-leak objects, as indicated by both low precision and recall.

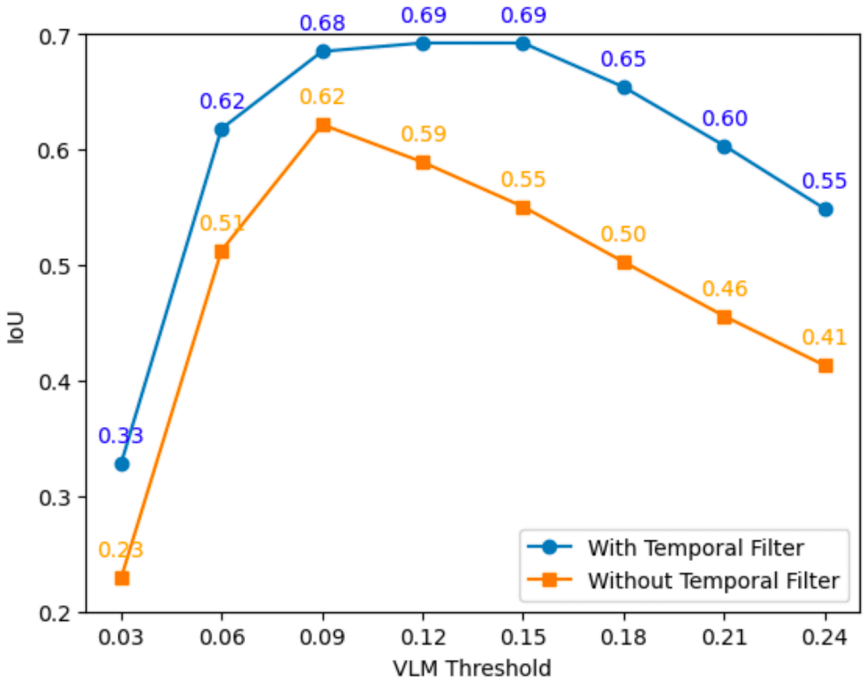

Figure 4. Performance of Method with and without Temporal Filter across Different VLM Thresholds (τ_{VLM}). The two configurations have different optimal τ_{VLM} . However, the method with the temporal filter achieves a higher best IoU. It demonstrates greater robustness to τ_{VLM} variations, maintaining strong performance over a broader threshold range (0.09–0.15), whereas the method without the temporal filter performs well only at 0.09.

Case 4 setup performed similarly to the background subtraction baseline but with a high precision and low recall. Upon visual inspection with a few images, we noted that some bounding boxes failed to encompass objects fully. Traditional segmentation could not extend the segmentation masks beyond bounding box limitations, whereas SAM 2 can use semantic information to capture entire objects.

With the full model (case 5), performance reaches 69% IOU, showing an almost 20% improvement over the MOG baseline and approximately a 40% boost compared to using only a visual language model, temporal filtering, and segmentation. Comparing this with case 2, where no temporal filtering is used, we achieved a 7% IoU boost in overall cases and a 15% increase in moving foreground. These two cases have a similar precision, but case 5 has a higher recall and higher frame-level accuracy. This could be due to the propagation mechanism in the filter. By comparing the performances of case 2 and case 5 across

different VLM Thresholds in Figure 4, we can also observe that with temporal filtering, the method is more robust to changes in the threshold.

5.4. Prompt Comparison

We hypothesized that the model might struggle to understand abstract or uncommon concepts such as “gas”, “methane,” and “leak.” To address this, we used “white steam” as a positive prompt in the object detection phase. To test our hypothesis, we experimented with different prompts. Since different prompts may have different optimal values τ_{VLM}, we performed a sweep across τ_{VLM} for each prompt. The results are shown in Figure 5.

In our comparison, “white steam” and “white smoke” achieved the best performance, which we attribute to their frequent occurrence in natural language. Additionally, the prompt “white methane leak ...” also performed well, likely due to its detailed description specifying the infrared modality and black background. However, “white gas” exhibited poor performance, which we hypothesize is because “gas” is generally invisible in RGB images, which is what most VLMs are trained on. For a more detailed analysis, please refer to the supplemental material.

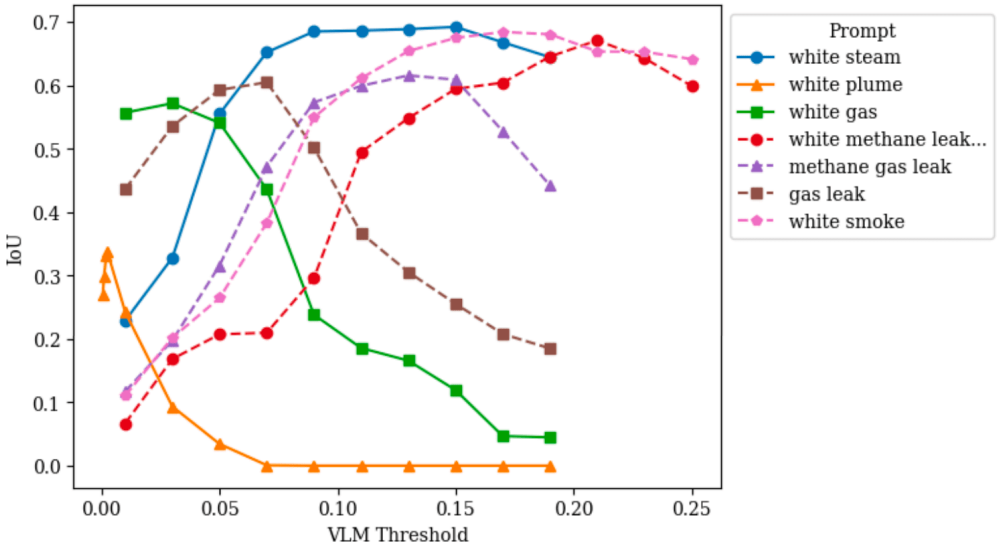

Figure 5. Performance of VLM with Different Prompts and Thresholds. The prompts with "white steam", "white smoke," and "white methane leak on black background in the infrared image" are the top performing prompts. Additionally, the prompt containing "white steam" demonstrates strong robustness, as its performance remains largely unchanged when the threshold varies between 0.10 and 0.15.

5.5. Qualitative Expenerments on GasVid

We performed qualitative experiments on GasVid to assess our method on real-world videos. Because high-quality mask annotations are unavailable, we evaluated the results visually. Please see the supplemental material for the results of the experiment.

6. Conclusion

In this work, we introduced a synthetic dataset with diverse backgrounds, interfering foreground objects, and precise segmentation of ground truth. To leverage this dataset, our proposed zero-shot method significantly improves segmentation performances. Our approach achieves an IoU of 69%, outperforming baseline methods relying solely on background subtraction or zero-shot object detection. Additionally, our analysis of prompt configurations and threshold settings provides further insights into optimizing segmentation performance. These findings highlight the potential of zero-shot learning in gas leakage detection and suggest future work in refining dataset quality and enhancing detection

robustness in real-world scenarios. Additionally, replacing background subtraction with the optical flow could enable adaptation to a wider range of scenarios, such as hand-held devices in the IIG dataset[5]. Moreover, this project could also be applied to fire detection, asteroid or exoplanet detection, and tracking microorganism movement under a microscope.

7. Methane Release From GasVid

The total amount of methane released during the capture of the GasVid dataset can be calculated using Equation 2, where m_{total} represents the total mass of methane released, n is the number of videos, i denotes the class label corresponding to the flow rate, and m_i is the flow rate of the i-th class in g/h. Each flow rate lasted for 3 minutes. Based on the flow rate data from the GasVid paper, the total methane release is 12906.385g.

$$\begin{aligned} m_{\text{total}} &= n \times \sum_{i=0}^{7} \left(\frac{m_i}{60} \times 3 \right) \\ &= \frac{31 \times 3}{60} \times \sum_{i=0}^{7} m_i \\ &= 1.55 \times 8326.7 \text{ g} \\ &= 12906.385 \text{ g} \end{aligned} \tag{2}$$

doi.org/10.32388/1HECO0

8. Further Review of VLMs

Algorithm 1: Temporal Filtering Algorithm

Input: *current boxes*, *past boxes*, *image size*
Output: *valid boxes*

```
Set valid boxes as an empty list;
for each current box in current boxes do
    if area of current box > image area × ignore large threshold then
        Skip this box;
    Set matched boxes to 0;
    for each past frame boxes in the last maximum past frames do
        Find overlap between current box and past frame boxes;
        Find position difference between current box and past frame boxes;
        if any overlap > IoU match threshold OR all position differences < absolute shift threshold then
            Increase matched boxes by 1;
    if matched boxes ≥ match threshold then
        Add current box to valid boxes;
Set matched boxes as an empty list;
if valid boxes is empty AND past boxes length ¿ 3 then
    for each first frame in the last 3 frames do
        for each second frame in the last 3 frames do
            if first frame == second frame then
                Skip this frame;
            for each box in first frame do
                if any overlap with boxes in second frame > IoU match threshold then
                    Add box in first frame to matched boxes;
return valid boxes;
```

CLIP[27] is a remarkable work that has inspired downstream work in segmentation, detection, etc. LSeg[50] adapted CLIP into segmentation by calculating the similarity of the text query with every pixel on the feature map of the image, classifying each pixel into one of the text queries. It then used a special

regulation block to decode the feature map into segmentations. This straightforward way has also been used in OwlVit[35] and Owl-V2[36]. In OwlVit, they pre-trained the CLIP encoder using contrastive loss and transferred the model into detection by removing the pooling operation with a classification and localization head to archive language-guided detection.

Besides the image-text contrastive loss function, align before fuse (ALBEF)[51] also used image-text matching and masked language modelling like in BERT[52]. Their model has an image encoder, a text encoder, and a multimodality encoder.

Although ALBEF was not trained on grounding or localization tasks, their Grad-CAM[53] has shown a strong localization correlation between phases and text. This is further improved by[39][40]. Grounding-DINO[37] and GLIP[34], on the other hand, are specifically trained on grounding tasks and trained in object detection fashion by producing bounding boxes for phases. Both the Grad-CAM and the bounding box can be used to prompt a segmentation model such as SAM[41], or SAM 2[47] to generate language-guided instance segmentation masks like in Grounding-SAM[42] and APOVIS[43].

Another line of work took a generative approach[28][29][31][32][30][54][55][56]. In these works, GPT-4 serials[28][29] and llama-like[31][32] models use pure language as an interface, take in instruction as text prompt and generate output as pure text (such as location information in coordination). Florence, on the other hand, uses special tokens for different tasks (such as segmentation, detection, etc) and also uses special tokens for generated results. Some other works[54][55][56] also used special tokens for segmentation results.

9. Qualitative Experiments on GasVid

We excluded videos recorded at 18.6 m (following VideoGasNet[7]) and selected examples showing two failure cases and two successful cases, as shown in Figure 6. The experiment used MOG2 as the background subtractor, OWLv2[36] as the visual language model with a threshold of 0.06, enhancement factor of 10, and both temporal filtering and SAM 2 enabled. The results indicate that the model can localize and segment leakage with reasonable performance, although worse than the synthetic dataset due to real-world noise, artifacts in background subtraction, etc. Future work should be done on how to improve this method on real-world captured videos.

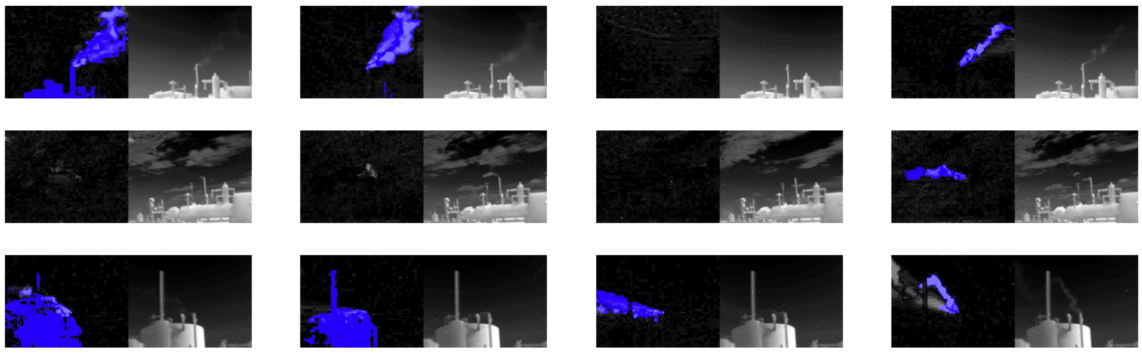

Figure 6. Selected Samples from GasVid: The two left columns display failure cases, and the two right columns show successful cases. In each pair, the left image shows the background subtraction result,t with blue indicating the segmentation output (artifacts may appear), while the right image is the original frame. The three rows correspond to videos with GasVid IDs 1239, 2570, and 2579, recorded at distances of 12.6m, 15.6m, and 6.9m, respectively.

In the success cases, two samples (from the third column and first two rows) are true negatives, showing that noise is not mistakenly segmented as a leak, while the remaining examples are true positives with well-aligned segmentation boundaries. In the sample in the fourth column of the third row, the model avoids an artifact from background subtraction that is not a leak. In the failure cases, the first and third videos show over-segmentation of non-leak objects, and in the second video, the leak is missed (false negative) due to the larger distance. We provided 4 full video results in the attached video.

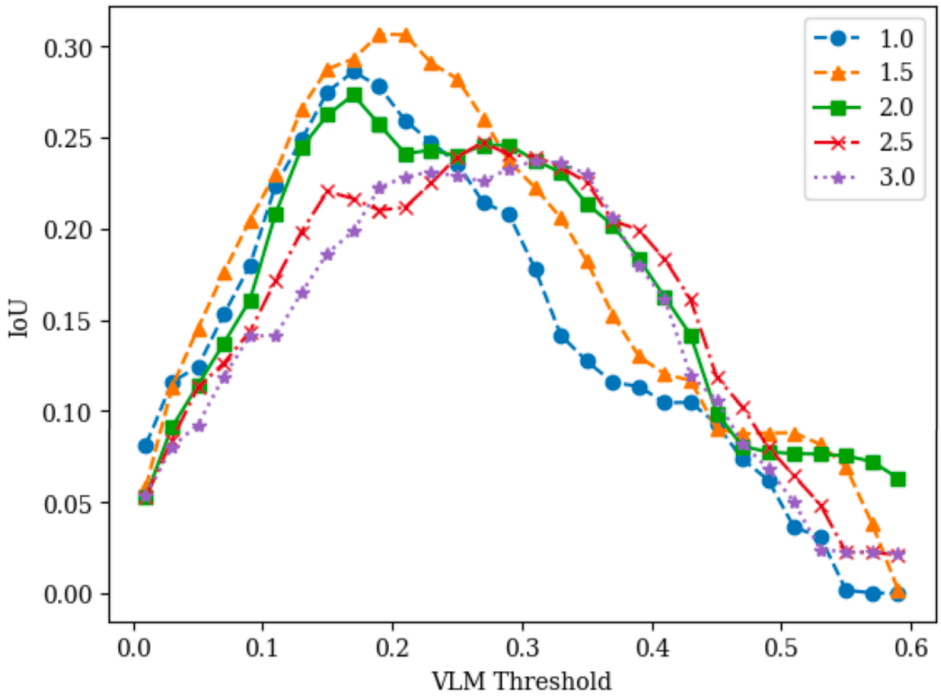

Figure 7. Grid Search For Configuration without Background Subtraction: We did a grid search on the enhancement factor and VLM threshold for the configuration without background subtraction. Different lines show different enhancement factors. The best performing point is when the enhancement factor is 1.5 and the VLM Threshold is 0.19. Results in this setting are values reported in Table 3.

10. Prompts Comparison

In our study on different prompts, “white steam” and “white smoke” performed the best, whereas “white plume” exhibited the worst performance. We hypothesize that the superior performance of “white steam” and “white smoke” is due to their explicit description of both the substance (smoke or steam) and its colour (white). In contrast, the poor performance of “white plume” is likely because “plume” is a relatively uncommon word.

Notably, the prompts “white gas” and “gas leak” also performed poorly. We attribute this to the fact that, in the training data of vision-language models (VLMs), “gas” is often associated with “gas station” rather than referring solely to a gaseous substance. As a result, the model may tend to link “gas” to “gas station” or “gas stove,” leading to suboptimal performance. Additionally, since gases are generally invisible in RGB

images, and RGB is likely the primary modality in the training dataset, the model may struggle to associate the term “gas” with its visual characteristics in infrared imagery. This suggests that the poor performance of prompts containing “gas” is likely due to a mismatch between the term’s associations in the training data and its expected visual representation in real-world scenarios.

Another notable observation is that the long prompt, “white methane leak on black background in the infrared image,” achieved near-optimal performance, only slightly worse than the best-performing prompts. We hypothesize that while the VLM may not have a strong understanding of “methane,” the explicit description of the black background and the infrared image modality provide sufficient context for the model to generate accurate outputs.

Acknowledgments

Thanks for the support by NFRF GR024473 and CFI GR024801.

Footnotes

[1] Note that this is not the same as background removal, which is to segment the saliency objects in a single frame image to remove the background; an example use case is to blur or replace the background in online video conferences. For background removal, readers can refer to[57][58].

[2] When we report the best results, we select the setting with the highest IoU, and report precision, recall, and frame level accuracy (FLA) in that setting.

References

1. ^*US EPA, OAR (2016). "Importance of Methane". https://www.epa.gov/gmi/importance-methane.*
2. ^*Curtis J, Metheny E, Sergent SR. Hydrocarbon Toxicity. Treasure Island (FL): StatPearls Publishing; 2025. A vailable from: http://www.ncbi.nlm.nih.gov/books/NBK499883/.*
3. a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q*Wang J, Tchapmi LP, Ravikumara AP, McGuire M, Bell CS, Zimmerle D, Sava rese S, Brandt AR (2019). "Machine vision for natural gas methane emissions detection using an infrared ca mera". arXiv. doi:10.48550/arXiv.1904.08500. Available from: http://arxiv.org/abs/1904.08500.*
4. a, b*Sarker TT, Embaby MG, Ahmed KR, Abughazaleh A. "Gasformer: A Transformer-based Architecture for S egmenting Methane Emissions from Livestock in Optical Gas Imaging." In: Proceedings of the IEEE/CVF Co nference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2024. p. 5489-5497.*

5. a, b, c, d, e, f Yu H, Wang J, Wang Z, Yang J, Huang K, Lu G, Deng F, Zhou Y (2024). "A lightweight network based on local–global feature fusion for real-time industrial invisible gas detection with infrared thermography". Applied Soft Computing. **152**: 111138. doi:10.1016/j.asoc.2023.111138. Link to article.
6. a, b, c, d, e, f Wang J, Lin Y, Zhao Q, Luo D, Chen S, Chen W, Peng X (2024). "Invisible Gas Detection: An RGB-Thermal Cross Attention Network and A New Benchmark". arXiv. doi:10.48550/arXiv.2403.17712. Available from: http://arxiv.org/abs/2403.17712.
7. a, b, c, d, e, f, g Wang J, Ji J, Ravikumar AP, Savarese S, Brandt AR (2022). "VideoGasNet: Deep learning for natural gas methane leak classification using an infrared camera". Energy. **238**: 121516. doi:10.1016/j.energy.2021.121516. Link to article.
8. ^ SHI X, Chen Z, Wang H, Yeung DY, Wong WK, WOO WC. "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting." In: Cortes C, Lawrence N, Lee D, Sugiyama M, Garnett R, editors. Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2015. **28**. Available from: https://proceedings.neurips.cc/paper_files/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf.
9. ^ Guo W, Du Y, Shehata M (2024). "3D2SMILES: Translating Physical Molecular Models into Digital DeepSMILES Notations Using Deep Learning". https://chemrxiv.org/engage/chemrxiv/article-details/673a9d62f9980725cf89abe1. doi:10.26434/chemrxiv-2024-zvcb4-v3.
10. ^ Wang G, Li H, Li P, Lang X, Feng Y, Ding Z, Xie S (2024). "M4SFWD: A Multi-Faceted synthetic dataset for remote sensing forest wildfires detection". Expert Systems with Applications. **248**: 123489. doi:10.1016/j.eswa.2024.123489. Link to article.
11. ^ Mao J, Zheng C, Yin J, Tian Y, Cui W (2021). "Wildfire smoke classification based on synthetic images and pixel- and feature-level domain adaptation". Sensors. **21** (23): 7785. doi:10.3390/s21237785. PMID 34883801.
12. ^ Zhu JY, Park T, Isola P, Efros AA. "Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks." In: Proceedings of the IEEE International Conference on Computer Vision (ICCV); 2017 Oct.
13. ^ GU X, LIN H, DING D, GU X (2023). "An infrared gas imaging and instance segmentation based gas leakage detection method". Journal of East China University of Science and Technology. **49** (1): 76–86. doi:10.14135/j.cnki.1006-3080.20210719001.
14. a, b Stauffer C, Grimson WEL. "Adaptive background mixture models for real-time tracking." In: Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149). Fort Collins, CO, USA: IEEE Comput. Soc; 1999. p. 246-252. doi:10.1109/CVPR.1999.784637. Link.
15. a, b, c, d Zivkovic Z. "Improved adaptive Gaussian mixture model for background subtraction." In: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004. Cambridge, UK: IEEE; 200

4. p. 28–31 Vol.2. doi:10.1109/ICPR.2004.1333992. Available from: http://ieeexplore.ieee.org/document/1333992/.

16. a, b, c, d, e, f, g *Zivkovic Z, Van Der Heijden F (2006). "Efficient adaptive density estimation per image pixel for the task of background subtraction". Pattern Recognition Letters. **27** (7): 773–780. doi:10.1016/j.patrec.2005.11.005. Link.*
17. a, b, c, d *Tezcan MO, Ishwar P, Konrad J (2020). "BSUV-Net: A Fully-Convolutional Neural Network for Background Subtraction of Unseen Videos". arXiv. doi:10.48550/arXiv.1907.11371. Available from: http://arxiv.org/abs/1907.11371.*
18. a, b *Tezcan MO, Ishwar P, Konrad J (2021). "BSUV-Net 2.0: Spatio-Temporal Data Augmentations for Video-Agnostic Supervised Background Subtraction". arXiv. doi:10.48550/arXiv.2101.09585. Available from: http://arxiv.org/abs/2101.09585.*
19. a, b *An Y, Zhao X, Yu T, Guo H, Zhao C, Tang M, Wang J (2023). "ZBS: Zero-shot Background Subtraction via Instance-level Background Modeling and Foreground Selection". arXiv. doi:10.48550/arXiv.2303.14679. Available from: http://arxiv.org/abs/2303.14679.*
20. a, b *Lim LA, Keles HY (2018). "Learning Multi-scale Features for Foreground Segmentation". arXiv. doi:10.48550/arXiv.1808.01477. arXiv:1808.01477.*
21. ^ *Lim LA, Keles HY (2018). "Foreground Segmentation Using a Triplet Convolutional Neural Network for Multiscale Feature Encoding". arXiv. doi:10.48550/arXiv.1801.02225. Available from: http://arxiv.org/abs/1801.02225.*
22. ^ *Lin C, Yan B, Tan W (2018). "Foreground Detection in Surveillance Video with Fully Convolutional Semantic Network". In: 2018 25th IEEE International Conference on Image Processing (ICIP), pages 4118–4122. doi:10.1109/ICIP.2018.8451816. Available from: https://ieeexplore.ieee.org/document/8451816. ISSN: 2381-8549.*
23. ^ *St-Charles PL, Bilodeau GA, Bergevin R (2015). "SuBSENSE: A Universal Change Detection Method With Local Adaptive Sensitivity". IEEE Transactions on Image Processing. **24** (1): 359–373. doi:10.1109/TIP.2014.2378053. Available online.*
24. ^ *Mandal M, Dhar V, Mishra A, Vipparthi SK (2019). "3DFR: A Swift 3D Feature Reductionist Framework for Scene Independent Change Detection". IEEE Signal Processing Letters. **26** (12): 1882–1886. doi:10.1109/LSP.2019.2952253. arXiv:1912.11891.*
25. ^ *Zhao C, Hu K, Basu A (2022). "Universal Background Subtraction Based on Arithmetic Distribution Neural Network". IEEE Transactions on Image Processing. **31**: 2934–2949. doi:10.1109/TIP.2022.3162961. Available from: https://ieeexplore.ieee.org/document/9749010.*

26. ^Yang Y, Ruan J, Zhang Y, Cheng X, Zhang Z, Xie G (2022). "STPNet: A Spatial-Temporal Propagation Network for Background Subtraction". IEEE Transactions on Circuits and Systems for Video Technology. **32** (4): 2145–2157. doi:10.1109/TCSVT.2021.3088130. Link.

27. a, b, cRadford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, Krueger G, Sutskever I (2021). "Learning transferable visual models from natural language supervision". arXiv. doi:10.48550/arXiv.2103.00020. Available from: http://arxiv.org/abs/2103.00020.

28. a, b, cOpenAI (2024). "GPT-4 Technical Report". arXiv. doi:10.48550/arXiv.2303.08774. arXiv:2303.08774 [cs].

29. a, b, cOpenAI (2024). "GPT-4o System Card". Available from: https://cdn.openai.com/gpt-4o-system-card.pdf.

30. a, b, cXiao B, Wu H, Xu W, Dai X, Hu H, Lu Y, Zeng M, Liu C, Yuan L (2023). "Florence-2: Advancing a unified representation for a variety of vision tasks". arXiv. doi:10.48550/arXiv.2311.06242. Available from: http://arxiv.org/abs/2311.06242.

31. a, b, cMeta AI. "The Llama 3 Herd of Models". arXiv. 2024 Nov. arXiv:2407.21783 [cs]. Available from: http://arxiv.org/abs/2407.21783. doi:10.48550/arXiv.2407.21783.

32. a, b, cLiu H, Li C, Wu Q, Lee YJ (2023). "Visual instruction tuning". arXiv. doi:10.48550/arXiv.2304.08485. Available from: http://arxiv.org/abs/2304.08485.

33. ^Li J, Li D, Xiong C, Hoi S (2022). "BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation". arXiv. doi:10.48550/arXiv.2201.12086. Available from: http://arxiv.org/abs/2201.12086.

34. a, bLi LH, Zhang P, Zhang H, Yang J, Li C, Zhong Y, Wang L, Yuan L, Zhang L, Hwang JN, Chang KW, Gao J (2022). "Grounded Language-Image Pre-training". arXiv. doi:10.48550/arXiv.2112.03857. Available from: http://arxiv.org/abs/2112.03857.

35. a, b, cMinderer M, Gritsenko A, Stone A, Neumann M, Weissenborn D, Dosovitskiy A, Mahendran A, Arnab A, Dehghani M, Shen Z, Wang X, Zhai X, Kipf T, Houlsby N. Simple open-vocabulary object detection with vision transformers. arXiv. 2022 Jul. Available from: http://arxiv.org/abs/2205.06230. doi:10.48550/arXiv.2205.06230.

36. a, b, c, d, eMinderer M, Gritsenko A, Houlsby N. Scaling open-vocabulary object detection. arXiv. 2024. doi:10.48550/arXiv.2306.09683. Available from: http://arxiv.org/abs/2306.09683.

37. a, b, cLiu S, Zeng Z, Ren T, Li F, Zhang H, Yang J, Jiang Q, Li C, Yang J, Su H, Zhu J, Zhang L (2024). "Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection". arXiv. doi:10.48550/arXiv.2303.05499. Available from: http://arxiv.org/abs/2303.05499.

38. ^Wang J, Yang Z, Hu X, Li L, Lin K, Gan Z, Liu Z, Liu C, Wang L (2022). "GIT: A Generative Image-to-text Transformer for Vision and Language". arXiv. doi:10.48550/arXiv.2205.14100. Available from: http://arxiv.org/abs/2205.14100.

39. a, bHe R, Cascante-Bonilla P, Yang Z, Berg AC, Ordonez V (2023). "Improved visual grounding through self-consistent explanations". arXiv. doi:10.48550/arXiv.2312.04554. Available from: http://arxiv.org/abs/2312.04554.

40. a, bYang Z, Kafle K, Dernoncourt F, Ordonez V (2024). "Improving visual grounding by encouraging consistent gradient-based explanations". arXiv. doi:10.48550/arXiv.2206.15462. Available from: http://arxiv.org/abs/2206.15462.

41. a, bKirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, Xiao T, Whitehead S, Berg AC, Lo WY, Dollár P, Girshick R. "Segment Anything". arXiv. 2023 Apr. Available from: http://arxiv.org/abs/2304.02643. doi:10.48550/arXiv.2304.02643.

42. a, bRen T, Liu S, Zeng A, Lin J, Li K, Cao H, Chen J, Huang X, Chen Y, Yan F, Zeng Z, Zhang H, Li F, Yang J, Li H, Jiang Q, Zhang L (2024). "Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks". arXiv. doi:10.48550/arXiv.2401.14159. Available from: http://arxiv.org/abs/2401.14159.

43. a, bMa Q, Yang S, Zhang L, Lan Q, Yang D, Chen H, Tan Y (2025). "APOVIS: Automated pixel-level open-vocabulary instance segmentation through integration of pre-trained vision-language models and foundational segmentation models". Image and Vision Computing. **154**: 105384. doi:10.1016/j.imavis.2024.105384. Available from: https://www.sciencedirect.com/science/article/pii/S026288562400489X.

44. ^Wu Z, et al. A thermal infrared video benchmark for visual analysis. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops. IEEE Xplore; 2014. p. 201-208. doi:10.1109/CVPRW.2014.39.

45. ^Saha P, Mudassar BA, Mukhopadhyay S. "Adaptive Control of Camera Modality with Deep Neural Network-Based Feedback for Efficient Object Tracking". In: IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS); 2018.

46. ^Gebhardt E, Wolf M (2018). "CAMEL Dataset for Visual and Thermal Infrared Multiple Object Detection and Tracking". In: IEEE International Conference on Advanced Video and Signal-based Surveillance (AVSS).

47. a, b, c, d, eRavi N, Gabeur V, Hu YT, Hu R, Ryali C, Ma T, Khedr H, Rädle R, Rolland C, Gustafson L, Mintun E, Pan J, Alwala KV, Carion N, Wu CY, Girshick R, Dollár P, Feichtenhofer C (2024). "SAM 2: Segment Anything in Images and Videos". arXiv. doi:10.48550/arXiv.2408.00714. Available from: http://arxiv.org/abs/2408.00714.

48. a, bRamesh A, Dhariwal P, Nichol A, Chu C, Chen M (2022). "Hierarchical text-conditional image generation with clip latents". arXiv. **arXiv:2204.06125**. doi:10.48550/arXiv.2204.06125. Available from: http://arxiv.org/abs/2204.06125.

49. ^a, b, cOtsu N (1979). "A threshold selection method from gray-level histograms". IEEE Transactions on Systems, Man, and Cybernetics. **9** (1): 62–66. doi:10.1109/TSMC.1979.4310076.

50. ^Li B, Weinberger KQ, Belongie S, Koltun V, Ranftl R (2022). "Language-driven Semantic Segmentation". arXiv. doi:10.48550/arXiv.2201.03546. Available from: http://arxiv.org/abs/2201.03546.

51. ^Li J, Selvaraju RR, Gotmare AD, Joty S, Xiong C, Hoi S (2021). "Align before Fuse: Vision and Language Representation Learning with Momentum Distillation". arXiv. **arXiv:2107.07651** [cs]. Available from: http://arxiv.org/abs/2107.07651. doi:10.48550/arXiv.2107.07651.

52. ^Devlin J, Chang MW, Lee K, Toutanova K (2019). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". arXiv. doi:10.48550/arXiv.1810.04805. arXiv:1810.04805.

53. ^Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2019). "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization". arXiv. doi:10.48550/arXiv.1610.02391. arXiv:1610.02391.

54. ^a, bLai X, Tian Z, Chen Y, Li Y, Yuan Y, Liu S, Jia J (2024). "LISA: Reasoning Segmentation via Large Language Model". arXiv. doi:10.48550/arXiv.2308.00692. Available from: http://arxiv.org/abs/2308.00692.

55. ^a, bWang J, Ke L (2024). "LLM-Seg: Bridging Image Segmentation and Large Language Model Reasoning". arXiv. doi:10.48550/arXiv.2404.08767. Available from: http://arxiv.org/abs/2404.08767.

56. ^a, bBai Z, He T, Mei H, Wang P, Gao Z, Chen J, Liu L, Zhang Z, Shou MZ (2024). "One Token to Seg Them All: Language Instructed Reasoning Segmentation in Videos". arXiv. doi:10.48550/arXiv.2409.19603. Available from: http://arxiv.org/abs/2409.19603.

57. ^Qin X, Dai H, Hu X, Fan DP, Shao L, Van Gool L (2022). "Highly accurate dichotomous image segmentation". arXiv. doi:10.48550/arXiv.2203.03041. Available from: http://arxiv.org/abs/2203.03041.

58. ^Qin X, Zhang Z, Huang C, Dehghan M, Zaiane OR, Jagersand M (2020). "U^2-Net: Going Deeper with Nested U-Structure for Salient Object Detection". Pattern Recognition. **106**: 107404. doi:10.1016/j.patcog.2020.107404. arXiv:2005.09007 [cs].

Declarations

Funding: Thanks for the support by NFRF GR024473 and CFI GR024801.

Potential competing interests: No potential competing interests to declare.