

Article

AI-Generated Fire Images for Object Detection-Based Fire Detection

Wangeun Ji ^{1,†}, Sugi Choi ^{2,†}, Heejun Kwon ² and Haiyoung Jung ^{2,*}¹ Department of Electrical Engineering, Inha University, 100 Inha-ro, Michuhol-gu, Incheon 22212, Republic of Korea; 22222037@inha.edu² Department of Fire and Disaster Prevention, Semyung University, 65 Semyung-ro, Jecheon-si 27136, Republic of Korea; zz55cc@naver.com (S.C.); he2jun@naver.com (H.K.)

* Correspondence: hyjung@semyung.ac.kr

† These authors contributed equally to this work as first author.

Abstract

Vision-based fire detection models are often limited by the insufficient diversity of annotated fire and smoke images, particularly in terms of fire location, flame scale, smoke density, ignition cause, and indoor scene context. This study investigates whether generative AI-based synthetic images can expand fire-image diversity and improve object detection-based fire detection performance. Real fire images were combined with conventional augmented images and synthetic images generated using ChatGPT-4.o and ChatGPT-5.5. The generated images were constructed using multivariable prompts considering fire location, scale, and cause, and unsuitable samples were screened using a pretrained fire detection model. YOLOv8n, YOLOv11n, and RT-DETR were trained under 48 dataset–detector conditions and evaluated using fixed validation and test datasets. The results showed that generated-image-based training generally maintained or improved detection performance compared with the original and conventional augmentation conditions. In particular, selected ChatGPT-4.o-based YOLOv11 conditions showed statistically supported improvements over matched augmentation conditions, with increases of +0.052 in Precision, +0.031 in Recall, +0.065 in mAP@0.5, and +0.038 in mAP@0.5:0.95. LPIPS and t-SNE analyses indicated that the generated images formed structured perceptual and feature-space distributions relative to real fire images. Scenario-based inference using location-specific video frames also showed stable model responses in several complex indoor fire environments. These findings suggest that validated generative AI-based images can supplement the limited visual diversity of real fire datasets and improve the robustness of vision-based fire detection models.

Keywords: fire detection; object detection; generative AI; synthetic fire image; RT-DETRAcademic Editors: Tao Chen,
Hongyong Yuan and Xiaole Zhang

Received: 11 March 2026

Revised: 19 June 2026

Accepted: 26 June 2026

Published: 2 July 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Recent research on object detection models has focused on improving real-time object detection performance by modifying and optimizing the backbone, neck, and head structures of deep learning-based object detection models, including You Only Look Once (YOLO), Detection Transformer (DETR), and Faster Region-based Convolutional Neural Network (Faster R-CNN) [1–3]. These models have been increasingly adopted in various applications, including license plate detection, product quality monitoring, and fruit

freshness classification [4–10]. This widespread application may be attributed to the availability and abundance of relevant datasets. However, the potential to enhance performance by analyzing factors such as the quality and quantity of image data, which are critical for optimizing the performance of artificial intelligence (AI) models, has not been fully explored. Particularly, research on AI-based fire detection is constrained by the scarcity of relevant datasets featuring flame and smoke images from forests, buildings, and indoor fires, which are key elements in disaster scenarios.

Various studies have analyzed the use of image augmentation and transfer learning to overcome this limitation [11–13]. Despite its effectiveness, image augmentation may lead to overfitting due to the similarity of augmented images, and transfer learning cannot enhance performance beyond a certain threshold.

Recent studies have addressed the limitations of fire-image datasets mainly through two approaches: improving detection architectures and increasing training-data diversity. In terms of model architecture, Fire-YOLO improved the detection of small flame and smoke targets by modifying YOLOv3 for fire inspection scenarios [14]. Xu et al. proposed Light-YOLOv5, a lightweight fire detection model designed for complex fire scenarios, by incorporating a separable vision transformer block, a lightweight bidirectional feature pyramid network, and attention-based feature enhancement [15]. Zhao et al. developed the Fire Segmentation–Detection Framework (FSDF), which combines flame feature enhancement with YOLOv8 and VQ-VAE to improve fire detection performance [16]. These studies show that architectural refinement can enhance the detection of small or visually ambiguous fire objects. However, the improvement obtained from model modification remains closely related to the diversity and representativeness of the training dataset.

Dataset-oriented studies have also emphasized the importance of data diversity for robust fire detection. Recent review papers have reported that the lack of large-scale annotated fire and smoke datasets remains a major obstacle to developing reliable vision-based fire detection systems [17,18]. To address this issue, Wu et al. introduced a fire and smoke object detection dataset containing fire, smoke, and fire-like objects [19], while Dewangan et al. proposed FIgLib and SmokeyNet for real-time wildland smoke detection using fixed-view camera imagery [20]. These datasets have contributed to benchmark construction and model evaluation, but real fire data collection is still difficult because fire events are hazardous, unpredictable, and highly dependent on scene conditions such as location, camera viewpoint, fire scale, and smoke density. In addition, the annotation of flame and smoke objects requires considerable effort because their boundaries are often irregular, blurred, or partially occluded. To reduce the dependency on real-image collection, synthetic data have been investigated in several object detection studies. Park et al. generated synthetic wildfire images using a GAN-based approach combined with weakly supervised object localization [21]. Ljungqvist et al. analyzed differences between detectors trained using synthetic and real images and showed that synthetic data can influence the internal representation of object detection networks [22]. Kiefer et al. also demonstrated that synthetic data can be useful for object detection in unmanned aerial vehicle scenarios where real data collection is costly or restricted [23]. These studies suggest that synthetic data can alleviate dataset scarcity, but they also indicate that synthetic images should be carefully validated because the distributional gap between real and synthetic images may affect detection performance. Recent advances in text-to-image generation provide a new opportunity for fire-image dataset expansion. DALL·E 3 improved prompt-following capability through detailed caption-based training, enabling users to generate images that reflect specific scene descriptions and visual attributes [24]. In fire detection, this capability is useful because fire scenarios can be defined by combinations of location, fire scale, ignition cause, smoke density, and surrounding objects. However, generated images should not be used without screening because unrealistic or irrelevant samples

may degrade training quality. Moreover, the effectiveness of generated-image-based training should be examined across different detector architectures. RT-DETR provides a useful comparison model because it differs from YOLO-based dense prediction models by using a query-based end-to-end detection mechanism [25]. The results of the comparison between these object detection-based fire and smoke detection and synthetic data-based research are shown in Table 1.

Previous studies have improved fire detection performance mainly by modifying object detection architectures, enhancing fire-related visual features, or constructing fire and smoke datasets. These studies demonstrate the effectiveness of model improvement and dataset expansion for fire and smoke detection. However, they have not sufficiently examined whether generative AI-based synthetic fire images can be systematically generated, screened, and incorporated into object detection training under identical evaluation conditions.

Table 1. Comparison of object detection-based fire and smoke detection and synthetic-data studies.

Models	Evaluation Matrix	Dataset	Input Class	Result	Reference
Fire-YOLO	Accuracy, F1, mAP	Fire small target dataset	Flame, Smoke	0.915, 0.730, 0.802	Zhao et al. [14]
Light-YOLOv5	mAP, FPS	Public + Web Dataset	Flame, Smoke	+3.3%, 91.1%	Xu et al. [15]
FSDF	F1-score, Accuracy	Forest + Urban Fire Dataset	Fire	94.2%, 93.5%	Zhao et al. [16]
GAN + WSOL	F1-score, AP	Wildfire Dataset	Wildfire	+71.9%, +6.41%	Park et al. [21]
YOLOv3	mAP	BDD + GTAV Dataset	Vehicle	0.890	Ljungqvist et al. [22]
Real/Synthetic					

Based on these considerations, this study investigates whether generative AI-based synthetic fire images can improve object detection-based fire detection performance when they are systematically generated, screened, and incorporated into the training dataset. Unlike previous studies that primarily focused on model-architecture modification, real-dataset construction, or conventional synthetic-data generation, this study compares GPT-4o- and GPT-5.5-generated image groups with conventional augmentation under identical validation and test dataset conditions. The generated images were constructed using multivariable prompts considering location, fire scale, smoke condition, and ignition cause, and were screened using a pretrained fire detection model before training. The resulting datasets were used to train YOLOv8n, YOLOv11n, and RT-DETR models under 48 detector–dataset conditions.

The main contributions of this study are as follows:

- We propose a generative AI-assisted fire-image dataset expansion pipeline that combines multivariable prompt engineering, generated-image screening, and object detection model training.
- We compare ChatGPT-4.o- and ChatGPT-5.5-generated image groups with conventional augmentation using YOLOv8n, YOLOv11n, and RT-DETR under the same validation and test dataset conditions.
- We verify the effectiveness and validity of generated-image-based training through test-dataset evaluation, bootstrap-based statistical analysis, LPIPS and t-SNE analyses, and scenario-based inference using location-specific fire video frames.

The remainder of this paper is organized as follows. Section 2 describes the research content and methods, including dataset construction, synthetic image generation, detection model training, and evaluation procedures. Section 3 presents the experimental results based on validation and test datasets. Section 4 evaluates the practical inference behavior of the developed models in diverse real-world fire scenarios. Section 5 discusses

the validity, implications, and limitations of the generated images. Finally, Section 6 concludes the paper and outlines future research directions.

2. Research Content and Methods

Figure 1 illustrates the overall workflow of this study. The proposed framework was designed to evaluate whether generative AI-based synthetic fire images can improve object detection-based fire detection performance when they are systematically generated, screened, and incorporated into the training dataset. The workflow consists of real fire-image dataset construction, conventional image augmentation, synthetic fire-image generation, generated-image screening, object detection model training, test-dataset evaluation, statistical verification, and scenario-based inference analysis. The resulting datasets were used to train YOLOv8n, YOLOv11n, and RT-DETR models under identical validation and test dataset conditions.

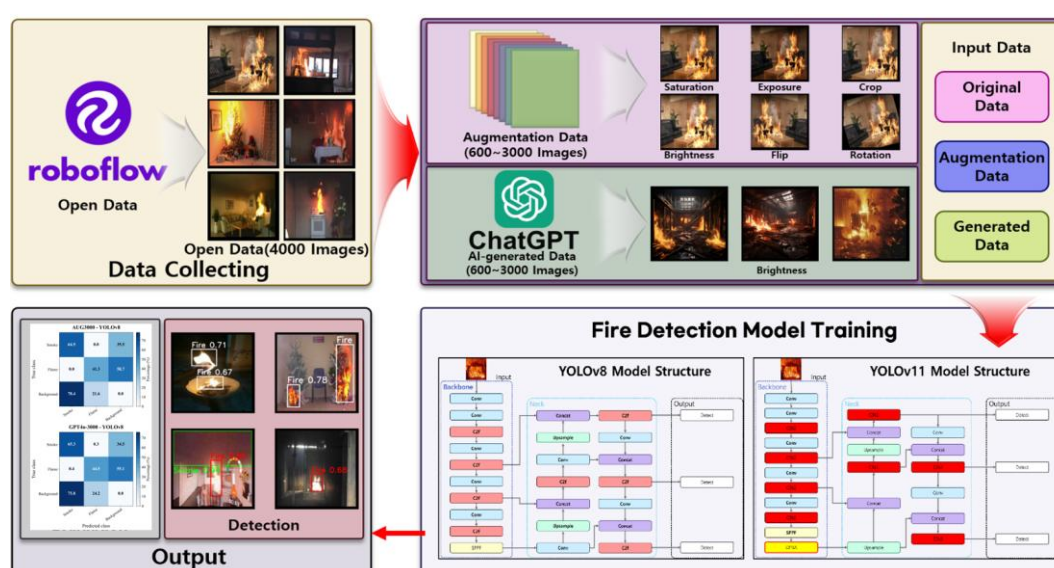


Figure 1. Development Process of the Proposed Fire Detection Model.

The following subsections describe the detailed procedures of the proposed framework. Section 2.1 introduces the detection models used in this study, including YOLOv8n, YOLOv11n, and RT-DETR. Sections 2.2–2.4 describe the construction of the real fire-image dataset, conventional augmentation, and synthetic fire-image generation and screening. Section 2.5 presents the experimental conditions and evaluation metrics, and Section 2.6 describes the generated-image validity analysis based on LPIPS and t-SNE.

2.1. Detection Models

This study employed YOLOv8n, YOLOv11n, and RT-DETR to evaluate whether the effect of generated-image-based training varies depending on detector architecture. The YOLO series has been widely adopted for real-time object detection because it provides a practical balance between detection accuracy, computational cost, and inference speed [26]. YOLOv8 is an anchor-free one-stage detector that improves multiscale feature representation through its backbone–neck–head structure. Its lightweight variant, YOLOv8n, was selected as a baseline detector because real-time fire detection requires both computational efficiency and stable detection performance. Recent fire and smoke detection studies have also shown that fine-tuned YOLOv8 models are effective for detecting flame and smoke objects under diverse visual conditions, supporting the use of YOLOv8 as a representative detection architecture in this study [27,28]. YOLOv11n was additionally evalu-

ated as a recent YOLO-based architecture. YOLOv11 introduces structural components such as C3k2 and C2PSA, which are designed to improve feature extraction efficiency and spatial attention capability [2]. These characteristics are relevant to fire detection because flame and smoke objects often appear with irregular shapes, scale variations, blurred boundaries, and low-contrast visual patterns. Thus, YOLOv11n was used to examine whether the generated-image-based training strategy remains effective in a more recent lightweight YOLO architecture.

In addition to the YOLO-based detectors, RT-DETR was included as a transformer-based real-time object detector. Unlike YOLO-based dense prediction models, RT-DETR performs end-to-end object detection using a query-based mechanism and an efficient hybrid encoder [25]. This architecture enables comparison between one-stage YOLO detectors and a transformer-based detection framework under the same dataset conditions. By evaluating YOLOv8n, YOLOv11n, and RT-DETR together, this study examined whether the contribution of generated synthetic fire images was limited to a specific detector type or could be observed across different object detection frameworks. Figure 2 shows the Architectures of YOLOv8n, YOLOv11n, and RT-DETR

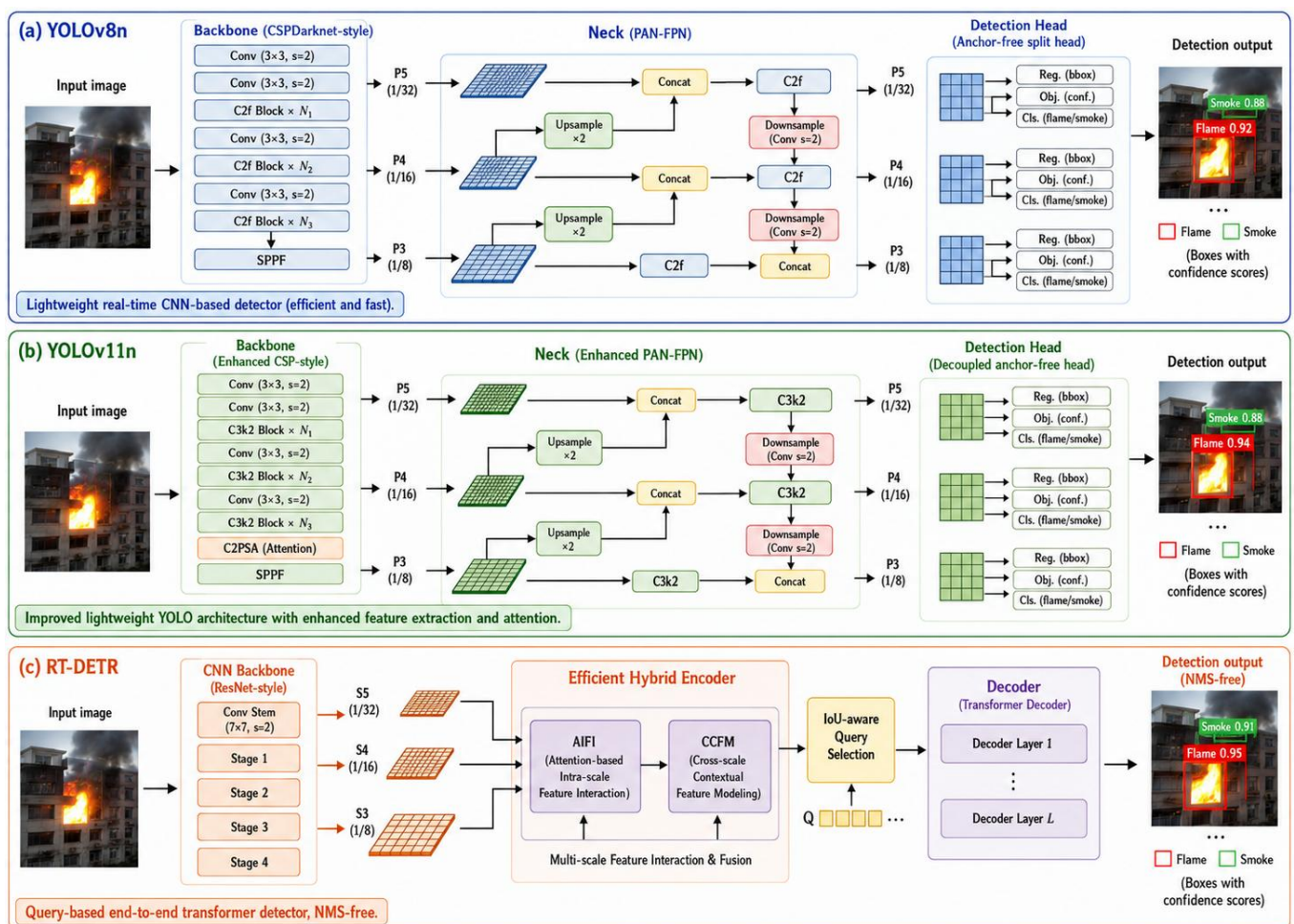


Figure 2. Architectures of YOLOv8n, YOLOv11n, and RT-DETR.

2.2. Real Fire Image Dataset Construction

In this study, fire-related images were collected from several online communities and open-data platforms, including Kaggle and Roboflow, to reduce overfitting that may occur when continuous or visually similar image data are used. Kaggle is a global platform for data science and machine learning that provides diverse datasets contributed by re-

searchers and practitioners worldwide. Roboflow is a computer vision dataset management platform that provides open datasets for various object detection tasks.

For this study, 4000 fire-related images were collected from public platforms, including Roboflow and Kaggle, and data provided by the Korea National Fire Agency [29–44]. Wildfire imagery was excluded because this study focused on indoor and structural fire detection involving flame and smoke objects. Duplicate and visually redundant images were removed, and the remaining images were curated. Label annotations provided by open-data sources were not used. Instead, all images were manually re-labeled in-house, assigning separate bounding-box annotations for flame and smoke. Of the 4000 images, 3000 images were allocated to the training set, 500 images to the validation set, and 500 images to the test set. Figure 3 presents examples of the collected real fire images.



Figure 3. Examples of Real Fire Image Dataset.

2.3. Conventional Image Augmentation

This study used various image-augmentation techniques to enhance the diversity of the training dataset and construct conventional comparison groups. Image augmentation artificially increases the diversity and size of a dataset by applying transformations to existing images [45]. The augmentation techniques applied in this study included saturation adjustment, brightness adjustment, exposure adjustment, flipping, cropping, and rotation. As shown in Table 2, 600 augmented images were generated for each augmentation stage, with 100 images assigned to each transformation category. These augmented images were progressively added to the original training dataset, resulting in AUG 600, AUG 1200, AUG 1800, AUG 2400, and AUG 3000 conditions.

Table 2. Conventional Image Augmentation Settings.

Name	Details	Number	Total
Augmentation	Saturation ($\pm 10\%$)	100	600
	Brightness ($\pm 10\%$)	100	
	Exposure ($\pm 10\%$)	100	
	Flip (Horizontal)	100	
	Crop ($\pm 10\%$)	100	
	Rotation ($\pm 10^\circ$)	100	

2.4. Synthetic Fire Dataset Generation and Screening

Synthetic fire and smoke images were generated to supplement the limited diversity of real-world fire image datasets. Two generative AI-based image-generation workflows, GPT-4o and GPT-5.5, were used to construct independent synthetic-image groups. To ensure that the generated images reflected fire-relevant visual conditions, the prompts were formulated using predefined scenario variables rather than simple free-form descriptions.

Three representative prompt templates were designed according to the major scene categories included in the generated dataset: industrial/logistics facilities, residential/commercial indoor spaces, and public facilities. These categories correspond to the representative examples shown in Figure 4, including logistics centers, factories, living rooms, res-

taurants, and airport terminals. Each prompt was composed by combining scene location, fire scale, ignition cause, smoke condition, surrounding objects, and camera viewpoint.

The prompt variables were defined as follows. Fire scale was expressed as the approximate proportion of visible flame area in the image and was discretized into ten levels ranging from less than 3% to 80%. Ignition causes included electrical short circuit, machinery overheating, cooking-related overheating, EV battery thermal runaway, heating-appliance failure, arson, discarded cigarette embers, and combustible-material ignition. Smoke condition and camera viewpoint were additionally specified when necessary to reflect different visibility levels and surveillance environments.

Using these variable combinations, 3000 synthetic images were generated using GPT-4o and another 3000 images were generated using GPT-5.5. The generated images were then subjected to an object detection-based verification procedure before being incorporated into the training dataset. Images were retained only when the target fire-related objects, namely flame and/or smoke depending on the prompt condition, were detected with a confidence threshold greater than 0.25. Images showing missing fire-related objects, severe mismatch with the intended scene, unrealistic flame or smoke appearance, cartoon-like rendering, duplicated composition, or irrelevant non-fire content were excluded.

Table 3 summarizes the representative prompt templates, variable definitions, and screening criteria used for synthetic fire-image generation. The accepted synthetic images were added to the GPT-4o and GPT-5.5 supplementary training sets and evaluated against conventional augmentation under identical validation and test conditions using YOLOv8n, YOLOv11n, and RT-DETR. In addition, examples of images generated in various locations are shown in Figure 4.

Table 3. Prompt templates, variable definitions and screening criteria for synthetic image generation.

Stage	Example
Template 1: Industrial/logistics facility	A high-resolution photographic image of an industrial or logistics facility. A small to medium-scale fire caused by an electrical short circuit or machinery overheating occupies approximately 10–30% of the scene. Flames and smoke are visible around shelves, pallets, production equipment, pipes, or electrical facilities.
Template 2: Residential/commercial indoor space	A high-resolution photographic image of a living room, restaurant, or commercial kitchen. A small fire caused by an electrical short circuit, cooking-related overheating, or combustible-material ignition occupies approximately 5–20% of the scene. Flames and moderate smoke are visible around furniture, kitchen appliances, tables, curtains, or interior fixtures.
Template 3: Public facility	A high-resolution photographic image of a large public indoor facility, such as an airport terminal or transit concourse. A small to medium-scale fire caused by an electrical short circuit or equipment failure occupies approximately 5–20% of the scene. Flames and smoke are visible around seats, signage, equipment, or terminal facilities.
Variable definitions	Location: indoor and structural fire-related environments; Fire scale: <3% to 80% of the image area; Ignition cause: electrical short circuit, machinery overheating, cooking-related overheating, EV battery thermal runaway, arson, discarded cigarette embers, or combustible-material ignition; Optional variables: smoke density, camera viewpoint, lighting condition, and surrounding objects.
Screening criteria	Generated images were retained only when flame and/or smoke objects were confirmed through an object detection-based verification procedure with a confidence threshold greater than 0.25. Images with missing fire-related objects, severe scene mismatch, unrealistic rendering, duplicated composition, or irrelevant non-fire content were excluded.

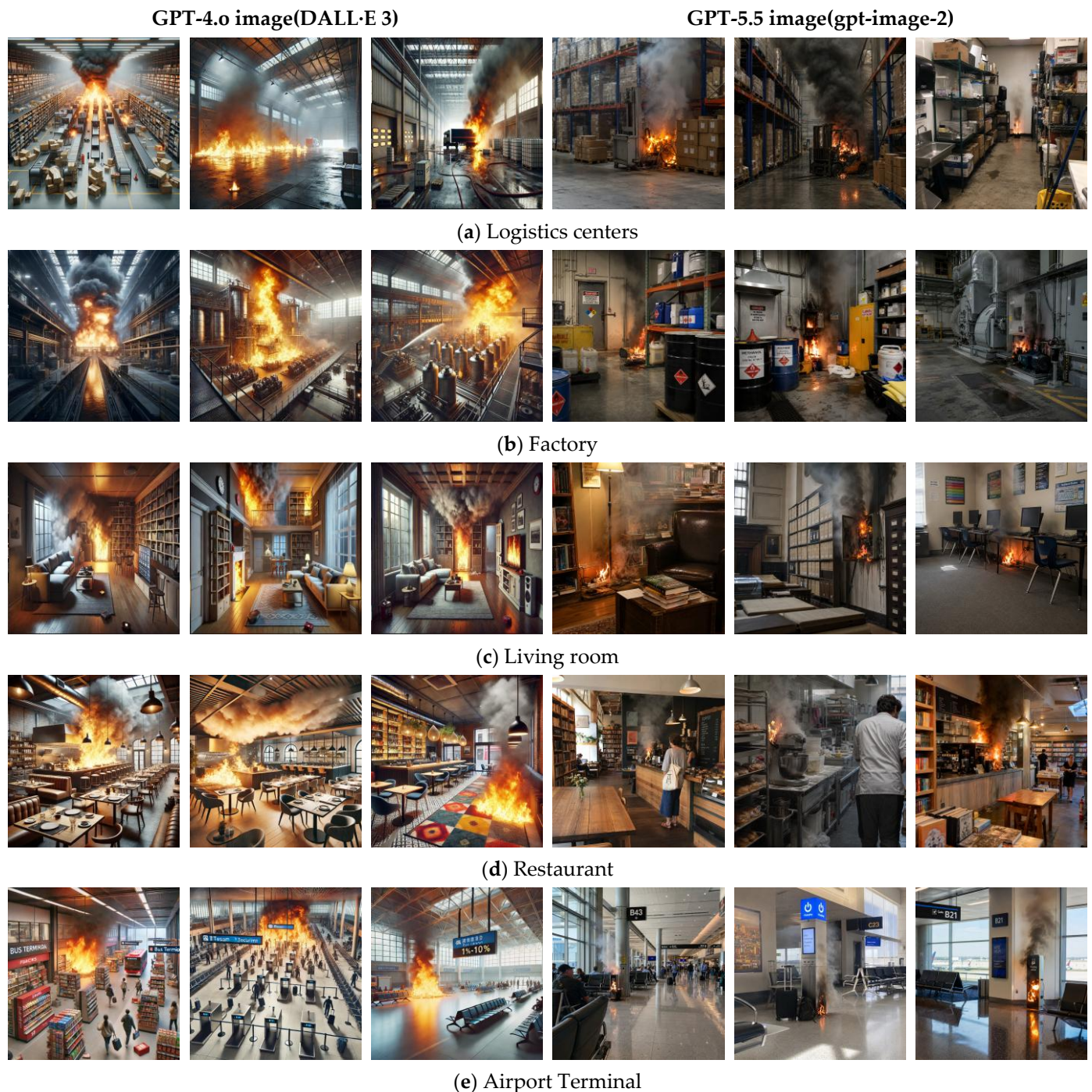


Figure 4. Examples of Synthetic Fire Dataset Generation Images. Dataset integration, Accepted images were incorporated into GPT-4o and GPT-5.5 supplementary training sets and compared with conventional augmentation under identical validation and test dataset conditions using YOLOv8n, YOLOv11n, and RT-DETR.

2.5. Model Evaluation Metrics and Experimental Conditions

Table 4 presents the hardware environment and hyperparameter settings used for training the object detection models. The experiments were conducted using YOLOv8n, YOLOv11n, and RT-DETR under the same training conditions. The same hyperparameter settings were applied across detector architectures and dataset conditions to ensure a fair comparison. The training and validation of the models were performed on a system equipped with a 10-core central processing unit, 128 GB Random Access Memory, and a graphics processing unit (GPU) based on the Nvidia RTX 5090. The deep-learning soft-

ware environment consisted of Compute Unified Device Architecture 12.8 and PyTorch 2.7.1, which were used to train the object detection models. The hyperparameters were set to Stochastic Gradient Descent with a learning rate of 0.01, 300 epochs, and a batch size of 16, corresponding to the default configuration for YOLOv8n. This setup was selected to ensure that each model was trained under comparable conditions by leveraging a GPU to achieve optimal performance.

Table 4. Hyperparameter and Hardware Settings of Network Training.

Hyperparameter		Hardware	
Training Parameters	Details	Name	Version
Epochs	300	Pytorch	2.7.1
Batch size	16	CUDA	12.8
Image size (Pixels)	640 × 640	CPU	10core
Optimizer algorithm	SGD	RAM	128
Learning rate	0.01	GPU	NVIDIA RTX 5090

Table 5 summarizes the number of input images used for each dataset condition. The Original condition consisted of 3000 training images, 500 validation images, and 500 test images. For the AUG, ChatGPT-4.o, and ChatGPT-5.5 conditions, 600, 1200, 1800, 2400, or 3000 additional images were progressively added to the original training set. Therefore, the total number of training images increased from 3000 to 6000 depending on the dataset condition. For each detector architecture, 16 dataset conditions were evaluated: Original, five AUG conditions, five ChatGPT-4.o conditions, and five ChatGPT-5.5 conditions. Since three detector architectures were used, a total of 48 training events were conducted.

Table 5. Number of Input Images by Dataset Condition.

Dataset Condition	Training Images	Validation Images	Test Images
Original	3000	500	500
AUG 600	3600	500	500
AUG 1200	4200	500	500
AUG 1800	4800	500	500
AUG 2400	5400	500	500
AUG 3000	6000	500	500
ChatGPT-4.o 600	3600	500	500
ChatGPT-4.o 1200	4200	500	500
ChatGPT-4.o 1800	4800	500	500
ChatGPT-4.o 2400	5400	500	500
ChatGPT-4.o 3000	6000	500	500
ChatGPT-5.5 600	3600	500	500
ChatGPT-5.5 1200	4200	500	500
ChatGPT-5.5 1800	4800	500	500
ChatGPT-5.5 2400	5400	500	500
ChatGPT-5.5 3000	6000	500	500

The performance of the model was evaluated using object detection metrics, including precision, recall, F1-score, and mean average precision (*mAP*). The key metrics of precision, recall, and F1-score are as follows: true positives (*TPs*) represent correctly identified positive instances, false positives (*FPS*) are negative instances incorrectly identified as positive, and false negatives (*FNs*) are positive instances incorrectly identified as negative. Precision and recall are defined using Equations (1) and (2), respectively. The *F1-score*, which represents the harmonic mean of precision and recall, is defined in Equation (3).

The average precision (AP) was calculated as the area under the precision-recall curve, as Equation (4) presents, and mAP was the average AP across all classes, as defined in Equation (5). Here, mAP@0.5 refers to the AP at an Intersection over Union (IoU) threshold of 0.5 for the test dataset. where $p(r)$ in Equation (4) is the maximum precision at a given recall, C in Equation (5) is the total number of classes.

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

$$F1_Score = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \quad (3)$$

$$AP = \int_0^1 p(r) dr \quad (4)$$

$$mAP = \frac{1}{C} \sum_{i=1}^C AP_i \quad (5)$$

To verify whether the selected performance differences were statistically reliable, paired bootstrap analysis was conducted using the test dataset [46]. Because object detection metrics, particularly mAP, are calculated over the entire test set rather than obtained as independent scalar values for individual images, test-image-level bootstrap resampling was adopted. In each bootstrap iteration, test images were resampled with replacement, and the performance difference between the generator-based model and the corresponding conventional augmentation model was recalculated. The 95% confidence interval of the performance difference was estimated using the percentile method. A performance improvement was considered statistically supported when the 95% confidence interval did not include zero.

2.6. Generated-Image Validity Analysis

LPIPS and t-SNE analyses were conducted to examine the validity of the generated fire images from perceptual and feature-distribution perspectives. LPIPS was used to quantify the perceptual distance between real and generated images in deep feature space [47]. Given two images, x and x_0 , the LPIPS distance is calculated using the weighted difference between normalized deep feature activations, as shown in Equation (6):

$$d(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \|\hat{y}_{hw}^l - \hat{y}_{0hw}^l\|_2^2 \quad (6)$$

where $d(x, x_0)$ denotes the perceptual distance between images x and x_0 , l denotes the network layer, H_l and W_l denote the height and width of the feature map at layer l , $\hat{y}_{hw}^l$ and $\hat{y}_{0hw}^l$ denote the unit-normalized feature vectors extracted from images (x) and (x_0) at spatial location (h, w) , respectively, w_l denotes the learned channel-wise weight. A lower LPIPS value indicates a smaller perceptual distance between the two images. In this study, LPIPS score distributions were used to compare the perceptual-distance characteristics of ChatGPT-4.0- and ChatGPT-5.5-generated images relative to the real-image dataset.

t-SNE was used to visualize the feature-space distributions of real and generated images [48]. This method converts high-dimensional feature similarities and low-dimensional embedding similarities into probability distributions, denoted as P and Q , respectively. The low-dimensional embedding is optimized by minimizing the Kullback–Leibler divergence between these two distributions, as shown in Equation (7):

$$C = \sum_i KL(P \parallel Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \quad (7)$$

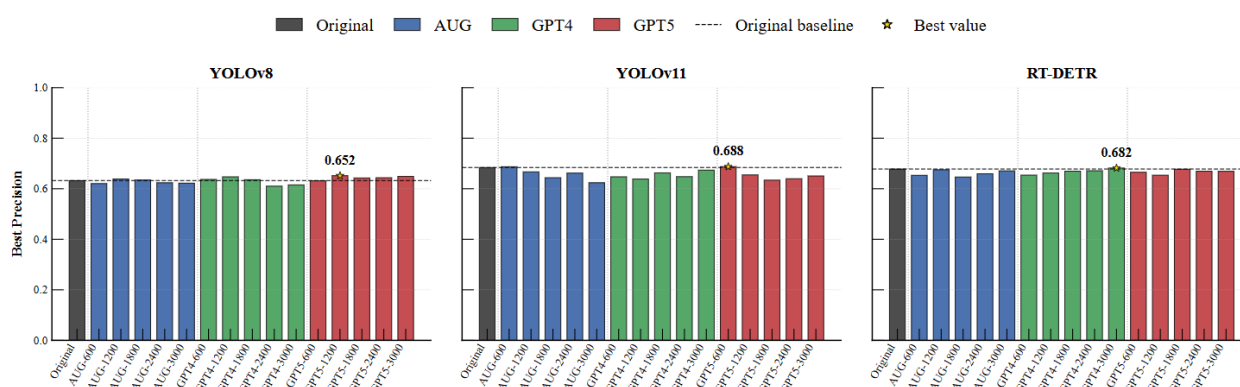
where p_{ij} represents the pairwise similarity between feature vectors in the original high-dimensional space, and q_{ij} represents the corresponding similarity in the low-dimensional embedded space. In this study, feature embeddings were extracted from real, ChatGPT-4.0-generated, and ChatGPT-5.5-generated images and projected into a two-dimensional space using t-SNE. This analysis was used to qualitatively examine whether the generated images formed structured feature distributions rather than irregular or random clusters.

3. Experimental Results

3.1. Fire Detection Model Training and Validation Results

Figure 5 shows the training and validation results obtained under different detector architectures and dataset configurations. The models were evaluated using Precision, Recall, mAP@0.5, and mAP@0.5:0.95. Overall, the generated-image-based training conditions produced stable validation performance across YOLOv8n, YOLOv11n, and RT-DETR, indicating that the generated images did not introduce substantial degradation during model training. For Precision, the best validation scores were 0.652 for YOLOv8n, 0.688 for YOLOv11n, and 0.682 for RT-DETR. The corresponding best Recall values were 0.566, 0.554, and 0.550, respectively. Although Recall remained relatively lower than Precision, the generated-image-based conditions maintained comparable or improved detection sensitivity across the three detector architectures. This trend suggests that the additional synthetic images contributed to improving the detection response to flame and smoke objects while maintaining stable prediction reliability. For mAP-based evaluation, YOLOv11n achieved the highest validation performance, with mAP@0.5 and mAP@0.5:0.95 values of 0.550 and 0.275, respectively. YOLOv8n achieved best values of 0.521 and 0.262, while RT-DETR achieved 0.539 and 0.266 for the same metrics. The lower mAP@0.5:0.95 values compared with mAP@0.5 indicate that accurate localization of flame and smoke regions remains challenging, particularly because smoke boundaries are often ambiguous and flame shapes vary depending on scene conditions.

These validation results indicate that ChatGPT-4.0- and ChatGPT-5.5-generated images can serve as effective supplementary training data for fire detection models. Among the evaluated architectures, YOLOv11n showed the strongest overall validation performance, while RT-DETR also produced competitive results. Therefore, the following test-dataset evaluation further analyzes whether the observed validation trends are maintained under fixed test conditions.



(a) Precision result

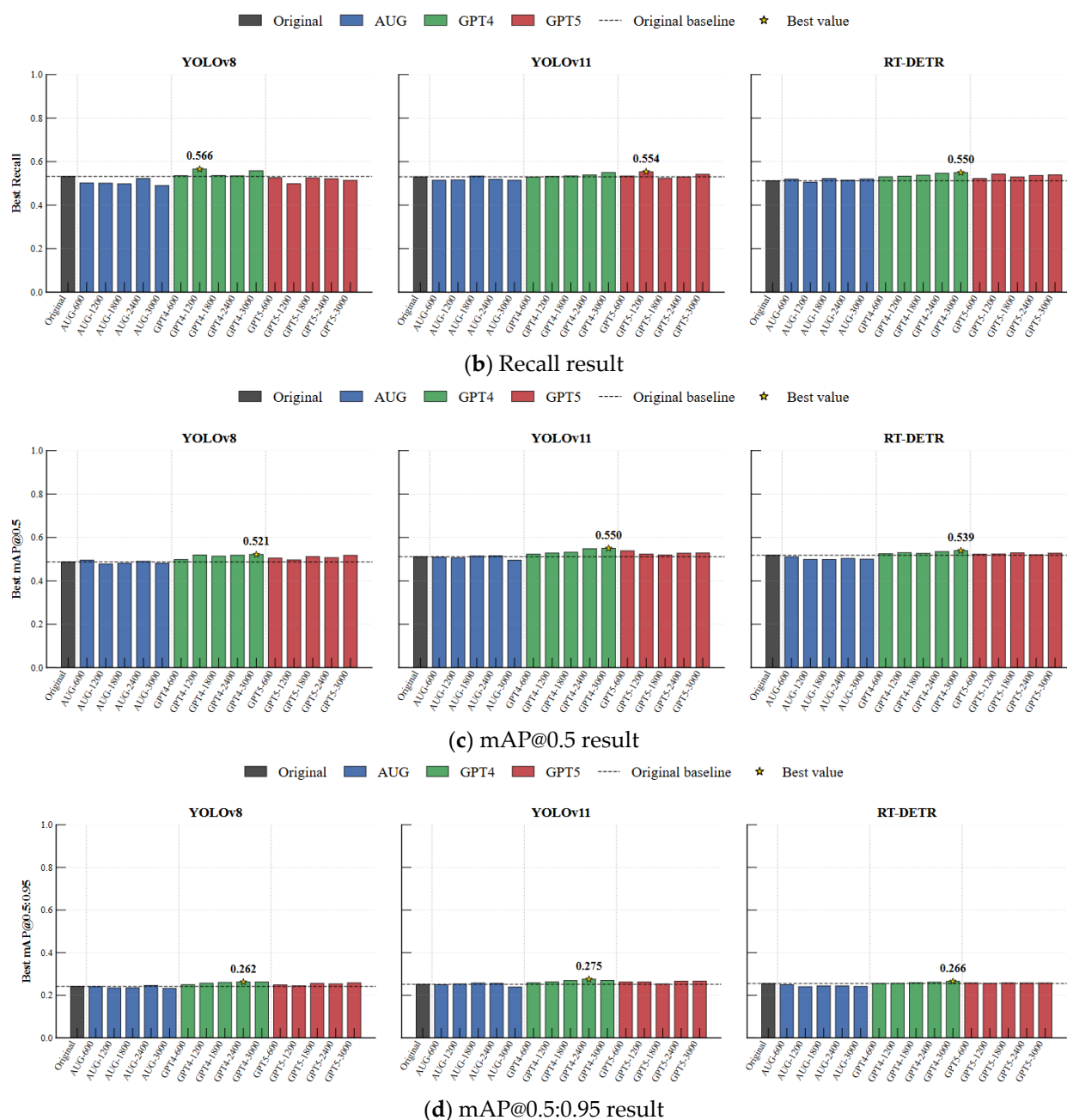


Figure 5. Model training and validation results according to detector architecture and dataset configuration. (a) Precision, (b) Recall, (c) mAP@0.5, and (d) mAP@0.5:0.95.

3.2. Test Dataset-Based Model Evaluation and Inference Results

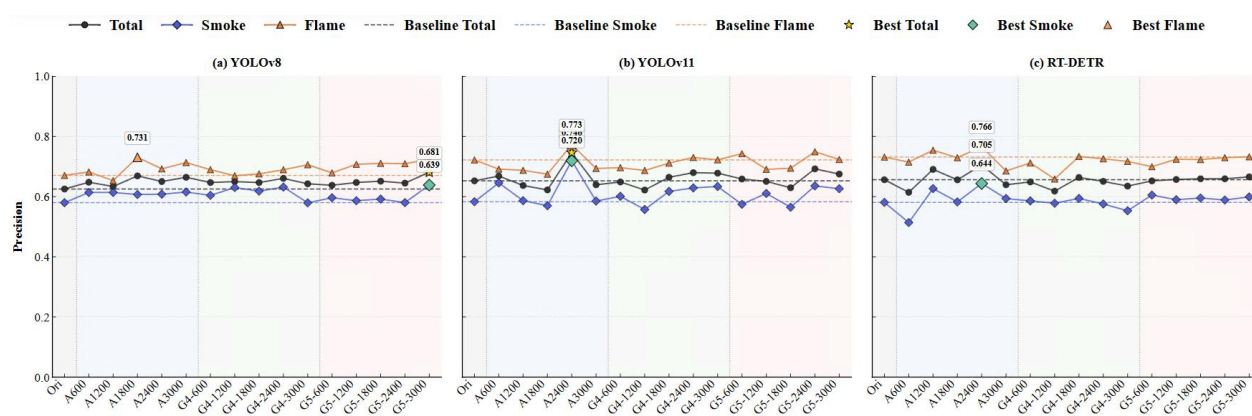
Figure 6 presents the test dataset-based evaluation results for YOLOv8n, YOLOv11n, and RT-DETR under the Original, AUG, ChatGPT-4.o, and ChatGPT-5.5 dataset conditions. The results were analyzed using Precision, Recall, mAP@0.5, and mAP@0.5:0.95 for Total, Smoke, and Flame classes. Overall, the generated-image-based training conditions produced competitive or improved results compared with the original and conventional augmentation conditions, although the degree of improvement varied depending on detector architecture, metric, class, and added-image scale. In the Precision results, YOLOv11n showed the most pronounced improvement under the ChatGPT-4.o condition. In particular, the ChatGPT-4.o 2400 condition produced the highest Precision values for YOLOv11n, reaching 0.740 for Total, 0.720 for Smoke, and 0.773 for Flame. This indicates that the generated images contributed to reducing false-positive detections while main-

taining class-level detection reliability. RT-DETR also showed competitive Precision performance, with best values of 0.705 for Total, 0.644 for Smoke, and 0.766 for Flame. In the case of YOLOv8n, the Flame Precision reached 0.731 under the AUG condition, whereas the ChatGPT-5.5 3000 condition showed high Precision values for Total and Smoke. These results suggest that the generated-image effect was not identical across detector architectures, but it generally helped maintain or improve prediction reliability.

The Recall results showed a different tendency from Precision. Across all detector architectures, Recall values were lower than Precision values, indicating that missed detections remained a more challenging issue than false-positive suppression. The best Recall values for Flame were 0.683 for YOLOv8n, 0.682 for YOLOv11n, and 0.665 for RT-DETR, whereas the corresponding Smoke Recall values remained lower, at approximately 0.470, 0.478, and 0.467, respectively. This class-wise gap indicates that Smoke detection was more difficult than Flame detection, most likely because smoke regions have ambiguous boundaries, low contrast, and large variations in density and shape. Nevertheless, the ChatGPT-4.o-based conditions improved or maintained Recall in YOLOv8n and YOLOv11n, suggesting that prompt-generated smoke and flame patterns provided useful supplementary training diversity.

For mAP@0.5, the generated-image-based conditions also showed meaningful performance tendencies. YOLOv11n achieved the highest overall mAP@0.5 among the three detector architectures, with best values of 0.632 for Total and 0.719 for Flame. YOLOv8n achieved best values of 0.628 for Total and 0.708 for Flame, while RT-DETR reached 0.580 for Total and 0.692 for Flame. These results indicate that YOLOv11n benefited most from the expanded dataset in terms of overall detection accuracy. In contrast, Smoke mAP@0.5 remained lower than Flame mAP@0.5 across all detector architectures, confirming that smoke detection remained the limiting factor in the overall fire detection performance.

The mAP@0.5:0.95 results further demonstrate the difficulty of precise fire-object localization. Compared with mAP@0.5, the mAP@0.5:0.95 values were substantially lower for all detector architectures and dataset conditions. The best Total mAP@0.5:0.95 values were 0.360 for YOLOv8n, 0.363 for YOLOv11n, and 0.315 for RT-DETR, while the best Flame values were 0.417, 0.413, and 0.384, respectively. This decrease is expected because mAP@0.5:0.95 applies stricter IoU thresholds and therefore requires more accurate bounding-box localization. Flame and smoke objects often have irregular shapes, blurred boundaries, and partial occlusions, which makes high-IoU localization more difficult than simple object presence detection.



(A) Precision result

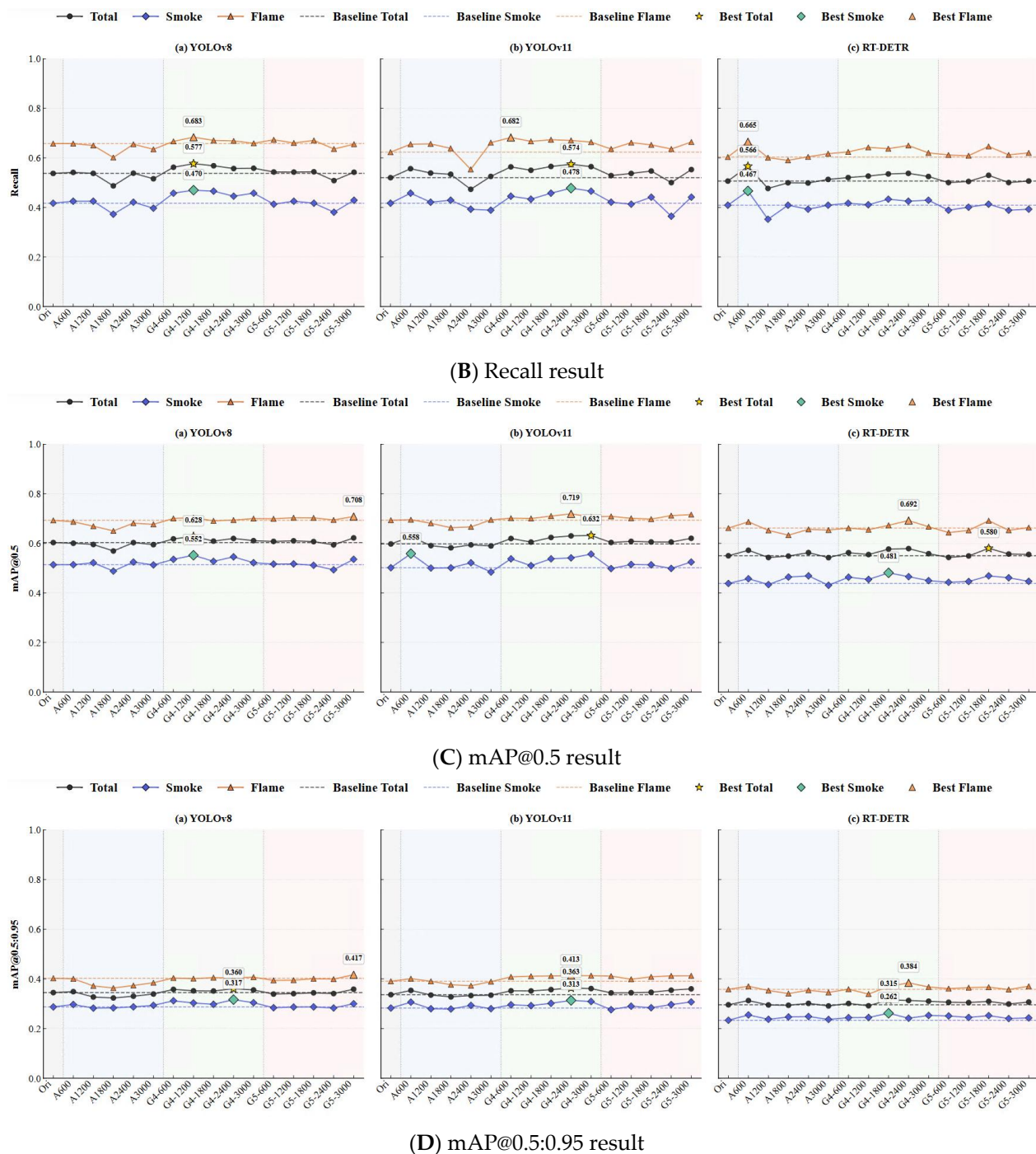


Figure 6. Test dataset-based model evaluation results.

Overall, the test dataset-based evaluation indicates three major findings. First, Flame detection consistently outperformed Smoke detection across all detector architectures and metrics. Second, ChatGPT-4.0- and ChatGPT-5.5-generated images improved or maintained model performance compared with conventional augmentation, but the optimal added-image scale differed by detector and metric. Third, the performance improvement was not simply proportional to the number of added images, suggesting that the quality and distributional relevance of the generated images were more important than dataset size alone. These results support the use of validated generated images as supplementary training data for fire detection models, while also showing that their effectiveness should be evaluated separately for each detector architecture, class, and evaluation metric.

Based on the test-dataset evaluation results, representative improved events were selected for statistical verification. Table 6 summarizes the paired bootstrap analysis conducted to determine whether the observed performance differences between the ChatGPT-4.o-based models and the corresponding conventional augmentation models were statistically reliable.

Table 6. Bootstrap-based statistical comparison of detection performance on the test dataset.

Detector	Comparison	Metric	Reference	Generator	Difference	95% CI
YOLOv11	ChatGPT-4.o 2400 – AUG 2400	Precision	0.635	0.687	+0.052	[0.021, 0.082]
YOLOv11	ChatGPT-4.o 3000 – AUG 2400	Recall	0.569	0.601	+0.031	[0.005, 0.054]
YOLOv11	ChatGPT-4.o 3000 – AUG 3000	mAP@0.5	0.506	0.571	+0.065	[0.037, 0.090]
YOLOv11	ChatGPT-4.o 3000 – AUG 3000	mAP@0.5:0.95	0.244	0.282	+0.038	[0.022, 0.054]

As shown in Table 6, the YOLOv11 model trained with the ChatGPT-4.o 2400 dataset improved Precision from 0.635 to 0.687 compared with the AUG 2400 model, corresponding to a difference of +0.052 with a 95% confidence interval (CI) of [0.021, 0.082]. For Recall, the YOLOv11 model trained with the ChatGPT-4.o 3000 dataset improved the value from 0.569 to 0.601 compared with the AUG 2400 model, yielding a difference of +0.031 with a 95% CI of [0.005, 0.054].

Since both confidence intervals were entirely above zero, the selected improvements in Precision and Recall were statistically supported under test-image-level bootstrap resampling. A similar trend was observed for the mAP-based metrics. The YOLOv11 model trained with the ChatGPT-4.o 3000 dataset improved mAP@0.5 from 0.506 to 0.571 compared with the AUG 3000 model, yielding a difference of +0.065 with a 95% CI of [0.037, 0.090]. In addition, mAP@0.5:0.95 increased from 0.244 to 0.282, with a difference of +0.038 and a 95% CI of [0.022, 0.054]. These results indicate that the selected ChatGPT-4.o-based training conditions produced statistically reliable improvements over the corresponding conventional augmentation conditions in both detection accuracy and stricter localization-based evaluation.

Figure 7 presents representative inference results on the test dataset using selected model weights from each data configuration. The Original and AUG models detected major flame regions in several cases, but their responses to smoke regions and partially occluded fire areas were less consistent. In contrast, the ChatGPT-4.o- and ChatGPT-5.5-based models showed more stable detection responses across diverse fire scenes, including building fires, dense smoke plumes, and complex multi-object fire situations. In particular, the generated-image-based models detected flame and smoke objects in scenes with irregular fire shapes and low-contrast smoke regions, which supports the quantitative improvements observed in the test-dataset evaluation.

Overall, the test-dataset evaluation, bootstrap analysis, and inference visualization consistently demonstrate that generated fire images can provide useful supplementary training information for object detection-based fire detection models. However, the performance gains were dependent on the detector architecture, evaluation metric, class type, and added-image scale. Therefore, the results suggest that generated images are effective when properly screened and incorporated into the training dataset, rather than indicating universal superiority over conventional augmentation under all conditions.

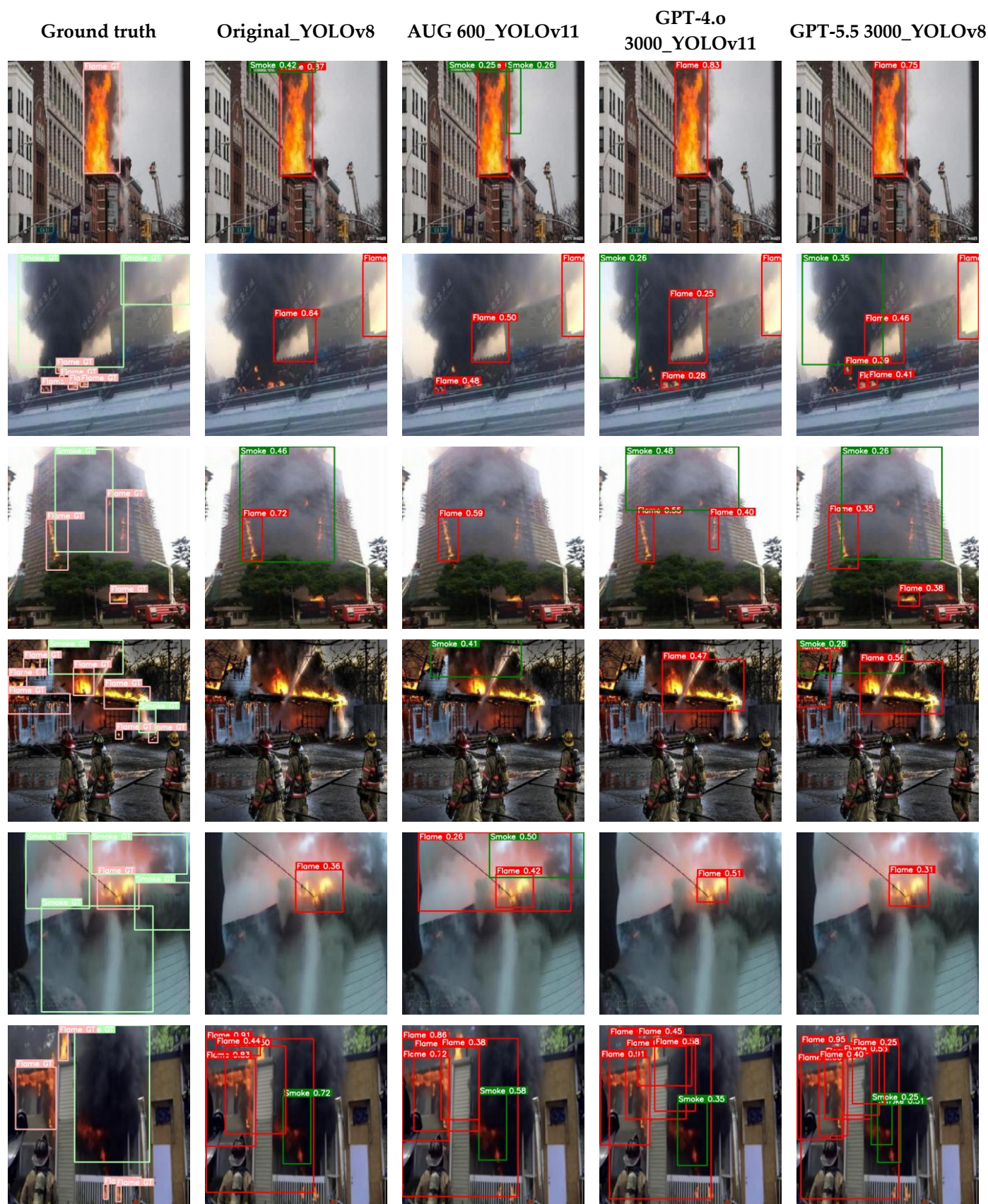


Figure 7. Test Dataset-Based Fire Detection Inference Results.

To examine the detection behavior for small fire-related objects, AP50 was additionally calculated for ground-truth objects whose normalized bounding-box area was less than 5% of the image area. As summarized in Table 7, the GPT-4o-3000 training conditions yielded

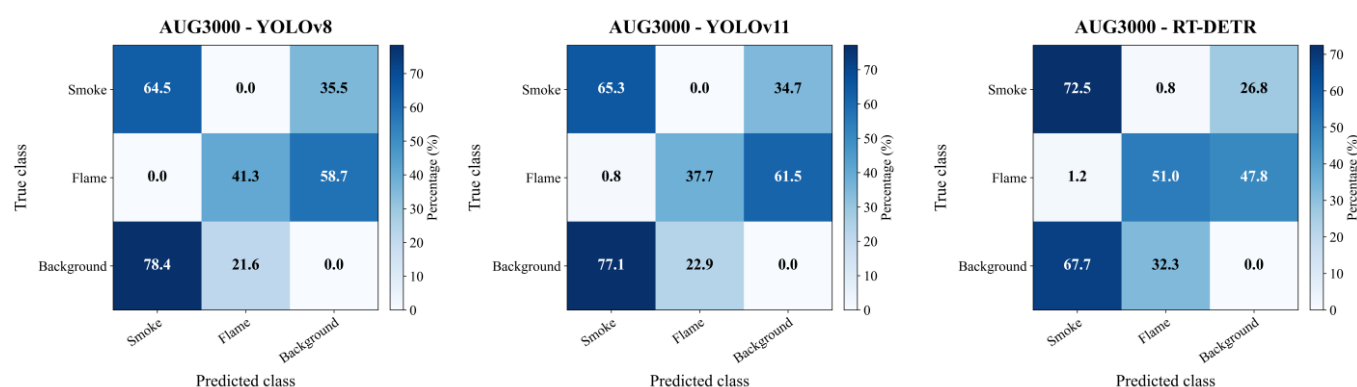
higher small-object mAP@0.5 values than the corresponding AUG3000 conditions for all detector architectures. The mAP@0.5 increased from 0.194 to 0.212 for YOLOv8, from 0.202 to 0.214 for YOLOv11, and from 0.180 to 0.188 for RT-DETR. These results indicate that the GPT-4o-generated images were beneficial for improving small-object detection performance, although the magnitude of improvement varied across detector architectures.

Table 7. Small-object AP50 results for targets occupying less than 5% of the image area.

Training Dataset	Detector	Smoke AP50 (<5%)	Flame AP50 (<5%)	mAP@0.5 (<5%)
GPT4o-3000	YOLOv8	0.392	0.033	0.212
GPT4o-3000	YOLOv11	0.400	0.029	0.214
GPT4o-3000	RT-DETR	0.357	0.020	0.188
AUG3000	YOLOv8	0.363	0.025	0.194
AUG3000	YOLOv11	0.378	0.026	0.202
AUG3000	RT-DETR	0.337	0.023	0.180

The class-wise AP50 results show that this improvement was mainly associated with smoke detection. Smoke AP50 was consistently higher than Flame AP50 in all experimental conditions, ranging from 0.337 to 0.400, whereas Flame AP50 remained low, ranging from 0.020 to 0.033. This result suggests that small flame regions remain difficult to detect, even when generated images are added to the training dataset. Small flame objects generally occupy only a limited number of pixels and often show unstable boundary characteristics, making them more susceptible to confusion with light sources, reflections, and complex background patterns.

The row-normalized confusion matrices in Figure 8 provide further insight into the error characteristics of the models. Compared with the AUG3000 models, the GPT-4o-3000 models generally showed higher diagonal values for both Smoke and Flame, indicating improved class-wise correct classification. For YOLOv11, the correct classification rate increased from 65.3% to 66.9% for Smoke and from 37.7% to 45.7% for Flame. For RT-DETR, the corresponding values increased from 72.5% to 75.4% for Smoke and from 51.0% to 57.5% for Flame. In addition, the proportion of Smoke and Flame objects assigned to the Background column decreased in most GPT-4o-3000 conditions, indicating that generated images contributed to reducing missed detections.



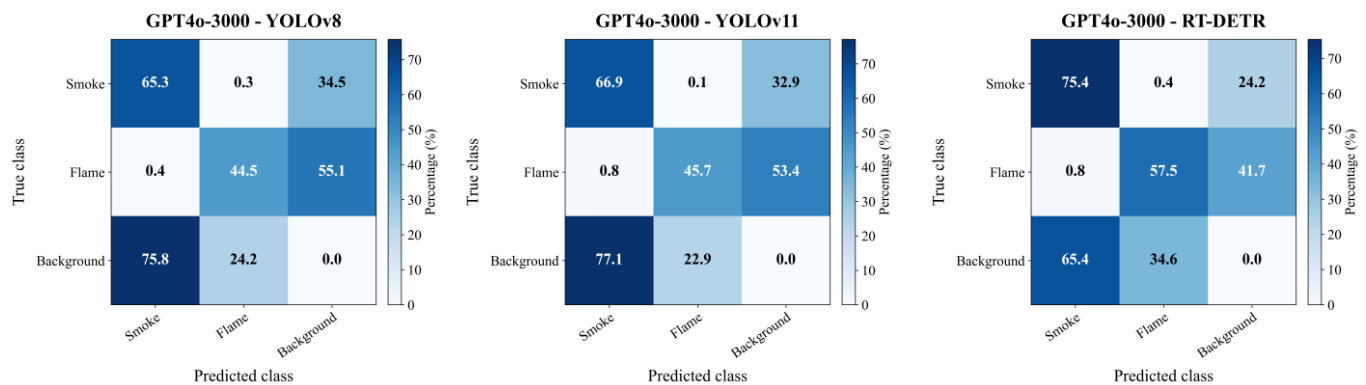


Figure 8. Confusion matrices of AUG3000 and GPT-4o-3000 models.

4. Evaluation of Real-World Fire Detection Performance for the Developed Model

Evaluation of Fire Detection Performance in Various Locations

To examine the practical applicability of the developed fire detection models, inference experiments were conducted using video-frame images collected from various fire-related locations. The evaluated locations included building interiors, subway stations, garages, basements, schools, house interiors, corridors, logistics centers, and hospitals. Because these scenario-based video frames did not include ground-truth annotations, the results were interpreted based on qualitative inference behavior and confidence score distributions rather than conventional accuracy-based metrics.

Figure 9 presents the location-based fire detection inference results obtained using representative models from each data configuration. Overall, all models detected visually distinct flame regions in relatively clear fire scenes, such as building, subway, garage, school, and house interior scenarios. However, differences among the models became more evident in scenes with low contrast, smoke diffusion, partial occlusion, or small fire regions. In particular, the models trained with generated images showed relatively stable responses to flame and smoke objects in complex indoor environments, suggesting that synthetic images contributed to improving model exposure to diverse fire appearances during training.

In the building and subway scenarios, most models detected flame regions with stable confidence, while smoke responses varied depending on the model. In the garage and school scenarios, where flame regions were visually distinct, the models generally showed strong flame detection performance. By contrast, the basement, corridor, logistics center, and hospital scenarios were more challenging because the fire regions were small, partially occluded, or embedded in cluttered backgrounds. These results indicate that location-specific visual characteristics strongly affect inference behavior, particularly when fire or smoke objects are not clearly separated from the background.

Figure 10 summarizes the scenario-based mean confidence scores of the selected models. The value of (n) shown above each bar represents the number of video frames extracted and analyzed for each location. For the YOLOv8 Original model, the garage scenario showed the highest mean confidence score, whereas the basement and logistics center scenarios showed relatively low values. This suggests that the original model responded strongly to visually distinct flame scenes but showed weaker responses under low-contrast or spatially complex conditions.

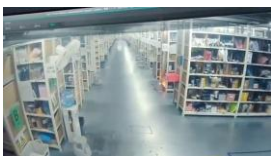
Location	Original_YOLOv8	AUG 1800_YOLOv11	GPT-4o 3000_YOLOv11	GPT-5.5 3000_YOLOv8
Building				
Subway				
Garage				
Basement				
School				
House interior				
Corridor				
Logistics center				
Hospital				

Figure 9. Fire Detection Inference Results by Location.

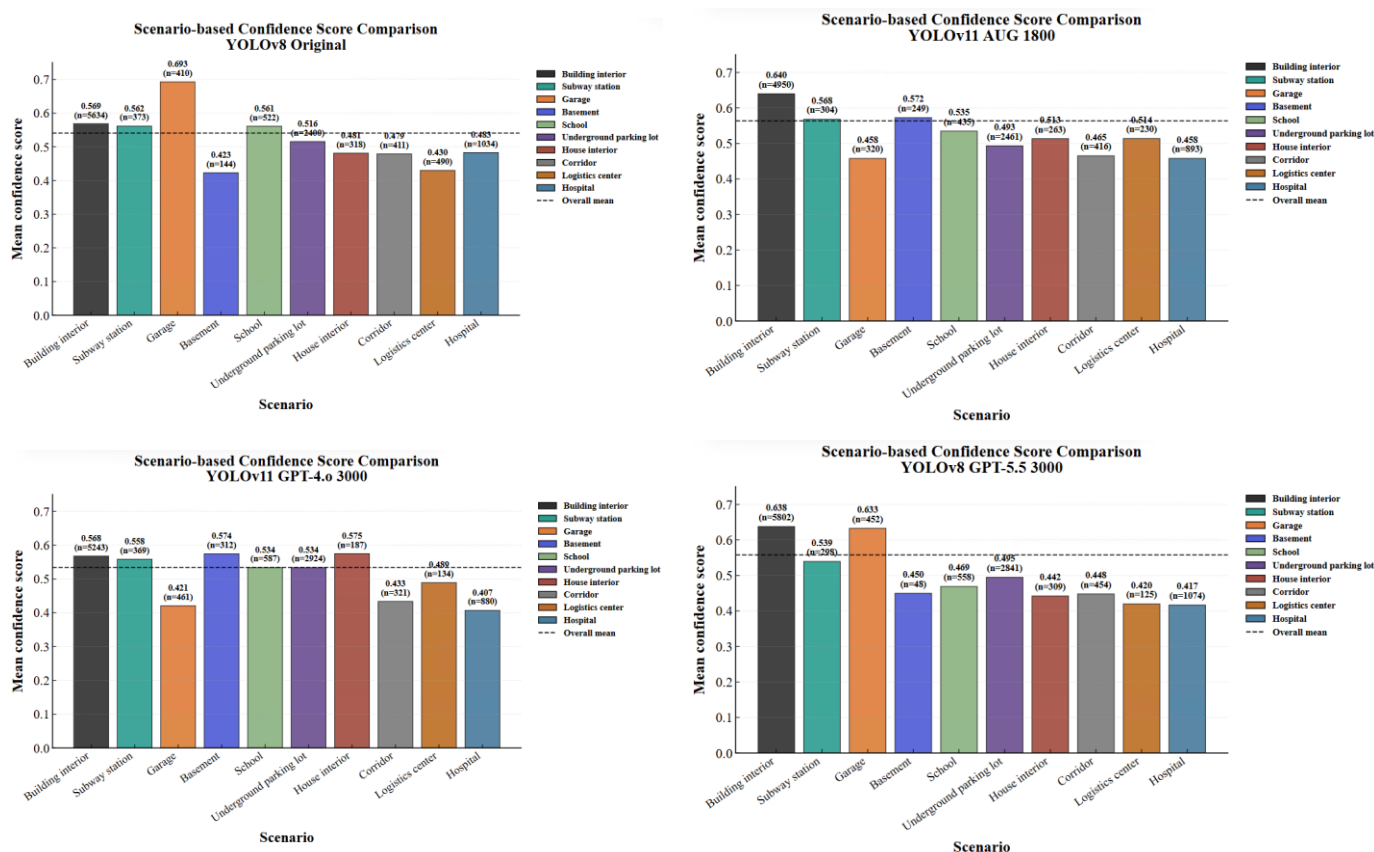


Figure 10. Scenario-based mean confidence comparison of selected fire detection models.

For the YOLOv11 AUG 1800 model, the building interior scenario showed the highest mean confidence score, followed by the basement and subway station scenarios. However, the garage and hospital scenarios showed relatively lower confidence values. This indicates that conventional augmentation improved model response in some indoor and structured environments, but its effect was not uniform across all location types.

The YOLOv11 ChatGPT-4.0 3000 model showed relatively balanced confidence scores across several indoor scenarios. In particular, the basement and house interior scenarios showed high confidence values compared with other locations. This result suggests that ChatGPT-4.0-generated images contributed to improving the model's response to visually diverse indoor fire scenes, including low-light and smoke-affected environments. However, the garage and hospital scenarios still showed lower confidence scores, indicating that some scene-specific conditions remained challenging.

The YOLOv8 ChatGPT-5.5 3000 model showed high confidence scores in the building interior and garage scenarios, indicating strong responses to clear flame-dominant scenes. However, confidence scores were relatively low in the basement, logistics center, and hospital scenarios. This tendency suggests that ChatGPT-5.5-generated images were effective in enhancing responses to certain fire appearances but did not uniformly improve confidence across all location categories.

Overall, the location-based inference results suggest that generative AI-based synthetic images can improve the practical response characteristics of fire detection models under diverse visual environments. Although the confidence score results varied depending on location and detector configuration, the ChatGPT-4.0- and ChatGPT-5.5-based models showed stable detection responses in several complex indoor fire scenarios, including smoke-affected and cluttered scenes. These findings indicate that generated fire images can supplement the limited diversity of real fire datasets by providing additional variations in fire scale, scene context, and smoke-flame appearance. Therefore, the use of

validated generative AI-based images can be considered an effective data expansion strategy for improving the applicability and robustness of vision-based fire detection models. However, because the scenario-based frames did not include ground-truth annotations, these results should be interpreted as complementary evidence of model response behavior rather than as direct accuracy-based performance evaluation.

5. Discussion

The experimental results demonstrate that generative AI-based synthetic images can function as effective supplementary data for fire detection model training. The generated-image-based training conditions improved or maintained detection performance compared with conventional augmentation in several detector–metric combinations, and the bootstrap analysis confirmed that selected ChatGPT-4.o-based YOLOv11 conditions produced statistically supported improvements. These findings suggest that prompt-controlled synthetic images can help compensate for the limited diversity of real fire-image datasets by providing additional variations in flame shape, smoke appearance, fire scale, and scene context.

The validity analyses further explain why the generated images contributed to model performance. The LPIPS distributions in Figure 11 show that ChatGPT-4.o- and ChatGPT-5.5-generated images had comparable perceptual distances from the real-image dataset. This indicates that both generative sources produced synthetic images with a consistent level of perceptual similarity to real fire images. In addition, the t-SNE visualization in Figure 12 shows that the generated images formed structured feature-space distributions rather than random or highly scattered patterns. Although the real and generated images were partially separated, this separation reflects the inherent difference between real-image acquisition and text-to-image generation. Therefore, the generated images should be interpreted as complementary training samples that expand the data distribution, rather than as direct substitutes for real fire images.

Nevertheless, several limitations should be considered. First, generated-image screening based on detection confidence cannot fully guarantee physical realism or scenario-level correctness. Second, the effectiveness of synthetic images varied depending on detector architecture, class type, evaluation metric, and added-image scale, indicating that more generated images do not necessarily lead to proportional performance improvement. Third, the scenario-based inference analysis used video frames without ground-truth annotations; therefore, confidence score results should be interpreted as model response characteristics rather than direct accuracy-based performance. Future research should strengthen generated-image quality control, expand annotated real-world fire scenario datasets, and incorporate multimodal data such as thermal images and sensor signals to improve early-stage fire detection.

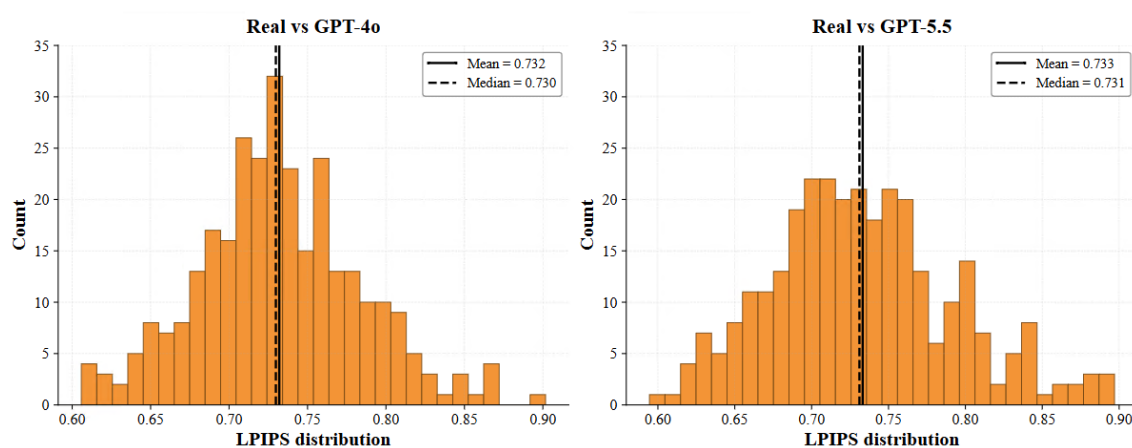


Figure 11. LPIPS distribution results between real and generated fire images.

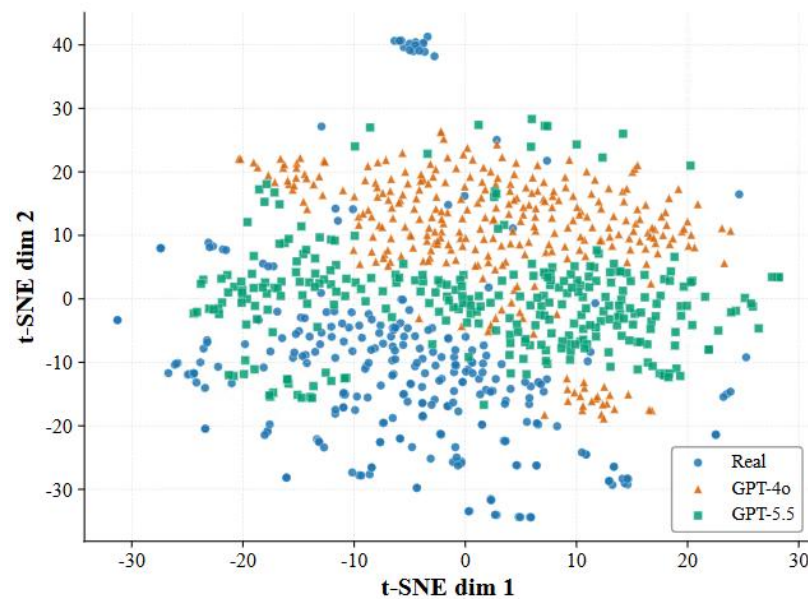


Figure 12. t-SNE visualization results of real and generated fire images.

6. Conclusions

This study proposed a generative AI-assisted data expansion strategy for improving object detection-based fire detection models. Real fire images, conventional augmented images, and synthetic images generated using ChatGPT-4.o and ChatGPT-5.5 were used to train YOLOv8n, YOLOv11n, and RT-DETR models. The models were evaluated using test-dataset-based detection metrics, bootstrap-based statistical verification, scenario-based inference, and generated-image validity analyses.

The results showed that properly screened synthetic fire images can contribute to improving the performance and practical response characteristics of fire detection models. In particular, selected ChatGPT-4.o-based YOLOv11 training conditions showed statistically supported improvements over matched conventional augmentation conditions in Precision, Recall, mAP@0.5, and mAP@0.5:0.95. The location-based inference results also showed that generated-image-based models can respond effectively to diverse indoor fire scenes involving flame, smoke, cluttered backgrounds, and low-contrast visual conditions.

Overall, the findings indicate that validated generative AI-based fire images can serve as useful supplementary training data for mitigating the limited diversity of real fire-image datasets. This approach has practical potential for enhancing the robustness and applicability of vision-based fire detection systems. Future work will therefore establish a structured generation record in which the intended scene category, visible fire-scale range, and ignition scenario are stored as image-level metadata at the time of synthesis. This will enable expert-based validity assessment of flame appearance, smoke characteristics, and scene consistency, as well as ablation analyses according to scenario variables with inter-rater agreement.

Author Contributions: Conceptualization, W.J.; methodology, S.C.; software, W.J. and H.K.; validation, H.J.; formal analysis, S.C.; investigation, W.J. and S.C.; data curation, H.K.; writing—original draft preparation, H.K.; writing—review and editing, H.J.; supervision, W.J. and H.J.; project administration, H.J.; funding acquisition, H.J. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by the Semyung University Research Grant of 2024.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding authors.

Acknowledgments: During the preparation of this work, the author used ChatGPT to improve the language and readability of the manuscript and to assist in generating synthetic images for training dataset construction. After using this tool, the author carefully reviewed and edited the content as necessary and takes full responsibility for the content of the publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Al-batat, R.; Angelopoulou, A.; Premkumar, S.; Hemanth, J.; Kapetanios, E. An end-to-end automated license plate recognition system using YOLO-based vehicle and license plate detection with vehicle classification. *Sensors* **2022**, *22*, 9477. <https://doi.org/10.3390/s22239477>.
2. Khanam, R.; Hussain, M. YOLOv11: An overview of the key architectural enhancements. *arXiv* **2024**, arXiv:2410.17725. <https://doi.org/10.48550/arXiv.2410.17725>.
3. Banachewicz, K.; Massaron, L. *The Kaggle Book: Data Analysis and Machine Learning for Competitive Data Science*; Packt Publishing Ltd.: Birmingham, UK, 2022.
4. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. End-to-end object detection with transformers. In *Computer Vision-ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M., Eds.; Springer: Cham, Switzerland, 2020; pp. 213–229. https://doi.org/10.1007/978-3-030-58452-8_13.
5. Chung, M.-A.; Lin, Y.-J.; Lin, C.-W. YOLO-SLD: An attention mechanism-improved YOLO for license plate detection. *IEEE Access* **2024**, *12*, 89035–89045. <https://doi.org/10.1109/ACCESS.2024.3419587>.
6. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* **2023**, *82*, 9243–9275. <https://doi.org/10.1007/s11042-022-13644-y>.
7. Flores-Calero, M.; Astudillo, C.A.; Guevara, D.; Maza, J.; Lita, B.S.; Defaz, B.; Ante, J.S.; Zabala-Blanco, D.; Armingol Moreno, J.M. Traffic sign detection and recognition using YOLO object detection algorithm: A systematic review. *Mathematics* **2024**, *12*, 297. <https://doi.org/10.3390/math12020297>.
8. Fonder, M.; Van Droogenbroeck, M. Mid-Air: A multi-modal dataset for extremely low altitude drone flights. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA, 16–20 June 2019; pp. 553–562. <https://doi.org/10.1109/CVPRW.2019.00081>.
9. Kwon, H.; Lee, B.; Jung, H. Research on improving the performance of YOLO-based object detection models for smoke and flames from different materials. *J. Korean Inst. Electr. Electron. Mater. Eng.* **2024**, *37*, 261–273. <https://doi.org/10.4313/JKEM.2024.37.3.4>.
10. Huang, J.; Zhou, J.; Yang, H.; Liu, Y.; Liu, H. A small-target forest fire smoke detection model based on deformable transformer for end-to-end object detection. *Forests* **2023**, *14*, 162. <https://doi.org/10.3390/f14010162>.
11. Jiang, P.; Ergu, D.; Liu, F.; Cai, Y.; Ma, B. A review of YOLO algorithm developments. *Procedia Comput. Sci.* **2022**, *199*, 1066–1073. <https://doi.org/10.1016/j.procs.2022.01.135>.
12. Jung, H.; Choi, S.; Lee, B. Rotor fault diagnosis method using CNN-based transfer learning with 2D sound spectrogram analysis. *Electronics* **2023**, *12*, 480. <https://doi.org/10.3390/electronics12030480>.
13. Junos, M.H.; Mohd Khairuddin, A.S.; Thannirmalai, S.; Dahari, M. An optimized YOLO-based object detection model for crop harvesting system. *IET Image Process.* **2021**, *15*, 2112–2125. <https://doi.org/10.1049/ipr2.12181>.
14. Zhao, L.; Zhi, L.; Zhao, C.; Zheng, W. Fire-YOLO: A small target object detection method for fire inspection. *Sustainability* **2022**, *14*, 4930. <https://doi.org/10.3390/su14094930>.
15. Xu, H.; Li, B.; Zhong, F. Light-YOLOv5: A lightweight algorithm for improved YOLOv5 in complex fire scenarios. *Appl. Sci.* **2022**, *12*, 12312. <https://doi.org/10.3390/app122312312>.
16. Zhao, H.; Jin, J.; Liu, Y.; Guo, Y.; Shen, Y. FSDF: A high-performance fire detection framework. *Expert Syst. Appl.* **2024**, *238*, 121665. <https://doi.org/10.1016/j.eswa.2023.121665>.
17. Cheng, G.; Chen, X.; Wang, C.; Li, X.; Xian, B.; Yu, H. Visual fire detection using deep learning: A survey. *Neurocomputing* **2024**, *596*, 127975. <https://doi.org/10.1016/j.neucom.2024.127975>.

18. Vasconcelos, R.N.; Franca Rocha, W.J.S.; Costa, D.P.; Duverger, S.G.; Santana, M.M.M.d.; Cambui, E.C.B.; Ferreira-Ferreira, J.; Oliveira, M.; Barbosa, L.d.S.; Cordeiro, C.L. Fire detection with deep learning: A comprehensive review. *Land* **2024**, *13*, 1696. <https://doi.org/10.3390/land13101696>.
19. Wu, S.; Zhang, X.; Liu, R.; Li, B. A dataset for fire and smoke object detection. *Multimed. Tools Appl.* **2023**, *82*, 6707–6726. <https://doi.org/10.1007/s11042-022-13580-x>.
20. Dewangan, A.; Pande, Y.; Braun, H.-W.; Vernon, F.; Perez, I.; Altintas, I.; Cottrell, G.W.; Nguyen, M.H. FIgLib & SmokeyNet: Dataset and deep learning model for real-time wildland fire smoke detection. *Remote Sens.* **2022**, *14*, 1007. <https://doi.org/10.3390/rs14041007>.
21. Park, M.; Tran, D.Q.; Bak, J.; Park, S. Advanced wildfire detection using generative adversarial network-based augmented datasets and weakly supervised object localization. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *114*, 103052. <https://doi.org/10.1016/j.jag.2022.103052>.
22. Ljungqvist, M.G.; Nordander, O.; Skans, M.; Mildner, A.; Liu, T.; Nugues, P. Object detector differences when using synthetic and real training data. *SN Comput. Sci.* **2023**, *4*, 281. <https://doi.org/10.1007/s42979-023-01704-5>.
23. Kiefer, B.; Ott, D.; Zell, A. Leveraging synthetic data in object detection on unmanned aerial vehicles. In Proceedings of the 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022; pp. 4468–4474. <https://doi.org/10.1109/ICPR56361.2022.9956710>.
24. Betker, J.; Goh, G.; Jing, L.; Brooks, T.; Wang, J.; Li, L.; Ouyang, L.; Zhuang, J.; Lee, J.; Guo, Y.; et al. Improving Image Generation with Better Captions. OpenAI Technical Report 2023. Available online: <https://cdn.openai.com/papers/dall-e-3.pdf> (accessed on 10 October 2025).
25. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. DETRs beat YOLOs on real-time object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 16965–16974. <https://doi.org/10.1109/CVPR52733.2024.01605>.
26. Terven, J.; Córdova-Esparza, D.-M.; Romero-González, J.-A. A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* **2023**, *5*, 1680–1716. <https://doi.org/10.3390/make5040083>.
27. Yaseen, M. What is YOLOv8: An in-depth exploration of the internal features of the next-generation object detector. *arXiv* **2024**, arXiv:2408.15857. <https://doi.org/10.48550/arXiv.2408.15857>.
28. Chetoui, M.; Akhloufi, M.A. Fire and smoke detection using fine-tuned YOLOv8 and YOLOv7 deep models. *Fire* **2024**, *7*, 135. <https://doi.org/10.3390/fire7040135>.
29. Roboflow. Fire and Smoke Dataset (Separate Fire and Smoke). Available online: <https://universe.roboflow.com/sepratefriends-smoke/fire-syvwe> (accessed on 1 October 2024).
30. Roboflow. Fire Dataset (Jos Buttler). Available online: <https://universe.roboflow.com/jos-buttler-7xxsn/fire-hxsld> (accessed on 1 October 2024).
31. Roboflow. Fire Detection Dataset (QWE LWESW). Available online: <https://universe.roboflow.com/qwe-lwesw/fire-o4970> (accessed on 1 October 2024).
32. Roboflow. Fire Detection Dataset (Arkan). Available online: <https://universe.roboflow.com/arkan-jrix2/fire-detection-my95a> (accessed on 1 October 2024).
33. Roboflow. Fire Detection Dataset (Fire ULBsk). Available online: <https://universe.roboflow.com/fire-ulbsk/fire-detection-ackkx> (accessed on 1 October 2024).
34. Roboflow. Fire Detection Dataset (ADSGFSHFHDGJ). Available online: <https://universe.roboflow.com/adsgfshfhhdgj/fire-detection-vzlsf> (accessed on 1 October 2024).
35. Roboflow. Fire Dataset (Test). Available online: <https://universe.roboflow.com/test-d0sr2/fire-e4rbw> (accessed on 1 October 2024).
36. Roboflow. Fire Dataset (Anika Kabir). Available online: <https://universe.roboflow.com/anika-kabir/fire-vrgjf> (accessed on 1 October 2024).
37. Roboflow. Fire Dataset (Sergey Kovalenko). Available online: <https://universe.roboflow.com/sergey-kovalenko/fire-orek4> (accessed on 1 October 2024).
38. Roboflow. Fire Dataset (Tree GPDDE). Available online: <https://universe.roboflow.com/tree-gpdde/fire-tbgul> (accessed on 1 October 2024).
39. Roboflow. Fire Dataset (INO UILSQ). Available online: <https://universe.roboflow.com/ino-uilsq/fire-2ejf7> (accessed on 1 October 2024).
40. Roboflow. Fire Dataset (Ebenezer LQXEG). Available online: <https://universe.roboflow.com/ebenezer-lqxeg/fire-k6xeb/browse?queryText=&pageSize=50&startingIndex=0&browseQuery=true> (accessed on 1 October 2024).

41. Roboflow. Fire Dataset (1st Project). Available online: <https://universe.roboflow.com/1st-project-k0wu9/fire-yp5px> (accessed on 1 October 2024).
42. Roboflow. Fire Detection Dataset (Fire C3H8X). Available online: <https://universe.roboflow.com/fire-c3h8x/fire-fger8> (accessed on 1 October 2024).
43. Kaggle. Fire Detection Dataset (Metin Mekiab Ullrahman). Available online: <https://www.kaggle.com/datasets/phy-lake1337/fire-dataset> (accessed on 1 October 2024).
44. Kaggle. Fire Detection Dataset (Fire-detection). Available online: <https://www.kaggle.com/datasets/metinmekiabullrahman/fire-detection> (accessed on 1 October 2024).
45. Yunusov, N.; Islam, B.M.S.; Abdusalomov, A.; Kim, W. Robust forest fire detection method for surveillance systems based on You Only Look Once version 8 and transfer learning approaches. *Processes* **2024**, *12*, 1039. <https://doi.org/10.3390/pr12051039>.
46. Efron, B.; Tibshirani, R.J. *An Introduction to the Bootstrap*; CRC: New York, NY, USA, 1993.
47. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 586–595. <https://doi.org/10.1109/CVPR.2018.00068>.
48. van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.