International Journal of Web of Multidisciplinary Studies
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
website: *https://ijwos.com*
Volume 03, Issue 11,November 2026
OPEN ACCESS
E-ISSN : 3049-2424
DOI: https://doi.org/10.71366/ijwos

Fire and Smoke Detection Using the YOLOv8 Object Detection Model

MUPPINA SIVAKRISHNA[1], NAKKA BHARATH[2], NARIKELLA JAGADEESH[3], M. CHIRANJEEVI[4]
[1,2,3] Master of Computer Applications Final Year Students, Department of CSE, Avanthi Institute of Engineering & Technology, Anakapalli, Andhra Pradesh, India
[4] Associate Professor, Department of CSE, Avanthi Institute of Engineering & Technology, Anakapalli, Andhra Pradesh, India

Article Info

Article History:
Published: 04 Nov 2026

Publication Issue:
Volume 03, Issue 11
November 2026

Page Number:
28-31

Corresponding Author:
MUPPINA SIVAKRISHNA

Abstract:
Fire and smoke incidents require rapid detection to reduce risks to people, property, and the environment. This paper presents a vision-based fire and smoke detection approach built around YOLO object detection, with an enhanced YOLOv8 configuration incorporating GnConv-based feature aggregation, attention mechanisms, and Omni-Dimensional Dynamic Convolution with Batch Statistics (ODConvBS). The system processes surveillance video through frame extraction and image pre-processing, then produces fire/smoke class labels, bounding boxes, and confidence scores. A Python implementation using PyTorch and the Ultralytics framework is exposed through a Flask-based inference application. The source study reports that a YOLOv5x6 baseline achieved 79.2% mAP, 80.6% precision, 74.0% recall, and 38 frames per second (FPS), while the enhanced YOLOv8 configuration achieved 89.4% mAP, 87.8% precision, 85.2% recall, and 46 FPS. These results indicate the potential of the enhanced detector for real-time visual monitoring. Dataset composition, training configuration, and test hardware must be documented in a reproducible evaluation.

Keywords: Fire detection, smoke detection, YOLOv8, object detection, GnConv, attention mechanism, ODConvBS, real-time surveillance.

1. Introduction

Fire accidents endanger human life, property, and the environment. Timely identification of flames and smoke can support early warning and intervention. Conventional fire-safety installations commonly use smoke, heat, or infrared sensors. Although useful in enclosed locations, sensor-only methods may provide limited visual confirmation and may be difficult to deploy uniformly across open or extensive areas.

Computer vision enables surveillance cameras to act as visual sensing devices. The You Only Look Once (YOLO) family performs object detection in a single-stage pipeline and is widely used where detection speed is important [1], [2]. However, flame and smoke are visually variable, non-rigid targets. Small or distant fire regions, changing illumination, and background clutter can make reliable localization difficult.

The source work studies YOLOv5x6 and YOLOv8 configurations and describes an enhanced YOLOv8 detector that combines GnConv-based feature aggregation, attention mechanisms, and ODConvBS. The intended system accepts surveillance video, extracts frames, performs detection, and generates alerts from the resulting fire/smoke predictions. The reported objective is to improve detection accuracy while retaining real-time inference performance.

2. LITERATURE REVIEW

Early vision-based fire detection methods used hand-crafted color, texture, and motion features. These methods can be computationally efficient, but their behavior may be sensitive to illumination changes and complex backgrounds. Deep convolutional detectors learn hierarchical visual representations and have become a common alternative for surveillance applications.

Copyright © 2026 The Author(s) : This is an open access article under the CC BY license (http://creativecommons.org/licenses/by-nc/4.0/)

Redmon et al. introduced YOLO as a unified real-time object detection framework [1]. The Ultralytics YOLOv8 implementation provides a further development of the YOLO detector family [2]. Rao et al. proposed HorNet, which uses recursive gated convolutions to model higher-order spatial interactions [3]. Li et al. introduced Omni-Dimensional Dynamic Convolution, applying attention across multiple dimensions of convolutional weights [4].

Muhammad et al. studied convolutional neural networks for fire detection in surveillance videos [5]. The source paper also cites recent YOLOv8-based visual smoke and forest-fire studies [6]–[8], and a 2026 multispectral YOLO26-based UAV fire-detection study [9]. These works provide context for improving feature extraction, small target sensitivity, and the balance between detection accuracy and computational cost.

The supplied source provides brief summaries rather than full comparative details for references [5]–[9]. Accordingly, this review uses those references as contextual background and does not attribute unreported quantitative results to them.

3. PROPOSED SYSTEM

A. System Overview

The proposed system is a camera-based fire and smoke detection pipeline. A surveillance stream is captured and converted into individual frames. Each frame is pre-processed and passed to an enhanced YOLOv8 model. The detector generates class predictions, bounding-box coordinates, and confidence scores for the target categories fire and smoke. The inference service exposes the results through a Flask-based web application, and an alert is generated based on the detection output.

B. Enhanced YOLOv8 Configuration

The system modifies the YOLOv8 architecture by integrating three components described in the source paper. (1) GnConv-based feature aggregation is intended to capture higher-order spatial relationships and represent the irregular structures of flame and smoke. (2) Attention mechanisms emphasize salient image regions and suppress less relevant background information. (3) ODConvBS, described as Omni-Dimensional Dynamic Convolution with Batch Statistics, is used to improve feature extraction across varying scales. The source identifies the backbone and neck as the areas in which the enhanced components are integrated, but it does not provide a layer-by-layer network specification.

C. Detection Output

For each processed frame, the detector is designed to return the predicted class, bounding-box location, and confidence score. These outputs support visual localization and downstream alert generation. The supplied paper does not specify the confidence threshold, non-maximum-suppression settings, alert persistence rule, or duplicate alert handling; these parameters should be reported in the final experimental description.

4. SYSTEM ARCHITECTURE AND WORKFLOW

Fig. 1 presents the high-level inference architecture derived from the implementation description. The video input is converted into frames, pre-processed, and passed to the enhanced YOLOv8 detector. The detector uses the specified feature modules and returns object-level predictions. The Flask layer makes the inference results available to the application and supports alert delivery.

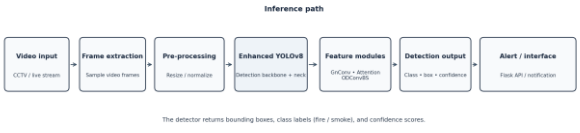


Fig. 1. High-level architecture of the proposed fire and smoke detection system.

The runtime workflow is shown in Fig. 2. After application initialization, the system continuously captures video and extracts frames for inference. If fire or smoke is detected, the prediction is forwarded to the alert interface. Otherwise, monitoring continues. The decision condition is represented generically because the source does not specify a numeric confidence threshold.

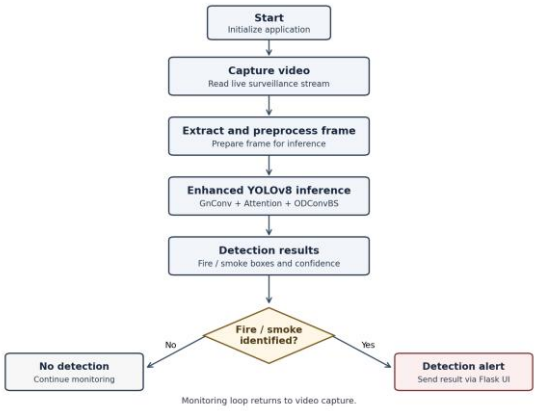


Fig. 2. Real-time video processing and fire/smoke alert workflow.

A. Model Development and Evaluation Flow

The source indicates that the customized model is trained using a curated fire/smoke dataset and evaluated using mAP, precision, recall, and inference speed. Fig. 3 summarizes this development path at the level supported by the provided description. Details such as dataset size, class balance, train/validation/test split, annotation format, augmentation policy, training epochs, and optimizer are not specified in the source.

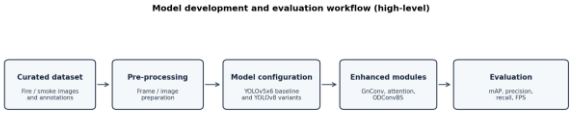


Fig. 3. High-level model development and evaluation workflow.

5. IMPLEMENTATION DETAILS

The implementation stack specified in the source consists of Python, PyTorch, and the Ultralytics framework. The trained detector is wrapped in a Flask-based web application that functions as a real-time inference API. The operational pipeline includes live video capture, frame extraction, image pre-processing, YOLOv8 inference, and alert generation from detection results.

For reproducibility, the final manuscript should identify the exact YOLOv8 model variant, input resolution, hardware and accelerator, software versions, confidence and NMS thresholds, and the precise placement of GnConv, attention, and ODConvBS modules. These implementation details are not present in the supplied document and have therefore not been inferred.

6. EXPERIMENTAL RESULTS AND DISCUSSION

The source evaluates the models using mean Average Precision (mAP), precision, recall, and inference speed in frames per second. Table I reproduces the reported comparison. The source refers to the enhanced model's value as mAP@0.5 in the detailed results; the final submission should confirm that the baseline value uses the same IoU definition and evaluation protocol.

TABLE I
REPORTED PERFORMANCE COMPARISON

Model	mAP (%)	**Precision**	**Recall**	**Speed (FPS)**
YOLOv5x6 baseline	79.2%	80.6%	74.0%	38
Enhanced YOLOv8	89.4%	87.8%	85.2%	46

A. Comparative Analysis

According to Table I, the enhanced YOLOv8 configuration reports an increase in mAP from 79.2% to 89.4%, an increase of 10.2 percentage points. Precision increases from 80.6% to 87.8% (7.2 percentage points), and recall increases from 74.0% to 85.2% (11.2 percentage points). Reported inference speed increases from 38 FPS for YOLOv5x6 to 46 FPS for the enhanced YOLOv8 configuration. These values are reproduced from the supplied paper and should be interpreted in the context of a common test set and hardware configuration, which must be confirmed by the authors.

B. Ablation Observation

The source reports that GnConv provided the largest individual improvement in recall by modeling the non-rigid spatial structure of flames and smoke. However, no per-module ablation values or experimental settings are provided. A complete ablation table should report the baseline, each individual module, and the combined model under the same dataset split and training settings.

C. Evaluation Considerations

The reported results are promising, but a reproducible study should document the number and source of images or video clips, class distribution, annotation procedure, data split, training parameters, test hardware, and evaluation software. It should also report false-positive and false-negative examples, performance under different lighting and distance conditions, and latency measured end to end rather than only model FPS. The supplied source does not contain these details.

7. CONCLUSION

This paper presents a vision-based fire and smoke detection system built on an enhanced YOLOv8 object detector. The described model integrates GnConv-based feature aggregation, attention mechanisms, and ODConvBS to improve feature representation across irregular shapes and varying target scales. The implementation uses Python, PyTorch, Ultralytics, and a Flask inference interface to process surveillance video and generate detection alerts. The source reports that the enhanced YOLOv8 configuration achieves 89.4% mAP, 87.8% precision, 85.2% recall, and 46 FPS, compared with 79.2% mAP, 80.6% precision, 74.0% recall, and 38 FPS for the YOLOv5x6 baseline. Further work should provide a reproducible dataset and training protocol, detailed ablation measurements, and field testing across different camera positions, lighting conditions, and deployment hardware.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2016.

[2] G. Jocher, A. Chaurasia, and J. Qiu, “Ultralytics YOLOv8,” Ultralytics GitHub repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[3] Y. Rao et al., “HorNet: Efficient High-Order Spatial Interactions with Recursive Gated Convolutions,” in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2022.

[4] C. Li, A. Zhou, and A. Yao, “Omni-Dimensional Dynamic Convolution,” in Proc. International Conference on Learning Representations (ICLR), 2022.

[5] K. Muhammad, J. Ahmad, I. Mehmood, S. Rho, and S. W. Baik, “Convolutional Neural Networks Based Fire Detection in Surveillance Videos,” IEEE Access, vol. 6, pp. 18174–18183, 2018.

[6] R. Kvyetnyy, “Enhancement of early fire detection using improved YOLOv8-based visual smoke detection model,” Proc. SPIE, 2025, doi: 10.1117/12.3100273.

[7] J. Liu, “SER-YOLOv8: An Early Forest Fire Detection Model Integrating Multi-Path Attention and NWD,” Forests, vol. 17, no. 1, Art. no. 93, 2026, doi: 10.3390/f17010093.

[8] V. Yukesh, “Detection of Forest Fire Using YOLOv8 Algorithm,” in Proc. International Conference on Data Science, Agents & Artificial Intelligence (ICDSAAI), 2025, doi: 10.1109/ICDSAAI11011830.2025.

[9] S. Zhang, “Fire-YOLO26: An NMS-free dual-stream RGB–thermal network for real-time UAV forest fire detection in smoke-challenging conditions,” Frontiers in Environmental Science, 2026, doi: 10.3389/fenvs.2026.1824597.