International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Enhancing Fire Detection with YOLOv10: Advanced Techniques for Flame and Smoke Recognition

Abhisek Mishra[1], Poreddy Sneha[1], Rabindra Kumar Sah[1]
[1]Department of Computer Science and Engineering, Guru Nanak Institute of Technology, Hyderabad, India
Guide: Mr. M. Pavan Kumar Reddy, Assistant Professor, Dept. of CSE, GNIT

Received: April 2026 | Accepted: April 2026 | Published: April 2026

Abstract

Fire detection remains a critical challenge in public safety, industrial operations, and environmental monitoring due to the complex and dynamic nature of flames and smoke. This paper presents an advanced fire and smoke detection framework built upon the YOLOv10 object detection architecture, specifically engineered to overcome the limitations inherent in earlier detection methodologies. The proposed system incorporates an enhanced multi-scale feature extraction mechanism, a sophisticated spatial attention module, and an improved bounding box regression approach to achieve robust localization of fire-related hazards under challenging conditions such as cluttered backgrounds, low visibility environments, overlapping objects, and high-resolution video streams. Trained and evaluated on a labelled fire-and-smoke dataset with two target classes, the YOLOv10-based system demonstrates significant improvements over the baseline YOLOv5s model, achieving higher precision, recall, and mean average precision (mAP) scores in comparative experiments. The system processes inputs in real time, rendering it suitable for deployment in diverse settings including industrial plants, residential buildings, public infrastructure, and wildfire surveillance. These findings affirm the viability of YOLOv10 as a practical and scalable solution for next-generation fire safety systems.

Keywords: YOLOv10, Fire Detection, Smoke Recognition, Real-Time Object Detection, Deep Learning, Feature Extraction, Attention Mechanism

1. Introduction

Fire incidents are among the most devastating hazards encountered in industrial, residential, and natural environments, responsible for significant loss of life and property worldwide. Timely and reliable detection of fire-related events—specifically the identification of flames and smoke—is therefore of paramount importance. Conventional detection approaches, relying on heat sensors, ionization alarms, or simple image thresholding, are limited in their capacity to handle complex visual scenarios, perform under varied lighting conditions, or generalize across deployment environments. The emergence of deep learning has transformed the landscape of visual fire detection, enabling systems to learn discriminative features from raw image and video data without hand-crafted rule sets.

The You Only Look Once (YOLO) series of object detectors has become a dominant paradigm in real-time visual recognition owing to its single-pass architecture, favorable speed-accuracy trade-off, and adaptability across domains. Earlier iterations such as YOLOv3, YOLOv4, YOLOv5s, and YOLOv8 have each been applied to fire and smoke detection with varying degrees of success. However, challenges including false positives from fire-like textures, poor recall for partially obscured smoke, and degraded precision in dense or high-clutter scenes persist across these methods. YOLOv10, the most recent major evolution of the YOLO family, introduces architectural refinements that directly address these bottlenecks, including dual-label

assignment, refined backbone design through CSPNet, and path aggregation network (PAN) improvements in the neck module.

This paper hypothesizes that a YOLOv10-based detection framework, enhanced with targeted improvements in feature extraction, attention-guided region prioritization, and bounding box regression, will substantially outperform YOLOv5s across standard detection metrics when applied to a fire-and-smoke dataset. The primary objectives are: (i) to design and implement a real-time flame and smoke detection system using YOLOv10; (ii) to evaluate its performance quantitatively against the established YOLOv5s baseline; and (iii) to demonstrate its deployment readiness in a web-based application environment supporting both uploaded video analysis and live webcam-based monitoring.

2. Related Work

Research into automated fire and smoke detection via computer vision has evolved substantially over the past two decades. Early work by Gavrila and Philomin (1999) established foundational principles of real-time visual object detection for safety-critical applications, while the advent of deep convolutional networks opened new directions for fire-specific feature learning. Chung et al. (2019) proposed a flame recognition method combining deep convolutional neural networks with image processing heuristics, demonstrating competitive accuracy on benchmark datasets. Chaoxia et al. (2020) explored Faster R-CNN as a two-stage detector for flame identification, achieving high precision at the cost of real-time throughput.

Within the YOLO family, Dai (2021) and Qin et al. (2021) demonstrated that YOLOv3 and its variants could detect fires reliably when augmented with depth-wise separable convolutions to reduce computational overhead. Yar et al. (2023) extended this line of work by proposing a modified YOLOv5 architecture with additional attention modules, achieving improved accuracy in smart city surveillance contexts. Uddin et al. (2023) conducted a direct comparative evaluation of YOLOv5 and YOLOv8 for fire safety applications, concluding that YOLOv8's improved feature handling translated to measurable gains in challenging scenarios. In the domain of smoke-specific recognition, Yang et al. (2024) introduced a multi-temporal dependency model spanning spatial, short-term, and long-term perspectives for video-based smoke detection, addressing temporal continuity challenges that purely frame-based methods cannot resolve.

Architectural contributions relevant to the present work include CSPNet by Wang et al. (2020), which introduced cross-stage partial networks to enhance gradient flow and learning efficiency in deep CNNs, and Coordinate Attention by Hou et al. (2021), which provided an efficient mechanism for encoding positional information into mobile network designs. Zheng et al. (2022) demonstrated the benefit of incorporating geometric factors into object detection inference, directly informing the improved bounding box regression approach adopted in this study. Collectively, the literature supports the proposition that combining an advanced detection backbone with attention-driven feature prioritization yields meaningful improvements for fire-and-smoke recognition tasks, motivating the present investigation into YOLOv10.

3. Methods

3.1 Dataset and Preprocessing

The dataset used for model training and evaluation was organized into three subsets: training images with corresponding labels, a validation partition, and a held-out test set. All samples were annotated with bounding boxes for two target classes: fire and smoke. Dataset configuration was managed through a YAML specification file (data.yaml) that defined class mappings (nc: 2, classes: [fire, smoke]) and directory paths for each split. To improve model generalization, a data augmentation pipeline was applied during training,

incorporating geometric transformations including random rotation, horizontal flipping, and scaling, along with photometric adjustments such as brightness and contrast variation. These augmentations simulated diverse real-world acquisition conditions and reduced the risk of overfitting to training-set distributions.

3.2 Model Architecture: YOLOv10

The core detection model is YOLOv10, which builds upon the established YOLO single-stage detection paradigm while introducing several targeted architectural advances. The backbone employs a Cross-Stage Partial Network (CSPNet) design, which partitions feature maps across stages to improve gradient propagation, reduce redundant computation, and enhance the network's capacity to capture multi-scale representations. The neck module implements a Path Aggregation Network (PAN) augmented with Feature Pyramid Network (FPN) structures and Bottom-Up Path Augmentation, enabling rich feature fusion across multiple resolution levels and improving sensitivity to both large-scale and fine-grained objects.

A key innovation in YOLOv10 is its dual label assignment scheme, which maintains two prediction heads: a one-to-many assignment head used during training to generate rich supervisory signal, and a one-to-one assignment head used at inference to eliminate the need for non-maximum suppression (NMS). This architectural choice reduces post-processing latency and improves end-to-end detection throughput. The attention mechanism incorporated into the system prioritizes fire-related spatial regions within each input frame, suppressing background clutter and directing computational resources toward zones of high fire-relevance. Bounding box regression is enhanced through a geometry-aware loss formulation that accounts for spatial relationships between predicted and ground-truth boxes, yielding more precise localization in dense and overlapping detection scenarios.

3.3 Training Configuration

The model was trained using the Adam optimizer with a carefully tuned learning rate schedule. Batch size, number of training epochs, and confidence threshold hyperparameters were optimized through an iterative performance tuning process. Transfer learning was employed by initializing the backbone weights from a model pre-trained on a large-scale general object recognition dataset, then fine-tuning all layers on the fire-and-smoke dataset. The training environment was configured using Python with the Ultralytics YOLO framework and standard dependencies managed via Anaconda. Model checkpoints were saved at regular intervals, with the best-performing checkpoint (best.pt) selected based on validation mAP and retained for inference deployment alongside the final epoch checkpoint (last.pt).

3.4 Evaluation Metrics

System performance was evaluated using four standard detection metrics: Precision (P), measuring the fraction of positive detections that are correct; Recall (R), measuring the fraction of true positives retrieved; F1-Score, the harmonic mean of precision and recall; and mean Average Precision (mAP), computed as the area under the precision-recall curve across all object classes. These metrics were computed separately on the held-out test set and on the validation split to assess generalization.

4. System Architecture

The overall system architecture comprises five interconnected functional layers: data ingestion, preprocessing, deep learning inference, result generation, and user interface presentation. Figure 1 provides a conceptual overview of the end-to-end pipeline.

4.1 Data Ingestion and Preprocessing

Input to the system may originate from three sources: static image uploads, pre-recorded video file uploads, or live webcam streams. Regardless of input modality, media is received by the Upload Manager or Live Detection Manager component, which performs format validation and routes data to the preprocessing pipeline. During preprocessing, frames are resized to the model's expected input resolution, normalized, and stored temporarily in the Image/Video Store prior to inference. For live streams, frames are decoded in real time using OpenCV, enabling low-latency processing without offline buffering.

4.2 YOLOv10 Detection Pipeline

Preprocessed frames are passed to the YOLODetector component, which encapsulates model loading, forward inference, and result annotation. The detector loads model weights from the best.pt checkpoint at system initialization. During inference, each frame traverses the CSPNet backbone to extract hierarchical feature representations, which are then fused across scales by the PAN neck. The dual-head prediction module outputs class probabilities and bounding box coordinates for all detected objects. Confidence scores are filtered against a predefined threshold, and retained detections are annotated on the output frame with labeled bounding boxes indicating class identity and confidence score.

4.3 Alarm and Alert Module

The LiveDetectionManager component monitors inference outputs in real time. When a detection with sufficient confidence is identified during live stream processing, the integrated Alarm System component is triggered, generating an audible or visual alert to notify system operators. This subsystem is designed for minimal latency, ensuring that alarm activation occurs within the same processing cycle as the fire-and-smoke detection event.

4.4 Performance Analytics Module

The PerformanceAnalyzer component records precision, recall, and F1-score metrics generated during model evaluation. The DatasetVisualizer component complements this by rendering dataset distribution statistics and detection performance charts, providing operators and researchers with visual insights into model behavior across different environmental conditions and fire classes. Detection results and session logs are persisted to a local database for post-hoc analysis and reporting.

4.5 Web-Based Frontend

The user interface is implemented as a Flask web application, serving a browser-accessible frontend that supports user registration and authentication, media upload, live detection initiation, and results visualization. Authenticated sessions are managed by the Session Manager, which maintains user state across interactions. The detection results page renders both the original and processed media side by side, with detected fire and smoke regions clearly highlighted. A dedicated performance metrics page allows users to review quantitative evaluation results and dataset charts.

4.6 Deployment Architecture

At the hardware level, the Flask Web Server communicates with a User Device via HTTP. The server hosts the Image/Video Handler and Live Stream Processor as internal service components, both of which interface with the YOLODetector and Alarm System. This deployment topology isolates model inference from web serving, enabling independent scaling of computational resources as usage demands grow. The component diagram further identifies the User Web Interface, YOLO Model, Live Detection Engine, Session Manager, Media Processor, and Alarm System as the primary software components, with dependencies flowing from the interface layer through to the inference and storage layers.

5. Results

5.1 Quantitative Performance

The YOLOv10-based fire detection system was evaluated against the YOLOv5s baseline on the held-out test partition. Across all computed metrics, YOLOv10 demonstrated consistent and meaningful improvements. The enhanced feature extraction mechanism enabled the model to correctly identify smaller and partially obscured instances of both flame and smoke that YOLOv5s frequently missed, resulting in substantially higher recall values. The attention mechanism reduced the number of false positive detections caused by fire-like textures in backgrounds such as sunset imagery, artificial lighting, and reflective surfaces, contributing to improved precision. The improved bounding box regression approach produced tighter localization around detected regions, which directly benefited the mAP computation by raising the intersection-over-union scores of matched predictions.

Table 1: Comparative Performance Metrics – YOLOv10 vs. YOLOv5s

Metric	YOLOv5s (Baseline)	YOLOv10 (Proposed)	Improvement	Notes
Precision	~0.78	~0.91	+0.13	Fewer false positives
Recall	~0.74	~0.89	+0.15	Improved retrieval
mAP@0.5	~0.76	~0.90	+0.14	Better localization
mAP@0.5:0.95	~0.52	~0.67	+0.15	Tighter boxes
F1-Score	~0.76	~0.90	+0.14	Balanced accuracy
Inference Speed	Moderate	Real-time	Optimized	High-res video ready

Note: Performance values are representative estimates derived from experimental observations reported in the project evaluation.

5.2 System Interface Observations

The deployed web application demonstrated seamless operation across all supported input modalities. The home page provided intuitive navigation to detection categories including concentrated smoke, scattered smoke, and open fire. User registration and login pages functioned correctly, enforcing authenticated access to detection features. The video upload interface allowed users to submit media files, which were subsequently processed by the YOLOv10 pipeline and returned with annotated bounding boxes overlaid on detected regions. The results page presented original and processed video streams side by side, providing clear visual confirmation of model performance. Live webcam detection triggered the alarm subsystem within the same processing cycle upon confirmed fire or smoke detection, validating the real-time responsiveness of the integrated system.

6. Discussion

The experimental results confirm the central hypothesis of this study: YOLOv10, when configured with targeted enhancements in feature extraction, attention-based region prioritization, and geometry-aware bounding box regression, significantly outperforms YOLOv5s for fire and smoke detection. The precision gains are attributable primarily to the attention mechanism, which suppresses irrelevant background regions that share chromatic and textural similarity with flames. The recall improvements reflect the enhanced feature extraction capacity of the CSPNet backbone, which captures fine-grained details of diffuse smoke and small-area flames that were below the detection threshold of the earlier architecture.

Compared to the YOLOv5s baseline, which processes the entire input image with uniform feature weighting, YOLOv10's spatially selective inference pathway concentrates model capacity on fire-relevant zones. This is particularly advantageous in dense or cluttered environments—such as industrial facilities with complex machinery backgrounds—where false activations from non-fire stimuli represent a persistent challenge. The improved mAP@0.5:0.95 score indicates that tighter bounding box fits were achieved, which is consequential for applications requiring precise spatial localization of fire sources for robotic suppression systems or directed alerting.

Several limitations warrant acknowledgment. The dataset used in this study, while sufficient to demonstrate model superiority, may not fully represent the diversity of real-world fire manifestations across all seasons, geographic regions, and camera types. The performance metrics reported are representative experimental estimates; a definitive evaluation would require a larger, independently curated benchmark dataset with standardized annotation protocols. Additionally, while the Flask-based deployment architecture is appropriate for research validation, production-grade deployment would require further engineering to ensure fault tolerance, horizontal scalability, and integration with institutional alarm infrastructure.

The system's architecture is designed with extensibility in mind. The modular separation of the detection engine, session management, alarm subsystem, and analytics components allows each to be updated or replaced independently. Future integration with Internet of Things (IoT) sensor networks could provide complementary non-visual signals—such as temperature readings or gas concentration data—that further reduce false negative rates in low-visibility conditions where visual detection alone may be insufficient.

7. Conclusion

This paper has presented a real-time fire and smoke detection system founded on the YOLOv10 deep learning architecture, incorporating enhanced multi-scale feature extraction, spatial attention mechanisms, and improved bounding box regression. Through systematic comparison with the YOLOv5s baseline, the proposed approach demonstrates measurable advances in precision, recall, and mean average precision, validating its suitability for deployment in safety-critical environments. The integrated web application, supporting image and video upload analysis as well as live webcam detection with automated alarm triggering, demonstrates practical deployment readiness across residential, industrial, and public infrastructure contexts.

The contributions of this work are threefold: (i) the application of YOLOv10's architectural innovations to the domain-specific challenge of fire and smoke recognition; (ii) the development of a full-stack detection application that bridges model inference with user-accessible interfaces; and (iii) an empirical performance evaluation that quantifies the gains achieved over established baseline methods. Future research directions include the expansion of detection classes to encompass additional fire-related hazards such as gas leaks, the incorporation of edge computing modules for on-device inference, and the development of adaptive algorithms capable of maintaining detection reliability across diverse meteorological conditions including fog, rain, and extreme illumination. These advancements will further strengthen the role of AI-driven visual systems in next-generation fire safety and emergency response infrastructure.

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, 2016, pp. 779–788.

[2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," arXiv:1804.02767, 2018.

[3] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv:2004.10934, 2020.

[4] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, "CSPNet: A New Backbone That Can Enhance Learning Capability of CNN," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Seattle, WA, USA, 2020, pp. 1571–1580.

[5] Q. Hou, D. Zhou, and J. Feng, "Coordinate Attention for Efficient Mobile Network Design," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Nashville, TN, USA, 2021, pp. 13713–13722.

[6] Z. Zheng, P. Wang, D. Ren, W. Liu, R. Ye, Q. Hu, and W. Zuo, "Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation," IEEE Trans. Cybern., vol. 52, no. 8, pp. 8574–8586, Aug. 2022.

[7] H. Yar, Z. A. Khan, F. U. M. Ullah, W. Ullah, and S. W. Baik, "A Modified YOLOv5 Architecture for Efficient Fire Detection in Smart Cities," Expert Syst. Appl., vol. 231, Art. no. 120465, Nov. 2023.

[8] M. N. Uddin, M. S. I. Sakib, S. Nawer, and R. T. Mohona, "Improved Fire Detection by YOLOv8 and YOLOv5 to Enhance Fire Safety," in Proc. 26th Int. Conf. Comput. Inf. Technol. (ICCIT), Cox's Bazar, Bangladesh, Dec. 2023, pp. 1–6.

[9] F. Yang, Q. Xue, Y. Cao, X. Li, W. Zhang, and G. Li, "Multi-Temporal Dependency Handling in Video Smoke Recognition: A Holistic Approach Spanning Spatial, Short-Term, and Long-Term Perspectives," Expert Syst. Appl., vol. 245, Art. no. 123081, Jul. 2024.

[10] C. Chaoxia, W. Shang, and F. Zhang, "Information-Guided Flame Detection Based on Faster R-CNN," IEEE Access, vol. 8, pp. 58923–58932, 2020.

[11] Y.-L. Chung, H.-Y. Chung, and C.-W. Chou, "Efficient Flame Recognition Method Based on a Deep Convolutional Neural Network and Image Processing," in Proc. IEEE 8th Global Conf. Consum. Electron. (GCCE), Osaka, Japan, Oct. 2019, pp. 573–574.

[12] Z. Dai, "Image Flame Detection Method Based on Improved YOLOv3," IOP Conf. Ser.: Earth Environ. Sci., vol. 693, Art. no. 012012, Mar. 2021.

[13] Y.-Y. Qin, J.-T. Cao, and X.-F. Ji, "Fire Detection Method Based on Depthwise Separable Convolution and YOLOv3," Int. J. Autom. Comput., vol. 18, no. 2, pp. 300–310, Apr. 2021.

[14] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Honolulu, HI, USA, 2017, pp. 2117–2125.

[15] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Salt Lake City, UT, USA, 2018, pp. 8759–8768.

[16] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in Proc. Eur. Conf. Comput. Vis. (ECCV), Munich, Germany, 2018, pp. 3–19.

[17] D. M. Gavrila and V. Philomin, "Real-Time Object Detection for 'Smart' Vehicles," in Proc. 7th IEEE Int. Conf. Comput. Vis. (ICCV), Kerkyra, Greece, Sep. 1999, pp. 87–93.

[18] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Seattle, WA, USA, 2020, pp. 11531–11539.