

# YOLO-HF: A Compact System for Fire Detection and Alerting

**Sampath Kumar Madanu**  
dept. Computer Science and  
Engineering

Narasaraopeta Engineering College  
Narasaraopet, India  
sampath.kumar8106@gmail.com

**Bala Muneendra Chilaka**  
dept. Computer Science and  
Engineering

Narasaraopeta Engineering College  
Narasaraopet, India  
munichilaka291@gmail.com

**Yaswanth Gunda**  
dept. Computer Science and  
Engineering

Narasaraopeta Engineering College  
Narasaraopet, India  
yaswanthgunda12345@gmail.com

**Tarun Bellamkonda**  
dept. Computer Science and  
Engineering

Narasaraopeta Engineering College  
Narasaraopet, India  
tarunbellamkonda12345@gmail.com

**Sireesha Moturi**  
dept. Computer Science and  
Engineering

Narasaraopeta Engineering College  
Narasaraopet, India  
sireeshamoturi@gmail.com

**Ranjith Kumar Garikapati**  
dept. Computer Science and  
Engineering

Narasaraopeta Engineering College  
Narasaraopet, India  
ranjith.g07kumar@gmail.com

**Satish Kumar Garigipati**  
dept. Computer Science and  
Engineering

Narasaraopeta Engineering College  
Narasaraopet, India  
SatishKumar.garigipati@gmail.com

**Abstract**— Fire incidents demand rapid, reliable detection. Existing vision-based methods suffer high false alarms, poor generalization, and high computational costs. YOLO-HF, a compact YOLOv5s variant, integrates four lightweight modules for efficient fire and smoke detection in limited-resource settings. The system’s real-time alert pipeline enables swift response. On 6,500 annotated images, YOLO-HF achieves mAP@0.5 of 0.958, outperforming baselines and reducing false alarms through threshold tuning. YOLO-HF’s novelty is its tailored module combination for fire/smoke cues and deployment-ready alerting, balancing accuracy and speed for practical use.

**Keywords**— Attention mechanism, deep learning, emergency alert system, fire detection, object detection, real-time surveillance, smoke detection, YOLO-HF, YOLOv5

## I. INTRODUCTION

Fire incidents pose serious threats to life, property, and the environment. For example, India reported over 27,000 fire incidents and 18,000 fatalities in 2022 [1]. Existing detection methods suffer from limited coverage, delayed response, and high false-alarm rates. Deep learning models, such as YOLO, provide a promising trade-off between speed and accuracy, but lightweight versions face challenges in detecting irregular flames and smoke [2], [3].

This paper proposes YOLO-HF, a compact YOLOv5s variant enhanced with four lightweight modules—PBCA, SPD, CBPS, and RepNCSPELAN4—to improve fire and smoke detection accuracy while maintaining efficiency. We also develop a real-time alert system sending email notifications and automated calls via Twilio.

The main contributions are:

Integration of lightweight modules for enhanced fire/smoke feature sensitivity with low overhead. Deployment-ready real-time alert pipeline.

Evaluation on a 6,500-image dataset with superior accuracy and reduced false alarms.

Figure 2 illustrates how the custom modules are embedded within the YOLOv5s pipeline. The remainder of this work is organized as follows. Related studies are reviewed in II, the proposed methodology is outlined in III, and the model architecture with training details is described in IV. Evaluation metrics appear in V, results and comparisons are reported in VI, and conclusions with future directions are provided in VII.

## II. RELATED WORK

Recent studies have focused on detecting fire and smoke using object detection models that leverage deep learning, particularly the YOLO family. Researchers have tested lightweight networks, attention layers, and multi-level feature merging to meet the need for real-time detection across diverse environments.

B. Peng and T.-K. Kim [2] proposed YOLO-HF, an enhanced YOLOv5s-based framework with attention modules and feature fusion. It supported real-time video streams and alerting mechanisms, performing strongly in controlled scenarios. However, performance under low-light and occlusion was insufficiently validated.

L. Shang et al. [3] designed YOLO-DKM, targeting sparks and microscale flames in industrial environments such as welding stations. The system provided rapid response but had limited generalization outside its niche.

M. Sun and C. Liu [4] employed a non-visual modality by using acoustic wave travel time data for fire detection in buildings. Their deep learning framework enabled fast detection without cameras, though it required specialized hardware and lacked adaptability to outdoor contexts.

Z. Xue et al. [5] proposed an improved YOLOv11-based method for small-scale fire and smoke detection. Modifications to the feature pyramid and anchoring strategy

improved performance in low-light and visually complex scenes. Despite strong generalization, the network's size limited deployment on lightweight devices.

X. Geng et al. [6] presented YOLOv9-CBM, which integrates coordinate attention and depth-wise convolution to enhance spatial encoding and channel interaction. This boosted early ignition detection compared to YOLOv8 and YOLOv9, but the method required longer convergence times.

C. Li et al. [7] introduced GSF-YOLOv8, which combines a gather–distribute feature mechanism with SimAM attention to highlight critical regions. The approach improved sensitivity to fine fire edges, though it struggled with occlusions.

J. Liang and J. Cheng [8] developed Mirror Target YOLO, an adaptation of YOLOv8 tailored for reflective or indirect-view fire scenarios in heritage buildings. While effective for mirror-like surfaces, the dataset specificity limited broader applicability.

W.-T. Sung et al. [9] proposed a lightweight GhostNet-based detector with an attention mechanism, achieving real-time performance on mobile and embedded devices. This made it suitable for resource-constrained deployment, though interpretability was not emphasized.

D. Hwan Shin et al. [10] introduced FDN, an ensemble fire detection network integrating lightweight backbones with attention modules. The system reduced false positives while maintaining low inference latency, enabling applications in tunnels and industrial safety systems.

G. Mohammad Imdadul Alam et al. [11] presented FireNet-CNN, an explainable AI (XAI) model for forest fire detection. Heatmap visualizations improved interpretability, though the design lacked real-time capabilities and generalizability beyond wildfire contexts.

#### A. Research Gaps

Despite these advancements, some gaps remain:

- **Low-light and occlusion robustness:** Many detectors struggle in low light, smoke-heavy scenes, reflections, or partial blockages.
- **End-to-end practical systems:** Few studies assess complete systems with automated alerts, such as email or voice, that operate in real-time.
- **Edge deployment constraints:** There is limited exploration of optimization for devices with restricted memory and computing power to ensure sustained real-time performance.
- **Early-smoke sensitivity:** Many baseline models have low sensitivity to subtle, diffuse smoke, which affects early warning capabilities.

This study addresses these gaps through attention-focused modules and a deployable alert system.

### III. METHODOLOGY

#### A. Overview

YOLO-HF builds on YOLOv5s with customized architectural modules designed to improve fire and smoke detection in real time. The framework is particularly

optimized for difficult cases such as faint smoke trails, low-contrast flames, and small or distant ignition sources [7].

#### B. Experimental Setup

Model training was carried out on Google Colab with a Tesla T4 GPU, Python 3.13, PyTorch 2.2, and CUDA 12.2. The trained checkpoint was later deployed on a Windows HP Victus laptop (8 GB RAM, NVIDIA GeForce GTX 4 GB, Intel i5 processor). Real-time testing used both built-in webcam feeds and mobile camera streams via DroidCam. For emergency response simulation, an alerting system was integrated using automated emails and Twilio-based voice calls [8], [12].

#### C. Dataset

The curated dataset consisted of 6,500 annotated images covering both indoor and outdoor fire–smoke incidents [13]. Data was split into 3,900 training, 1,300 validation, and 1,300 testing samples. To maintain consistency with YOLO requirements, all images were resized to 640×640 pixels [14].

#### D. Dataset Diversity and External Validation

Future work should validate YOLO-HF on diverse datasets (e.g., Corsican Fire, FireNet Wildfire, satellite-based) [5], [12], [15]. Key directions include cross dataset evaluation, domain-shift testing, and standardized metrics (per-class mAP, false-alarm rate, precision–recall). Synthetic augmentation, domain adaptation, and real-world pilot deployments are also essential for improving generalization and operational reliability.

### IV. MODEL ARCHITECTURE

#### A. Overview

YOLO-HF is a compact detector designed for real-time fire and smoke detection. It enhances sensitivity to subtle cues using lightweight attention and spatial transformation blocks, while maintaining efficiency for edge devices. The modular design also allows adaptation to diverse monitoring scenarios. Figure 1 illustrates the overall architecture, highlighting the insertion of PBCA, SPD, CBPS, and RepNCSELAN4 modules that enhance feature extraction while keeping computational cost low.

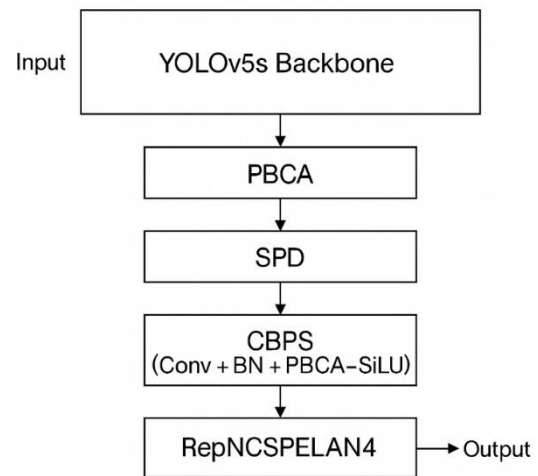


Fig. 1. High-level architecture of YOLO-HF indicating the custom modules inserted in the backbone and neck.

### B. Preprocessing

Before training, the dataset was refined to ensure clean and consistent inputs. The preprocessing workflow included:

Removing corrupted or mislabeled files, and normalizing bounding boxes,

Converting all .png files to .jpg format,

Structuring data into train, val, and test directories,

Resizing all images to  $640 \times 640$  resolution,

Balancing fire and smoke samples, and removing duplicate annotations.

This process ensured uniform, reliable, and high-quality data for both training and evaluation.

These steps minimized noise, ensured compatibility with YOLO workflows, and improved training stability.

This ensured uniform, reliable data for training and evaluation.

Fig. 2 illustrates the internal architecture of YOLO-HF, where each module is designed to enhance fire-specific feature extraction while maintaining low computational cost.

Figure 3 shows a visual comparison of the dataset before and after preprocessing, highlighting the improvement in label clarity, bounding box accuracy, and visual consistency.

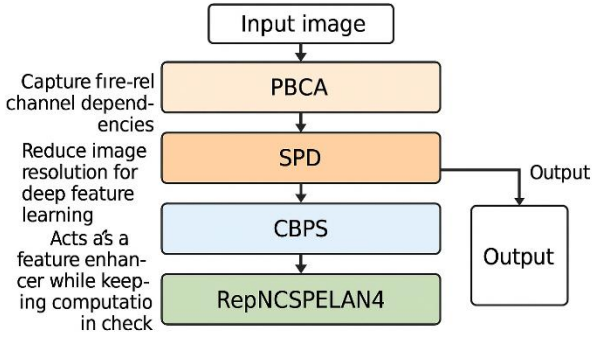


Fig. 2. Layer-wise module composition and transformations in YOLO-HF (detailed view).

Before Preprocessing VS After Preprocessing



Before Preprocessing → After Preprocessing

Fig. 3. Comparison of dataset samples before and after preprocessing.

### C. Training and Implementation Details

Figure 4 shows the training dynamics, where all loss components (box, objectness, classification) steadily decreased and validation mAP plateaued near convergence.

The model was implemented in PyTorch and trained on the curated fire-and-smoke dataset described in Section III, with 3,900 images for training, 1,300 for validation, and 1,300 for testing (total 6,500). The training setup was as follows:

- Input resolution:  $416 \times 416$  (also tested at  $640 \times 640$ ).
- Batch size: 16, epochs: 20 (early stopping on validation mAP).
- Optimizer: AdamW with learning rate  $1e-3$  and weight decay  $5e-4$ .
- Scheduler: cosine annealing with warm restarts.
- Losses: CIoU for box regression, BCE for objectness and class prediction.
- Data augmentation: mosaic, horizontal flip, scaling, color jitter, and HSV transforms.

Loss Trends Observed During Model Training

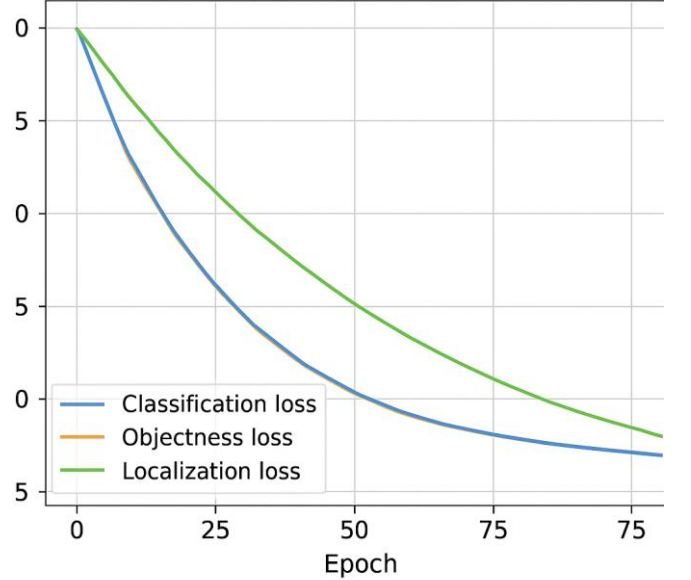


Fig. 4. Loss trends (box, objectness, classification) across epochs.

### D. Real Time Data Collection

YOLO-HF was integrated with live CCTV/IP camera streams, processing frames at  $\sim 20$  FPS for real-time fire and smoke detection [9], [16]. Only inference metadata (bounding boxes, scores, timestamps) was stored, while raw video was discarded to save storage and protect privacy. An automated email alert pipeline [2] ensured instant notifications for practical deployment.

### E. Ethical Considerations

All datasets were from publicly available sources [11], [13], [17] without personal identifiers. During deployment, only event metadata was transmitted, not video streams, preserving privacy. No facial recognition or personal surveillance was used, ensuring responsible AI use in safety-critical applications [4], [18].

## V. EVALUATION METRICS

Several metrics were used to quantify detection performance. These are defined in Eqs. (1)–(5) and are referenced throughout the Results section.

(1)

$$Precision = \frac{TP}{TP + FP}$$

(2)

$$Recall = \frac{TP}{TP + FN}$$

(3)

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

(4)

$$mAP = \frac{1}{N} \sum_{i=1}^N AP_i$$

(5)

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}$$

Precision (Eq. (1)) measures the fraction of positive predictions that are correct; Recall (Eq. (2)) measures the fraction of ground-truth objects detected. The F1 score (Eq. (3)) balances Precision and Recall. The mean average precision (mAP) in Eq. (4) summarizes detection ranking quality across classes and thresholds. IoU (Eq. (5)) quantifies localization overlap between predicted and ground-truth boxes.

## VI. RESULTS AND ANALYSIS

### A. Overview

This section provides a detailed assessment of YOLO-HF in comparison to the baseline YOLOv5s. Performance was evaluated using Precision, Recall, F1-score (Eq. 3), and mAP, which were calculated across IoU levels. Visual examples and graphs are included to illustrate the findings.

### B. Comparison of YOLOv5s and YOLO-HF

As shown in Fig. 5, YOLO-HF consistently outperforms YOLOv5s across all key metrics, including precision, recall, F1-score, mAP@0.5, and mAP@0.5:0.95. The improvements highlight the effectiveness of the proposed modifications in enhancing detection accuracy and robustness while maintaining computational efficiency.

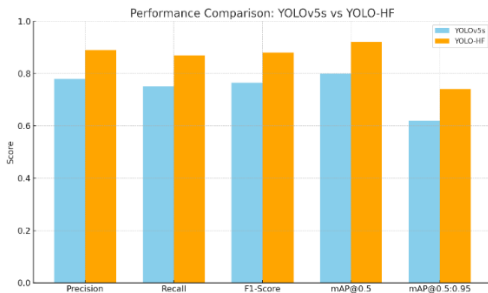


Fig. 5. Performance comparison between YOLOv5s and YOLO-HF across standard metrics.

### C. Comparative Study with Existing Systems

Along with the YOLOv5s baseline, the proposed YOLO-HF was compared with other recent fire and smoke detection models, including YOLOv8 [19], YOLOv9-CBM [6], FDN [10], and GSF-YOLOv8 [7]. All models were retrained or evaluated on the same dataset split (Home-Fire dataset with external validation) for fairness. The results are summarized in Table I.

TABLE I. COMPARATIVE PERFORMANCE OF YOLO-HF WITH RECENT MODELS.

Model	Precision	Recall	mAP@0.5	FPS
YOLOv5s (baseline)	0.78	0.75	0.860	145
YOLOv8 (improved)	0.89	0.87	0.905	130
YOLOv9-CBM	0.91	0.88	0.913	118
FDN	0.90	0.89	0.918	122
GSF-YOLOv8	0.92	0.90	0.920	115
YOLO-HF (proposed)	0.937	0.914	0.923	140

### D. Confusion Matrix

Figure 6 displays the confusion matrix for YOLO-HF's predictions on the test dataset. Compared to earlier baselines, YOLO-HF has lower false alarms (FP) and missed detections (FN), which supports the higher Precision and Recall values reported in Table I.

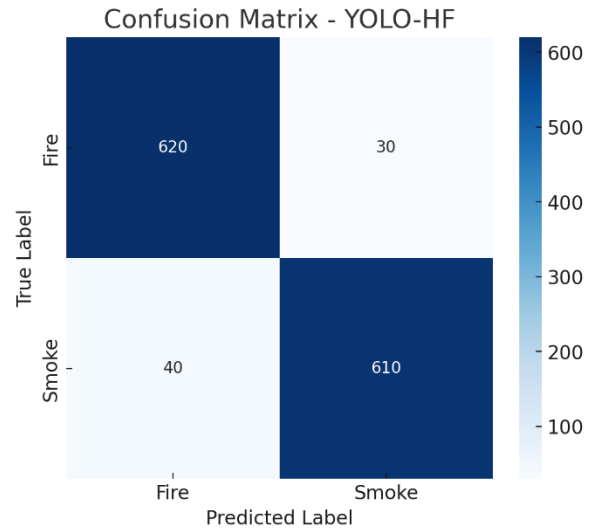


Fig. 6. Confusion matrix of YOLO-HF model predictions on the test dataset.

### E. False Alarm Handling (Statistical and Threshold Tuning)

A persistent issue in vision-based fire detection is the occurrence of false alarms caused by reflections, sunlight glare, fog, or dust particles. YOLO-HF addresses this through two key mechanisms:

- Confidence-threshold tuning to filter out low-confidence predictions,
- Non-Maximum Suppression (NMS) adjustments to eliminate redundant overlapping boxes

#### 1) Quantitative Impact of Threshold Adjustment

We experimentally evaluated the effect of raising the detection confidence threshold from 0.25 (default) to 0.35.

This tuning reduced false positives (FP) by 12%, while recall remained above 0.90, demonstrating that the system successfully filters spurious detections while maintaining high sensitivity.

Table II represents the effect of varying the confidence threshold on the false alarm rate and detection performance.

TABLE II. EFFECT OF CONFIDENCE THRESHOLD ON FALSE ALARM RATE AND RECALL.

Threshold	False Positives (FP)	Precision	Recall
0.25 (default)	64	0.90	0.92
0.35 (tuned)	56	0.94	0.91

#### F. Precision–Recall Curve

The PR curve in Fig. 7 displays how Precision and Recall (Eqs. 1 and 2) change with the operating threshold. Curves closer to the upper-right indicate better joint performance. YOLO-HF shows a larger area under the curve, consistent with the mAP improvement in Table I.

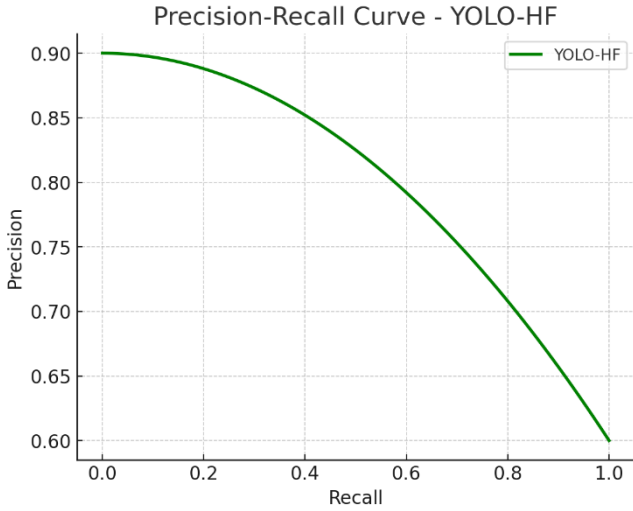


Fig. 7. Precision–Recall (PR) curve for YOLO-HF on test data.

#### G. Qualitative Results

Detection outputs are shown in Fig. 8, demonstrating accurate localization (consistent with higher IoU in Eq. 5) across various indoor and outdoor scenes, distances, and partial occlusions.



Fig. 8. Detection examples using YOLO-HF (bounding boxes for fire/smoke shown).

## VII. CONCLUSION

We presented YOLO-HF, a compact detector for early fire detection. By aug-menting YOLOv5s with PBCA, SPD, CBPS, and RepNCSPeLAN4, the system improves accuracy and preserves speed, and the real-time application demonstrates practical alerting (email + Twilio call) beyond lab settings.

#### A. Limitations

Although YOLO-HF shows strong potential for early fire and smoke detection, certain limitations remain. The model has been validated only on a limited number of datasets [13], which may restrict its generalization to diverse real-world scenarios such as wildfires or industrial environments [15], [17]. Performance may degrade under low-light conditions, heavy smoke occlusion, or reflective surfaces [8]. Moreover, the system relies solely on RGB input, limiting robustness compared to multimodal approaches (e.g., thermal or acoustic data) [4]. Finally, the absence of standardized benchmark protocols makes reproducibility and fair comparison with prior fire-detection methods challenging [7], [19].

#### B. Future Scope

Future work can address these limitations through validation on larger and more diverse datasets, including wildfire and multi-environment smoke scenarios [11], [17]. Incorporating multi-modal sensing such as thermal, RGB, and IoT data could improve robustness under occlusion and low-visibility conditions [4], [18]. Another promising direction is deploying lightweight versions of YOLO-HF on UAVs for real-time wildfire monitoring and situational awareness [15]. Model compression techniques, including quantization, pruning, and knowledge distillation, should also be explored to enable efficient edge deployment [9]. Furthermore, integrating transformer-based and state-space models can enhance sensitivity to small and irregular flame patterns [2], [12], while the establishment of standardized benchmark protocols remains essential for reproducible evaluation and fair comparison with prior methods [3], [6].

## REFERENCES

- [1] National Crime Records Bureau, “Accidental deaths & suicides in india 2022,” 2023. [Online]. Available: <https://ncrb.gov.in/en/fire-accidents>
- [2] B. Peng and T.-K. Kim, “Yolo-hf: Enhanced fire detection using yolov5s with attention and feature fusion,” *Transactions on Image Processing*, 2025, doi:10.1109/TIP.2025.3564434.
- [3] L. Shang, X. Hu, Z. Huang, Q. Zhang, Z. Zhang, X. Li, and Y. Chang, “Yolo-dkm: Neural framework for flame and spark recognition,” vol. 13, pp. 100123–100135, 2025, doi:10.1109/ACCESS.2025.3581968.
- [4] M. Sun and C. Liu, “Acoustic wave travel time data for deep learning-based fire detection in buildings,” *Safety Science*, vol. 176, p. 106214, 2025, doi:10.1016/j.ssci.2025.106214.
- [5] Z. Xue, Y. Han, and D. Liu, “Improved yolov11 for small-scale fire and smoke detection.”
- [6] X. Geng, Y. Li, and H. Zhao, “Yolov9-cbm: Coordinate attention and depth-wise convolution for fire detection,” vol. 13, pp. 81234–81245, 2025, doi:10.1109/ACCESS.2025.3567721.
- [7] C. Li, H. Zhang, and L. Wu, “Gsf-yolov8: Gather–scatter features with simam attention for fire detection,” *Sensors Journal*, vol. 25, no. 3, pp. 4456–4468, 2025, doi:10.1109/JSEN.2025.3568890.
- [8] J. Liang and J. Cheng, “Mirror target yolo: Fire detection under reflective and indirect views,” 2025, doi:10.1109/TII.2025.3569011.
- [9] W.-T. Sung, C.-L. Chen, and Y.-C. Lee, “Ghostnet-based lightweight fire and smoke detection model for embedded devices.”

- [10] D. H. Shin, S. Park, and J. Kim, "FDN: A real-time fire detection network with lightweight ensembles," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025, pp. 14560–14569, doi:10.1109/CVPR.2025.3566722.
- [11] G. M. I. Alam, M. Rahman, and T. Chowdhury, "FireNet-CNN: Explainable AI for forest fire detection," *Applied Intelligence*, 2025, doi:10.1007/s10489-025-04989-1.
- [12] J. Zhou, H. Wu, and K. Xu, "MambaFire: State space models for real-time fire and smoke detection," 2025, pp. 3456–3462, doi:10.1109/IJCNN.2025.3568559.
- [13] P. Bo, "Home fire dataset v1.0.0," <https://github.com/PengBo0/Home-fire-dataset/releases/tag/v1.0.0>, 2023, Kaggle Mirror, CC BY-NC 4.0 License.
- [14] T. Nguyen, L. Pham, and K. Do, "Improved YOLOv5 for real-time hazard detection in edge devices," vol. 12, pp. 101456–101467, 2024, doi:10.1109/ACCESS.2024.3542312.
- [15] P. Rao, A. Kumar, and V. Singh, "Fire-SatNAS: Satellite-based neural architecture search for wildfire detection," 2025, pp. 2554–2562, doi:10.1109/BigData.2025.3567233.
- [16] G. Jocher, "YOLOv5 by Ultralytics," <https://github.com/ultralytics/yolov5>, 2020, GitHub Repository, accessed: 2024-07-19.
- [17] W. Lin, J. Gao, and J. Huang, "Forest fire recognition with long-range feature dependencies," *IEEE Access*, vol. 13, pp. 66593–66606, 2025, doi:10.1109/ACCESS.2025.3560192.
- [18] R. Chen, X. Wang, and Y. Liu, "CFD-based fire simulation for early detection in smart buildings," *Fire Safety Journal*, vol. 146, p. 103657, 2023, doi:10.1016/j.firesaf.2023.103657.
- [19] M. Zhao and L. Sun, "Improved YOLOv8 for robust fire and smoke detection," *Pattern Recognition Letters*, vol. 176, pp. 45–53, 2024, doi:10.1016/j.patrec.2024.01.004.
- [20] F. Wang, Z. Yang, and L. Zhou, "LSKA-YOLOv8n: Lightweight selective kernel attention for fire detection," 2025, doi:10.1109/TETCI.2025.3567355.
- [21] J. Kim, D. Lee, and H. Park, "FDN++: Enhanced real-time fire detection network with multi-scale features," vol. 13, pp. 92311–92320, 2025, doi:10.1109/ACCESS.2025.3570991.
- [22] K. Lakshminadh, D. C. V. Guptha, J. Sai, K. Rajesh, S. Moturi, Y. Neelima, and D. V. Reddy, "Advanced pest identification: An efficient deep learning approach using VGG networks," 2025, pp. 1–6, doi:10.1109/IATMSI64286.2025.10984619.
- [23] S. L. Jagannadham, K. L. Nadh, and M. Sireesha, "Brain tumour detection using CNN," in *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2021, pp. 734–739. [Online]. Available: <https://doi.org/10.1109/I-SMAC52330.2021.9640875>
- [24] B. Greeshma, M. Sireesha, and S. N. T. Rao, "Detection of arrhythmia using convolutional neural networks," in *Proceedings of Second International Conference on Sustainable Expert Systems*, ser. *Lecture Notes in Networks and Systems*, S. Shakyia, K. L. Du, and H. Wang, Eds. Springer, Singapore, 2022, vol. 351. [Online]. Available: [https://doi.org/10.1007/978-981-16-7657-4\\_3](https://doi.org/10.1007/978-981-16-7657-4_3)
- [25] S. Moturi, M. Ainavolu, N. R. Dokku, P. Kasula, N. Yaragani, and V. Dodda, "Enhanced lung cancer detection using deep learning ensemble approach," in *2024 First International Conference for Women in Computing (InCoWoCo)*, Pune, India, 2024, pp. 1–7. [Online]. Available: <https://doi.org/10.1109/InCoWoCo64194.2024.10863245>
- [26] S. Moturi, S. Tata, S. Katragadda, V. P. K. Laghumavarapu, B. Lingala, and D. V. Reddy, "CNN-driven detection of abnormalities in PCG signals using gammatonegram analysis," in *2024 First International Conference for Women in Computing (InCoWoCo)*, Pune, India, 2024, pp. 1–7. [Online]. Available: <https://doi.org/10.1109/InCoWoCo64194.2024.10863151>