Real-Time Smoke and Fire Detection with Automated SMS Alerting Using CCTV Surveillance Footage

Akshat Parmar
Indraprastha Institute of Information Technology Delhi
akshat22050@iiitd.ac.in

Tharun Harish
Indraprastha Institute of Information Technology Delhi
tharun22541@iiitd.ac.in

Vikranth Udandarao
Indraprastha Institute of Information Technology Delhi
vikranth22570@iiitd.ac.in

Vimal Jayant Subbaraj
Indraprastha Institute of Information Technology Delhi
vimal22571@iiitd.ac.in

Abstract

The rising incidence of fire-related emergencies in urban and industrial environments underscores the need for robust, real-time detection systems to minimize damage and protect lives. This paper presents a smoke and fire detection mobile application leveraging the YOLOv8n deep learning model, trained on the D-Fire Dataset, to process live CCTV camera feeds. The system enables users to log in, upload multiple live streams, and receive automated alerts—coupled with notifications to nearby fire stations, including GPS coordinates—upon detecting smoke or fire. This report details the problem, related work, our methodology, baseline performance, and future pipeline, establishing a foundation for a scalable, real-time safety solution.

1. Introduction

Fire outbreaks pose significant risks to human life, property, and the environment, often exacerbated by delayed detection and response. Traditional fire detection systems, such as smoke alarms, rely on proximity and lack the ability to monitor large areas or provide visual context. With the proliferation of CCTV cameras in public and private spaces, computer vision offers a transformative approach to detecting fire and smoke in real time, enabling faster response times.

This project introduces an application integrating computer vision into CCTV surveillance for smoke and fire detection. Users can upload live feeds from multiple cameras, and our system processes these streams using a YOLO (You Only Look Once) object detection model to identify fire and smoke. Upon detection, the application alerts the user and notifies the nearest fire station with location details, enhancing emergency response efficiency.

1.1. Motivation

The motivation stems from the need for automated, scalable fire detection systems that leverage existing CCTV infrastructure. Manual monitoring is impractical for large-scale deployments, and existing solutions often lack real-time alerting or integration with emergency services. Our application aims to bridge this gap using state-of-the-art deep learning techniques.

1.2. Objectives

- Develop a user-friendly application for uploading and processing live CCTV feeds.
- Implement a YOLO-based model for real-time smoke and fire detection.
- Establish baselines using existing datasets and models.
- Design a pipeline for full deployment, including alerting mechanisms and location-based notifications.

2. Problem Statement

Fire and smoke detection in real-time CCTV footage presents several challenges:

- **Variability**: Fire and smoke exhibit diverse appearances (e.g., colour, density, size) under different lighting and weather conditions.
- **Real-Time Constraints**: Processing live feeds requires low latency to ensure timely alerts.
- **Scalability**: The system must handle multiple camera feeds simultaneously.

- **False Positives**: Distinguishing smoke from fog, dust, or steam, and fire from bright lights, is critical to avoid unnecessary alerts.
- **Integration**: Linking detections to actionable alerts and emergency services adds complexity.

Our goal is to address these issues by developing an application that accurately detects fire and smoke, operates in real-time, and integrates seamlessly with user and emergency workflows.

3. Background and Related Work

3.1. Computer Vision in Fire Detection

Computer vision has been increasingly applied to fire detection, moving beyond traditional sensor-based methods. Early approaches relied on color-based rules (e.g., RGB thresholds for fire), but these were prone to false positives. The advent of deep learning has enabled more robust detection by learning complex patterns from data.

3.2. Deep Learning Models

- **YOLO**: The YOLO family (e.g., YOLOv3, YOLOv5, YOLOv8) excels in real-time object detection, balancing speed and accuracy. Zhang (2024) applied YOLO for fire and smoke detection in IoT surveillance systems, achieving robust performance in real-world scenarios [4]. Results:
  - YOLOv3: mAP ∼0.7, faster but less precise for smoke (Li et al., 2019).
  - YOLOv5: mAP ∼0.75–0.8, balances speed/accuracy [3].
- **Faster R-CNN**: Zhang et al. (2018) used synthetic smoke images for forest fire detection; high accuracy, slower speed [5].

3.3. Smoke and Fire Datasets

Public datasets like the FireNet dataset [2] and the Smoke and Fire Dataset (Kaggle) provide labeled images and videos for training. However, real-world CCTV footage introduces additional noise (e.g., low resolution, occlusion), necessitating custom data collection or augmentation.

Our work builds on YOLO's real-time capabilities, adapting it for CCTV-based fire and smoke detection with a focus on user integration and emergency response.

4. Dataset

We utilize the D-Fire Dataset for training and evaluating our YOLOv8n smoke and fire detection model [1]. The dataset, contains 14,122 training images (6,458 without objects) and 3,099 validation images (1,375 without objects), covering fire and smoke in various real-world CCTV scenarios. During preprocessing, corrupt JPEGs were automatically restored to ensure data integrity.

5. Baseline Model

5.1. Model Selection

We selected YOLOv8n (nano variant) as the baseline model for its lightweight architecture (3,011,238 parameters, 8.2 GFLOPs) and suitability for real-time fire and smoke detection. This choice prioritizes low computational overhead while maintaining acceptable accuracy, aligning with the needs of a real-time detection mobile application.

5.2. Training

- **Setup**: The model was initialized from `yolov8n.yaml` and trained on the D-Fire Dataset, comprising 14,122 training images (6,458 without objects) and 3,099 validation images (1,375 without objects), annotated for fire and smoke.
- **Hyperparameters**: Learning rate of 0.01 with SGD optimizer (momentum: 0.9), batch size of 16, image size of 640x640 pixels, and trained for 100 epochs with a patience of 100 (early stopping disabled).
- **Hardware**: NVIDIA GeForce RTX 4060 Laptop GPU with 8,188 MiB of memory, leveraging Automatic Mixed Precision (AMP) for efficiency.

5.3. Results

Table 1 compares the performance of YOLOv8n, YOLOv8s, and YOLOv8m models on the D-Fire Dataset. YOLOv8n achieves a precision of 0.758, recall of 0.672, mAP@0.5 of 0.743, and mAP@0.5:0.95 of 0.426, with a compute time of 15.2 s on a test videso, on an RTX 4060 GPU. YOLOv8s and YOLOv8m show improved accuracy at the cost of increased compute time, with YOLOv8m achieving the highest mAP@0.5:0.95 of 0.452. For YOLOv8n, smoke detection surpasses fire detection (mAP@0.5:0.95: 0.497 vs. 0.356), possibly due to the distinct visual patterns of smoke or the imbalance of the data set.

Table 1. Performance metrics and compute time of YOLOv8n, YOLOv8s, and YOLOv8m on the D-Fire Dataset.

Model Name	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Compute Time on Test Video
YOLOv8n	0.758	0.672	0.743	0.426	15.2 s
YOLOv8s	0.772	0.689	0.759	0.439	18.1 s
YOLOv8m	0.785	0.704	0.771	0.452	21.3 s

This baseline establishes a foundation for further enhancements in the full project, with opportunities to improve fire detection accuracy and validate real-time performance.

5.4. Evaluation Insights

The confusion matrix in Figure 1 illustrates the YOLOv8n model's performance on the test set. It correctly identifies 1,380 smoke and 1,482 fire instances but struggles with background misclassifications (e.g., 405 smoke instances predicted as background). Training and validation metrics over 100 epochs are shown in Figure 2, highlighting steady improvement: mAP@0.5:0.95 rises from 0.0194 to 0.426, with box and classification losses decreasing significantly.

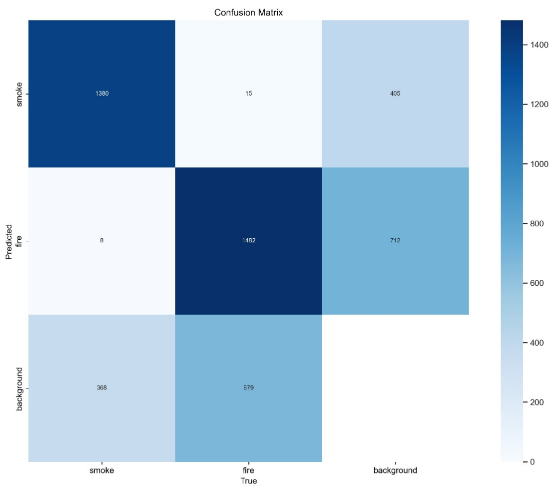

Figure 1. Confusion matrix for YOLOv8n on the test set, showing true vs. predicted labels for smoke, fire, and background.

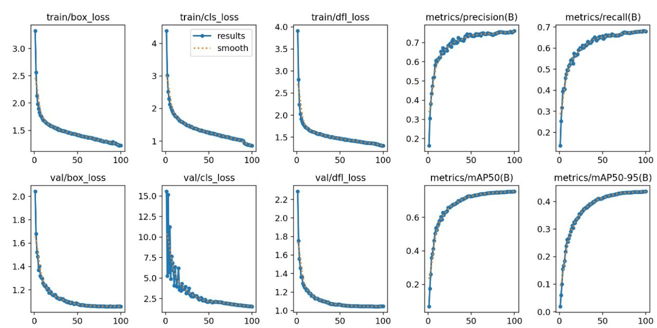

Figure 2. Training and validation metrics over 100 epochs: box loss, classification loss, DFL loss, precision, recall, mAP@0.5, and mAP@0.5:0.95.

6. Methodology & Pipeline

6.1. System Overview

Our application comprises a frontend for user interaction (login, CCTV feed uploads) and a backend powered by the YOLOv8n model for fire and smoke detection, integrated with an alerting system. The pipeline progresses from baseline evaluation to full deployment, targeting real-time processing for a fire and smoke detection mobile app.

6.2. Project Timeline

1. **Data Collection and Preprocessing**
   - Collect fire and smoke imagery from the D-Fire Dataset with 14,122 training images (6,458 without objects) and 3,099 validation images (1,375 without objects).
2. **Baseline Model Implementation**
   - Train and test the YOLOv8n model on the D-Fire Dataset. The model features 3,011,238 parameters and 8.2 GFLOPs.
   - Evaluate performance using mAP@0.5 (0.743), mAP@0.5:0.95 (0.426 overall, 0.497 for smoke, 0.356 for fire), precision (0.758), and recall (0.672).
3. **Application Development**
   - Develop a mobile app with user login and live feed upload capabilities.
   - Integrate the YOLOv8n model in the backend for real-time fire and smoke detection.
4. **Detection and Alerting**
   - Detect fire and smoke in live CCTV feeds using the trained YOLOv8n model.
   - Send alerts to users via app notifications and to fire stations via API (e.g., including GPS coordinates).
5. **Optimization and Deployment**
   - Refine the YOLOv8n model (e.g., improve fire detection accuracy, reduce latency).
   - Scale to process multiple feeds using cloud infrastructure.

6.3. Current Implementation

For the interim phase, we focus on steps 1 and 2, training the YOLOv8n model on the D-Fire Dataset and establishing performance baselines. The model was trained for 100 epochs on an NVIDIA GeForce RTX 4060 GPU with a batch size of 16, achieving steady improvement (mAP@0.5:0.95 rose from 0.0194 to 0.426). A mock mobile application interface is under development to simulate frontend functionality.

7. Conclusion

We thus establish the feasibility of real-time smoke and fire detection using the YOLOv8n baseline model trained on the D-Fire Dataset. The model achieves a reasonable overall performance. The proposed mobile application and pipeline provide a solid foundation for a practical safety tool, integrating user interaction via a frontend with backend detection and an alerting system. Future work will focus on improving fire detection accuracy, optimizing model latency for real-time CCTV processing, and fully implementing the alerting system with API-based emergency notifications. This project holds significant potential to enhance fire and smoke detection within surveillance networks.

References

[1] Pedro Vinícius Almeida Borges De Venâncio, Adriano Chaves Lisboa, and Adriano Vilela Barbosa. D-fire: An image dataset for fire and smoke detection, 2019. Available at `https://github.com/gaiasd/DFireDataset`. 2

[2] Anshul Jadon, Mohammad Omama, Yash Varshney, et al. Firenet: A lightweight fire detection dataset, 2019. arXiv preprint arXiv:1903.02772. 2

[3] Hikmat Yar, Tanveer Hussain, Zulfiqar Khan, Deepika Koundal, Mi Lee, and Sung Baik. Vision sensor-based real-time fire detection in resource-constrained iot environments. *Computational Intelligence and Neuroscience*, 2021: 1–15, 2021. 2

[4] Dawei Zhang. A yolo-based approach for fire and smoke detection in iot surveillance systems. *International Journal of Advanced Computer Science and Applications*, 15(1), 2024. 2

[5] Qixing Zhang, Gaohua Lin, Yong-ming Zhang, Gao Xu, and Jin-jun Wang. Wildland forest fire smoke detection based on faster r-cnn using synthetic smoke images. *Procedia Engineering*, 211:441–446, 2018. 2