Fire and Smoke Detection System

---

1. System Architecture and Operation

The system uses a YOLOv8s model (`best.pt`) trained to detect two classes:

- **Class 0:** Fire
- **Class 1:** Smoke

Key detection logic:

- **Confidence Threshold:** ≥ 0.5
- **Time Thresholds:** Fire for ≥ 4 seconds, Smoke for ≥ 5 seconds
- **Frame Thresholds:** Fire for ≥ 120 frames, Smoke for ≥ 150 frames
- **Grace Period:** 2-second buffer after detection loss to avoid false resets

When an alert is triggered:

- Plays an audible alert
- Saves an annotated frame
- Logs the detection event with a timestamp

---

2. Model Performance (Validation Results)

Training Details:

- **Epochs:** 25
- **Batch Size:** 8
- **Hardware:** Tesla T4 GPU(Collab)

Validation Dataset:

- 2,730 images
- 6,798 instances

Metric	Overall	Fire	Smoke
Precision	0.800	0.745	0.855
Recall	0.676	0.614	0.738
mAP50	0.773	0.712	0.833
mAP50–95	0.547	0.473	0.621

Inference Speed: ~8.9 ms per image (~112 FPS on Tesla T4)
The model shows reliable real-time detection, with stronger performance for smoke. Further training and tuning will improve fine localization and reduce false positives.

3. Code Structure and Key Features

Key Python implementation components:

Load Model & Predict:

```
model = YOLO('11.pt')
results = model.predict(...)
```

Video Capture:

```
cap = cv2.VideoCapture(0)
```

Detection Filtering:

```
if class_id in [0, 1] and conf > 0.5:
    # Process detection
```

- **Alert Logic:** Manages time/frame thresholds and grace period
- **Alert Actions: `play_alert_sound()`, `cv2.imwrite()`**, logging
- **FPS:** Real-time calculation and overlay

---

4. Essential Enhancements for Production

Recommended improvements for robust production use:

Extended Training:

- Train for 100 epochs with early stopping(Needs better GPU)
- Expand dataset: thousands more images with diverse conditions and accurate annotations

Adequate Hardware:

- GPU: 12–16 GB VRAM minimum (RTX 3060/3080/4070). For larger models: 24 GB+ (RTX 4090, A100)
- System RAM: 16–32 GB
- Storage: Fast NVMe SSD
- GPU Access: Consistent cloud GPU usage (Colab Pro+, AWS, GCP)

Model Optimization:

- Tune hyperparameters, augmentation, and confidence thresholds
- YOLOv8m or YOLOv8l for higher accuracy if speed allows and YOLOv8x better

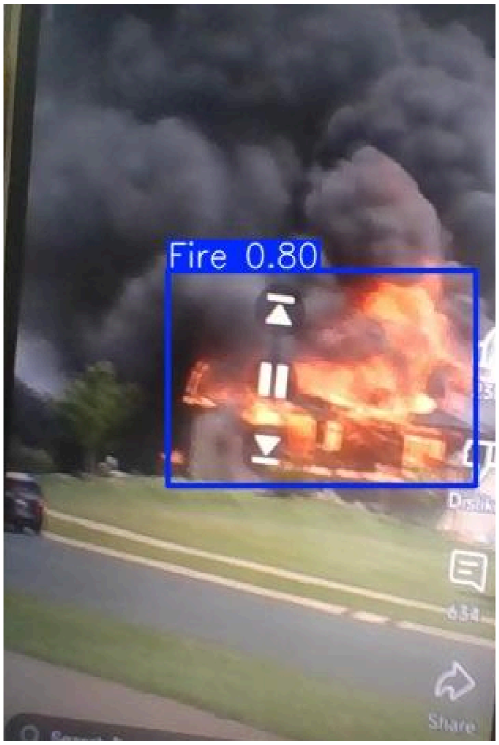
Advanced Alerting:

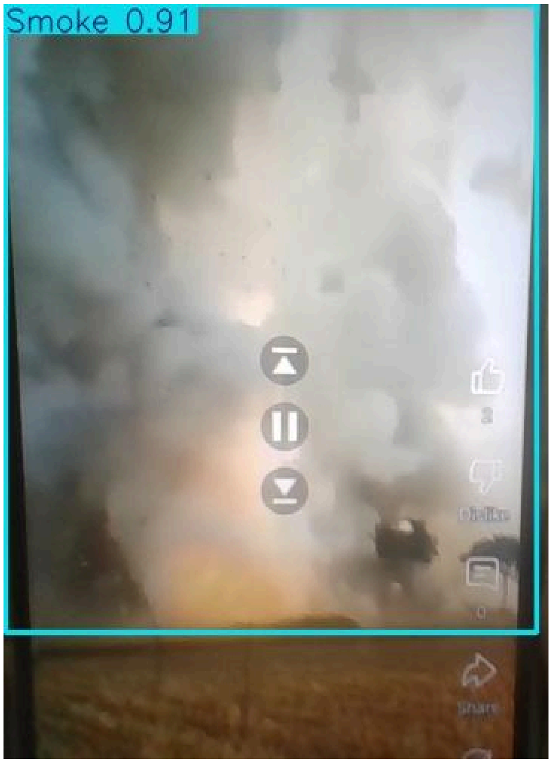
- Integrate email/SMS notifications
- Build detailed logging for audits and reports

Achieved Results (Example Screenshots)

When documenting results, include clear annotated examples:

- **Fire Detection:** Bounding box and alert overlay
- **Smoke Detection:** Bounding box and alert overlay


Fire 0.80
Dislik
634
Share


Smoke 0.91
Dislike
Share