Disaster Detection Framework for Smart Cities : An AI YOLOv8 and IoT Approach

Hossam Kamel

Swiss School of Business and Management Geneva
SSBM

Contents

01

I. Introduction

Smart Disaster Detection in Cities

Rapid urban population growth demands innovative disaster preparedness.

Smart cities utilize technology to enhance safety and resilience.

AI, IoT devices, and real- time data processing contribute to improved urban safety.

Ongoing research aims to monitor and manage in- city disasters at both macro and micro levels.

Purpose of the Paper

Present a comprehensive disaster detection framework.

Employ IoT- based environmental sensing and AI-driven image processing.

Use real- time data transmission to an IoT operations center.

Focus on detecting and responding to fires and floods using various technologies.

02

Challenges in Traditional Disaster Detection Systems

Sensing Devices

Conventional disaster detection systems deploy various types of sensing devices, which can be categorized into different groups.

Static sensing devices like seismometers and weather sensors cumulate data over time.

Mobile sensing devices like smartphones and UAVs can be deployed strategically.

Technological Challenges

Despite the technological advancements in disaster sensing and detection devices, the currently deployed disaster detection systems face several challenges in effectively detecting disasters

01

Slow disaster response time due to reliance on fixed sensors, communication delays, and centralized processing.

02

Lack of integration among different disaster detection systems.

03

Deficiency in high- resolution data hampers precise disaster localization.

Operational Challenges

01

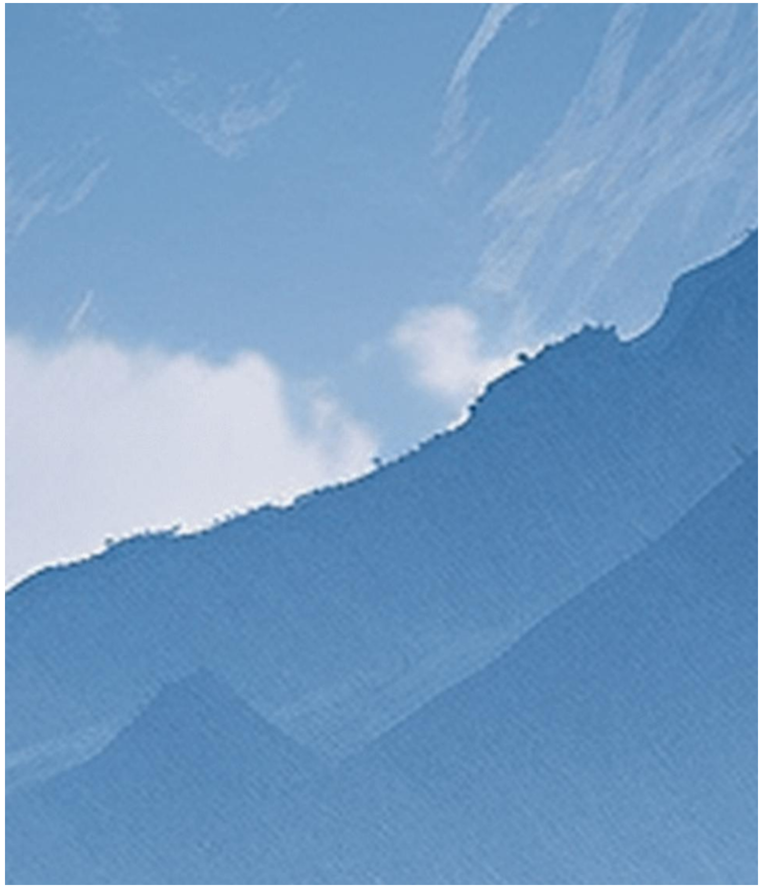
Limited trained human resources.

02

High maintenance costs for sensing devices.

03

Low community engagement in emergency alert systems.



03

Key Components of Smart Cities Frameworks

Smart City Frameworks

01

The foundation for the frameworks is a definition of the essential components of smart city systems, describing what needs to be integrated and how this integration works

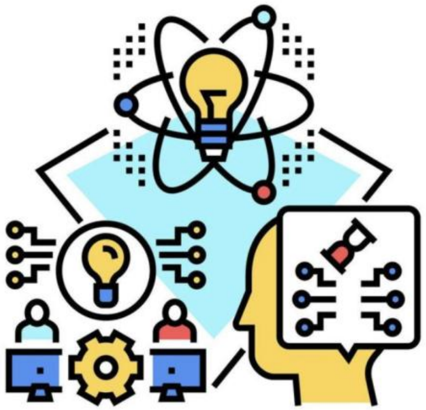
Smart City Pillars

A

1-Smart Governance & Education
2-Smart Living,
3-Smart Healthcare,
4-Smart Transportation,
5- Smart Economy,
6-Smart Environment.

B

Promote technological advancement, innovation, and urban quality of life.



C

Aim for sustainable future and community engagement.

Proposed Framework Layers

01. IoT Sensor and Camera LayerRaspberry Pi- based edge AI devices, CCTV cameras, environmental sensors.

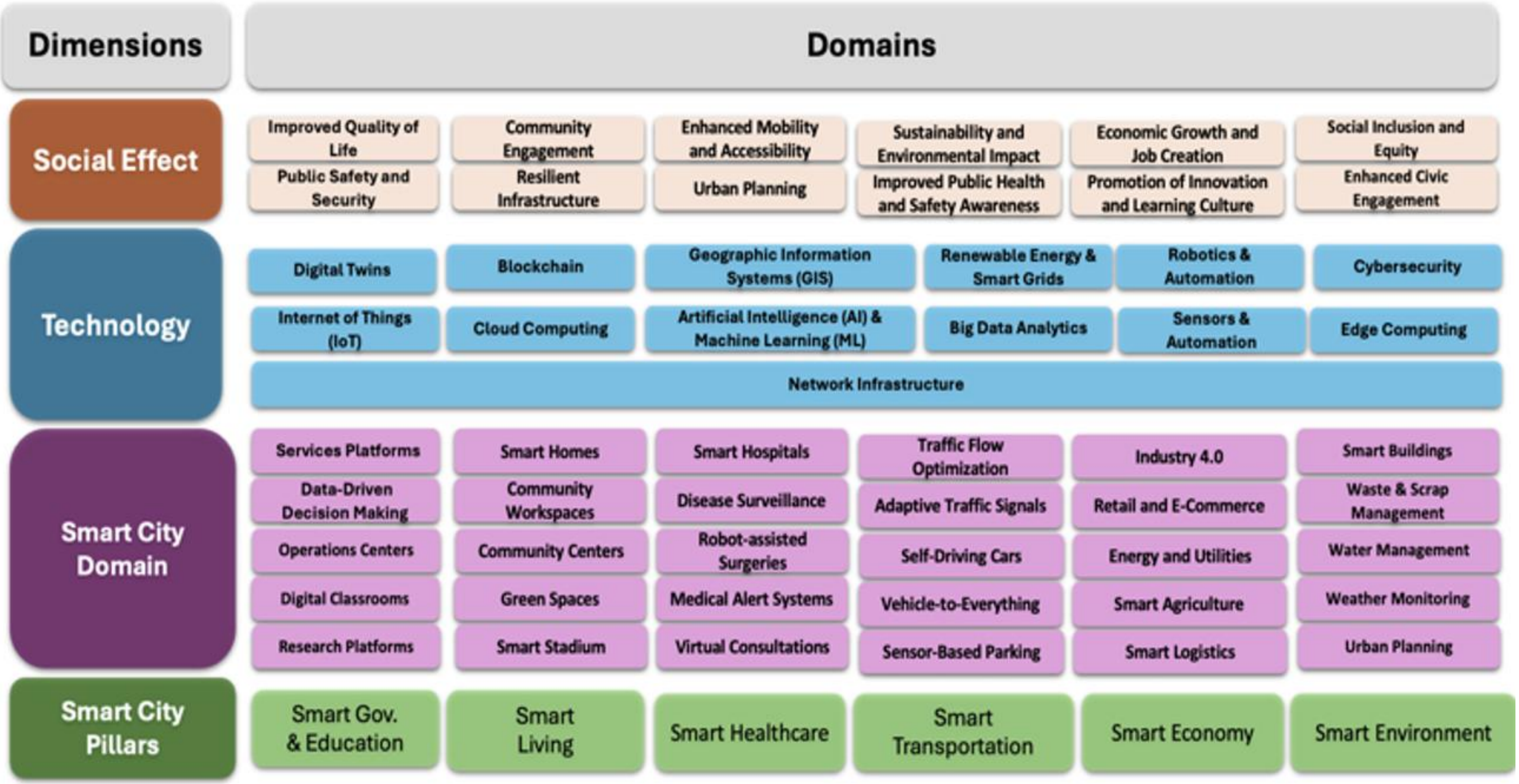
02. AI Processing and Detection LayerYOLOv8 algorithm on Raspberry Pi detects fire and flood incidents.

03. IoT Operations Center and Response Layer Event data transmission, emergency service notifications, and resident alerts.

04

Smart City Framework

Smart City Framework



Smart City Framework -2

The smart city framework

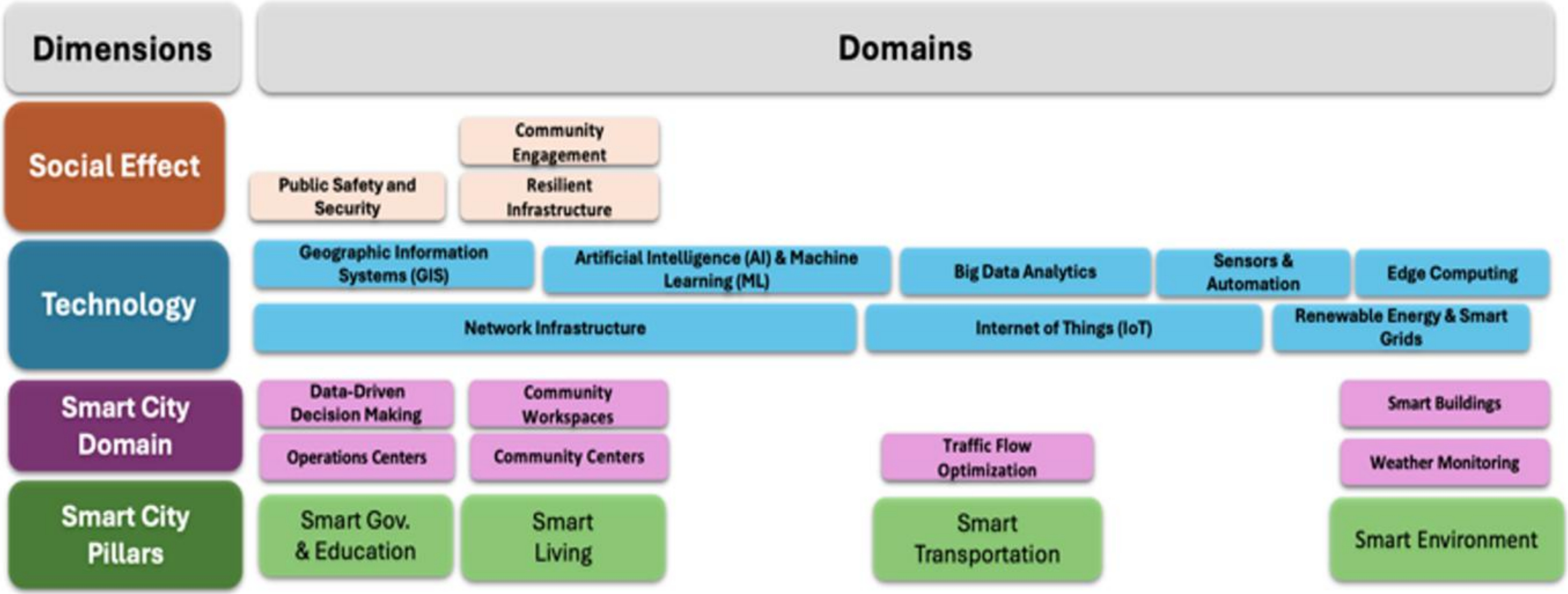
- It is an adaptive model
- Integrates key components of smart city development.
- Combining core pillars, specialized domains, advanced technologies, and social impacts.
- When focusing on a specific activity—such as waste management or smart healthcare only relevant elements are retained.
- Streamlining efforts and emphasizing expected social benefits like public health, sustainability, and economic growth.

YOUR LOGO

05

Proposed Disaster Detection Smart City Framework

Proposed Disaster Detection Framework (DDF)



Proposed Disaster Detection Framework (DDF) - 2

A Disaster Detection Framework (DDF) for Smart Cities, as, integrates advanced technologies and smart city components to enhance fire prevention, detection, and response while promoting sustainability

Integration of IoT, GIS, weather monitoring, AI, and edge computing for early disaster detection.

Coordination through IoT operations centers and community engagement.

This holistic approach combines technology, smart city domains, and proactive strategies to minimize fire risks and enhance safety in urban environments.

06

AI & ML Role in DDF

Enhancing Urban Resilience

AI and ML improve monitoring, identification, and resource estimation for disasters.

AI and ML offer data analysis tools that can enhance performance beyond traditional methods, fostering innovation in diverse fields.

Data Quality and Privacy

01

Challenges in data quality and privacy concerns.

Many cities provide access to public data, such datasets are often not collected with the same parameters or standards, limiting their comparability

YOLOv8 Real-Time Object Detection

“

Smart Cities Need Safety: Urban growth increases flood and fire risks.
AI Object Detection: Technology like YOLO leads modern detection systems.
YOLOv8 Advantage: Fast, accurate detection with real-time response.
Powerful Architecture: Five-stage model ensures high performance.

”

YOUR LOGO

07

Fire and Flood Detection in Smart Cities Using YOLOv8

Fire Detection

"Challenges in urban fire detection due to obstructions and rapid flame spread.

"Utilization of existing CCTV cameras for cost- effective fire detection and real- time surveillance.

Flood Detection

Complexity of flood detection involving simultaneous events.

Real- time image processing- based flood detection using YOLOv8.

Integration with other sensors and GIS for improved flood monitoring.

08

IoT Role in the Detection

Role of Raspberry Pi

Raspberry Pi as an IoT edge device running YOLOv8 for real- time hazard detection.

Local processing for reduced latency and quick response.

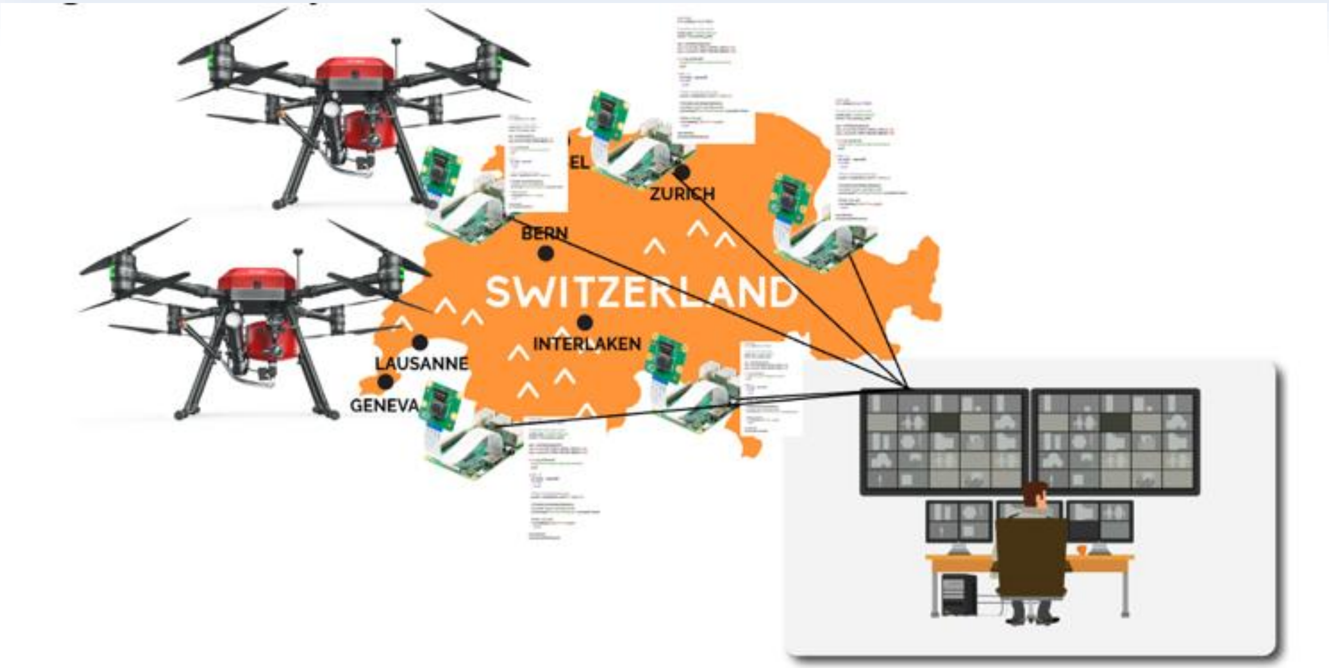
YOUR LOGO

09

The Proposed Model Algorithm

Solutions Architecture

Integration of UAVs, cameras, and IoT network for real- time fire detection and response.



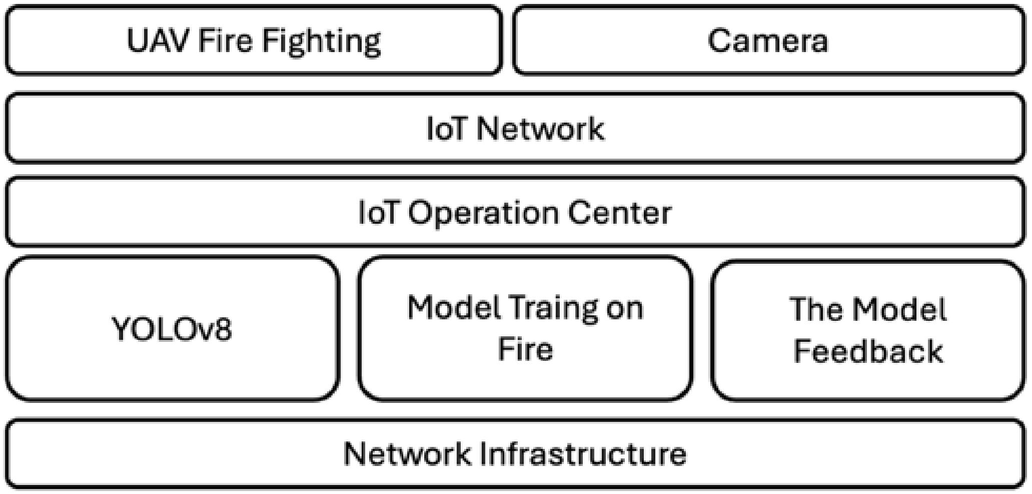
The Proposed Model Algorithm

YOLOv8 for Detection: Real-time fire detection using UAV and camera footage.

Smart IoT Network: Enables communication and rapid response coordination.

Self-Improving System: Continuous training and feedback boost accuracy.

Seamless Integration: Combines AI, IoT, and UAVs for smarter fire safety.



YOUR LOGO

10

Examine the Algorithm

Initial and Improved Experiments

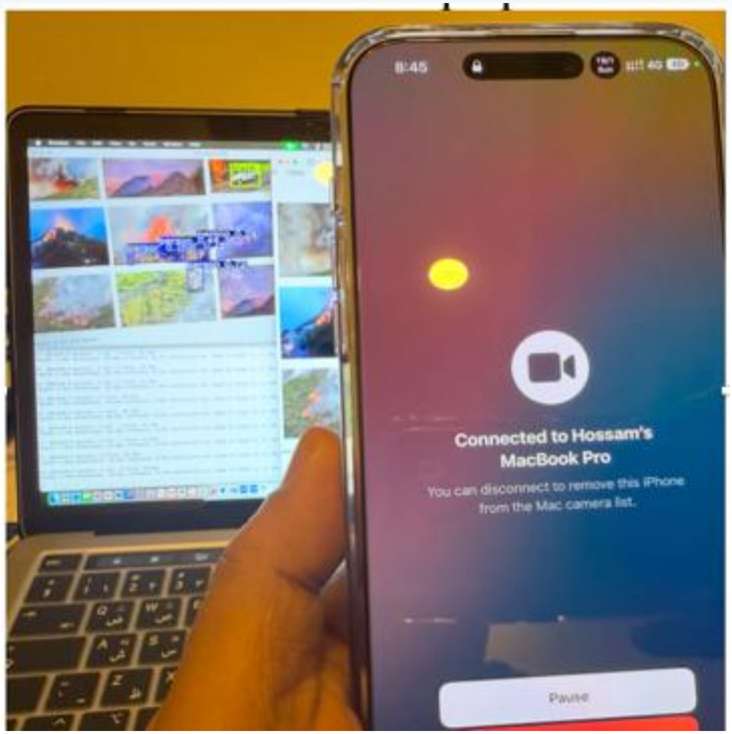
01 Initial experiments with YOLOv8 on a picture without fire- specific learning.

02 Improved experiments with fire learning detection demonstrating higher accuracy.

```python
import cv2
from ultralytics import YOLO

# Load the multi-class model
model_path = '/path/to/best.pt'
model = YOLO(model_path)

cap = cv2.VideoCapture(0)
cap.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)

if not cap.isOpened():
    print("Error: Could not open the camera")
    exit()

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # Perform fire & flood detection
    results = model(frame, conf=0.7, iou=0.4)

    # Annotate and display detections
    annotated_frame = results[0].plot()
    cv2.imshow('Fire & Flood Detection', annotated_frame)

    # Press 'q' to quit
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

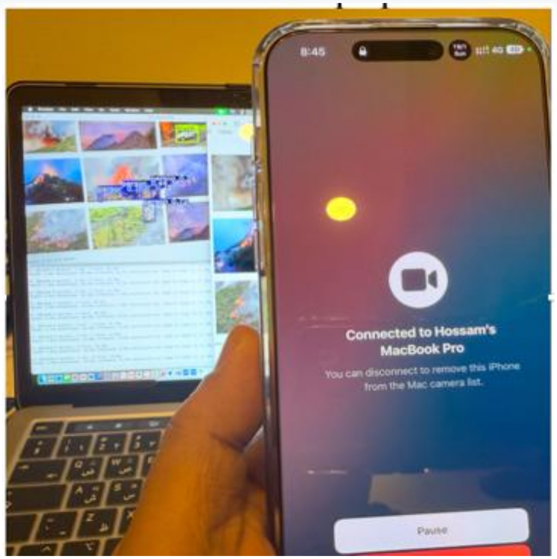
Practical Implementations

Implementation using a laptop and mobile phone to simulate real-world scenarios.



Practical Implementations - 2

- **Laptop:**
  Used for running the YOLOv8 model, processing the video input, performing fire detection, and displaying results. It handled model inference and visual output using local computing resources.
- **Mobile Phone:**
  Served as a real-time camera to capture live video streams of the environment. The phone acted as a flexible, portable input device, simulating practical deployment in smart city scenarios.

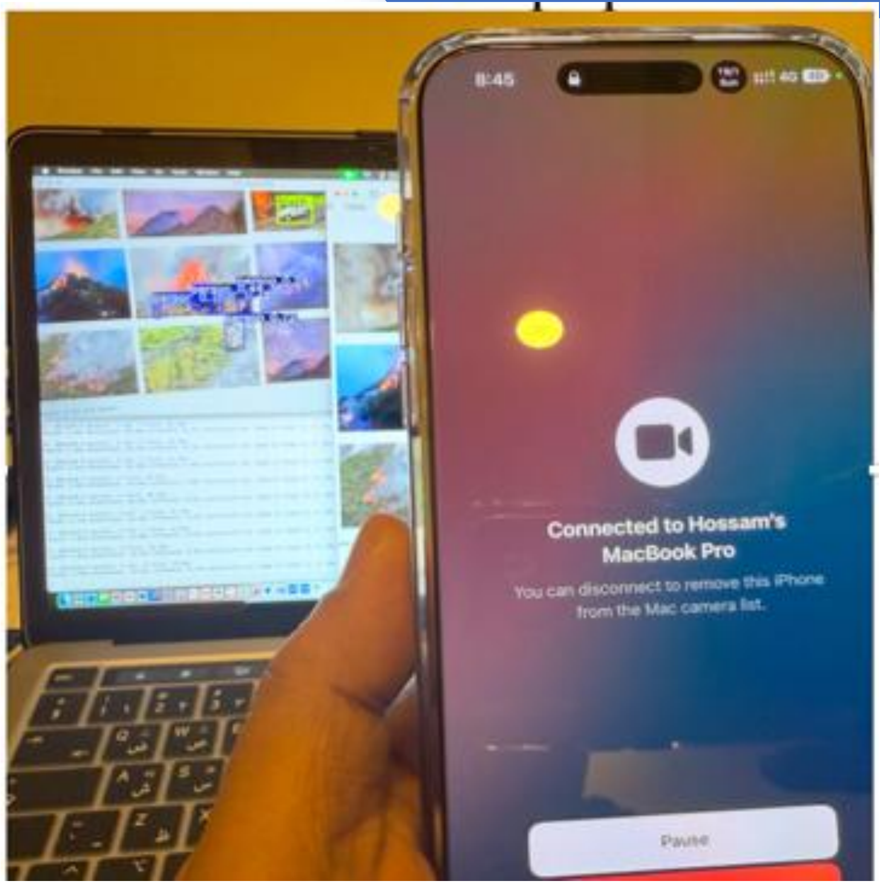


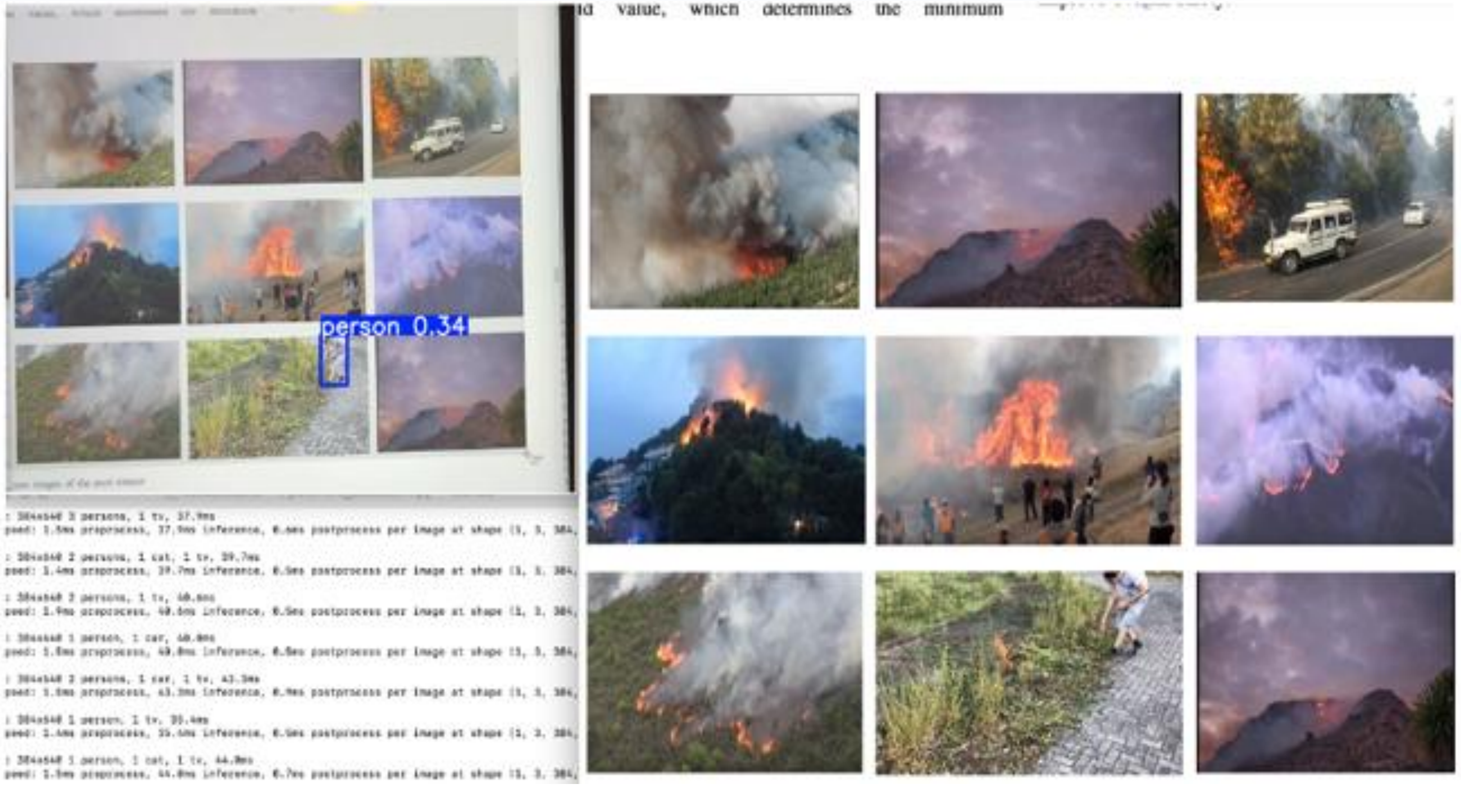
Practical Implementations - 3

Initial Test (Without Fire Training)

•The YOLOv8 model was used "as-is" without specific fire detection training.

•**Result**: It detected general objects (e.g., people, furniture) but **failed to recognize fire**.





Practical Implementations - 4

Fire-Specific Training Added

•The model was retrained with labeled fire images to help it learn fire features.

•**Result**: After training, the model **successfully detected fire areas** in test images.

11

Conclusion and Future Work

Summary of Contributions

01

- **Comprehensive Framework**: Targets both fire and flood detection using AI, IoT, and real-time video analysis.
- **YOLOv8-Powered:** Utilizes the advanced YOLOv8 algorithm for high-speed, accurate object recognition.
- **Real-World Experiment:** Demonstrated using a laptop and mobile phone simulating real-time detection in rural settings.

02

- **Flood Detection Roadmap**: Ongoing plans to train on flood datasets and integrate IoT water level sensors.
- **Urban Integration Plan:** Future work includes connecting the system to CCTV city surveillance for urban fire detection.

Thanks