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

Hossam Kamel
Swiss School of Business and Management Geneva
SSBM Geneva
Geneva, Switzerland
email: dr.hossamkamel.a@gmail.com

***Abstract*— Disaster detection is vital for smart city resilience and public safety. This paper presents a framework for detecting fire and flood incidents using the You Only Look Once version 8 (YOLOv8) algorithm on a Raspberry Pi Internet of Things (IoT) device, which transmits data to IoT operation center. An initial experiment using a laptop and mobile phone demonstrated the effectiveness of machine learning in fire detection.**

Keywords-Smart Cities; Disaster Detection; Fire Detection; Flood Detection; IoT; YOLOv8; Artificial Intelligence (AI); Machine Learning (ML); Raspberry Pi.

I. INTRODUCTION

The rapid expansion of urban populations has placed increasing pressure on city infrastructure, requiring innovative solutions to enhance resilience and disaster preparedness. Smart cities leverage technology to mitigate risks posed by natural and man-made disasters, integrating AI, IoT devices, and real-time data processing to improve urban safety. The goal of smart cities is to leverage technology and data analytics to improve the quality of life in urban areas [1]. There are several ongoing research efforts focused on using technology to monitor and manage in-city disasters, either at the macro level or for specific types of disasters [2].

This paper presents a comprehensive disaster detection framework that integrates IoT-based environmental sensing, AI-driven image processing, and real-time data transmission to an IoT operations center. The framework is designed to detect and respond to disasters such as fires and floods using a combination of Closed-Circuit Television (CCTV) surveillance, edge AI processing on Raspberry Pi devices, and automated alerts to emergency responders.

The core contribution of this work is the implementation and validation of an AI-based disaster detection system within the proposed smart city framework. Specifically, we evaluate the effectiveness of YOLOv8, a state-of-the-art object detection algorithm, in identifying fire hazards using real-time image analysis. Additionally, we discuss the potential for extending the system to flood detection.

The remainder of the document is organised as follows: Section II discusses the challenges in traditional disaster detection systems, emphasizing the need for real-time AI-based solutions. Section III introduces the proposed disaster detection framework, outlining its key components and role in smart city resilience. Section IV details the system implementation, explaining how YOLOv8 and IoT components work together for fire and flood detection. Section V presents experimental results, highlighting the effectiveness of the proposed system in identifying fire incidents. Finally, Section VI discusses the broader implications, including potential improvements for flood detection and integration into smart city infrastructure.

II. CHALLENGES IN TRADITIONAL DISASTER DETECTION SYSTEMS

Conventional disaster detection systems deploy various types of sensing devices, which can be categorized into different groups. Static sensing devices are permanently located at a specific geographical site, cumulating data over time. Examples include seismometers and weather sensors. On the other hand, mobile sensing devices are portable and can be strategically deployed at various locations or moved over time. Such devices include smart phones and Unmanned Aerial Vehicles UAVs [3].

Despite the technological advancements in disaster sensing and detection devices, the currently deployed disaster detection systems face several challenges in effectively detecting disasters [4]. These challenges can be categorized into three groups: technological challenges, operational challenges, and situational challenges [5].

The first technological challenge is slow disaster response time. The disaster response time consists of three time delays: sensing time, processing time, and communication time. The traditional disaster detection systems take considerable time to detect disasters due to their reliance on fixed and static sensors. These sensors must first sense the arrival of a disaster. Then, the cumulative data are sent to a centralized location for processing, which results in significant communication delays. The second technological challenge is a lack of integration. Currently, the disaster detection systems in developed countries do not integrate with one another, even though multiple disasters are detected by using different devices on numerous occasions. As a result, significant time delays occur in detecting and preventing disasters. It is vital that systems be developed that can share disaster information in real time. The third technological challenge is a lack of high-resolution data. Currently, most of the data used in the detection systems are remotely gathered data, which restrict the detection systems from generating high-resolution data [6]. This limitation results in difficulties in precisely locating the site of a disaster.

The first operational challenge is the lack of trained human resources [7]. Most of the developing countries' governments have limited resources to employ. The second operational challenge is maintenance costs [8]. The structural deterioration of the sensing devices is a key reason for inoperable disaster detection systems. The third operational challenge is community engagement [9]. A lack of community awareness and participation can lead to ignored emergency alerts, delaying evacuation and reducing disaster response effectiveness. While many disaster detection systems have been successfully implemented in developing countries, the local community remains largely uninterested. As a result, several systems are rendered useless because the community does not provide adequate resources for operating the systems.

III. Key Components of Smart Cities Frameworks

In recent years, numerous cities have adopted smart city frameworks, which outline principles, policies, and goals for smart city development [10]. 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 [11].

The comprehensive framework for establishing advanced smart city systems is fundamentally centered around six key and crucial Smart City Pillars: Smart Governance & Education, Smart Living, Smart Healthcare, Smart Transportation, Smart Economy, and Smart Environment [12]. These vital pillars serve as the essential focus areas, meticulously aimed at significantly enhancing urban life, fostering community engagement, and ensuring a sustainable future for all residents. Each of these pillars plays a unique and impactful role in fostering innovation while improving overall quality of life within metropolitan areas. By integrating these pillars effectively, cities can promote technological advancement and create a conducive environment for growth and development. Furthermore, the interconnectedness of these pillars increases the potential for synergies, enabling cities to tackle complex urban challenges more efficiently and sustainably. The holistic approach of the framework ensures that every aspect of city living is considered, providing a comprehensive strategy for modern urban management and planning, thereby encouraging a well-rounded development that benefits everyone.

The proposed framework consists of three core layers, each playing a distinct role in disaster detection:

1. IoT Sensor and Camera Layer – This includes Raspberry Pi-based edge AI devices, CCTV cameras, and environmental sensors deployed across the city. These devices capture real-time visual and environmental data.
2. AI Processing and Detection Layer – The captured data is analyzed using YOLOv8 running on Raspberry Pi to detect fire or flood incidents. This edge computing approach ensures faster detection and reduces reliance on cloud processing.
3. IoT Operations Center and Response Layer – Detected events are transmitted via wireless or wired networks to a central operations center, where emergency services are notified. Alerts can also be sent to residents via mobile applications or warning systems.

By structuring the solution within this framework, we ensure that the proposed system is scalable, adaptable, and aligned with existing smart city initiatives.

IV. Smart City Framework

A plethora of smart city frameworks can be unearthed through a wide-ranging investigation of publicly accessible smart city features, models, methodologies, scaffolds, architectures, and pilot schemes [13] .

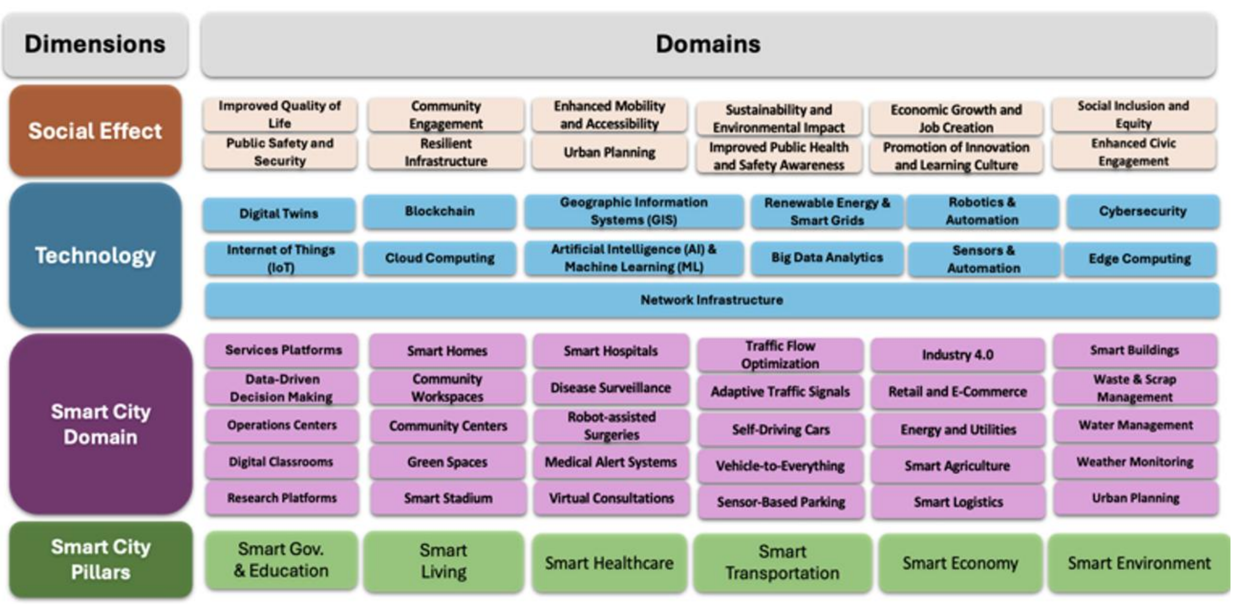

Figure 1. Smart City Framework [14] .

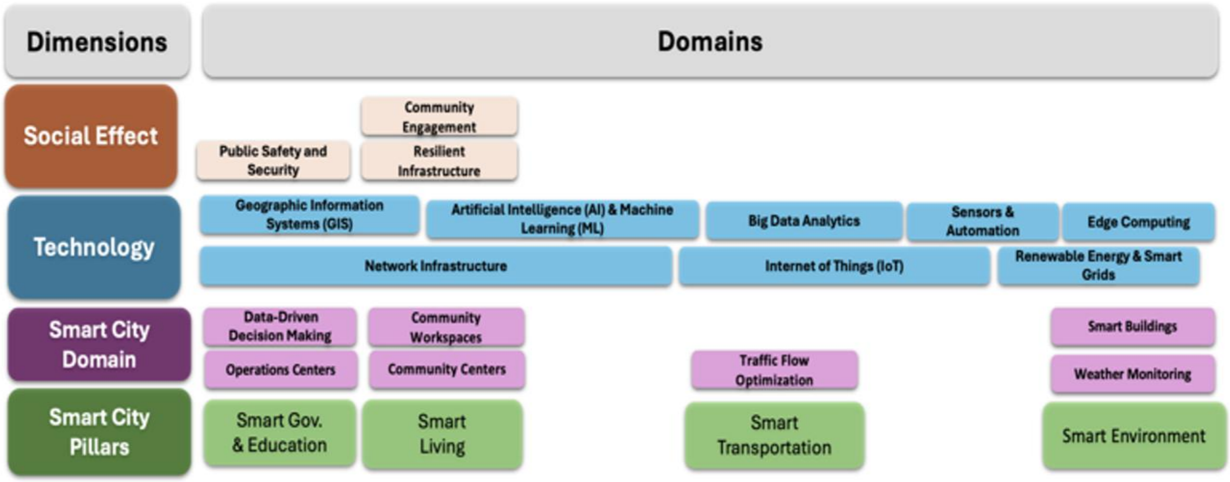

Figure 2. Proposed Disaster Detection Smart City Framework.

The smart city framework in Figure 1 is an adaptive model that 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.

Structured around six central pillars, the framework aligns targeted domains such as smart homes, traffic optimization, and waste management with data-driven decisions and robust infrastructure. It highlights cutting-edge technologies, including IoT, AI, blockchain, and renewable energy, ensuring seamless operation through a strong network infrastructure.

By linking domains and technologies to tangible outcomes, the framework offers a focused, efficient, and socially impactful roadmap for smart city initiatives. it is important to address data security, particularly in data transmission. Ensuring the authenticity and integrity of transmitted data can prevent cyber threats, misinformation, and unauthorized access. Implementing encryption, secure communication protocols, and blockchain-based verification could strengthen the system against tampering or data manipulation, enhancing trust and reliability in disaster response operations.

V. PRPOSESD DISASTER DETECTION SMART CITY FRAMEWORK

A Disaster Detection Framework (DDF) for smart cities, as shown in Figure 2, integrates advanced technologies and smart city components to enhance fire prevention, detection, and response while promoting sustainability. IoT sensors, Geographic Information System (GIS), and weather monitoring systems are deployed to track fire risks and detect incidents in real time, while AI and edge computing analyze data for early detection and predictive modeling. Operations centers coordinate responses, leveraging adaptive traffic systems for evacuation routes and notifying communities through automated alerts. Smart buildings and resilient infrastructure are equipped with automated safety measures, and waste management systems handle post-fire debris sustainably.

Community engagement and public safety initiatives educate citizens on fire prevention, while renewable energy sources power detection systems, ensuring sustainability. This holistic approach combines technology, smart city domains, and proactive strategies to minimize fire risks and enhance safety in urban environments.

VI. AI & ML ROLE IN DDF

The advancements of AI and ML have made urban centers smarter and more self-sufficient [15]. However, the sustainable development of smart cities is still an ongoing challenge, especially in disaster-prone areas. These technologies can monitor and identify a disaster, as well as estimate the resources needed to handle it.

AI and ML offer data analysis tools that can enhance performance beyond traditional methods, fostering innovation in diverse fields. Natural and man-made disasters significantly affect societal development, underscoring the importance of early detection to reduce risks, economic losses, and casualties [16].

Despite great interest in using artificial intelligence and machine learning for disaster detection, several obstacles must be overcome to ensure successful implementation. Primarily, the quality and availability of data pose significant challenges. While many cities provide access to public data, such datasets are often not collected with the same parameters or standards, limiting their comparability. Moreover, the implementation of smart city technologies is frequently hindered by privacy concerns, particularly regarding the use of personal data.

In modern smart cities, disaster detection systems are crucial for safety. Urbanization has increased flood and fire vulnerabilities. The object detection technology has evolved significantly in recent years, driven by its successful applications in various domains. The development of these detectors follows a “model zoo” approach, where different

 ISBN: 978-1-68558-251-7

models trained using varying methodologies are made publicly available [17].

The YOLO series is one of the most well-known object detector families. In particular, YOLO version 8 (YOLOv8) is a complete object detection and instance segmentation model that overtakes its predecessors [18]. It uses the framework for model implementation, training, evaluation, and inference. The YOLO family of models has played a pioneering role in advancing real-time object detection, owing to their unique architecture that integrates model training and inference on a single neural network. YOLOv8 excels as a real-time object detection model, quickly identifying and classifying objects within diverse classes in images and videos. It utilizes a single convolutional neural network to simultaneously predict bounding boxes, class probabilities, and object counts for detected classes [19]. YOLOv8 architecture consists of five key stages: image preprocessing and augmentation, backbone, neck, detector, and postprocessing [20].

VII. Fire and Flood Detection in Smart Cities using YOLOv8

Fire is one of the disasters that poses a significant threat to human life in urbanized areas. This is compounded by other potential hindrances based on the infrastructure. Fire detection in cities is particularly difficult as they are typically crowded spaces, leading to obstructions in the view field of the cameras. Furthermore, flames in general spread rapidly, leading to the idea of having an early detection mechanism [21].

Based on the above, integrating the algorithm with preexisting CCTV cameras on the roads would be an efficient alternative. Currently, most detection systems rely on either thermal cameras or a combination of both thermal and visual cameras. This necessitates the need for a separate camera system installed in addition to the standard CCTV cameras on roadways. Consequently, a new detection framework that makes use of road surveillance CCTV cameras for fire detection is presented. Recent advancements in the YOLO family, namely YOLOv8, are utilized to train a model that can detect fires. There are various approaches to integrating this model, either with an already preexisting detection system based on image processing techniques or outside detection systems based on just monitoring the images.

The urban environment poses an additional challenge for fire detection since fires are anticipated to be detected at a greater distance as opposed to other environments like industrial complexes. Hence, it is critical to have cameras that can cover a wider area. Empirical results with real-world implementation to monitor and detect fire in the surroundings of a highway are provided. Detection systems of this nature are necessary, particularly in high-speed roadways, as the response time for vehicles approaching an accident is crucial. Since the detection system is based on image processing techniques, the data can be processed in real-time on the edge to ensure rapid detection and a timely response. A thorough discussion of the framework is provided, along with case studies and examples where YOLOv8 has been useful in detecting fires [22]. This aims to provide effective implementations of such technologies and inspire the future and current endeavors in this field.

Flooding is one of the most serious calamities in urban settings that arise due to sudden and massive downpour events with gradual drainage of the platform. A flood is a complex catastrophe that involves many crucial and complicated occurrences, which happen concurrently.

There are many cataclysmic events that create flooding in a city, such as storms, tsunamis, dam breaking, heavy rain, melting snow, landslides, etc. Though there are many elements accountable for a flood catastrophe, urbanization is found to be the most evident one [23].

The real-time image processing-based flood detection is designed to detect flooding swiftly using the video stream taken from the camera placed on the roadside. The framework for flood detection in smart cities is implemented using YOLOv8. Flood can be detected using infrared, visible, or depth images taken from the camera installed in public places like traffic signals, malls, parking areas, etc. The video stream from the camera is processed using YOLOv8 to detect the flood situation. When flood is detected, it generates an alert which can be sent to the control room or concerned authority. The framework can be integrated with other sensors like rainfall, water depth, temperature, humidity etc. to take precautionary measures. The framework can also be used with the GIS system to view the flood affected area on the city map. The flood detection using YOLOv8 is tested with various videos taken from real urban flood scenarios during heavy downpour. The framework is successfully able to detect the flood condition.

Fire detection can benefit from CCTV cameras, even if they are not infrared, due to their widespread installation in urban areas, reducing the need for additional infrastructure. Leveraging existing CCTV networks allows for cost-effective fire monitoring, real-time surveillance, and integration with AI-based detection systems like YOLOv8, enabling early detection and response without requiring specialized thermal imaging cameras.

VIII. IoT Role in the Detection

Raspberry, as shown in Figure 3, can serve as a powerful IoT edge device capable of running advanced AI models such as YOLOv8 to detect fire and flood incidents in real time. Equipped with camera modules and environmental sensors, the Raspberry Pi can analyze visual and sensor data locally, leveraging its processing power to identify potential hazards with high accuracy. Once a threat is detected, the device can connect to the country's Internet

Figure 3. Raspberry with Camera.

network via Wi-Fi, Ethernet, or cellular modules to transmit critical data, including alerts and images, to a centralized IoT operation center. This seamless integration enables authorities to respond swiftly to emergencies, enhancing disaster management efforts with a cost-effective and scalable solution.

Local processing with YOLO could enhance disaster detection by reducing latency and reliance on network connectivity, enabling faster responses. However, back-end processing allows for centralized analysis, resource optimization, and integration with larger datasets, making it more scalable. A hybrid approach, combining local edge processing for real-time detection with back-end verification, could improve efficiency and reliability.

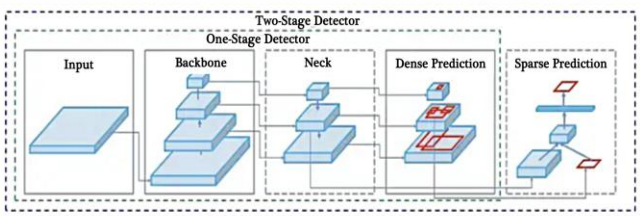

Figure 4. YOLO Model.

Figure 4 represents the architecture of object detection models. The one-stage detector processes input through a backbone for feature extraction, a neck to refine features, and a dense prediction layer to detect objects directly. The two-stage detector refines detection by using a sparse prediction layer, improving accuracy by first generating region proposals before classification.

IX. THE PROPOSED MODEL ALGORITHM

Figure 5 presents the solutions architecture for an IoT-enabled fire detection and response system, integrating UAVs and cameras. The system utilizes the YOLOv8 algorithm for real-time fire detection by analyzing visual data from UAVs and surveillance cameras. An IoT network facilitates communication, with the IoT operation center managing data processing and response coordination. Continuous model training and feedback loops enhance detection accuracy and system performance. The network infrastructure ensures reliable connectivity, enabling rapid UAV deployment for fire suppression. This architecture demonstrates the integration of AI, IoT, and UAV technologies to improve fire safety in smart city environments.

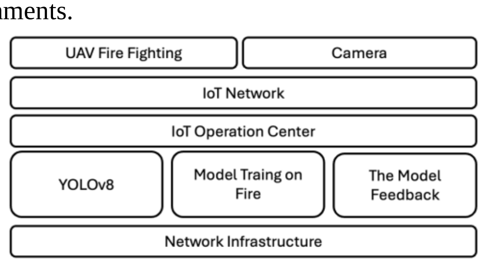

Figure 5. The Proposed Solutions Architecture.

Although YOLOv8 is used in different works, the authors only focus on one detection, either fire or flood [23] [24] [25]. Detecting fire and flood events in real time is a critical task for mitigating potential disasters and protecting people and property. This algorithm is proposed as a solution to the problem of automated fire and flood detection using the YOLO deep learning framework. By leveraging a single multi-class model, we can efficiently identify both threats within the same scene, simplifying the deployment process and reducing the computational overhead.

High-quality data is the foundation of any successful detection model. We begin by gathering a wide range of images showing fire and flood scenarios under various conditions (different lighting, angles, scales). We include some negative examples (images without fire or flood) to help the model learn what backgrounds typically look like. We use a labeling tool—such as LabelImg, Roboflow, or CVAT—to draw bounding boxes around the areas containing fire or flood. Each bounding box should be labeled with the appropriate class name: fire or flood. Once labeled, we split the data into training and validation sets, maintaining a similar distribution of classes in both sets. The directory structure typically follows YOLO's expected format, and we will need a data.yaml file that specifies paths to images, the number of classes, and their names.

By developing and training a unified multi-class YOLO model with meticulously labeled datasets encompassing both fire and flood scenarios, the proposed algorithm effectively facilitates simultaneous real-time detection of these two critical hazards. The process begins with the collection and annotation of diverse images representing fire, flood, and non-threatening environments, which are subsequently organized into training and validation subsets adhering to YOLO's standardized format. Utilizing the Ultralytics YOLO framework, the model undergoes extensive training to learn distinguishing features of each class, resulting in a robust best.pt weight file capable of identifying both fire and flood instances with high accuracy.

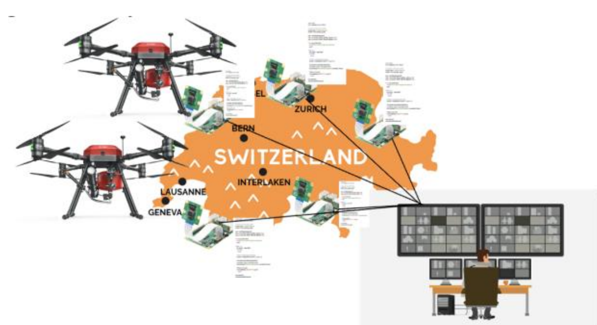

Figure 6. Overview of The Proposed Solutions.

The main Python script integrates OpenCV to capture live video streams, wherein each frame is processed by the trained YOLO model to perform detections based on predefined confidence and Intersection over Union (IoU) thresholds. Detected objects are annotated with bounding boxes and class labels directly on the video feed, enabling immediate visualization and potential activation of alert mechanisms. This streamlined approach not only enhances computational efficiency by employing a single model for dual-class detection but also ensures prompt and reliable identification of fire and flood events, thereby contributing significantly to automated disaster monitoring and mitigation systems.

Figure 6 presents the proposed solution, which integrates a Raspberry Pi running the YOLOv8 algorithm to detect fire and flood, transmitting real-time data to an IoT operation center. Additionally, UAVs are deployed to respond rapidly and assist in firefighting efforts. It also highlights the role of geographic distribution in detecting and tracking spreading disasters. By using IoT sensors and AI, the system enables real-time monitoring across urban areas, improving early detection and response efficiency to ensure that the framework is not just theoretical, but directly supports the proposed system, Table I clarifies how each framework component integrates with the disaster detection model.

TABLE I. PROPOSED FRAMEWORK COMPONENTS AND THEIR ROLES IN DISASTER DETECTION

Framework Component	Role in the Proposed Disaster Detection System
IoT Sensors & Cameras	Capture visual and environmental data (fire, smoke, water levels)
Raspberry Pi (Edge AI)	Runs the YOLOv8 model to detect disasters locally
YOLOv8 AI Model	Identifies fire and flood incidents with real-time object detection
Communication Networks	Sends detection alerts to the IoT Operations Center
IoT Operations Center	Processes incoming disaster data and triggers emergency responses
Public Alert Systems	Notifies authorities and affected populations via smart notifications

X. EXAMINATION OF THE ALGORITHM

In our experiment, we utilized the YOLOv8 model - and algorithm as in Figure 7 - with a picture from [24] (see Figure 8) to detect fire as a potential hazard within a given environment. Initially, the model was employed without incorporating fire-specific learning, resulting in the detection of various objects present in the scene, but failing to identify fire accurately, as shown in Figures 9 and 10. However, after integrating fire learning detection into the model, it successfully recognized fire occurrences with improved precision. The experiment was conducted using a standard laptop for processing and presenting the fire images situations while a mobile phone is used as a real-time camera to capture live video streams, demonstrating the feasibility of implementing fire detection in practical scenarios, as shown in Figures 11 and 12.

```
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()
```

Figure 7. The Proposed Python Script.

Figure 8. Test Picture from [24].

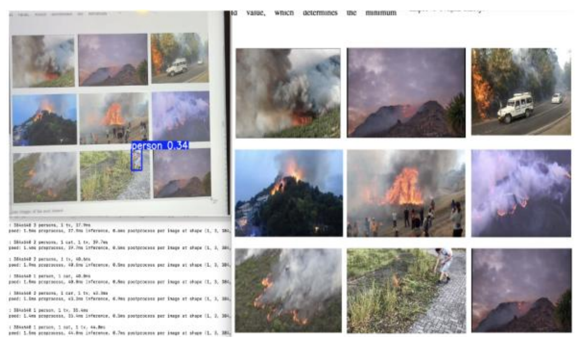

Figure 9. Screenshot from the Laptop - No Fire Detection.

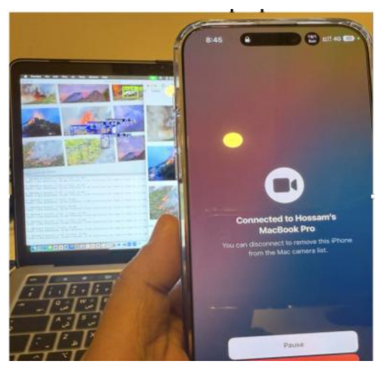

Figure 10. Experiment on the Laptop - No Fire Detection.

 ISBN: 978-1-68558-251-7

Figure 11. Screenshot from the Laptop - Fire Detection.

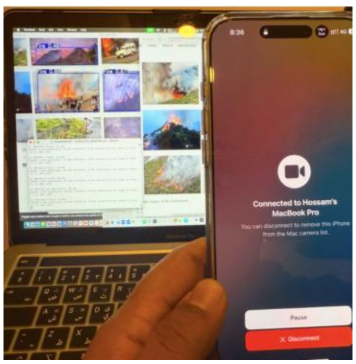

Figure 12. Experiment on the Laptop - Fire Detection.

XI. CONCLUSION AND FUTURE WORK

Smart city disaster detection is a critical topic that requires innovative solutions to enhance urban resilience and safety. In this work, we presented a comprehensive framework for disaster detection, focusing on fire and flood scenarios. We proposed a solution leveraging the YOLOv8 algorithm for real-time fire and flood detection, demonstrating its potential for effective disaster response. To validate our approach, we conducted an experiment using a laptop and a mobile phone, which successfully proved the effectiveness of machine learning in detecting fire incidents. As part of our future work, we aim to extend the validation to flood detection and conduct further tests under various flood scenarios to ensure the robustness and reliability of our solution.

In this study, we tested the fire detection capabilities of the YOLOv8 algorithm using laptop connected to mobile camera as a simulation of camera in a rural environment. The choice of a rural test setting was intentional, as wildfires are a major threat to smart cities. In many cases, early detection of wildfires in forests and suburban areas can prevent fires from spreading into urban zones, which lack sufficient open-space fire barriers.

While urban areas typically use CCTV-based fire detection, the proposed system is also designed for deployment in forests and highways where traditional fire detection is limited. Our initial tests in rural environments demonstrate the feasibility of detecting fire hazards before they reach populated areas.

Planned Urban Testing: Future work will integrate the same YOLOv8-based detection system with CCTV feeds from city surveillance cameras, allowing detection of street fires, car fires, and industrial fires in real-time.

The system is designed to complement existing fire detection methods by focusing on early wildfire detection in rural and peri-urban areas, where traditional fire alarms are not available. While urban buildings have smoke detectors and suppression systems, wildfires pose a greater risk to city outskirts, requiring AI-powered monitoring. The proposed approach leverages image-based detection using Raspberry Pi and UAVs, ensuring early intervention before fires spread to cities. Given recent events like the LA wildfires, this use case is both timely and necessary.

Flood detection is planned as future work, with efforts focused on training YOLOv8 on flood datasets and integrating IoT water level sensors for real-world validation.

REFERENCES

[1] H. M. Kim, S. Sabri, and A. Kent, “Smart cities as a platform for technological and social innovation in productivity, sustainability, and livability: A conceptual framework,” *Smart Cities Technol. Soc. Innov.*, pp. 9–28, 2021, doi: 10.1016/B978-0-12-818886-6.00002-2.

[2] A. Sharifi, A. R. Khavarian-Garmsir, and R. K. R. Kummitha, “Contributions of Smart City Solutions and Technologies to Resilience against the COVID-19 Pandemic: A Literature Review,” *Sustainability*, vol. 13, no. 14, Art. no. 14, Jan. 2021, doi: 10.3390/su13148018.

[3] R. Gaire *et al.*, “Internet of Things (IoT) and Cloud Computing Enabled Disaster Management,” in *Handbook of Integration of Cloud Computing, Cyber Physical Systems and Internet of Things*, R. Ranjan, K. Mitra, P. Prakash Jayaraman, L. Wang, and A. Y. Zomaya, Eds., Cham: Springer International Publishing, 2020, pp. 273–298. doi: 10.1007/978-3-030-43795-4_12.

[4] A. Khan, S. Gupta, and S. K. Gupta, “Emerging UAV technology for disaster detection, mitigation, response, and preparedness,” *J. Field Robot.*, vol. 39, no. 6, pp. 905–955, 2022, doi: 10.1002/rob.22075.

[5] W. He, Z. (Justin) Zhang, and W. Li, “Information technology solutions, challenges, and suggestions for tackling the COVID-19 pandemic,” *Int. J. Inf. Manag.*, vol. 57, p. 102287, Apr. 2021, doi: 10.1016/j.ijinfomgt.2020.102287.

[6] H. Jiang, M. Peng, Y. Zhong, H. Xie, Z. Hao, J. Lin, X. Ma, and X. Hu “A Survey on Deep Learning-Based Change Detection from High-Resolution Remote Sensing Images.” Accessed: Jan. 15, 2025. [Online]. Available: https://www.mdpi.com/2072-4292/14/7/1552

[7] H.-T. Nguyen, N. Ngoc, and D. Anh, “The situation of high-quality human resource in FDI enterprises in Vietnam: Exploitation and development solutions,” Jan. 2020.

[8] Z. Ren, A. S. Verma, Y. Li, J. J. E. Teuwen, and Z. Jiang, “Offshore wind turbine operations and maintenance: A state-of-the-art review,” *Renew. Sustain. Energy Rev.*, vol. 144, no. C, 2021, Accessed: Jan. 15, 2025. [Online]. Available: https://ideas.repec.org//a/eee/rensus/v144y2021ics136403212 1001805.html

[9] N. T. Nwosu, “Reducing operational costs in healthcare through advanced BI tools and data integration,” *World J. Adv. Res. Rev.*, vol. 22, no. 3, pp. 1144–1156, Jun. 2024, doi: 10.30574/wjarr.2024.22.3.1774.

[10] E. Ismagilova, L. Hughes, Y. K. Dwivedi, and K. R. Raman, “Smart cities: Advances in research—An information systems perspective,” *Int. J. Inf. Manag.*, vol. 47, pp. 88–100, 2019.

[11] E. Ismagilova, L. Hughes, N. P. Rana, and Y. K. Dwivedi, “Security, Privacy and Risks Within Smart Cities: Literature Review and Development of a Smart City Interaction Framework | Information Systems Frontiers.” Accessed: Jan. 12, 2025. [Online]. Available: https://link.springer.com/article/10.1007/s10796-020-10044-1

[12] H. M. Alshuwaikhat, Y. Adenle, and T. Almuhaidib, “A Lifecycle-Based Smart Sustainable City Strategic Framework for Realizing Smart and Sustainability Initiatives in Riyadh City.” Accessed: Jan. 13, 2025. [Online]. Available: https://www.mdpi.com/2071-1050/14/14/8240

[13] P. Bellini, P. Nesi, and G. Pantaleo, “IoT-Enabled Smart Cities: A Review of Concepts, Frameworks and Key Technologies,” *Appl. Sci.*, vol. 12, no. 3, Art. no. 3, Jan. 2022, doi: 10.3390/app12031607.

[14] H. Kamel, “Smart Scrap Management for the Smart City: Enhancing Urban Sustainability through Technology-Driven Smart Circular Economy Model Framework,” *Urban. Sustain. Soc.*, 2025, Under Review.

[15] F. Cugurullo, “Frankenstein Urbanism: Eco, Smart and Autonomous Cities, Artificial Intelligence and the End of the City,” Routledge & CRC Press. Accessed: Jan. 18, 2021. [Online]. Available: https://www.routledge.com/Frankenstein-Urbanism-Eco-Smart-and-Autonomous-Cities-Artificial-Intelligence-and-the-End-of-the-City/Cugurullo/p/book/9781138101784

[16] M. M. Kuglitsch, I. Pelivan, S. Ceola, M. Menon, and E. Xoplaki, “Facilitating adoption of AI in natural disaster management through collaboration,” *Nat. Commun.*, vol. 13, p. 1579, Mar. 2022, doi: 10.1038/s41467-022-29285-6.

[17] W. Ouyang *et al.*, “BioImage Model Zoo: A Community-Driven Resource for Accessible Deep Learning in BioImage Analysis,” Jun. 08, 2022, *bioRxiv*. doi: 10.1101/2022.06.07.495102.

[18] R. Sapkota, D. Ahmed, and M. Karkee, “Comparing YOLOv8 and Mask R-CNN for instance segmentation in complex orchard environments,” *Artif. Intell. Agric.*, vol. 13, pp. 84–99, Sep. 2024, doi: 10.1016/j.aiia.2024.07.001.

[19] P. M. Harikrishnan, A. Thomas, V. P. Gopi, P. Palanisamy, and K. A. Wahid, “Inception single shot multi-box detector with affinity propagation clustering and their application in multi-class vehicle counting,” *Appl. Intell.*, vol. 51, no. 7, pp. 4714–4729, Jul. 2021, doi: 10.1007/s10489-020-02127-y.

[20] J. Chappidi and D. M. Sundaram, “Novel Animal Detection System: Cascaded YOLOv8 With Adaptive Preprocessing and Feature Extraction,” *IEEE Access*, vol. 12, pp. 110575–110587, 2024, doi: 10.1109/ACCESS.2024.3439230.

[21] O. Zell, J. Pålsson, K. Hernandez-Diaz, F. Alonso-Fernandez, and F. Nilsson, “Image-Based Fire Detection in Industrial Environments with YOLOv4,” in *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods*, Lisbon, Portugal: SCITEPRESS - Science and Technology Publications, 2023, pp. 379–386. doi: 10.5220/0011689400003411.

[22] O. M. Khare, S. Gandhi, A. M. Rahalkar, and S. Mane, “YOLOv8-Based Visual Detection of Road Hazards: Potholes, Sewer Covers, and Manholes,” Oct. 31, 2023, *arXiv*: arXiv:2311.00073. doi: 10.48550/arXiv.2311.00073.

[23] P. R. Kumbam, and *K. M. Vejre*., “FloodLense: A Framework for ChatGPT-based Real-time Flood Detection.” Accessed: Jan. 18, 2025. [Online]. Available: https://arxiv.org/html/2401.15501v1

[24] F. M. Talaat and H. ZainEldin, “An improved fire detection approach based on YOLO-v8 for smart cities,” *Neural Comput. Appl.*, vol. 35, no. 28, pp. 20939–20954, Oct. 2023, doi: 10.1007/s00521-023-08809-1.

[25] S. Norkobil Saydirasulovich, A. Abdusalomov, M. K. Jamil, R. Nasimov, D. Kozhamzharova, and Y.-I. Cho, “A YOLOv6-Based Improved Fire Detection Approach for Smart City Environments,” *Sensors*, vol. 23, no. 6, Art. no. 6, Jan. 2023, doi: 10.3390/s23063161.