Urban Intel Platform – A Smart Response System for Emergency Services

Aditya Korde , Chinmay Gaikwad , Om Divekar , Tanvi Deshmukh

Aditya Korde, Computer & Zeal college of Engineering
Chinmay Gaikwad, Computer & Zeal college of Engineering
Om Divekar, Computer & Zeal college of Engineering
Tanvi Deshmukh , Computer & Zeal college of Engineering

---***---

Abstract - The Urban Intel Platform is a real-time emergency response and traffic reporting system designed to improve the communication between citizens and emergency service agencies.
Users can upload multimedia content, like images, videos, and text, along with incident details and location data. The inputs are processed using image-processing techniques and rule-based logic to determine the urgency level of each incident. The system automatically notifies the nearest emergency service—police, fire, or ambulance—based on the incident type and priority. Additionally, the system detects traffic hotspots by analyzing spatial data to find areas with frequent reports or congestion. A dynamic dashboard visualizes real-time incidents, response statuses, and hotspots. The Urban Intel Platform helps improve emergency response time, accuracy, and coordination, supporting the vision of safer and smarter cities.

1. INTRODUCTION

Urbanization and rapid population growth have created more challenges for city emergency services, leading to delayed responses, poor communication, and lack of verified incident evidence.

Traditional reporting methods, like phone calls, often fail to provide accurate locations or visual context, leading to inefficient decisions and resource use. The Urban Intel Platform addresses these issues with an integrated web-based system that lets citizens report emergencies by uploading images, videos, and brief descriptions. The platform uses image processing and metadata analysis to check incident details, assess urgency through rule-based prioritization, and automatically notify the nearest emergency service based on location. It also detects traffic hotspots using spatial clustering to identify and monitor accident-prone or congested zones. By combining multimedia reporting, real-time alerts, and data-driven spatial analysis, the Urban Intel Platform improves emergency response and supports proactive urban safety.

2. Functional And Non Functional Requirements Of Urban Intel Platform

2.1 Functional Features

- Incident Reporting: Citizens can upload images, videos, and incident details directly to the platform. Each submission is automatically geotagged using GPS data.
- Image Processing: Image enhancement and color segmentation help determine the nature of the incident, like smoke for fire, or debris for accidents.
- Rule-Based Priority Assessment: Incidents are classified into High, Medium, or Low priority using predefined logic based on keywords, report type, and visual patterns.
- Traffic Hotspot Detection: The platform uses clustering analysis, such as DBSCAN, on location data to highlight frequently reported or congested areas.

- Agency Notification: Alerts are automatically sent to the nearest relevant emergency service based on the incident type and location.
- Dashboard Visualization: Real-time dashboards show incidents, response statuses, and hotspot zones for monitoring and performance analysis.

2.2 Non-Functional Features

- Performance: Ensures nearly real-time alert generation and dispatch, usually within seconds.
- Scalability: Supports many users at the same time with cloud-hosted services.
- Security: Uses encrypted communication and role-based access for agencies.
- Reliability: Includes error handling, queuing, and backup systems for reliable performance.
- Usability: Designed with a responsive, accessible, and multilingual interface for both citizens and responders.

3. METHODOLOGY

The methodology of the Urban Intel Platform includes three main parts: image processing, rule-based priority assessment, and hotspot detection.

3.1 Image Processing

Uploaded images are preprocessed to improve clarity and reliability:

- Noise Reduction: Uses Gaussian filters and resizing to standard resolution.
- Color and Edge Detection: Helps identify features like fire (orange/red hues), accidents (broken structures, damaged vehicles), or smoke.
- Metadata Extraction: Automatically reads GPS and timestamp data from image files to ensure spatial accuracy.

3.2 Rule-Based Priority Assessment

Each incident is evaluated using deterministic rules instead of machine learning.

Priority levels are assigned using a weighted sum model:

Parameter Description Weight

Incident Type (Fire, Accident, Medical, Crime) w_1

Citizen Input – User-selected urgency w_2

Keywords – Text-based indicators like “injured,” “fire,” or “unconscious” w_3

Visual Indicators – Derived from image properties w_4

Parameter	Description	Weight
Incident Type	(Fire, Accident, Medical, Crime)	w_1
Citizen Input	User-selected urgency	w_2
Keywords	Text-based indicators like “injured,” “fire,” or “unconscious”	w_3
Visual Indicators	Derived from image properties	w_4

Table 1: Parameters Used in Rule-Based Priority Assessment.

Priority Score:

$$P = w_1 + w_2 + w_3 + w_4$$

High-priority cases are automatically sent to the nearest emergency department.

3.3 Traffic Hotspot Detection

To detect traffic and emergency-prone areas, the platform uses Density-Based Spatial Clustering (DBSCAN) on geolocation data.

Clusters that exceed a predefined density threshold are marked as hotspots and displayed on the dashboard.

These insights help city authorities deploy preventive measures, optimize traffic signals, or place surveillance systems.

3.4 Notification and Routing

Once the incident data is validated and prioritized:

The system finds the nearest emergency agency using Google Maps API.

A multimedia alert, with images, coordinates, and priority level, is sent through Socket.io in real time.

The dashboard updates automatically, ensuring that agencies can monitor and respond efficiently.

4. SYSTEM ARCHITECTURE AND DASHBOARD

The Urban Intel Platform is built on a three-tier architecture:

- Frontend: Developed using React.js for interactive citizen and agency interfaces.
- Backend: Built with Node.js and Express for handling APIs, rule-based logic, and notifications.
- Database: MySQL/PostGIS for incident storage and spatial queries; Firebase or AWS for media file storage.

The dashboard displays:

- Real-time map with incident markers (color-coded by urgency).
- Traffic hotspot overlays using clustering results.
- Performance metrics such as response times and frequency of incidents.

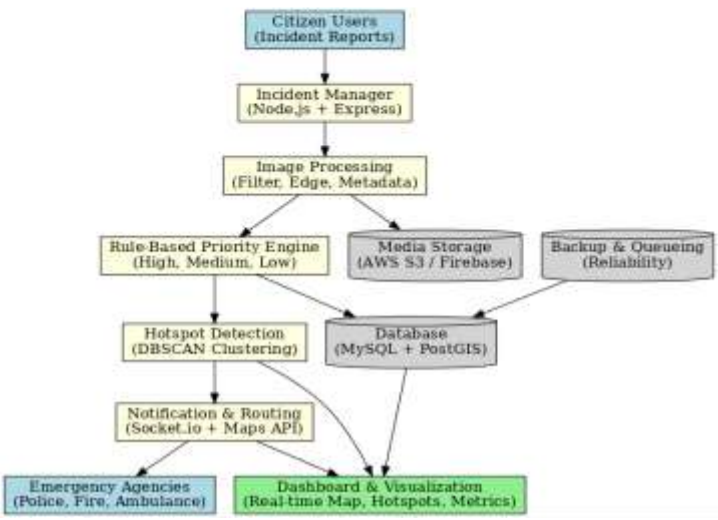

Fig. 1 System Architecture

5. CONCLUSION

The Urban Intel Platform helps bridge the gap between citizens and emergency responders through a structured, real-time reporting system.

By combining image processing, spatial clustering, and logical prioritization, the system improves the efficiency of urban emergency services. The platform also supports proactive measures by identifying high-risk traffic zones. Future developments might include integration with IoT sensors, live video streaming, and predictive hotspot analytics for smarter city management.

ACKNOWLEDGEMENT

The authors thank Prof. Dhanashri R. Londhe for her consistent guidance and valuable insights.

The team also appreciates the Department of Computer Engineering, Zeal College of Engineering & Research, Pune, for providing the infrastructure and technical support throughout this project.

REFERENCES

[1] A. D. Rosyidah, M. A. Akbar, I. Rahmat, and A. D. Rahmadi, “Emergency Response App for Enhancing Location-Based Services Using Socket.io and WebRTC,” Proc. IEEE Int. Conf. on Electrical, Electronics, and Communication Engineering (ELEXCOM), 2024, doi: 10.1109/ELEXCOM58812.2023.10370056.

[2] H. Bodkhe, C. Bilade, D. Naik, O. Deshmukh, A. Bulakh, P. Potdar, K. Shirbavikar, and S. Komble, “Track-Me-Down Emergency Location Service Provider,” Engineering Proceedings, vol. 59, p. 235, 2025, doi: 10.3390/engproc2023059235.

[3] D. Cicek and B. Kantarci, “Use of Mobile Crowdsensing in Disaster Management: A Systematic Review, Challenges, and Open Issues,” Sensors, vol. 23, no. 3, p. 1699, 2025, doi: 10.3390/s23031699.

[4] H. B. Firmansyah, V. Lorini, M. O. Mulayim, J. Gomes, and J. L. Fernandez-Marquez, “Improving Social Media Geolocation for Disaster Response by Using Text from Images and ChatGPT,” Proc. 11th Multidisciplinary Int. Social Networks Conf. (MISNC 2024), ACM, Bali, Indonesia, Aug. 2024, doi: 10.1145/3675669.3675696.

[5] J. P. J. Peixoto, J. C. N. Bittencourt, T. C. Jesus, D. G. Costa, P. Portugal, and F. Vasques, “Exploiting Geospatial Data of Connectivity and Urban Infrastructure for Efficient Positioning of Emergency Detection Units in Smart Cities,” Computers, Environment and Urban Systems, vol. 107, 102054, 2024, doi: 10.1016/j.compenvurbsys.2023.102054.

[6] L. Cao, “AI and Data Science for Smart Emergency, Crisis and Disaster Resilience,” Int. J. Data Science and Analytics, vol. 15, pp. 231–246, 2023, doi: 10.1007/s41060-023-00393-w.

[7] L. Grassi, M. Ciranni, P. Baglietto, C. T. Recchiuto, M. Maresca, and A. Sgorbissa, “Emergency Management through Information Crowdsourcing,” Information Processing and Management, vol. 60, 103386, 2023, doi: 10.1016/j.ipm.2023.103386.

[8] M. Aboualola, K. Abualsaud, T. Khattab, N. Zorba, and H. S. Hassanein, “Edge Technologies for Disaster Management: A Survey of Social Media and Artificial Intelligence Integration,” IEEE Access, vol. 11, pp. 117600–117620, 2023, doi: 10.1109/ACCESS.2023.3274839.

[9] M. J. Shehab, I. Kassem, and A. A. Kutty, “5G Networks Towards Smart and Sustainable Cities: A Review of Recent Developments, Applications and Future Perspectives,” IEEE Access, vol. 9, pp. 155314–155338, 2021, doi: 10.1109/ACCESS.2021.3128532.

[10] J. M. L. Dela Cruz, A. Feliciano, A. J. Hinay, and L. I. K. Regala, “ResQMe: An Emergency Services Locator for Manila’s Hospitals, Police, and Fire Stations Utilizing the A* Algorithm,” ResearchGate Preprint, 2020, doi: 10.13140/RG.2.2.23028.77449.