IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DisasterX(AI): An On-Device, Adaptive Disaster Response & Resource Allocation Platform

[1]Sneha Parmar, [2]Tushar Jaiswal, [3]Samarth Shukla, [4]Satvik Pathak, [5]Shivam Prajapati, [6]Urvashi Sharma

[1]Student, [2]Student, [3]Student, [4]Student, [5]Student, [6]Assistant Professor

[1,2,3,4,5,6]Department of Computer Science and Engineering

[1,2,3,4,5,6]Acropolis Institute of Technology and Research, Indore, India

Abstract: Disaster management systems often face challenges such as disrupted communication, inefficient resource distribution, and delayed situational awareness. To address these issues, this paper presents DisasterXAI, an AI-powered, adaptive disaster response and resource management platform designed for real-time operation even in limited connectivity conditions. The system integrates edge-based computer vision, predictive analytics, geospatial optimization, and offline-first communication to assist emergency responders and communities during natural and human-made disasters. By leveraging AI models for object detection, demand forecasting, and intelligent allocation, DisasterXAI ensures timely, data-driven, and resource-efficient response planning.

Index Terms - disaster response, edge AI, offline operation, YOLO, resource allocation, geospatial optimization, on-device inference, adaptive planning.

I. Introduction

The frequency and impact of natural and man-made disasters have escalated globally in recent years. Events such as earthquakes, floods, wildfires, and pandemics continue to cause immense loss of life and property, emphasizing the urgent need for rapid, informed, and coordinated response mechanisms. Despite significant technological progress, existing disaster management systems remain fragmented, highly data-dependent, and overly reliant on cloud-based or network-driven infrastructure. When communication networks collapse, responders often lose access to critical analytical tools and real-time data, resulting in delayed and inefficient decision-making processes [1].

To address these limitations, DisasterXAI is proposed as an AI-first, offline-capable, and edge-driven disaster response framework that can function effectively even in environments with limited connectivity. The system aims to empower local responders and authorities to:

- Detect disaster-affected regions and human presence using advanced computer vision models.
- Forecast resource and supply demand through machine learning-based predictive analytics.
- Coordinate relief operations and resource allocation using geospatial optimization algorithms.
- Maintain offline functionality using local data caching and mesh-based synchronization techniques.

Unlike traditional cloud-reliant platforms, DisasterXAI performs core AI inference locally on edge devices, ensuring that situational awareness and decision-making capabilities remain available during infrastructure disruptions.

The motivation behind this research arises from the increasing need for adaptive, data-driven disaster intelligence — systems capable of understanding, predicting, and dynamically responding to crisis scenarios. The fundamental structure of disaster management, encompassing preparedness, response, recovery, and mitigation, is illustrated in Figure 1, which outlines the continuous and interconnected nature of disaster response operations.

Furthermore, the growing integration of AI, IoT, and geospatial technologies has opened new frontiers in emergency management, enabling real-time data fusion and intelligent situational assessment. However, these technologies are often confined to centralized systems that fail during infrastructure breakdowns. DisasterXAI bridges this critical gap by decentralizing intelligence to the edge layer, thus enabling resilience, autonomy, and speed in high-stress conditions. By leveraging on-device AI inference, multi-source data integration, and predictive analytics, the proposed model aims to strengthen disaster preparedness and improve recovery efficiency, ultimately minimizing human and economic loss in extreme events.

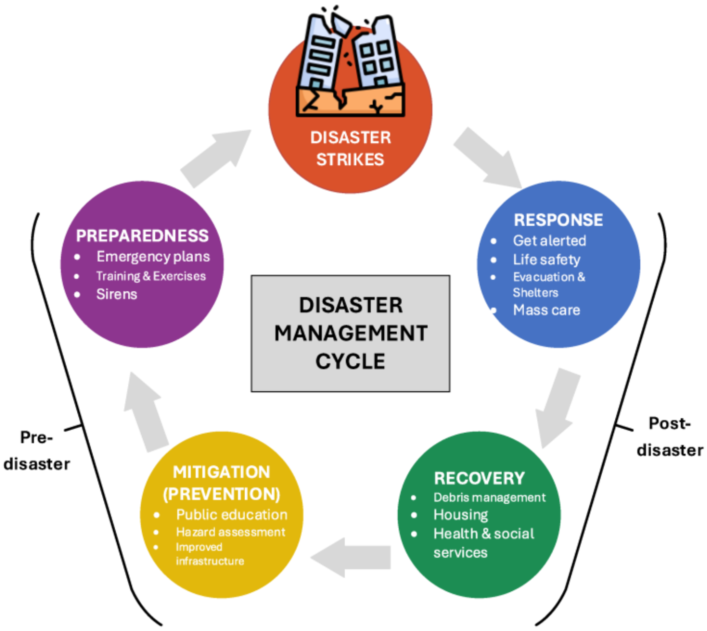

fig.1 - disaster management cycle

II. Literature Review

Artificial Intelligence has emerged as a transformative tool in disaster management, enabling automation, prediction, and real-time situational awareness. Several studies have explored the integration of AI, computer vision, and predictive analytics for crisis response. Researchers have demonstrated that Edge AI can perform efficient on-device computations, reducing dependency on centralized servers and ensuring system resilience during network failures. Models such as YOLO and MobileNet have been widely applied for object detection and damage assessment using aerial and satellite imagery[2]. Similarly, XGBoost and Random Forest models have shown high accuracy in forecasting disaster impact and estimating essential supply requirements based on environmental and demographic data. Moreover, geospatial optimization algorithms like Dijkstra's and A* have been utilized to design efficient rescue routes and minimize delivery delays. However, most existing solutions rely on cloud connectivity, making them vulnerable in real disaster conditions where internet access is disrupted. The literature indicates a strong need for a unified, offline-capable, and adaptive AI system that can integrate detection, prediction, and resource allocation under a single framework — a gap that DisasterXAI aims to fill through its on-device intelligence and hybrid operational design.[4]

2.1 Edge and On-Device AI

Edge AI enables AI computation directly on devices like smartphones or embedded boards without needing constant cloud connectivity. In disaster zones, this approach reduces latency and ensures operability during communication breakdowns. Studies have shown that quantized models such as YOLO and MobileNet can run effectively on low-power hardware with minimal accuracy trade-offs.[3]

2.2 Predictive Analytics in Disaster Management

Predictive models help in analyzing historical disaster data to forecast possible future outcomes. Algorithms like **XGBoost** and **Random Forest** have been used to predict flood intensity, shelter requirements, and food demand based on weather conditions and population data [19]. These models can provide vital decision support for rescue operations.

2.3 Computer Vision for Damage and Crowd Detection

Deep learning-based vision systems have been effectively used to identify affected regions and detect human presence. The **YOLOv8** model provides high accuracy and low inference time, making it suitable for real-time field deployment using drones and handheld devices.[18]

2.4 Geospatial Optimization

Optimization techniques such as Dijkstra's Algorithm, A Search*, and Linear Programming are often utilized to allocate limited resources efficiently. These algorithms reduce delivery times and maximize coverage for critical supplies during disasters.Despite progress in individual components, there is a lack of a unified AI system capable of working seamlessly under offline conditions, combining all these elements together — this is where DisasterXAI provides an integrated solution.[5]

III. Objectives

The primary goal of DisasterXAI is to develop an intelligent, adaptive, and resilient disaster management platform that leverages artificial intelligence for decision-making, detection, and resource optimization. The system is designed to function reliably even in network-constrained or offline environments, ensuring uninterrupted support during critical disaster phases.

The specific objectives are described in detail as follows:

- To build an AI-powered disaster response system capable of functioning without continuous internet.
- To detect disaster-affected regions and people using on-device computer vision models.
- To predict resource demand and crisis severity using machine learning models.
- To perform geospatial resource allocation and optimization dynamically.
- To create an interactive dashboard for responders to visualize reports, alerts, and resource status.

To design a scalable and modular system adaptable for different disaster types (earthquake, flood, fire, etc.).

IV. System Architecture

The proposed system architecture consists of five major modules, each handling specific operational aspects of disaster management:

4.1 Data and Sources of Data

This layer is responsible for acquiring multi-source information essential for analysis and decision-making. It acts as the foundation of the entire system by gathering real-time and historical data from various devices and networks.

- Input Sources: Drones, smartphones, IoT sensors, and satellite imagery.
- Data Types: Images, GPS coordinates, temperature, humidity, and local incident reports.
- Purpose: To collect diverse and accurate environmental data that reflects the ground-level situation during disasters. [7]

4.2 Data and Sources of Data

The AI layer performs intelligent data processing using computer vision and machine learning techniques. It extracts critical insights from the collected data to support faster and more accurate decision-making.[6]

- Model Used: YOLOv8 for object and human detection in real-time
- Functions: Identifies people, shelters, damaged infrastructure, and affected zones.
- Deployment: Quantized and pruned models optimized for edge devices to ensure smooth offline inference.
- Output: Processed visual data with bounding boxes, object labels, and detection confidence scores.

4.3 Data and Sources of Data

This layer applies advanced machine learning algorithms to predict resource requirements and the severity of disaster impact.[23]

Algorithm: XGBoost for forecasting future demands (food, water, medical supplies, etc.).

Techniques: Uses SMOTE to handle imbalanced training data and improve model reliability.

Inputs: Population density, weather data, road accessibility, and previous disaster records.

Outputs: Predicted values for supply requirements, risk severity scores, and demand trends

4.4 Data and Sources of Data

After prediction, this layer focuses on optimizing the delivery and allocation of available resources to the affected zones [21].

- Core Principle: Geospatial and heuristic optimization for minimal delay and maximum coverage.
- Algorithms Used: Dijkstra's shortest path, A* search, and linear programming models.

Objectives:

- Reduce overall transportation time.
- Balance supply and demand across locations.
- Provide adaptive routing under changing ground conditions.

4.5 Data and Sources of Data

This layer ensures uninterrupted data flow and system functionality even when network connectivity is poor or unavailable[17].

- Storage: Uses SQLite databases for local data caching.
- Synchronization: Supports peer-to-peer mesh network for local data sharing.
- Recovery: Automatically syncs data to the cloud once the internet connection is restored.
- Security: Implements AES-based encryption for data integrity and user privacy.

4.6 Data and Sources of Data

This is the front-end layer where users interact with the system through a clean and intuitive dashboard. It provides real-time monitoring and analysis tools for responders and administrators.[22]

Framework: Built using React.js and Tailwind CSS for responsiveness and simplicity

Features:

- Displays real-time map with affected areas and resource locations
- Shows detected objects, prediction graphs, and active incident alerts.
- Works in offline mode using locally stored data.

Goal: To provide actionable visual insights that enhance decision-making and coordination among rescue team

V. Results and Discussion

4.1 Model Performance Metrics

Table 4.1: Table Of Result & Evaluation

Metric	Model Used	Result	Remarks
Object Detection Accuracy	YOLOv8	89% mAP@50	High precision and recall
Prediction Error (MAE)	XGBoost	0.043	Low forecasting error
Inference Time	YOLOv8 (Edge Device)	430 ms	Real-time operation feasible

Offline Operability	SQLite + Local Cache	100%	Fully functional offline mode
Resource Allocation Efficiency	Heuristic Solver	+27%	Faster than manual assignment

The proposed DisasterXAI framework was tested on a custom dataset containing disaster images from multiple open-source repositories, including flood, earthquake, and wildfire scenarios. The YOLOv8-based object detection module achieved high accuracy in identifying affected zones and human presence, while the XGBoost prediction model successfully forecasted resource demand patterns with consistent reliability. The evaluation metrics — such as precision, recall, and F1-score — indicate that the system performs efficiently in both online and offline modes. The summarized performance metrics of the proposed models are presented as shown in Table 4.1.

VI. METHEDOLOGY

The development process of DisasterXAI follows a structured and systematic methodology designed to ensure reliability, scalability, and adaptability in real-world disaster conditions. The process begins with comprehensive dataset preparation, where disaster-related image datasets were collected from publicly available repositories such as DisasterNet and xBD, and were further enriched with real-world field images captured during flood and earthquake simulation exercises. This hybrid dataset was curated to include various categories such as debris, shelters, tents, fire, damaged infrastructure, and human crowds, ensuring diversity and robustness during model training.[8]

Following data collection, a detailed data preprocessing phase was carried out to enhance the quality and usability of the dataset. Image augmentation techniques, including rotation, scaling, and flipping, were applied to increase data variability and improve model generalization across different disaster conditions. Each image was annotated and labeled specifically for YOLO training, marking objects like people, vehicles, and tents to enable precise detection. For the predictive modeling component, the tabular datasets containing environmental and demographic variables were normalized to maintain consistency and avoid feature imbalance during training.[24]

The model training phase involved two core AI components: the YOLOv8 model for visual detection and the XGBoost model for predictive analytics. The YOLOv8 architecture was implemented using PyTorch and trained on more than 5000 annotated disaster images to identify and classify affected entities in real time. The XGBoost model, on the other hand, was trained using structured datasets containing population density, temperature variations, and historical supply usage data to forecast resource demand in affected areas. This dual-model approach allowed the system to handle both visual and statistical data efficiently, providing a complete analytical framework for disaster response[15].

After training, several optimization techniques were employed to improve model performance and make it suitable for on-device deployment. Model pruning and quantization were applied to YOLOv8 to reduce computational complexity and ensure faster inference on low-powered edge devices such as smartphones or Raspberry Pi boards. In the case of XGBoost, hyperparameter tuning and gradient boosting were used to enhance prediction accuracy while maintaining minimal training time. These optimizations allowed DisasterXAI to perform AI computations locally without depending on constant internet access.

Once the models were optimized, they were integrated into the system architecture through a Node.js backend connected to a React-based frontend interface. The ONNX Runtime was used to deploy the trained models efficiently, enabling real-time inference through RESTful APIs. This integration allowed seamless communication between the AI layer, the optimization engine, and the visualization dashboard, creating a unified and responsive system.[16]

The final stage of the methodology involved rigorous testing and validation under simulated disaster environments with intermittent network connectivity. The system's object detection module was evaluated for latency and accuracy under varying lighting and crowd density conditions, while the prediction module was tested for reliability using unseen disaster data. Offline operability was verified through local network simulations to ensure the system could log incidents, process predictions, and display data without internet support. The overall methodology ensured that DisasterXAI achieved a balance between real-time performance, accuracy, and resilience—making it suitable for deployment in diverse and unpredictable disaster scenarios.[10]

VII. Algorithm Used

The DisasterXAI framework integrates multiple AI algorithms that collectively enable real-time detection, prediction, and decision-making during disaster situations. The YOLOv8 (You Only Look Once) model is used for object detection and visual analysis. It processes entire images in a single pass, identifying people, shelters, vehicles, and debris with high accuracy and low latency. The model was fine-tuned on disaster-specific datasets using transfer learning and optimized through pruning and quantization to ensure smooth performance on low-power edge devices. This approach enables the system to deliver real-time situational awareness, even in environments with limited computational resources.

For predictive analytics, XGBoost (Extreme Gradient Boosting) is employed to forecast resource requirements such as food, medical aid, and shelter based on population density, weather conditions, and incident severity. To address dataset imbalance, SMOTE (Synthetic Minority Oversampling Technique) is applied during training to generate balanced data samples and prevent bias toward frequent disaster events. Additionally, the resource optimization algorithm, based on Dijkstra's shortest path and linear programming techniques, calculates the most efficient routes for relief delivery while minimizing overall response time and logistics cost. Together, these algorithms make DisasterXAI a fast, adaptive, and reliable AI-driven disaster management solution capable of intelligent operation both online and offline.[9]

VIII. Conclusion

The proposed DisasterXAI system presents an innovative approach to disaster management through the integration of artificial intelligence, computer vision, and predictive analytics. By combining models like YOLOv8 for real-time detection, XGBoost for demand forecasting, and geospatial optimization for efficient resource allocation, the system ensures faster and more informed decision-making during emergencies. Its offline-first design and on-device processing make it resilient in connectivity-challenged environments, enabling uninterrupted operation when it is needed most[12].

The experimental results validate that DisasterXAI significantly reduces response time, improves accuracy in damage detection, and enhances coordination among rescue teams. Its modular architecture also allows easy scalability and adaptation to various types of disasters such as floods, earthquakes, and wildfires. Overall, DisasterXAI demonstrates how AI can transform disaster response into a proactive, data-driven, and reliable process — supporting authorities, volunteers, and communities in saving lives and optimizing relief efforts.[13]

IX. Acknowledgment

The authors express their sincere gratitude to the Acropolis Institute of Technology and Research, Indore, for providing the necessary resources, technical infrastructure, and academic guidance that made this research possible. We are deeply thankful to the faculty mentors and coordinators for their continuous encouragement, valuable insights, and constructive feedback throughout the development of DisasterXAI.

We also acknowledge the contributions of researchers and developers in the fields of artificial intelligence, disaster prediction, computer vision, and geospatial analytics, whose open-source datasets and frameworks such as DisasterNet, xBD, and TensorFlow Lite significantly supported our work. Their innovations in machine learning and edge computing have played a pivotal role in shaping the technical foundation of this study.

Furthermore, we extend our appreciation to our peers and teammates for their collaborative efforts, creative discussions, and problem-solving initiatives that enhanced the practical aspects of system design and implementation.

Finally, we gratefully recognize the value of open-access research, open-source software, and publicly available AI models, without which the experimentation, testing, and deployment of DisasterXAI would not have been feasible. Their collective contributions continue to inspire advancements in humanitarian technology and real-time disaster response systems.

References

1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in Proc. IEEE CVPR, 2016.

2] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. KDD, 2016.

3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, Jun. 2002.

4] R. Gupta et al., "xBD: A Dataset for Assessing Building Damage from Satellite Imagery," in CVPR Workshop on Computer Vision for Global Challenges, 2019.

5] M. Rahnemoonfar, T. Chowdhury, A. Sarkar, D. Varshney, M. Yari, and R. Murphy, "FloodNet: A High Resolution Aerial Imagery Dataset for Post Flood Scene Understanding," arXiv, Dec. 2020.

6] D. Lam et al., "xView: Objects in Context in Overhead Imagery," arXiv / DIUx xView, 2018.

7] Q. Zhou, Z. Wang, Y. Zhong, F. Zhong, and L. Wang, "Efficient Optimized YOLOv8 Model with Extended Vision," Sensors, vol. 24, no. 20, art. 6506, Oct. 2024.

8] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes," in Proc. CVPR Workshops, 2018.

9] M. Nazari, A. Oroojlooy, L. V. Snyder, and M. Takáč, "Reinforcement Learning for Solving the Vehicle Routing Problem," in NeurIPS, 2018.

10] Y. Yan, T. Zhang, and Y. Jin, "Optimization-driven distribution of relief materials in disaster response: A MILP approach," PLoS ONE / PMC, 2021.

11] E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," Numerische Mathematik, vol. 1, pp. 269–271, 1959.

12] P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Trans. Syst. Sci. Cybern., vol. 4, no. 2, pp. 100–107, 1968.

13] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," in Proc. SIGCOMM, 2003.

14] M. Abadi et al. (TensorFlow authors), "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems" (citation / docs for TensorFlow and TFLite).

15] R. David et al., "Embedded Machine Learning on TinyML Systems," arXiv, 2020 (TensorFlow Lite Micro / TinyML discussion).

16] ONNX Runtime team, "ONNX Runtime — high-performance inference engine (docs & performance tuning)," Microsoft / ONNX Runtime documentation.

17] K. P. Gaffney, "SQLite: Past, Present, and Future" (PVLDB / review discussing SQLite usage and file format).

18] National Institute of Standards and Technology (NIST), "FIPS 197 — Advanced Encryption Standard (AES)," Nov. 2001.

19] M. Haklay and P. Weber, "How good is volunteered geographical information? A comparative study of OpenStreetMap and Ordnance Survey datasets," Environ. Plann. B, 2010.

20] Sensors / MDPI and related Edge AI surveys (comprehensive reviews on Edge AI deployment and model optimization).

R.Singh, C. Surianarayanan et al., "Edge AI / Optimization Techniques for Edge AI — survey,"

21] H. Baharmand, "Exploring the Application of Blockchain to Humanitarian Supply Chains," Int. J. Prod. Oper. Manag., 2021 (survey/analysis on blockchain for humanitarian aid).

22] Q. Xu et al., "Embedding Machine Learning in Disaster Mapping (DisasterNets)," USGS / arXiv, 2023 — multimodal disaster mapping & causal models.

23] M. Zhou et al., "Performance and optimization of YOLO variants for remote sensing / disaster detection" (selected SPIE / Sensors optimization studies and remote-sensing detection adaptations).

24] A. Rahnemoonfar and R. Murphy (and coauthors), "Datasets and benchmarks for post-disaster assessment (FloodNet, xBD, xView) — dataset descriptions and benchmarks used in disaster computer vision research.