AI-Based Emergency Evacuation System

Mubashir Afzal 231AI021 Aashi Kumari 231AI002 Hasini Jaishetty 231AI012 Prajwal Meshram 231AI019
Department of Information Technology
National Institute of Technology Karnataka

***Abstract*—This paper presents a comprehensive AI-based emergency evacuation system that combines deep learning-based detection and intelligent path planning. The system uses an integrated YOLO model to simultaneously detect fire and crowd congestion from video feeds. It employs ZeroMQ for real-time networking between the server and clients. The server computes an optimal evacuation route based on a weighted function considering fire intensity, crowd density, and distance, and sends it to clients for visualization using Pygame. Clients continuously update their location to the server, enabling adaptive rerouting. This scalable, modular approach enhances safety and efficiency in emergency scenarios.**

***Index Terms*—Emergency Evacuation, YOLO, Fire Detection, Crowd Detection, ZeroMQ, Pygame, Path Planning, AI, Networking**

I. Introduction

In emergency situations such as fires in crowded indoor spaces, swift and informed evacuation is critical to saving lives. Traditional systems often lack real-time adaptability and intelligence. To address this, we propose an AI-powered solution that integrates fire and crowd detection using a YOLO model, real-time path computation, and dynamic client-server communication using ZeroMQ. Our system not only detects hazards but also computes the safest evacuation route and updates it based on live conditions.

II. System Architecture

The system is divided into the following components:

- **YOLO-Based Fire and Crowd Detection:** A modified YOLO model detects both hazards from a single frame.
- **ZeroMQ Communication:** Bi-directional messaging between server and multiple clients.
- **Evacuation Path Computation:** Based on custom weights using fire intensity, crowd density, and distance.
- **Pygame Visualization:** Graphical display of environment, hazards, and optimal path on the client side.

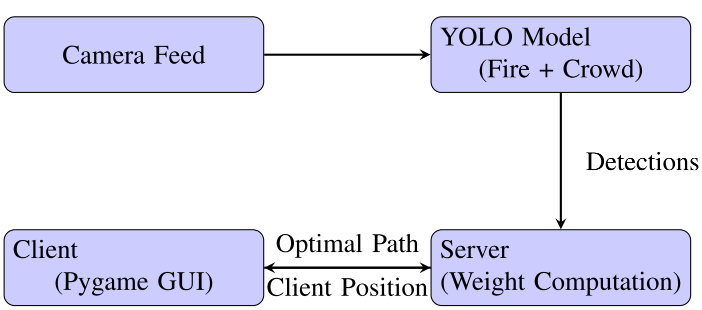

Fig. 1. Block Diagram of System Architecture

III. Integrated Detection Model

We employ a custom YOLOv5 model trained to detect both fire and crowd. This eliminates the need for separate models, reducing computation and latency. The dataset comprises labeled images for both fire (flames, smoke) and crowd (people clusters). The output includes bounding boxes with classes and confidence scores, transmitted to the server for route planning.

IV. Networking using ZeroMQ

ZeroMQ enables fast, asynchronous communication. The server uses PUB-SUB and REQ-REP patterns to:

- Receive fire/crowd coordinates from the detection node.
- Send optimal paths to clients.
- Receive client position updates.

This architecture supports multiple clients in parallel.

V. Path Planning Algorithm

We use a modified A* algorithm where the edge cost w is computed as:

$$w = \alpha \cdot I_{fire} + \beta \cdot D_{crowd} + \gamma \cdot d_{distance} \tag{1}$$

Where:

- I_{fire} = fire intensity (normalized)
- D_{crowd} = crowd density (estimated from YOLO bounding boxes)
- $d_{distance}$ = physical distance between nodes
- α, β, γ = tunable weights based on importance

This cost function helps avoid crowded and fire-prone areas.

VI. Visualization with Pygame

The client GUI is implemented using Pygame and shows:

- A grid-based map of the environment
- Detected fire and crowd positions
- Current location of the client
- Safest computed path to exit

Clients update their location periodically and reroute if the environment changes.

VII. Conclusion

We present an intelligent, robust AI-based evacuation system capable of real-time hazard detection and adaptive path planning. The integration of deep learning, networking, and visualization makes it suitable for real-world deployment in public buildings, offices, or campuses. Future work will focus on incorporating IoT sensors and extending it to multi-floor buildings.

ACKNOWLEDGMENTS

We thank the Department of Information Technology, NITK Surathkal, for their support. Special thanks to our mentors and peers who provided feedback during the development phase.

REFERENCES

[1] J. Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection,” CVPR, 2016.
[2] P. Hintjens, ”ZeroMQ: Messaging for Many Applications”, O’Reilly Media, 2013.
[3] P. Hart, N. Nilsson, and B. Raphael, “A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” IEEE Transactions on Systems Science and Cybernetics, 1968.