Multimodal RAG Pipeline for GAT-Based Fire and Accident Detection

Minor project report submitted in partial fulfilment of the

requirements for the degree of

Bachelor of Technology

in

Computer Science & Engineering

by

Peddi Sai Preetham (2022UCP1687)
Meghavath Rahul Srinivas Sai Nayak (2022UCP1464)

under the supervision of

Dr. Mahipal Jadeja

December, 2025

Department of Computer Science & Engineering
Malaviya National Institute of Technology Jaipur,
Jaipur, Rajasthan, India

Department of Computer Science & Engineering
Malaviya National Institute of Technology Jaipur

Supervisor's Certificate

This is to certify that the project report titled **Multimodal RAG Pipeline for GAT-Based Fire and Accident Detection** is being submitted by **Peddi Sai Preetham (2022UCP1687), and Meghavath Rahul Srinivas Sai Nayak (2022UCP1464)** , is a record of original research carried out by them under my supervision and guidance in partial fulfillment of the requirements of the degree of Bachelor of Technology in Computer Science & Engineering. Neither this report nor any part of it has been submitted earlier for any degree or diploma to any institute or university in India or abroad.

Dr. Mahipal Jadeja

Department of Computer Science & Engineering,
Malaviya National Institute of Technology Jaipur,
JLN Marg, Jaipur, Rajasthan 302017, INDIA.

Acknowledgement

The team wishes to thank Dr. Mahipal Jadeja for continuous support and guidance throughout the development of this project. We acknowledge MNIT Jaipur for computational facilities, and the authors of the open-source frameworks (PyTorch, Ultralytics YOLO, Torch-Geometric, Sentence-Transformers, FAISS[1]) that formed the backbone of our implementation. We also thank contributors of publicly available fire and accident datasets used in training our models.

Place: MNIT, Jaipur

Date : December 4, 2025

Signature of students

(2022UCP1687)

(2022UCP1464)

Declaration of Originality

We hereby declare that this work entitled **Multimodal RAG Pipeline for GAT-Based Fire and Accident Detection** presents our original work carried out as a under graduate student of MNIT Jaipur and, to the best of our knowledge, contains no material previously published or written by another person, nor any material presented by us for the award of any degree or diploma of MNIT Jaipur or any other institution. Any relevant material taken from the the work of others (whether published or unpublished) has been properly acknowledged and referenced in accordance with the institute's guidelines.

Place: MNIT, Jaipur

Date : December 4, 2025

Signature of students

(2022UCP1687)

____

(2022UCP1464)

____

Abstract

This project presents an end-to-end, multimodal Retrieval-Augmented Generation (RAG) pipeline for hazard detection and response. The system detects critical events such as *fire* and *road accidents* using a hybrid vision–graph architecture. Our approach integrates YOLOv8[2] for localized detection, a custom global encoder(ResNet18), and a Graph Attention Network (GAT)-based relational model. FAISS[1] retrieval of relevant safety guidelines paired with SBERT embeddings enables task-specific advisory generation through an LLM. Finally, the pipeline optionally triggers email/SMS alerts and local TTS announcements.

The motivation for this system lies in automating early detection and structured safety responses for emergency situations. Unlike plain image classifiers, our method models relationships between scene objects and global surroundings using a graph-based architecture, enabling richer reasoning about incidents. Experiments show strong performance in fire and accident detection using our hybrid GNN trained on curated datasets.

Keywords: *Multimodal RAG*; *GAT*; *Fire Detection*; *Accident Detection*; *FAISS[1] Retrieval*

Contents

List of Figures

List of Tables

1 Introduction

This project develops a unified multimodal pipeline that classifies incoming image frames into three categories — *normal* [3], *fire* [4], and *accident* [5] — and, when required, generates safety advisories based on retrieved domain documents. The system is designed to work in near-real-time, running efficiently even on CPU, while leveraging offline or cloud-based LLMs for advisory generation.

1.1 Problem Statement

Traditional image classifiers treat each frame as independent and often fail to understand contextual relationships between detected objects. For example, a small flame may be harmless in a kitchen but dangerous near a fuel container. Similarly, a damaged vehicle alone does not always indicate an accident unless combined with human presence or object spatial relationships. This motivates a relational modeling approach, which our GAT[6]-based hybrid system enables.

1.2 Motivation and Objectives

Our key objectives are:

1. Build a hybrid graph-based hazard classifier combining global scene context and local object features.
2. Train a three-class classifier (*normal*,*fire*, *accident*) using a curated dataset.
3. Construct a complete RAG pipeline that retrieves relevant safety instructions using FAISS[1] and generates advisory text using an LLM.
4. Integrate optional alerting modules (SMTP, SMS) into a deployable pipeline.

1.3 Scope and Limitations

Our system strictly detects three classes: normal, fire, and accident. It does not include flood, smoke-only alerts, or medical event detection since no training data for these was used. The LLM output is guided solely by retrieved documents to minimize hallucinations.

1.4 Contributions

- A fully implemented hybrid GNN hazard classifier integrating YOLOv8[2] detections with global ResNet embeddings.
- A robust graph builder generating per-frame fully connected graphs with meaningful node features.
- A practical FAISS[1]-based retrieval engine with SBERT[7] embeddings for guideline enrichment.
- A complete safety advisory pipeline including LLM generation, TTS, and alerting.

2 Literature Survey

2.1 Object Detection

YOLOv8[2] provides accurate real-time object detection and is used here without additional training. It supplies bounding boxes, class IDs, and confidence scores forming the local features for graph construction.

2.2 Scene-Level Feature Extraction

ResNet18, pretrained on ImageNet, is used to extract global image representations. The final fully-connected layer is removed, producing a 512-dimensional vector later compressed to 128 dimensions.

2.3 Graph Neural Networks

Graph Attention Networks (GAT) dynamically weight neighbor importance using attention coefficients. Since hazards often depend on relational structure (fire near objects, accident with multiple entities), GATs are suitable for reasoning over object-object interactions.

2.4 Retrieval-Augmented Generation

RAG pipelines enhance LLM outputs by grounding them in actual retrieved documents, helping reduce hallucinations. FAISS[1] is a fast vector search tool used to index safety guideline corpora.

3 Methodology

3.1 Complete Pipeline

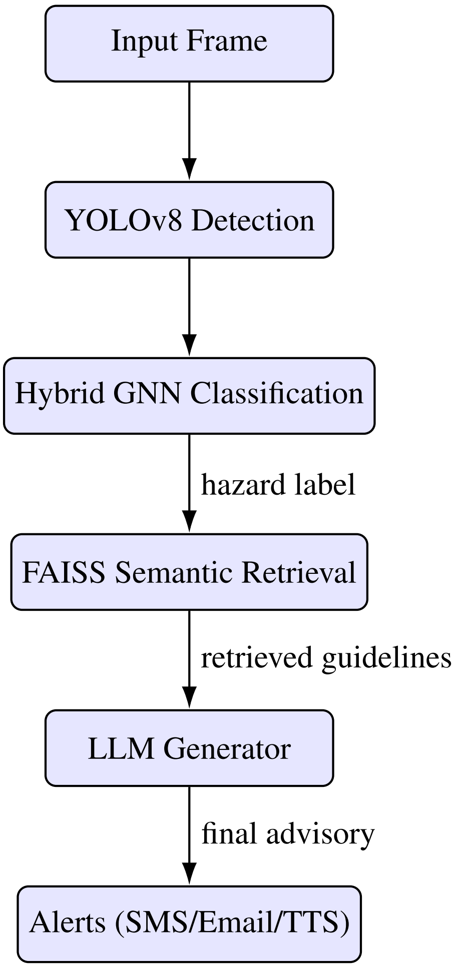


Figure 1: Complete RAG pipeline used for hazard detection and advisory generation.

3.2 System Overview

The system consists of the following sequential modules:

1. YOLOv8[2] object detection

2. Global scene embedding using ResNet18 [8]

3. Hybrid graph construction with global and local node features

4. GAT-based hazard classification

5. FAISS[1]-based retrieval of relevant guidelines
6. LLM-based advisory generation
7. Optional alert dispatch via SMTP/SMS and TTS announcement

3.3 Detector Module (YOLOv8[2])

YOLOv8[2]n is used in **inference-only mode**. For each image it returns:

- bounding box (x_1, y_1, x_2, y_2)
- class id
- confidence score

These form the *local features* of graph nodes.

3.4 Global Feature Encoder (ResNet18)

The image is resized to 224×224 and passed through a frozen ResNet18. The output 512-dimensional feature is compressed using a learnable linear layer into a 128-dimensional global context vector.

3.5 Graph Construction

Each detected object yields a node. Node feature vector:

$$\mathbf{x}_i = [g \mid\mid c_i \mid\mid conf_i \mid\mid cx_i \mid\mid cy_i \mid\mid area_i]$$

where:

- $g \in R^{128}$: global ResNet feature
- c_i: YOLO class id (float)
- $conf_i$: confidence score
- cx_i, cy_i: normalized center coordinates
- $area_i$: normalized bounding box area

Fully connected edges: All node pairs $i \neq j$ form directed edges. If YOLO detects no objects, a single dummy node is created.

3.6 Hybrid GAT Architecture

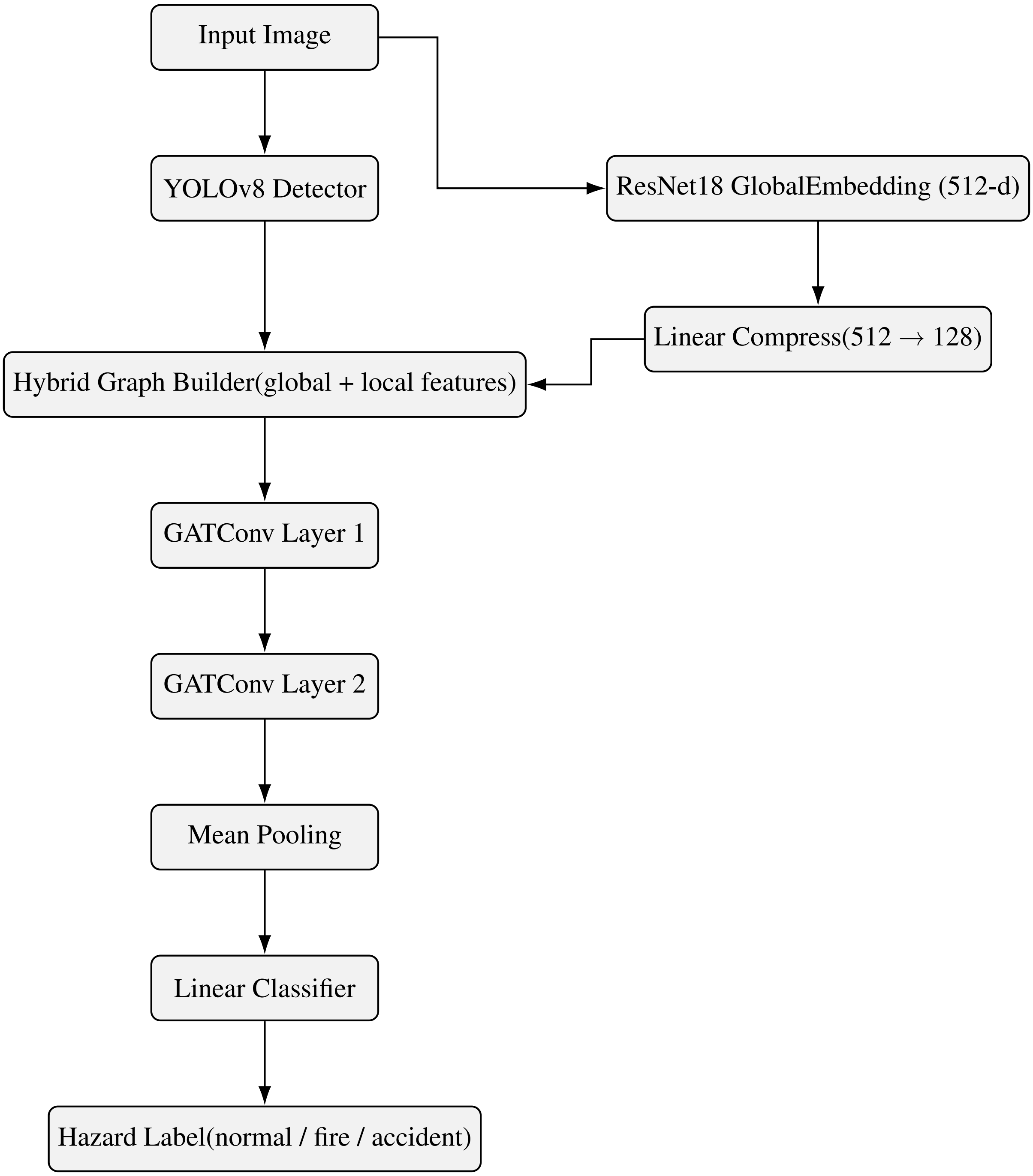

Input Image
YOLOv8 Detector
ResNet18 GlobalEmbedding (512-d)
Linear Compress(512 → 128)
Hybrid Graph Builder(global + local features)
GATConv Layer 1
GATConv Layer 2
Mean Pooling
Linear Classifier
Hazard Label(normal / fire / accident)

Figure 2: Hybrid GNN Architecture combining YOLOv8[2] local object cues with ResNet18 global context.

The model consists of:

- GATConv 1: input = 133, output = 256 × 2 heads
- GATConv 2: input = 512, output = 256
- Mean pooling over nodes
- Linear layer → 3 classes (normal, fire, accident)

The graph attention learns which objects are relevant for incident reasoning.

3.7 Training Pipeline

- Dataset organized as three folders: `normal/`, `fire/`, `accident/`
- WeightedRandomSampler used to balance classes
- Loss: Cross-Entropy Loss
- Optimizer: Adam
- LR = $1e^{-4}$
- Compress layer and GAT jointly trained
- Checkpoints store both model and compress-layer weights

3.8 FAISS Index Creation and Retrieval

Guidelines are stored as text files in a folder. Steps:

1. Encode each document using SBERT[7] (all-mpnet-base-v2)
2. Store embeddings in a FAISS[1] index (L2 or cosine)
3. Save metadata file mapping embeddings to document text
4. At inference time, query text is embedded and top-k neighbors retrieved

3.9 LLM Advisory Generation

The final prompt includes:

- hazard class detected
- safety score (confidence)
- retrieved guideline summaries

Output format:

1. one short announcement
2. 3–5 actionable instructions
3. 1 cautionary “Do not” instruction

3.10 Alerting Module

The system supports:

- SMTP email alerts
- Twilio SMS (optional)
- TTS announcements

The pipeline triggers alerts only for *fire* and *accident* predictions.

4 Experiments and Results

4.1 Datasets and Preprocessing

The complete training dataset is constructed from three independent sources corresponding to the three classes used in our Hybrid GNN classifier:

- **Fire Class:** We used the public *“Flame Dataset – Fire Classification”* released by **Smruti Sanchita Das** on Kaggle. This dataset contains labeled images of indoor and outdoor flames and served as the primary fire class for training.

- **Accident Class:** Accident images were taken from a public Kaggle collection titled *“Road Accident Images Dataset”*, containing real incident photographs of vehicle collisions, road crashes, and emergency scenes.

- **Normal Class:** A curated collection of daily-scene images not containing hazards. These were sourced from publicly available normal-scene datasets and from manually gathered images sampled from Open Images and internet sources.

For evaluation of fire vs non-fire performance, an additional dataset split was created:

- **Fire Evaluation Set:** The same Kaggle *Flame Dataset* was used, but using images reserved exclusively for evaluation.

- **Non-Fire Evaluation Set:** A curated folder containing general indoor and outdoor images with no flames, used for binary fire-vs-non-fire benchmarking.

All images were normalized to valid RGB format. YOLO inference operates directly on original-scale images, while the ResNet18 encoder uses resized 224×224 inputs. Corrupted or grayscale images were automatically filtered.

4.2 Training Details

The three-class Hybrid GNN model was trained using the datasets described earlier: the Kaggle Flame Dataset for fire images, the Road Accident Images Dataset for accident images, and a curated normal-scene dataset. The class distribution was naturally imbalanced, with fire images being fewer than accident or normal images. To address this, we used a **WeightedRandomSampler** where sample weights were set to $1/\sqrt{\text{class frequency}}$.

4.3 Evaluation

Evaluation was performed on a held-out set constructed by:

- reserving a portion of the Kaggle Flame Dataset as an unseen fire-test subset,

- using images from the Road Accident Dataset not included during training,

- using additional unseen normal images collected separately.

For fire-vs-non-fire benchmarking, an additional evaluation dataset was created with to compare with a previously proposed model:

- the Flame Dataset (fire),
- curated non-fire images (non-fire).

Standard metrics such as accuracy, precision, recall, F1-score, and confusion matrix were computed for both the 3-class model and the binary fire evaluation model.

- Accuracy
- Precision / Recall / F1 per class
- Confusion matrix

4.4 Quantitative Results

Table 1: Trained Models Comparision

Metric	With ResNet18	Without ResNet18
Accuracy	0.97	0.63
Precision	0.96	0.65
Recall	0.97	0.71
F1	0.96	0.68

- **Global context from ResNet improves discrimination** between small flames and red-colored false positives.
- **Hybrid node representation (global + local)** captures scene-level cues (e.g., smoke, lighting, occlusions) unavailable to YOLO detections alone.

4.5 Comparison with Prior Work

Yuan et al. [9] proposed a graph-embedded YOLOv5 architecture for forest fire detection. Their method integrates graph reasoning directly into the YOLO detector head, improving sensitivity to small flames and context-dependent features. They evaluated their model on the *Flame Dataset – Fire Classification*.

Our approach differs in two ways:

- **Decoupled architecture:** YOLOv8[2] performs object detection, while a separate Hybrid GAT performs relational reasoning. This modularity simplifies training, debugging, and future extension to multimodal RAG.
- **Three-class hazard reasoning:** Unlike prior fire-only systems, our Hybrid GNN performs classification across *fire*, *accident*, and *normal* scenes.

We evaluated three variants of our system:

1. **Hybrid GNN (3-class)** – uses ResNet global context + local YOLO features.
2. **Binary Fire Classifier** – same architecture but trained only on fire vs. non-fire.
3. **Earlier GNN Baseline (No ResNet)** – uses only YOLO local features for each node (class ID, confidence, center, area). No global scene embedding was used.

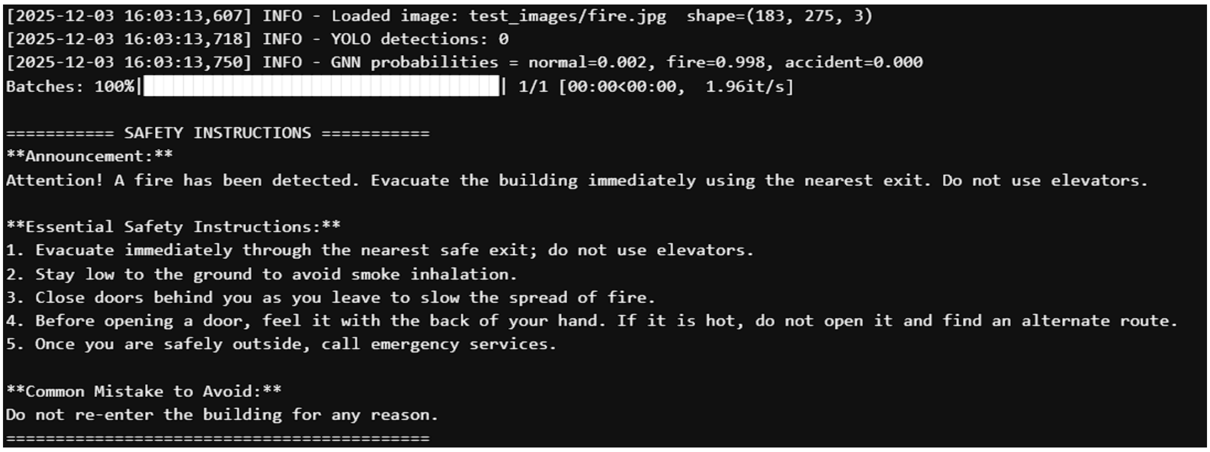
Table 2 shows a comparison of these models evaluated on the *Flame Dataset*.

Table 2: Comparison with Yuan et al. (2024) on the Flame Dataset

Model	Precision	Recall	F1
Graph-Embedded YOLOv5 (Yuan et al.)	0.92	0.85	0.88
Hybrid GNN (3-class)	1	0.005	0.01
Hybrid GNN (Fire-only, 2-class)	0.66	0.91	0.77

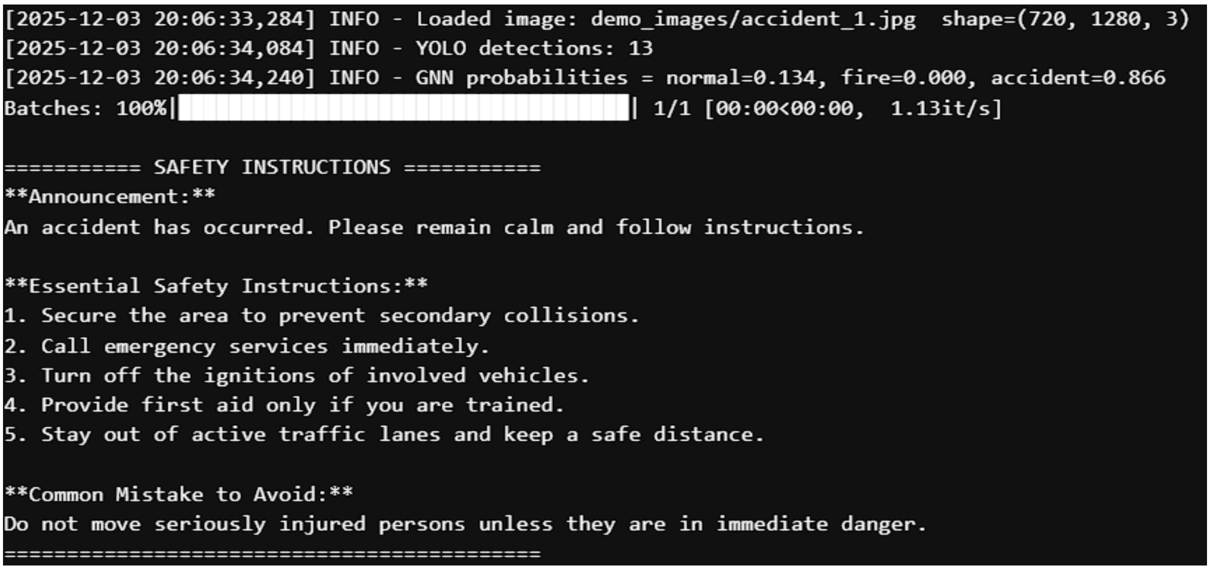
Although the specialized graph-embedded YOLOv5 model achieves the highest accuracy, our binary fire-only Hybrid GNN approaches it while retaining the flexibility to classify multiple hazard types and integrate into a full RAG pipeline.

4.6 Qualitative Results



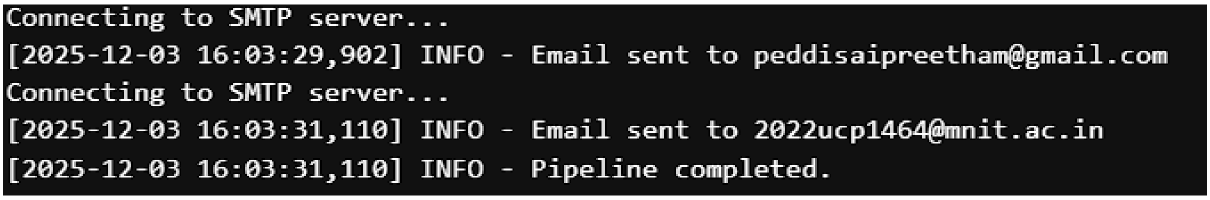
```
[2025-12-03 16:03:13,607] INFO - Loaded image: test_images/fire.jpg  shape=(183, 275, 3)
[2025-12-03 16:03:13,718] INFO - YOLO detections: 0
[2025-12-03 16:03:13,750] INFO - GNN probabilities = normal=0.002, fire=0.998, accident=0.000
Batches: 100%|████████████████████████████████████████| 1/1 [00:00<00:00,  1.96it/s]

=========== SAFETY INSTRUCTIONS ===========
**Announcement:**
Attention! A fire has been detected. Evacuate the building immediately using the nearest exit. Do not use elevators.

**Essential Safety Instructions:**
1. Evacuate immediately through the nearest safe exit; do not use elevators.
2. Stay low to the ground to avoid smoke inhalation.
3. Close doors behind you as you leave to slow the spread of fire.
4. Before opening a door, feel it with the back of your hand. If it is hot, do not open it and find an alternate route.
5. Once you are safely outside, call emergency services.

**Common Mistake to Avoid:**
Do not re-enter the building for any reason.
===========================================
```

Figure 3: Fire detected



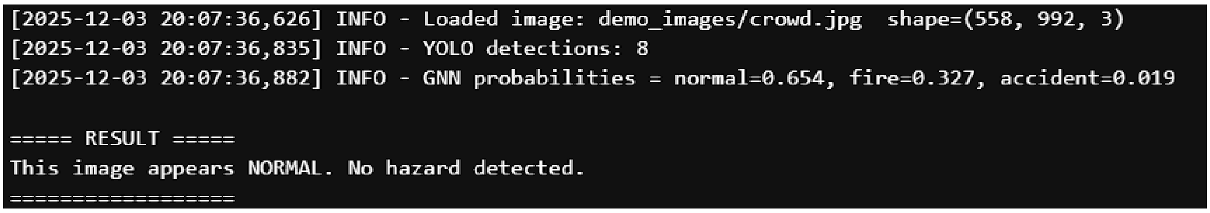
```
[2025-12-03 20:06:33,284] INFO - Loaded image: demo_images/accident_1.jpg  shape=(720, 1280, 3)
[2025-12-03 20:06:34,084] INFO - YOLO detections: 13
[2025-12-03 20:06:34,240] INFO - GNN probabilities = normal=0.134, fire=0.000, accident=0.866
Batches: 100%|████████████████████████████████████████| 1/1 [00:00<00:00,  1.13it/s]

=========== SAFETY INSTRUCTIONS ===========
**Announcement:**
An accident has occurred. Please remain calm and follow instructions.

**Essential Safety Instructions:**
1. Secure the area to prevent secondary collisions.
2. Call emergency services immediately.
3. Turn off the ignitions of involved vehicles.
4. Provide first aid only if you are trained.
5. Stay out of active traffic lanes and keep a safe distance.

**Common Mistake to Avoid:**
Do not move seriously injured persons unless they are in immediate danger.
===========================================
```

Figure 4: Accident detected



```
Connecting to SMTP server...
[2025-12-03 16:03:29,902] INFO - Email sent to peddisaipreetham@gmail.com
Connecting to SMTP server...
[2025-12-03 16:03:31,110] INFO - Email sent to 2022ucp1464@mnit.ac.in
[2025-12-03 16:03:31,110] INFO - Pipeline completed.
```

Figure 5: Alert Sent



```
[2025-12-03 20:07:36,626] INFO - Loaded image: demo_images/crowd.jpg  shape=(558, 992, 3)
[2025-12-03 20:07:36,835] INFO - YOLO detections: 8
[2025-12-03 20:07:36,882] INFO - GNN probabilities = normal=0.654, fire=0.327, accident=0.019

===== RESULT =====
This image appears NORMAL. No hazard detected.
==================
```

Figure 6: Normal image

5 Conclusion and Future Scope

This project successfully implements a multimodal hazard detection and advisory system. By leveraging YOLO detections, global scene embeddings, and a hybrid GAT architecture, the model learns relational cues important for detecting fire and accident scenarios. The RAG pipeline grounds LLM outputs in factual domain guidelines, producing reliable safety instructions. The system demonstrates strong performance and real-time capability on CPU.

5.1 Limitations

- Model trained only on three classes; cannot detect other hazards.
- YOLO misdetections can propagate errors to the GNN.
- Advisory quality depends on LLM availability and retrieved documents.

5.2 Future Work

- Incorporate temporal reasoning from live video streams instead of single-frame input.
- Deploy on distributed edge devices (drones, surveillance systems) for low-latency detection.
- Expand dataset to cover disasters of other classes(flood, earthquake, etc.)
- Fine-tuning YOLO for current domain.

References

[1] J. Johnson, M. Douze, and H. Jégou, “Billion-scale similarity search with gpus,” *IEEE Transactions on Big Data*, 2019.

[2] Ultralytics, “YOLOv8: Next-generation real-time object detection,” https://github.com/ultralytics/ultralytics, 2023.

[3] C. from multiple royalty-free datasets, “Normal scene image set (negative class),” 2023, used for non-hazard normal classification in Hybrid GNN training.

[4] S. Das, “Flame dataset – fire classification,” Kaggle, 2022, https://www.kaggle.com/datasets/smrutisanchitadas/flame-dataset-fire-classification.

[5] V. Contributors, “Car crash / accident detection dataset,” Kaggle, 2021, a curated dataset used for accident classification.

[6] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” *arXiv preprint arXiv:1710.10903*, 2017.

[7] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

[8] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[9] H. Yuan, Z. Lu, R. Zhang, J. Li, S. Wang, and J. Fan, “An effective graph embedded yolov5 model for forest fire detection,” *Computational Intelligence*, vol. 40, no. 2, p. e12640, 2024.