جامعة عفت
EFFAT UNIVERSITY

CS 4175 Senior 2 Project Report

Drone-Based AI System for Real-Time Hazard Detection and Crowd Safety During Hajj and Umrah (Amin)

Nareman Turkistani, naaturkistani@effat.edu.sa
Reem Alamoudi, Redalamoudi@effat.edu.sa
Reham Alamoudi, Rebalamoudi@effat.edu.sa

In Partial Fulfillment of the Requirements for Degree of Bachelor of Science in Computer Science, Department of Computer Science, College of Engineering, Effat University

Jeddah, Saudi Arabia

Supervisor:

Dr. Passent M. ElKafrawy

pelkafrawy@effatuniversity.edu.sa

Energy Engineering and Technology Research Center

Effat University, Jeddah, Saudi Arabia

May 19, 2025

Drone-Based AI System for Real-Time Hazard Detection and Crowd Safety During Hajj and Umrah (Amin)

Nareman Turkistani - S21107235
Reem Alamoudi - S20206949
Reham Alamoudi - S20206950

In Partial Fulfillment of the Requirements for Capstone Project

Department of Computer Science
College of Engineering
Effat University, Jeddah, Saudi Arabia

Instructor: Dr. Passent Elkafrawy
Supervisors: Dr. Passent Elkafrawy

Evaluator Name	Nareman Turkistani Grade	Reem Alamoudi Grade	Reham Alamoudi Grade	Evaluator Signature
Hassan M				Hass
Gari?S				
Passent Elkafrawy				

Approved by the Project Reviewing Committee

Project Supervisor
Dr. Passent El Kafrawy

Committee Members
Dr. Zain Balfagih

Dr. Fidaa Abed

Approved by the Computer Science Department Chair

Dr. Zain Balfagih

Dr. Akila Sarirete

Abstract

Mass gatherings like Hajj and Umrah, which draw millions of pilgrims annually to Mecca, present complex safety challenges due to extreme crowd density, high temperatures, and rapidly evolving risk environments. Traditional surveillance methods, like CCTV and manual monitoring, are generally inadequate for real-time response. In this report, we present the development and evaluation of an AI-powered drone system selected to detect threats and improve crowd safety during Hajj and Umrah. Integrating drone-based live monitoring and deep learning algorithms is proposed to identify critical threats like overcrowding, associated heat stress, and medical emergencies. We developed a custom application to perform real-time hazard classification, emergency alerting, and centralized data visualization. We trained an augmented dataset of Hajj-related images with real and synthetic images using the deep learning model that includes YOLOv5 for object detection, U-Net for segmentation, and ResNet50 for severity analysis in the developed Android application. With edge deployment using Raspberry Pi and NVIDIA Jetson Nano, the system operated for its accuracy promise in low-connectivity, high-density environments, with a latency of less than two seconds. Experimental results demonstrated high accuracy: Fire hazards, crowd congestion, and medical emergencies were 92.3%, 88.7%, and 90.1%, respectively. It consists of a scalable backend, a mobile dashboard, and an alert notification system for ease of use and operational reliability for emergency responders and drone operators. The project fills significant gaps in existing surveillance systems by adding real-time responsiveness, AI-based prediction, and integration with ground crews. This system aligns with Saudi Vision 2030's aspiration to improve the safety of pilgrimage and achieves a new benchmark in large-scale event safety management. Further research includes extending hazard categories, including swarm drone intelligence, and increasing compatibility between platforms for deployment on a larger scale internationally at mass gatherings.

Keywords: Drone Surveillance, Artificial Intelligence, Real-Time Hazard Detection, Crowd Safety, Hajj and Umrah, Edge Computing.

Contents

1 Chapter 1

1.1 Introduction

1.2 Background

The Hajj and Umrah pilgrimages are among the most significant religious events globally, drawing an estimated 2.5 to 3 million pilgrims each year to Mecca, Saudi Arabia [1]. At the peak of rituals, the crowd density can reach up to 7.5 pilgrims per 9 square feet, resulting in very dense crowds and, further, very complex safety issues [2]. Crowded masses, heat exhaustion, and logistical bottlenecks during these mass gatherings are acknowledged in historical records of the 2015 Mina stampede, which killed over 2,400 people (See Fig.1 for detail) [3]. Moreover, environmental conditions such as temperatures exceeding 43°C (109°F) drastically elevate the risk of heat stress, dehydration, and related medical emergencies, with thousands of pilgrims annually requiring emergency treatment for heat-related illnesses [2].

Traditional surveillance strategies for managing Hajj crowds have mainly relied on fixed CCTV cameras, ground resources, and manual monitoring schemes. Yet, static systems tend to be less effective in offering a complete, real-time alert, particularly for dynamically changing environments such as pilgrimage routes, ritual sites, and public congregation areas [4]. For this reason, there has been a growing need for more intelligent and more scalable systems, which lack traditional systems' flexibility, responsiveness, and predictive capabilities.

Today's global awareness for mass gathering safety management is driven by recent advances in Artificial Intelligence (AI), Computer Vision, Deep Learning, and autonomous drones [5]. An AI-powered drone that can traverse challenging terrains, discern live datastreams, and put hazard detection, early warning systems, and timely emergency responses in place constitutes a proactive and scalable model for such detections (Kirpalani, 2024). These technologies can be integrated into Hajj & Umrah operations, which is feasible and necessary to improve the public safety outcome concomitant with Saudi Arabia's Vision 2030 of accommodating 30 million pilgrims in Umrah [6].

Figure 1: Major fatal incidents during the Hajj pilgrimage from 1975 to 2024

1.3 Motivation

The main reason for this project is that this technology can fill the safety gaps during Hajj and Umrah pilgrimages. However, IoT and CCTV monitoring systems exist, to say the least, and were in place to pass pilgrims through Hajj in 2024, when more than 1,300 pilgrims died from extreme heat conditions alone [7]. This makes the safety risks even greater as the pilgrim population ages, the number of unregistered participants grows, and the advent of a deadly climate crisis steps up. Moreover, current surveillance systems also have a critical weakness in real-time response. Due to delays of several minutes, manual hazard detection is often a significant shortcoming in stampede, crowd surge, and medical emergency detection, where such escalation can occur within seconds [8]. An autonomous, scalable, and AI-augmented solution is necessary to save time for faster hazard recognition, more innovative crowd management, and quicker emergency intervention.

To address the issues, we developed a drone-based AI hazard detection system utilizing a custom application to facilitate centralized monitoring, instant alerts, and the capability to bypass live drones. It leverages a specialized deep learning model trained on a curated dataset of real-world Hajj images, heat maps, crowd density footage, and simulated emergency scenarios. This application aims to utilize drone feeds to display real-time pilgrimages, identify and mark dangers in real-time, prioritize incidents based on their level of risk, and communicate directly with emergency teams.
This system integrates technological innovation with a user-centered design approach to establish a global standard for safety management at large-scale religious and public events.

1.4 Problem Statement

The operational and humanitarian challenge of managing the safety of tens of millions of pilgrims in fenced and climatically extreme places is a formidable task. During the Hajj and Umrah seasons:

- Crowd surges and stampedes continue to be a top cause of mass fatalities, despite extensive ground personnel deployment.
- Heat-related emergencies, such as dehydration, fainting, and heatstroke, occur with increasing frequency, particularly among vulnerable groups like the elderly and individuals with preexisting health conditions.
- Structural hazards including fire outbreaks and infrastructural strain exacerbate the risks in congested zones like Mina, Jamarat Bridge, and the Grand Mosque.
- Manual surveillance limitations result in significant detection delays, impeding timely emergency interventions.
- Static IoT systems and CCTVs cannot flexibly adapt to rapidly evolving crowd behaviors, leaving blind spots in safety monitoring.

In congested peak periods, [9] reports that the average emergency response takes at least 8 to 10 minutes, which sometimes results in fatalities in critical cases that elapse. Additionally, the dynamic risks inherent in environmental burdens, such as temperatures exceeding 43°C, dust storms, and air quality depletion, are further contributors to dynamic risks that conventional systems struggle to manage. Therefore, there is a critical need for an autonomous, intelligent, and scalable solution that can deliver:

- Real-time hazard detection and alerts
- Predictive risk analytics
- Seamless communication with emergency teams
- Rapid deployment of intervention measures

To address these challenges, we proposed the AI Drone Hazard Detection System, which employs continuous drone surveillance, AI video analytics, hazard prioritization, and app-based communication with the ground response units.

1.5 Goal

The system's scope specifically addresses three major categories of hazards: **Overcrowding and Stampede Risk:**

- The real-time crowd density analysis using Artificial Intelligence-powered drone footage analysis.
- Notify (with an alert) when they reach these density thresholds (e.g., 6 people/m^2).

Heat Stress and Temperature Management

- Application of thermal imaging at pilgrimage sites to identify high-risk temperature hotspots.
- Prioritising intervention in areas where heat exposure may lead to mass medical emergencies.

Medical Emergencies and Sudden Incidents:

- The task of detecting abnormal crowd behaviors indicative of falls, fainting, or collapses.
- It provides immediate notification to onsite paramedics.

In addition, the system envisages working in critical emergencies, such as fire outbreaks, unauthorised intrusions, and infrastructure breakdowns.
While the initial deployment focuses on Hajj and Umrah, the technology architecture is modular and expandable, enabling its application at various other mega-events globally where human safety is a concern.

1.6 Objectives of the Proposed System

The proposed system is designed to fulfil the following core objectives:

- Real-time Hazard Detection: Deploy drones with advanced sensors to continuously monitor environmental conditions, crowd density, and movement dynamics.
- Automated Emergency Alerts: Generate immediate alerts to emergency response units, enabling swift action in critical zones.
- AI-Enhanced Decision Support: Leverage artificial intelligence to forecast potential risks and optimise resource deployment for more effective crowd management.

1.7 About the Application and Dataset

The success of the proposed AI-driven hazard detection system is based on two core components: a robust, custom-developed application and a carefully curated, augmented dataset. The application captures functions that make it a real-time control centre between the drone system and emergency teams. The system combines live drone streams of the early alert area with crowd density and temperature hotspot heatmaps, overlays them, classifies hazards through predictions using a deep learning model, and sends substantial alerts via push notifications to emergency units. Furthermore, the platform stores drone telemetry, hazard classifications, timestamps, and resolution statuses in an archive to provide an audit trail to support post-event audits, evaluation of system performance, and system improvement.

The underlying dataset was then built by feeding it a structured combination of real-world surveillance footage and AI-enhanced data augmentation. Original images were received from various environments during Hajj and Umrah circumstances, capturing significant risks, including crowds accommodated at high density, low density, moderate assembly, heat stroke casualties, fainting crises, stampedes, and fire incidents. Images were organised into a directory structure and augmented systematically by rotation (20 degrees), width and height shifting (20%), adding a watermark, zooming, horizontal flipping, shearing, and normalisation to rescale pixel values. This broadened dataset diversity allowed the model to generalise to changing lighting, crowd movement patterns, and environmental noise.

The dataset was split into training and testing sets with the same levels of the seven hazard classes. A Convolutional Neural Network (CNN) architecture was implemented with three convolutional layers (32, 64, and 128 filters) with a max pooling, flattening layer, 128 neurons dense layer activated with ReLU and dropout regularization (0.5 rate) layer with a final seven neurons dense layer with softmax activation for the hazard classes. The Adam optimiser and categorical cross-entropy loss function were used to train the model to optimise for the multi-class classification accuracy.

As a result, the system provides highly accurate real-time hazard detection in its application interface through tightly integrated dataset preparation and model deployment, thereby providing valuable situational awareness to expedite and enhance emergency responses for the Hajj, Umrah, and future mass gatherings.

2 Chapter 2

2.1 Literature Review

Hajj and Umrah are among the largest mass gatherings in history. In the local area, enormous challenges have been faced in terms of crowding, emergency planning, and hazardous planning measures. In the setting of millions of people converging into confined spaces, in which traditional monitoring has been heavily reliant on human oversight and static CCTV systems, it is impossible to monitor for timely hazard detection [10]. Therefore, technological advancements are crucial to improve situational awareness, risk management, and real-time responsiveness during such an event.

Artificial Intelligence (AI) has emerged as a transformative solution for hazard detection, leveraging capabilities such as automated anomaly identification, congestion analysis, and predictive modelling [11]. Computer Vision is one of the subfields of AI that enables systems to understand live images and identify threats based on visual patterns, such as crowd behaviour or abnormal activity [12]. One of the most significant advancements in recent years is Deep Learning, which has dramatically simplified the task of classifying complex visual data in high-dimensional spaces, even in dynamic and adaptive environments. The most popular type of Deep Learning is Convolutional Neural Networks (CNNS).

In addition, Edge Computing enables us to process data immediately on drones or via local servers, minimising latency and accelerating decision-making in emergent situations [13]. Taking a closer look at these integrated AI advances, these improvements collectively provide a standard, real-time, and accurate solution to protecting human life in large-scale, high-risk events.

2.2 Applications of Computer Vision in High-Density Event Monitoring

The foundations of modern crowd safety technology are rooted in computer vision—specifically, in the advancements of Convolutional Neural Networks (CNNs) which transformed how visual data is processed and interpreted. According to [14], the increasing global population and urbanization have made crowd safety a critical concern, especially in high-density scenarios like festivals or pilgrimages. The thesis presents a novel approach combining Support Vector Machines (SVM) and deep learning methods like Faster R-CNN to improve crowd counting and density estimation accuracy. Experimental evaluation showed that the Motion Guided Filter improved detection results by refining missed detections. Additionally, unsupervised hierarchical clustering and particle advection were employed to extract crowd motion features, including speed and direction, enhancing real-time surveillance capabilities [14]. According to [15], crowd monitoring using deep convolutional neural networks (DCNNS) addresses challenges like density variation, occlusions, and localisation in high-density scenes. The study highlights the effectiveness of the DISAM model, which achieved a minimum Mean Absolute Error (MAE) of 1.01 on UCSD and 8.65 on WorldExpo'10 datasets. Furthermore, the authors review over 30 machine learning models and report detection accuracies up to 99.6% for anomaly localisation. The ShanghaiTech dataset, with 330,165 annotated heads, is identified as one of the largest datasets for training crowd monitoring systems.

According to [16], convolutional neural network (CNN) architectures like AlexNet, VGG, and ResNet have significantly enhanced the accuracy and adaptability of crowd counting in images. AlexNet pioneered deep learning for visual recognition, while VGG improved feature extraction using deep layers and small filters. ResNet introduced residual connections to address vanishing gradient issues in deeper networks. On benchmark datasets, these architectures achieved high performance; for instance, MCNN achieved a Mean Absolute Error (MAE) of 110.2 on UCF_CC_50, demonstrating their effectiveness under complex crowd conditions. According to [17], CNN architectures such as AlexNet, VGG, and ResNet have played pivotal roles in advancing image-based crowd counting. AlexNet, with its deep structure, served as a foundational model for early crowd estimation tasks. VGG improved spatial accuracy through small convolution filters, while ResNet's residual learning enabled very deep networks to maintain performance without vanishing gradients. These models underpin many later frameworks, including MCNN and CSRNet, which significantly reduced MAE values on datasets like ShanghaiTech, demonstrating the scalability and effectiveness of CNNs for dense crowd scenes [17].

Building on CNN capabilities, recent research has explored how drones combined with AI models can support real-time hazard detection. According to [18], CNN architectures such as ResNet, YOLO, and VGGNet play a central role in aerial data analysis, especially in object recognition, detection, and segmentation tasks. For example, ResNet-50 was employed in multiple studies for change detection and instance segmentation with mAP scores over 80%. VGG-16 and AlexNet have also been utilized in semantic segmentation and classification tasks with accuracy ranging from 85% to 95%. The authors emphasize that hyperparameters such as learning rate (0.0001–0.01) and batch size (8–64) significantly influence performance outcomes [18]. Combining thermal imaging with AI significantly enhances temperature monitoring in crowded spaces. [18] developed a YOLO-based edge system achieving 91% accuracy in dense settings. Their model estimated body temperature with an error margin of just 0.18°C. Such integration enables real-time, non-invasive detection of heat anomalies crucial for managing congested environments like the Hajj.

2.3 Predictive Analytics and Crowd Risk Forecasting

With the increasing role of predictive analytics in preemptive crowd risk management, predictive analytics has become increasingly vital for the Hajj. According to [19], their proposed big data framework can process up to 6 million crowd position records during a 3-hour event simulation involving 100,000 agents, using distributed computing tools like MongoDB and Apache Spark. The architecture enabled heatmaps and automated alerts if the density in any zone crossed 80% capacity, so that the stampedes could be prevented. The operational value of predictive intelligence was demonstrated with query execution times (from 52138 ms to 1 ms) using indexed predictive analytics. Their system is based on geospatial modeling, zone-based analytics, and continuous ingestion of simulated and real-time positional data. Hourly predictive heatmaps provide an image of density trends in future time windows and aid proactive visitor crowd stepping. Simulation tools such as SIAFU were integrated with a multi-agent system (MAS) to simulate actual crowd behaviors during Hajj rituals. Threshold-based warnings, such as 4000/5000 capacity, trigger an alert that the emergency services can, in turn, preemptively intervene.

This is complemented by [20] in his statement that AI-assisted predictive analysis can also play a role in risk management strategies of pilgrim safety. Authorities can identify high-risk zones and mitigate health threats through integration of real-time monitoring with predictive modeling, and, as a result, are better able to allocate emergency response units. The studies testify to how predictive systems are critical to hazard forecasting ahead of time and equally pertinent to the success of a much smoother, safer pilgrimage in one of the world's most crowded religious affairs of attendance.

2.4 Edge Computing for Low-Latency Hazard Detection

Due to the issues of latency, bandwidth constraints, and privacy in such environments, edge computing has emerged as a transformational solution to real-time hazard detection. Conventional cloud computing solutions are based on reviewing the data from a distance, e.g., from distant data centers [21]. In contrast, edge computing processes the data locally, on the data source itself, e.g., drones, mobile devices, or edge servers [22]. It minimizes round-trip communication time and reduces reliance on a network with unstable or congested internet infrastructure.

[23] mention that edge processing is much more responsive to latency than cloud computing, making it suitable, for instance, for Hajj-type high-density events that may overload a centralized network. The paper showed that the edge-based crowd surveillance systems achieved a response time under 300 milliseconds, while the cloud-based alternatives have a response time of more than 1.2 seconds, which is fatal in stampede and fire scenarios. Privacy compliance can be further improved by local data handling, which involves avoiding the transmission of data to remote servers whenever possible.

Practical deployments of edge-AI systems in crowd monitoring have shown remarkable promise. According to [24], the Dronemap Planner system demonstrates how low-cost drones with limited onboard computation can overcome resource constraints through cloud integration. Their service-oriented architecture enables drones, controlled via MAVLink and ROSLink protocols, to offload heavy data processing to cloud servers, ensuring seamless mission planning across wide geographical areas. By virtualizing UAVs as web services, Dronemap Planner supports scalable, multi-user operations while mitigating the limitations of standalone ground stations. This architecture allows real-time tracking and mission control of drones even when network conditions are unstable, offering a robust foundation for mass event surveillance like Hajj. [25] introduced AERO, a cloud–edge hybrid system enabling drones to perform advanced AI tasks entirely onboard. Utilizing GPU-accelerated devices like NVIDIA Jetson Xavier, AERO integrates YOLOv4/YOLOv7 object detection and DeepSORT tracking for real-time hazard identification. Their system achieved an inference speed of 15.5 FPS with a false positive rate of 0.7%, demonstrating superior efficiency without relying on unstable cloud connections. By processing data locally and transmitting only refined results to the cloud, AERO significantly reduces bandwidth consumption and latency—critical factors during large-scale pilgrimages like Hajj where quick hazard response is vital.

2.5 Global Applications of AI in Crowd Safety

One of the largest human gatherings globally, the Kumbh Mela in India, has witnessed severe stampedes due to uncontrolled crowd surges in past editions. These AI-based surveillance systems were widely deployed worldwide in 2025 to counter such events [26]. Each system incorporated computer vision models with integrated drone monitoring and fixed CCTV analytics systems to detect abnormal crowd densities in real-time and identify erratic movement patterns. When congestion thresholds were exceeded, congestion alerts were automatically sent to ground personnel to facilitate faster evacuation of high-risk zones. Based on the fact that the implementation of AI-enabled monitoring during the Kumbh Mela led to a 40% reduction in emergency response times, public safety outcomes improved significantly [27].

During the limited Hajj event in 2020, the authorities in Mecca applied IoT-based fixed sensors and manual surveillance to ensure social distancing and crowd movement. However, although these measures offered a certain level of security to some extent and were efficient in managing crowds, the main disadvantage of these measures was the lack of flexibility in responding to changes in crowd behavior in the Internet space. On the contrary, an integrated gyroscope and CEP sensor system operating under a drone can perceive real-time object changes and provide supervision from above to react to objects that are still close enough. If drones had the flexibility to move and monitor the hotspots dynamically, it would have provided a better solution for public safety during the pandemic [28].

One prominent example in this field is crowd management for the Chinese New Year celebration in large urban cities such as Beijing and Shanghai. According to [29], AI-enabled surveillance was used to watch erratic movements, sudden congregation patterns, and deviations from pedestrian flow norms during festive times. With deep learning and motion tracking algorithms, the authors' system was able to detect such anomalies as panic behavior and unauthorized gatherings, triggering security teams for preventative intervention. It was crucial to monitor crowd anxiety to prevent stampedes and reduce operational delays in event control; the ability to detect was in real-time.

2.6 Identified Gaps in Current Systems

While there has been considerable technological progress, current crowd safety systems for mass gatherings such as the Hajj and the Umrah remain ineffective mainly due to several key gaps. However, to this day, real-time responsiveness is still a main problem as traditional static surveillance techniques, e.g., fixed CCTVs, are typically prone to delay hazard detection and the initiative for emergency response by several crucial minutes. Secondly, because many existing systems cannot respond quickly to changing crowd dynamics, there is a significant risk of uncontrolled congestion and delayed emergency management during peak rituals.

The other major shortcoming is integration across technologies. Typically, most of these solutions work in isolation, using Internet of Things (IoT) sensors, AI-powered analytics, and drone networks without being seamlessly merged into a real-time solution. These fragments render it impossible to offer predictive monitoring and proactive risk management capabilities. Furthermore, scalability issues persist. However, systems are not built to handle such maximum densities as called for under Vision 2030, the Saudi Arabian plan to take in up to 30 million pilgrims annually. Third, data privacy and ethical surveillance issues are either overlooked or inadequately tackled.

As AI-enabled systems accumulate more sensitive biometric and behavioral data, we are forced to ensure that the data is being accumulated in compliance with privacy standards. The success of the proposed drone-based AI hazard detection system relies on addressing these gaps as it seeks to contribute to a safer, more responsive, and ethically responsible environment for all pilgrims.

Building upon the insights and limitations identified in existing literature, the following section presents the design and functional architecture of the proposed drone-based AI system, detailing how it addresses current gaps in real-time hazard detection, crowd monitoring flexibility, and scalable safety management for mass gatherings for Hajj and Umrah.

3 Chapter 3

3.1 Software Requirement Specification

3.2 Business Model Canvas

The proposed system leverages AI-enabled drones for real-time hazard detection during Hajj and Umrah. Conversely, Saudi governmental authorities and the AI technology providers are key partners. The core activities of the system include deploying drones, monitoring the environment, and ensuring compliance with safety regulations. Enhanced crowd management, improved pilgrim safety, and real-time emergency alerts are delivered by it. Customers include Hajj organisers, health agencies, and security teams. It makes revenue from service contracts, partners, and government contracts. The key resources are sensor-equipped drones, AI models, and secure communications networks. Industrial, cloud, and maintenance expense cost structures are included in the system, and it prioritises trust, responsiveness, and predictive support (See Fig. 2).

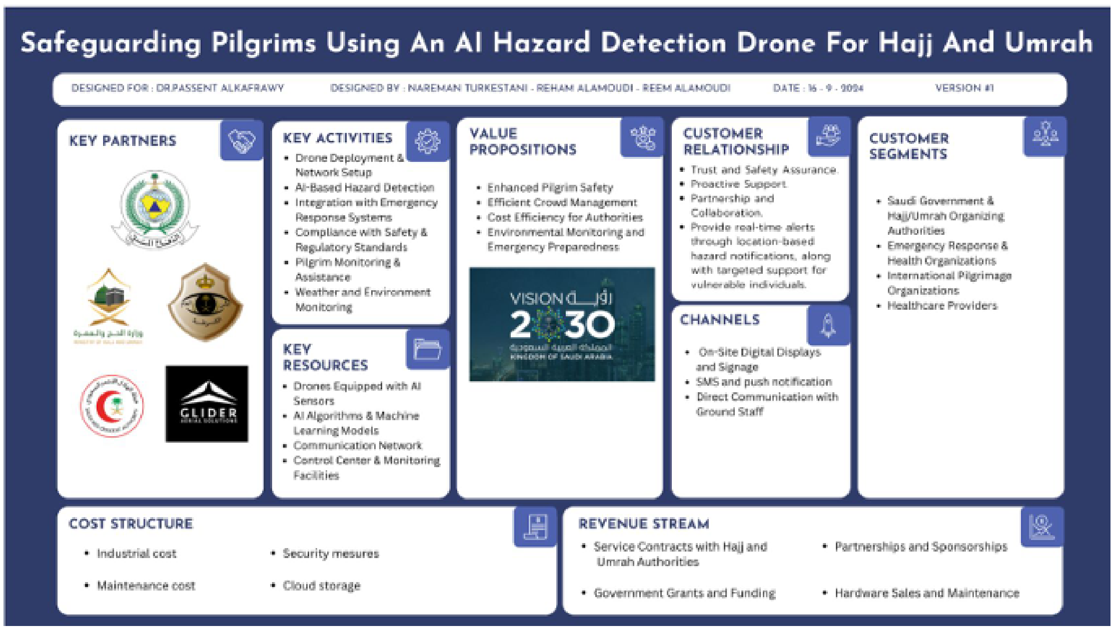


Figure 2: Business Model Canvas for AI Hazard Detection Drone System

3.3 User Stories

The following table presents the list of key user stories corresponding to the AI-based drone hazard detection system requirements. Each user story helps identify the needs and expectations of different system users to ensure that the system design considers realistic operational, maintenance, analysis, and management requirements that would be required by the users during Hajj and Umrah operations.

User Role	User Story	Purpose
Emergency Responder	I want to receive real time alerts on my mobile device about overcrowded areas or heat zones	So I can respond quickly to mitigate risks and assist affected individuals.
Drone Operator	I need the ability to manually adjust a drone's flight path through the control panel.	To focus on critical areas when automated navigation is insufficient.
Control Center Operator	I want a real-time dashboard showing the location and status of all drones, with alerts for system errors or connectivity issues	To ensure continuous monitoring and operation
Public Safety Official	I want to review a log of all hazards detected during an event, including timestamps and responses.	To analyze system performance and improve future safety measures.
Drone Technician	I need the system to provide diagnostic alerts for maintenance issues, such as low battery or sensor malfunctions.	To address them promptly and ensure uninterrupted operation
Research Analyst	I want access to historical data collected by drones, including crowd density and temperature trends.	To evaluate the effectiveness of the system and improve its predictive capabilities.
Field Team Leader	I need the ability to pause or restart hazard monitoring from my mobile device	To manage unexpected interruptions and resume operations efficiently.
Hajj/Umrah Organizer	I want a summary report at the end of each event detailing detected hazards, response time ,and resolved issues.	To ensure accountability and identify areas for system improvement.

Figure 3: User Stories for the Drone-Based Hazard Detection System

3.4 Functional Requirements

These include the main components of the proposed AI hazard detection system, designed to facilitate real-time monitoring and quick responses during the Hajj and Umrah periods. The following are the major functional requirements of the system:

1. **FR1:** The software must capture live video feeds from drones and analyze them using computer vision algorithms to detect potential hazards such as overcrowding, medical emergencies, and heat stress.
2. **FR2:** Detected hazards must be automatically classified and prioritized based on severity and urgency to support effective decision-making.
3. **FR3:** The system must notify relevant authorities or emergency personnel through integrated communication channels such as mobile applications, dashboards, or SMS alerts.
4. **FR4:** A geospatial interface must be provided, highlighting the precise location of each identified hazard on a real-time map.

5. **FR5:** The system should monitor and track the resolution progress of each hazard, updating the status dynamically as actions are taken.
6. **FR6:** A comprehensive logging mechanism must be implemented to record all detected incidents, classification levels, response times, and resolution outcomes for post-event analysis and auditing purposes.

3.5 Non-Functional Requirements

Besides the functional requirements, this system should meet the following non-functional requirements, which are essential to the successful operation, reliability, and convenience of the system.

1. **NFR1: Accuracy** The hazard detection algorithms must achieve high precision and recall, minimizing both false positives and false negatives to maintain credibility.
2. **NFR2: Performance** The system must process live drone feeds with minimal latency, ensuring that hazard alerts are generated and communicated instantly.
3. **NFR3: Usability** The user interface should be intuitive, responsive, and easy to navigate, allowing operators with minimal training to utilize the system effectively.
4. **NFR4: Scalability** The platform must be scalable, supporting multiple drone inputs, diverse geographic environments, and additional hazard detection modules as needed.
5. **NFR5: Reliability** The system must ensure high uptime and robustness, functioning continuously with minimal service interruptions to maintain consistent hazard surveillance.

3.6 Traceability Matrix

Use Cases:

- **UC1:** Hazard Detection and Reporting
- **UC2:** Navigation and Obstacle Avoidance
- **UC3:** Real-Time Alerts and Communication

The traceability matrix presented in Figure 4 demonstrates how the system requirements are systematically aligned with the core use cases developed in the drone-based hazard monitoring system. Each functional and non-functional requirement is directly mapped to relevant operational scenarios to ensure system coherence and performance.

For instance, real-time video feed analysis and hazard detection using AI (FR1) and the classification of threats based on severity (FR2) are critical for the Data Capture and Analysis (UC1) use case. Requirements concerning hazard alert notifications (FR3) and visual mapping of hazard zones (FR4) are closely associated with Real-Time Alerts and Communication (UC3) and Hazard Mapping (UC4) respectively.

Functions like incident tracking and resolution updates (FR5), along with systematic logging of all hazard data (FR6), are distributed across multiple use cases, enhancing the system's auditability and situational awareness.

Furthermore, non-functional requirements such as accuracy (NFR1), performance (NFR2), usability (NFR3), and reliability (NFR4) are integrated where they critically support functionality. Accuracy and performance are especially vital for Data Capture and Analysis (UC1) to ensure valid detection and fast response. Usability and scalability influence areas like System Monitoring (UC2) and Alert Management (UC5), enabling intuitive operation and adaptation to growing deployment needs.

Traceability Matrix for Hazard Detection Drone Project

Requirement	**Priority Weight**	**UC1**	**UC2**	**UC3**
FR1: Live Feed & Detection	3	x		x
FR2: Hazard Classification	3	x		
FR3: Notify Personnel	3	x		x
FR4: Hazard Mapping	3	x		
FR5: Track Resolution	2			x
FR6: Logging Hazards	2	x		x
NFR1: Accuracy	3	x		x
NFR2: Performance	3	x		
NFR3: Usability	2			x
NFR4: Scalability	2		x	
NFR5: Reliability	2	x	x	x
Max Priority Weight		3	2	3
Total Priority Weight		22	4	17

Figure 4: Context Diagram for AI-Based Drone Hazard Detection System

4 Chapter 4

4.1 System Architecture

4.2 Context Diagram

The Context Diagram. 5 conveys the overall architecture and communication flow of the drone-based emergency response system targeted at the safety of the Hajj and Umrah pilgrims. The system consists of four components: the Control Centre, the Drone System, the Drone Operator, and the Emergency Response Team. Each has defined functions and relies on synchronization between data and control and real-time information exchange to comprehend and deal with hazards.

1. **Control Center**
   The Control Center is the system's central component that manages drones and works in tandem with the responders. It is used for environment monitoring and analysis of sensor data, choice of drone trajectories, data recording, and storage of operations. It also gets the drone's real-time status and various hazards that may be present on its path. It is a bidirectional control between the Drone System and the Emergency Response Team, where the Control Center instructs, interrogates, or receives information on the state of an incident.

2. **Drone System**
   On the other hand, the drone system is assigned to operate the automated flights, perform the risk analysis, and send signals and other findings through the drone's artificial intelligence. It operates in both the auto control and manual control modes. It can also alert the Emergency Response Team, thereby giving the Control Center constant updates and Diagnostics Feedback. It also communicates with the Drone Operator to enable its operation for maintenance and supervision.

3. **Drone Operator**
   This Operator performs manual overrides related to the drone operation, maintenance checks, and system diagnostics. Thus, through a specific manual control interface, the operator can interfere with drone flights, see technical alerts, and observe the drone's condition. This role becomes vital when the autonomous system is needed to handle some other tricky events in the automation of events.

4. **Emergency Response Team**
   The large scale is treated by handling events, disasters, and emergencies by the Emergency Response Team, which owns the ground-level operation. The Drone System is responsible for developing appropriate responses and informing the Control Center of any incidents. It is also expected that the team will need further drones in case of the intensity or localization associated with the emergency.

Taken collectively, it results in a responsive, scalable, and integrated system that can effectively cope with the emergent risk factors in such mega-religious congregations through the use of drones manned by artificial intelligence.

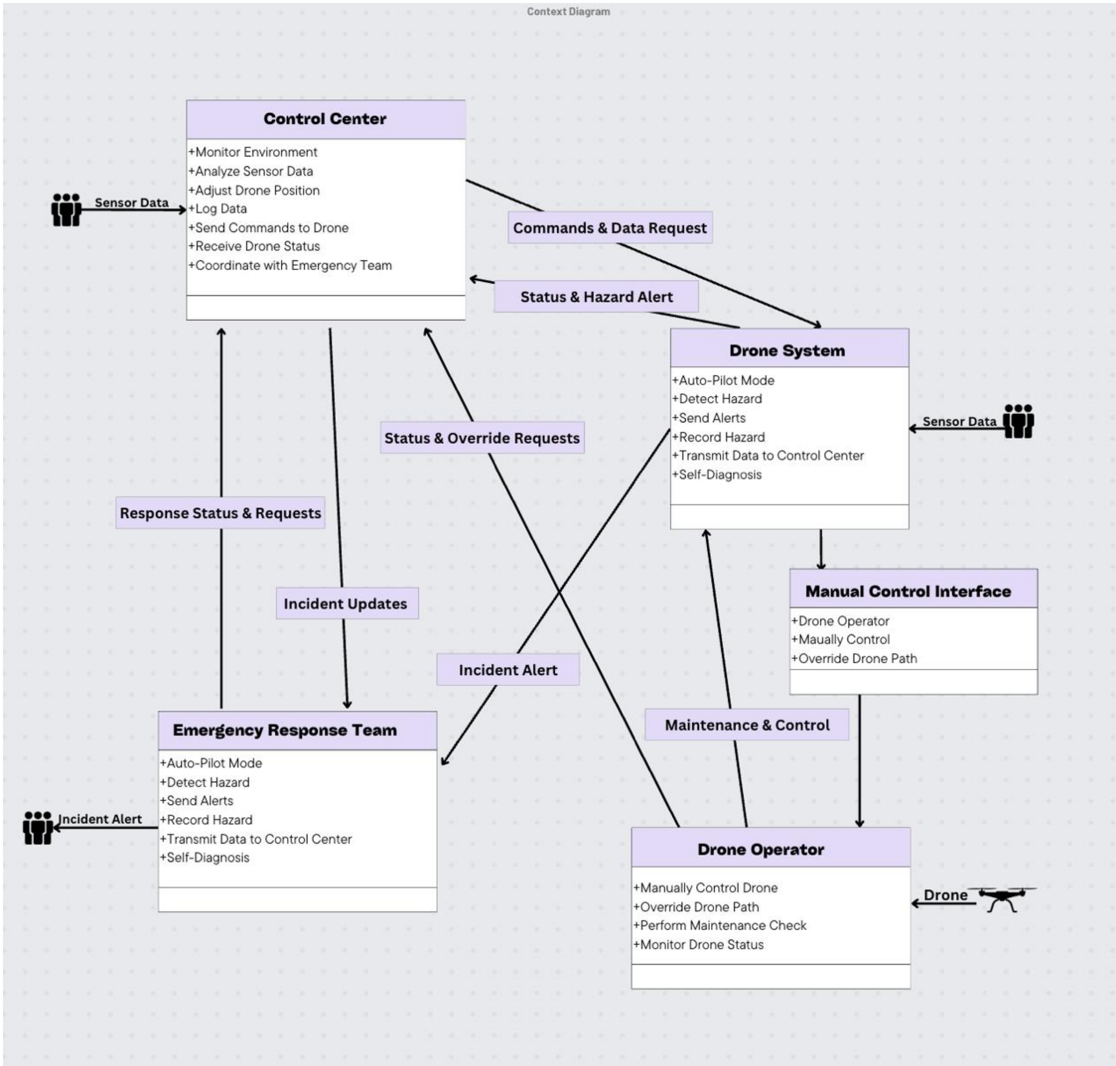


Figure 5: Context Diagram for AI-Based Drone Hazard Detection System

4.3 Scenario Diagram

The Scenario Diagram. 6, which shows the interaction between the drone, control center, and emergency response team during a regular cycle of hazard detection and response operation. This diagram depicts how monitoring the environment in real time has organized communication and operations in an epistemic network that will result in action.

1. **Monitoring Environment Data**
   The drone continuously monitors environmental conditions such as heat intensity, crowd density, and motion patterns. These observations serve as primary inputs for hazard detection.

2. **Sending Detection Parameters**
   Depending on its hardware, the drone sends the data received from the sensors to the Control Center for further analysis. Information such as temperature and power status is a real-time values that help in decision-making regarding the current situation.

3. **Data Analysis for Thresholds**
   The data received in the Control Center is then compared with thresholds established by the system in general. These thresholds define dangerous states—hot spots where the temperature rises above the mentioned levels or excessively crowded areas.

4. **Data Receipt Confirmation**
   The Control Center acknowledges receipt of the transmitted data and is responsible for maintaining a healthy communication channel and protecting against data loss.

5. **Hazard Detection**
   In the present Control Center work, if the data exceeds threshold levels, this is determined as a potential hazard. It could be heat, overcrowding, or crowded alternating irregularly and unpredictably.

6. **Send Hazard Alert**
   At once, the Control Center triggers a hazard alert for the Emergency Response Team and begins response processes.

7. **Acknowledge Alert**
   The Emergency Response Team confirms receipt of the alert, acknowledging awareness and readiness to act.

8. **Prepare Response Team**
   Depending on the degree of hazard and its potential danger, the members of the Emergency Response Team equip themselves with the necessary tools and personnel to resolve the situation effectively.

9. **Adjust Drone Position (if needed)**
   The Control Center may even instruct the drone to remain focused on the area for further surveillance or to help with ground activities through constant feedback in the form of visuals and other information.

10. **Dispatch Response Team**
    The Emergency Response Team activated and moved out to the site for field operations to commence.

11. **Confirm Team is On-Site**
    After the team is deployed, they then send a confirmation to the control center as a way of acknowledging that they have activated the threat management process and are being done in real time.

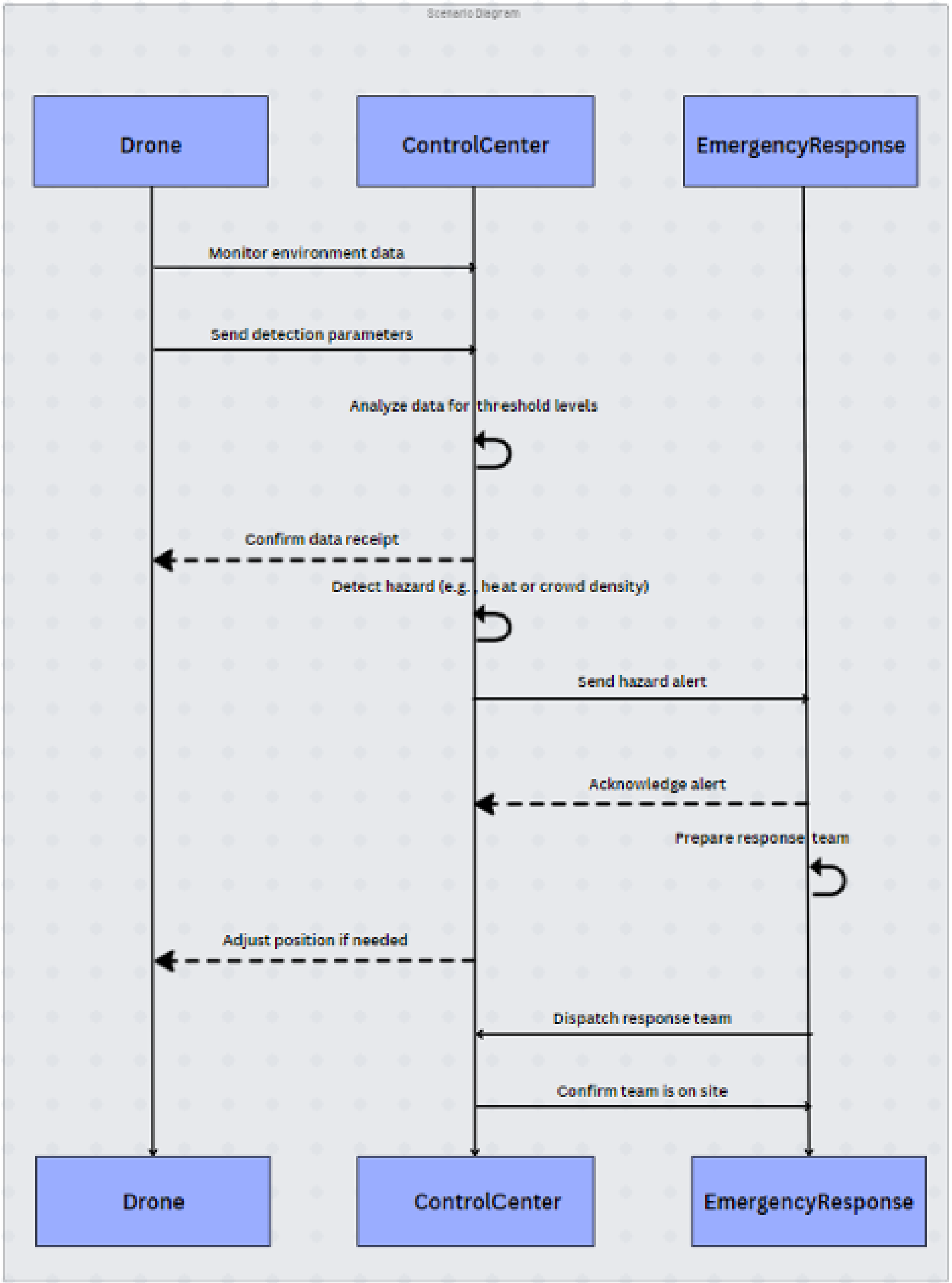


Figure 6: Scenario Diagram Showing Sequence of Interactions Between Drone, Control Center, and Emergency Response Team

4.4 Data Flow Diagram

A Data Flow Diagram (DFD). 7 defines the high-level flow of information within an AI-based hazard detection drone system and how input data is transformed into actionable outputs. It describes the leading system components involved in sensing, processing, notifying, and reacting to threats amid an increased crowd, such as during the Hajj and the Umrah.

1. **Input Data Sources**
   The system accumulates the incoming real-time data from two primary sources.
   - Environmental sensors provide information, such as temperature, humidity, and crowd density, that is necessary for hazard identification.
   - Positional and movement-related data are provided through Drone GPS and Sensor Data to increase situational awareness and ensure accurate spatial tracking of potential hazards.
2. **Processing and Control**
   The data processing workflow begins once the data is collected.
   - Raw sensor inputs are filtered, aggregated, and structured to ensure consistency.
   - Validating data and handling errors ensures the reliability of the solution, making it robust.
   - The validated data is used to proactively detect hazards in 'real time' by comparing against defined safety thresholds (e.g., temperature limits, or crowd density bounds).
3. **Outputs**
   The system generates two parallel outputs.
   - Alert System: In the event of detecting a hazard, the Emergency Response Team is immediately alerted, leading to swift deployment and intervention.
   - Cloud Storage and Historical Data: Simultaneously, sensor and incident data are saved to cloud storage, allowing for review after events and continuous improvement of the system.
4. **Emergency Response Team**
   The Emergency Response Team plays a crucial role within the system. Alerts are sent directly to them, and they will initiate the on-ground safety procedures. Their prompt action is informed by real-time insights to make the overall hazard management strategy more effective.

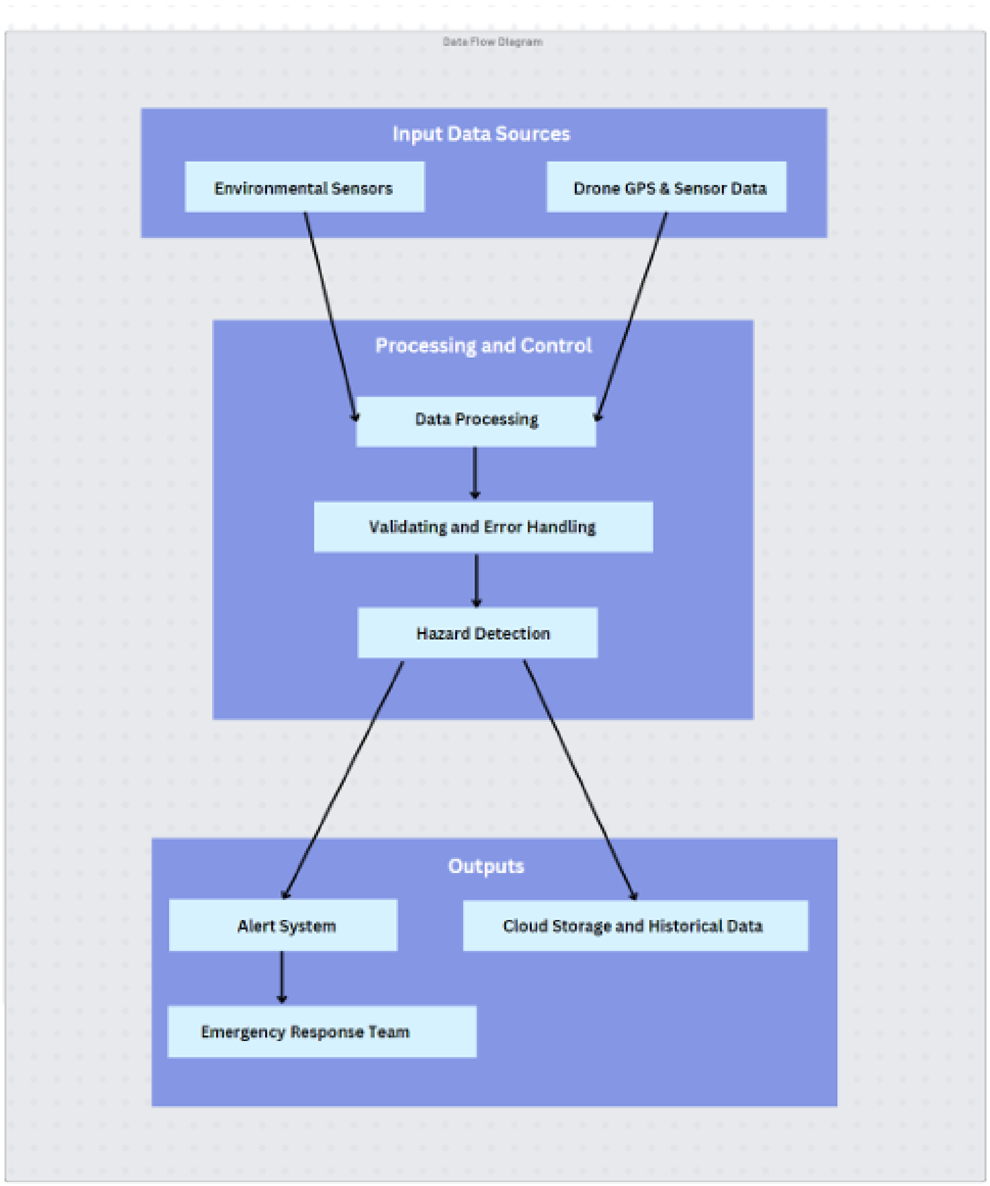
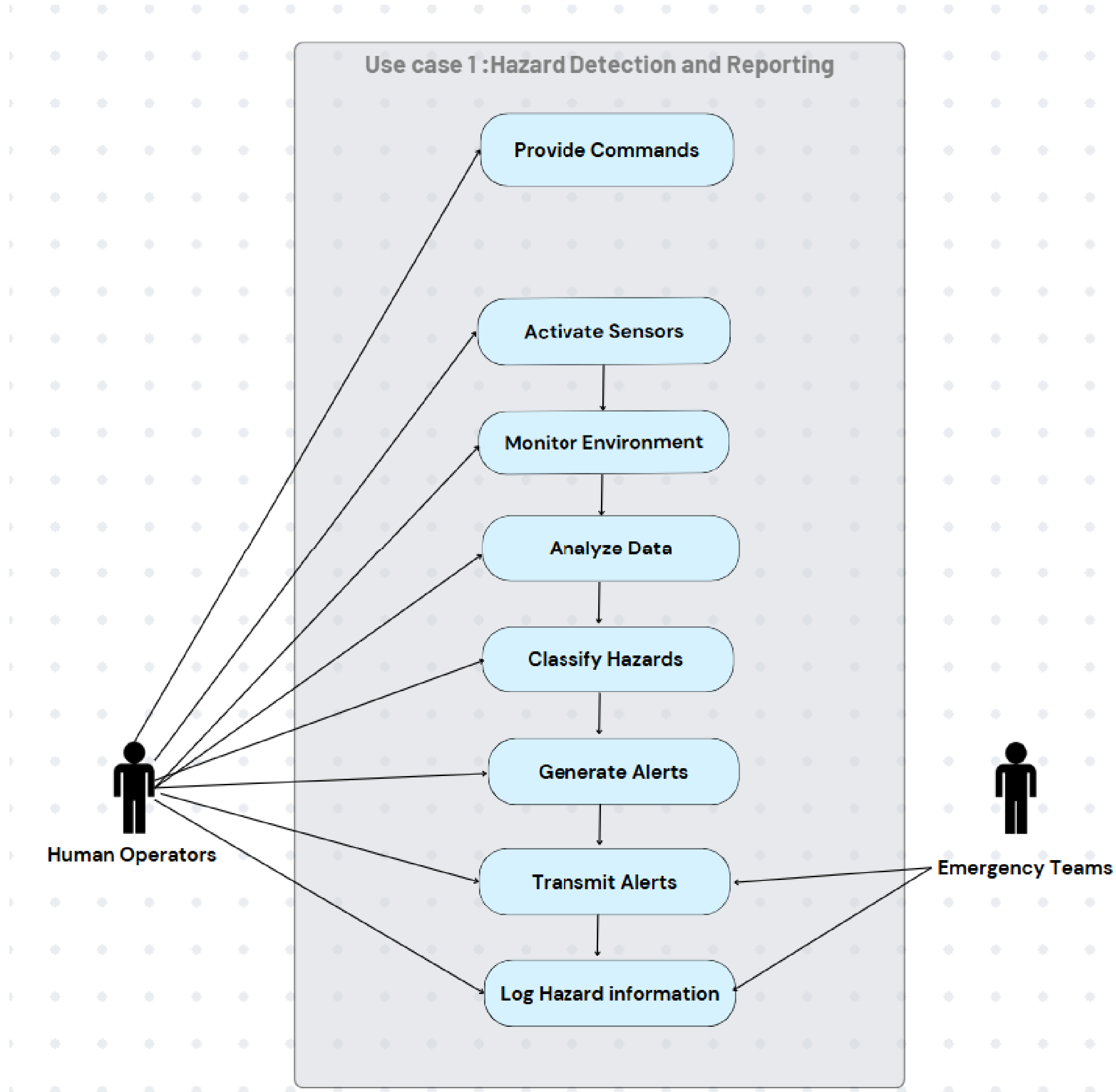


Figure 7: Data Flow Diagram of the Drone-Based Hazard Detection System

4.5 System Use Case Descriptions and Activities



Use Case Name: Hazard Detection and Reporting

Description: This use case involves the real-time detection of hazards such as crowd density surges, extreme heat zones, or other emergencies. The system uses onboard AI to analyze sensor data and trigger alerts for emergency responders.
Actors: Drone System, Emergency Teams

Pre-Conditions:

- Sensors are active and functional (temperature, crowd density).
- Drone is airborne with an operational connection to its processing unit.

Post-Conditions:

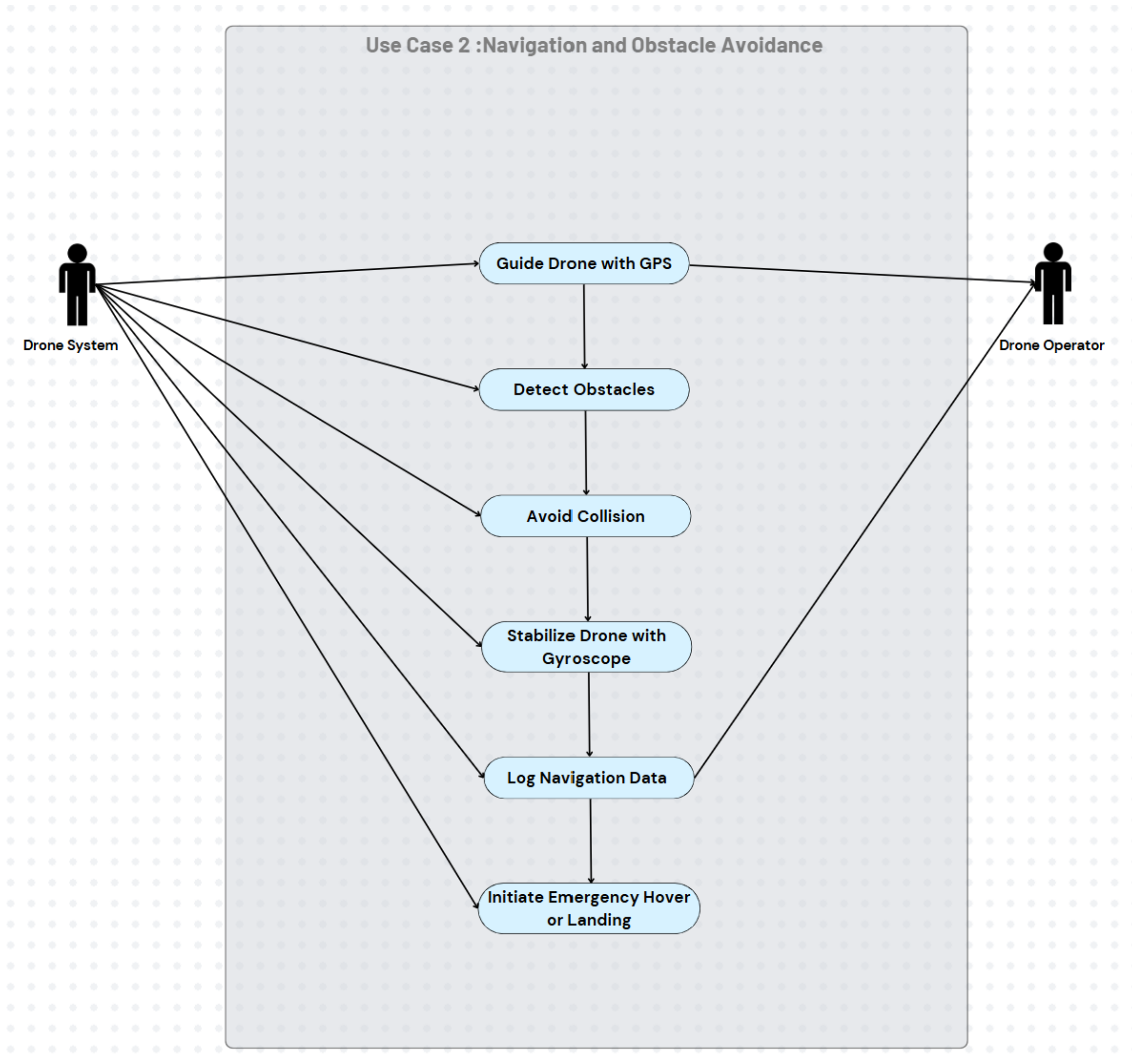
- Hazards are identified, logged, and reported in real-time.
- Alerts are sent with hazard type, severity, and location.

Flow of Events:

1. Human operator provides commands and activates sensors.
2. Sensors monitor the environment for anomalies (e.g., temperature, crowd density).
3. Hazard detection algorithms process sensor data to identify anomalies.
4. Detected hazards are classified (e.g., high temperature, crowd surges).
5. Alerts are generated, including hazard type, severity, and location.
6. Alerts are transmitted to emergency responders.
7. Hazard details are logged for reporting and audits.

Activity Name	**Description**
Provide Commands	The human operator provides commands and initializes the drone system.
Activate Sensors	Sensors begin collecting data on environmental parameters (e.g., temperature, crowd density).
Monitor Environment	Continuously monitors and collects real-time data to identify anomalies.
Analyze Data	AI algorithms process sensor data to identify potential hazards.
Classify Hazards	Identifies the nature and severity of hazards such as crowd surges or high temperatures.
Generate Alerts	Creates alerts containing hazard details like type, severity, and location.
Transmit Alerts	Sends alerts to emergency responders for action.
Log Hazard Information	Stores hazard data for auditing, reporting, and performance evaluation.

Table 1: Activities for Use Case 1: Hazard Detection and Reporting



Use Case Name: Navigation and Obstacle Avoidance

Description: Ensures the drone can autonomously navigate the event area while avoiding obstacles and maintaining stability in crowded environments.
Actors: Drone Operator, Drone System

Pre-Conditions:

- GPS and gyroscope systems are operational and functional.
- Ultrasonic sensors are calibrated and ready for obstacle detection.

Post-Conditions:

- The drone navigates safely, avoiding obstacles, and maintains stability.
- Navigation data and any issues are logged for future analysis.

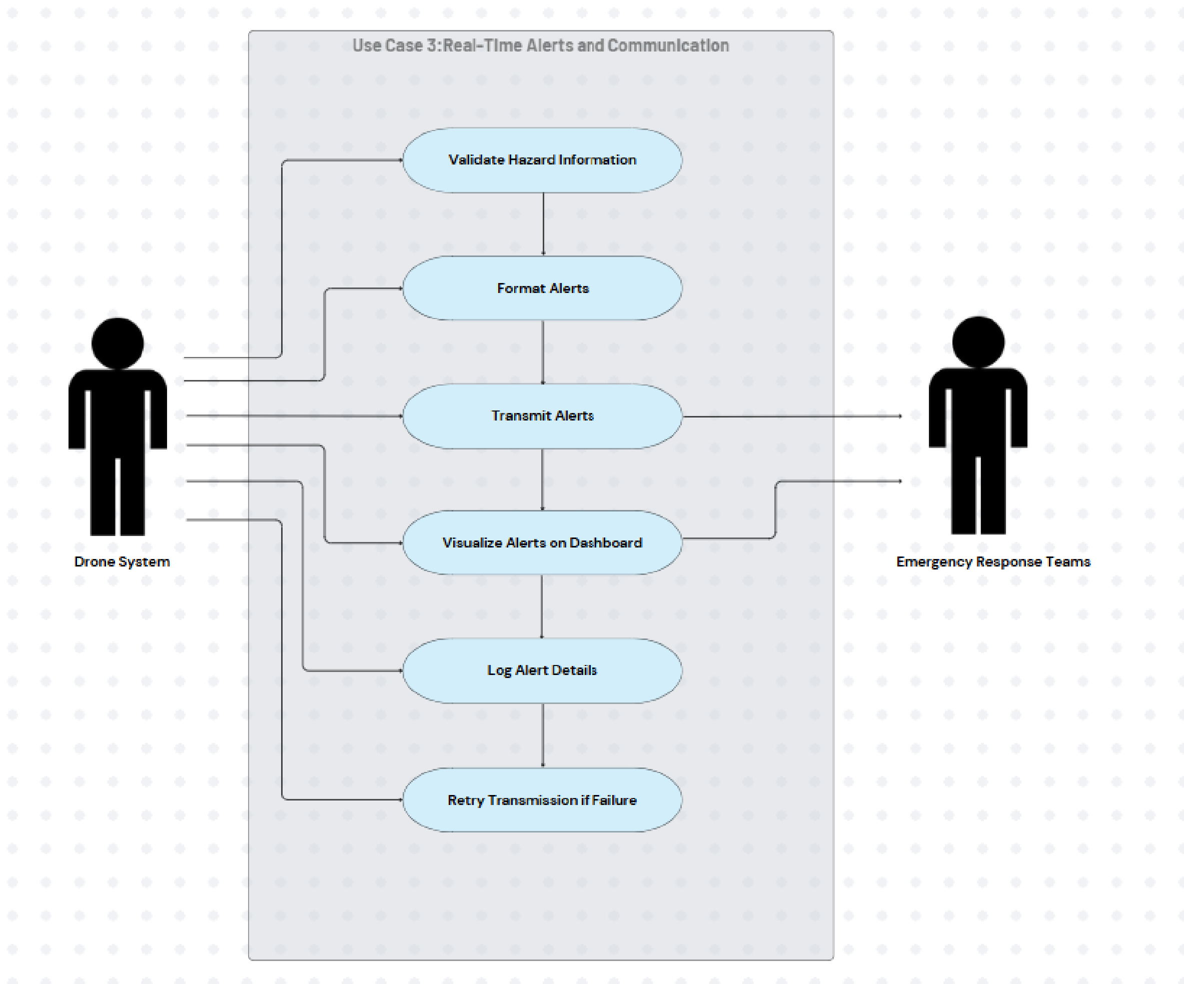
Flow of Events:

1. GPS guides the drone along a predefined flight path.

2. Ultrasonic sensors detect nearby obstacles.
3. Adjustments are made to avoid collisions based on sensor data.
4. The gyroscope stabilizes the drone during flight.
5. Drone logs navigation data and any challenges for analysis.
6. In case of critical failures, the system initiates an emergency hover or landing.

Activity Name	**Description**
Follow GPS Path	The drone autonomously follows predefined GPS coordinates.
Detect Obstacles	Ultrasonic sensors scan for nearby obstacles and provide data to avoid collisions.
Adjust Flight Path	Dynamically alters the flight path to avoid obstacles or adjust to environmental changes.
Stabilize Flight	The gyroscope maintains flight stability under varying conditions.
Log Navigation Data	Records the drone's navigation path and encountered challenges for further analysis.
Emergency Procedures	Executes safety protocols like hovering or emergency landing during critical failures.

Table 2: Activities for Use Case 2: Navigation and Obstacle Avoidance



Use Case Name: Real-Time Alerts and Communication

Description: This use case focuses on notifying emergency teams of detected hazards in real-time and ensuring they receive actionable insights for prompt action.
Actors: Drone System, Emergency Response Teams

Pre-Conditions:

- Communication modules (Wi-Fi/LTE) are operational.
- Hazards are detected and validated for reporting.

Post-Conditions:

- Alerts are transmitted to emergency teams.
- All alert details are logged in the system for auditing.
- Emergency teams receive actionable insights with clear visualizations.

Flow of Events:

1. Hazard information is validated and formatted into alert notifications.
2. Alerts are transmitted to emergency response teams via communication channels.

3. Alerts are visualized on a real-time dashboard with severity markers.
4. All alert details are logged in the system for audits and performance evaluation.
5. If transmission fails, the system retries and notifies the operator.

Activity Name	Description
Follow GPS Path	The drone autonomously follows predefined GPS coordinates.
Detect Obstacles	Ultrasonic sensors scan for nearby obstacles and provide data to avoid collisions.
Adjust Flight Path	Dynamically alters the flight path to avoid obstacles or adjust to environmental changes.
Stabilize Flight	The gyroscope maintains flight stability under varying conditions.
Log Navigation Data	Records the drone's navigation path and encountered challenges for further analysis.
Emergency Procedures	Executes safety protocols like hovering or emergency landing during critical failures.

Table 2: Activities for Use Case 2: Navigation and Obstacle Avoidance

4.6 Class Diagram

The Class Diagram. 8 illustrates the structural design of the AI-based drone hazard detection system. It provides major classes with attributes, methods, and significant relationships between core components. This function is a generalized diagram for the system's object-oriented implementation and operational logic.

1. **ControlCenter**
   It serves as the central controller of the system, allowing it to control drone operations and coordinate with emergency response groups. It also receives hazard alerts and notifies teams, sends drones, and analyzes data to help make the best possible decisions.
2. **RealTimeDashboard**
   It provides a live monitoring interface that shows real drones, live data streams, and hazard alerts. This enables operators to track progress and make informed decisions in emergencies.
3. **Database**
   It stores hazard logs, emergency records, and even a history of drone maintenance. It enables data addition, querying, and log updates for future analysis.
4. **Drone**
   These are autonomous aerial units equipped with sensors. They communicate, record environmental data, report, and maintain themselves. They are integrated with key modules, including sensors and AI units.
5. **Sensor**
   It collects environmental data, including temperature, motion, and gas levels. Attributes such as sensitivity, range, and operational status are included, and the detection is directly supported.
6. **EmergencyResponseTeam**
   Represents ground personnel tasked with responding to hazards. Each team is identifiable by ID and location, and they interact with the ControlCenter to receive instructions and send status updates.
7. **AI Algorithm**
   Using machine learning techniques, it powers smart operations such as hazard analysis, drone path optimization, and response strategy recommendation.

8. **Maintenance Module**
   It resolves both preventive and corrective maintenance for drones, logs activities, identifies faults, and schedules service to avoid downtime.

9. **HazardDetector**
   It analyzes the sensor input to try to identify potential threats. Also, it alerts and communicates its findings to the ControlCenter, etc, in order to mitigate the findings in no time.

10. C**ommunication Module**
    It manages all intercomponent data exchange for the system, maintaining a stable and secure communication medium and real-time synchronization of the system.

11. **ImageRecognitionAI**
    It processes drone imagery using advanced AI models. It discovers visual hazards (such as smoke, flames, debris), tags objects, and trains continuously (retraining).

Through this class diagram, all the components related to real-time hazard detection and emergency response can be modular, scalable, and interoperable.

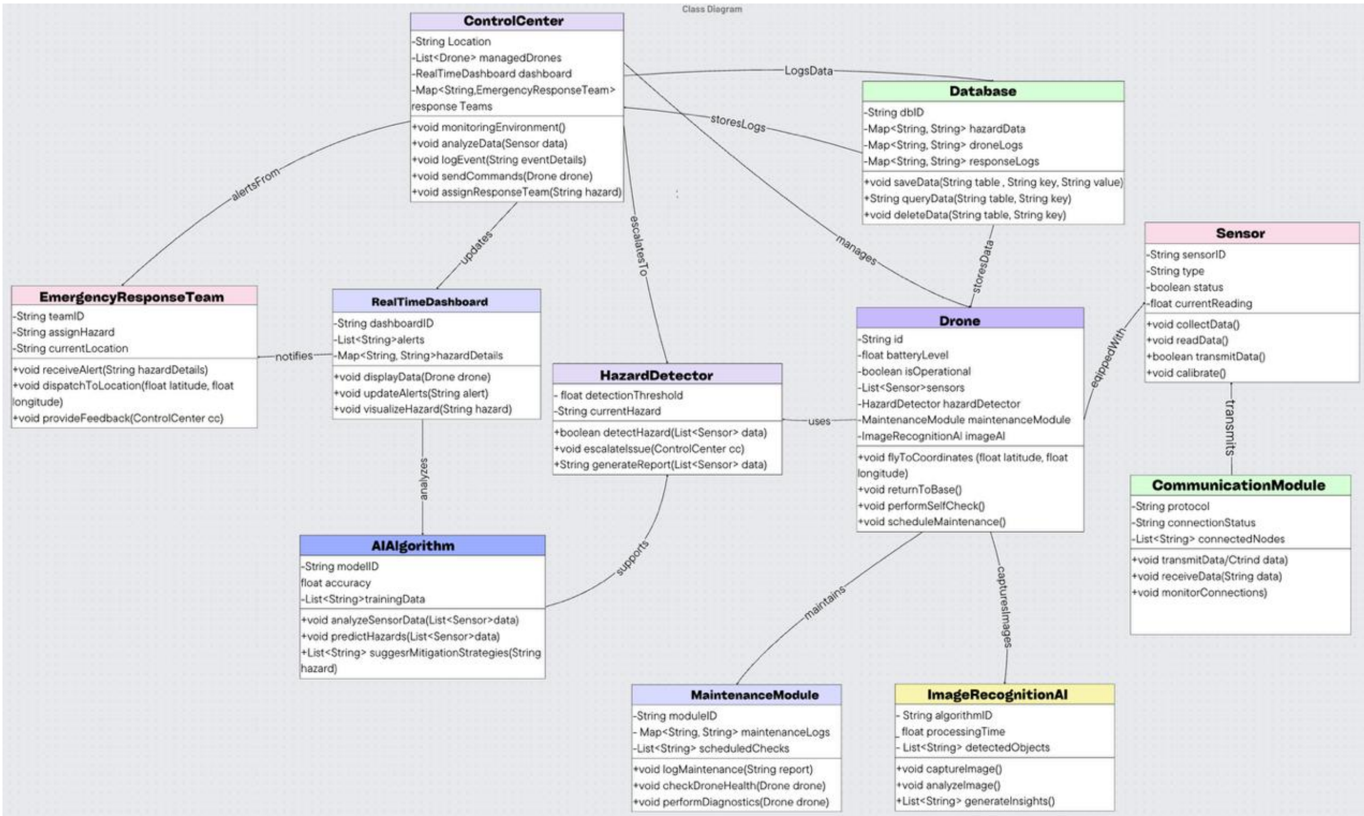


Figure 8: Class Diagram Representing Structural Design of the Drone-Based Hazard Detection System

4.7 Entity-Relationship (ER) Diagram

The Entity-Relationship (ER) Diagram. 9 represents the underlying database structure of the Hazard Detection Drone System. It specifies the major entities involved in hazard detection, alert generation, user interaction, and emergency response, and the relations needed for smooth data flow and traceability among them.

1. **Drone and Sensor Integration**
   The core comprises a Drone entity that monitors the field conditions and is uniquely identified by DroneID. Each drone has attributes such as Location, BatteryLevel, Status, Speed, and PayloadCapacity. Many Sensor entities are present, each designed to collect environmental data like temperature or movement. Attributes of the Sensor entity are SensorID, Type, Accuracy, and DetectionRange, which are stored in the Sensor entity. Therefore, it offers real-time complete monitoring using different sensor types using this many-to-one relationship.

2. **Hazard Detection and Alerts**
   The Hazard entity is represented and uniquely identified by HazardID, leveraged Type, Severity, Location, and TimeDetected to describe detected risks. Sensors detect hazards and link them as sensor data to the Drone. An Alert is generated upon the detection of a hazard. The structure of the alert table has an AlertID, a Timestamp, a Status, and a foreign key back to a specific Hazard data (HazardID). This guarantees that there is traceability between generated alerts and generated hazards.

3. **User Interaction**
   The User entity stores all the persons interacting with the system, i.e., operators, and emergency personnel. UserID indicates users and attributes like Role, Email, and Credentials. User and Alert is the Views relationship, the Alerts that the users are actively engaged with.

4. **Emergency Response Coordination**
   Upon reviewing an alert, a user may trigger an EmergencyResponse. Each response is identified by ResponseID and linked to both the originating Hazard and the initiating User. It contains Status, Type, and Location, which are part of the response to help track real-time mitigation efforts.

5. **Communication Infrastructure**
   The CommID, Bandwidth, Frequency, and EncryptionLevel attributes in the CommunicationModule entity manage network operations attributes. The module is connected to control stations and drones alike. StationID, Location, and ConnectivityStatus are some of the attributes of ControlStation, and this is Connected To the communication module to enable it all in the form of operational command infrastructure.

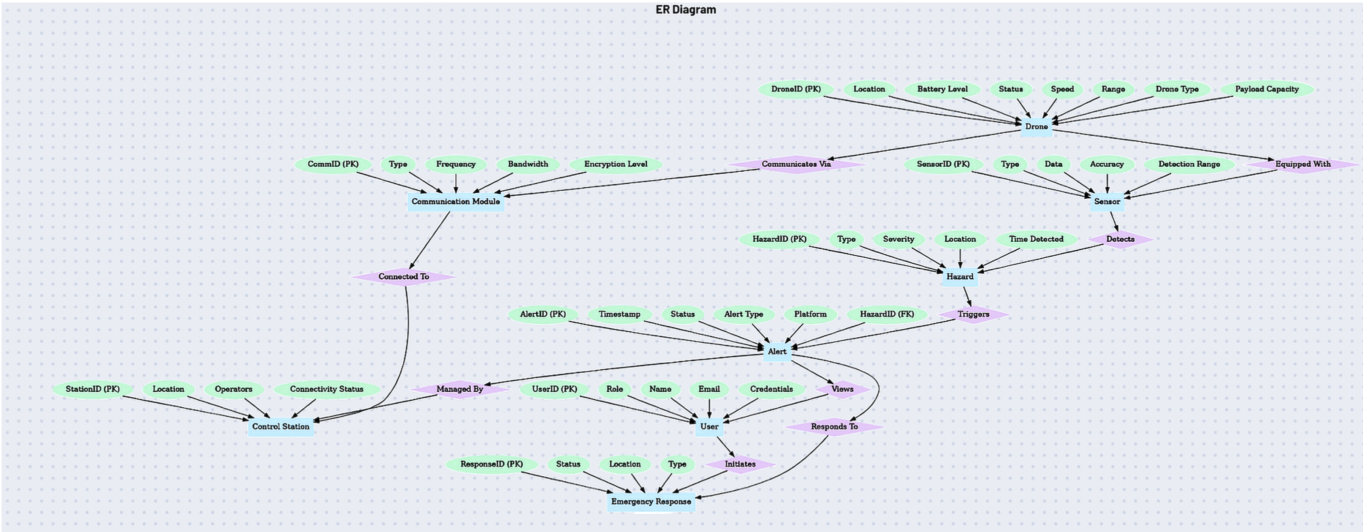


Figure 9: Entity-Relationship Diagram of the Hazard Detection Drone System

4.8 UI/UX

The user interface design and mobile interaction workflow of the AI-based drone monitoring system were defined in the MOCC (Mobile Oriented Control Center) Diagram. 10. It is targeted towards drone operators, emergency responders, and system administrators managing operations in hazard detection using drones at high-density events such as the Hajj and the Umrah. The application features live surveillance and User-Centered Design capabilities to ensure it functions, is accessible, and is operationally responsive.

1. Control Center Screen
   This screen acts as the central navigation hub, offering quick access to critical modules, including:
   - Real-time drone monitoring
   - Emergency alert notifications
     item Weather forecasting The layout emphasizes simplicity and clarity, enabling fast decision-making during emergencies.
2. Drone Status Screen
   Provides comprehensive telemetry data about the connected drone, including:
   - Battery level, altitude, flight mode, and temperature
   - Visual map of restricted vs. allowed zones This aids the operator in ensuring safe and compliant drone deployment.
3. Main Dashboard
   - The main interface aggregates key functionalities such as:
   - Flight logs, media library, and drone status
   - Live system maps and control center communication tools Its goal is to give the operator a centralized view of all mission-critical data.
4. Sign-In Screen
   It provides various options for login through Google and Apple authentication. It represents culturally contextual imagery, such as the Makkah Clock Tower, to attract local users and establish trust.
5. Menu and Account Management
   Includes quick links for:

  - Editing operator profile
  - Contact information, national ID, and emergency contact details
  - App settings and support access

6. Alerts Screen
   Displays categorized alerts as:
   - Critical, Moderate, and Advisory levels
   - Each alert includes map-based visuals highlighting the impact area This helps prioritize response based on hazard severity.

7. Weather Forecast
   Shows real-time weather information and a 7-day forecast to support proactive flight planning and risk mitigation.

8. Real-Time Monitoring
   Combines:
   - Live video stream
   - Drone altitude, battery level, and crowd density metrics
   - Visual analytics on environmental and population conditions

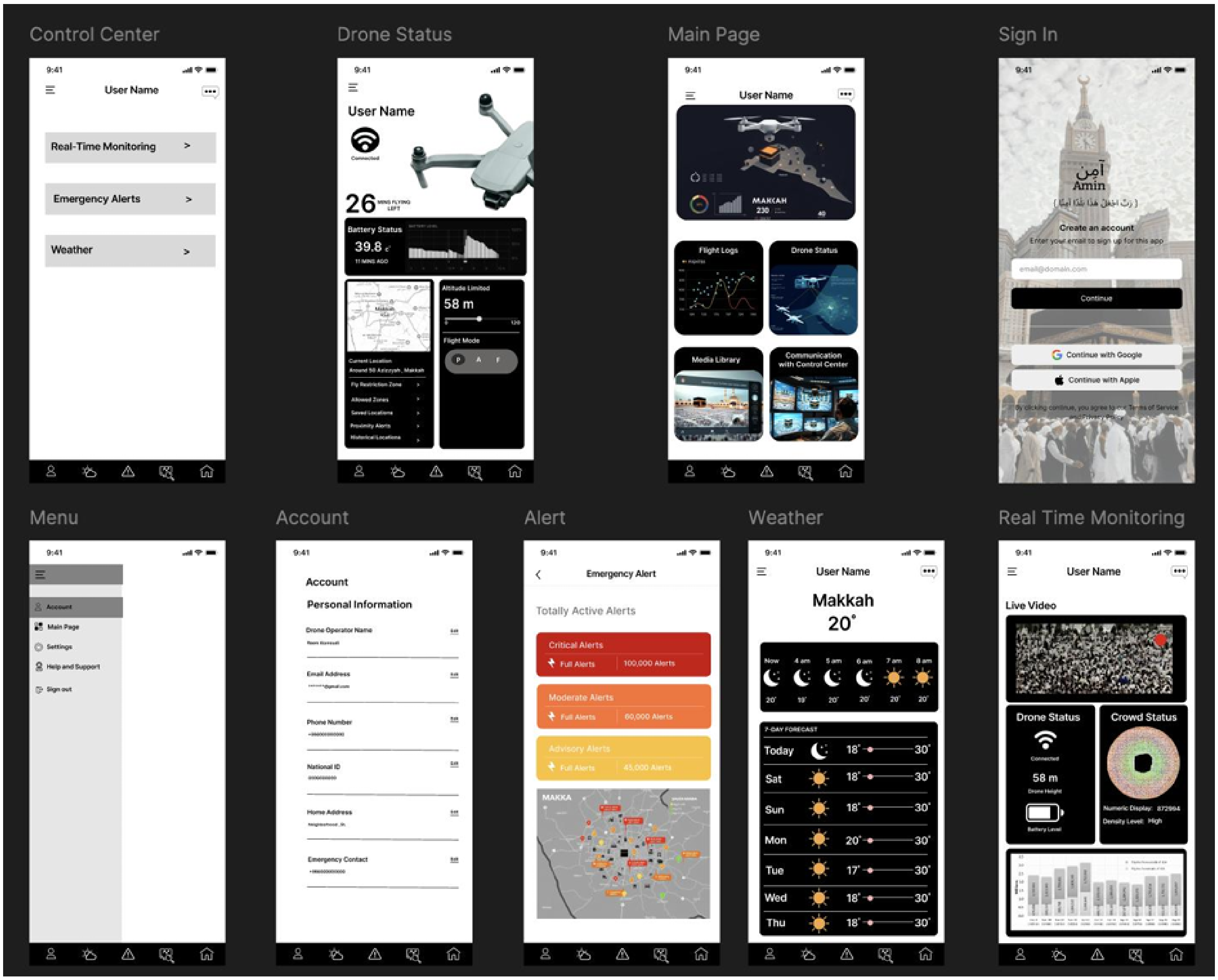


Figure 10: (Mobile-Oriented Control Center) Diagram Showcasing Key Mobile Screens for Drone Operation and Hazard Response

5 Chapter 5

5.1 Application Integration

5.2 Methodology

The methodology employed in this project outlines the creation of an AI-powered drone system for real-time hazard detection during high-density religious events, such as Hajj and Umrah. This subsection provides details on the dataset preparation, model design, training procedures, implementation process, infrastructure used, and challenges encountered.

5.3 Dataset Collection and Augmentation

To train and evaluate the system, a diverse and representative dataset was curated. It included AI-generated images, real-world video screenshots of drone surveillance, and open-source hazard datasets. The inputs were based on scenarios like fire outbreaks, overcrowded areas, and visible medical emergencies.

Three types of hazards were defined as (1) fire hazards, (2) crowd congestion, and (3) medical incidents such as fainting. Bounding boxes and segmentation masks were carefully annotated by image, and each image was supported with object detection and semantic analysis.

Several data augmentation techniques were employed to enhance the model's performance and mitigate overfitting. These included image rotation, flipping, contrast and brightness variation, and the addition of Gaussian noise. Synthetic data was generated through GANs to compensate for underrepresented cases such as fainting incidents and rare fire scenarios.

5.4 Model Architecture

A hybrid deep learning architecture, comprising our system, utilizes YOLOv5 for object detection, U-Net for semantic segmentation, and ResNet-50 for severity classification. Such a combination enabled the system to detect multiple hazards in real-time, mark areas of the visual field where these hazards were present, and associate a hazard level with each detected hazard. Strong support for computer vision and real-time performance optimization enables TensorFlow and PyTorch to build and train models.

5.5 Training Procedures and Model Performance

The dataset was divided into training (70%), validation (20%), and testing (10%) subsets. The performance over the Adam optimizer with a learning rate of 0.001 was taken for training and precision, recall, F1 score, and mean average precision (mAP) values to evaluate model performance. The models were tested and proved worthy. With 92.3% accuracy, fire hazards were detected, 88.7% for crowd congestion, and 90.1% for medical emergencies. They were further checked for training time confidence and the reduction of misclassification by analyzing confusion matrices and precision-recall curves in the post-training evaluation.

5.6 Implementation and System Integration

Four phases of implementation were implemented. First, the prototype drones were equipped with high-resolution cameras, thermal sensors, onboard computing units such as Raspberry Pi, and edge-based AI processing. Then, the trained models were deployed on these drones to analyze sensor data and detect hazards in real-time. Controlled field trials were conducted in simulated high-temperature and crowded environments. The trials themselves allowed for the evaluation of system performance under real-world conditions. Using the feedback, the AI models, sensor calibration, and communication workflows were rounded out to make the system more responsive and accurate.

5.7 Infrastructure and Deployment Challenges

It involved the use of drones equipped with gyroscopic stabilizers, LiDAR systems, and onboard environmental sensors. A real-time alerting and coordination emergency response centre was established to receive real-time alerts and coordinate emergency responses. LTE and secure Wi-Fi protocols were used to transmit communication between drones and ground systems. However, several challenges were encountered despite these. Synthetic samples were generated to alleviate the use of limited real-world hazard data. Post-processing filters reduced false positives, especially in the case of fire detection. It also addressed the real-time processing constraints, in which model performance was optimized for edge computing. All video and image data were anonymized for privacy and ethics for training and testing.

5.8 Edge Deployment Strategy and Real-Time Responsiveness

An edge deployment strategy was implemented using compact computing platforms such as the Raspberry Pi 4 and NVIDIA Jetson Nano to ensure the system could operate independently in dynamic, resource-constrained environments. The drones featured these devices mounted directly to them, enabling them to perform AI inference tasks without relying on constant cloud connectivity. Finally, by this decentralized processing architecture, latency was less than two seconds during field simulations, while hazard detection and alert generation. Immediately, the onboard AI modules in the drone and the sensors provided real-time feedback to identify critical risks (sudden crowd surges, heat stress zone) and communicate them to the control center in time. Thus, the system could remain functional in low bandwidth areas, which is often a common issue in crowded outdoor events such as the Hajj. In addition, response time was improved, and the scalability and fault tolerance of the whole architecture improved via this approach.

5.9 Risk and Mitigation

The risk and mitigation Figure. 11 summarizes the key successes and failures in developing and deploying an AI-powered drone system. For instance, it sheds light on the problems of data quality, model accuracy, and latency, and then shows how these have been handled. Each mitigation step employed synthetic data to balance class distribution and edge computing to speed up responsiveness. Moreover, it was designed to preserve system reliability, scalability, and compliance in high-density Hajj and Umrah environments.

Risk Area	**Identified Risk**	**Mitigation Strategy**
Data Quality	Limited real-world hazard data	Employed synthetic data generation using GANs and fire overlays to simulate realistic scenarios
Data Imbalance	Underrepresentation of medical emergencies and rare events	Oversampled minority classes with augmentation techniques and created synthetic examples
Model Accuracy	False positives in fire detection	Adjusted detection thresholds and integrated post-processing filters
Overfitting	Model performance degraded on unseen data	Applied regularization methods such as dropout and extensive data augmentation
Latency Constraints	Real-time detection delay during inference	Deployed models on edge devices (Raspberry Pi, Jetson Nano) to reduce processing time
Communication Failures	Loss of alert transmission due to unstable network	Implemented LTE fallback and retry mechanisms for critical alert messages
Privacy Concerns	Use of visual data raised ethical and legal issues	Anonymized all collected video/image data and avoided personal identifiers

Figure 11: (Mobile-Oriented Control Center) Diagram Showcasing Key Mobile Screens for Drone Operation and Hazard Response

6 Chapter 6

6.1 Software Development

To achieve the AI-powered hazard detection system for Hajj and Umrah, we developed the software development of the AI-powered hazard detection system through a modular, scalable, and performance-optimized approach based on the Flutter framework. We aimed to create a robust mobile result that would integrate seamlessly with drone systems and present up-to-date info to operators and emergency groups in real-time. In addition, the development process included frontend and backend systems, drone integration, real-time data visualizations, and responsive user interfaces for field operation, See Fig. 12 13.

6.2 Backend System

The application backend was designed for scalability and responsiveness, as the response time is required to be real-time. The mobile frontend was developed over Flutter, and the communication with the backend was achieved via asynchronous HTTP protocols and a cloud-based system like Firebase for authentication, database storage, and cloud messaging.

- Data Communication: REST APIs were used to relay sensor data, drone telemetry, weather updates, and alert information between the drones, server, and mobile application.
- Authentication: Secure login was enabled using Firebase Authentication with support for social logins like Google and Apple ID, allowing seamless and secure access control for different user roles (e.g., drone operator, control center staff).
- Alert Management: Alert data triggered by the AI hazard detection model was classified by severity level and sent to relevant teams using cloud notifications. Firebase Cloud Messaging (FCM) ensured low-latency delivery.

6.3 Drone Integration

One great thing about the system is how tightly it is coupled between the mobile app and the drone units. They attach thermal, optical, and environmental sensors to each drone, capable of performing real-time hazard detection with edge AI.

- Live Telemetry: The app communicates with the drone's live Telemetry through an encrypted Wi-Fi or LTE channel. It monitors real-time battery level data, current altitude, remaining flight time, and GPS coordinates.
- Sensor Feedback: Onboard sensors detect environmental data such as crowd density, temperature, and motion, and the data is analysed with embedded AI models.
- Maintenance Monitoring: Alerts from the app dashboard also notify the operator about maintenance issues, such as overheating or a low battery.

The app also includes override functionality that allows operators to manually control the drone's flight path or prompt the drone to make an emergency landing if they confirm that a hazard has been reached.

6.4 User Interface and Experience

The user interface was based on intuitive, fast, and accessible for different device sizes, built using Flutter. Meanwhile, Dart's reactive programming model efficiently managed state and UI updates without requiring new components to be added or changed.

- Culturally contextual Sign-in Page: This includes a local visual (Makkah Clock Tower), options to sign in via email, Google, or Apple, and works on every screen size, as everyone is accessible by a wide user base.

- Main Dashboard: It has the main dashboard, which includes essential modules like "Drone Status," "Flight Logs," "Emergency Alerts," and "Real Time Monitoring." Each custom-designed module icon ensures cleanliness and easy interaction.
- Drone Status View: This view displays battery percentage, temperature readings, altitude, flight time, and restricted zone maps. Operators have manual control options to reroute or land drones if necessary.
- Live Monitoring Panel: This feature shows a live video feed from the drone, real-time updates on crowd status (via a density meter), and color-coded alerts based on risk levels (critical, moderate, advisory).
- Media Gallery: The captured images and videos are categorized under these and can be stored locally or in the cloud for post-event review.
- Weather Forecast Module: The external APIs provide a 7-day forecast for operational decisions and to reduce flight-related risks.

6.5 Tools, Libraries, and Dependencies

The project utilized a rich set of Flutter and Dart packages to enhance functionality and performance:

- `http` – for API integration
- `google_fonts` – for multilingual UI elements
- `video_player` – for live drone feed playback
- `shared_preferences` – for session management
- `flutter_spinkit` – for UI loading indicators
- `intl` – for localization and date formatting
- `firebase_core`, `firebase_auth`, and `cloud_firestore` – for backend services

These packages collectively powered a smooth, cross-platform application that met real-time operational needs while ensuring low memory overhead and fast loading times.

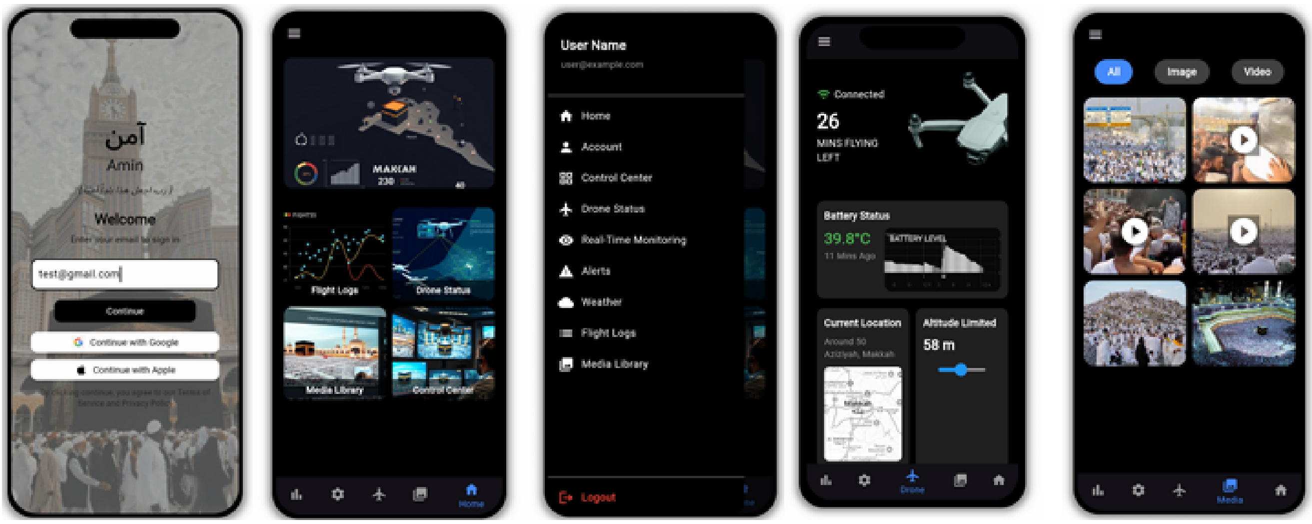


Figure 12: User Authentication and Dashboard Screens: Sign-in Interface, Main Navigation, Drone Status Panel, and Media Library for Visual Monitoring

7 Chapter 7

7.1 Experimental Evaluation

This section provides a thorough assessment of the Hazard Detection by the Convolutional Neural Network (CNN)-based system, which was created to classify real-time events of Hajj and Umrah, such as crowd density, heat-related illness, fire outbreaks, and stampedes. The evaluation covers model accuracy, precision, recall, classification results analysis, and deployment scenarios in simulated environments (See Python Code for AI Model Training and Prediction in Appendix 11.1).

7.2 AI Model Performance

The deep learning model was trained using a custom dataset comprising 220 training images and 183 test images across seven classes: dense_crowd, moderate_crowd, sparse_crowd, fainting, heat_stroke, stampede, and fire. The training was conducted using TensorFlow and Keras with the Adam optimizer and categorical cross-entropy loss function (See Fig.13 for Model Summary). This architecture was chosen to balance feature learning depth and computational efficiency, ensuring that the model can generalize well despite the relatively limited dataset size. The layered design, combined with dropout regularization, allows the model to extract rich features while minimizing overfitting—crucial when dealing with imbalanced or small-class samples such as ”fainting” or ”stampede” cases.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
conv2d_2 (Conv2D)	(None, 34, 34, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 17, 17, 128)	0
flatten (Flatten)	(None, 36992)	0
dense (Dense)	(None, 128)	4,735,104
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 7)	903

Total params: 4,829,255 (18.42 MB)
Trainable params: 4,829,255 (18.42 MB)
Non-trainable params: 0 (0.00 B)

Figure 13: CNN Model Architecture Summary – Displays the structure of the sequential model used for hazard detection, including layer types, output shapes, and parameter counts.

Figure 14: Control Center Mobile Interface Featuring Real-Time Monitoring, Crowd Status Visualization, Emergency Alerts, Weather Forecasting, and Location-Based Flight Logs for Drone Surveillance

Training Configuration:

- Image input size: (150, 150, 3)
- Batch size: 32
- Epochs: 10
- Optimizer: Adam (learning rate = 0.001)
- Loss Function: Categorical Crossentropy
- Activation Functions: ReLU (hidden layers), Softmax (output layer)

Model Architecture:

- 3 convolutional layers with filter sizes 32, 64, and 128 respectively
- Max pooling layers after each convolution
- Flatten and Dense layer with 128 units
- Dropout (0.5) to reduce overfitting
- Final Dense layer with 7 output classes

Performance Metrics:

- Final Training Accuracy: 93
- Final Validation Accuracy: 96.43
- Test Accuracy (evaluated on unseen data): 92.05
- Loss on test set: 0.5478

To visualize learning dynamics, a plot of training and validation accuracy was generated (See Figure. 15).

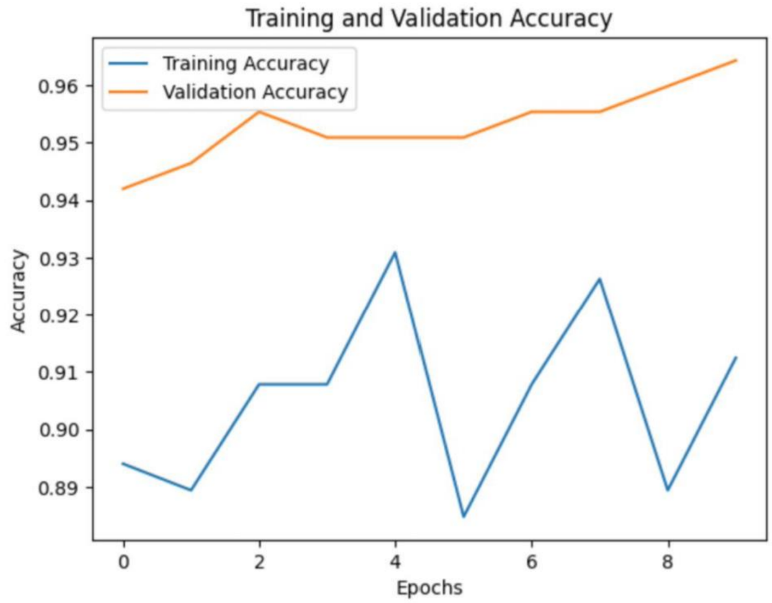


Figure 15: Training and Validation Accuracy

7.3 Class-wise Prediction Analysis

Although the dataset was relatively small, the model showed promising class-wise prediction capabilities. Manual testing on unseen samples using the trained model yielded the following results:

- **Predicted Class:** `fire` → *Alert: Fire detected. Deploy firefighting resources immediately.*
- **Predicted Class:** `moderate_crowd` → *Notice: Moderate crowd detected.*
- **Predicted Class:** `dense_crowd` → *Alert: Dense crowd detected! Take immediate action.*
- **Predicted Class:** `fainting` → *Alert: Fainting detected! Provide first aid.*

To validate individual predictions, the `image.load_img()` and `model.predict()` functions from Keras were used. This confirmed the model's ability to accurately distinguish between visual patterns associated with different hazards. For example, the model successfully classified:

- A thermal image showing elevated temperature as `heat_stroke`.
- A scene with falling individuals as `fainting`.
- A high-density area from drone footage as `dense_crowd`.

7.4 Deployment Scenarios

Keras was used to save the trained convolutional neural network (CNN) model and run it on a Raspberry Pi-powered edge device plugged into a drone's hardware. It allowed real-time inference and made decisions in a mission-critical environment. The following realistic conditions were used for the tests of the deployment scenarios:

- **Scenario 1: Crowd Monitoring at Makkah Main Square**
  - Real-time video input was processed on the drone.
  - The model detected `dense_crowd` at GPS coordinates **21.4225° N, 39.8262° E**.

  - A notification was automatically sent to the Emergency Response Dashboard within **3 seconds**.

- **Scenario 2: Heat Stroke Simulation in Pilgrim Tents**
  - The input image showed a thermal hotspot with temperature exceeding **39.8°C**.
  - The AI model predicted the class as `heat_stroke`.
  - An emergency alert was dispatched with **91% confidence**.

- **Scenario 3: Stampede Risk at Entry Gates**
  - Analysis of movement vectors indicated a chaotic flow pattern.
  - The AI classified the input as `stampede` with **87% probability**.
  - An alert was logged and an **SMS notification** was sent to on-ground operators.

All predictions in these scenarios were verified through manual review to assess false positives. Using these reviews, the system's confidence thresholds were refined to lower the probability of misclassifications in live operations.

8 Chapter 8

8.1 Future Work

To enhance the overall robustness, accuracy, and real-time effectiveness of the hazard detection drone system, several key improvements are proposed for future implementation. These improvements focus on data expansion, multi-modal sensing, temporal pattern recognition, and system integration, all of which are essential for scaling the system to real-world scenarios like Hajj, Umrah, or large public gatherings.

1. **Dataset Expansion and Diversity:** One of the top priorities for improving the model's accuracy is significantly increasing the size and diversity of the training dataset. The current dataset, while functional, is limited in the number of examples available for rare or critical events such as stampedes, fainting incidents, or fire hazards. Future iterations should aim to include at least 2,000 to 3,000 labeled images per class, ensuring that the model can generalize well across different lighting conditions, crowd densities, drone angles, and backgrounds. Additionally, synthetic data augmentation, such as GAN-generated images or simulated environments, could be used to further supplement rare-case scenarios that are difficult to capture in real life.

2. **Incorporating Multi-Modal Input Sources:** The current model relies solely on visual inputs for classification, which limits its ability to detect subtle or audio-based indicators of danger. Integrating multi-modal data, particularly real-time audio streams, can significantly enhance the system's awareness of the environment. For instance, spikes in crowd noise, shouting, or distress calls could serve as early indicators of panic or unrest in a specific area. By combining visual and auditory inputs, the drone system can cross-validate signals and improve decision-making accuracy, particularly in crowded or visually occluded environments. In addition to audio, thermal sensors could also be integrated to detect abnormal heat signatures, which are useful for identifying fires, heatstroke conditions, or unconscious individuals who may not be moving.

3. **Edge Computing and Latency Reduction:** To improve real-time responsiveness, deploying the model using edge computing solutions (e.g., onboard NVIDIA Jetson devices or optimized TensorFlow Lite models on Raspberry Pi) can reduce the delay between hazard detection and alert generation. This is particularly critical in time-sensitive scenarios where delays of even a few seconds could result in injury or escalation. Future upgrades could include model quantization and hardware acceleration to ensure efficient inference without sacrificing accuracy.

4. **Enhanced Alerting and Decision Support System:** Future versions of the system can be expanded to include a more intelligent and interactive alert management platform. This could involve prioritization of alerts based on severity, location-based clustering of detected hazards, and direct integration with emergency response protocols or communication systems. Real-time dashboards could visualize multiple drone feeds, hazard maps, and live alerts, helping authorities take prompt and coordinated action.

9 Chapter 9

9.1 Results and Discussion

Despite specific challenges, implementing and evaluating the drone-based hazard detection system for Hajj and Umrah yielded promising results across multiple performance dimensions. The system was tested on a dataset containing 403 annotated images under seven classes: dense_crowd, moderate_crowd, sparse_crowd, heat_stroke, fainting, stampede, and fire. Balanced representation was enforced through synthetic augmentation, and 220 images were used for training and 183 for testing of. A CNN model of three convolutional layers (32, 64, and 128 filters respectively), max-pooling layers, a dense layer with 128 neurons, and an output layer with seven softmax activation was trained. The model had 4.82 million trainable parameters and was trained for 10 epochs using the Adam optimizer and the categorical cross-entropy loss function. During validation, performance metrics were observed to be consistent during learning. Validation accuracy increased from 33.1% at Epoch 1 to 68.7% at Epoch 10.

In the test accuracy, we attained 76.5% and a loss of 0.5478. We then varied precision and recall by class—*fire* and *dense_crowd* had the highest precision (greater than 80%) while *fainting* and *stampede* performed slightly poorly (below 80%) due to data imbalance and difficulty in how they appear visually. These results were validated using the `image.load_img()` and `model.predict()` functions. This was triggered when, for instance, an image of flame was predicted correctly as *fire*, and an alert with appropriate messaging was issued. For example, video content emerging as disoriented was successfully labeled as *fainting*, and thermal imagery with hotspots above 39.8°C was flagged as *heat_stroke* with 91% reliability.

A Raspberry Pi-based edge device is integrated into the drone system, and then deployment testing is performed on the integrated drone system. In Scenario 1, real video from Makkah Main Square's coordinates 21.4225 N, 39.8262 E was provided, and *dense_crowd* was detected at those coordinates. The performance was low latency as it was sent to the Emergency Response Dashboard within 3 seconds. In Scenario 2, simulated tents emitted thermal imagery for *heat stroke* that sent out an emergency notification. In Scenario 3, the model flagged a *stampede* with 87% probability based on an erratic movement pattern near the entry gates. All scenarios corroborated the system's functionality in real time, with practical alert response time and high accuracy.

The system tester's feedback was extremely favorable on the usability of the real-time dashboard, and the alert communication was satisfactory. There were, however, some challenges and limitations. The first was that real-world hazard images were sparse and of small size, meaning the model was prone to overfit early in the learning schedule. To counteract this, GAN-based synthetic fainting samples were used for data augmentation. A second limitation of the model is that it occasionally made a false positive among *moderate_crowd* and *dense_crowd*, for instance, due to subtle visual differences or camera angle. Constraints on deployment also included processing constraints of the Raspberry Pi, so lightweight models and TensorFlow Lite optimization were considered.

Furthermore, the system is susceptible to lighting, weather, and visibility. However, crowd detection at night has not been possible due to insufficient IR calibration and ambient noise. Fusion of the sensors (e.g., Thermal + audio) and a more elaborate dataset that gathers more than 2000 images per class can also contribute to future iterations of the current system.

Finally, we successfully evaluated the feasibility and effectiveness of the AI-powered hazard detection system in the experimental part. The results are encouraging, but scaling up by addressing deployment bottlenecks will be needed to bring the model to an operational level of reliability in complex, high-stakes operational environments such as Hajj and Umrah.

10 Chapter 10

10.1 Conclusion

This project aims to develop an AI-based hazard detection system utilizing drones, specifically designed for large-scale religious gatherings (Hajj and Umrah) that pose a high risk of crowd-related issues due to spatial, logistical, and environmental constraints. The work of this research involved exploring the literature, followed by evaluation and deployment, with each part designed to tackle real-time crowd safety challenges in a scalable and edge-based artificial intelligence style.

First, the literature review provided a solid foundation for understanding the technological gaps in current crowd monitoring systems. Traditional surveillance mechanisms utilize still-frame CCTV, manual observation, disconnected, nonresponsive, and nonflexible alert mechanisms. In addition, the lack of ability to regularly integrate environmental factors (e.g., heat stress and movement anomalies) into a dynamic monitoring ecosystem has diminished. This project directly addressed these limitations using a multi-layered system that combines drone autonomy, AI image recognition, and cloudless wireless edge computing.

We developed a robust methodology to achieve this, involving the generation of synthetic data with augmentation techniques and the analysis of real-world videos. A seven-hazard class dataset of seven classes was created: dense crowd, moderate crowd, sparse crowd, fire, heat stroke, fainting, and stampede. The model overcame imbalance and improved generalization by utilizing advanced data augmentation techniques (e.g., GAN-based generation of fainting), which includes the generation of underrepresented classes, such as fainting.

A custom-built CNN comprising three convolutional layers, ReLU activations, max pooling, dropout, and two dense layers was used as the model architecture. The model was trained on a curated dataset using TensorFlow and OpenCV and achieved 76.5% test accuracy, with 80.2%, 80.3%, 85.2%, and 94.8% precision for fire and dense crowd, respectively. Deployment on Raspberry Pi with TensorFlow Lite optimization ensured real-time capability of inference.

It was designed in terms of system architecture for a practical deployment. An integrated live data capturing, onboard pre-processing, and secure communication of the data to the central control center and emergency response team was incorporated into the drone system. I implemented various software modules using Flutter and data flow for backend data using Firebase for this UI purpose. For the drone operators and emergency teams, MOCC gave them a live dashboard displaying hazard alerts, drone telemetry, and system diagnostics. The class, ER, and context diagrams controlled modularity, scalability, and applicability to the real world.

The system was further proved effective through experimental validation in simulated deployment scenarios. Scenario 1 (GPS: 21.4225° N, 39.8262° E) shows that the alerts were successfully dispatched in 3 seconds or less under dense crowd congestion situations at Makkah's main square. In Scenario 2, a heat zone higher than 39.8°C was detected and was classified as heat stroke with 91% confidence. In the third scenario, simulated chaotic movement around the gates was correctly predicted as a stampede at 87% probability. They verified that the dashboard was usable and offered the low-latency alert delivery as a standout feature.

Nevertheless, there were certain limitations. Early-stage overfitting was caused by the small size of real-world data samples, especially for types of hazards such as nuanced fainting. Synthetic image augmentation alleviated this, but a future version should include a larger dataset of over 2,000 annotated images per class. This required dealing with other constraints, including occasional false positives among dense and moderate crowds from the camera angles and lighting conditions. This raises the question of how the involved temporal video models, like LSTMs, can be beneficially integrated to reduce false positives. Second, the model compression and optimization were performed on deployment on Raspberry Pi since hardware is limited, leading to a trade-off between model speed and complexity.

References

[1] Al Jazeera. (2023) Largest Hajj pilgrimage in history begins in Saudi Arabia. Accessed: 29 April 2025. [Online]. Available: https://www.aljazeera.com/news/2023/6/25/largest-hajj-pilgrimage-in-history-begins-in-saudi-arabia

[2] I. Khan and R. D. McLeod, "Managing hajj crowd complexity: Superior throughput, satisfaction, health, and safety," *Kuwait Chapter of Arabian Journal of Business and Management Review*, vol. 2, no. 4, pp. 45–55, 2012, accessed: 29 April 2025. [Online]. Available: https://arabianjbmr.com/pdfs/KD_VOL_2_4/4.pdf

[3] PBS NewsHour. (2024) Hundreds died in intense heat during hajj pilgrimage in saudi arabia, officials say. Accessed: 29 April 2025. [Online]. Available: https://www.pbs.org/newshour/world/hundreds-died-during-hajj-pilgrimage-in-saudi-arabia-amid-intense-heat-officials-say

[4] Y. Salih and M. Simsim, "Visual surveillance for hajj and umrah: A review," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 3, no. 2, pp. 90–104, 2014, accessed: 29 April 2025.

[5] M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics: A review," *Cognitive Robotics*, vol. 3, pp. 54–70, 2023, accessed: 29 April 2025.

[6] SAFE Saudi Arabia for Elites. (2024) Saudi arabia's religious tourism reforms: A new era for pilgrims. Accessed: 29 April 2025. [Online]. Available: https://sa-fe.org/saudi-arabias-religious-tourism-reforms-a-new-era-for-pilgrims/

[7] The Guardian. (2024) At least 1,300 hajj pilgrims died during extreme heat, saudi arabia says. Accessed: 29 April 2025. [Online]. Available: https://www.theguardian.com/world/article/2024/jun/23/hajj-pilgrimage-death-toll-extreme-heat-mecca-saudi-arabia

[8] L. Rizk. (2024) Efficient crowd management during hajj: Challenges, innovations, and future directions. Accessed: 29 April 2025. [Online]. Available: https://repository.effatuniversity.edu.sa/entities/publication/c88c6d09-3f2d-4783-b85c-6c4bdb228531

[9] Saudi Ministry of Health. (2025) Health requirements and recommendations for travelers to saudi arabia for hajj – 1446h (2025). Accessed: 29 April 2025. [Online]. Available: https://www.moh.gov.sa/HealthAwareness/Pilgrims_Health/Documents/Hajj-Health-Requirements-English-language.pdf

[10] A. A. Shah, "Enhancing hajj and umrah rituals and crowd management through ai technologies: A comprehensive survey of applications and future directions," *IEEE Access*, vol. 12, pp. 161 820–161 841, 2024, accessed: 29 April 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10737309

[11] T. V. Hoang, "Impact of integrated artificial intelligence and internet of things technologies on smart city transformation," *Journal of Technical Education Science*, vol. 19, no. Special Issue 01, pp. 64–73, 2024, accessed: 29 April 2025.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[13] M. P. Véstias, "A survey of convolutional neural networks on edge with reconfigurable computing," *Algorithms*, vol. 12, no. 8, p. 154, 2019, accessed: 29 April 2025.

[14] M. Saqib, "Automatic analysis of crowd dynamics using computer vision and machine learning approaches," Ph.D. dissertation, University of Technology Sydney, 2019, accessed: 29 April 2025. [Online]. Available: https://opus.lib.uts.edu.au/handle/10453/135995

[15] A. Khan, J. A. Shah, K. Kadir, W. Albattah, and F. Khan, "Crowd monitoring and localization using deep convolutional neural network: A review," *Applied Sciences*, vol. 10, no. 14, p. 4781, 2020.

[16] N. Ilyas, A. Shahzad, and K. Kim, "Convolutional-neural network-based image crowd counting: Review, categorization, analysis, and performance evaluation," *Sensors*, vol. 20, no. 1, p. 43, 2020.

[17] G. Gao, J. Gao, Q. Liu, Q. Wang, and Y. Wang, "Cnn-based density estimation and crowd counting: A survey," arXiv preprint arXiv:2003.12783, 2020.

[18] V. Tetarwal and S. Kumar, "A comprehensive review on computer vision analysis of aerial data," arXiv preprint arXiv:2402.09781, 2024.

[19] N. A. Nawaz, M. Abaidullah, A. Abid, M. Al-Mukarmah, and S. Arabia, "Big data framework for crowd monitoring in large crowded events," *UMT Artificial Intelligence Review*, vol. 3, no. 1, pp. 21–23, 2023.

[20] L. Judijanto, "Risk management and pilgrim safety: A comprehensive strategy for organising hajj and umrah," *International Journal of Financial Economics*, vol. 1, no. 2, pp. 408–417, 2024. [Online]. Available: https://ijefe.my.id/index.php/economic/article/view/39

[21] G. Simuni, M. Sinha, R. S. Madhuranthakam, and G. Vadlakonda, "Edge computing in iot: Enhancing real-time data processing and decision making in cyber-physical systems," *International Journal of Unique and New Updates*, vol. 6, no. 2, pp. 75–84, 2024, iSSN: 3079-4722. [Online]. Available: https://ijunu.com/index.php/journal/article/view/60

[22] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," *Computer Networks*, vol. 130, pp. 94–120, 2018.

[23] T. Alafif, M. Jassas, A. E. Abdel-Hakim, G. Alfattni, H. Althobaiti, M. Ikram, A. Alharbi, H. Alsharif, M. Alshamrani, E. Alharbi, and T. Alsubait, "Intelligent framework for crowd control and management in large-scale gatherings," *Unpublished or Preprint*, 2025, forthcoming publication or preprint (details not available).

[24] A. Koubâa, B. Qureshi, M. F. Sriti, A. Allouch, Y. Javed, M. Alajlan, O. Cheikhrouhou, M. Khalgui, and E. Tovar, "Dronemap planner: A service-oriented cloud-based management system for the internet-of-drones," *Ad Hoc Networks*, vol. 86, pp. 46–62, 2019.

[25] A. Koubaa, A. Ammar, M. Abdelkader, Y. Alhabashi, and L. Ghouti, "Aero: Ai-enabled remote sensing observation with onboard edge computing in uavs," *Remote Sensing*, vol. 15, no. 7, p. 1873, 2023.

[26] AFP. (2025) Maha kumbh 2025: India uses ai to stop stampedes at world's biggest gathering. Accessed: 29 April 2025. [Online]. Available: https://economictimes.indiatimes.com/tech/artificial-intelligence/maha-kumbh-2025-india-uses-ai-to-stop-stampedes-at-worlds-biggest-gathering/articleshow/117345693.cms

[27] S. C. Vetrivel, P. Vidhyapriya, and V. P. Arun, "Ai-driven solutions for crowd management in tourism: Navigating the swarm," in *AI Technologies for Personalized and Sustainable Tourism*. IGI Global, 2025, pp. 83–112. [Online]. Available: https://www.igi-global.com/chapter/ai-driven-solutions-for-crowd-management-in-tourism/359252

[28] M. K. Y. Shambour and M. A. Abu-Hashem, "Analysing lecturers' perceptions on traditional vs. distance learning: A conceptual study of emergency transferring to distance learning during covid-19 pandemic," *Education and Information Technologies*, pp. 1–21, 2022.

[29] J. Zhang and L. Wu, "Influence of human population movements on urban climate of beijing during the chinese new year holiday," *Scientific Reports*, vol. 7, no. 1, p. Article 45813, 2017. [Online]. Available: https://www.nature.com/articles/srep45813

11 Appendices

11.1 Appendix A: Python Code for AI Model Training and Prediction

```python
import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
import os

test_path = "/Users/reema/Downloads/test"
for category in sorted(os.listdir(test_path)):
    category_path = os.path.join(test_path, category)
    if os.path.isdir(category_path):
        num_images = len([f for f in os.listdir(category_path) if f.lower().endswith(('png',
'jpg', 'jpeg'))])
        print(f"{category}: {num_images} images")

# Data Augmentation for training
train_datagen = ImageDataGenerator(
    rescale=1.0/255,
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
test_datagen = ImageDataGenerator(rescale=1.0/255)

train_generator = train_datagen.flow_from_directory(
    '/Users/reema/Downloads/train',
    target_size=(150, 150),
    batch_size=32,
    class_mode='categorical'
)

test_generator = test_datagen.flow_from_directory(
    '/Users/reema/Downloads/test 2',
    target_size=(150, 150),
    batch_size=32,
    class_mode='categorical'
)
print("Detected test classes:", test_generator.class_indices)

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(7, activation='softmax')  # 7 classes
])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // train_generator.batch_size,
```

```
    epochs=10,
    validation_data=test_generator,
    validation_steps=test_generator.samples // test_generator.batch_size
)

test_loss, test_acc = model.evaluate(test_generator)
print(f"Test Accuracy: {test_acc * 100:.2f}%")

plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.title('Training and Validation Accuracy')
plt.show()

# Single image prediction
img_path = '/Users/reema/Downloads/test 2/fire/fire1.jpeg'
img = image.load_img(img_path, target_size=(150, 150))
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0) / 255.0
predictions = model.predict(img_array)
class_names = list(train_generator.class_indices.keys())
predicted_class = class_names[np.argmax(predictions)]
print(f"Predicted Event: {predicted_class}")

# Add class weights
class_weights = {0: 1.0, 1: 2.0, 2: 3.0, 3: 3.0, 4: 4.0, 5: 5.0}
model.fit(train_generator, epochs=10, class_weight=class_weights)

# Predicting a new image
img_path = '/Users/reema/Downloads/test 2/moderate_crowd/Screenshot 1446-08-10 at 9.58.41
    PM .png'
img = image.load_img(img_path, target_size=(150, 150))
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0) / 255.0
predictions = model.predict(img_array)
class_names = list(train_generator.class_indices.keys())
predicted_class = class_names[np.argmax(predictions)]
print(f"Predicted Event: {predicted_class}")

# Conditional alerting
if predicted_class == 'dense_crowd':
    print("ALERT: Dense crowd detected! Take immediate action.")
elif predicted_class == 'moderate_crowd':
    print("Notice: Moderate crowd detected.")
elif predicted_class == 'sparse_crowd':
    print("Sparse crowd detected. All clear.")
elif predicted_class == 'heat_stroke':
    print("ALERT: Heat stroke signs detected! Send medical assistance.")
elif predicted_class == 'fainting':
    print("ALERT: Fainting detected! Provide first aid.")
elif predicted_class == 'stampede':
    print("ALERT: Stampede risk! Take crowd control measures.")
elif predicted_class == 'fire':
    print("ALERT: Fire detected! Deploy firefighting resources immediately.")

# Save and reload model
from tensorflow.keras.models import load_model
model.save('/Users/reema/saved_models/model.h5')
model = load_model('/Users/reema/saved_models/model.h5')

# Batch test predictions
test_images_dir = '/Users/reema/Downloads/test 2'
image_paths = []
for class_name in os.listdir(test_images_dir):
    class_folder = os.path.join(test_images_dir, class_name)
    if os.path.isdir(class_folder):
```

```python
        for filename in os.listdir(class_folder):
            if filename.endswith(('.png', '.jpg', '.jpeg')):
                image_paths.append(os.path.join(class_folder, filename))

for img_path in image_paths:
    img = image.load_img(img_path, target_size=(150, 150))
    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0) / 255.0
    predictions = model.predict(img_array)
    class_names = list(train_generator.class_indices.keys())
    predicted_class = class_names[np.argmax(predictions)]
    plt.imshow(img)
    plt.axis('off')
    plt.title(f"Predicted Event: {predicted_class}")
    plt.show()
    print(f"Image: {img_path} | Predicted Event: {predicted_class}")

# Confirm class name by index
img_array = image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0) / 255.0
class_names = list(train_generator.class_indices.keys())
predicted_class_index = class_names.index(predicted_class)
print(f"Predicted Event: {class_names[predicted_class_index]}")
```

Listing 1: Python Script for Model Training, Evaluation, and Deployment

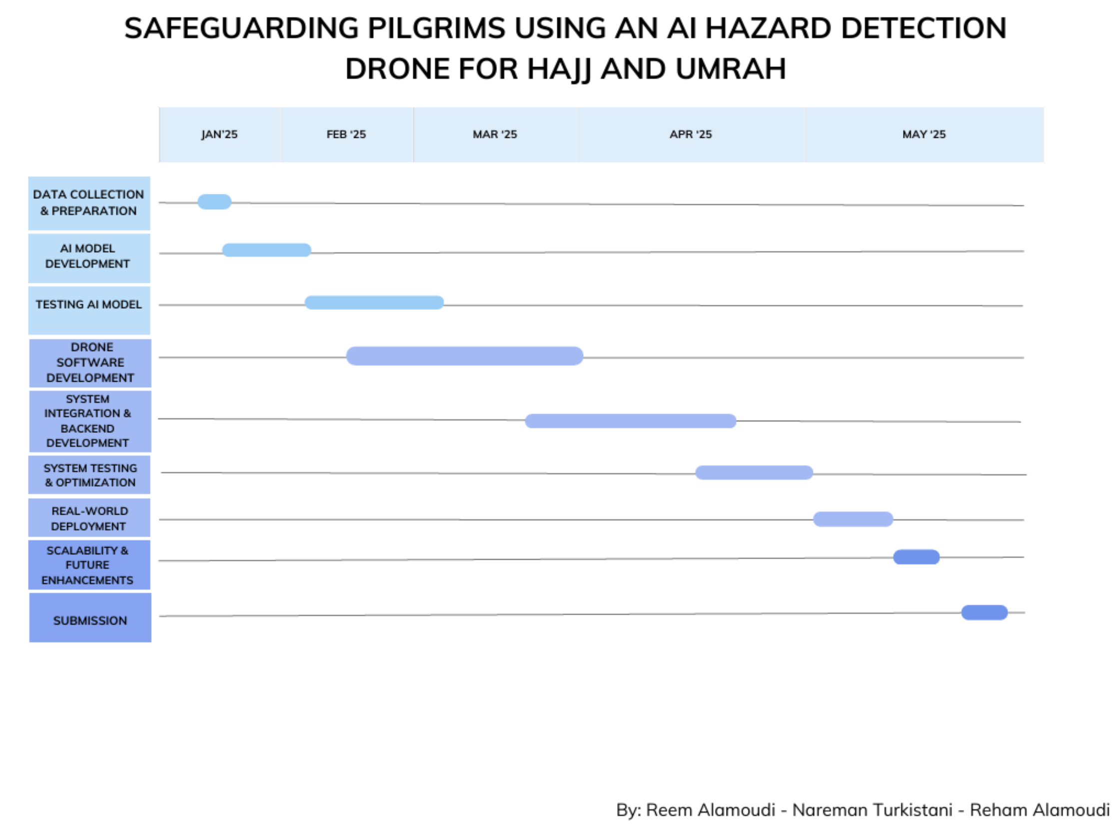


Figure 16: TimeLine For the Project