RIMC Volume 2, Issue 2, December 2025 | http://dx.doi.org/10.21622/RIMC.2025.02.2.1414
Journal of Robotics: Integration, Manufacturing & Control - eISSN 3009-7967

Rescue-Bots: A proposed Multi-robot Architecture for Rescue Missions

Yehia Abdulghafar [1], Feras Hamdan [2], Husain AlQuraini [3], Ali Bohamad [4], Ahmed AlRokhami [5], and Yehia Kotb [6*]

[1,2,3,4,5,6*] American University of the Middle East, Egila, Kuwait.

eng.yehia_abdelghafar@hotmail.com, ferasohamdan@gmail.com, husain.alquraini@gmail.com, alihamadbohamad@gmail.com, a.alrokhami@gmail.com, yehia.kotb@aum.edu.kw

Received: 03 June 2025 Accepted: 18 November 2025 Published: 11 December 2025

Abstract

The growing frequency of natural disasters and armed conflicts has created an urgent need for rapid, reliable, and autonomous rescue solutions, particularly in situations where traditional methods become inefficient or unsafe for human responders. This work presents the design and implementation of a fully autonomous rescue system aimed at detecting survivors and delivering immediate assistance without exposing rescue personnel to risk. The system integrates three main components: an aerial drone, a control center, and a ground vehicle. The drone performs autonomous search operations and transmits detected survivor locations to the control center, which handles decision-making and dispatches commands to the ground vehicle. The vehicle then navigates to the identified location to provide essential aid. A comprehensive high-level and low-level design is developed, detailing the system architecture, detection algorithm, communication framework, and hardware components. The implementation of the drone platform, ground vehicle, pre-trained detection model, and inter-device communication is presented based on this design. The system undergoes multiple tests evaluating drone search patterns, communication reliability, and detection performance. Results demonstrate accurate human detection and effective guidance of the ground vehicle to target locations, confirming the feasibility and robustness of the proposed autonomous rescue solution.

Key-words: Autonomous, Rescue System, Self-Centralized, Low-Level Design, High-Level Design, Pre-trained Model

I. Introduction

In recent times, the world has experienced significant loss of lives due to a combination of natural disasters[3] and armed conflicts[4,5]. A notable example is the devastating earthquake that impacted Turkey and Syria. This event resulted in the deaths of approximately 50,000 people, with many more injured. Additionally, tens of thousands of individuals were missing, and over 100,000 have been displaced, facing a lack of shelter. Figure 1 depicts the aftermath of the earthquake in Turkey and Syria, highlighting the destruction [1].

* Corresponding Author

Figure 1: Aftermath of the Turkey–Syria earthquake, illustrating large-scale structural destruction and debris. Adapted from [1].

Additionally, Figure 2 depicts the escalating trend of natural disasters year by year, attributed to global warming and other sources of climate change. It also categorizes the types of disasters [2].

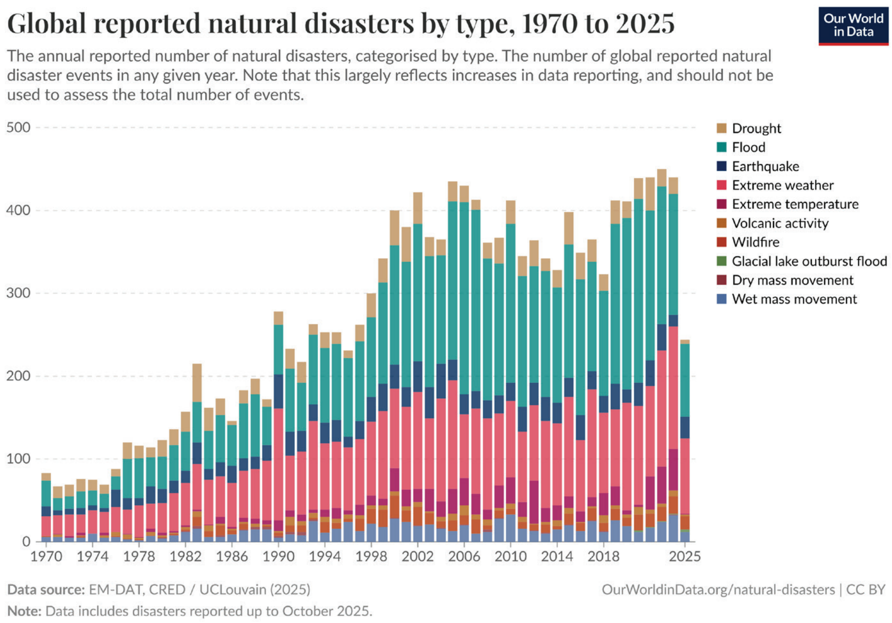


Figure 2: Reported natural disasters by type from 1970 to 2023, showing long-term variation in disaster frequency. Data sourced from [2].

To address the increasing number of natural and human-made disasters, it's important to recognize the urgency caused by rising global temperatures. This will likely face many challenges from these disasters, and it's crucial to find ways to reduce the number of people harmed or killed.

Modern technology plays a crucial role in disaster management [6-10]. Its primary strength lies in its ability to spread awareness and knowledge. This report focuses on leveraging technological advancements to reduce human casualties and losses in disaster situations. Recent developments in artificial intelligence [11,12] and robotics [13,14] have significantly enhanced the disaster response capabilities [15,16]. These technological innovations offer substantial promise in transforming search and rescue operations, providing hope in dire situations. Integrating advanced AI systems into disaster management procedures is a promising strategy for saving lives and mitigating the effects of these disastrous events.

In response to urgent and challenging situations, this study's primary goal is to protect lives at risk from being trapped under collapsed structures, lost in vast deserts, or stranded in harsh terrains. Utilizing advanced technology, it was proposed to develop a robot capable of locating missing individuals and transmitting their location. This approach enables effective tracking and rescue, either by the robot directly or through the automatic and adaptive dispatch of necessary assistance, thus reducing the need for human involvement. This innovative approach is central to this research and has the potential to save not only the lives of rescue personnel but also those of the survivors.

In Table 1, it can be seen that, in disaster response, humans and robots each have their own strengths and weaknesses. Humans are adaptable and experienced, but they face risks in dangerous environments and get tired. They may also struggle in tight spaces. Robots, on the other hand, can work continuously without getting tired and can go into risky areas. They can be equipped with special tools. However, they may have trouble quickly changing situations and rely on programming for decisions. Humans are good at communication, while robots use devices. By combining human skills with robot capabilities, it can improve disaster response efforts.

Table 1: Comparison of human and robotic capabilities in rescue missions, highlighting key strengths and limitations of each.

Aspect	Humans	Robots
Risk to Rescuers	High risk in hazardous environments	Low risk, can navigate dangerous areas
Physical Limitations	Limited, fueled by strength and endurance	No physical limitations, operates continuously
Adaptability	Can quickly adapt to changing situations	May face challenges in rapidly changing environments
Specialized Capabilities	Limited by human capabilities	Can be equipped with specialized sensors and tools
Decision-Making	Based on experience, intuition, and training	Rely on programming and sensor data for decision-making
Access to Confined Spaces	Limited by size and physical constraints	Can navigate tight or small spaces
Remote Operation	Not applicable	Can be operated remotely
Surveillance	Visual and auditory senses	Equipped with cameras for visual information
Communication	Verbal and non-verbal communication	Equipped with communication devices

II. Design development

A. *Proposed design*

The proposed solution is a self-centralized autonomous rescue system consisting of a drone, a central unit, and a ground vehicle. The drone will continuously stream its camera feed and location data to the central unit, which acts as the operation's brain. This central unit will receive data from the drone, perform decision-making, and conduct object recognition on the drone's stream. If a survivor is detected, the central unit will send the survivor's location to the ground vehicle, which will then proceed to provide the necessary aid. Utilizing UAVs and AI significantly enhances capabilities in post-disaster scenarios beyond human abilities. Consequently, the disaster response time will be reduced, and the probability of rescuing survivors will be much higher.

Solution requirements and criteria:

1. **Computer Vision System:** The drone must have an advanced computer vision system to identify people effectively in various conditions.

2. **Navigation and Autonomy:** Both the drone and the ground vehicle require sophisticated navigation systems for autonomous operation in unpredictable areas.

3. **Communication:** Robust communication is essential between the drone, vehicle, and base station, capable of efficient data transfer even in low connectivity areas.

4. **Safety and Reliability:** The system should include fail-safe mechanisms and thorough safety testing to ensure reliability and prevent accidents.

5. **Natural Language Processing:** Advanced natural language processing is needed for clear communication with and assessment of found individuals.

6. **Operational Endurance:** The drone and vehicle should have long battery life and energy efficiency for extended missions.

7. **Simplified Controls:** The system should feature straightforward and easy controls for efficient operation and quick response by the team.

8. **Scalability and Adaptability:** Design should be scalable and adaptable for future technological integrations.

Solution constraints:

1. **Financial Constraints**: Operating within a limited budget restricts the ability to acquire top-tier equipment and software, potentially impacting the overall effectiveness of the rescue system.

2. **Technological Limitations**: The drones and autonomous vehicles have inherent limitations in terms of weight capacity, battery life, and physical dimensions, which can limit their operational capabilities.

3. **Regulatory Compliance**: Adherence to existing laws and regulations governing the use of drones and autonomous vehicles is mandatory, which can complicate deployment in certain areas.

4. **Environmental Challenges**: Diverse weather conditions, difficult terrain, and natural obstacles pose significant challenges to the efficiency and success of rescue missions.

5. **Communication Barriers**: Limited or unreliable internet and phone connectivity in certain areas can hinder communication with the drones.

6. **Interaction with Survivors**: Challenges in interacting with survivors, including language barriers, fear, and injuries, can complicate rescue efforts.

7. **Public Perception and Trust**: Gaining public acceptance and trust is a challenge due to concerns about privacy, safety, and reliance on technology in life-saving scenarios.

8. **Operational Constraints**: The physical weight and size of the drones and vehicles may restrict their access and utility in certain environments, particularly in confined areas.

B. *Detailed high-level specifications*

After starting the operation, the central control unit (PC server) establishes communication with both the drone and vehicle to enable the transmission and reception of requests. Initially, both the drone and vehicle are instructed to calibrate their current locations as the home position $(X, Y, Z = 0, 0, 0)$. Afterwards, the vehicle enters a standby mode, awaiting further instructions. Simultaneously, the drone is tasked to perform reconnaissance within a 300m2 area in the vicinity of the central unit. During this operation, it continuously streams video back to the central unit along with real-time positional data (X, Y, Z). As illustrated in Figure 3.

The central unit, equipped with AI image analysis capabilities, specifically utilizes a Convolutional Neural Network (CNN) algorithm [17], processes the incoming video stream by capturing a snapshot every 20 seconds and analyzing these images. Upon identifying a survivor within these images, the precise coordinates are transmitted to both the drone and vehicle. In response, the drone may either broadcast a prerecorded message or engage its speaker system as determined by the situation, then maintain a hovering position near the identified survivor to scout for additional survivors.

At the same time, the vehicle is navigating through the terrain towards the survivor's location. The vehicle will either wait for a preset duration or visual confirmation via onboard cameras to ensure the survivor has entered it. Following the successful entry of the survivor, both the drone and vehicle are instructed to return to their designated home position (0, 0, 0).

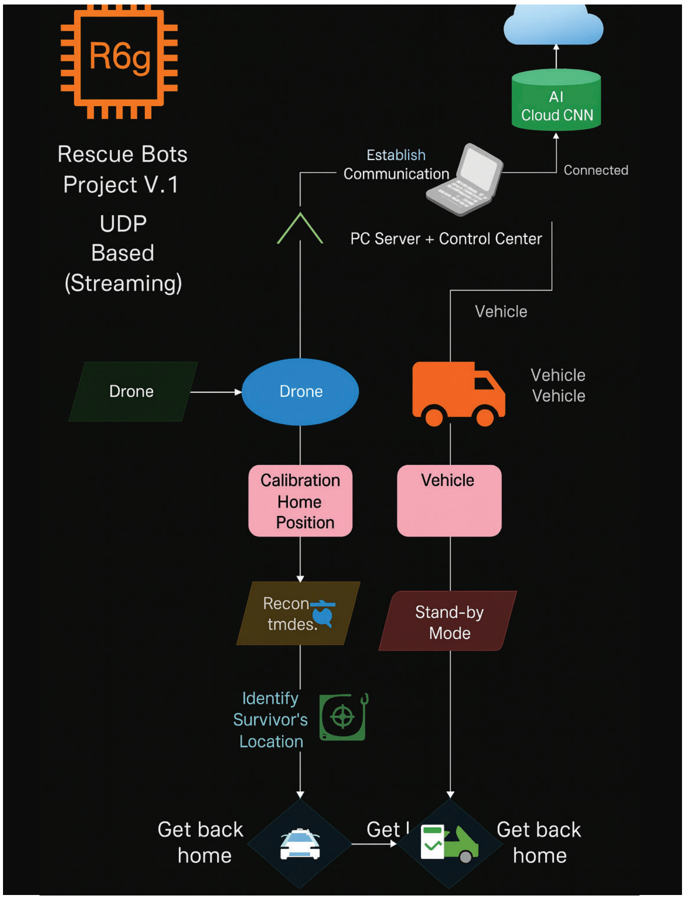


Figure 3: High-level architecture of the autonomous rescue system, showing interactions among the drone, central unit, and ground vehicle.

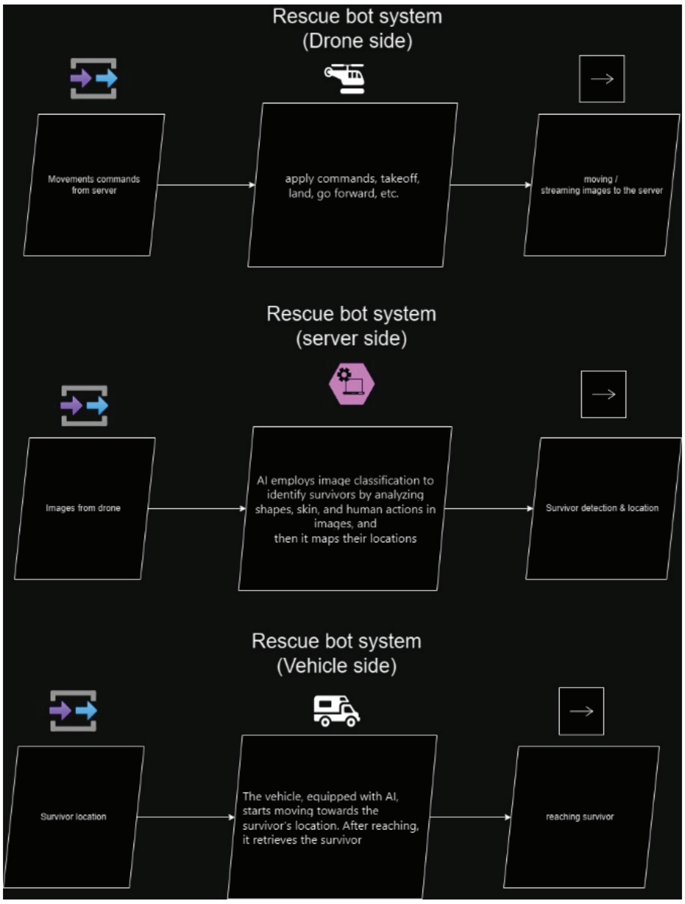


Figure 4: Input–output diagram illustrating communication flows among the server, drone, and ground vehicle.

The vehicle system is designed around four key components: a Raspberry Pi 4, a webcam, a motor driver, and a pair of DC motors, as shown in Fig. 4. The Raspberry Pi 4 serves as the central processing unit, orchestrating the vehicle's overall functionality. It operates as the main controller, managing both the reception and transmission of commands to and from the central server.

Communication with the server is facilitated through a secure SSH port, enabling the Raspberry Pi to receive coordinates and other operational commands. Upon receipt of these coordinates, the Raspberry Pi directs the vehicle, through the integration of the motor driver and DC motors, to navigate towards the designated survivor's location, as shown in Fig. 5. This motor assembly is crucial for controlling the vehicle's movements, allowing for precise adjustments in direction and speed as required.

Once the vehicle arrives at the specified region or coordinates, the webcam camera, acting as an optical sensor, is activated to commence the search for the nearest survivor. The camera, in conjunction with image processing algorithms running on the Raspberry Pi, scans the area to detect and identify any individuals in distress. This combination of hardware and software enables the vehicle to fulfill its mission of reaching and assisting survivors effectively.

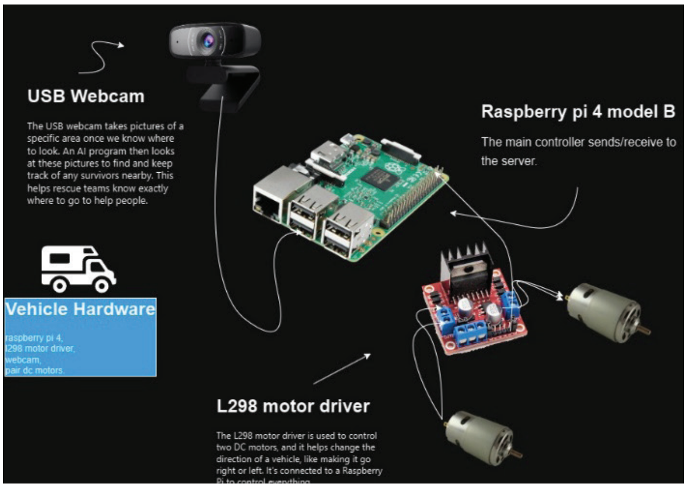


Figure 5: Hardware configuration of the ground vehicle, including the Raspberry Pi 4, motor driver, DC motors, and onboard camera.

The drone constitutes a central component of the rescue system, operating as the primary field unit responsible for environmental scanning and real-time data acquisition, as demonstrated in Fig. 6. It performs systematic aerial reconnaissance across the designated area and transmits continuous image streams to the server for processing and survivor identification. The drone executes flight commands issued by the control server while simultaneously returning visual data, thereby forming an essential link in the overall search-and-rescue workflow.

The drone's software architecture integrates autonomous navigation, visual detection, and communication functions into a unified operational framework. It establishes a stable connection with the aerial platform, manages live video acquisition, and employs advanced image-recognition algorithms to detect potential survivors. Upon detection, the system activates predefined alert mechanisms and initiates coordination procedures with the ground vehicle. The drone's programmed flight pattern enables systematic coverage of the search area, while the detection subsystem provides continuous analysis of captured frames to identify human figures or distress signals.

The ground vehicle is controlled through a dedicated software module designed to execute remote navigation commands securely. Communication is established via an SSH-based protocol that enables the transmission of movement instructions and mission parameters to the vehicle's onboard processor. The vehicle navigates toward specified coordinates while maintaining communication with the control server and subsequently activates its own detection procedures upon arrival at the target location. This software framework ensures reliable coordination between the aerial and ground units and supports the system's overall goal of autonomous survivor localization and assistance. This logic is exemplified in Appendices 1 and 2 by a flowchart.

Figure 6: Drone platform used for aerial imaging, showing the mounted camera and live-streaming components.

III. Realization & performance optimization

A. Planned implementation and experiments

In this innovative work, the drone plays a crucial role as a primary tool for survivor detection. It is equipped with a camera whose angle can be adjusted from 0 to 90 degrees, as illustrated in Figure 7, to provide a comprehensive bird's eye view for horizontal detection. Additionally, a specially designed vehicle, powered by a Raspberry Pi 4 and programmed using Python, acts as a secondary element. This vehicle is responsible for retrieving the survivor and transporting them to the rescue station. To implement this modification, the drone's casing was opened, the original camera connection was detached, and the camera was repositioned vertically to obtain a downward-facing, bird's-eye view. This configuration was required to accommodate the YOLO-based detection method [19] and the characteristics of the custom dataset used.

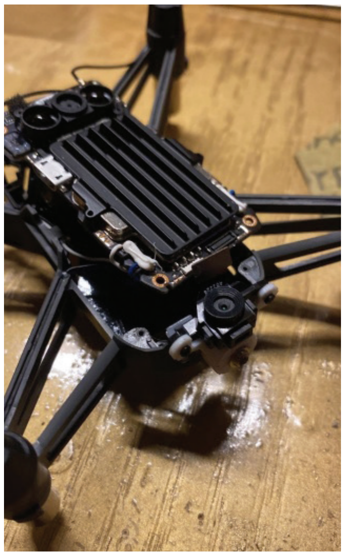


Figure 7: Modified drone prototype with a repositioned downward-facing camera to enable vertical image capture for YOLO-based detection.

The operational plan for the search and rescue system seamlessly integrates three pivotal devices: the drone *in Figure 8,* the vehicle in *Figure 9*, and a central server, each playing a critical role. Upon initiating the Python script on the server, automatic connections are established with both the drone and vehicle via a LAN network. The drone then takes off, embarking on a search within a specific randomized area, utilizing computer vision techniques. Leveraging advanced technologies like artificial intelligence and convolutional neural networks (CNN), the drone intelligently identifies the survivor's location, sending the images back to the server for further analysis.

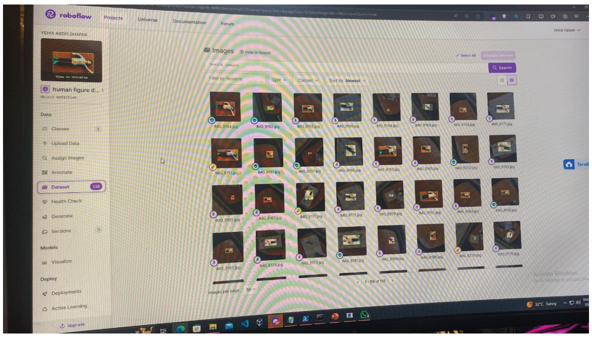
Figure 8: Final drone prototype used in field testing, incorporating the integrated camera and communication modules.

On the server side, advanced image-processing operations are executed to interpret the visual data captured by the mobile application. The incoming frames are analyzed using the YOLOv8 object-detection model, which provides rapid and highly accurate classification of pedestrians and other relevant scene elements. This stage relies on a custom dataset developed through the Roboflow platform to ensure robust detection performance. The subsequent section presents a detailed description of the dataset construction process, including annotation procedures, augmentation strategies, and quality-control measures. It also outlines the complete training pipeline implemented on Google Colab, covering model configuration, hyperparameter optimization, and evaluation methodologies.

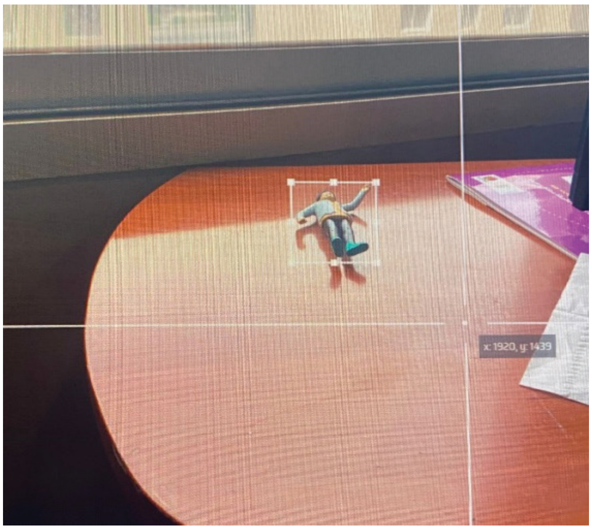
Figure 9: Ground vehicle prototype equipped with a Raspberry Pi 4, motor system, and optical sensing module.

Once the survivor's coordinates are pinpointed, the server promptly transmits this information to the vehicle. Embedded within the vehicle is a Raspberry Pi, which, upon receiving the coordinates via an SSH port, executes another Python script. This enables the vehicle to autonomously navigate towards the survivor, facilitate their rescue, and safely return to base. This sophisticated interplay between the drone, vehicle, and server exemplifies a highly interdisciplinary approach, merging robotics, computer vision, and artificial intelligence to significantly enhance the efficacy of search and rescue operations.

In this paper, the YOLOv8 deep-learning model is employed as the primary method for survivor detection. A dedicated dataset consisting of 118 images of a survivor figure, as shown in Figure 10, was developed to ensure robust model performance. Dataset construction followed a structured workflow: images were first captured under varied angles and environmental conditions to improve generalization capability; subsequently, precise annotations were performed using the Roboflow platform. Model training was conducted on Google Colab, utilizing its available GPU resources to accelerate computation and optimize learning efficiency. Following training, the model underwent extensive evaluation to verify performance in realistic operational scenarios. The overall pipeline—from dataset collection and annotation to training and validation—was computationally demanding yet essential for achieving high accuracy and reliability in survivor detection.



(a)



(b)

Figure 10: Annotation process used in dataset preparation: (a) sample of annotated training images; (b) example of a single annotated survivor instance.

Figure 10 displays a variety of images annotated with different poses, ensuring both accuracy and diversity in the dataset. This approach enhances the model's ability to recognize the target object survivors, in this case, across a wide range of scenarios and positions. By incorporating multiple poses, the dataset effectively trains the deep-learning model to identify survivors in various conditions and orientations, significantly improving the model's robustness and performance in real-world applications. Figure 10 shows the overall annotation process and dataset results, including Dimension Insights and the total number of trained and annotated images. This visualization offers key insights into the diversity of image sizes and the scale of data preparation, crucial for assessing the dataset's robustness and effectiveness in training the model.

Figure 11 is valuable for assessing the quality and comprehensiveness of the dataset used for training machine learning models, ensuring that the model is trained on well-rounded and representative data.

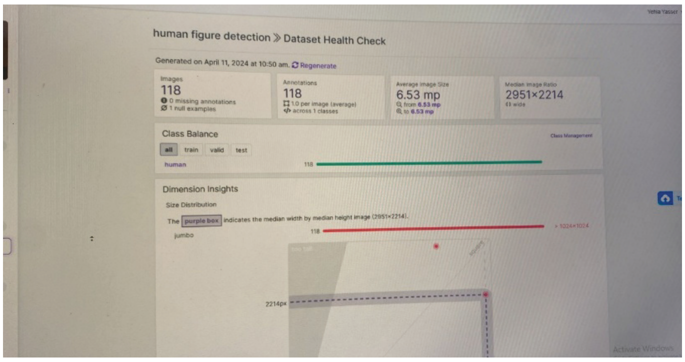


Figure 11: Dataset statistics generated through Roboflow, including image dimensions, class distribution, and annotation completeness.

B. *Design analysis and feedback*

Comprehensive testing has been conducted on the prototype to ensure it meets all the requirements. The first set of tests focused on key aspects of the system:

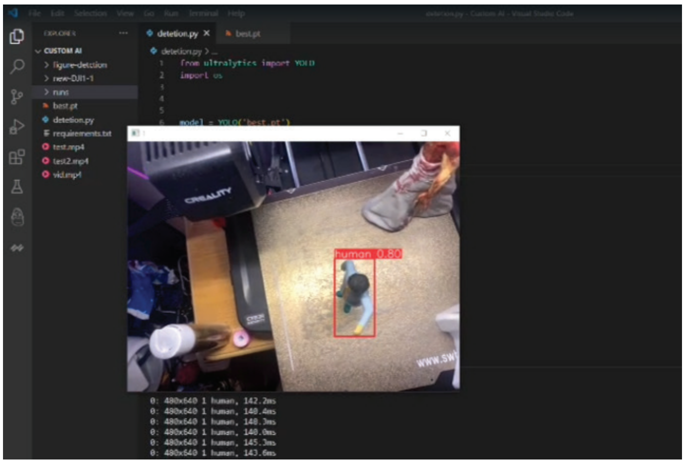
Quick response is crucial in rescue operations. Extensive testing ensured the system's quick response to disasters. As a self-centralized rescue system, decision-making regarding the survivor's location and the vehicle's movement initially caused delays. To address this, improvements in detection speed and vehicle pathing were tested as shown in Figure 12.

The prototype uses a map, so it was essential to ensure the drone's search paths stayed within the map boundaries. Multiple tests confirmed the drone remained within these borders and thoroughly searched every necessary section of the map.

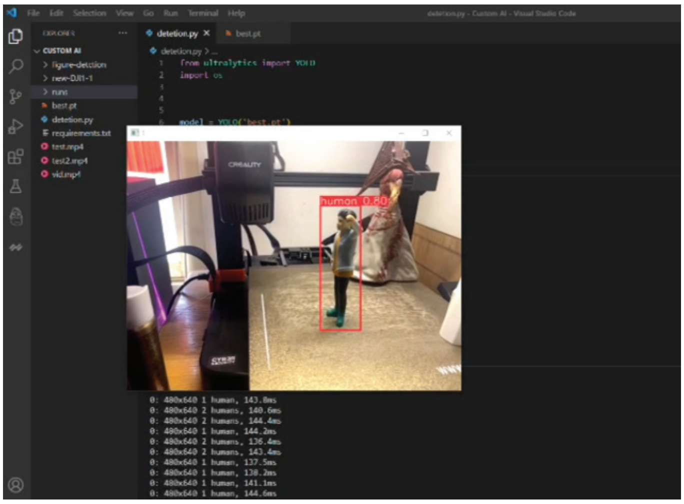
Comprehensive tests checked for issues in communication between the drone, server, and vehicle. Centralized systems can fail at a single point, so the server's operation must be robust to prevent system-wide failures. The prototype was tested under connection obstacles, such as crowded connections, to assess how much connection disruption it could handle before failing.

Figure 12: Experimental setup used to evaluate drone detection accuracy, communication latency, and vehicle navigation.

The drone was put in a series of trials to check the accuracy of survivor detection in various environments, as shown in Figure 13. The vehicle, using the same pre-trained model but detecting survivors from different angles, was also tested. These tests assessed how the camera angles of both the drone and the vehicle, as well as different poses of the survivor on the map, affected detection accuracy.



(a)



(b)

Figure 13: Experimental detection results for two test scenarios: (a) trial 1 showing successful aerial detection; (b) trial two under different survivor pose and lighting conditions.

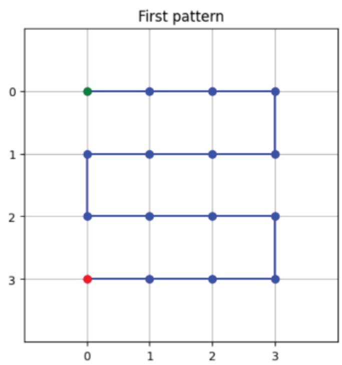
C. Design optimization and improvements

Using Yolov9 significantly improved detection speed. However, vehicle pathing still needs refinement. Testing various pathing methods revealed that directly approaching the survivor is the fastest, but this can be problematic if obstacles are present, since the vehicle lacks a collision avoidance system. Further tests are needed to develop a reliable path assessment.

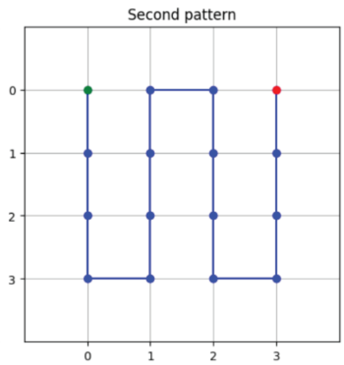
After extensive testing, the fourth pattern proved the fastest for finding survivors and returning home, especially since the map is not square. Further tests are needed to integrate SLAM [22-23], which would eliminate the need for predefined search patterns in a localized map.

Testing various communication methods showed that a LAN network connecting the vehicle and server to the central unit was the most stable. However, interference from multiple people using Wi-Fi or cellular networks can disrupt communication, necessitating further testing with alternative methods.

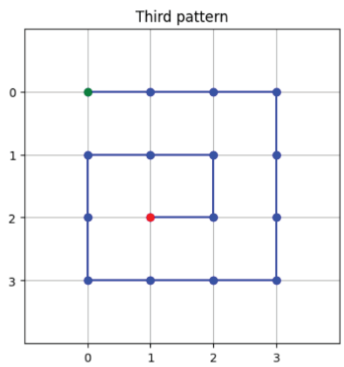
Several algorithms were tested as demonstrated in Figure 14, including pretrained Yolov8, a pretrained TensorFlow CNN model, custom-trained Yolov8, and a custom-trained TensorFlow CNN. The custom-trained Yolov8 was superior for the drone's stream, offering great accuracy and detection speed. For the vehicle, the pretrained TensorFlow model performed better due to the Raspberry Pi 4 limitations. Further testing on different algorithms and datasets is required.



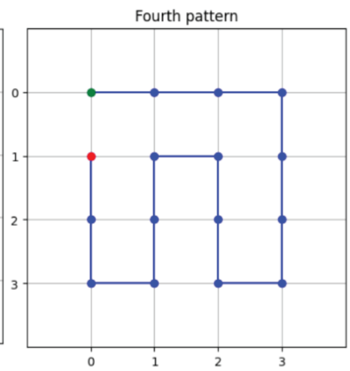
(a)



(b)



(c)



(d)

Figure 14: evaluated search strategies for drone reconnaissance: (a) row-by-row search; (b) column-by-column search; (c) outside-inward pattern; (d) hybrid approach where the drone scans the first column followed by row-by-row coverage.

IV. Conclusion & Future works

In conclusion, a self-centralized autonomous rescue system has been developed, consisting of an aerial drone, a central control unit, and a ground vehicle. The drone performs reconnaissance, identifies potential survivors, and transmits their coordinates to the control center, which subsequently directs the ground vehicle to the detected location. A custom YOLOv8 model is employed for detecting survivors or visual signals indicating a need for assistance. The ground vehicle is equipped with a Raspberry Pi 4, providing the computational capability necessary for real-time processing and navigation.

Future enhancements include the integration of drone swarms to improve search coverage and overall system efficiency. Additional sensors will be incorporated to enhance detection performance under poor visibility or challenging environmental conditions. Furthermore, the use of Simultaneous Localization and Mapping (SLAM) is planned to enable operation in previously unmapped or unknown environments, allowing the drone to construct and navigate a map in real time.

V. Appendix

Appendix1:

Autonomous drone rescue system flowchart for YOLO human detection.

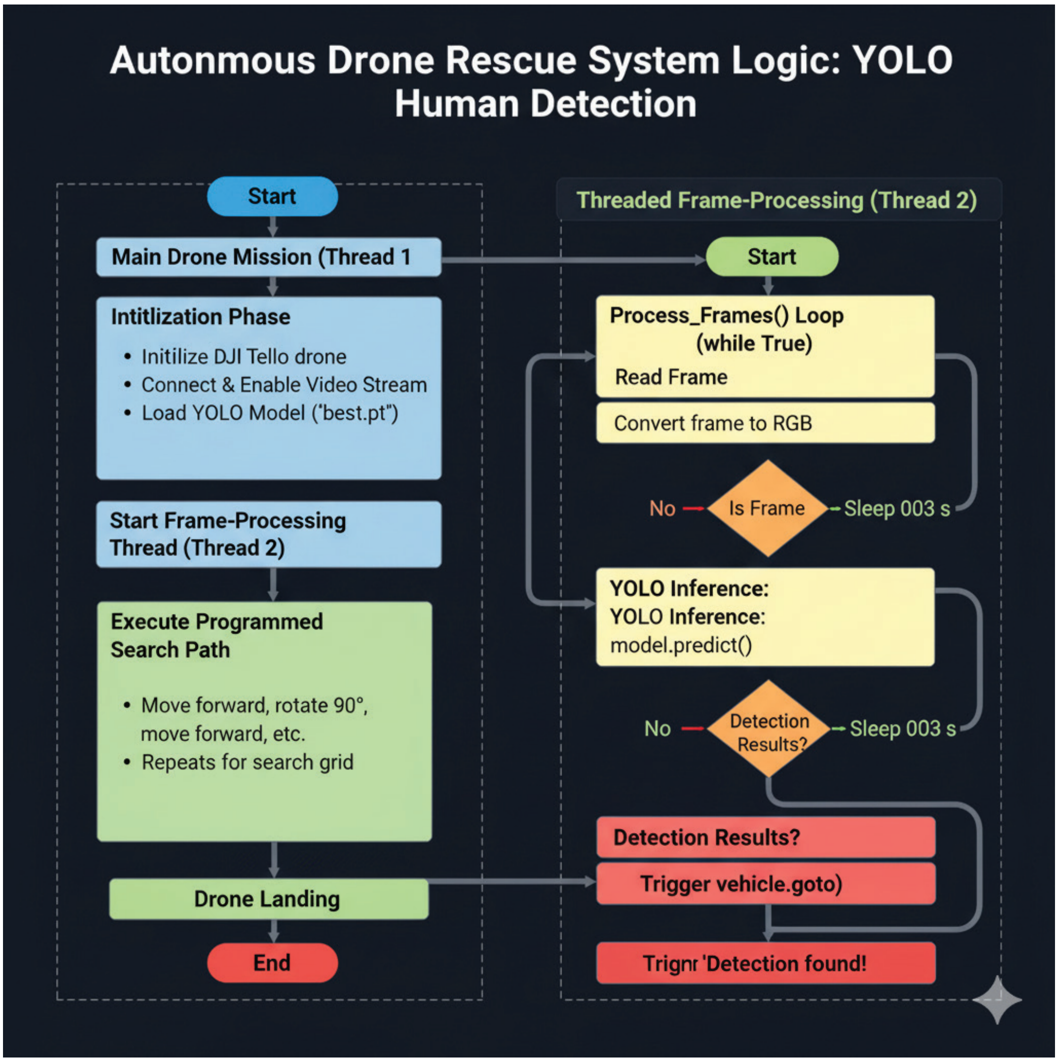


Figure A1

Appendix 2:

Raspberry Pi Ground Vehicle flowchart for pedestrian detection and avoidance.

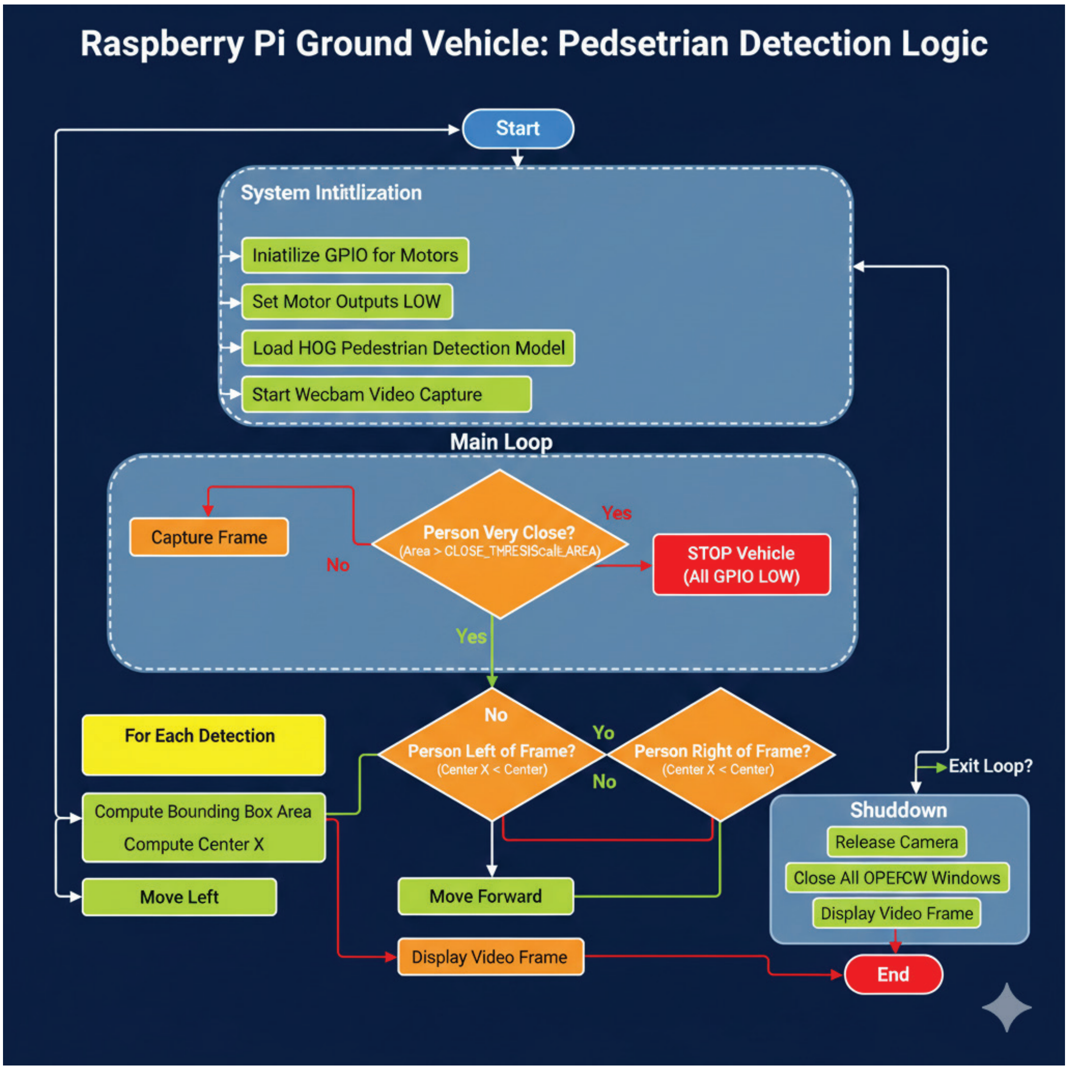


Figure A2

REFERENCES

[1] I. S. E. K. C. Liakos Amarachi Orie, "Death toll climbs to 33,000 people in Turkey-Syria earthquake," Nov. 2023. [Online]. Available: https://edition.cnn.com/2023/02/12/middleeast/deaths-turkey-syria-earthquake-intl/index.html

[2] "Global reported natural disasters by type." [Online]. Available: https://ourworldindata.org/grapher/natural-disasters-by-type

[3] B. R. Cross, "Turkey and Syria earthquakes: latest news," Nov. 2023, *British Red Cross*. [Online]. Available: https://www.redcross.org.uk/stories/disasters-and-emergencies/world/turkey-syria-earthquake

[4] K. McIntosh, "Deaths from global conflicts hit 30-year high, since 2021," *The Guardian*, 2024.

[5] J. Erdemir, M. Ghassany, N. Fernandez, and et al, "The impact of wars and natural disasters on the emergence of communicable diseases," *Ann Clin Microbiol Antimicrob*, vol. 22, no. 1, 2023.

[6] M. Sakurai and Y. Murayama, "Information technologies and disaster management – Benefits and issues -," *Progress in Disaster Science*, vol. 2, p. 100012, Jul. 2019, doi: 10.1016/j.pdisas.2019.100012.

[7] B. Mukhopadhyay, "Use of Information Technology in Emergency and Disaster Management," *American Journal of Environmental Protection*, vol. 4, no. 2, p. 101, 2015, doi: 10.11648/j.ajep.20150402.15.

[8] Y. Murayama, H. J. Scholl, and D. Velev, "Information Technology in Disaster Risk Reduction," *Information Systems Frontiers*, vol. 23, no. 5, pp. 1077–1081, Sep. 2021, doi: 10.1007/s10796-021-10204-x.

[9] R. Gaire *et al.*, "Internet of Things (IoT) and Cloud Computing Enabled Disaster Management," 2018. [Online]. Available: https://arxiv.org/abs/1806.07530?utm

[10] A. Adeel *et al.*, "A Survey on the Role of Wireless Sensor Networks and IoT in Disaster Management," 2019, pp. 57–66. doi: 10.1007/978-981-13-0992-2_5.

[11] W. Sun, P. Bocchini, and B. D. Davison, "Applications of artificial intelligence for disaster management," *Natural Hazards*, vol. 103, no. 3, pp. 2631–2689, Sep. 2020, doi: 10.1007/s11069-020-04124-3.

[12] S. K. Abid *et al.*, "Toward an Integrated Disaster Management Approach: How Artificial Intelligence Can Boost Disaster Management," *Sustainability*, vol. 13, no. 22, p. 12560, Nov. 2021, doi: 10.3390/su132212560.

[13] H. Surmann *et al.*, "Lessons from robot-assisted disaster response deployments by the German Rescue Robotics Center task force," *J Field Robot*, vol. 41, no. 3, pp. 782–797, May 2024, doi: 10.1002/rob.22275.

[14] A. Sebastian, "Soft Robotics for Search and Rescue: Advancements, Challenges, and Future Directions," *arXiv preprint arXiv:2502.12373*, 2025.

[15] J. Cani *et al.*, "TRIFFID: Autonomous Robotic Aid For Increasing First Responders Efficiency," *arXiv ID: arXiv: 2502.09379*, 2025.

[16] R. V. Kumar, M. V. D. P. Kumar, Mamatha B, and A. V. Hanuman, "Swarm Robotics for Disaster Management," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 4, pp. 5584–5589, 2024.

[17] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Trans Neural Netw Learn Syst*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.

[18] Ryze Tech, "Tello Drone Specifications," *Ryze Robotics*, 2025.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91.

[20] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[21] S. Thrun and M. Montemerlo, "The Graph SLAM Algorithm with Applications to Large-Scale Mapping of Urban Structures," *Int J Rob Res*, vol. 25, no. 5–6, pp. 403–429, May 2006, doi: 10.1177/0278364906065387.

[22] Y. Bai, Y. Wang, and S. Shen, "Deep Visual-Inertial Odometry with Semantic Features for Indoor Navigation," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 1–14, 2020.

[23] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," *Int J Rob Res*, vol. 34, no. 7, pp. 1–19, Jul. 2015, doi: 10.15607/RSS.2014.X.007.