Developing a Vision-Guided Tracked Robot for Fire Emergency Missions

Pham Thuc Anh Nguyen [1], An Hai Nguyen [2], and Son Hoang [3,*]

[1] Department of Automation Engineering, School of Electrical and Electronic Engineering,
Hanoi University of Science and Technology, Hanoi, Vietnam
[2] Control, Automation in Production and Improvement of Technology Institute,
Academy of Military Science and Technology, Hanoi, Vietnam
[3] Faculty of Electronics Engineering 1, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam
Email: anh.nguyenphamthuc@hust.edu.vn (P.T.A.N.); anhaicpt@gmail.com (A.H.N.); hoangson@ptit.edu.vn (S.H.)
*Corresponding author

Abstract—**Emergency fire suppression activities subject rescue personnel to severe thermal conditions, hazardous fumes, and blast risks, creating extremely perilous environments for human operators. Rapid urban development has amplified fire emergency occurrences, necessitating the deployment of advanced autonomous firefighting platforms. This study presents an innovative tracked firefighting robot designed to navigate complex terrain and autonomously detect and approach fire sources. The system integrates a You Only Look Once version 8 (YOLOv8)-based deep learning model for real-time fire detection and employs depth imaging to calculate angular deviation and distance to the fire. These measurements are transmitted to a Programmable Logic Controller (PLC)-based control unit via a Modbus RS485 interface for responsive control. To enable autonomous navigation, the proposed robot combines an enhanced Bug-2 pathfinding algorithm with LiDAR-based environmental mapping and Hector Simultaneous Localization and Mapping (SLAM) for real-time localization and mapping. The core innovation lies in the integration of YOLOv8-based fire detection with deviation-angle-optimized Bug-2 navigation and a PLC-Robot Operating System (ROS) control architecture, enabling precise fire localization and obstacle avoidance in dynamic environments. Experimental validation confirms the effectiveness of the proposed firefighting robot in identifying fire sources and navigating around obstacles, demonstrating its potential as a reliable solution for autonomous firefighting in hazardous scenarios.**

***Keywords*—autonomous firefighting systems, robotic fire suppression, You Only Look Once version 8 (YOLOv8) neural networks, obstacle navigation, flame detection**

I. Introduction

Fire plays a vital role in human life and development. In modern society, fire is an essential energy source in industrial production, transportation, healthcare, and daily living. However, fire accident is one of the most dangerous catastrophes which may threaten seriously human life and health, and destroy national resources [1]. Recently in Vietnam, the speed of urbanization and industrialization has been increasing, more and more factories, industrial parks, export processing zones have been built, combined with harsh weather conditions, have led to significant amount of fire accidents and explosions with complex progression. As reported from National Police Department of Fire Fighting and Prevention, there were 1710 fire incidents across the country in the first half of 2025, resulting in 42 fatalities, 48 injuries, and an estimated property loss of 160.98 billion VND, along with 216.94 hectares of forest destroyed. Additionally, 13 explosions occurred causing 5 deaths and 27 injuries.

During fire emergencies, emergency personnel serve as frontline forces entering hazardous zones to conduct evacuation operations, suppress combustion, and limit damage escalation. These responsibilities demand physical resilience and technical competencies, coupled with courage to confront potentially fatal circumstances. Fires often escalate rapidly and tend to spread larger, with solid material fires reaching flame temperatures of 800 °C–1000 °C, and ambient temperatures within 5 m ranging from 100 °C–300 °C. In gas fires, flames can reach 1000 °C–1300 °C, with surrounding temperatures rising to 150 °C–350 °C due to heat radiation. Thick smoke causes drastically reducing visibility, making spatial orientation extremely difficult and suffocation. In addition, high temperatures and toxic gases released from burning materials can cause burns, poisoning, or unconsciousness if firefighters are not equipped properly with specialized protective gear.

Structural instability, particularly in aging buildings experiencing extensive fire damage, poses additional threats that may entrap response personnel. The presence of combustible materials including gas, chemicals, and electrical equipment creates explosion risks. These combined hazards render firefighting operations extremely dangerous, necessitating advanced autonomous robotic platforms in emergency response [2].

Manuscript received August 22, 2025; revised September 12, 2025; accepted October 20, 2025; published February 28, 2026.

doi: 10.18178/ijmerr.15.1.102-113

Recent advances in firefighting robotics include remote-operated systems like the Colossus robot (developed by Shark Robotics, France) and the TAF-20 system (Magirus/EmiControls), the latter deployed in New South Wales, Australia. These systems feature thermal imaging, remote control, and enhanced mobility over complex terrain. They illustrate increasing incorporation of sensor, computer vision, and AI-assisted technologies in emergency response, particularly for situations too hazardous for direct human intervention [3, 4].

The designed firefighting robot has a tracking-belt crawler structure, that provides flexibility for operation across various terrains. For fire detection, the platform employs a You Only Look Once version 8 (YOLOv8) deep neural network trained on a diverse dataset of flame imagery, enabling high-confidence recognition in both static and dynamic conditions. Fire localization is achieved through the fusion of pixel coordinates and depth data from a depth-sensing camera, allowing accurate computation of the fire's spatial position and inclined angle. An enhanced Bug-2 algorithm enables obstacle avoidance using URG-04LX LiDAR data. The robot proceeds along a direct path unless impeded, in which case it traces obstacle boundaries until the original trajectory is resumed. The deviation angle is utilized to optimize exit decisions, thereby improving navigation efficiency and response time. The control architecture is based on a Programmable Logic Controller (PLC), which governs the movement of the robot toward the detected fire locations. Next, the Robot Operating System (ROS) has applied for managing vision sensing data, coordinating control logic, and facilitating inter process communication. Additionally, the system employs Hector Simultaneous Localization and Mapping (SLAM) to simultaneously map the environment and track the trajectory of the robot, further enhancing autonomous navigation capabilities [5]. Experimental results validate the system's ability to detect fire sources and avoid obstacles. The research contributes a practical and scalable solution to the growing demand for intelligent firefighting robots capable of operating in high-risk scenarios.

II. Literature Review

A. Remote-Controlled Firefighting Systems

Currently, extensive investigation and innovation in fire suppression robotics are being pursued globally, with particular emphasis in the United States, Japan, and various European enterprises [1]. Contemporary navigation developments encompass advanced simultaneous localization and mapping [5] and multi-sensor integration [3] for demanding operational environments. Fire suppression robotic platforms can function through manual operation or autonomous control, determined by specific application requirements and environmental conditions. During initial developmental phases, multiple remotely-operated systems were established and exhibited successful performance under dangerous circumstances. These platforms generally utilize wireless command interfaces, enabling personnel to conduct navigation and fire suppression activities from secure locations. The Thermite RS3 platform, manufactured by Howe & Howe Technology, represents one of the most advanced and robust fire suppression robots globally. The Thermite operates via remote command with an operational range extending up to 400 m. Utilizing portable control units, operators receive live video streams that facilitate navigation across challenging terrain and obstacle displacement while enduring severe environmental conditions. The Thermite platform serves as essential equipment for fire departments operating in high-risk, hazardous environments. The "Scrum Force" fire suppression unit operates at the emergency station in Ichihara City, Chiba Prefecture, Japan. This wheeled platform demonstrates high maneuverability and rapid movement capabilities, though it exhibits limited obstacle traversal performance [4]. Various remotely-controlled fire suppression platforms including LUF60, FIRREROB, FFR-1, and FIRE-MOTE4800 have been documented regarding their design parameters and operational characteristics [2]. The FlameShieldBot serves as a prominent illustration, incorporating both manual and autonomous operational modes. During manual operation, the system utilizes Wi-Fi connectivity for remote command, providing real-time video transmission and precise manipulation of locomotion and suppression mechanisms [6].

B. Autonomous Firefighting Approaches

These platforms are wirelessly commanded to access combustion zones and can deploy water to suppress flames, disperse gases and redirect fire progression while establishing clear visibility for rescue operations. It is evident that during initial development phases, most platforms were operated remotely by human controllers, necessitating the involvement of emergency personnel at safe distances to observe fire scenarios and direct robotic operations [7–9]. Wirelessly-controlled fire suppression platforms remain unable to completely replace emergency personnel in essential operations including victim extraction, medical emergency support, or decision-making in intricate situations requiring environmental awareness and human judgment. Additionally, operational dependability can be compromised when command signals experience disruption from electromagnetic disturbances, extensive communication ranges, or hostile environmental factors, potentially causing delayed or unsuccessful responses. To overcome constraints linked with wirelessly-controlled fire suppression platforms, specifically their reliance on consistent command signals and restricted capability for sophisticated decision-making, autonomous fire suppression systems offer a viable solution. These platforms utilize sophisticated sensing technology and machine intelligence to identify, evaluate, and react to combustion events without direct human control. Advanced pathfinding systems currently incorporate movement forecasting and route optimization [10] for independent operations. Li *et al.* [11] and colleagues engineered an independent fire suppression platform through integration of simultaneous localization and mapping with You Only Look Once version 4 (YOLOv4) deep neural network for flame identification and assessment. The prototype system could transport multiple extinguisher types and select suitable suppression agents based on fire magnitude. Dai *et al.* [12] presented a methodology combining YOLOv8 with laser ranging data for accurate spatial assessment in robotic and autonomous vehicle applications. Jindal *et al.* [13] developed an independent ground platform featuring a manipulator arm for metropolitan fire suppression operations. Bhattarai and Martinez-Ramon [14] implemented deep Q-learning methodology for navigation decision-making utilizing observed and archived information in combustion environments. Investigation by Tan *et al.* [15] suggested integrating image analysis with A pathfinding algorithm for route optimization within indoor settings using evacuation plans as input parameters. Their work employed Particle Swarm Optimization (PSO) enhanced Convolutional Neural Networks (CNN) combined with image analysis methods for identifying and localizing combustion sources within visual data.

C. Vision-Based Fire Detection Methods

One of the critical functions of autonomous firefighting robots is their ability to detect and localize fire using computer vision systems. These systems may use real time video analysis powered by deep learning algorithms to identify fire sources, smoke patterns, and hazardous zones with high precision. By enabling robots to perceive and interpret complex visual environments, computer vision significantly enhances their autonomy, allowing them to navigate dynamic fire scenes, assess risks, and initiate suppression actions without direct human input. When combined with autonomous navigation and decision-making capabilities, computer vision transforms firefighting robots from passive tools into proactive agents capable of operating in high-risk environments with minimal supervision. Several advanced vision-based systems have been developed and integrated to enhance the robot's ability to detect, localize, and respond to fire-related hazards in complex environments. Contemporary object identification approaches and machine learning frameworks include Faster R-CNN, Single Shot MultiBox Detector, and RetinaNet architectures. YOLO emerges as the predominant methodology regarding precision, processing speed, and computational efficiency, with YOLOv8 representing an advanced instantaneous object recognition platform that has garnered significant research interest [16–20]. Yu *et al.* [21] applied YOLOv8 and its optimization enhance the accuracy of real-time detection of fire, the efficiency of path planning, and the precision of firefighting operation in drone swarms during fire emergencies.

D. Recent Fire Detection Technologies

Modern fire detection has evolved from sensor-based to computer vision approaches using deep learning. YOLOv8 incorporates improved feature extraction, enhanced loss functions, and anchor-free detection heads, demonstrating enhanced performance in real-time object detection tasks. Multi-modal systems combining thermal and visible imaging improve detection reliability under adverse conditions like smoke and low visibility.

E. Research Gap and Motivation

Despite advances in firefighting robotics, several critical challenges remain:

Vision-control integration: Existing systems use either remote control or autonomous navigation, but few integrate real-time vision-based fire detection with industrial-grade PLC control for reliable operation.

Navigation efficiency: Traditional Bug-2 algorithm, while simple, lacks optimization for dynamic fire scenarios where response time is critical.

System architecture: Current autonomous systems lack scalable middleware frameworks that can coordinate multi-sensor data (vision, LiDAR, depth) with real-time control logic.

These challenges motivate the need for an integrated autonomous firefighting platform combining YOLOv8 vision, optimized Bug-2 navigation, and ROS-PLC architecture for reliable operation in hazardous environments.

III. Materials and Methods

Firefighting robots must access fires and stops at the appropriate distance to extinguish fires. The structure of the proposed firefighting robot consists of mechanical structure, machine vision unit, control unit, actuator unit, sensing unit and management unit (Fig. 1).

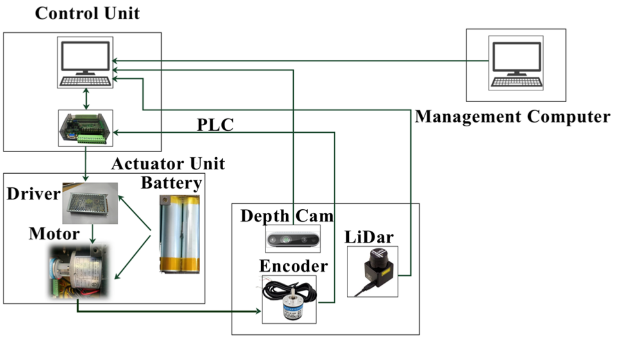
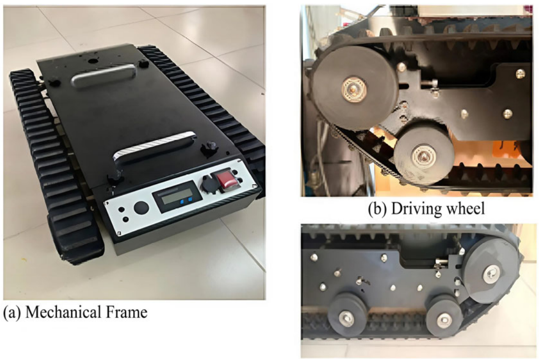


Fig. 1. Overall system architecture of the proposed firefighting robot.

A. Description of Mechanical Structure

Fig. 2 illustrates mechanical structure of the prototype fire fighting robot.

To ensure operational stability under high-temperature conditions, the robot's structural components—including the frame, chassis, wheels, and belt—are fabricated from steel alloy and coated with electrostatic paint. This combination enhances both thermal resistance and corrosion protection, enabling the robot to function effectively in environments with temperatures reaching approximately 300 °C. The robot is equipped with a mobility system comprising a pair of driving wheels located at the rear, a pair of idler wheels at the front, and three pairs of road wheels distributed along its body. The driving wheels are powered by Direct Current (DC) gear motors, which transmit motion through a continuous track-belt system. This belt is constructed from fire-resistant rubber and serves to transfer torque from the driving wheels to the idler wheels, while also facilitating traction and movement. The idler wheels are strategically positioned to maintain tension in the track-belt, thereby preventing slack and ensuring consistent performance. Internally, the track-belt features high grousers that interlock with the convex grooves of the driving wheels, enhancing grip and mechanical engagement. This tracked configuration increases the robot's ground contact area, thereby improving traction and enabling efficient navigation across uneven and high-friction terrains.


(a) Mechanical Frame

(b) Driving wheel

(c) Idler and road wheels

Fig. 2. Mechanical structure of the firefighting robot.

B. Driver and Control Units

The robot's power supply, drive system, and control unit are integrated within the chassis, as illustrated in Fig. 3. The system is powered by an 8-cell lithium-ion battery with a capacity of 400 Ah, which is designed for rapid replacement to facilitate operational continuity. This power configuration enables the robot to perform uninterrupted tasks for up to four hours, thereby supporting extended deployment in demanding environments.

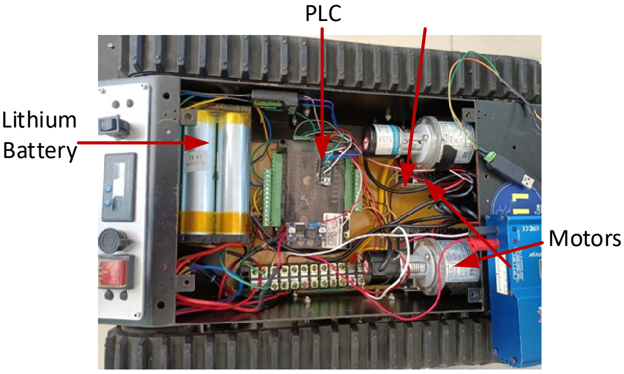


Fig. 3. Driver system and control unit.

The robot is engineered to operate in two distinct modes: remote control and autonomous control. This study primarily emphasizes the autonomous mode. The control architecture comprises a host computer interfaced with a Mitsubishi PLC FX3U-24MT/6AD/2DA via RS485 communication protocol. This configuration facilitates the transmission of control commands to two motor drivers, which in turn regulate the operation of the DC motors responsible for driving the robot's wheels.

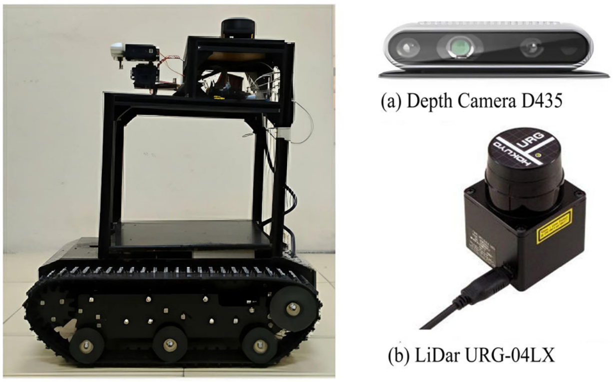
Through this control mechanism, the robot is capable of navigating toward fire-affected areas while simultaneously detecting and avoiding obstacles along its path.

At the initiation of each operational cycle, a depth camera mounted on the robot captures video data and transmits it to the host computer. Utilizing the YOLOv8 object detection algorithm, the system identifies fire-affected regions within the video frames. Subsequently, the image processing module computes both the tilt angle between the camera's optical axis and the centroid of the detected fire area, as well as the spatial coordinates of the fire zone. These parameters are then transmitted to the PLC for further processing.

The PLC, based on the received data, generates control signals that are sent to the motor drivers, which in turn regulate the DC motors responsible for driving the rear wheels. This enables the robot to autonomously navigate toward the fire location and halt at a predefined safe distance of 30 cm from the fire source. Communication between the host computer and the PLC is established via the Modbus RTU protocol over an RS-485 interface.

The robot's initial motion trajectory is planned as a straight line from its current position to the target location near the fire. In the event that an obstacle is detected along this path, the robot dynamically adjusts its trajectory by following the contour of the obstacle using the Bug2 algorithm [12]. Obstacle detection is facilitated by a LiDAR sensor mounted at the front of the robot, which continuously transmits environmental data to the host computer.

Throughout its movement, the robot continuously estimates its current position using feedback from wheel encoders. When the contour-following path intersects the original straight-line trajectory, the robot reorients itself to resume the direct path toward the target. This contour navigation process is repeated as necessary until the robot successfully reaches the designated position. The complete system configuration is illustrated in Fig. 4.


(a) Depth Camera D435

(b) LiDar URG-04LX

Fig. 4. Robot is equipped depth-camera and LiDar

C. Vision System Development

Autonomous firefighting robots should have a machine vision unit to detect and identify fires and determine exact locations of fires to transmit the information to control unit.

In the study, the machine vision unit is designed from a host computer and Depth-Camera. The duty of the machine vision unit is to detect fire-areas in real-time. Depth-Camera D435i takes pictures and send it to the host computer and a CNN-based detector has been trained for fire detection.

Fire detection is a necessary task to ensure accuracy and quick-fire extinguishing. Traditional methods for fire detection commonly used sensor-based detection system using temperature sensors, gas sensors, and smoke sensors. However, these detection systems are often applied for small fires, for large fires that are at risks of breaking out quickly, they are no longer effective. Recent development in deep-learning methods have proven highly effective in machine vision fields, including fire detection. Deep-learning is a machine learning method based on deep neural networks capable of processing data and automatically extracting high-level features in machine vision problems. Deep neural networks- such as CNN are used to automatically analyze and classify data, process information in real-time, extract feature from images and point out there is detected object in the input image or not.

Currently, deep learning-based methods for object detection can be divided into two groups: two-stage detectors and one-stage detectors. Two-stage detectors, such as Region-based CNN (R-CNN) and its variants, first generate object proposals and then classify each proposal. One-stage detectors, such as YOLO and its variants, directly predict the presence and location of objects in a single step. Overall, YOLOv8 expresses great potential as an object detection model that can enhance real-time detection capabilities. YOLOv8 is evaluated as an advancement in the realm of computer vision and is likely to stimulate additional exploration and progress in this domain. In this study, YOLOv8 detector has been trained and applied.

D. Build Up YOLOv8 Model for Detecting Fires

The YOLOv8 framework comprises three primary modules: the backbone network extracts feature representations from input imagery; the neck module consolidates features from multiple layers before forwarding to prediction components; ultimately, the detection head determines object presence and spatial coordinates.

The training process encompasses the three key steps:

(1) Step 1. Data collection and augmentation

First, a dataset of about 2800 images containing a diverse range of fires has been gathered from Police Department of Fire Fighting and Prevention, Kaggle, and self-collected. The dataset includes indoor, small and large, dim and clear fire conditions to ensure the model’s robustness across different demographics. Duplicated images should be removed.

Due to deep learning models rely on large volumes of diverse data to increase accurate predictions of the detection model. when training data is limited or not diverse enough, data augmentation is vital. Data augmentation is a technique used to create new variants from original data samples by applying random transformation. Three transformations of flip, noise addition and brightness change have been applied and increase the dataset to 11,200 images. By using rich datasets, the training model can learn more complex features, leading to higher prediction accuracy. Furthermore, data augmentation helps to eliminate overfitting, when the model provides accurate predictions for training data but not for new data. Before feeding the images into the model, several preprocessing steps were performed to prepare the data. All images were resized to a standard resolution of 640×640 pixels to ensure uniformity and reduce computational complexity. Pixel values were normalized to a range of 0 to 1, which is crucial for faster convergence during training. Representative samples from the fire image dataset used for training the YOLOv8 detection model are illustrated in Fig. 5.

Fig. 5. Representative samples from the fire image dataset used for training the YOLOv8 detection model.

(2) Step 2. Labelling

We used the Makesense.ai tool to create a box bounding around the fire areas in all data images with the label “Fire”. Labelling is for delineate accurately fire locations in the images. Bounding boxes were adjusted to be appropriately sized and positioned to ensure the model can effectively detect all instances of fire, regardless of their scale. After labeling is completed, we will have a file .txt saving corresponding information of fire area in all images.

The labeled data are split into three sets: training set (70%), validation set (20%) and testing set (10%). Other step is resizing and normalizing the data is necessary to optimize the training model. With a well-labelled and sufficient dataset, the accuracy can be high. Also using a Graphics Processing Unit (GPU) over a Central Processing Unit (CPU) is recommended for the training process to further enhance the performance by decreasing computation time.

(3) Step 3. Training YOLOv8 for fire detection

In this step, the YOLOv8 model is trained on the labeled dataset. The model is trained to recognize the features of fire. Model training was executed on Google Colab infrastructure utilizing Tesla P100 graphics processing units with 12GB memory capacity. The software stack comprised Python 3 and PyTorch 1.7 frameworks.

Training iterations extended across 100 epochs, implementing early termination protocols that activate when validation performance metrics show no enhancement over 20 successive iterations. This approach ensures effective feature learning while mitigating overfitting risks during optimization for peak performance. Each training cycle processed 32-sample batches to maintain optimal equilibrium between parameter update frequency and computational resource utilization. Furthermore, this batch size ensures adequate GPU memory utilization without causing memory overflow. The initial learning rate was configured at 0.001 to enable moderate convergence velocity while avoiding overshooting optimal parameter configurations or inefficient resource consumption. Learning rate reduction by factor 0.01 occurred at five-epoch intervals. The trained model's performance evaluation is presented in Section IV.A.

E. Determine Deviation Angle from the Depth Camera to the Fires

The camera is fixed rigidly on the robot, with its viewing direction aligned with the robot's forward motion. Upon system initialization, the host computer begins receiving data from the depth camera. The YOLOv8 model, installed on the host machine, processes the incoming images to detect fire sources and generates bounding boxes around identified regions. Subsequently, the system calculates the angular displacement between the camera's axis and the center of the detected fire, along with the spatial coordinates of the fire location. These parameters are then transmitted to the PLC-based control unit, which utilizes them to execute actuator commands for navigation and fire suppression tasks.

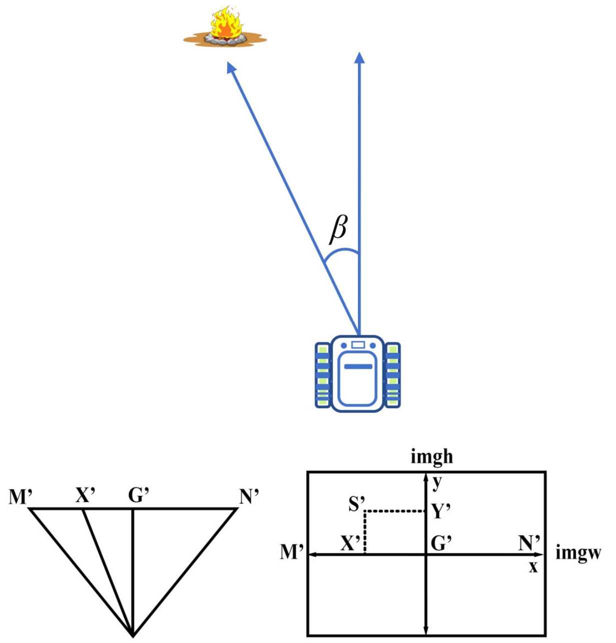


Fig. 6. Coordinates of fire in image plane.

Parameters necessary for calculating as follows:

- The camera is placed at constant height (Fig. 4) coordinates of the depth camera F(0, 0, FO);
- Image size: 640 (imgw)×640 (imgh);
- Limit of view angle in horizontal plane $\alpha = \widehat{P'FQ'} = 69{,}4°$;
- Limit of view angle in vertical plane $\gamma = \widehat{M'FN'} = 42{,}5°$.
- Optical axis FG;
- Information from Depth Camera;
- Coordinates of center of fire in the image plane (x_0, y_0);
- Pixel depth of fire FE (Deep).

Next step we propose algorithm for calculating $\widehat{X'FG'}$ angle between the camera and the fire.

Geometrical transformation from Fig. 6 provides the deviation angle between the depth camera and the fire as Eq. (1):

$$\widehat{X'FG'} = arctan\left(\frac{\left(x_0 - \frac{imgw}{2}\right) tan\frac{\gamma}{2}}{\frac{imgh}{2}}\right) \tag{1}$$

F. Algorithm for Calculating Position of the Detected Fire

Geometric transformation from Fig. 7 provides the distance FG' from the depth camera to the center of the image plane as Eq. (2) and the distance FS' from depth camera to center of the detected fire in the image plane as Eq. (3):

$$FG' = hPixel = \frac{1}{2}\left(\frac{imgw}{2\,tan\frac{\alpha}{2}} + \frac{imgh}{2\,tan\frac{\beta}{2}}\right) \tag{2}$$

$$FS' = depthPixel = \sqrt{x^2 + y^2 + (hPixel)^2} \tag{3}$$

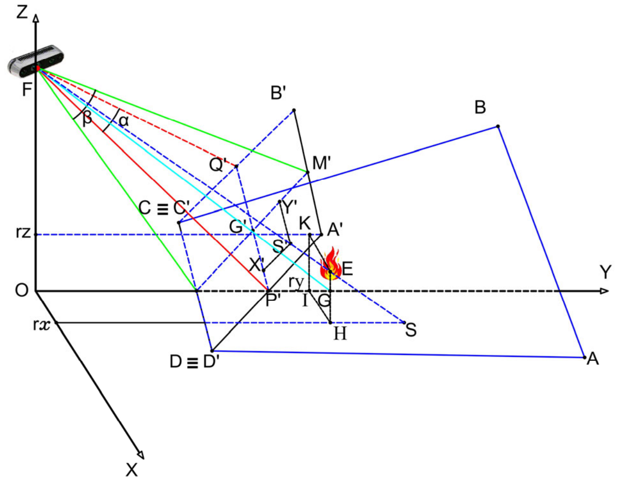


Fig. 7. Geometric scheme for calculating position of the center of fire (point E).

The coordinates (r_x, r_y, r_z) of the detected fire center E in the reference coordinate system are obtained from Eqs. (2) and (3) via geometric transformation as follows:

$$r_x = EK = S'Y'\frac{FE}{FS'} = x\frac{depth}{depthPixel} \tag{4}$$

$$r_y = \frac{depth}{depthPixel}\sqrt{y^2 + hPixel^2}\,\sin\left(\begin{matrix} arctan\,\frac{y}{hPixel} \\ +\widehat{GFO} \end{matrix}\right) \quad (5)$$

$$r_z = FO - \sqrt{depth^2 - r_x^2 - r_y^2} \quad (6)$$

G. Navigation and Obstacle Avoidance Algorithm

The deviation angle and position concerning to fires calculated above are transmitted to the control unit. Although the robot has flexibility to cross complex terrains, but large obstacles that may be encountered in the working environment become problematic. Then tasks of the control unit are to guide the Robot approach fires and avoid obstacles on the path.

For obstacle avoidance, almost firefighting robots are equipped ultrasonic sensors. However, these sensors have low reliability, leading to inaccuracy in detecting and avoiding obstacles. In this research, RPLiDar URG-04LX device is proposed instead to determine distance to obstacles. The device uses high powered laser and light energy to generate large amounts of data points to measure the distance and direction of target objects from afar. RPLiDar URX-04LX can scan 270 degrees and allow quick and accurate data collection of surrounding objects. Depend on practical experience, scanning angle in front of the robot can be divided into 6 areas (V1–V6) as pointed in Fig. 8. Scanning information of LiDar provides whether or not there is an obstacle in each area and determine necessary movements of the Robot as shown in Table I.

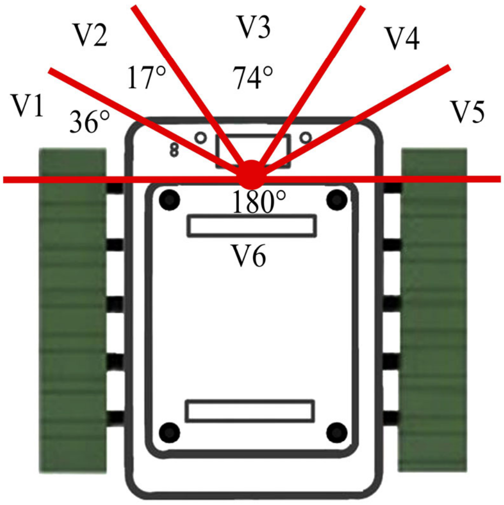


Fig. 8. Partition of working areas for LiDar.

TABLE I. CASE BY CASE IMPLEMENTATION

Case	V1	V2	V3	V4	V5	Necessary Movement
1	√	√	-	√	√	Forward
2	-	√	√	√	-	Turn right
3	-	√	√	-	-	Turn right
4	-	-	√	√	-	Turn left
5	-	√	√	-	-	Turn right
6	-	-	√	√	√	Turn left

Among current algorithms robot such as Potential Fields, Bug, A*, DWA for obstacle avoidance for mobile, Bug-2 algorithm is simple and effective. The conventional Bug-2 pathfinding approach establishes a trajectory from starting point to target location, with the platform following this route until obstruction occurs. Upon encountering barriers, the system circumnavigates obstacle boundaries while continuously computing updated trajectories from each position until the revised path aligns with the original route. After reaching on point having same line as previous, it starts moving to destination by following pervious generated path. It can be seen that, traditional Bug 2 algorithm is still not optimal in time and distance. In this study, the Bug 2 algorithm is improved by using deviation angle of the robot with the detected fire to determine when to leave a contour around an obstacle, leading the trajectory is more optimal in time and distance.

After encountering obstacle and determining necessary movement, Robot turns and tracks a contour of the obstacle. Then the LiDar rays scanning in V1 and V5 will be used to determine the distance from the Robot and the obstacle. The movement of the Robot has to ensure that its distance to the obstacle always within the range of 20–60 cm as shown in the Fig. 9. If the distance is less than 30 cm, Robot will be controlled to move away from the obstacle. On the contrary, if the distance is greater than 60 cm, it will move closer to the obstacle.

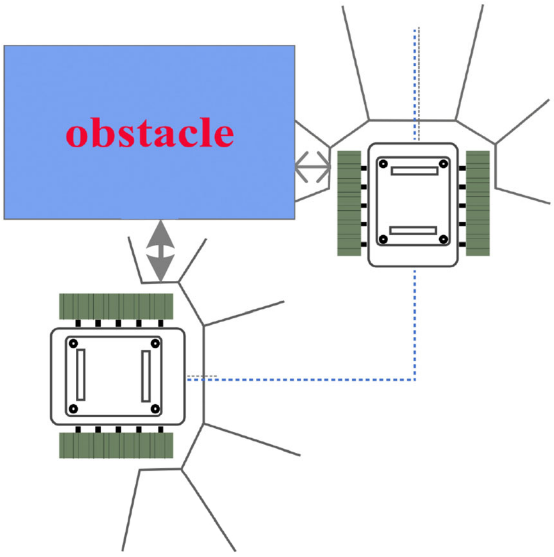


Fig. 9. V1 and V5 areas of LiDar scanning help the robot keep a constant distance with the obstacle.

Proposed improved Bug 2 algorithm:

As starting, when no fire is detected, the robot maintains rotation 360° and move randomly in the vicinity. When YOLOv8 model detects fire already, the host computer will calculate the deviation angle between the robot and fire using Eq. (1), and the position of the fire center using Eqs. (4)–(6), then transmit these data to the control unit. As pointed out by Bug 2 algorithm, the robot will change its current orientation and head to the detected fire, then move straight until it reaches to the goal if no obstacle appears. If it encounters an obstacle on trajectory of the robot, scanning data from LiDar will activate working area and create necessary movement for the robot to avoid obstacle but maintain an optimal contour around it. During movement, the robot continuously calculates its own coordinate position based on information received from encoders. When the trajectory around the obstacle contour intersects the original straight trajectory, the robot will turn and follow the straight trajectory. The contour loop process will continue if the Robot encounters other obstacles until it reaches the target position. An illustration of moving to approach a fire and avoid obstacles is depicted in Fig. 10.

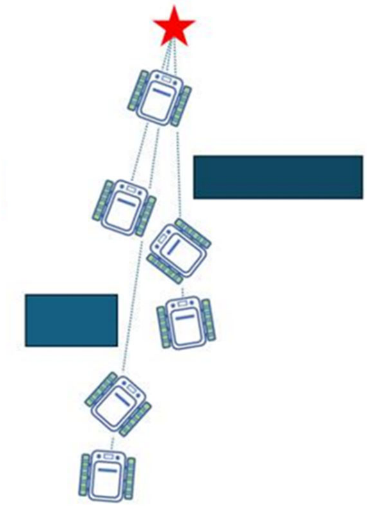


Fig. 10. Motion directs to the detected fire area and avoids obstacles by means of improved Bug 2 algorithm.

H. Software Framework Implementation

As mentioned above, the firefighting robot must possess advanced perceptual capabilities to identify fire zones and comprehend operational environment. It is required to make real-time decisions regarding navigation and fire suppression activities. To support such functionalities, a flexible and robust software framework is necessary to efficiently manage multi-sensor data, coordinate control logic, and facilitate reliable inter-process communication. The ROS serves as a powerful open-source middleware that enables the development of distributed, modular, and highly scalable robotic systems, making it particularly well-suited for safety-critical applications such as autonomous firefighting. The ROS-based architecture of the robot, as illustrated in Fig. 11, comprises five primary nodes, each responsible for a distinct functional module within the robot's control framework:

(1) Camera and Lidar Nodes: These two nodes interface directly with the Depth Camera and LiDar respectively, acquiring real-time environmental data. The camera captures visual information, while the Lidar provides spatial measurements essential for obstacle detection and mapping.

(2) Detection Node: This node processes image data from Depth Camera to identify fire sources. Upon detection, it publishes messages to a topic named 'Topic_goal', which contains the distance and angular position of the fire. This information is critical for guiding the robot's navigation and targeting behavior.

(3) PLC node: This node is programmed in Python to communicate with a PLC via RS485 protocol. It is designed to implement a class named RS485, which specializes in interacting with the PLC through this widely used industrial communication standard.

(4) Avoid Obstacles Node: This node analyzes Lidar data to segment the frontal area of the robot into five distinct zones, enabling localized obstacle detection. Based on the proximity of objects in each zone, the node generates control commands such as turn left, turn right, move forward, or stop. Additionally, it subscribes to the "Topic_goal" to receive fire location data and transmits both navigation commands and target coordinates to the robot's PLC for execution.

(5) SLAM Node: Utilizing Lidar scan data from the "scan" topic, this node performs SLAM to construct a map of the surrounding environment. The generated map is stored for future use, supporting enhanced navigation and situational awareness.

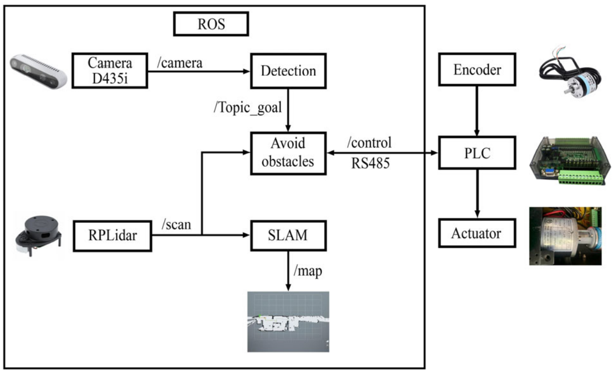


Fig. 11. ROS-based architecture of the system.

IV. Result and Discussion

A. Fire Detection Performance

1) Training and validation metrics

It is can be observed in Fig. 12 that as the box loss, classification loss, and distribution focal loss decrease gradually in both the training and validation phases as the train progresses. These decreation expressed improved accuracy in bounding box prediction, object classification, and confidence in prediction, respectively. It is evident that the model is correctly placing those boxes where fires are and the model is not overfitting as well as able to generalize well with unseen data.

Although the precision metrics had small fluctuating, but in general showed a general upward trend, indicating gradual improvement in the accuracy of identifying fires. The recall metrics were variable, but overall showed upward trajectory, providing an increase in the ability to detect fires consistently. The mean average precision (mAP) at IoU thresholds of 0.50 (mAP@50) and 0.50–0.95 (mAP@50–95) also demonstrated a notable increase throughout the training epochs, underscoring a significant improvement overall precision and reliability of the model at these thresholds.

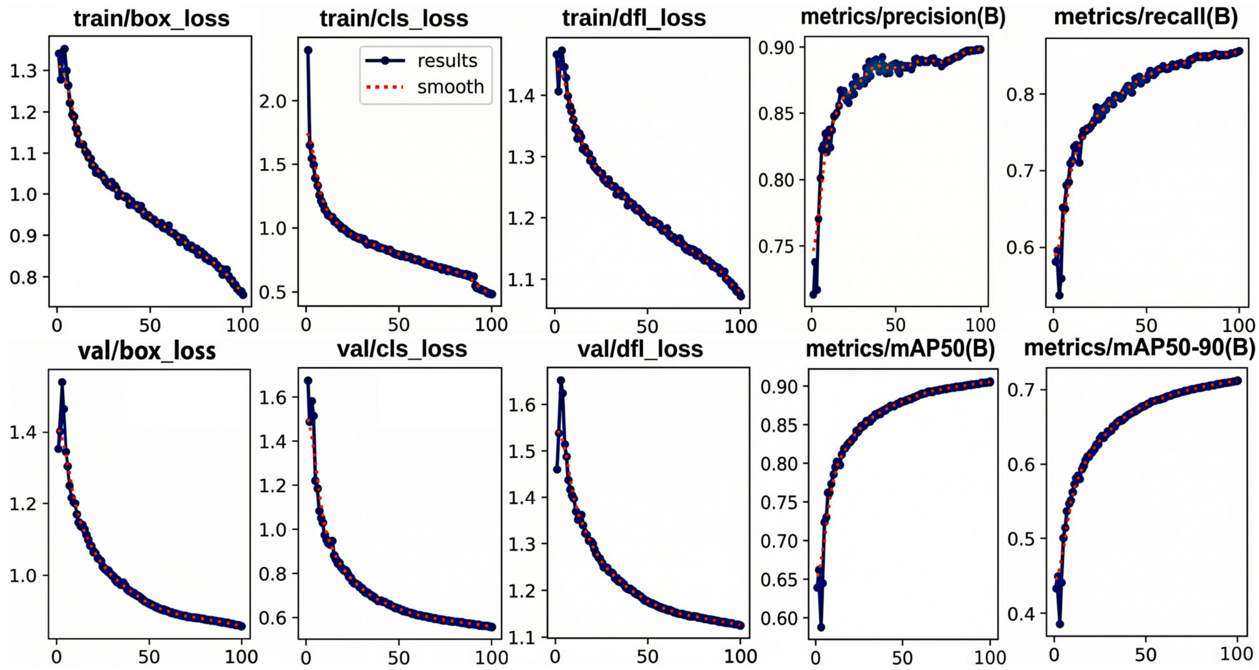


Fig. 12. Training and validation performance metrics for proposed fire detection model.

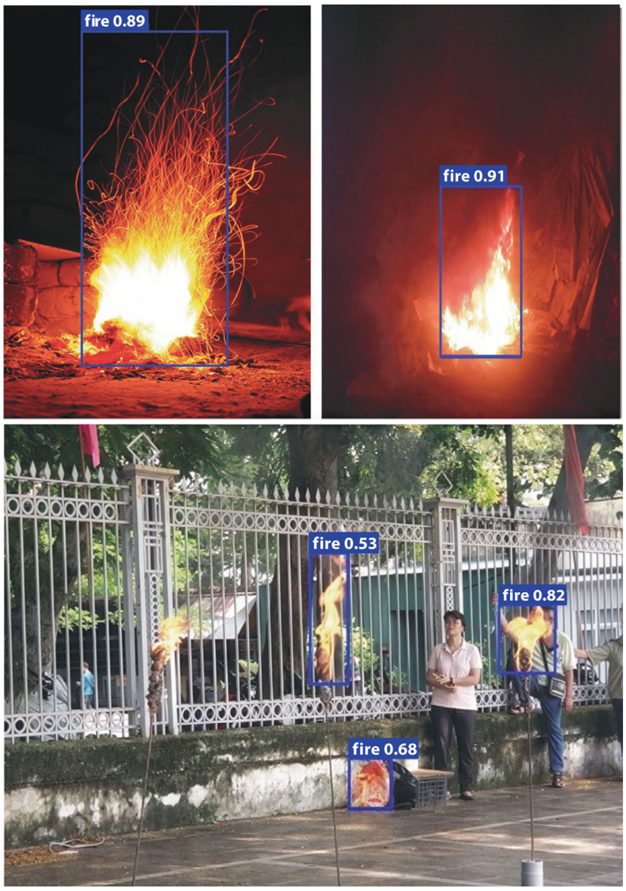


Fig. 13. Fire detection results using input images from the testing set.

2) *Testing set evaluation*

Following the training phase, a YOLOv8 neural network model with updated weights capable of detecting and identifying fire was successfully deployed. The trained model was installed on the host computer of the machine vision unit integrated into the firefighting robot. To evaluate the model's effectiveness, a series of test images (Fig. 13) were input into the system. Additionally, real-time images captured by the camera mounted on the robot were transmitted to the host computer for live inference and validation (Fig. 14). This dual-source testing approach enabled comprehensive assessment of the model's performance in both controlled and operational environments.

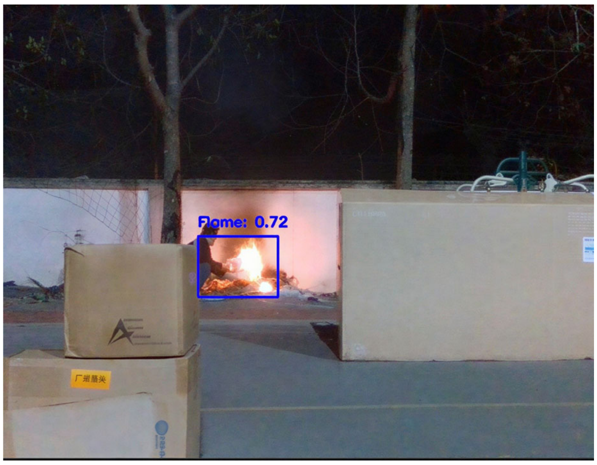


Fig. 14. Real-time fire detection results using the trained YOLOv8 model.

The results of detection as input some images of the testing set are shown in Fig. 13. It can be seen that when the image is bright and the fire area is clear, the confidence score is high (>0.89) and when the image is dimmer, the confidence score decreases, but is still acceptable (in the range 0.53–0.89). With this result, it is completely possible to integrate the model into the Robot system to perform the required problem.

B. Fire Localization Accuracy

Measurement of deviation angle and distance from the camera and the fire's center is implemented and compared with calculated as shown in Table II.

TABLE II. EXPERIMENT RESULTS MEASURING THE DISTANCE AND DEVIATION ANGLE FROM THE DEPTH-CAMERA TO THE FIRE AREA

No	Calculated Values		Measured Values	
	Distance (m)	Deviation angle (°)	Distance (mm)	Deviation Angle (°)
1	2.137	7.5	1.9	7.5
2	1.675	−13	1.6	−12.5
3	1.406	−19.8	1.4	−20.0
4	1.289	10.7	1.2	11.0
5	1.014	−26.5	1.0	−26.0
6	0.942	8.3	0.9	8.0
7	0.821	15.2	0.8	15.5
8	0.775	−5.6	0.7	−5.0
9	0.663	12.5	0.6	13.0
10	0.615	−9.5	0.5	−10.0

C. Autonomous Navigation and Obstacle Avoidance

To evaluate the robot's capabilities in fire detection, navigation toward hazardous areas, and obstacle avoidance, a series of controlled experiments were conducted. The test environment was arranged as illustrated in Fig. 15, where a fire source was positioned 20 m from the robot, surrounded by two obstacles.

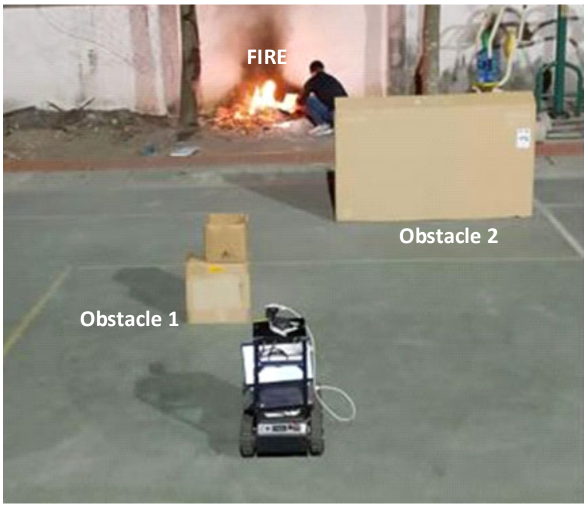


Fig. 15. Obstacle and fire distribution map.

The experimental procedure and corresponding results are depicted in Fig. 16. First, the robot rotates to find the target until the YOLOv8 detector identifies a fire (Fig. 16(a)). When the fire area is detected, the robot will head to the fire (Fig. 16(b)). The global trajectory is established as a straight line connecting the actual position of the Robot to the position in front of the fire area at a safe distance of 50 cm. The robot moves in a straight line until an obstacle appears (Fig. 16(c)). When encountering an obstacle, the Robot will move forward to maintain a safe distance from the obstacle (Fig. 16(d)). The robot then rotates to avoid the obstacle (Fig. 16(e)) and continues to avoid and keep distance (Fig. 16(f)). The robot encounters obstacle 2 (Fig. 16(g)) and rotates to avoid obstacle 2 (Fig. 16(h)). After passing the obstacles, the robot heads to the fire (Fig. 16(i)), approaches the fire (Fig. 16(j)) and moves closer to the fire (Fig. 16(k)). Finally, the robot stops 50 cm from the fire to prepare to extinguish the fire (Fig. 16(l)).

The experimental results validate the effectiveness of the machine vision system in accurately identifying fire zones and demonstrate the robustness of the robot's motion control algorithms in navigating complex environments and approaching hazardous areas with precision.

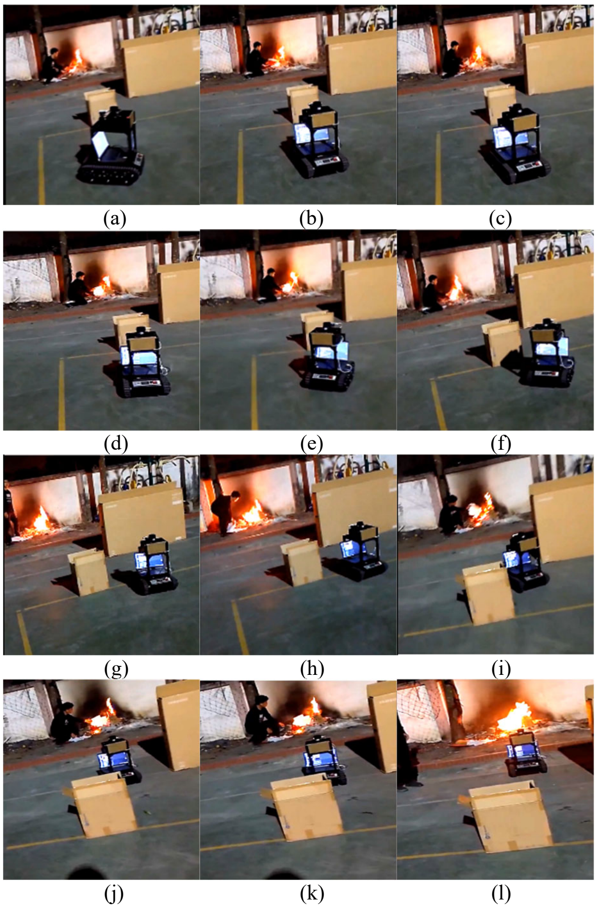


Fig. 16. Experimental output: navigating toward fire while avoiding obstacles. (a). rotate at site; (b). head to fire; (c). obstacle appears; (d). move forward to maintain distance; (e). rotate to avoid; (f). avoid and keep distance; (g). encounter obstacle 2; (h). rotate to avoid obstacle 2; (i). head to the fire; (j). appoach the fire; (k). closer to the fire; (l). stops at 50 cm from the fire.

Our system achieves fire detection confidence scores ranging from 0.53 to 0.89 across different lighting conditions, with higher confidence (>0.89) in optimal visibility scenarios. The autonomous navigation successfully completed obstacle avoidance tasks in controlled environments, demonstrating the practical feasibility of the integrated YOLOv8 and improved Bug-2 approach for firefighting applications.

During navigation, the robot employs Hector SLAM to simultaneously construct a map and record its trajectory. As illustrated in Fig. 17, the resulting map demonstrates a relatively high degree of accuracy in comparison to the actual environment. However, the presence of noise introduces certain distortions that affect the overall map quality.

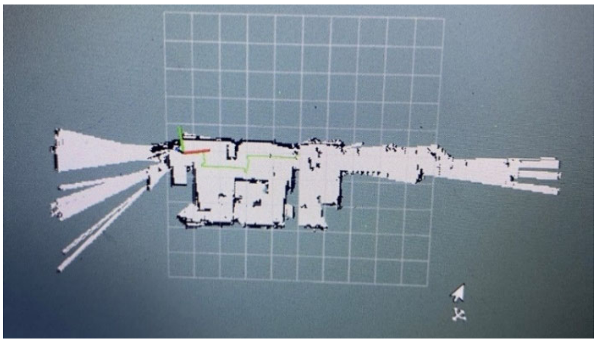


Fig. 17. Trajectory and map construction via hector SLAM.

D. Discussion

This study successfully integrates YOLOv8-based fire detection with PLC-ROS control architecture for autonomous firefighting operations. The YOLOv8 model achieved robust fire detection with confidence scores ranging from 0.53 to 0.89 depending on lighting conditions, and real-time detection performance validated its deployment readiness. The geometric transformation method accurately calculated fire position and deviation angle as shown in Table II, enabling precise navigation. The improved Bug-2 algorithm successfully guided the robot to fire locations while avoiding obstacles, and the ROS-based architecture facilitated efficient multi-sensor coordination between vision, LiDAR, and control subsystems.

However, several limitations were identified. SLAM mapping exhibited noise artifacts affecting map quality as observed in Fig. 17. Detection confidence decreased under low-light conditions with minimum values of 0.53, which may affect performance in smoke-filled environments. Furthermore, testing was conducted in controlled environments, while real fire scenarios present additional challenges including extreme heat, thick smoke, and dynamic conditions.

Future work should focus on integrating thermal imaging for enhanced detection under smoke conditions, implementing advanced SLAM algorithms for noise reduction, conducting field testing in realistic fire environments, and incorporating actual fire suppression mechanisms such as water delivery systems or foam dispensers to complete the autonomous firefighting capability.

V. Conclusion

This research presents the development of a mobile robotic platform for autonomous fire detection and navigation in hazardous environments. The belt-tracking locomotion system of the proposed robot ensures stable movement across uneven terrain, while the YOLOv8-based vision model—trained on a diverse fire image dataset—delivers high-confidence flame detection in both static and dynamic conditions. Fire localization is achieved with high accuracy through depth-sensing and geometric transformation techniques. A key innovation is the enhancement of the Bug-2 algorithm using deviation angle analysis and LiDAR data, which significantly improves obstacle avoidance and navigation efficiency. The integration of ROS enables seamless coordination between vision, LiDAR, and control systems, while supporting SLAM for real-time mapping and localization. Experimental results confirm the robot’s effectiveness in fire source identification, terrain adaptability, and autonomous operation. By combining deep learning, optimized path planning, and intelligent control, this research offers a scalable and practical solution to the growing demand for autonomous firefighting technologies in dynamic and high-risk scenarios.

Future research will focus on expanding capabilities of the robot in real-world firefighting scenarios. This includes integrating thermal imaging for smoke-obscured fire detection, enhancing multi-robot coordination for large-scale fire suppression, and developing adaptive decision-making algorithms for dynamic hazard assessment. Additionally, efforts will be made to improve energy efficiency, real-time communication with emergency response systems, and robustness under extreme environmental conditions. These advancements aim to further establish autonomous firefighting robots as reliable assets in modern disaster response strategies.

Ethical Approval and Informed Consent

This study did not involve human participants or animals.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

SH conducted the research, developed the dynamic model, and performed the control design; PTAN supervised the study and provided critical revisions to the manuscript; AHN carried out the simulations, analyzed the results, and contributed to the manuscript preparation; all authors had approved the final version.

Funding

This research was funded by Hanoi University of Science and Technology (HUST) under project number T2022-PC-004.

Acknowledgement

The authors thank Nguyen Dac Toan, Nguyen Hai Dang, Ho Van Hoat, Nguyen Thanh Tam, and the members of the AI & Robotics Lab for supporting this research.

References

[1] Kiran, K. Keerthana, M. Meghana *et al.,* "A review on fire fighting robot," *Int. Res. J. Eng. Technol.*, vol. 9, no. 4, pp. 3623–3626, 2022.

[2] C. F. Tan, S. M. Liew, M. R. Alkahari *et al.*, "Fire fighting mobile robot: state of the art and recent development," *Aust. J. Basic Appl. Sci.*, vol. 7, no. 10, pp. 220–230, 2013.

[3] T. N. Canh, T. S. Nguyen, C. H. Quach *et al.*, "Multisensor data fusion for reliable obstacle avoidance," in *Proc. 11th Int. Conf. Control, Automation and Information Sciences (ICCAIS)*, 2022, pp. 385–390.

[4] B. S. Sampath, "Automatic fire extinguisher robot," in *Proc. 8th Int. Conf. Ubiquitous Robots Ambient Intell. (URAI)*, 2011, pp. 215–218.

[5] T. N. Canh, B. N. Quoc, H. Zhang *et al.*, "IRAF-SLAM: An illumination-robust and adaptive feature-culling front-end for visual SLAM in challenging environments," arXiv preprint, arXiv: 2507.07752, 2025.

[6] K. Lalitha, B. K. Kumar, K. Satvik *et al.*, "Autonomous and manual fire suppression using FlameShieldBot," *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 7, no. 3, pp. 12921–12929, 2025.

[7] J. Zhao, Z. Zhang, S. Liu *et al.*, "Design and research of an articulated tracked firefighting robot," *Sensors*, vol. 22, no. 14, 5086, 2022.

[8] V. Mangayarkarasi, "Remote controlled fire fighting robot," *Int. J. Trend Sci. Res. Dev.*, vol. 2, no. 5, pp. 820–826, 2018.

[9] P. W. Aung and W. Y. Win, "Remote controlled fire fighting robot," *Int. J. Sci. Eng. Technol. Res.*, vol. 3, no. 24, pp. 4830–4835, 2014.

[10] T. N. Canh, X. HoangVan, and N. Y. Chong, "Enhancing social robot navigation with integrated motion prediction and trajectory planning in dynamic human environments," in *Proc. 24th Int. Conf. Control, Automation and Systems (ICCAS)*, 2024, pp. 731–736.

[11] S. Li, J. Yun, C. Feng *et al.*, "An indoor autonomous inspection and firefighting robot based on SLAM and flame image recognition," *Fire*, vol. 6, no. 3, 93, 2023.

[12] Y. Dai, D. Kim, and K. Lee, "An advanced approach to object detection and tracking in robotics and autonomous vehicles using YOLOv8 and LiDAR data fusion," *Electronics*, vol. 13, no. 12, 2250, 2024.

[13] K. Jindal, A. Wang, D. Thakur *et al.*, "Design and deployment of an autonomous unmanned ground vehicle for urban firefighting scenarios," *Field Robotics*, vol. 1, pp. 1–17, 2021.

[14] M. Bhattarai and M. Martinez-Ramon, "A deep Q-learning based path planning and navigation system for firefighting environments," in *Proc. 13th Int. Conf. Agents Artif. Intell. (ICAART)*, 2021, vol. 2, pp. 267–277.

[15] S. Q. Y. Tan, V. J. Karthik, A. Govind *et al.*, "An approach into navigation and vision for autonomous fire fighting robots," *International Journal of Advanced Mechatronic Systems*, vol. 10, no. 3, pp. 156-164, 2023.

[16] M. Aliff, N. S. Sani, M. I. Yusof *et al.*, "Development of fire-fighting robot (QROB)," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 1, pp. 142–147, 2019.

[17] C. B. Liu and N. Ahuja, "Vision-based fire detection," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR 2004)*, 2004, vol. 4, pp. 134–137.

[18] J. H. Kim, S. Jo, and B. Y. Lattimer, "Feature selection for intelligent firefighting robot classification of fire, smoke, and thermal reflections using thermal infrared images," *Journal of Sensors*, vol. 2016, no. 1, 8410731, 2016.

[19] J. Jalani, D. Misman, A. S. Sadun *et al.*, "Automatic fire-fighting robot with notification," in *IOP Conf. Ser.: Mater. Sci. Eng.*, 2019, vol. 637, no. 1, 012002.

[20] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint, arXiv: 2004.10934, 2020.

[21] B. Yu, S. Yu, Y. Zhao *et al.*, "Intelligent firefighting technology for drone swarms with multi-sensor integrated path planning: YOLOv8 algorithm-driven fire source identification and precision deployment strategy," *Drones*, vol. 9, no. 5, 348, 2025.