An Autonomous Arduino-based Firefighting Robot for Laboratory Environments

Md. Mohiuddin Maruf
Department of Software Engineering
Daffodil International University, Dhaka, Bangladesh.

M. Mahamudul Hasan Sagor
Department of Software Engineering
Daffodil International University, Dhaka, Bangladesh.

Sadiya Khanom Kanta
Department of Software Engineering
Daffodil International University, Dhaka, Bangladesh

Md. Hafizul Imran
Department of Software Engineering,
Daffodil International University,
Dhaka, Bangladesh.

ABSTRACT
The laboratory fires are increasing due to flammable chemicals, high-voltage tools, and complex experiments. Traditional firefighting methods often fail in such environments. They respond slowly and cannot adapt to rapid fire spread. This study introduces an autonomous firefighting robot designed for laboratory use. The system is built using Arduino and includes three parts. These are a 180° infrared flame detection unit, a servo-controlled water sprayer, and an autonomous movement system. Control algorithms help the robot detect and extinguish flames quickly and accurately. Tests show a 95.8% accuracy in flame detection. The robot can put out fires from up to 90 cm away. The average time to extinguish a flame is 18.9 seconds. In 250 tests, the robot showed a 98.0% success rate. These results confirm the system's reliability and stability. The robot offers a low-cost and effective tool for fire safety. It can improve lab protection through intelligent automation.

Keywords
Firefighting Robot, Autonomous Fire Suppression, Arduino-Based Robot, Firefighting Robot for Laboratory Fire Safety.

1. INTRODUCTION
Laboratories are highly vulnerable to fire hazards due to the presence of flammable chemicals, high-voltage equipment, and complex experiments. The National Fire Protection Association (NFPA) reports an increasing trend in laboratory fire incidents. These fires pose threats to life, equipment, and property [1-6, 35-40]. Traditional fire safety methods, such as manual extinguishers and fixed ventilation systems, often fail in hidden or hard-to-reach areas. Rapid flame spread can make manual intervention difficult.

Robotic systems have emerged as a solution to address this issue. However, most current robotic systems have limited sensing range, slow response times, and require remote control. These factors reduce their ability to act independently in real-time fire conditions.

This paper presents an autonomous firefighting robot designed for laboratory use. The system is based on an Arduino microcontroller and includes three core subsystems. These are the Flame Detection System, the Fire Extinguishing System, and the Robot Movement Control System. A 180° infrared flame sensor allows wide-angle flame detection. A servo-based nozzle and a relay-operated water pump handle the suppression task. The robot uses decision-making algorithms to detect, approach, and extinguish flames.

Testing was conducted under controlled conditions. The robot achieved 95.8% accuracy in flame detection. It could extinguish fire from a distance of up to 90 cm. The average time to extinguish a fire was 18.9 seconds. In 250 trials, the robot maintained a 98.0% success rate. These results confirm the robot's reliability, stability, and efficiency.

The paper is structured as follows: Section 2 reviews related work. Section 3 explains the methodology. Section 4 discusses hardware and software components. Section 5 presents experiments and results. Section 6 concludes the paper and suggests future work.

2. LITERATURE REVIEW
The Robotics and embedded systems have advanced rapidly in recent years. These advances have improved autonomous firefighting technologies. Firefighting robots are now used to reduce risks in dangerous environments [7-9]. Many systems have been developed to detect and extinguish fires. These systems rely on sensors, microcontrollers, and wireless communication. Most operate without human help and can suppress fire using onboard mechanisms.

This section reviews various firefighting robots. The systems are analyzed based on sensors, control methods, autonomy, and structure. Key limitations are also discussed. These include poor sensor fusion, weak decision-making, and inefficient actuation. The proposed system addresses these challenges. It uses a mathematical model and focuses on energy and response optimization.

Cakir and Ezzulddin (2016) designed a robot that uses a fan to put out fire [10]. The system worked in controlled settings. However, it was ineffective in open environments. The fan's airflow could spread fire instead of extinguishing it. Also, the robot detected fire in only one direction. The proposed system improves on this. It uses water to suppress fire and covers a 180-degree area.

Kadam et al. (2018) built a robot using Arduino Uno R3. The robot included gas and temperature sensors [11]. It also used ultrasonic sensors for avoiding obstacles. However, gas and temperature sensors can cause delays in fire detection. The proposed system avoids this delay. It uses IR flame sensors, which respond faster and more accurately. It also works without remote control.

Taha et al. (2018) created a robot for enclosed places. It used one flame sensor and could detect fire within a 60-degree range [12]. Its deployment was complex and limited its usability. The proposed system uses three flame sensors. These sensors cover 180 degrees and allow easy setup.

Pan and Zhu (2017) explained how to build a fire-fighting robot using Arduino [13]. Their book included design steps and component details. However, their work did not involve real-world testing. In contrast, the proposed system has been tested in practical scenarios.

Diwanji et al. (2019) presented a robot with three flame sensors [14]. It also used a servo motor to spray water. Their system was similar in design to the proposed one. But they did not explore algorithms in depth. They also lacked detailed testing. The current research adds advanced algorithms and extensive testing.

Sharma and Singh (2021) developed the FEBO robot. It featured fire detection, suppression, live video, and voice communication [15]. It could be controlled by a smartphone. Multiple versions were designed for various locations. However, the system lacked real-world testing. The proposed robot is tested and verified in practical conditions. It is also easier to deploy.

Sangewar et al. (2021) built a robot with a gas sensor and camera. It used Arduino Mega 2560. It could detect fire but relied on manual operation [16]. The proposed system improves on this. It automates both detection and suppression tasks. This improves speed and efficiency.

Anuradha et al. (2023) designed a robot similar in structure to the proposed one. Their robot used Arduino and could detect fire [17]. But it did not use a complex algorithm. It also lacked proper testing. The current system includes a refined algorithm. It has been tested in different environments.

Najm et al. (2024) created a robot for indoor fire emergencies. They described the system but did not include detailed algorithms [18]. The robot was also not tested under extreme conditions. The proposed system includes a full algorithm. It has been tested in harsh situations.

Kucukdermenci and Ilten (2025) built a remote-controlled fire robot. It had a detection angle over 120 degrees. However, it lacked autonomous operation [19]. It also did not undergo real-world testing. The proposed robot covers 180 degrees and works without human control. It has been validated through field testing.

From this review, three main problems were found in existing systems:

- Limited Detection Range: Many robots use one flame sensor. This limits the field of view and reduces detection accuracy.
- Lack of Real-World Testing: Most systems were not tested in practical fire situations. This affects their reliability.
- Dependence on Manual Control: Several systems depend on remote or Wi-Fi control. These may fail during emergencies.

This study solves the above problems. The system includes the following improvements:

- Wider Detection Range: Three flame sensors offer 180-degree fire detection. This is more effective than older systems.
- Autonomous Operation: The robot detects and suppresses fires without human help. This improves speed and reduces human error.
- Tested System with Strong Algorithm: The robot has been tested under various conditions. The fire detection algorithm is accurate. The system also controls water use efficiently.

This research provides a better alternative to existing robots. It is reliable, fast, and easy to use. The system can be deployed in real-world fire emergencies. It is suitable for use in Laboratory.

3. METHODOLOGY

3.1 System Architecture

The proposed firefighter robot system is composed of one Robot Control System and three primary subsystems: Flame Detection System, Fire Extinguishing System, and Robot Movement Control System. These subsystems collaborate to autonomously detect fire, navigate towards the flame source, and extinguish it effectively. Figure 1 displays the proposed architecture for the Firefighter Robot.

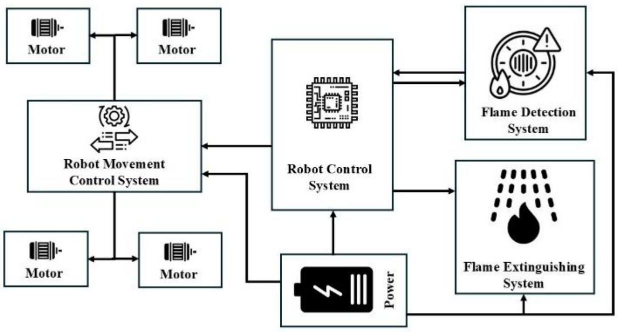

Fig 1: Architecture of the proposed system.

3.1.1 Robot Control System

The firefighter robot uses the Arduino Uno as its main controller. The Arduino Uno is based on the ATmega328P microcontroller. This microcontroller uses the Harvard architecture. It separates program memory and data memory for better performance. The ATmega328P has 32 KB of flash memory, with 0.5 KB used for the bootloader. It also has 2 KB of SRAM and 1 KB of EEPROM. The processor runs at 16 MHz, allowing real-time operations [20].

The robot receives input from several sensors. These sensors provide data such as flame presence, direction, and path status. The sensor input is defined as:

$$S(t) = \{s_1(t), s_2(t), \ldots\ldots, s_n(t)\} \quad (1)$$

Here, $S_i(t)$ is the reading from the *i*-th sensor at time t. The Arduino uses this input to make decisions in real time. The control decision function is represented as:

$$\mathrm{D(S)} = \begin{cases} Navigation\ Mode, & if\ no\ flame\ is\ detected \\ Extinguishing\ Mode, & if\ flame\ is\ detected \end{cases} \quad (2)$$

This function determines the robot's operational mode. After deciding the mode, a control signal $C(t)$ is generated. This signal depends on both *D(S)* and the instruction set *P*:

$$\mathrm{C(t)} = f(D(S), P) \quad (3)$$

The function f maps decisions and instructions to actuator commands. These include motor actions and fire suppression.

The Arduino has a built-in Analog-to-Digital Converter (ADC). It converts analog signals from sensors into digital values. The ADC has 10-bit resolution. It maps 0–5V input into values from 0 to 1023. The robot also uses digital input pins and interrupt routines. This ensures fast and prioritized responses [21].

Sensor data is collected periodically. This is controlled by the sampling frequency f_s:

$$f_s = \frac{1}{t_s} \quad (4)$$

Here, T_s is the sampling period. This regular sampling supports real-time response to changing conditions [22].

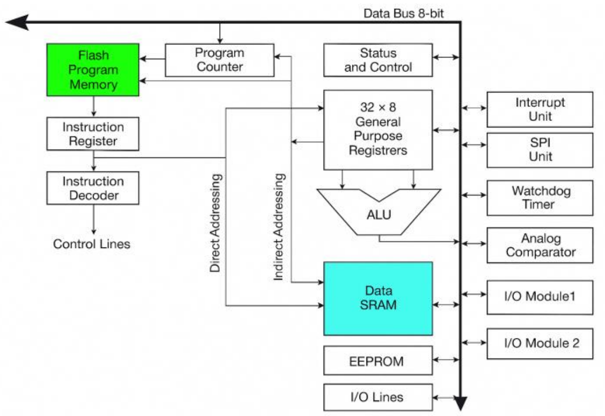

Fig 2: Arduino Architecture [20].

3.1.2 Flame Detection System

Accurate flame detection is a key feature of the firefighter robot. The robot uses infrared (IR) flame sensors. These sensors are sensitive to wavelengths from 760 nm to 1100 nm. This range matches the emission spectrum of most hydrocarbon fires. Each IR sensor converts infrared radiation into an electrical signal. This signal is used for real-time decisions in navigation and fire suppression.

Each sensor contains a photodiode. The photodiode detects incoming IR radiation and produces a current. This current is known as photocurrent and is defined as:

$$I_{ph} = R \cdot F \quad (5)$$

Here, I_{ph} is the photocurrent, R is the photodiode responsivity (A/W), and F is the flame intensity [23].

This small current is amplified and changed into voltage. A transimpedance amplifier performs this conversion. The voltage is filtered to reduce noise. Then the filtered voltage (V_f) is compared to a set threshold (V_{th}). A comparator like the LM393 outputs a binary signal based on this comparison:

$$V_{out} = \begin{cases} 1, & if\ V_f \geq V_{th}\ (Flame\ Detected) \\ 0, & if\ V_f < V_{th}\ (No\ Flame\ Detected) \end{cases} \quad (6)$$

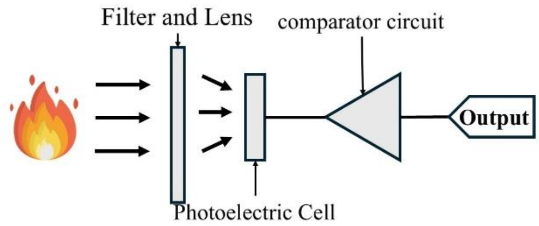

Fig 3: Structure Diagram of the IR Flame Sensor [24].

The analog output can also reflect flame intensity. This is modeled as:

$$V_f = \alpha F + \epsilon \quad (7)$$

Here, α is the system gain, and ϵ is the noise and environmental error [24].

The robot uses multiple IR sensors to cover a wide area. Three sensors are placed on the front side. Each sensor is angled at about 60 degrees. This setup gives 180-degree coverage and reduces blind zones. The robot estimates flame direction by calculating voltage differences:

$$\Delta V = V_{f_1} - V_{f_2} \quad (8)$$

This difference helps the robot face the flame source correctly. Flame distance is estimated using the inverse-square law:

$$F = k \cdot (1/d^2) \quad (9)$$

Here, k is a calibration constant, and d is the flame distance [25].

3.1.3 Fire Extinguishing System

The fire extinguishing mechanism employed in this study integrates multiple hardware components, including a relay-controlled water pump, a servo motor for directional nozzle control, and a precision nozzle designed to optimize water dispersion while minimizing waste. Upon detection of a flame, the control system processes sensor data to compute the required servo angular position θ for accurate nozzle alignment. This angle is determined as a function of the voltage differential ΔV between the flame sensors:

$$\theta = g(\Delta V) \quad (10)$$

where $g(\cdot)$ represents a mapping function that correlates sensor voltage differences to the appropriate servo angle. The relay module governs the activation of the water pump through a digital control signal R(t) defined by:

$$R(t) = \begin{cases} 1, & Pump\ Activated \\ 0, & Pump\ Dectivated \end{cases} \quad (11)$$

The water pump delivers a continuous flow of extinguishing agent, and the flow rate Q can be estimated based on the mechanical power of the pump using:

$$Q = \frac{P}{\rho g h} \quad (12)$$

where P denotes the pump power, ρ is the density of water, g is gravitational acceleration, and h is the pump head (vertical distance the water is lifted [26]. This targeted extinguishing approach ensures that the water is applied precisely to the fire source, reducing overall water usage while maximizing suppression efficiency. Figure 4 display the operational mechanism of the Fire Extinguishing System.

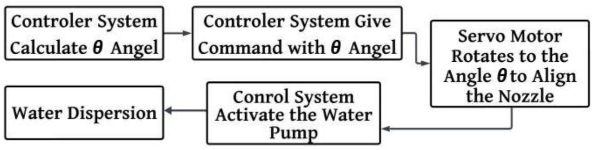

Figure 4: Operational Mechanism of the Fire Extinguishing System.

3.1.4 Robot Movement Control System

The movement system of the robot is based on electronic motor control and classical electromechanical theory. Four BO DC motors drive the wheels of the robot. These motors support both linear and rotational movements. The motors are controlled using an L298N dual H-bridge driver. This module allows current to flow in both directions through the motors. As a result, the robot can move forward and backward.

DC motor operation relies on electromagnetic force. When current flows through a conductor in a magnetic field, a force is produced. This is defined by the Lorentz force law:

$$F = B \cdot I \cdot L \cdot \sin(\theta) \quad (13)$$

Here, F is the force, B is the magnetic flux density, I is the current, L is the conductor length, and θ is the angle between B and I. This principle is represented by Fleming's Left-Hand Rule (Figure 5) [27]. The resulting force turns the rotor, which rotates the wheels.

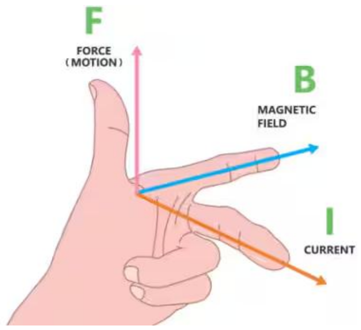

Fig 5: Fleming's Left-Hand Rule [27]

The direction of motor rotation depends on the polarity of voltage across the terminals. An H-bridge circuit is used to switch polarity. It consists of four switches arranged in an "H" shape. Two diagonal switches are turned on to change the current direction. This allows the robot to move forward, backward, and turn. The L298N driver includes this H-bridge system. It simplifies motor control using digital signals from the Arduino.

Motor speed is controlled using Pulse Width Modulation (PWM). PWM controls the voltage by switching it on and off rapidly. The duty cycle (D) determines the average voltage:

$$V_{PWM} = \frac{D}{100} \cdot V_{max} \quad (14)$$

Where VPWM is the average voltage, D is the duty cycle, and Vmax is the supply voltage. Figure 6 shows the PWM waveform.

The Arduino generates the PWM signal. It sends the signal to the motor driver. This enables speed control with low energy loss. The motor's angular speed ω is given by:

$$\omega = (V_{PWM} - V_{drop}) / K \quad (15)$$

Here, V_{drop} is the voltage loss in the circuit, and K is the motor's voltage constant [28].

The use of PWM, H-bridge control, and electromagnetism enables complex movements. The robot can move straight, reverse, rotate, and steer accurately. This is important for aiming the robot at the fire for effective suppression.

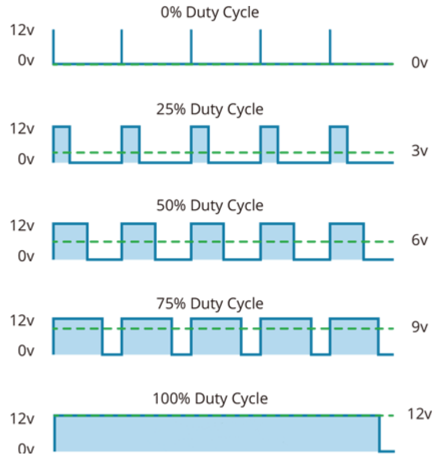

Fig 6: Pulse Width Modulation Technique [28].

3.2 Software Architecture

The software architecture of the proposed firefighter robot functions as the central logic control layer, orchestrating the coordination between sensory input, decision-making algorithms, actuator control, and system safety loops. The embedded software was developed using the Arduino IDE, programmed in C/C++, and deployed directly onto the microcontroller to ensure real-time responsiveness, low latency, and deterministic task scheduling.

The complete operational flow of the software architecture is illustrated in Figure 7: Software Architecture Flowchart of Firefighter Robot.

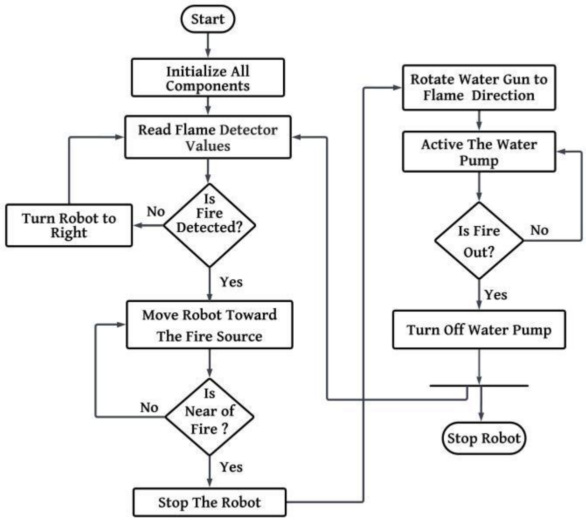

Fig 7: Software Architecture Flowchart of Firefighter Robot.

4. IMPLEMENTATION

The complete implementation of the firefighter robot is divided into two main parts. These are 4.1 Hardware Implementation and 4.2 System Software Implementation.

4.1 Hardware Implementation

This part focuses on building and assembling the robot. It includes the connection of all electromechanical components.

The goal is to ensure proper communication between the control unit and the subsystems. The Arduino microcontroller is used as the central controller. It is an open-source platform that supports flexible hardware design. All sensors and actuators are connected to the Arduino. Each component is integrated to ensure smooth operation. The Arduino manages and coordinates all peripheral devices. The hardware is arranged according to the proposed system design. The full circuit diagram is shown in Figure 8. The robot control system is also highlighted in the same figure under the Robot Control System section.

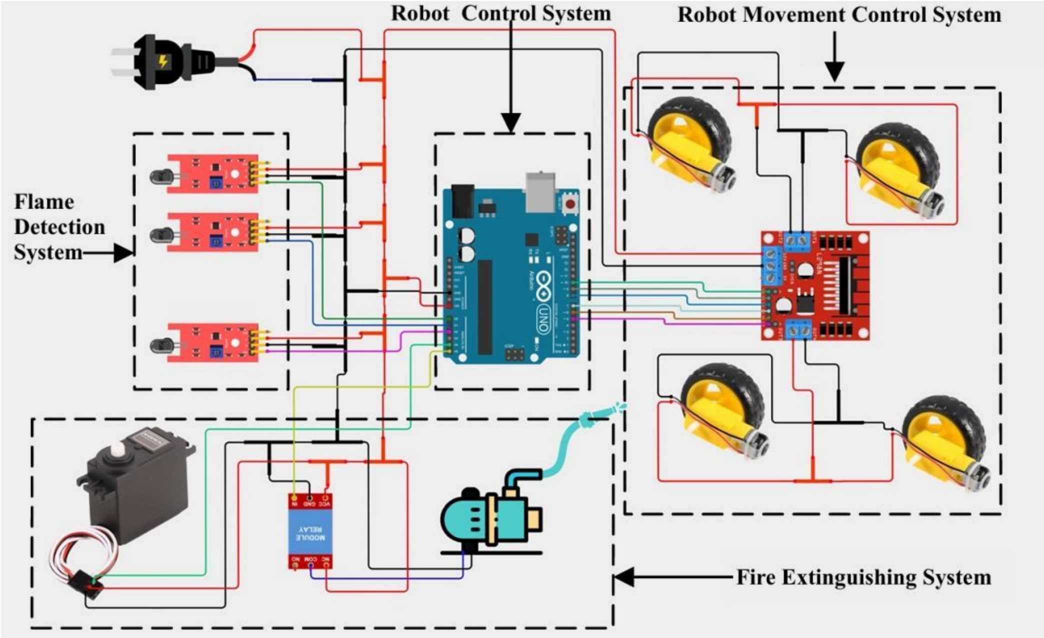

Fig 8: Circuit diagram of the firefighter robot

4.1.1 Flame Detection System
The flame detection system is a vital part of the firefighter robot. It detects flames and guides the robot's response. The system uses three infrared (IR) flame sensors. These sensors are arranged to cover a wide field of view. Each IR sensor can detect fire within a 60-degree range. Together, the three sensors provide 180 degrees of flame detection. This ensures that the robot can monitor its environment effectively. The physical layout of the sensors is shown in Figure 9.

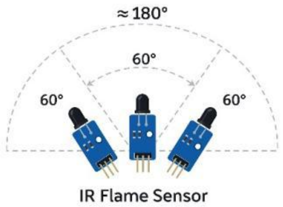

Fig 9: The physical arrangement of the IR flame sensors

Each IR flame sensor is connected to the Arduino microcontroller. The analog outputs of the sensors are linked to analog ports A0, A1, and A2. The circuit layout of this system is illustrated in the Flame Detection System section of Figure 8.

4.1.2 Fire Extinguishing System
The fire extinguishing system is used to suppress the detected flame. It includes a water pump, a relay module, a nozzle, and a servo motor. The relay module controls the pump. The servo motor adjusts the nozzle to target the flame. The nozzle is mounted on the servo motor for directional control.

The water pump is placed at the rear of the robot. The servo motor is placed at the front. The nozzle is directly connected to the servo motor. The robot uses flame detection data to set the nozzle's angle (θ). The pump is connected to the Arduino through a relay module. The relay is connected to analog pin A5. This allows the Arduino to switch the pump on or off. The servo motor is connected to analog pin A4. It controls the nozzle's direction based on flame position. The complete wiring of this system is shown in the Fire Extinguishing System section of Figure 8.

4.1.3 Robot Movement Control System
The The movement of the robot is controlled using two DC motors. These motors drive the left and right wheels. The L298N motor driver controls the motors. The Arduino sends control signals to the motor driver. These signals set the motor direction and speed. For the right motor, the ENA pin is connected to digital pin ~10 to control speed, while IN1 and IN2 are connected to digital pins ~9 and 8 to manage the rotational direction. The left motor's speed is controlled through the ENB pin, which is connected to digital pin ~5, and its direction is controlled by IN3 and IN4 connected to digital pins 7 and ~6. The right motor is connected to output pins 1 and 2 of the motor driver, and the left motor is connected to output pins 3 and 4. By adjusting the logic levels of the IN1 to IN4 pins and modifying the PWM signals applied to ENA and ENB, the robot can perform forward motion, reverse movement, and turning maneuvers as required by the navigation algorithm. The full circuit for this subsystem is shown in the Robot Movement Control System section of Figure 8.

4.2 System Software Implementation
The The software system of the firefighter robot is designed for autonomous flame detection and suppression. It integrates sensor data processing, motor control, and actuator

coordination. The software is developed using the Arduino Integrated Development Environment (IDE). The programming language used is C++, which allows precise control of hardware components. It is suitable for real-time tasks in embedded systems.

The software is structured into several functional algorithms. Each algorithm controls a specific subsystem. The overall control flow is managed by the main operation algorithm.

4.2.1 Main Operational Algorithm

The core logic of the robot is handled by the firefighter robot operation algorithm. It reads data from three IR flame sensors. When a flame is detected, the system calculates the servo angle. It then activates the water pump and monitors the extinguishing process. If no flame is found, the robot starts a scanning routine.

Algorithm 1: Firefighter Robot Operation

```
function FirefighterRobot();
Input: Flame sensor data V = (V1, V2, V3), where each Vi is an analog voltage from IR flame sensors. System initialized with water pump, relay, servo motor, and movement control.
Output: Fire detected and extinguished.

Initialize all hardware components;
while true do
    V1 ← readSensor(A0);
    V2 ← readSensor(A1);
    V3 ← readSensor(A2);
    if (V1 ≥ Tfire OR V2 ≥ Tfire OR V3 ≥ Tfire ) then
        θ ← computeAngle(V1, V2, V3);
        rotateServo(θ);
        activateRelay();
        while (firePresent()) do
            continue extinguishing;
        end
        deactivateRelay();
    else
        moveRobot();
    end
end
end
```

4.2.2 Nozzle Angle Computation Algorithm

The computeAngle() algorithm determines the angle θ for nozzle positioning. It uses a weighted average of the sensor values. This ensures the nozzle points directly at the flame source. It improves targeting and reduces water & time usage.

Algorithm 2: computeAngle()

```
function computeAngle(V1, V2, V3);
Input: V1, V2, V3 — analog voltages from flame sensors (A0, A1, A2).
Output: Servo angle θ for nozzle alignment.

total ← V1 + V2 + V3;
if total = 0 then
    return θ = defaultAngle;
end
weight1 ← (V1 / total) × (-60);
weight2 ← (V2 / total) × 0;
weight3 ← (V3 / total) × 60;
θ ← weight1 + weight2 + weight3;
return θ;
end
```

4.2.3 Robot Search and Movement Algorithm

The moveRobot() algorithm manages robot navigation. When no flame is present, the robot rotates and scans the area. If a flame is detected, the robot moves toward it and prepares to extinguish it.

Algorithm 3: moveRobot()

```
function moveRobot();
Input: Flame sensor data V1, V2, V3 Threshold value Tfire
Output: Robot navigates toward the fire or stops after search

searchCount ← 0;
while searchCount < 4 do
    V1 ← readSensor(A0);
    V2 ← readSensor(A1);
    V3 ← readSensor(A2);
    if (V1 ≥ Tfire OR V2 ≥ Tfire OR V3 ≥ Tfire) then
        moveRobo();
        while (V1 ≥ Tfire OR V2 ≥ Tfire OR V3 ≥ Tfire) do
            V1 ← readSensor(A0);
            V2 ← readSensor(A1);
            V3 ← readSensor(A2);
        end
        stopMoving();
        return;
    else
        turnRight();
        delay(turnDelay);
        searchCount ← searchCount + 1;
    end
end
stopMoving();
end
```

4.2.4 Software Functional Components

To ensure modular and maintainable code, several key functions were implemented in the Arduino IDE. The readSensor(Ax) function reads analog input from a flame sensor. The computeAngle(V1, V2, V3) function calculates the nozzle angle θ based on sensor data. The rotateServo(θ) function adjusts the nozzle direction. The activateRelay() and deactivateRelay() functions control the water pump. The moveRobot() function manages navigation. The firePresent() function checks for flame presence. Additional functions include turnRight() for directional control and stopMoving() to halt movement.

5. RESULTS & DISCUSSION

This section presents the experimental evaluation and comprehensive performance analysis of the proposed autonomous firefighter robot. The evaluation focuses on multiple key aspects, including flame detection accuracy, servo-based nozzle alignment precision, fire extinguishing efficiency, system stability, and failure rate assessment. A series of controlled laboratory experiments was conducted under varying flame intensities, distances, and environmental conditions to thoroughly assess the robot's operational capabilities. The collected data were systematically analyzed to quantify the system's responsiveness, precision, and reliability. Comparative analysis with several recent research works was also performed to validate the superior performance of the proposed system. The results confirm that the firefighter robot offers significant improvements in extinguishing accuracy, reduced extinguishing time, and enhanced system stability, demonstrating its practical applicability for real-world indoor firefighting scenarios.

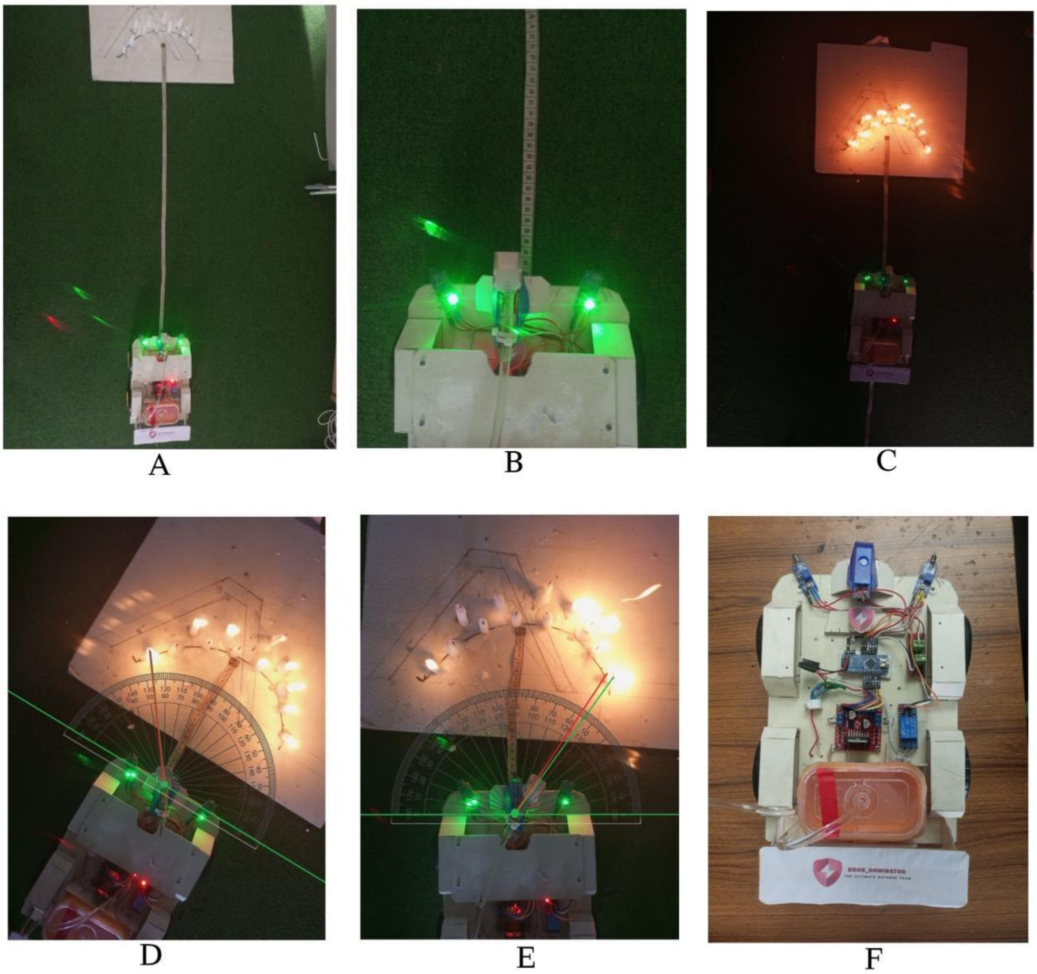

Fig 10: Experimental Test Scenarios of the Firefighter Robot Under Various Fire Conditions

Figure 10 presents key stages of the firefighter robot's testing process under experimental conditions. Part (A) shows the robot detecting fire from a measured distance. Part (B) displays a clear line-of-sight view between the robot and the fire source. Part (C) captures the robot after it detects the fire and moves beyond the halfway point toward the flame. Parts (D) and (E) illustrate the firefighting operation and nozzle alignment accuracy. In part (D), the nozzle is aligned with zero-degree error. In part (E), the nozzle alignment shows a minor two-degree error. Part (F) provides a top view of the robot, highlighting the complete component setup.

5.1 Flame Detection Accuracy Evaluation and Comparative Analysis

The flame detection performance of the firefighter robot was evaluated through controlled experiments. Flames were placed at different angles ranging from -60° to +60° relative to the robot. The robot used three IR flame sensors positioned at -60°, 0°, and +60°. The sensor data were processed using a weighted-angle computation algorithm to estimate the flame's direction.

The system showed high accuracy in detecting the flame position. The estimation error was between 0° and 3°. The robot achieved 100% accuracy at 0°, which is the center. At the edge positions, the accuracy was 96.67% at -60° and 95% at +60°. The average detection accuracy across all angles was about 95.8%. This is shown in Figure 11 and Figure 12.

These results confirm that the detection system is reliable and accurate. The low error rate ensures proper nozzle alignment. This improves fire suppression efficiency.

To validate performance, results were compared with other studies. Ding et al. (2025) reported an average accuracy of 94.6% [29]. Ahamed et al. (2024) achieved 95% but noted failures at wider angles [30]. The proposed system showed better accuracy with consistent performance across all angles. This is due to the use of a weighted-angle model and real-time sensor feedback. These features help correct small errors during flame detection and targeting.

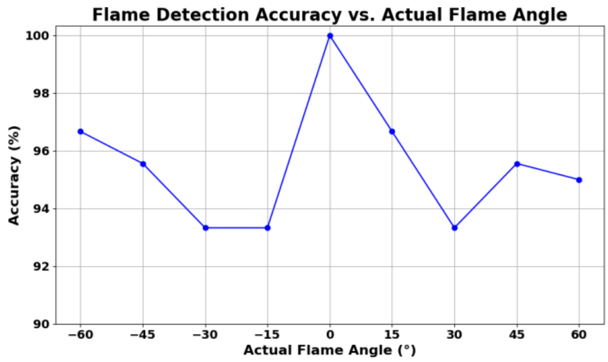

Fig 11: Flame Detection Accuracy vs. Actual Flame Angle

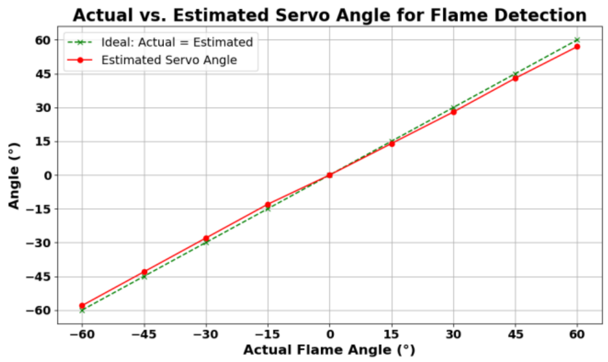

Fig 12: Actual vs. Estimated Servo Angle for Flame Detection

5.2 Fire Extinguishing Time Evaluation and Comparative Analysis

The extinguishing time was tested by placing the robot at four different distances from the fire source. Twelve standard candles were used to simulate a small indoor fire. The robot was placed at distances of 30 cm, 50 cm, 70 cm, and 90 cm. Each test recorded the time needed to extinguish the flame.

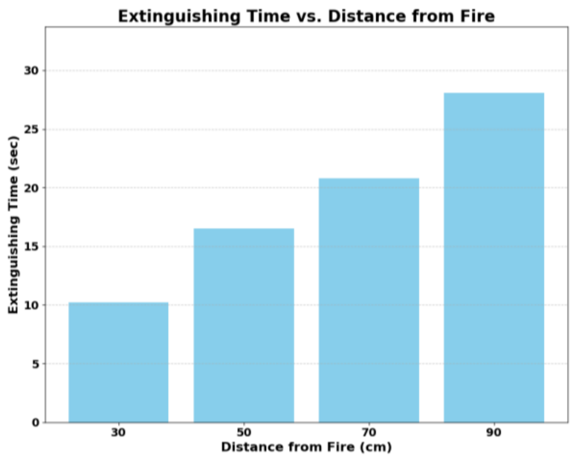

Fig 13: Extinguishing Time vs. Distance from Fire

At 30 cm, the robot extinguished the fire in 10.2 seconds. At 50 cm, the time increased to 16.5 seconds. At 70 cm, it took 20.8 seconds. At 90 cm, the extinguishing time was 28.1 seconds. These results are shown in Figure 13.

The data show that the robot remains effective even at longer distances. The increase in time is due to the reduction in water pressure over distance. However, the robot's precise nozzle alignment helped maintain high efficiency.

Comparative studies support these findings. Gao et al. (2019) reported an average extinguishing time of 38.87 seconds [31]. The proposed system is about 51% faster. This is due to servo-based nozzle control and adaptive water spraying. Sani et al. (2021) reported times between 9 to 21 seconds at 30 to 50 cm [32]. Our robot performed within 10 to 30 seconds at 30 to 90 cm. This shows better performance at larger distances.

The robot's nozzle angle optimization and servo control helped reduce water loss. The system also adapts to small flame movements. This improves the extinguishing process..

5.3 Operational Stability and Performance Evaluation

The robot was tested in five separate trials. Each trial had 50 operations, totaling 250 runs. Each run was evaluated for success or failure. The success rate and failure causes were recorded.

The robot performed with a high success rate. It completed most operations without issues. The average success rate was 98.0%. Only four failures were observed. Table 1 presents the detailed results.

Table 1: Summary of Firefighter Robot Tests

Trial	Total Operations	Failures	Success Rate	Reasons for Failure
1	50	1	98.0%	Sensor noise spike
2	50	0	100.0%	—
3	50	2	96.0%	Servo response delay
4	50	1	98.0%	Relay mome-ntary error
5	50	0	100.0%	—
Average			98.0%	

These results show that the robot is stable and reliable. The low failure count supports its readiness for real applications.

Other studies offer limited data on performance. Edozie et al. (2024) mostly described system design without trial data [33]. Sathiabalan et al. (2021) reported general performance without clear metrics [34]. In contrast, this study provides a detailed evaluation with real-world testing. The robot shows strong performance with minimal errors, proving its robustness and effectiveness in fire emergencies.

6. CONCLUSION

This paper presented an autonomous Arduino-based firefighter robot designed to improve laboratory fire safety through real-time flame detection, accurate nozzle control, and fully automated fire suppression. The system integrates a 180-degree IR flame sensor array, a servo-controlled water spray mechanism, and a microcontroller-driven control unit to detect and extinguish flames without human intervention.

Experimental validation under varying distances and flame intensities demonstrated the system's reliability and efficiency. The robot achieved an average flame detection accuracy of

95.8% and extinguished fires at distances up to 90 cm, with an average extinguishing time of 18.9 seconds, and also its dynamic servo alignment and real-time sensor integration. Furthermore, it achieved a 98.0% operational success rate across 250 trials, confirming its system stability and repeatability. The developed robot provides a cost-effective, autonomous, and scalable solution for real-world laboratory fire risks.

Overall, the system offers a deployable and robust platform for fire detection and suppression in indoor hazardous environments. Future developments may include AI-enhanced fire behavior prediction, multi-sensor fusion for obstacle avoidance, and cloud-based data logging for incident analytics. Integration with edge computing could further reduce response latency and enable smart coordination in multi-agent firefighting systems. The results of this study demonstrate the practical viability of autonomous robotics in reducing fire-related hazards and advancing laboratory safety protocols.

7. REFERENCES

[1] Fire & Safety Journal Americas. (n.d.). *NFPA discusses laboratories and their fire safety challenges.* Retrieved June 17, 2025, from https://fireandsafetyjournalamericas.com/nfpa-discuss-laboratories-and-their-fire-safety-challenges.

[2] Cheng-Chan, S., Horng, R. S., & Shin-Ku, L. (2016). Investigation of lab fire prevention management system of combining root cause analysis and analytic hierarchy process with event tree analysis. Mathematical Problems in Engineering, 2016, Article ID 3161823. https://doi.org/10.1155/2016/3161823.

[3] Hafizul Imran, M., Ziaul Haque Zim, M., & Ahmmed, M. (2021). PIRATE: Design and implementation of pipe inspection robot. In R. R. Samin & R. Ghazali (Eds.), *Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2020* (pp. 77–88). Springer Singapore. https://doi.org/10.1007/978-981-16-0586-4_7.

[4] Bai, M., Liu, Y., Qi, M., Roy, N., Shu, C.-M., Khan, F., & Zhao, D. (2022). Current status, challenges, and future directions of university laboratory safety in China. *Journal of Loss Prevention in the Process Industries*, 74, 104671. https://doi.org/10.1016/j.jlp.2021.104671.

[5] Hoeltge, G. A., Miller, A., Klein, B. R., & Hamlin, W. B. (1993). Accidental fires in clinical laboratories. Archives of Pathology & Laboratory Medicine, 117(12), 1200–1204. https://pubmed.ncbi.nlm.nih.gov/8250687/.

[6] Zainol, S. M., Daud, S., Ahmad, R. B., & Imran, M. H. (2024). A comprehensive systematic review: Sleep strategy on microcontroller for power management. *Journal of Advanced Research in Micro and Nano Engineering*, 23(1), 91–105. https://doi.org/10.37934/armne.23.1.91105

[7] Alsammak, I. L. H., Mahmoud, M. A., Aris, H., AlKilabi, M., & Mahdi, M. N. (2022). The use of swarms of unmanned aerial vehicles in mitigating area coverage challenges of forest-fire-extinguishing activities: A systematic literature review. Forests, 13(5), 811.

[8] Aizat, M., Kamarudin, K., Qistina, N., Heng, H., Imran, H., & Rahiman, W. (2025). Parameters tuning for enhanced automated guided vehicle navigation in ROS/Gazebo simulation environment. *Journal of Advanced Research in Applied Mechanics*, 1, 63–77. https://doi.org/10.37934/aram.133.1.6377

[9] Imran, M. H., Mahi, R. B., Saha, R., Islam, M. H., & Mahmud, I. (2022). NISHASH: A reasonable cost-effective mechanical ventilator for covid affected patients in Bangladesh. *Heliyon, 8*(5), e09375. https://doi.org/10.1016/j.heliyon.2022.e09375.

[10] Cakir, A., & Ezzulddin, N. F. E. (2016). Fire-extinguishing robot design by using Arduino. I*OSR Journal of Computer Engineering (IOSR-JCE)*, 18(6, Ver. V), 113–119. https://www.researchgate.net/publication/386869834.

[11] Kadam, K., Bidkar, A., Pimpale, V., Doke, D., & Patil, R. (2018). Fire fighting robot. *International Journal of Engineering and Computer Science*, *7*(1), 23383–23485. https://doi.org/10.18535/ijecs/v7i1.02.

[12] Taha, I. A., & Marhoon, H. M. (2018). Implementation of controlled robot for fire detection and extinguish to closed areas based on Arduino. *TELKOMNIKA*, *16*(2), 654–664. https://doi.org/10.12928/TELKOMNIKA.v16i2.8197 .

[13] Pan, T., & Zhu, Y. (2017). A fire-fighting robot using Arduino. In Designing embedded systems with Arduino (Chapter 7). *Springer*. https://doi.org/10.1007/978-981-10-4418-2_7.

[14] Diwanji, M., Hisvankar, S., & Khandelwal, C. (2019, September 28–29). Autonomous fire detecting and extinguishing robot. In *2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT)*. Manipal University Jaipur.

[15] Sharma, Y., & Singh, S. (2021). The fire extinguisher robot "FEBO". *International Research Journal of Engineering and Technology (IRJET)*, *8*(6). https://www.irjet.net/archives/V8/i6/IRJET-V8I6761.pdf.

[16] Sangewar, V. J., Thorat, A. S., Kholamkar, M. M., & Shinde, V. V. (2021). Design and development of multi-purpose fire extinguisher robot. *International Journal of Science and Research (IJSR)*, *10*(7). https://doi.org/10.21275/SR21705175854 .

[17] T, A., Ahmed, M. F., & Khan, F. U. (2023). Arduino controlled fire detection robot. *Technoarete Transactions on Advances in Computer Applications (TTACA)*, 2(2).

[18] Najm, N. M., Hussain, A. K., Mustafa, S. I., Rashit, B., & Lukashenka, V. (2024, April). Design and implementation of a robot firefighter for indoor applications. In *2024 35th Conference of Open Innovations Association (FRUCT)* (pp. 482–491). IEEE.

[19] Kucukdermenci, S., & Ilten, E. (2025). Arduino based fire extinguisher vehicle design and application. *International Journal of Advanced Natural Sciences and Engineering Researches*, *9*(2), 223–232. https://www.researchgate.net/publication/389206587.

[20] Elprocus. (n.d.). *Arduino basics and design*. Retrieved June 17, 2025, from https://www.elprocus.com/arduino-basics-and-design/.

[21] Arduino. (n.d.). *ReadAnalogVoltage*. Arduino Documentation. Retrieved June 18, 2025, from https://docs.arduino.cc/built-in-examples/basics/ReadAnalogVoltage/.

[22] Wikipedia contributors. (n.d.). *Sampling (signal processing)*. Wikipedia. Retrieved June 18, 2025, from https://en.wikipedia.org/wiki/Sampling_%28signal_processing%29.

[23] Arias, L., Torres, S., Sbarbaro, D., & Farias, O. (2008). Photodiode-based sensor for flame sensing and combustion-process monitoring. *Applied Optics, 47*(30), 5541–5549. https://doi.org/10.1364/AO.47.005541.

[24] Arias, L., Torres, S., Sbarbaro, D., & Farias, O. (2008). Photodiode-based sensor for flame sensing and combustion-process monitoring. *Applied Optics*, 47(29), 5541-5549.

[25] Nugraha, A. R. (2016). Fabrication and Characterization of Photodiode Flame Distance Sensor. https://doi.org/10.14203/INSTRUMENTASI.V39I1.65.

[26] Yu, J., Zhang, T., & Qian, J. (2011). Efficiency testing methods for centrifugal pumps. In J. Yu, T. Zhang, & J. Qian (Eds.), *Electrical motor products* (pp. 125–172). Woodhead Publishing. https://doi.org/10.1533/9780857093813.125 .

[27] Careers360. (n.d.). *Fleming's left-hand rule and right-hand rule*. Careers360. Retrieved June 17, 2025, from https://www.careers360.com/physics/flemings-left-hand-rule-and-right-hand-rule-topic-pge.

[28] L298N DC stepper driver Arduino tutorial. (n.d.). *Last Minute Engineers.* Retrieved June 18, 2025, from https://lastminuteengineers.com/l298n-dc-stepper-driver-arduino-tutorial.

[29] Ding, W., Yang, T., Li, J., Hua, C., & Mu, D. (2025). A real-time flame detection and situation assessment algorithm for firefighting robots. *Fire Technology, 61*. https://doi.org/10.1007/s10694-024-01698-3.

[30] Ahamed, S., MS, S., & Sampangi, S. (2024). Automated fire sensing and fire extinguisher module. *International Journal of Innovative Science and Modern Engineering, 12*(11), 11–16

[31] Gao, S., Zha, Z., Zhao, Z., & Jamali, M. M. (2018). Vision and infra-red sensor based fire fighting robot. *2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS)*, 873–876. https://doi.org/10.1109/MWSCAS.2018.8624080 .

[32] Sani, M. A. A., Sani, N. S., Yusof, M. I., & Zainal, A. (2019). Development of fire fighting robot (QRob). *International Journal of Advanced Computer Science and Applications,* 10(1), 142–147

[33] Edozie, E., Twijukye, D., Okafor, W., & Eze, V. (2024). Design and validation of advancing autonomous firefighting robot. *KIU Journal of Science, Engineering and Technology, 3*(1), 56–62. https://doi.org/10.59568/KJSET-2024-3-1-06.

[34] Sathiabalan, N., Lokimi, A., Ong, J., Hasrin, N., Md Zain, A. S., Ramli, N., Zakaria, H., Firuz, S., Mohd Hashim, N. B., & Taib, M. (2021). Autonomous robotic fire detection and extinguishing system. *Journal of Physics: Conference Series, 2107*, 012060.

[35] Imran, M. H., Shaha, R., Mahi, R. B., & Ujjaman, M. R. (2021). Vertical axis wind turbine: A novel approach to development and modeling. *International Journal of Computer Applications,* 183(18), 25–30. http://dx.doi.org/10.5120/ijca2021921405.

[36] Bhuiyan, T., Ahmed, M. S., Arman, M. S., Tasnim, N., Imran, M. H., & Smmak, M. (2025). Rooftops detection with YOLOv8 from aerial imagery and a brief review on rooftop photovoltaic potential assessment. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 14(3), 2282–2290. http://doi.org/10.11591/ijai.v14.i3.pp2282-2290.

[37] Mahi, M. J. N., Chaki, S., Humayun, E., Imran, H., Barros, A., & Whaiduzzaman, M. (2023). A review on VANET security: Future challenges and open issues. Indonesian Journal of Electrical Engineering and Informatics (IJEEI), 11(1), 180-193. http://doi.org/10.52549/ijeei.v11i1.4295

[38] Aizat, M., Kamarudin, K., Qistina, N., Heng, H., Imran, H., & Rahiman, W. Parameters Tuning for Enhanced Automated Guided Vehicle Navigation in ROS/Gazebo Simulation Environment. http://doi.org/10.37934/aram.133.1.6377

[39] Asha, M. L. A., Rafi, M. A., Ahamed, M. S., & Imran, M. H. (2024, March). Suggesting Playlist and Playing Preferred Music Based on Emotion from Facial Expression. In 2024 3rd International Conference for Innovation in Technology (INOCON) (pp. 1-5). IEEE. http://doi.org/10.1109/INOCON60754.2024.10511575

[40] Islam, T., Imran, M. H., Hossain, M. R., Monshi, M. T., Himu, H. D., Rahman, M. A., & Surjo, G. S. (2022). Deep Learning Approaches to Predict Future Frames in Videos. Int. J. Recent Contributions Eng. Sci. IT, 10(3), 63-79. http://doi.org/10.3991/ijes.v10i03.33893