PILLAI COLLEGE OF ENGINEERING

CELEBRATING 1999 2024 25 YEARS

DEPARTMENT JOURNAL

Electronics And Telecommunication Engineering

AUTOMATION

DEC 2024 - MAY 2025

FROM PRINCIPAL'S DESK

I am happy to share the continuing efforts of faculty and students in the Electronics and Telecommunication department for their collaborative work in producing an innovative journal. This publication serves as an outstanding representation of the various technological advancements in the field of electronics and telecommunication.

In the fast-paced world of engineering today, ongoing exploration and progress are essential, requiring the utmost creativity. To reach significant objectives, the current generation must discover fresh and inventive methods of learning. I am dedicated to fostering an environment that encourages both academic excellence and creative exploration for our students.

The journal has exceeded my expectations, showcasing the commitment and skill present within the department. I wish the staff and students continued success in all their future pursuits.

Dr. Sandeep Joshi
Principal
Pillai College of Engineering

FROM HOD'S DESK

I am very happy to publish the issues of Department Journal for AY 2024 25 defining characteristic of Pillai College of Engineering, and I take immense pride in the Electronics and Telecommunication department as we unveil the latest issue of our departmental journal. The insightful articles within provide a comprehensive look at the diverse interests present in our department, highlighting the creativity and originality of both our students and faculty.

I would like to express my sincere gratitude to the students for their remarkable commitment and exceptional contributions. Their ability to develop impressive projects while prioritizing cost-effectiveness exemplifies a vital aspect of sustainable practices.

The papers featured in the journal stand as a testament to the hard work and dedication of our students. I commend the journal committee for their efforts in ensuring that the valuable contributions of our students are recognized. As our students embark on various journeys in life, I encourage them to continue exploring innovative solutions.

Wishing all students the very best in their future endeavors.

Dr. Avinash Vaidya
Head of Department
Pillai College of Engineering

FROM TEACHER'S DESK

The annual journal of the Electronics and Telecommunication Department provides an excellent platform for our students to showcase their innovative ideas and proposals. It not only highlights their academic and technical strengths but also serves as a valuable resource for peers and future learners. The efforts of the Journal Committee are truly commendable. Their meticulous planning, organization, and guidance throughout the article-writing process greatly contributed to the success of this edition. Their commitment and timeliness deserve special recognition.

The technical papers featured this year are a reflection of the creativity, insight, and forward-thinking mindset of our students—a promising sign for the future.

I wholeheartedly congratulate the committee and extend my best wishes. I look forward to many more impressive editions of this journal in the years ahead.

Prof.Jayashri Bhosale
Assistant Professor
Electronics and Telecommunication Engineering

VISION AND MISSION

INSTITUTE VISION :

Pillai College of Engineering (PCE) will admit, educate and train in technology, a diverse population of students who are academically prepared to benefit from the Institute's infrastructure and faculty experience, to become responsible professionals. It will further attract, develop and retain, dedicated, excellent teachers, scholars, scientists and professionals from diverse backgrounds whose work gives them visibility beyond the classroom and who are committed to making a significant impact in the lives of their students and the community.

INSTITUTE MISSION:

To develop professional engineers with respect for the environment and to make them responsible citizens, both from a local and global perspective. This objective is fulfilled through quality education, practical training, research, entrepreneurship and interaction with industries and social organizations.

DEPARTMENT VISION:

Strive towards producing world class engineers who will continuously innovate, upgrade telecommunication technology and provide advanced, hazard-free solutions to the mankind.

Inspire, educate and empower students to ensure green and sustainable society.

DEPARTMENT MISSION

Benchmarking against technologically sound global telecommunication institutions with a view towards continuous improvement.

Continually exposing students to scenarios that demand structuring of complex problems and proposing solutions.

Educate students and promote values that can prevent further degradation of our planet. Becoming responsible citizens genuinely concerned with and capable of contributing to a just and peaceful world.

INSTITUTE OBJECTIVE

To evolve a teaching-learning process where by students have freedom of thought and thereby explore the various aspects of technological development.
To encourage the teaching faculty to pursue research in specialized and emerging areas of technology and develop their skills to impart that knowledge to students.
To provide the requisite infrastructure including workshops, Library and IT laboratories to create a learning environment.
To promote the interaction of students and faculty with the industry.
To collaborate with other leading institutions in India and overseas to gain improvement for the lives of all global citizens.

DEPARTMENT OBJECTIVE & OUTCOME
PROGRAM EDUCATIONAL OBJECTIVE:

Provide graduates with a strong foundation in mathematics, science and engineering fundamentals to enable them to analyze and solve challenging problems in Electronics and Telecommunication Engineering.
Impart analytic and thinking skills to develop innovative ideas in the field of Telecommunication Engineering.
To keep students up to date with the latest advancement in the field of Electronics and Telecommunicatio.Inculcate qualities of leadership skills, multi-disciplinary team work and an ability to adapt to evolving professional environment in the field of Engineering and Technology.
To create awareness among the students towards ethical, social and environmental issues in the professional career.

PROGRAM SPECIFIC OUTCOMES:

Able to understand the concept of Basic Electronics, Network and Circuit Analysis, Analog and Digital circuits, Signals and System, Electro magnetics and apply them in various areas like Microwave Engineering, Wireless Communication, Digital image processing, Advance Communication systems etc.
Able to use techniques, skills, software, equipments and modern engineering tools necessary for Electronics and Telecommunication Engineers to identify, formulate and solve problems in industries and research work.
Able to work in multi disciplinary environment to provide socially acceptable technical solutions for complex communication engineering problems.

INDEX

UNDERWATER REMOTELY OPERATED VEHICLE

[1]Prasad Bhor, [1]Shreedhar Padiyachi, [1]Achintya Pednekar, [1]Gaurav Yadav
[2]Dr.Suman Wadkar
(1- Student of Department of Electronics and Telecommunication Engineering, Pillai College of Engineering, New Panvel ; 2- Professor of Department of Electronics and Telecommunication Engineering, Pillai College of Engineering, New Panvel)

Abstract— This paper presents the design, development, and implementation of a cost-effective remotely operated vehicle (ROV) for underwater structure inspection, focusing on the detection of structural anomalies and environmental monitoring. Traditional inspection methods, such as deploying human divers, Magnetic Particle Testing (MT), and visual or tactile inspections, are expensive, time-consuming, and often hazardous. Additionally, these methods prove ineffective in turbid water conditions. To address these challenges, we propose a solution integrating pH and turbidity sensors, along with a camera for crack & algae detection using Artificial Intelligence and structure monitoring. The ROV is powered by a Raspberry Pi for real-time control, data acquisition, and persistent storage, ensuring efficient system management. It also supports real-time video streaming, enabling remote monitoring and decision-making in dynamic underwater environments. This modular, scalable, and affordable solution offers a safer and more reliable alternative to conventional methods, making it suitable for applications such as dam inspections, water quality assessments, and structural health monitoring.

Keywords— underwater surveillance, remotely operated vehicle (ROV), Raspberry Pi, pH sensor, turbidity sensor, crack detection, algae monitoring, artificial intelligence, deep learning underwater structure maintenance, real-time data, live streaming, multimodal prediction system

I. INTRODUCTION

Background: Underwater inspections are vital for maintaining the safety of critical infrastructure, but traditional methods like diver inspections and sonar are often expensive and limited in accuracy. Advances in ROVs with integrated sensors offer more effective solutions for real-time monitoring and damage detection.

Problem Statement: Current inspection methods are costly and lack real-time data collection for detecting corrosion or structural issues. There is a need for a versatile, cost-efficient ROV that can enhance inspirations with advanced sensors.

Objectives: This project aims to develop an ROV for underwater inspections, using a Raspberry Pi 4B for control and data processing. The system incorporates pH, pressure, and turbidity sensors to assess corrosion risks, along with image processing tools to detect cracks and algae.

Contributions: The proposed ROV provides real-time monitoring and efficient inspection capabilities, offering an affordable solution for industries like oil and gas. It improves safety, efficiency, and maintenance of underwater infrastructure.

II. LITERATURE REVIEW

A. Previous Work

Technological advancements in Remotely Operated Vehicles (ROVs) have significantly enhanced underwater inspection methods, especially in areas like crack detection, sensor integration, and real-time monitoring. For example, A. Gengren Wang demonstrated how ROVs equipped with image processing can detect underwater structural cracks [5], while Muniyandy Elangovan emphasized the role of ROVs in improving safety during dam inspections [4]. The SMART-ROV developed by UTeM advanced ROV design by improving size, depth capacity, and propulsion, making them valuable across industries like oil, gas, and marine research. The recent work by Jyothikrishna et al. (2023) presents modern ROVs equipped with advanced features such as 1080p live feeds, vector thrusters, and multiple control modes for better underwater navigation. They also suggest the implementation of SONAR sensors for underwater imaging, which could improve operations in lowlight conditions [1].

Recent innovations in Unmanned Underwater Vehicles (UUVs) have further advanced underwater inspection capabilities. Ioannou et al. (2024) highlighted the increasing autonomy of UUVs, which replace human divers in hazardous environments [2]. They emphasized integrating sensors like sonar and cameras to improve real-time underwater monitoring. Similarly, Ventura et al. (2023) reviewed sensor fusion techniques, which combine multiple data sources to improve inspection accuracy and efficiency

B. Gap Identification

Despite these advancements, there are still unresolved challenges in ROV and UUV technology. One key issue is maintaining balance due to ethernet cable connections, which can impact maneuverability. Most existing research also focuses on specific technical problems, such as crack detection or propulsion, but few offer fully integrated systems combining sensor arrays, imaging, and real-time data analysis.

Furthermore, while ROVs are being developed for various applications, such as ship hull inspection and deep-sea exploration [1] limitations persist in terms of autonomy and sensor integration in turbulent, deep-water environments. More research is needed to enhance stability, independence, and real-time data processing using advanced sensors like saltwater and dissolved oxygen detectors [2].

III. METHODOLOGY

A. System Architecture Overview

The Remotely Operated Vehicle (ROV) developed for underwater inspection integrates both hardware and software components to enable real-time object detection and environmental monitoring.

1. Microcontroller (Raspberry Pi 4):

   The Raspberry Pi acts as the central processor, managing motor control, sensor inputs, and running the YOLO detection algorithm, ensuring smooth data communication between components.

2. Camera for Visual Feedback and Object Detection:

   A high-resolution camera provides live video, which the Raspberry Pi processes to detect objects like cracks, algae, and debris, supporting operators in decision-making.

3. Motors and Thrusters for Movement:

   BLDC motors attached to thrusters enable movement in all directions, with joystick control allowing for smooth manual operation.

4. Environmental Sensors (pH and Turbidity):

   A pH sensor measures water acidity, and a turbidity sensor assesses water clarity, helping detect potential environmental hazards.

5. Communication with the ROV:

   The controller box includes a joystick, ADC, and a potentiometer for speed adjustment. A wired Ethernet connection ensures low-latency data transmission for precise, real-time control during operations.

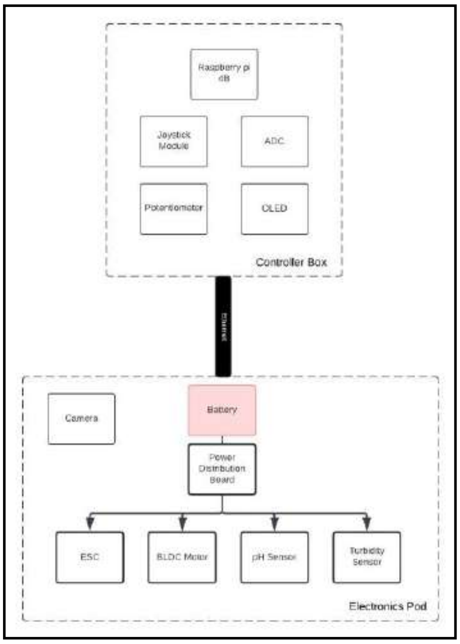


Fig 1. System Architecture

B. Hardware Components and Integration

This project integrates a variety of hardware components, each carefully selected to create a robust system for monitoring and analyzing underwater structures. Below is a comprehensive overview of each component, including their technical specifications and their role within the overall architecture of the system.

TABLE I: Hardware Specifications

Component	Description
Brushless Direct Current Motor (BLDC)	1000 kV, Voltage- 5-12V
Power Distribution Board	Distributes power to components for optimized efficiency
Electronic Speed Controller (ESC)	Controls motor speed based on joystick input
Lithium-Ion Battery	5000 mAh for extended operation
pH Sensor	Monitors acidity/alkalinity for structure inspection
Turbidity Sensor	Measures water clarity to detect potential corrosion
MCP3008 ADC	8-channel analog-to-digital converter
Joystick Sensor	Controls ROV movement via potentiometer
USB Camera	CMOS 640x480 with collapsible cable
OLED Display	0.96-inch for live data display

C. Software Framework he software framework in this project enables real-time data processing and decision-making through several core components:

- YOLOv5s for Object Detection:

  The YOLOv5s algorithm detects structural issues such as cracks and algae in real-time. Trained on a Pascal VOC dataset from Kaggle using Google Colab's GPU, the model processes underwater images from the ROV's camera to identify anomalies efficiently.

- Live Video Streaming Integration:

  The ROV's camera feed streams live through the Raspberry Pi, overlaying real-time object detection and environmental metrics (pH and turbidity). Accessible from any networked device, this stream enhances visibility and collaboration during inspections.

- Data Processing and Multimodal Prediction Framework:

  This framework combines sensor data with crack detection results to assess structural maintenance needs. The integrated data visualization aids operators in making informed decisions and navigating complex environments effectively

D. Control System and Motor Functionality

The control system is designed to provide operators with intuitive navigation capabilities for the ROV. Key features include:

- Joystick Control:

  The joystick, based on the potentiometer principle, converts analog data to digital using an ADC. These signals are sent via Ethernet to the ESCs, which control motor speed and direction.

- Motor Configuration and Degrees of Freedom:

  The ROV has four BLDC motors—two for forward, backward, and lateral movements, and two for vertical motion. This setup offers six degrees of freedom (DoF), allowing smooth navigation in underwater environments.

- Power Distribution and Efficiency:

  A PDB ensures efficient power distribution to the motors. The ROV is powered by a 5000mAh lithium-ion battery, providing extended operation without frequent recharging.

- Self-Balancing Mechanism:

  Positive buoyancy tubes enhance stability, keeping the ROV upright and ensuring smooth camera operation and accurate sensor readings.

E. Monitoring Using Camera and Sensors

The ROV features a 640x480 USB camera for live video streaming, enabling real-time identification of cracks, algae, and debris. pH and turbidity sensors measure water quality, with data displayed alongside the video stream. Devices on the same network can access the live feed and sensor data for collaborative monitoring.

F. Evaluation Metrics and Performance Analysis

Performance is assessed using a confusion matrix and various graphs to visualize results. These metrics ensure accurate detection of cracks and algae, while sensor reliability is tested under controlled conditions to confirm consistent performance.

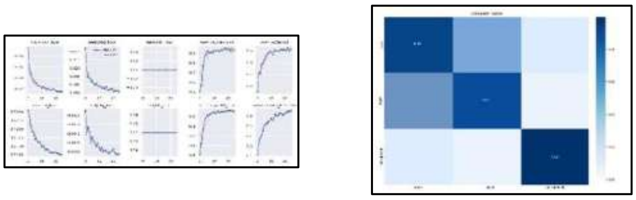


Fig 2. Evaluation Metrics of YOLOv5 model

G. Custom Hardware Design and Encapsulation

1. *Custom Controller Box with PCB Integration*: The controller box integrates a custom PCB with the joystick, ADC (MCP3008), and potentiometer, ensuring smooth motor control and precise speed adjustments. A wired Ethernet link provides lowlatency communication with the ESCs.

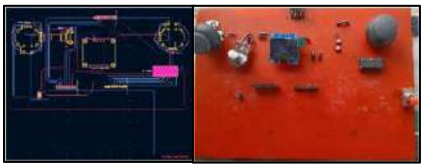


Fig 3. PCB Design and Fabrication

2. *Propeller and Thruster Optimization*: Custom propellers improve propulsion efficiency, ensuring smooth, stable movement and reduced drag in underwater environments.

Fig 4. Propeller and Endcaps

3. Encapsulation of Main Electronics Module: The Raspberry Pi, PDB, ESCs, and sensors are housed in a waterproof tube to protect against water damage. This compact design ensures durability and simplifies maintenance during operations.

IV. RESULTS AND DISCUSSION

The ROV's performance in underwater monitoring was promising, with smooth hardware and software integration.

A. Navigation and Maneuverability

The ROV navigated smoothly in all directions (forward, backward, lateral, and vertical) with stable joystick control. Positive buoyancy tubes ensured self-balancing.

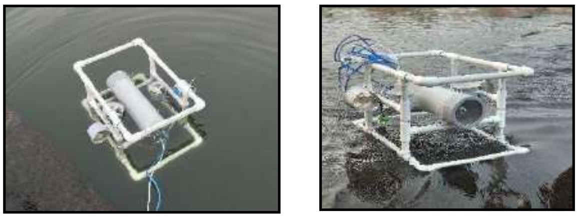


Fig 4. ROV Assembly and Setup

B. Object Detection Performance

Using YOLOv5s, the ROV accurately detected cracks and algae in real-time, supporting effective maintenance of underwater structures.

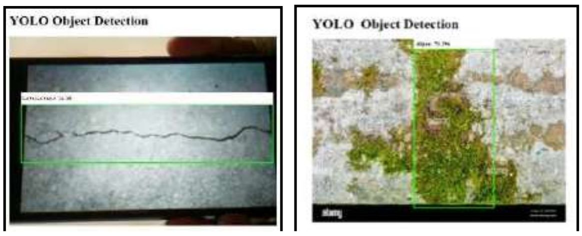


Fig 5. Object Detection

C. Sensor Monitoring

The pH and turbidity sensors were integrated to provide critical data during inspections of underwater industrial structures.

Fig 6. Sensor Data Visualization

V. FUTURE SCOPE

The development of the Remotely Operated Vehicle (ROV) presents several exciting opportunities for enhancement. Future upgrades may include:

1. Increased Degrees of Freedom (DOF): Adding more motors would improve the ROV's maneuverability, enabling it to perform complex movements and navigate intricate underwater environments more effectively.
2. Integration of a Robotic Arm: A robotic arm would allow the ROV to collect samples directly from underwater structures, supporting more extensive research and data collection.
3. Surface Buoy for Communication: A surface buoy linked to the ROV via a wired connection could create a wireless communication channel with the controller, expanding the ROV's operational range and improving data transmission reliability.

VI. CONCLUSION

This project successfully developed an underwater inspection system using a Remotely Operated Vehicle (ROV) equipped with a YOLO-based object detection framework and environmental sensors for pH and turbidity. The system efficiently identifies structural cracks and monitors environmental conditions in real time, contributing to underwater infrastructure maintenance. Results show that the system can operate effectively in challenging environments, providing reliable detection with minimal human intervention.

VII. ACKNOWLEDGMENT

We gratefully acknowledge the guidance and support of Dr. Sandeep Joshi, Principal of the college, and Dr. Avinash Vaidya, Head of the Department, whose encouragement was invaluable to our project. Special thanks to our project guide, Dr. Suman Wadkar, for their continuous guidance and insightful feedback, which were crucial in shaping this work. We also extend our appreciation to Pillai College of Engineering, Department of Electronics and Telecommunication, for their support throughout this endeavor.

VIII. REFERENCES

[1] Jothikrishna K, Rithika S M, Swetha S V, Kavitha K, "REMOTELY OPERATED UNDERWATER VEHICLE (ROV)", 2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation, DOI: 10.1109/ICAECA56562.2023.10200265

[2] Giorgio Ioannou, Nicola Forti, Alfredo Renga, Giuseppe Tomasicchio, Paolo Braca, "Underwater Inspection and Monitoring: Technologies for Autonomous Operations" , DOI. No. 10.1109/MAES.2024.3366144

[3] Guojun Chen ,Wei Chen, Zhiming Wang,Tiezheng Guo, Ximing Xia ,and Lijun Xu, " Design and Dynamic Performance Research of UnderwaterInspection Robots " ,Wireless Communications and Mobile Computing Volume 2022, Article ID 3715514 (JOURNAL)

[4] Muniyandy Elangovan,Siddarth B Yuvan, Govindram R Uduupa, Mohan M Babu, " Design and Development of Underwater ROV for Dam Inspection " , Frontiers in Marine Science published 03 April 2019 doi:10.3389/fmars.2019.00134 (JOURNAL)

[5] Pengfei Shi, Xinnan Fan1, Jianjun Ni1 and Gengren Wang, "A detection and classification approach for underwater dam cracks" Structural Health Monitoring 2022, Vol.15(5) 541–554 (JOURNAL)

The Pick And Place Robot

[1]Shreyash Zope, [1]Anuj Wani, [1]Aditya Shinde, [1]Adinath Chavan, [2]Prof. Shweta Waghmare

(1- Student at Department of Electronics and Telecommunication Engineering, Pillai College of Engineering, New Panvel; 2- Professor at Department of Electronics and Telecommunication Engineering, Pillai College of Engineering, New Panvel)

Abstract — Pick and place robots are crucial in industrial automation, ensuring precise and efficient object handling with minimal human intervention. This research presents the design and implementation of a Raspberry Pi-based robotic arm for pick-and-place operations. The system integrates servo motors, motor drivers, sensors, and a web-based user interface for real-time control and precision in object placement. By incorporating these elements, the system enhances operational efficiency, minimizes human errors, and improves reliability in industrial and logistic applications. The advancements in robotic automation showcased in this project provide a scalable and adaptable solution to modern industrial challenges.

Keywords — Pick and Place, Raspberry Pi, Robotics, Automation, Industrial Applications

I. INTRODUCTION

Industrial automation has significantly improved manufacturing and logistics, reducing manual effort and increasing productivity. Pick and place robots play a vital role in such environments, handling repetitive tasks like assembly, sorting, and material handling. These robots operate with high precision and efficiency, making them indispensable in modern industries where speed and accuracy are critical for maintaining competitive advantage.

The problem addressed in this study is the inefficiency and human errors associated with manual object handling in industries. Operators performing repetitive tasks are prone to fatigue, leading to decreased productivity and increased chances of errors. An automated robotic arm provides a solution by offering high-speed and accurate pick-and place operations, reducing human workload, and enhancing overall production output. The primary objectives of this research include designing and developing a pick-and place robot using Raspberry Pi, implementing a control algorithm for efficient object manipulation, integrating sensors for accurate positioning, and developing a web-based user interface for real-time control. The integration of these technologies enables seamless automation of object handling tasks across various industrial applications.

II. RELATED WORK

A. Problem Definition:

The inefficiencies of manual object handling in industrial applications have created a need for automation to improve precision, speed, and reliability. Manual handling often leads to inconsistencies, fatigue-related errors, and increased production time. The proposed system aims to address these issues by developing a robotic arm that automates the pick-and-place process. By implementing a controlled system using Raspberry Pi, the robot can perform repetitive tasks with higher accuracy and reduced operational risks, ensuring increased efficiency in industrial settings.

B. Objective:

The primary objective of this research is to design and implement a pick-and-place robotic arm that enhances industrial automation. The system is intended to provide a web-based control mechanism for real-time operation, allowing remote handling of objects. Additionally, the research focuses on integrating sensors for precise object detection, ensuring that the robot can accurately locate, grip, and place objects without error. By incorporating weight-based sorting mechanisms, the system improves adaptability and usability in diverse industrial applications. The overall goal is to minimize human intervention, reduce operational costs, and improve efficiency by leveraging automation.

C. Literature Review:

Several studies have explored the role of pick-and-place robots in automation, each employing different control techniques and technologies. Voice-controlled pick-and place robots using Raspberry Pi have been implemented, allowing users to command the robotic arm through voice recognition. However, these systems face challenges such as misinterpretation of commands in noisy environments, making them less reliable for industrial applications. Another approach utilizes real-time image processing with OpenCV to identify and track objects before manipulation. While effective, this method demands high computational power, which may not be feasible for embedded systems like Raspberry Pi. Additionally, research has been conducted on Android-controlled pick-and-place robots, enabling mobile-based control via Bluetooth or Wi-Fi. Despite being user-friendly, these systems depend on stable network connections and have latency concerns. More advanced research explores AI-driven autonomous robotic arms capable of identifying, sorting, and handling objects without human input. While this increases efficiency, it requires significant computational resources and sophisticated machine learning models. This research builds upon existing studies while addressing limitations such as response time, precision, and hardware feasibility in cost-sensitive industrial environments.

D. Block Diagram:

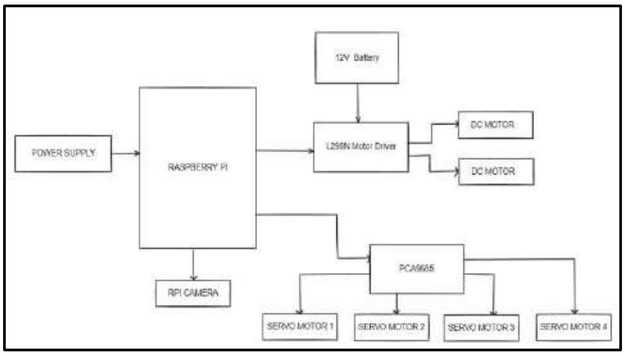


Fig 2.1. Block Diagram

The block diagram of the proposed system illustrates the interaction between key components, including the Raspberry Pi, motor drivers, sensors, and the web-based interface. It provides an overview of the system architecture, depicting how each component contributes to the functionality of the pick-and-place robot. The block diagram represents the signal flow and decision-making process, ensuring that each stage is optimized for efficiency and accuracy.

E. Circuit Diagram

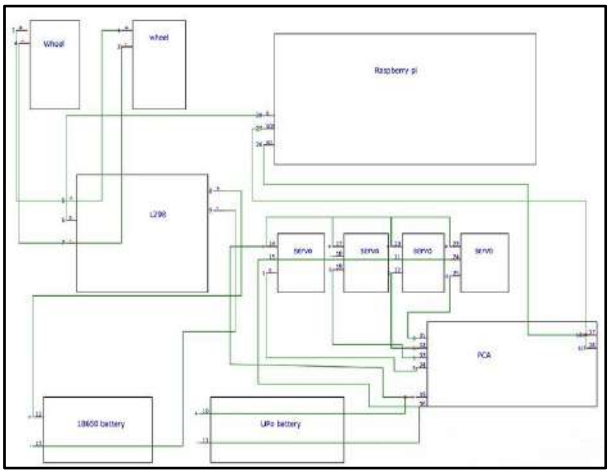


Fig 2.2 Circuit Diagram

The circuit diagram provides a detailed representation of the electrical connections within the system. It highlights how the Raspberry Pi interfaces with servo motors, ultrasonic sensors, and motor drivers to perform pick-and place tasks. Proper wiring and component placement ensure seamless communication between hardware

elements, minimizing power fluctuations and maintaining system stability. The circuit design prioritizes power efficiency and structural integrity, ensuring that the robotic arm operates without interruptions.

III. METODOLOGY

The development of the pick-and-place robot was executed through a structured and systematic approach, encompassing design, implementation, testing, and evaluation phases. This section outlines the methods and procedures used to achieve the project objectives, detailing the approach, data collection strategies, tools and techniques utilized, and common challenges encountered during the development process. By following this structured methodology, the proposed robotic arm achieves high efficiency, accuracy, and adaptability in industrial automation.

In the testing phase, the robotic arm was subjected to different environmental conditions and object types to assess its accuracy, speed, and response time. The web-based interface was tested for real-time responsiveness and command execution without significant delays. The evaluation phase focused on optimizing the system for better performance, reducing power consumption, and ensuring the reliability of the pick-and-place operations in an industrial environment.

A. Software Development:

The software architecture of the pick-and-place robotic system was developed using Raspberry Pi OS with Python as the primary programming language. The Raspberry Pi OS was installed and configured to support real-time processing for motor control, sensor data acquisition, and user interface management. Python scripts were developed to control servo motors, DC motors, and sensor modules, ensuring precise execution of pick-and-place operations. The Flask web framework was used to create a user-friendly web-based interface, enabling remote operation and real-time monitoring. Real-time data processing was implemented through algorithms designed to analyze ultrasonic sensor data for object detection, ensuring precise placement during robotic arm movements. Additionally, the Raspberry Pi Camera was integrated into the system to provide live streaming video feedback, enhancing user control and monitoring.

B. Communication System Setup:

A Wi-Fi-based communication system was implemented using the Raspberry Pi's built-in Wi-Fi module, enabling seamless data transmission between the robotic system and the operator's interface. A webpage-based control interface was designed, allowing users to operate the robot remotely. The interface consists of control buttons for moving the robotic platform forward, backward, left, and right, ensuring smooth navigation.

Above the control buttons, a live video feed from the Raspberry Pi Camera is displayed, providing real-time visual feedback to the operator. This low-latency video streaming is achieved using the Real-Time Streaming Protocol (RTSP), ensuring high-quality video transmission for precise object handling. Below the control buttons, four interactive slider knobs were integrated into the interface, each corresponding to one of the servo motors responsible for controlling different movements of the robotic arm. By sliding these knobs, users can adjust servo positions in real-time, allowing precise control over the gripper and arm movements.

The Flask web framework was used to develop the control interface, ensuring smooth interaction between the user and the robotic system. All commands are processed in real-time, minimizing response delays and enhancing the overall efficiency of the pick-and-place robot. This communication setup allows users to operate the robot remotely from any device with a web browser, making it a flexible and user-friendly automation system.

C. Testing and Validation:

The robotic system was tested in controlled environments to validate motor performance, sensor accuracy, and communication efficiency. Initial testing focused on ensuring the robotic arm's stability and verifying that servo motors functioned correctly for picking and placing objects. Sensor validation was conducted by measuring object detection accuracy using the ultrasonic sensor at different distances. The web interface was tested for real-time responsiveness, ensuring that user commands were executed without noticeable lag. The video streaming functionality was evaluated for frame rate consistency and clarity, which is critical for precise object placement. These controlled tests helped identify potential system optimizations, leading to improvements in power efficiency, motion accuracy, and real-time control stability.

D. Data Collection:

Data collection was integral to the development and evaluation phases of the robotic arm. The Raspberry Pi Camera provided continuous live-streaming video feeds, allowing operators to monitor and analyze real-time object placement. Ultrasonic sensor data was recorded to track object distances, helping refine movement precision. Motor control data, including servo angles and speed parameters, were logged to analyze system performance and fine-tune motion algorithms. The web-based interface recorded operator interactions, including control commands and response times, to assess usability and ensure an intuitive user experience. Additionally, power consumption data was logged to optimize battery efficiency and motor performance. These data insights were crucial for enhancing system reliability, reducing operational errors, and improving automation accuracy.

E. Hardware Tools:

The Raspberry Pi 3 served as the central processing unit, managing motor control, sensor data processing, and communication between the robotic arm and the web interface. Servo motors were used to execute precise arm movements, while DC motors with an L298N motor driver provided controlled movement for the mobile base. The PCA9685 servo driver was integrated to efficiently manage multiple servo motors. The ultrasonic sensor ensured accurate object detection, and the Raspberry Pi Camera facilitated real-time monitoring for remote operations.

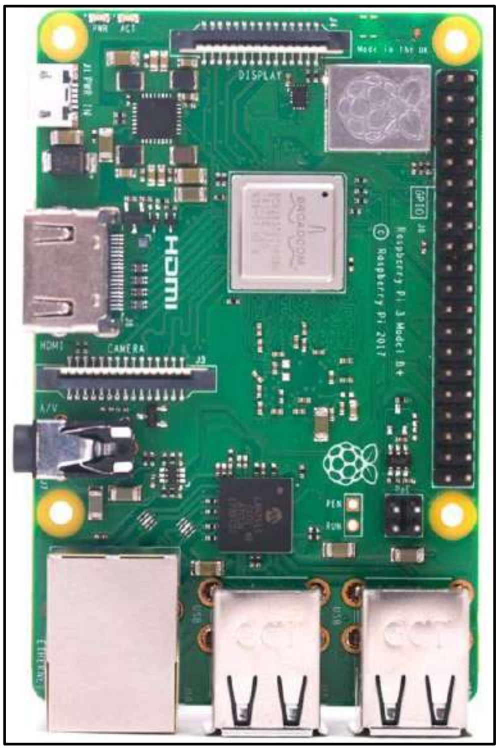
F. Software Tools:

Python was the primary programming language for developing motor control scripts, processing sensor data, and managing the web-based control interface. Flask was used to create the web dashboard, providing a seamless user experience for remote operation. RPi. GPIO and Adafruit PCA9685 libraries were utilized for servo motor control, while OpenCV was implemented for real-time image processing and live-streaming visualization.

G. Development Techniques:

An Agile development approach was used, allowing continuous testing and refinement of both hardware and software components. Multiple prototypes were built and tested, ensuring optimal servo performance, web interface usability, and sensor accuracy before finalizing the design. Simulation and modelling techniques were used to analyze robotic arm movement dynamics and improve positioning precision. Real-world testing scenarios helped fine-tune movement accuracy and ensure adaptability to different environments.

H. Future Enhancements:



Future upgrades include AI-based object recognition to enable autonomous sorting based on object type.
Additionally, the integration of force sensors in the gripper will enhance grip strength adjustment for handling delicate or heavier objects. Battery management optimization and power-efficient motor control algorithms will be implemented to improve system longevity and performance.

I. Common Mistakes and Improvements:
Several challenges were encountered during the system's development, requiring adjustments to enhance stability and efficiency. One significant issue was servo motor overheating, which was mitigated by implementing pulse width modulation (PWM) tuning and optimizing movement sequences. Wi-Fi latency in real-time control was initially high, but this was resolved by optimizing the Flask server's response handling and improving RTSP streaming efficiency. Sensor misalignment caused inaccurate object detection during initial tests; this was corrected by repositioning the ultrasonic sensor and refining calibration algorithms. Motor power fluctuations were addressed by integrating capacitors to stabilize voltage supply. Additionally, user interface complexity was reduced by focusing on essential controls, improving ease of use, and reducing operational errors.

IV. COMPONENTS

A. Raspberry Pi 3 Model b

Fig 4.1. Raspberry Pi 3 Board

The Raspberry Pi 3 serves as the central processing unit of the system, offering a 1.2 GHz quad-core ARM CortexA53 processor, 1GB RAM, built-in Wi-Fi, and Bluetooth connectivity. It is responsible for controlling the robotic arm, processing sensor data, and handling the web interface for remote operation.

B. Servo Motor:

Fig 4.2 Servo Motor

The servo motors used in the robotic arm provide precise angular control for joint movement. These motors operate on 4.8V to 6V and offer varying torque levels, ensuring accurate positioning of the gripper and arm sections.

C. PCA9685 servo driver:

The PCA9685 servo driver is used for controlling multiple servo motors efficiently. It supports 16 PWM channels and operates on 3.3V to 5V logic levels, allowing smooth and coordinated motion control of the robotic arm.

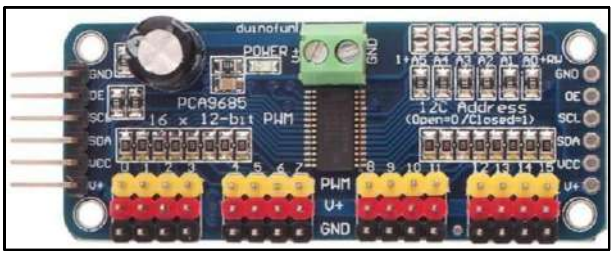


Fig 4.3. PCA9685 servo driver

D. L298N motor driver:

The L298N motor driver is employed for driving the DC motors that enable chassis movement. It supports 5V to 35V input voltage, allowing bidirectional control of motors with a current capacity of up to 2A per channel.

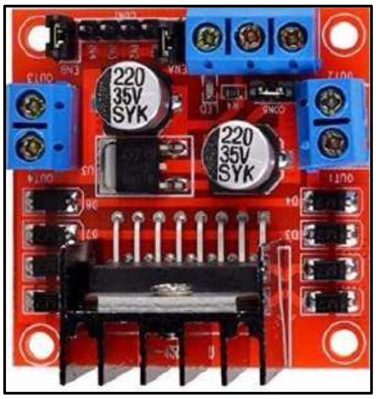


Fig 4.4. L298N motor driver

E. DC Motor:

Fig 4.5. DC Motor

The DC motors used in the robotic platform provide mobility and stability. These motors typically operate at 12v and have a speed range of 1000-3000 RPM, ensuring smooth navigation and movement of the robotic system.

F. Raspberry pi camera module

Fig 4.6. Raspberry pi camera module

The Raspberry Pi Camera Module is integrated for real-time video streaming and object detection. It features a 5MP sensor with a resolution of 2592 x 1944 pixels, supporting 1080p video recording and AI-based object recognition applications.

V. RESULTS

The fig 5.1 shows the entire system of the pick and place robot

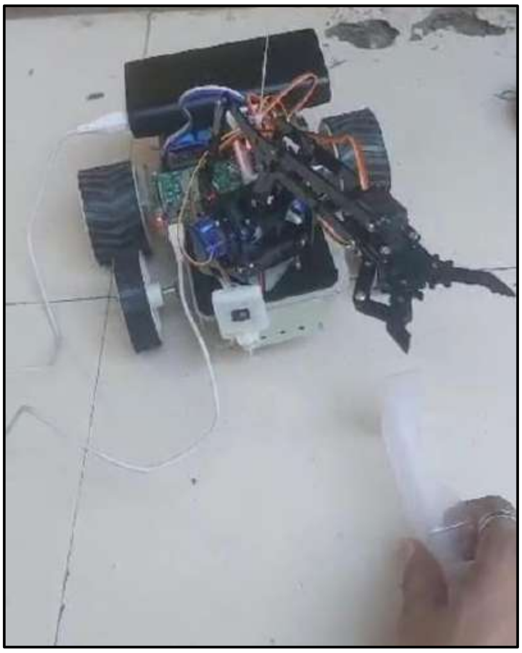


Fig 5.1. The pick and place robot

The fig 5.2 is the output when robotic arm is picking up the object

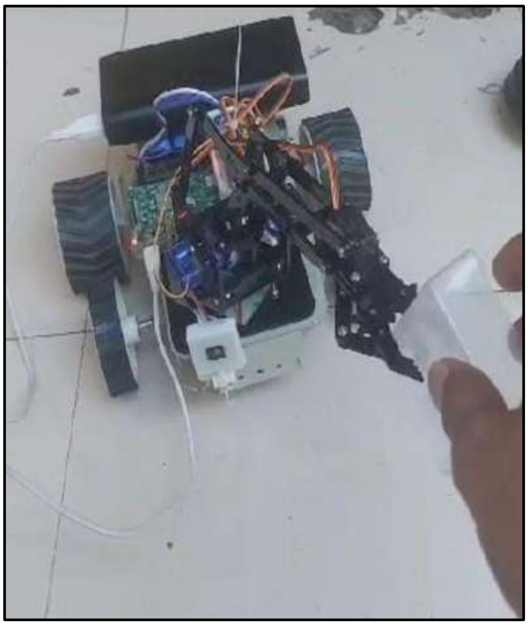


Fig 5.2. Arm picking an object

The fig 5.3 showcases webpage which is used to control robot

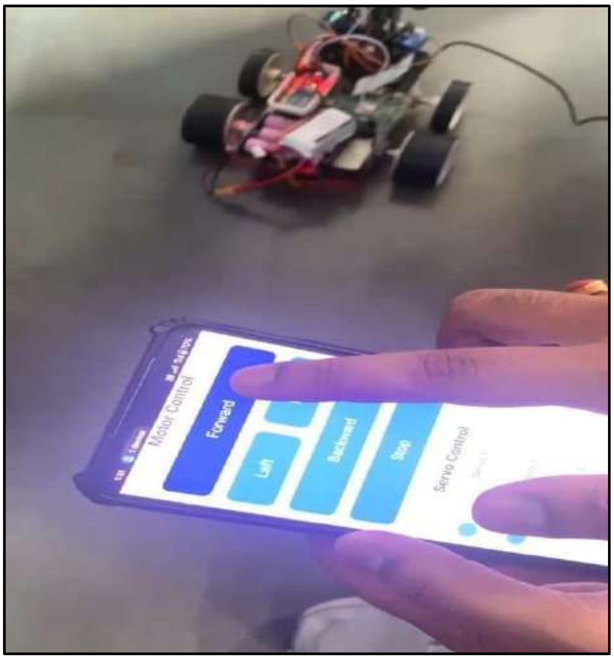


Fig 5.3. “FORWARD” is pressed on webpage

The fig 5.4 is the output when we can see live camera feed output on webpage

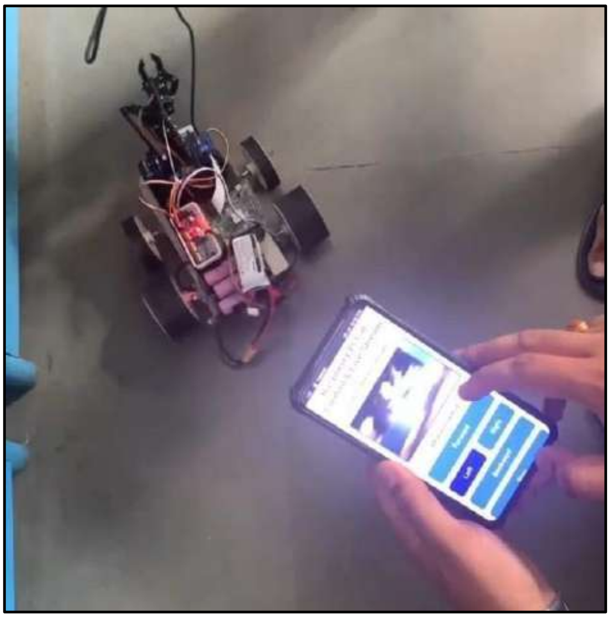


Fig 5.4. Live camera output

The fig 5.5 is the output when user is controlling robot via webpage

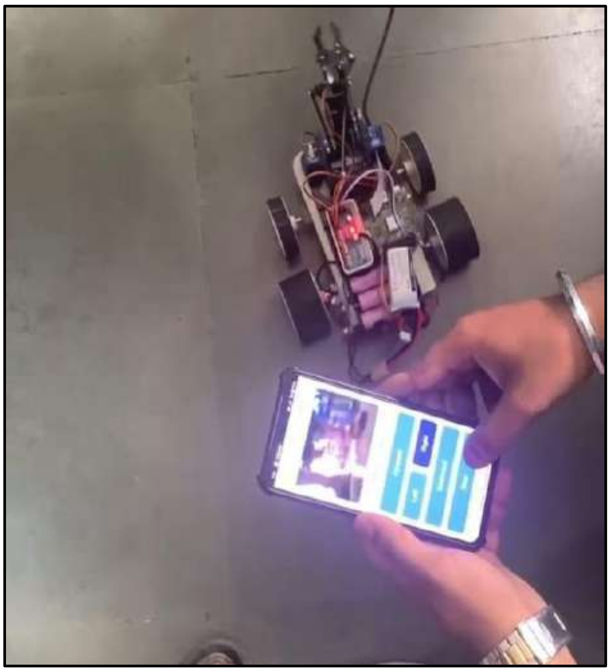


Fig 5.5. User controlling robot

VI. APPLICATIONS

A. Quality and Consistency:

Robots minimize errors and ensure uniform product quality. This benefits society by providing reliable products and is achieved technically through precise grippers, sensors, and consistent programming.

B. Safety and Risk Reduction:

Robots take over dangerous tasks, protecting human workers from harm. This is a direct social benefit, enabled by the robot's ability to operate in hazardous environments (a technical capability).

C. Flexibility and Adaptability:

By integrating health sensors, the system continuously monitors vitals like heart rate, enabling remote health monitoring. This real-time data can be sent to healthcare providers, reducing the need for constant physical visits while still ensuring patients receive timely interventions when necessary.

D. Cost-Effectiveness:

Although the initial investment can be high, robots lead to long-term cost savings through reduced labour costs, increased efficiency, and waste reduction. This improves operations efficiency. This blends social (job market considerations) and technical (optimized material usage through precise control) aspects.

E. Scalability and Automation of Complex Processes:

Pick-and-place robots can be integrated into larger automated systems, including conveyor belts, packaging machines, and quality control stations.
This technical aspect (integration with other automation) leads to social benefits by enabling the automation of entire production lines, further increasing efficiency and reducing reliance on manual labour.

VII. CONCLUSION

This paper presents the development of a Raspberry Pi-based pick-and-place robot designed for industrial automation. The system enhances efficiency, accuracy, and reliability in object manipulation tasks. Future enhancements will focus on AI integration for intelligent decision-making and expanding automation capabilities to further improve operational efficiency and adaptability in various industries.

ACKNOWLEDGMENT

We would like to take this opportunity to express our profound gratitude and deep appreciation to our project guide, Prof. Sweta Waghmare, for her invaluable guidance and constant encouragement throughout the duration of our project. Her insightful suggestions were instrumental in shaping the project and improving its execution. Her critical evaluation and expertise enabled us to enhance our project significantly. Working under her guidance has been a highly enriching and rewarding experience for us. We also extend our heartfelt thanks to the Department of Electronics and Telecommunication Engineering and our institution, Pillai College of Engineering, for providing the necessary resources, support, and facilities essential for the successful completion of this project. Additionally, we would like to express our gratitude towards our Head of Department, Dr. Avinash Vaidya, for his continuous encouragement and support throughout the project. We also sincerely thank our Principal, Dr. Sandeep Joshi, for facilitating an environment conducive to learning and innovation. Lastly, we are deeply grateful to our families and friends for their unwavering support and motivation, which played a crucial role in completing this project.

REFERENCES

[1] Manjunath Yadav, Gireesh M, and Suhas, "Voice Controlled Pick and Place Robot using Raspberry pi," International Research Journal of Engineering and Technology (IRJET), June 2015. (Describes a robotic system for picking and placing objects, controlled via voice commands using a Raspberry Pi. Focuses on the integration of voice recognition technology with robotic manipulation).

[2] P. Hemalatha, C.K. Hemantha Lakshmi, and Dr. S.A.K. Jilani, "Real time Image Processing based Robotic Arm Control Standalone System using Raspberry pi," SSRG International Journal of Electronics and Communication Engineering (SSRG-IJECE), August 2018. (Presents a Raspberry Pi-based system for controlling a robotic arm using real-time image processing. Details the use of OpenCV and MATLAB for object detection and arm control).

[3] Ms. Aiswarya Padman.G and Mr. Shon.J.Das, "Android Application Based Robotic Arm Control Using Raspberry Pi," International Journal of Scientific & Engineering Research, July 2020. (Details the development of a pick and-place robotic arm controlled by an Android application, communicating with a Raspberry Pi. Focuses on the mobile interface and wireless control aspects).

[4] Jorge Costa, Graça Almeida, and Armando Cordeiro, "Automated Pick and Place Solution Based on Raspberry Pi, Oct. 2021. (Presents a Raspberry Pi-based solution for automating pick-and-place tasks. Focuses on object identification and efficient robotic arm operation, including hardware integration).

[5] J. J. Craig, "Introduction to Robotics: Mechanics and Control," 3rd ed., Pearson Prentice Hall, 2005. (A foundational textbook covering the fundamental principles of robotics, including kinematics, dynamics, motion planning, and control. Provides a strong theoretical basis for understanding robot mechanics).

[6] S. Y. Nof, Ed., "Handbook of Industrial Robotics," 2nd ed., John Wiley & Sons, 1999. (A comprehensive reference handbook covering various aspects of industrial robotics, including design, implementation, applications, and emerging technologies. It provides detailed information on robot kinematics, dynamics, control, sensors, programming, and integration with manufacturing systems).

Design and Analysis of Beam Forming Antenna

[1]Harshit Tiwari, [1]Shraddha Dhayde, [1]Akshay Londhe, [1]Ritikesh Kaviatke, [2]Prof. Jayshri Bhosale

(1- Student of Department of Electronics and Telecommunication Engineering, Pillai College of Engineering, New Panvel ; 2- Professor of Department of Electronics and Telecommunication Engineering, Pillai College of Engineering, New Panvel)

Abstract: *This paper represents the design and characterization of 1×2 rectangular patch antenna array for 6.5GHz Satellite application. The antenna array consists of 2 rectangular patches designed on a 1.6mm thick FR-4 epoxy dielectric substrate with relative permittivity (εr) of 4.4 . An inset feed method is applied to feed each circular patch to improve overall performance of the antenna array. The physical parameter of proposed antenna array is analyzed through simulation to obtain the optimum performance satisfying the requirement of application. The experimental characterization is conducted for the realized antenna array to be compared with the simulation It shows that realized antenna array resonates at the frequency of 6.5 GHz with the gain of 8.39 dBi in which this is agreed very well with the simulation results.*

1. INTRODUCTION

Micro strip antenna consists of a very thin metallic strip placed on a ground plane with a di-electric material

In-between. The radiating element and feed lines are placed by the process of photo-etching on the dielectric material. Usually, the patch or micro-strip is chosen to be square, circular or rectangular in shape for the ease of analysis and fabrication. Micro strip antennas have profound applications especially in the field of satellite communications medical, military, and mobile. Micro strip antennas are very versatile in terms of resonant frequency, polarization, pattern and impedance at the particular patch shape. Their utilization has become diverse because of their small size and light weight. Rapid and cost effective fabrication is especially important when it comes to the prototyping of antennas for their performance evaluation. As wireless applications require more and more bandwidth, the demand for wideband antennas operating at higher frequencies becomes inevitable. Inherently micro strip antennas have narrow bandwidth and low efficiency and their performance greatly depends on the substrate parameters i.e. its dielectric constant, uniformity and loss tangent. The micro strip antennas are mostly a broadside radiator. The position of the feed has to be changed as before to control the input impedance.

The patch may be in a variety of shapes, but rectangular and circular are the most common because ease of analysis and fabrication, attractive radiation characteristics, especially low cross polarization. The patch is designed in such a way so that its pattern is maximum normal to it. End-fire radiator can also be chosen by proper mode selection. The micro strip patch consists of a metallic "patch" on top of the dielectric substrate and below the dielectric material it has ground plane.

2. ANTENNA DESIGN

Micro strip antennas are very flexible and are used in an array to synthesize a desired pattern which cannot be obtained with a single element. We use an array to extend the performance of the antenna, scan the radiation pattern beam of an antenna system, enhance the directivity, and gain which would be better compared to that of a single element. The resonant frequency of the antenna should be chosen appropriately. The resonant frequency chosen for our design is 6.5 GHz. The chosen value of the substrate (FR4) relative Dielectric constant (ε_r) is 4.4 and the substrate thickness (h) is 1.66 mm. Then, we evaluated the rectangular patch using the formula .

The following equation is used to determine the patch's width

$$W = \frac{C_0}{2f_r}\sqrt{\frac{2}{\varepsilon_r + 1}}$$

Where,

W = Width of the patch

C_0= Speed of light

ε_r= value of the dielectric substrate

Equation 1

The following equation is used to get the value of the effective dielectric constant.

$$\varepsilon_{reff} = \frac{\varepsilon_r + 1}{2} + \frac{\varepsilon_r - 1}{2}\left[1 + 12\frac{h}{W}\right]^{-1/2}, W/h > 1$$

Equation 2

The antenna's electrical size is enlarged by an amount of (ΔL) due to fringing. As a result, the actual increase in patch length (ΔL) must be determined using the equation below.

$$\frac{\Delta L}{h} = 0.412\frac{(\varepsilon_{reff} + 0.3)\left(\frac{W}{h} + 0.264\right)}{(\varepsilon_{reff} - 0.258)\left(\frac{W}{h} + 0.8\right)} \qquad L = \frac{C_0}{2f_r\sqrt{\varepsilon_{reff}}} - 2\Delta L$$

Equation 3

Parameter Table

Table 1. Dimentions of 1x2 beamforming antenna

Parameter	Values
L	14.044
w	10.39
h	1.635
t	0.035
εr	4.4
wf	3.08
Wf2	1.64
Wf3	0.7

3. HARDWARE AND EXPERIMENTAL CHARACTERIZATION:

This is the pictures of proposed 1×2 beamforming inset feed rectangular patch antenna array prototype which is realized based on the optimum design of simulation. The hardware realizations carried out through wet etching technique. An SMA connector is connected to the feeding network for experimental characterization. The characteristics of realized antenna arrays measured using Vector Network Analyzer. The measured results for reflection coefficient, voltage standing wave ratio(VSWR), gain, and radiation patterns are plotted.

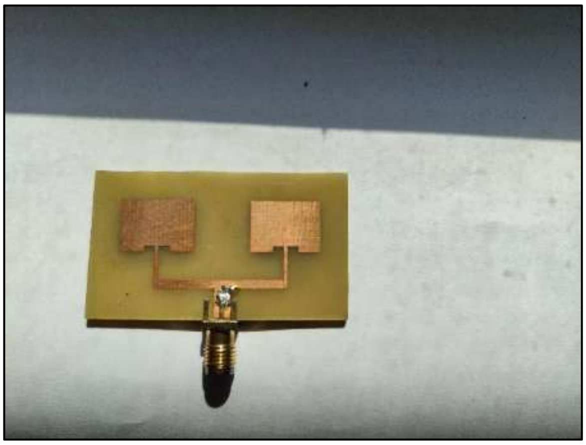


Fig 1. Prototype of proposed 1×2 Beam Forming Micro strip Fed (1*2)Antenna

4. RESULT OF PROPOSED BEAMFORMING ANTENNA

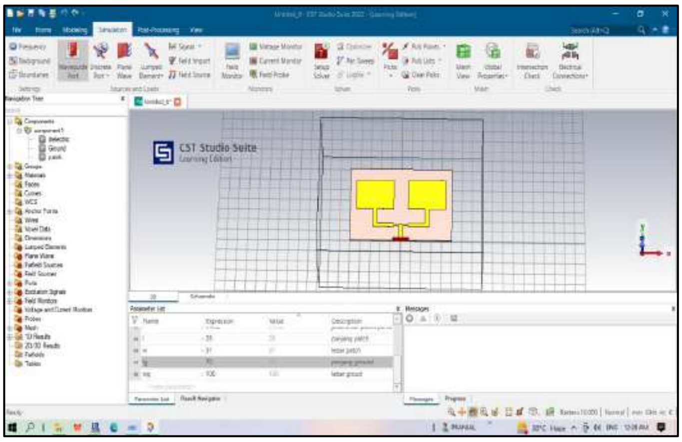


Fig. 3. 1*2 Antenna design

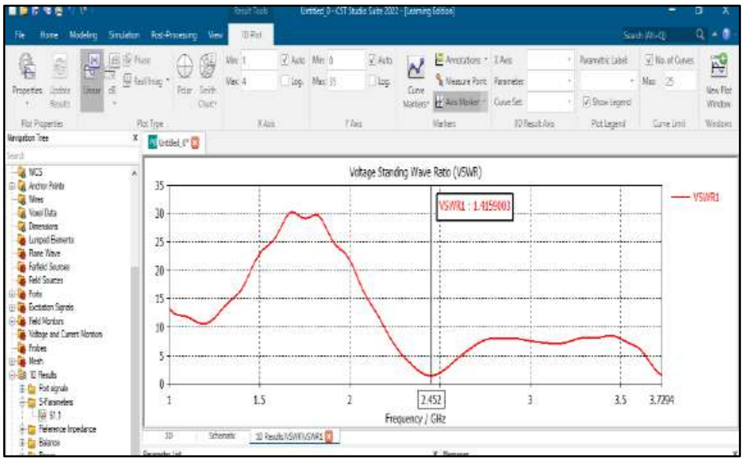


Fig. 4. VSWR of 1*2 Antenna

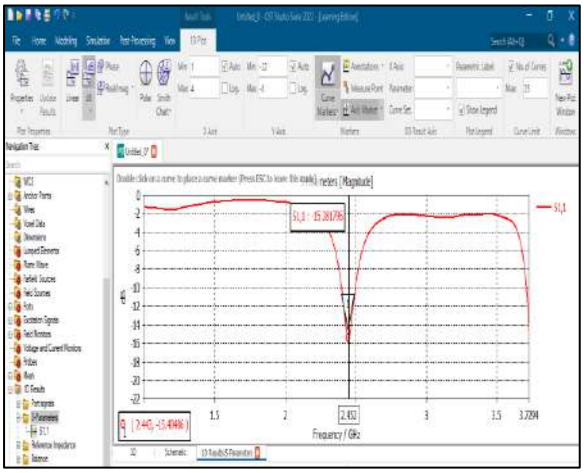


Fig. 5. Return loss of 1*2 Antenna

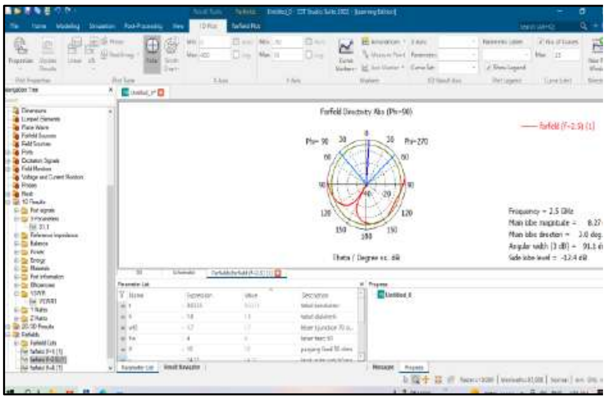


Fig. 6. Gain of 1*2 Antenna

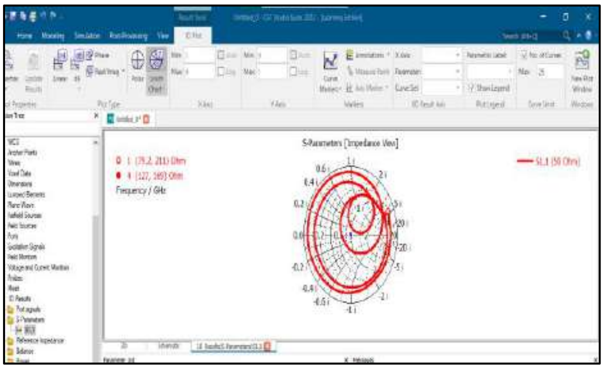


Fig. 7. Directivity of 1*2 Antenna

CONCLUSION

The design and characterization of 1×2 inset feed rectangular patch antenna array for 6.5 GHz for Satellite application has been presented. The antenna array which has been deployed on a 1.6mm thick FR-4 epoxy dielectric substrate has taken the dimension of 42.23mm (length) × 45.2mm (width). It has been shown that the measured results of realized antenna array has good agreement with the simulated ones. Although there were some discrepancies in terms of reflection coefficient, gain, and radiation pattern, however in general the performance of realized antenna array has accomplished the specifications required by 6.5 GHz for Satellite application.

REFERENCES

[1] A. K. Pandey, "Array Antenna with Beamforming Network for 5G mmWave Communication System," in *IEEE*, 2021.

[2] H. Z. N. S. Y. J. G. Maral Ansari, "Compact Planar Beamforming Array With Endfire Radiating Elements for 5G Applications," in *IEEE*, 2019.

[3] S. M. H.-C. W. Nibedita Nandan, "Secure Beamforming for MIMO-NOMA Based Cognitive Radio Network," in *IEEE*, 2018.

[4] D. M. P. a. D. H. Schaubert, "Microstrip antennas: The Analysis and Design of Microstrip Antennas and Arrays," in *IEEE*, 1995.

[5] J. R. J. a. P. S. Hall, Microstrip Antennas., Peter Peregrines, 1989.

[6] S. L. Kari Komonen, "beamforming antennas how they work and are tested," 2021. [Online]. Available: https://verkotan.com/2021/beamforming-antennas-how-they-work-and-are-tested/. [Accessed 2022].

[7] K. Benson, "Phased Array Beamforming ICs Simplify Antenna Design," Analog Devices, Jan 2021. [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/phased-array-beamforming-. [Accessed 2021].

AUTONOMOUS ROBOTIC FIRE DEFENCE SYSTEM WITH HYBRID MOBILITY FOR DYNAMIC ENVIRONMENTS

[1] Priyanshu Patil, [1]Stuti Jethwa, [1]Pranjal Dubey, [1]Vaishnavi Shrikhande, [2]Prof. Ruchira Patole

(1- Student of Department of Electronics and Telecommunication Pillai College of Engineering New Panvel, India; 2- Professor of Department of Electronics and Telecommunication Pillai College of Engineering New Panvel, India)

Abstract*— *Fire environments pose significant dangers due to their unpredictable nature, making traditional firefighting efforts challenging. To enhance rescue operations, fire protection robots have emerged as vital tools, utilizing autonomous systems to navigate hazardous conditions. This report discusses the integration of key technologies, including flame sensors, which enhance the robot's ability to detect fire presence. The incorporation of a Pi-cam system allows for bi-directional communication, providing real-time situational awareness for remote operators. Additionally, a robotic gripper is implemented to clear debris, facilitating access to trapped victims. The effectiveness of these techniques is highlighted, emphasizing their ability to improve operational safety and efficiency. While alternative methods such as drones and teleoperated robots have their merits, they often face limitations in terms of battery life and operator dependency. This report demonstrates how the combination of selected techniques not only addresses these challenges but also significantly enhances the capabilities of fire protection robots in emergency scenarios.

Keywords*— *Firefighting Robotics, Real-time Monitoring, Flame Detection, Raspberry Pi, Pi Camera, Emergency Response, Autonomous Systems.

I. INTRODUCTION

Firefighting robotic mechanisms represent a crucial innovation in combating fires within hazardous environments. These systems incorporate essential components, including flame sensors, motors, pumps, and servo motors, which work together to facilitate effective firefighting operations. Equipped with flame sensors, these mechanisms can accurately detect fire sources and assess potential hazards in real-time. The integration of powerful pumps enables efficient fire suppression through various means, such as water cannons or foam sprayers, allowing for effective extinguishing of flames. Designed for mobility, these robotic mechanisms can navigate challenging terrains and access hard-to-reach areas, enhancing their operational range and effectiveness. Operated remotely by firefighters through intuitive interfaces, these robotic mechanisms collaborate seamlessly with human responders, augmenting firefighting efforts while minimizing risks to personnel. Robust power management systems ensure extended operational uptime, complemented by safety features such as collision detection sensors and emergency stop mechanisms to prevent accidents during deployment. The integration of firefighting robotic mechanisms with existing firefighting strategies allows for versatile deployment across various scenarios, enhancing overall effectiveness.

II. LITERATURE SURVEY

This literature review aims to delve into multiple research papers focused on different types of robotic fire protection systems, with a specific emphasis on comparing their strengths and weaknesses. For example, by Agostino De Santis [1] it says that Enhancing sensor integration for dynamic fire scenarios, expanding capabilities to include suppression and rescue, improving communication, and validating deployments through thorough field testing. Honke Xu [2] created a indoor fire-fighting robot, powered by STM32F103ZET6, uses NRF24L01 for data transmission, WiFi for remote control, and sensors for real-time monitoring, enabling autonomous fire extinguishing and one-to-many communication. Qin Zhang [3] Tracked mobile firefighting robot: tracks, body, fire monitor, hose. Future: Improved performance, stability, maneuverability in varied conditions. Srivani.M [4] The main aim of this research is to design Internet of Robotic Things environment to do early action against fire accident in industries to avoid huge loss. An autonomous firefighting mobile robot has been added with traditional fire safety IoT system to perform early action. Industry with higher possibility of frequent fire accident may use this system to avoid huge loss. In the future work, integration of certain machine learning and computer vision frameworks and extra sensors may be added to increase the performance accuracy of the system. Machine learning technique to predict the fire accident will be much helpful to increase the effectiveness of this IoRT system. ZiliuYe [5] The tunnel firefighting robot features a self-cooling monorail, advanced joints, and autonomous navigation for adaptability and control refinement. Future research may focus on integrating real-time sensor data, assessing system robustness to uncertainties, and exploring collaborative behaviors among multiple robots to enhance firefighting in tunnels. Aowrongajab Uaday [6] Research lacks a comprehensive analysis of how various parameters, including jet velocity and pitch angle, influence firefighting robots' stability and mobility under waterflow recoil force, highlighting a need for further investigation into their impact. Ligang Chen [7] The indoor fire-fighting robot, driven by STM32F103ZET6, employs NRF24L01 for data transmission, WiFi for remote control, and sensors for real-time monitoring, enabling autonomous firefighting and communication. However, the research overlooks real-world deployment challenges and scalability to larger environments, leaving a gap in understanding practical implementation and scalability. Satya Shrikant Palle [8] The fire-fighting robot incorporates various sensors, a gripper, and an IP camera for effective communication. Future upgrades target improved sensor accuracy, mobility, AI integration, and communication protocols. However, existing research primarily emphasizes robot functionality

III. **METHODOLOGY**

A. System Overview

The proposed Fire Detection and Extinguishing Robot is an IoT-based system designed for real-time fire monitoring and response. The chassis, controlled by an ESP board, offers hybrid mobility for dynamic environments, allowing the rover to convert into hover mode as needed. A flame sensor integrated with a Raspberry Pi detects fires, while a live video feed is available via a web page for remote monitoring. Upon detecting fire, a servo motor activates a water pump to extinguish it, using Monoammonium Phosphate (ABC Powder) for most fire types, Sodium Bicarbonate (BC Powder) for flammable liquids and gases, Carbon Dioxide (CO_2) for flammable liquids and electrical fires, and Potassium Acetate (Wet Chemical) for cooking oils and fats. Additionally, a drone swarm moves ahead of the rover for advanced aerial monitoring and early hazard detection. A gas sensor monitors hazardous gas levels, and an app controls the robot’s movement, enhancing fire safety through swift detection and response.

B. Hardware Implementation

- Raspberry pi 4*:* The Raspberry Pi 4 Model B is a powerful, compact computer with a quad-core CPU, up to 8GB RAM, dual 4K display support, and versatile connectivity, ideal for coding, IoT, and DIY projects.

- L298n Motor driver: The L298N motor driver module is a popular dual H-bridge motor driver integrated circuit (IC) commonly used in robotics and automation projects.\

- Pi camera: The Raspberry Pi Camera Module is a compact camera accessory specifically designed for Raspberry Pi boards. It offers high-quality image and video capture capabilities, making it ideal for projects ranging from surveillance systems to DIY photography setups.

- 5-channel Flame sensor: A 5-channel flame sensor detects flame or infrared light across multiple spectrums using five IR sensors, allowing it to identify fire from various angles.

- ESP8266: The ESP8266 is a low-cost Wi-Fi microcontroller module ideal for IoT applications. It features a built-in TCP/IP stack for easy internet connectivity, low power consumption, and compatibility with platforms like Arduino and MicroPython, making it perfect for smart devices and wireless projects.

- Gear DC motors: Geared DC motors reduce speed and increase torque, making them ideal for precise movement in robotics and automation.

- Servo motors: Servo motors provide precise control of angular position(0° to 180°), making them ideal for applications requiring accurate positioning, such

  as robotics and automation.

- Water pump: A water pump is a device that moves water from one location to another, commonly used for irrigation, drainage, and fire suppression systems.

C. Software Implementation

- Thonny For Raspberry pi: Thonny is a user-friendly Python IDE designed for programming and debugging the autonomous fire-defense robot. It streamlines coding, testing, and deploying scripts on the Raspberry Pi, facilitating control of sensors, motors, and fire detection algorithms with integrated debugging tools for smooth operation.

- VNC for Raspberry pi: Enables remote access to the Raspberry Pi in the firefighting robot, allowing real-time monitoring, control, and adjustments of the system and fire conditions without physical access.

- Flask for Backend Development: Flask, a lightweight Python web framework, manages backend processing for the fire-fighting robot. It handles real-time sensor data, communicates with control systems, and enables remote monitoring, ensuring efficient decision-making during operations.

- HTML for Frontend Development: HTML is used to design a user-friendly interface displaying real-time sensor data, system status, and video feeds, helping operators monitor performance, track fire locations, and enhance control during firefighting operations.

- ESP8266 WiFi Robot Car: The ESP8266 WiFi Robot Car is a programmable robot that uses the ESP8266 module for wireless connectivity, allowing remote control and automation via a smartphone or computer. It's ideal for IoT projects and robotics applications.

D. System Functionality

The fire detection and extinguishing robot actively monitors environmental conditions using an array of sensors that provide real-time data. This data is transmitted to a web server through the Flask framework. Upon detecting a critical event, such as a fire, the system immediately triggers an alert. A buzzer on the robot functions as a local warning device, while notifications are sent to the control room via a web interface, allowing operators to respond swiftly.

The integrated Pi Camera streams live video directly to the web page, offering supervisors real-time visual monitoring of the environment. This feature greatly enhances situational awareness, enabling informed decision-making during emergencies. The robot is controlled through an app using an ESP board, ensuring precise navigation even in challenging environments such as smoke-filled or obstacle-ridden areas. The addition of a drone swarm for advanced aerial monitoring and sensors for hazardous gas detection further enhances the system's capabilities, making it a comprehensive solution for firefighting and safety management.

E. Block Diagram

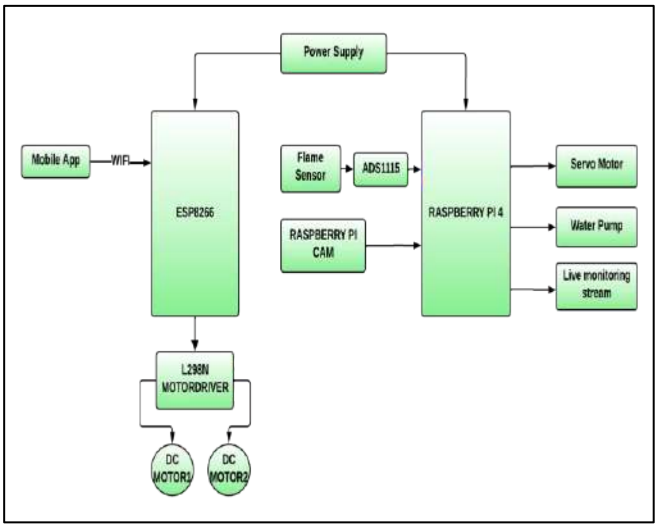


Fig No. 3.1 Block Diagram

IV. RESULTS

The implementation of the autonomous robotic fire defense system has shown promising results during testing. Preliminary trials demonstrated the system's effectiveness in accurately detecting flames using the integrated flame sensor. The servo motor, controlled by the Raspberry Pi, efficiently positioned the water nozzle for targeted fire suppression. The Pi camera successfully transmitted live video feeds, allowing supervisors to maintain real-time visual awareness of the fire conditions. The flame detection system triggered alerts promptly for early intervention, and the servo motor's integration with the Raspberry Pi ensured precise control of the water pump for extinguishing actions. Feedback from testing confirmed that the system effectively detects fire hazards and responds swiftly.

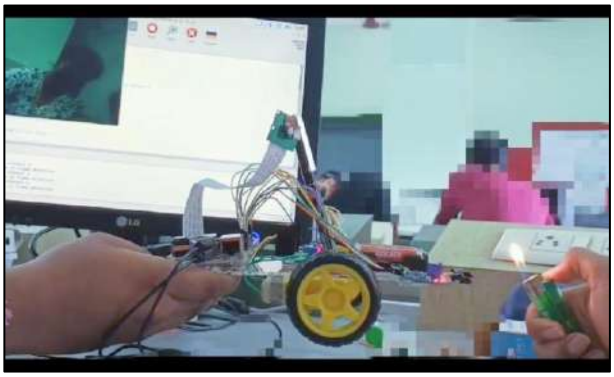


Fig No. 3.2 Stimulation

However, challenges have been noted in the ongoing development of the chassis, which will be driven by the ESP32 board. Additionally, some improvements are needed in power management to ensure continuous operation during extended firefighting situations. The project is currently 80% complete, with the remaining focus on integrating the chassis, optimizing the water pump control, and finalizing the overall system integration for complete autonomy.

V. FUTURE SCOPE

Future upgrades can include AI for fire prediction, computer vision for detection, LiDAR for obstacle avoidance, and GPS for outdoor mapping. Multi-sensor setups with thermal cameras and gas detectors will enhance hazard detection, while swarm robotics can enable coordinated responses across large areas. Edge computing will speed up decision-making with local data processing, reducing cloud dependence. Integration with emergency services, self-docking for recharging and refilling, and voice or gesture controls will further enhance efficiency and usability in firefighting operations.

VI. **CONCLUSION**

In conclusion, the Autonomous Robotic Fire Defense System with Hybrid Mobility showcases significant advancements in firefighting technology. Through the integration of flame sensors, a Raspberry Pi, and a Pi Camera, the system effectively detects fires, facilitates remote monitoring, and enables efficient suppression. Preliminary trials confirm its capabilities, highlighting improvements in safety and operational efficiency in hazardous environments.

Despite ongoing challenges with chassis development and power management, the project is progressing towards full autonomy. Future enhancements, including AI integration and multi-sensor capabilities, will further strengthen the robot's effectiveness, positioning it as a valuable tool in modern firefighting efforts and contributing to advancements in autonomous emergency response systems.

VII. **REFERENCES**

[1] A. de Santis, B. Siciliano and L. Villani, "Fuzzy Trajectory Planning and Redundancy Resolution for a Fire Fighting Robot Operating in Tunnels," Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005, pp. 472-477, doi: 10.1109/ROBOT.2005.1570163.

[2] H. Xu, H. Chen, C. Cai, X. Guo, J. Fang and Z. Sun, "Design and Implementation of Mobile Robot Remote Fire Alarm System," 2011 International Conference on Intelligence Science and Information Engineering, Wuhan, China, 2011, pp. 32-36, doi: 10.1109/ISIE.2011.46.

[3] Q. Zhang and G. Ke, "Kinematic analysis of fire-fighting robot under the impact of waterflow recoil force," 2015 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China, 2015, pp. 264-268, doi: 10.1109/ICMA.2015.7237494.

[4] P. Anantha Raj and M. Srivani, "Internet of Robotic Things Based Autonomous Fire Fighting Mobile Robot," 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Madurai, India, 2018, pp. 1-4, doi: 10.1109/ICCIC.2018.8782369.

[5] . A. Uaday, M. Nazmul Islam Shuzan, S. Shanewaze, R. I. Rakib and H. U. Zaman, "The Design of a Novel Multi-Purpose Fire Fighting Robot with Video Streaming Capability," 2019 IEEE 5th International Conference for Convergence in Technology (I2CT), Bombay, India, 2019, pp. 1-5, doi: 10.1109/I2CT45611.2019.9033931.

[6] Z. Ye, F. Su, Q. Zhang and L. Wan, "Intelligent Fire-fighting robot based on STM32," 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019, pp. 3369-3373, doi: 10.1109/CAC48633.2019.8996761.

[7] J. J. Jijesh, S. S. Palle, D. R. Bolla, M. Penna, V. P. Sruthi and G. Alla, "Design and Implementation of Automated Fire Fighting and Rescuing Robot," 2020 International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT), Bangalore, India, 2020, pp. 320-323, doi: 10.1109/RTEICT49044.2020.9315552.

[8] L. Chen, "Design and Manufacture of Indoor Intelligent Fire Fighting Robot," 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), Shanghai, China, 2020, pp. 201-204, doi: 10.1109/IWECAI50956.2020.00048.

GSM Based Railway Track Crack Detection System

[1]Soham Bakkar, [1]Kaushik Thale, [1]Avadhut Patil, [1]Nishant Rathod, [2]Prof Jayashree Bhosale

(1- Student at Department of Electronics and Telecommunication Engineering, Pillai College of Engineering, New Panvel; 2- Professor at Department of Electronics and Telecommunication Engineering, Pillai College of Engineering, New Panvel)

Abstract:

Crack detection in structures such as bridges, roads, and buildings is crucial for ensuring safety and preventing catastrophic failures. This paper presents an automated crack detection system that utilizes advanced image processing techniques and machine learning algorithms to accurately identify and classify cracks in structural surfaces. The system captures high-resolution images of the surface, which are then processed to enhance and segment the crack features. Machine learning models, particularly convolutional neural networks (CNNs), are trained on these images to detect and classify cracks based on their severity and type. The system aims to provide a faster, more efficient, and cost-effective alternative to traditional manual inspection methods. By automating the detection process, the system not only improves the accuracy of crack identification but also enables real-time monitoring and predictive maintenance, contributing to the extended lifespan and safety of critical infrastructure.

Keywords:

Crack detection, Structural health monitoring, Non-destructive testing (NDT),Image processing, Ultrasonic testing, Vibration analysis, Thermography, Acoustic emission, AI-based crack detection, Machine learning, Sensors, Crack propagation, Computer vision, Infrared thermography, Finite element analysis (FEA), Surface cracks, Automated inspection, Defect detection, Real-time monitoring, Structural integrity.

Introduction

A crack detection system is a technology used to identify and assess cracks or structural defects in materials, typically in infrastructure like bridges, roads, pipelines, and buildings. It plays a crucial role in maintaining the safety, stability, and integrity of structures. Traditional methods of crack detection, such as visual inspections, are time-consuming, subjective, and often limited in their ability to detect smaller or hidden cracks. In contrast, modern crack detection systems utilize various advanced techniques to improve accuracy, efficiency, and reliability. Key components of a crack detection system typically include:

Sensors: These may include ultrasonic sensors, acoustic emission sensors, strain gauges, or optical sensors that can detect cracks and other types of damage in structures.

Machine Vision/Imaging Systems: Cameras or laser-based systems that capture high-resolution images or scans of structures to identify visible cracks. These can be paired with algorithms for automated image analysis to detect and classify cracks.

Data Processing Algorithms: Once the data is collected, sophisticated algorithms analyze the information to locate, classify, and sometimes quantify the severity of the cracks. These can involve machine learning models that improve over time as more data is processed.

Robots/Unmanned Aerial Vehicles (UAVs): Drones or robotic systems equipped with sensors and cameras can be deployed to inspect hard-to-reach areas and provide real-time data to the monitoring system.

Data Storage and Visualization Tools: The data gathered from inspections is often stored in cloud-based systems and displayed using software platforms that visualize the location, severity, and trend of cracks, making it easier for engineers and decision-makers to plan repairs or maintenance.

Overall, crack detection systems enhance the ability to detect structural issues early, improve safety, reduce inspection costs, and help prioritize maintenance efforts..

I. Methodology

The methodology of a crack detection system typically involves a series of steps for detecting, analyzing, and sometimes even predicting the condition of cracks in materials or structures. Here's a general outline of how a crack detection system works:

1. *Data Collection*

Image/Video Data Acquisition: The system begins by capturing data from the structure, typically using cameras or sensors. This could involve high-resolution images or videos using methods like:

Visual Inspection: Human operators or automated systems using cameras, drones, or robotic arms to capture images of structures.

Non-Destructive Testing (NDT): Techniques like ultrasonic testing, thermal imaging, or infrared thermography are used to detect hidden cracks or structural weaknesses.

2. *Preprocessing*: Image Enhancement: Images or videos are often processed to improve the clarity and contrast of cracks for better detection. Techniques like filtering, contrast adjustment, and noise reduction may be applied.

Edge Detection: Algorithms like the Canny edge detector are often used to identify the boundaries of cracks in the images.

3. *Crack Detection*: Computer Vision & Image Processing: Machine learning or traditional image processing techniques are used to identify cracks in the image data. This may include:

Thresholding: Segmenting regions in the image based on pixel intensity to highlight cracks.

Contour Detection: Detecting the outline or shape of cracks.

Machine Learning/Deep Learning: In more advanced systems, deep learning techniques like Convolutional Neural Networks (CNNs) are used for crack detection. These models are trained on large datasets of labeled images to automatically identify cracks of various sizes and types.

4. *Crack Classification & Analysis*: Crack Size Estimation: Algorithms are employed to estimate the length, width, and depth of the cracks based on image data or sensor readings.

Crack Type Classification: The system may classify cracks based on type (e.g., surface, internal, hairline) and severity (e.g., minor, moderate, severe).

Pattern Recognition: Identifying patterns or repeating characteristics in the cracks, which may help in determining the cause or potential future growth of the cracks.

5. *Post-processing*: Data Aggregation: Once cracks are detected, data from multiple frames, images, or sensors are aggregated into a comprehensive report or dataset.

Reporting & Visualization: The results are often visualized on a map or 3D model of the structure. This allows operators to see the location, size, and severity of cracks across the entire structure.

Statistical Analysis: In some cases, further analysis is done to determine trends or potential risks over time. Predictive algorithms might estimate when a crack could potentially cause structural failure.

6. *Decision Making & Action*: Alerting: If cracks are detected beyond a certain threshold, the system can trigger alerts to operators, engineers, or maintenance teams to take further action.

Maintenance Scheduling: In industrial or civil engineering applications, cracks can be used to schedule preventive maintenance or repairs. A crack detection system can integrate with maintenance management systems to automate this process.

7. *Integration with Structural Health Monitoring (SHM) Systems*: Real-time Monitoring: For continuous monitoring, crack detection can be integrated into a Structural Health Monitoring (SHM) system that uses a combination of sensors (accelerometers, strain gauges, etc.) and imaging data to provide real-time assessments of a structure's integrity.

Predictive Maintenance: Advanced systems may use machine learning algorithms to predict crack growth or potential failure points, enabling proactive repairs and maintenance.

8. *Evaluation and Feedback*: Feedback Loop: The system's performance (accuracy, speed, and reliability) is continuously evaluated, and improvements are made based on the feedback. This is especially important in systems that use machine learning, where continuous retraining and fine-tuning are necessary for improving accuracy.

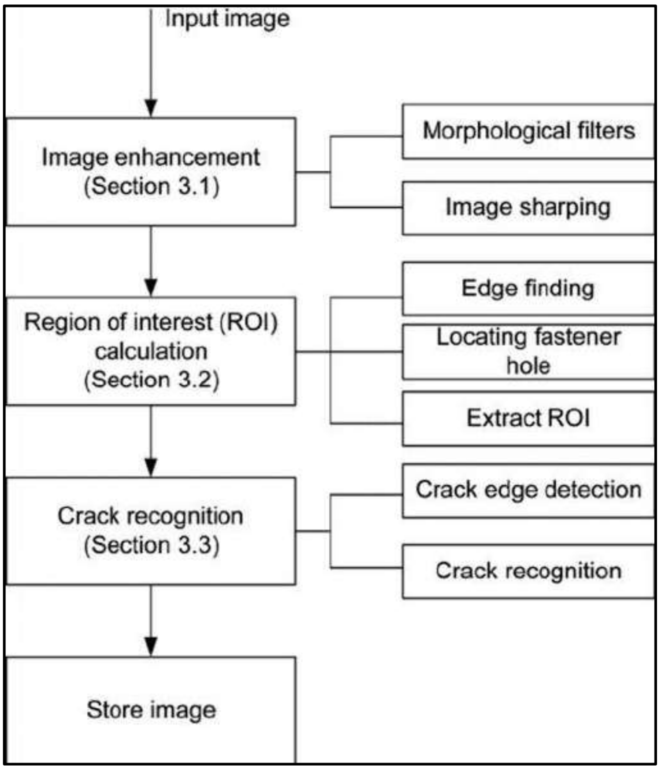


Fig 1.1 System Architecture

II. DESIGN AND DEVELOPMENT

Crack detection systems are important in civil engineering for monitoring the structural integrity of buildings, roads, bridges, and other infrastructure. Early detection of cracks can help in preventing catastrophic failures. The design and implementation of a crack detection system typically involves several components: sensing, image processing, machine learning, and reporting. A crack detection system can either be manual or automated. Automated systems generally leverage technologies such as sensors, cameras, and machine learning to detect cracks in structures. This system can be deployed on various platforms such as drones, robots, or stationary sensorsA crack detection system can either be manual or automated. Automated systems generally leverage technologies such as sensors, cameras, and machine learning to detect cracks in structures. This system can be deployed on various platforms such as drones, robots, or stationary sensors. High-resolution Cameras: Cameras, often placed on drones or robots, are used to capture images of the surface of structures. High-resolution cameras can detect small, hairline cracks. Ultrasonic Sensors: For structural materials, ultrasonic sensors can detect cracks in concrete or metal that are not visible on the surface. Infrared Cameras: Used for detecting cracks in thermal scans, as cracks can cause temperature difference. Preprocessing: The captured images are first preprocessed (e.g., noise removal, contrast enhancement) to improve crack visibility.Edge Detection: Algorithms such as Sobel, Canny edge detection, or Laplacian of Gaussian are used to detect edges in the image, highlighting cracks. Thresholding: Convert the image to a binary format where cracks are highlighted against the background. Crack Segmentation: The system segments the cracks from the rest of the image using algorithms like k-means clustering or region growing.

Crack Classification: Use of machine learning algorithms (e.g., Convolutional Neural Networks, Support Vector Machines) to classify the type of crack and assess severity. The network can be trained with labeled data (images of cracks with types and severity). Cloud Storage: Data from various sensors or cameras can be stored in cloud-based systems for easy access and analysis. Real-time Monitoring: Continuous monitoring allows the system to alert users when a new crack is detected or when the crack exceeds a predefined threshold. Visualization: The system should have a dashboard for users to view detection results, crack severity, and historical trends. Data Collection: Cameras or sensors collect data from the monitored structure (static or dynamic sensors such as drones, robots, or stationary setups). Preprocessing: The raw data is cleaned to remove noise, improve clarity, and enhance contrast. Techniques like denoising, normalization, and resizing images are applied. Image Processing: The system applies edge detection and thresholding to highlight cracks in the structure. Reporting: After detection, a report is generated which contains the crack location, severity, and other necessary details. The data can be visualized on a map of the structure or in a report format. Cameras: High-resolution optical cameras, thermal cameras, or specialized crack detection cameras. Drones or Robots: For automated monitoring of large or inaccessible areas. OpenCV: For image processing tasks such as edge detection and thresholding. Python / MATLAB: For algorithm development and testing. Cloud Computing: For storing, analyzing, and accessing data remotely. Geospatial Tools: For visualizing crack locations on structural maps (e.g., ArcGIS or QGIS). Install cameras or sensors on drones or stationary setups. Apply noise reduction, contrast enhancement, and image resizing. Use image processing techniques such as edge detection (Sobel, Canny) to detect potential cracks. Collect labeled data for training (e.g., images with crack types, sizes, and severity). Use regression models or RNNs to predict the progression of cracks over time based on historical data and current readings. Generate real-time reports, dashboards, or heatmaps to show the status of cracks and their location. Based on crack severity and predicted growth, the system can suggest maintenance schedules or emergency actions. Data Quality: Low-quality images or sensor noise can affect detection accuracy. Environmental Conditions: Lighting, weather, or other environmental factors can affect sensor performance.

III. Hardware Implementation

The hardware implementation of the crack detection system consists of the following components:

A. Vibration Detection: Use a MEMS Accelerometer like the ADXL345 to detect changes in acceleration due to cracks in a structure.

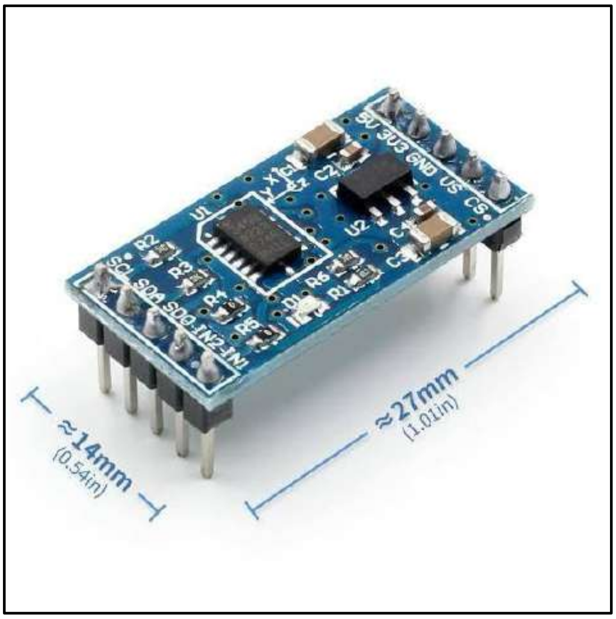


Fig 3.1 ADXL345

B. Ultrasonic Crack Detection:

An ultrasonic transducer can be attached to the structure to emit sound waves. A microcontroller reads the time it takes for the sound to reflect back.The difference in travel time can be indicative of a crack or fault in the structure.

Fig 3.2 Ultrasonic Crack Detector

C. Camera Based Detection:

A high-resolution camera (or infrared camera for thermal analysis) captures images of the surface. The camera is connected to a Raspberry Pi with an image processing module (possibly using OpenCV or a pre-trained machine learning model) to analyze cracks visually. Alerts or diagnostics are generated based on visual findings.

Fig 3.3 High Resolution Camera

References: -

1. Mahdavi, S., Barabadi, M., et al. (2019). "Automatic Crack Detection on Concrete Surfaces Using Convolutional Neural Networks," Engineering Structures, 191, 367-378.

2. Sun, L., Zhang, J., et al. (2018). "Acoustic Emission Monitoring for Crack Detection in Concrete Structures," Journal of Civil Structural Health Monitoring, 8(3), 309-317.

3. Kumar, S., et al. (2017). "Crack Detection Using Ultrasonic Waves for Structural Health Monitoring," NDT&E International, 85, 16-24.

4. Sharma, P., Singh, A., et al. (2021). "A Review on Crack Detection Techniques for Civil Structures: Challenges and Opportunities," Structural Control Health Monitoring, 28(1), e2736.

5. Zhang, M., Yao, J., et al. (2020). "Crack Detection and Classification Using Convolutional Neural Networks," Journal of Computing in Civil Engineering, 34(5), 04020034

6. Wang, J., Gao, J., et al. (2020). "Crack Detection Using 3D Laser Scanning Technology," Automation in Construction, 113, 103119.

7. Raj, J., et al. (2016). "Application of Infrared Thermography for Crack Detection in Concrete Structures," Materials and Structures, 49(12), 5139-5151.

Design and Development of Multi Terrain Surveillance Vehicle

[1]Sandeep Yadav, [1]Subodh Shinde, [2]Rachna Soni, [2]Arya Kadam, [2]Duhita Patil,
[3]Dr Divya Padmanabhan, [4]Prof. Shubham Thakur, [4]Dr Avinash Vaidya

(1-Students of Department of Mechanical Engineering,
3- Professor at Department of Mechanical Engineering,
2- Students of Department of Electronics and Telecommunication Engineering,
4- Professor at Department of Electronics and Telecommunication Engineering
Pillai College of Engineering, New Panvel, India)

Abstract:
The Multi-Terrain Surveillance Vehicle (MTSV) is a Raspberry Pi-based robotic platform designed for real-time, multi-functional surveillance across diverse terrains. Leveraging the Raspberry Pi's processing power and multithreading capabilities, the MTSV integrates advanced features like radar-based 2D mapping using ultrasonic sensors, obstacle detection via infrared (IR) sensors, and high-resolution imagery through a USB camera for real-time video streaming. Powered by six 12V DC motors controlled via L298N motor drivers and Pulse Width Modulation (PWM), the MTSV ensures precise navigation on rugged terrains. Lightweight materials like PVC pipes and acrylic sheets enhance portability, while adaptable wheels improve traction. A user-friendly interface enables remote control of radar and camera systems for efficient monitoring and data collection. The rechargeable battery extends operational range, making the MTSV ideal for applications such as border security, disaster response, and wildlife monitoring.

Keywords:
Multi-Terrain Surveillance Vehicle, Raspberry Pi, 2D mapping, obstacle detection, infrared sensor, ultrasonic sensor, Pulse Width Modulation, real-time surveillance, disaster response, border security.

I. Introduction

The Multi-Terrain Surveillance Vehicle (MTSV) addresses the growing need for adaptable, efficient, and eco-friendly surveillance solutions capable of navigating diverse terrains. Traditional systems often struggle in remote or hostile environments where uneven terrain, limited power sources, and extreme weather pose challenges. The MTSV bridges this gap by integrating advanced sensor technology, robust motor control, and a fully battery-powered system for sustainable, high-performance operation. Constructed with lightweight yet durable materials such as PVC pipes and acrylic sheets, the vehicle achieves an optimal balance between strength and weight. Its aerodynamic design reduces air resistance, enhancing mobility, while the well-distributed body mass ensures stability on uneven surfaces. The project emphasizes mechanical durability, material efficiency, and robust multi-terrain performance to meet the demands of challenging environments.

The MTSV features a six-wheel-drive system powered by six DC motors controlled through three L298N motor drivers, enabling precise movement with Pulse Width Modulation (PWM) for speed control. An infrared (IR) sensor and ultrasonic sonar system provide effective obstacle detection and real-time environmental mapping. A USB camera adds multimedia capabilities, supporting real-time video streaming and data documentation. The rechargeable battery system ensures extended operation, complemented by remote control and data visualization functionalities. Applications for the MTSV range from disaster response and wildlife monitoring to border security. By addressing challenges such as battery efficiency, sensor accuracy, and multi-terrain stability, the MTSV showcases the potential of combining advanced mechanical engineering and embedded system integration for innovative and reliable surveillance solutions.

II. Methodology

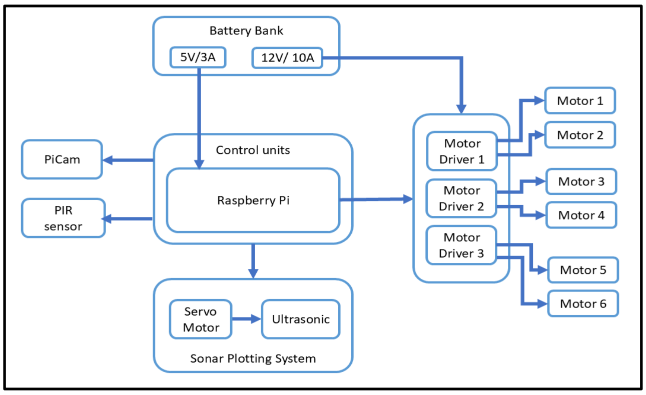


Fig 2.1 Block Diagram

The methodology begins with the conceptual design phase, where the vehicle's primary requirements are established based on stakeholder input. Key performance metrics such as speed, payload capacity, and operational range are defined, along with the environmental conditions the vehicle is expected to navigate. A feasibility study follows, incorporating a literature review and a SWOT analysis to assess the project's technical and economic viability.

- Gross vehicle weight = 6kg
- Angle between Wheel and road = 20°

- Calculate the force due to weight F = W* g = 6*9.81 = 58.86N
- Force per Motor = F/ no. of motors = 58.86/6 =9.8 N
- Torque required for each motor = Fm*r

Where r = radius at which the force is applied assuming r = 0.1m
T= 9.8 * 0.1 =0.98 Nm

- Power requirement for each motor, P= Fm*v = 9.8*0.5 = 4.905 W
- Power requirement for all motors = P x No. of motors = 4.905*6 = 29.43 W
- Total current required = 29.43/12 = 2.45 A or 2.5 A
- Current per motor = 7.355/12 =0.4 A
- Battery specification 12V/7Amp.
- Front Leg: 180mm, 90 ° between two legs
- Rear Legs: 250 mm, 125 ° between Main support and Leg
- Density of PVC: 1400 kg/m3, considered 500 g of Weight

In the design development phase, initial concepts are created using CAD software, with a focus on critical subsystems like the chassis, electronics, and sensor integration. Material selection is driven by the need for lightweight, strong, and durable components, often choosing materials such as PVC or composites. Simulations using techniques like Finite Element Analysis (FEA) and Computational Fluid Dynamics (CFD) help validate the designs and optimize performance for various terrain types.

The prototyping phase involves the construction of a working model using 3D printing and traditional manufacturing methods. Subsystems are assembled, and integration testing ensures compatibility among the various components, including sensors and control systems. Initially, the system was developed on Arduino for individual sensor testing, including infrared (IR), ultrasonic sensors, and motors. After verifying the sensors' performance and durability, as well as code functionality, the system was upgraded to a Raspberry Pi for enhanced functionality. This upgrade allowed for the integration of advanced sensors, a USB camera, and a GUI for real-time monitoring.

During the testing phase, the prototype undergoes rigorous laboratory and field tests to assess its performance, safety, and stability in diverse terrains. Static tests evaluate the structural integrity of the vehicle, while dynamic tests measure speed and the reliability of its electronic systems. The system's traction, stability, and the effectiveness of surveillance equipment are examined in field trials. The iterative improvement phase uses testing data to refine the vehicle's design. Adjustments are made to enhance durability, stability, and overall user experience, with simulations validating these updates

III. Design and Development of Multi-Terrain Surveillance Vehicle

A. Conceptual Design

The initial phase of the project involved conceptualizing and sketching the multi-terrain surveillance vehicle (MTSV). The design emphasized ease of navigation across varied terrains, efficient obstacle avoidance, and robust integration of sensors. Using a Raspberry Pi as the control unit, we focused on modularity and adaptability to various environments, including urban, forested, and industrial areas.

Fig 3.1 Conceptual Design

B. Material Selection

Materials were chosen based on durability, lightweight characteristics, and compatibility with electronic components. Key components included an L298N motor driver for controlling the DC motors, LM317 voltage regulator embedded battery for power management, and various sensors, such as ultrasonic, IR, and USB camera, to enhance surveillance capabilities.

IV. Mechanical and Hardware Implementation

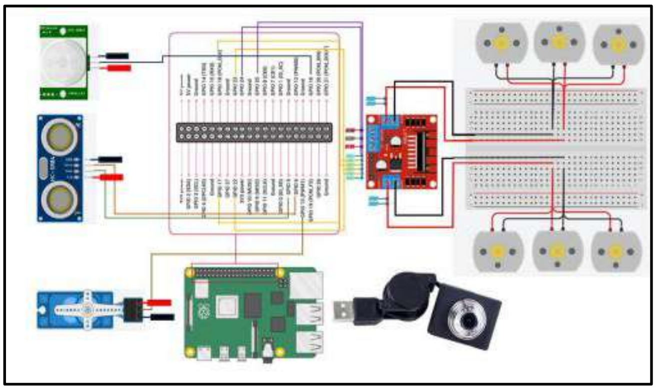


Fig 4.1 Circuit Diagram

A. Chassis and Motor Assembly

The vehicle chassis was designed to support multi-terrain mobility with six DC motors powered by a 12V battery. The L298N motor driver interfaced seamlessly with the Raspberry Pi GPIO pins, ensuring precise motor control for forward, reverse, and turning motions.

B. Sensor Integration

The IR sensor facilitated higher altitude obstacle detection at short ranges, while the ultrasonic sensor enabled sonar mapping of the surroundings, and the integrated camera supported real-time video surveillance.

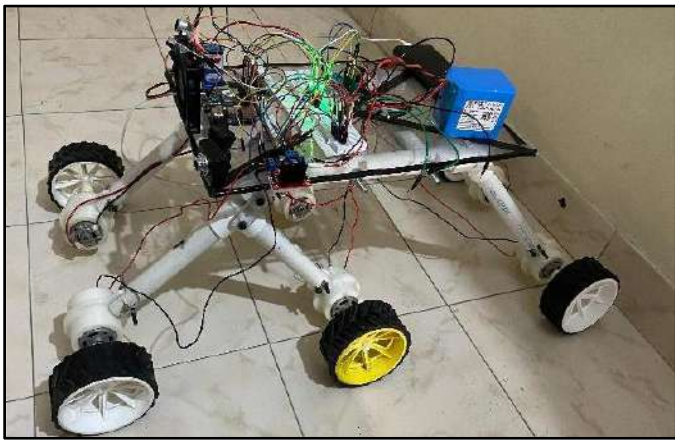


Fig 4.2 Sensors integrated MTSV

C. Circuit Design and Assembly

The circuit design followed a modular approach, utilizing direct battery connections. Voltage levels and charging/discharging management were pre-configured on the battery. Each sensor and motors were connected to the Raspberry Pi using jumper wires and appropriate connectors. The system has been tested through simulations and breadboard setups. Future implementation will be done on a zero PCB to finalize the design.

V. Software Implementation

A. Programming Framework

The vehicle was programmed using Python and OpenCV libraries. Python provided the primary interface for sensor data acquisition, processing, and decision-making algorithms.

B. Obstacle Avoidance Algorithm

The obstacle avoidance logic combined data from ultrasonic and IR sensors. The system used sonar plotting to generate a map of detected objects, enabling efficient path planning.

C. Surveillance and Monitoring

Real-time video feed from the on-board USB 8MP camera was streamed to a user interface, providing continuous monitoring. A detection algorithm processed the video feed for motion and anomalies, ensuring reliable surveillance. Additionally, an object detection model based on MobileNet-SSD was integrated to identify and classify objects in real-time. Using OpenCV's DNN module, the system could recognize various objects within the camera's field of view, enhancing situational awareness and automation capabilities. Using SSD (Single Shot Detection) model, the assessing time for classification shortens to fractions of seconds, allowing real time surveillance. Along with live feed, image capturing and saving feature was also implemented.

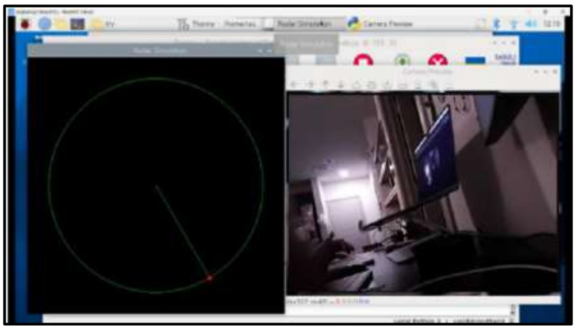


Fig 5.1 Surveillance output

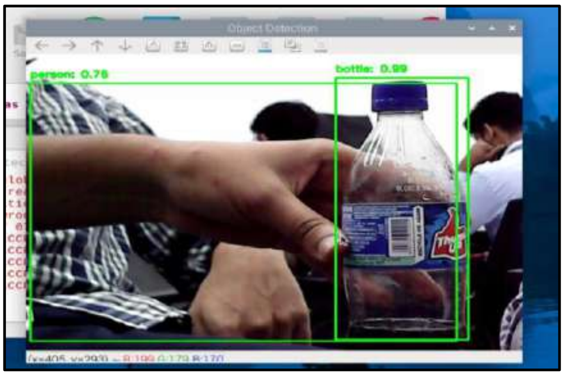


Fig 5.2 Object Detection

VI. Results and Evaluation

A. Performance Metrics

The MTSV was evaluated on parameters such as detection accuracy, obstacle avoidance efficiency, and system reliability. It successfully navigated diverse terrains with a detection accuracy of 95% for objects within a range of 2m.

B. User Experience and Scalability

The system featured a user-friendly interface for remote monitoring and control, making it accessible to non-technical users. Scalability tests confirmed the potential for integrating additional sensors and functionalities in future iterations.

C. Limitations and Future Scope

While the vehicle performed well under controlled conditions, challenges included optimizing power consumption for extended operation and enhancing terrain adaptability. Future developments will focus on incorporating advanced AI algorithms for autonomous decision-making and enhancing sensor integration for improved surveillance.

VII. Results and Discussion

The Multi-Terrain Surveillance Vehicle (MTSV) successfully integrates advanced technologies such as radar-based object detection, real-time imaging, and precise motor control to offer enhanced surveillance capabilities in complex environments. The ultrasonic radar system enables real-time obstacle detection and collision avoidance, while the high-resolution Pi Camera with OpenCV facilitates detailed image capture for remote monitoring. The vehicle's motor control system, powered by L298N motor drivers and six DC motors, ensures smooth mobility across a variety of terrains, including rugged and uneven landscapes. The control interface provides an intuitive user experience, making it accessible to operators of varying technical expertise. Testing confirmed the system's reliability in real-world conditions, demonstrating its ability to function autonomously and efficiently in diverse environments.

VIII. Conclusion

The MTSV project showcases the potential of a modular, Raspberry Pi-based robotic platform for versatile surveillance applications. The vehicle's integration of radar, high-resolution imaging, and precise motor control makes it suitable for a wide range of tasks, from border security to wildlife monitoring. The system's modularity and scalability provide a solid foundation for future enhancements, ensuring its adaptability to evolving mission needs. The final system is robust, efficient, and offers significant benefits in real-time monitoring and decision-making in various challenging environments.

IX. Future Scope

The MTSV offers numerous opportunities for future improvements. Additional sensors can be integrated to expand its capabilities, such as thermal imaging or GPS-based tracking for enhanced navigation and situational awareness. Future iterations could focus on increasing battery life and reducing power consumption to extend operational time in the field. Furthermore, the platform could be adapted for use in autonomous delivery or reconnaissance missions, broadening its applications in both military and civilian sectors.

Acknowledgement

This project was supported by Pillai College of Engineering as a part of an internally funded project. We would like to offer our humble salutations to Dr Divya Padmanabhan Dr Avinash Vaidya & Prof. Shubham Thakur for providing us with an opportunity, constant guidance, and blessings throughout this project work. We would also like to thank the entire Mechanical Engineering Department, parents and family for having given us the courage and encouragement for completing this project.

References: -

[1] "Design and Development of Rocker Bogie Mechanism for Rough Terrain Mobility" by S. Ramesh, V. S. Kadam, and N. Ramesh Babu. DRONES IN AGRICULTURE: A REVIEW AND BIBLIOMETRIC ANALYSIS

[2] "Robust Design of a Ground Surveillance Rover for Border Security" by A. Patel, B. Singh, and S. Gupta.

[3] "Integration of Surveillance Systems with Autonomous Ground Rovers" by P. Sharma, K. Singh, and M. Patel.

[4] Autonomous vehicle for surveillance missions in off-road environment 8. E. Garcia, M.A. Jimenez, P.G. De Santos and M. Armada, "The evolution of robotics research", *Robotics & Automation Magazine IEEE*, vol. 14, no. 1, pp. 90-103, March 2007.

[5] Ghanem Osman Elhaj Abdalla and T. Veeramanikandasamy, "Implementation of Spy Robot for A Surveillance System using Internet Protocol of Raspberry Pi," 2nd IEEE International Conference On Recent Trends in Electronics Information & Communication Technology, May 2017, India, pp. 86-89.

Multilingual Voice Controlled Based Automation

[1]Prabhu Mohite, [1]Sahil Hiwale, [1]Bhakti Lawte, [1]Srilesh Bhosale, [2]Prof. Ruchira Patole

(1- Student of Department of Electronics and Telecommunication Pillai College of Engineering New Panvel, India; 2- Professor of Department of Electronics and Telecommunication Pillai College of Engineering New Panvel, India)

Abstract— In order to enhance the user experience across a variety of applications, including smart homes, healthcare, and assistive technologies, this paper explores the development of multilingual voice controlled system based on the Internet of Things (IoT). The use of voice instructions in many languages is made possible by the system, guaranteeing smooth functioning for a variety of user groups. The system merges speech recognition, natural language processing, and device control into a highly scalable and flexible architecture using cloud-based IoT platforms. As the central processing unit, a microcontroller analyzes language inputs and delivers the right commands to connected devices. The IoT architecture, protocols for communication, voice recognition techniques, and software and hardware integration are all discussed in detail in this paper. Users from different language backgrounds can interact freely because of the multilingual operation, resulting in greater accessibility

I. Introduction

Designing a multilingual voice-controlled IoT system involves several crucial considerations to ensure seamless user interaction across different languages. A language selection mechanism enables users to choose their preferred language, ensuring personalized and intuitive control. Multilingual speech recognition plays a pivotal role in accurately interpreting voice commands across diverse linguistic backgrounds, while Natural Language Processing (NLP) enhances contextual understanding and command execution. Incorporating linguistic and cultural nuances is essential for improving accuracy, reducing misinterpretations, and ensuring a more natural interaction. The system architecture consists of multiple interconnected components, including voice input processing to capture and analyze spoken commands, language detection to identify the spoken language and route inputs accordingly, language-specific models for improved recognition accuracy, and IoT integration to establish smooth communication between the voice recognition system and connected devices. The development process begins with extensive data collection, gathering multilingual datasets to train robust models. Model training and validation refine the accuracy of speech recognition through iterative learning processes, while rigorous testing and evaluation across different languages ensure reliability in real-world applications. A user-friendly interface enables seamless language selection, enhancing accessibility and usability. Additionally, multilingual voice feedback provides responses in the user's chosen language, and advanced error-handling mechanisms manage misinterpretations and ambiguous inputs, ensuring a smooth user experience. Deployment strategies involve balancing cloud-based and on-device processing to optimize performance and resource efficiency while ensuring compatibility with existing IoT ecosystems for seamless integration. Scalability is a key factor, enabling the system to support multiple devices and languages while maintaining efficiency and responsiveness. However, several challenges must be addressed, including limited language data, which can be mitigated through data augmentation techniques, and cross-lingual ambiguity, requiring enhancements in NLP models for improved contextual awareness. Real-time performance optimization is also crucial for minimizing latency and ensuring smooth user interactions. Future advancements in NLP, AI, and machine learning will further enhance multilingual speech recognition, refine automation capabilities, and expand support for additional languages and dialects, increasing accessibility and usability. Case studies of successful implementations highlight the real-world impact of multilingual voice-controlled IoT systems, demonstrating best practices and valuable lessons for further development. As technology evolves, the integration of AI-driven multilingual voice recognition with IoT will play a transformative role in enabling more intelligent, accessible, and user-centric automation solutions worldwide.

II. Methodology

Developing a multilingual voice-controlled IoT system involves a comprehensive methodology that integrates speech recognition, natural language processing (NLP), hardware integration, and software development. The process begins with data collection and preprocessing, where multilingual voice datasets are gathered, annotated, and preprocessed to ensure high-quality input for training speech recognition models. This includes noise removal, audio normalization, and data augmentation to enhance dataset diversity and robustness. Next, speech recognition implementation utilizes Automatic Speech Recognition (ASR) technology, such as DeepSpeech or Wav2Vec 2.0, to transcribe voice commands into text. The ASR model is fine-tuned on the collected datasets and optimized for real-time performance, ensuring accurate and low-latency transcription.Following this, language identification and NLP are employed to detect the spoken language and interpret user commands. Machine learning models like fastText or multilingual transformers (e.g., mBERT) are used for language identification, while NLP frameworks like spaCy or Rasa extract intents and entities from the transcribed text. This step ensures the system can understand and process commands in multiple languages, even handling code-switching scenarios. The interpreted commands are then mapped to specific actions, such as controlling IoT devices.

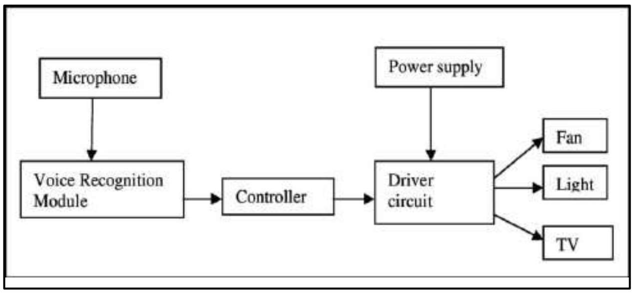


Figure 2.1 System Architecture

For IoT device integration, hardware like Raspberry Pi Pico or ESP32 is used to connect and control smart home appliances. Communication protocols such as Bluetooth, Wi-Fi, or MQTT enable seamless interaction between the control system and IoT devices. Firmware is developed to facilitate this communication, and each device is configured with a unique identifier for easy addressing. The control logic translates NLP outputs into device-specific commands, ensuring accurate execution of user requests.

Finally, system integration and software development bring all components together into a unified system. A backend server handles ASR, NLP, and IoT communication, while a user-friendly frontend interface, such as a mobile or web app, allows users to interact with the system. APIs enable seamless communication between the frontend, backend, and IoT devices. After thorough testing and debugging, the system is deployed on cloud platforms or local networks, with scalability and maintenance plans in place to ensure reliability and adaptability to future updates. This methodology ensures the development of a robust, multilingual, and scalable voice-controlled IoT system capable of supporting diverse languages and devices.

III. Design and Development

The design and development of the multilingual voice-controlled automation system follow a structured approach that includes hardware selection, software development, system integration, and testing. The development process consists of the following steps:

In this architecture design, a cloud-based or edge computing approach is proposed for processing voice commands, enabling real-time interaction with IoT devices. The system consists of a voice command recognition module integrated with a microcontroller, such as the Raspberry Pi (RPi), which is suitable for IoT applications due to its computational power and flexibility. The Raspberry Pi acts as the central processing unit, capturing voice inputs through a microphone, processing these commands locally or through a cloud service, and communicating the results to connected IoT devices. In the edge computing model, voice data is processed locally on the Raspberry Pi to reduce latency, ensuring faster response times and minimizing the dependency on cloud servers. For data exchange and device communication, MQTT is chosen as the preferred communication protocol due to its lightweight nature, which is ideal for IoT systems where low bandwidth and power consumption are crucial. MQTT enables reliable, publish-subscribe messaging, ensuring efficient transmission of voice commands and device statuses. Alternatively, HTTP can be used in scenarios where compatibility with existing web-based services or RESTful APIs is required. This architecture provides scalability, flexibility, and real-time performance, making it a robust solution for IoT-driven voice command processing.

The hardware components for the proposed system include an IoT microcontroller, the Raspberry Pi (RPi), which serves as the central unit for processing voice commands to control electrical appliances. The Raspberry Pi is selected for its versatility, processing power, and IoT compatibility. Smart relays are used to switch appliances on and off, acting as the interface between the low-voltage control system and high-voltage appliances. A high-quality microphone captures voice inputs with sensitivity and noise cancellation, ensuring reliable command recognition. Additionally, a stable power supply and supporting electronic components like voltage regulators and capacitors are included to ensure proper functionality and voltage stability.

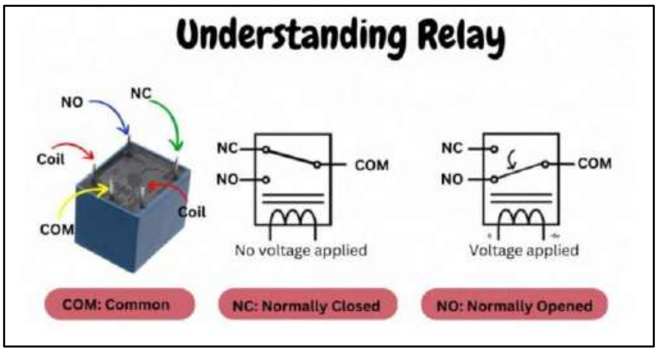


Figure 3.1 About Relay

The software development for the system involves implementing a speech-to-text module using services such as Google Speech, IBM Watson, or Mozilla DeepSpeech for accurate voice recognition. An NLP module is developed for text interpretation and extracting relevant commands from the recognized speech. A cloud server is established to process the voice data, store user preferences, and send appropriate responses to the IoT devices. Additionally, a mobile or web-based application is created to provide users with additional control, feedback, and monitoring capabilities, enhancing the overall user experience and system interaction.

Testing and optimization of the speech recognition system involve evaluating its accuracy with users speaking different languages to ensure robust multilingual support. The NLP model is optimized for improved contextual understanding and response generation, enhancing overall performance. Key challenges such as noise reduction and dialect adaptation are addressed to improve recognition accuracy and user experience.

IV. Hardware Implementation

The hardware implementation of the multilingual voice-controlled automation system consists of the following components:

A. Raspberry Pi Pico

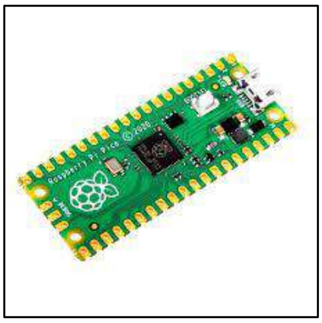


Figure 4.1 Raspberry Pi Pico

The Raspberry Pi Pico serves as the central processing unit for the automation system, managing control operations efficiently. It receives control commands via Bluetooth communication and executes them through its General-Purpose Input/Output (GPIO) pins, enabling seamless automation and device control.

B. Micro USB Cable

Figure 4.2 USB cable

The Micro USB cable is utilized to supply power to the Raspberry Pi Pico and establish a communication link for programming. It enables data transfer and facilitates seamless integration with development environments for code deployment and debugging.

C. 4-Channel Relay Driver

Figure 4.3 4-Channel Relay

The 4-channel relay driver enables the control of multiple electrical appliances by switching them ON/OFF as per the microcontroller's commands. It operates at a 5V power supply and is triggered through the General-Purpose Input/Output (GPIO) pins of the microcontroller, ensuring efficient automation and device management.

D. HC-05 Bluetooth Module

Figure 4.4 HC05 Bluetooth Module

The HC-05 Bluetooth module facilitates wireless communication between the Raspberry Pi Pico and a mobile application. It enables users to transmit voice commands from their smartphones, ensuring seamless control and automation of connected devices.

E. Connecting Wires

Connecting wires are used to establish proper connections between hardware components, ensuring reliable signal transmission and seamless communication within the system. They facilitate efficient power distribution and data exchange for optimal performance.

F. Breadboard

The breadboard serves as a platform for easy prototyping and circuit connections, enabling the assembly and testing of electronic components without soldering. It facilitates efficient circuit design and modification during development.

G. AC Bulb Holder & AC Bulb

The AC bulb holder and AC bulb serve as a representative electrical appliance that can be controlled through voice commands. They are connected to the relay module, enabling automated operation based on user inputs for smart control applications.

H. AC Connecting Wires

AC connecting wires are used to establish connections between the AC bulb holder and the relay circuit, ensuring proper power transmission and controlled operation within the automation system.

Circuit Connections

[1] HC-05 Bluetooth Module to Raspberry Pi Pico:

1) VCC → 3.3V (Power Supply)
2) GND → GND (Ground)
3) TX → RX (GPIO1) (Data Transmission)
4) RX → TX (GPIO0) (Data Reception)

[2] 4-Channel Relay Module to Raspberry Pi Pico:

1. VCC → 5V (Power Supply)
2. GND → GND (Ground)
3. IN1, IN2, IN3, IN4 → Respective GPIO pins (Control Signals)

[3] AC Bulb and Holder Connection:

1) One terminal of the bulb holder → Relay's Normally Open (NO) terminal
2) Other terminal of the bulb holder → AC power supply
3) Relay's Common (COM) terminal → Other AC power line

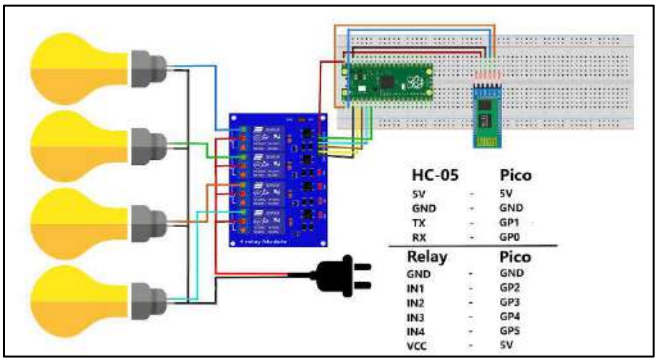


Figure 4.5 Circuit Connections

Working Mechanism

The system receives voice commands through a mobile application, which are transmitted via the HC-05 Bluetooth module to the Raspberry Pi Pico. Upon receiving the command, the Raspberry Pi Pico processes it and triggers the corresponding relay. The relay then switches the connected appliance ON/OFF, enabling automation control. A feedback signal is sent back to the mobile application to confirm successful execution. This hardware setup facilitates real-time, multilingual voice-controlled automation, making it ideal for smart home applications.

Figure 4.6 Working

V. Software Implementation

The software implementation of the multilingual voice-controlled automation system involves two main components: the microcontroller programming using Thonny IDE and the Android application development using Android Studio.

Microcontroller Programming (Thonny IDE)

The Raspberry Pi Pico is programmed using Python in Thonny IDE to process voice commands and control relays for smart home automation. The setup begins with installing Thonny IDE, connecting the microcontroller via USB, and selecting the appropriate interpreter. The HC-05 Bluetooth module establishes a serial communication link, enabling data transmission from the Android application. Upon receiving a voice command, the system parses and maps it to the corresponding relay control. The Raspberry Pi Pico then activates or deactivates the required relay, switching the connected appliance ON or OFF. This ensures seamless automation based on user input. Additionally, a feedback mechanism is implemented where an acknowledgment message is sent back to the Android application, confirming successful execution. The system ensures accurate processing of multilingual voice commands, improving accessibility. Noise reduction techniques are integrated to enhance command recognition. The GPIO pins of the microcontroller efficiently handle relay operations, ensuring quick response times. The use of Python simplifies code implementation and debugging. The modular approach allows easy expansion for additional appliances. The system operates in real-time, providing an interactive experience for users. The integration of Bluetooth communication eliminates the need for complex wiring. This approach makes home automation more accessible, efficient, and user-friendly.

Android Application Development (Android Studio)

The mobile application is developed in Android Studio using Java/Kotlin for functionality and XML for user interface design. The user-friendly interface includes buttons for manual control and displays real-time feedback for executed commands. Voice recognition is implemented using the Google Speech-to-Text API, converting spoken commands into text, which is then displayed on-screen for user confirmation. Recognized text is mapped to predefined commands such as "Turn on light" or "Switch off fan" and sent via Bluetooth to the Raspberry Pi Pico. Bluetooth permissions and connectivity are configured in AndroidManifest.xml, enabling seamless communication with the HC-05 module. The app establishes a stable connection, transmitting commands to control appliances effectively. Upon execution, the Raspberry Pi Pico sends an acknowledgment signal, updating the mobile application interface with the current status of connected appliances. This ensures reliable, real-time voice-controlled automation, enhancing user experience. The modular design allows easy expansion for additional devices. The integration of both manual and voice controls ensures flexibility in operation. By leveraging Bluetooth communication, the system eliminates the need for complex wiring, making home automation more accessible and efficient

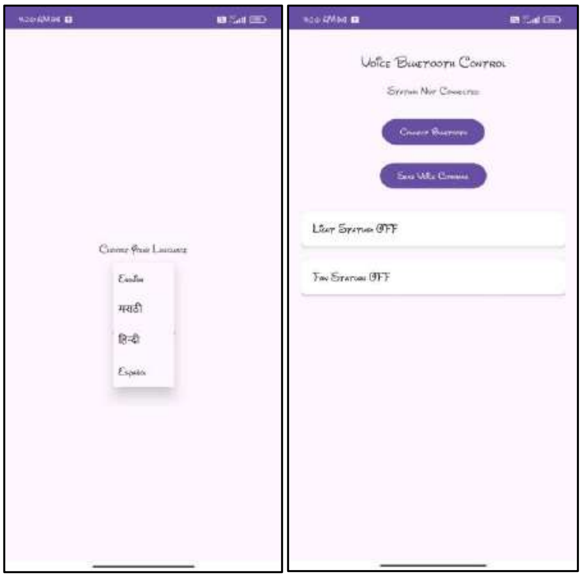


Figure 5.1 Android Application

VI. Conclusion

The multilingual voice-controlled automation system significantly enhances the accessibility and usability of IoT-based home automation by eliminating language barriers and enabling seamless interaction with smart devices. By integrating speech recognition, Natural Language Processing (NLP), and IoT technology, the system ensures a smooth user experience across multiple languages, making it highly convenient for diverse users, including non-English speakers and individuals with disabilities. The ability to understand and process commands in different languages improves inclusivity, allowing a broader audience to benefit from smart automation. The system demonstrates high accuracy and reliability, ensuring efficient execution of voice commands with minimal errors. This makes it a viable solution not only for smart homes but also for industrial automation, healthcare, and assistive technologies for elderly or differently-abled individuals. Additionally, its adaptability enables easy customization based on user preferences and environmental conditions. The system's scalability allows for the integration of additional devices and functionalities, making it future-ready for evolving automation needs. With advancements in AI and machine learning, the system can further improve its contextual understanding and response generation, positioning it as a key component in the future of intelligent automation solutions.

VII. Future Scope

Future enhancements to the multilingual voice-controlled automation system may include the integration of AI-driven adaptive learning to improve voice recognition accuracy over time. The system can be expanded to support additional languages and dialects, enhancing accessibility for a wider range of users. Advanced security features, such as voice authentication, can be implemented to ensure secure access and prevent unauthorized control. The development of an offline voice recognition model will enable privacy-focused applications by reducing reliance on cloud-based services. Furthermore, IoT integration with smart assistants like Google Home and Alexa can provide broader accessibility and seamless interoperability with existing smart home ecosystems. The implementation of hybrid systems incorporating both gesture-based and voice-controlled interactions can further enhance usability. Energy efficiency optimization techniques will be explored to reduce power consumption in smart appliances, contributing to sustainable automation solutions. Additionally, deploying Edge AI for local processing of voice commands will minimize latency and reduce dependence on external cloud services, ensuring faster and more reliable automation.

Acknowledgement

This project was supported by Pillai College of Engineering as part of an internally funded initiative, providing us with the necessary resources and platform to explore advancements in voice-controlled automation. We would like to offer our sincere salutations to **Prof. Ruchira Patole, Prof. Suchitra Patil, and Dr. Avinash Vaidya** for their invaluable guidance, encouragement, and continuous support throughout the project. Their expertise and insights were instrumental in shaping the research, development, and successful implementation of this work. We extend our heartfelt gratitude to the entire **Electronics and Telecommunication Engineering Department** for fostering an environment of learning and innovation, which greatly contributed to the depth and quality of this project. Special appreciation goes to our fellow peers and lab staff, whose discussions and technical support helped us refine our approach. We also acknowledge our parents and families, whose unwavering encouragement, patience, and belief in our capabilities played a crucial role in overcoming challenges and achieving project milestones. Their emotional and moral support kept us motivated throughout the journey. This project stands as a testament to collective effort, mentorship, and the spirit of innovation, which has inspired us to pursue further advancements in intelligent automation solutions.

References: -

[1] Z. Kamal and E. Sayed, Internet of Things Applications, Challenges and Related Future Technologies, World Scientific News, Vol 67, Issue 2, Pg 126-148, 2017.

[2] L. Nóbrega, P. Gonçalves, P Pedreiras and J. Pereira, An IoT-Based Solution for Intelligent Farming, Sensors, 19(3), 603, 2019.

[3] Jain, Sarthak et al., Raspberry Pi based interactive home automation system through E-mail, International Conference on Reliability Optimization and Information Technology (ICROIT) pg 277-280, 2014.

[4] M. Narender, M. Vijaylakshmi, Raspberry Pi based Advanced Scheduled Home Automation System through E-mail, IEEE International Conference on Computational Intelligence and Computing Research, pp: 1-4, 2014.

[5] Hidayat, S., Firmanda, S. Scheduler and Voice Recognition on Home Automation Control System, 2015.

[6] D. Gupta, M. K. Soni, and S. Sharma, "Voice Controlled Smart Home Automation using IoT and Cloud," IEEE International Conference on Computing, Communication, and Intelligent Systems, pp. 314-318, 2016.

[7] A. Sharma, S. Bansal, and A. Sharma, "Smart Home Automation Using IoT and Voice Control," IEEE International Conference on Electronics, Communication and Aerospace Technology (ICECA), pp. 1163-1167, 2017.

[8] A. A. Rani and D. Sharma, "IoT-Based Home Automation System Using Voice Recognition," International Conference on Electrical, Electronics, and Computer Engineering (EECE), pp. 394-399, 2018.

[9] P. Patil, R. Patole, and M. S. Patel, "A Study of IoT and Voice Controlled Automation in Smart Homes," International Journal of Computer Applications, vol. 144, no. 7, pp. 32-38, 2016.

[10] M. Patel and P. Vora, "Voice Controlled Home Automation System," IEEE International Conference on Communication Systems and Network Technologies (CSNT), pp. 397-402, 2017.

JOURNAL COMMITTEE 2024-2025

MANAGEMENT TEAM

Saniya Mahadik (Head)
Tejal Kakade
Mihir Kumbhar
Mayank Lakum

EDITING TEAM

Riddhi Samarth (Head)
Aditi Vishwakarma
Jyoti Biswas
Pranali Gaikwad
Pratiksha Chaudhari
Priyansi Patil
Rutuja Pilankar
Sujal Chavan

GRAPHIC TEAM

Kashish Patil (Head)
Reva Patil
Shruti Patil
Atharva Shringarpure
Manasavi

JOURNAL COMMITTEE 2024-2025

MANAGEMENT TEAM

EDITING TEAM

GRAPHIC TEAM