



Implementation of an Intelligent Robotic Surveillance System with Real-Time Video Streaming

M. Venisha¹, CH. V. K. Vardhan², A. Jyoshna³, E. Sai⁴, J. Likhith Sagar⁵, Dr. S. Sridhar⁶

^{1,2,3,4,5}B.Tech Students, Department of Electronics and Communication Engineering,

⁶ Professor, Department of Electronics and Communication Engineering, Nadimpalli Satyanarayana Raju Institute of Technology



<https://doi.org/10.55041/ijst.v2i4.395>

Cite this Article: Venisha, M., Vardhan, C. V. K., Jyoshna, A., Sai, E. & Sagar, J. L. (2026). Implementation of an Intelligent Robotic Surveillance System with Real-Time Video Streaming. *International Journal of Science, Strategic Management and Technology*, 02(04). <https://doi.org/10.55041/ijst.v2i4.395>

License: This article is published under the Creative Commons Attribution 4.0 International License (CC BY 4.0), permitting use, distribution, and reproduction in any medium, provided the original author(s) and source are properly credited.

Abstract—In many real-world situations, especially in areas where safety and quick awareness are important, relying only on human monitoring can be difficult and sometimes risky. To overcome this, this work presents an intelligent robotic surveillance system that allows users to monitor environments remotely while reducing direct human involvement. The system combines Raspberry Pi and Arduino to handle processing and control efficiently, along with a 5 MP camera that provides live video through a web-based interface.

The robot is capable of real-time video streaming, remote navigation, and basic motion-based monitoring, making it useful in applications such as security surveillance, industrial inspection, and disaster-prone areas. During testing, the system maintained stable video streaming at around 30 FPS with low latency between

0.2 and 0.5 seconds. It also responded quickly to user commands and demonstrated reliable control during operation.

Overall, the system is simple, cost-effective, and practical for real-world deployment. In the future, it can be further improved by adding intelligent features such as object detection, facial recognition, and autonomous navigation.

Keywords - Intelligent robotic surveillance, real-time video streaming, Raspberry Pi, Arduino, embedded systems, remote monitoring, mobile robot, IoT-based surveillance, web-based control, motion detection, smart security systems

I. INTRODUCTION

The growing demand for safety, security, and continuous situational awareness has driven the development of intelligent surveillance systems. Traditional fixed CCTV systems, although widely used, are limited by restricted coverage, lack of mobility, and poor adaptability in dynamic environments. As a result, monitoring areas where human presence is unsafe or impractical remains a challenge.

To address these limitations, the integration of robotics, embedded systems, and networking technologies offers a more flexible and efficient approach. Mobile robotic platforms enable real-time monitoring larger areas, allowing users to remotely observe and respond to situations. Such systems are especially useful in applications such as security surveillance, industrial inspection, and disaster management, where timely information and reduced human risk are critical.

In this work, an intelligent robotic surveillance system is developed to provide real-time video streaming and remote navigation. The system utilizes a Raspberry Pi for processing,

communication, and video handling, along with an Arduino microcontroller for precise control of motors and hardware components. A web-based interface allows users to access live video and control the robot from any location without requiring dedicated applications.

The proposed system is cost-effective, scalable, and suitable for real-world deployment. By enabling continuous monitoring while reducing direct human involvement, it supports safer and more efficient surveillance practices and provides a foundation for future intelligent and autonomous system enhancements.

The following sections present the proposed methodology, system architecture, and working principle, followed by performance evaluation and concluding remarks.

II. LITERATURE SURVEY

Over time, the rapid expansion of the Internet of Things (IoT) has reshaped the way surveillance systems are used. As more devices become interconnected, remote monitoring and data exchange have become more efficient and accessible. Research in this area highlights how IoT middleware and communication frameworks enable smooth interaction between devices, improving coordination in distributed environments [1], [2].

Alongside this, mobile robotic surveillance systems have gained attention as a more adaptable alternative to conventional fixed CCTV setups. Unlike stationary cameras, these robots can move across different locations, offering better coverage in changing environments [3], [4]. The use of platforms such as Raspberry Pi and Arduino has made these systems more practical by balancing performance with affordability, while also allowing efficient distribution of processing and control tasks [5], [6].

A key aspect of such systems is the ability to stream video in real time. Lightweight communication protocols, including MQTT, are commonly used to ensure quick and reliable transmission of video data. This helps maintain continuous monitoring with minimal delay, which is essential in surveillance applications [7], [8].

With the growth of computer vision and deep learning, surveillance systems have become more capable and intelli-

gent. Instead of only observing, they can now identify events and objects. Models such as YOLO are widely used due to their ability to perform fast and accurate detection in real-time scenarios [9], [10].

To further enhance system performance, approaches like cloud robotics and edge computing have been introduced. These methods distribute computational tasks between local devices and cloud infrastructure, reducing latency and improving scalability in real-time operations [11], [12], [17], [18].

However, increasing connectivity also brings challenges, particularly in terms of security and reliability. Protecting sensitive data and ensuring secure communication remain critical concerns. Various studies focus on encryption techniques and robust communication methods to address these issues [13], [14], [20]. In addition, efforts are being made to reduce delays in communication to improve overall responsiveness [15].

Frameworks such as ROS2 have simplified robotic system development by providing modular and scalable architectures [16]. Even with these advancements, issues such as dependence on network availability, energy consumption, and performance under varying conditions continue to be important areas for further research.

In summary, existing research shows considerable progress in intelligent surveillance systems. However, there is still a need for solutions that combine efficiency, scalability, and simplicity while remaining cost-effective and suitable for real-world deployment.

III. PROPOSED METHODOLOGY

A. System Design

The system is designed with the idea of making surveillance more flexible and accessible. Instead of relying on fixed cameras, a mobile robot is used to move through the environment while streaming live video. The Raspberry Pi handles video processing and communication, while the Arduino takes care of controlling the robot's movement. This separation keeps the system simple and efficient.

B. Component Selection

The components were chosen by balancing performance and cost. The Raspberry Pi was selected for its ability to handle video streaming and networking, while the Arduino was used for controlling motors and hardware connections. A camera module is used to capture live video, and a motor driver helps in controlling the movement of the robot.

C. Video Streaming Mechanism

The camera continuously captures the surroundings, and the Raspberry Pi processes this video and sends it over the network. The live stream is displayed on a web interface, allowing users to view the environment in real time from anywhere.

D. Control Mechanism

The system allows users to control the robot through a web interface. When a command is given, it is received by the Raspberry Pi, processed, and then sent to the Arduino. The Arduino then drives the motors accordingly, enabling smooth and responsive movement of the robot.

E. System Integration

All components are connected to work as a single system. The Raspberry Pi manages communication and video streaming, while the Arduino controls the robot's movement. Together, they ensure that the system runs smoothly without delays.

F. Performance Evaluation

The system was tested under different conditions to observe its performance. It was able to provide stable video streaming with low delay and respond quickly to user commands. Overall, the system performed reliably, making it suitable for real-time surveillance applications.

IV. SYSTEM ARCHITECTURE

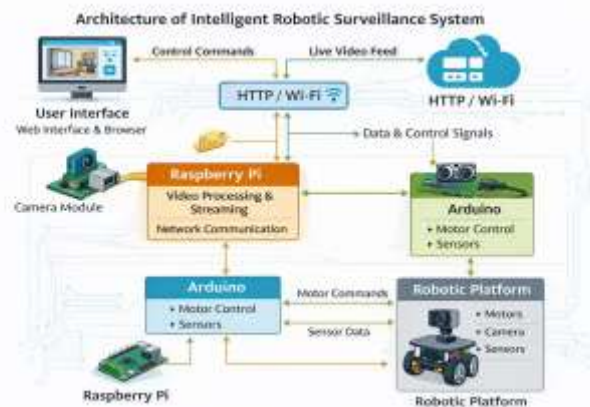


Fig. 1. Architecture of Intelligent Robotic Surveillance System

The system architecture of the intelligent robotic surveillance system is designed to enable real-time video streaming, remote control, and efficient data communication. It brings together key components such as the user interface, communication layer, processing unit, control unit, and the robotic platform. All these parts work together to ensure smooth interaction between hardware and software, allowing reliable and continuous surveillance.

A. User Interface

The user interface serves as the main point of interaction for the user. It is developed using web technologies and can be accessed through a browser on devices like laptops or smartphones. Through this interface, users can view the live video feed and control the movement of the robot, making the system easy to use without the need for any additional applications.

B. Communication Layer

The communication layer connects the user interface with the robotic system. It uses Wi-Fi along with HTTP protocols to transmit commands and video data. This setup ensures stable and real-time communication, allowing the user to operate the system remotely from different locations.

C. Processing Unit (Raspberry Pi)

The Raspberry Pi acts as the central unit of the system. It captures video from the camera, processes it, and streams it to the user interface. At the same time, it receives commands from the user and converts them into signals that can be understood by the Arduino. It also handles overall communication and keeps the data flow smooth across the system.

D. Control Unit (Arduino)

The Arduino is responsible for executing the commands at the hardware level. It receives instructions from the Raspberry Pi and controls the motor driver accordingly. This allows the robot to move in different directions with accuracy and respond effectively to user inputs.

E. Robotic Platform

The robotic platform includes DC motors, a camera module, sensors, and a power supply unit. The motors enable movement, while the camera captures live video of the surroundings. The platform is designed to operate efficiently in different environments, making it suitable for surveillance applications.

F. Data Flow and Operation

The system works through continuous interaction between its components. When the user gives a command, it is processed by the Raspberry Pi and passed to the Arduino for controlling the robot's movement. At the same time, the camera captures live video and streams it back to the user, providing real-time monitoring and feedback.

V. WORKING PRINCIPLE

The working of the system is based on continuous interaction between the user and the robot. It allows real-time video streaming and remote navigation by converting user inputs into actions while providing live visual feedback.

A. Command Input and Transmission

The process starts when the user gives commands through a web interface. These commands are sent over Wi-Fi to the Raspberry Pi, enabling remote control.

B. Video Capture and Streaming

The camera captures live video, which is processed by the Raspberry Pi and streamed to the user interface for real-time monitoring.

C. Command Processing and Execution

The Raspberry Pi processes user commands and sends control signals to the Arduino. The Arduino then controls the motor driver to move the robot in the desired direction.

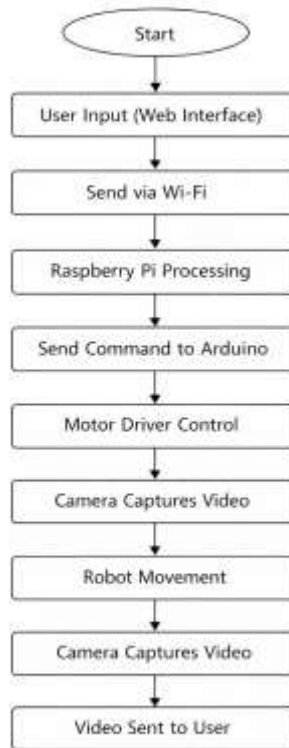


Fig. 2. Flowchart illustrating the working of the proposed robotic surveillance system

D. Feedback and Operation

The system works in a continuous loop, where commands are executed and video feedback is returned to the user. This ensures smooth control and real-time observation.

1) *Real-Time Performance:* Due to efficient communication and processing, the system operates with low latency and provides reliable real-time performance.

VI. IMPLEMENTATION

The implementation of the proposed robotic surveillance system combines hardware and software to achieve real-time video streaming and remote control. The system is designed to be simple and effective for practical use.

A. Hardware Implementation

The hardware setup includes a Raspberry Pi, Arduino, camera module, motor driver, and robotic chassis. The Raspberry Pi manages processing and communication, while the Arduino controls the motors. The camera captures live video of the surroundings.

B. Software Implementation

The software is developed using Python and Flask, enabling communication between the user interface and the robot.

RESULTS

The implemented system was tested for real-time video streaming and remote control using a web-based interface and VNC connectivity. The following figures illustrate the working of the system.



Fig. 3. Internal setup of the system

```

from flask import Flask, jsonify, request
from camera import detect_objects

app = Flask(__name__)
app.config['DEBUG'] = True
app.config['SECRET_KEY'] = 'secret-key'

@app.route('/api/status')
def get_status():
    status = {
        'forward': 0,
        'backward': 0,
        'left': 0,
        'right': 0,
        'stop': 0
    }
    for pin in pins:
        status[pin] = GPIO.read(pin)

    return jsonify(status)

@app.route('/api/control')
def control():
    data = request.json
    command = data.get('command')

    if command == 'forward':
        GPIO.write(forward_pin, 1)
    elif command == 'backward':
        GPIO.write(backward_pin, 1)
    elif command == 'left':
        GPIO.write(left_pin, 1)
    elif command == 'right':
        GPIO.write(right_pin, 1)
    elif command == 'stop':
        GPIO.write(forward_pin, 0)
        GPIO.write(backward_pin, 0)
        GPIO.write(left_pin, 0)
        GPIO.write(right_pin, 0)

    return jsonify({'status': get_status()})

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=5000)
  
```

Fig. 4. Flask-Based Control and Video Streaming

```

<!DOCTYPE html>
<html>
<body align="center">
<h3>Live Stream</h3>


<button onclick="c('forward')">F</button>
<button onclick="c('left')">L</button>
<button onclick="c('stop')">S</button>
<button onclick="c('right')">R</button>
<button onclick="c('backward')">B</button>

<script>
function c(x){
  fetch('/control',{
    method:'POST',
    headers:{'Content-Type':'application/x-www-form-urlencoded'},
    body:'command='+x});
}
</script>
</body>
</html>
  
```

Fig. 5. HTML code for web-based interface enabling live video streaming and robot control panel

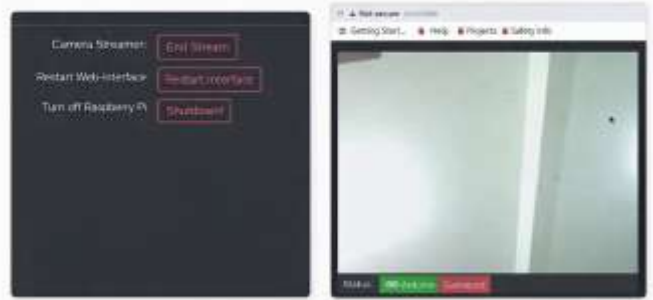


Fig. 6. Web-based interface showing real-time video streaming and robot control panel

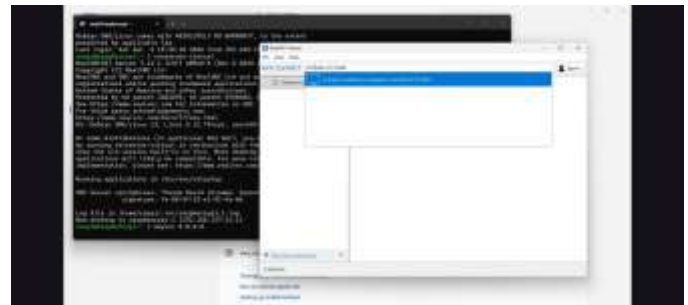


Fig. 7. VNC Viewer used to establish remote connection with Raspberry Pi

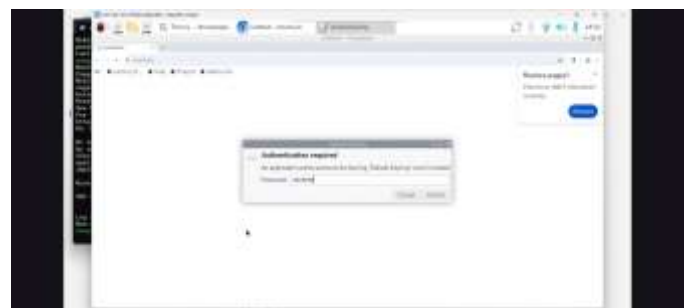


Fig. 8. Authentication process for accessing Raspberry Pi desktop via VNC

The developed intelligent robotic surveillance system was tested using a web-based interface consisting of a control panel and live video feed. The interface enables efficient management of operations such as camera streaming, interface restart, and system shutdown, ensuring quick and responsive execution.

TABLE I
PERFORMANCE EVALUATION OF THE PROPOSED SYSTEM

Parameter	Observed Value
Camera Resolution	5 MP
Video Quality	720p
Frame Rate	~30 FPS
Video Latency	0.2–0.5 s
Response Time	<1 s
Communication Delay	100–300 ms
Control Accuracy	~90%

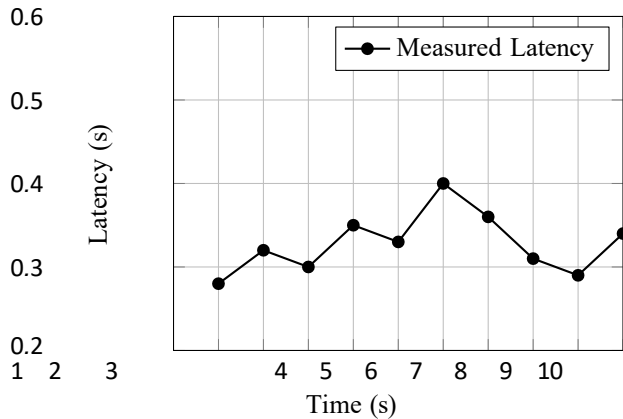


Fig. 9. Measured video latency of the proposed system under normal network conditions

The live video module provides real-time streaming with minimal latency and stable quality under normal network conditions. In addition, status indicators for components such as Arduino and gamepad enhance system usability and monitoring. Communication between the user interface, Raspberry Pi, and Arduino is achieved through Wi-Fi and HTTP protocols, ensuring reliable and low-latency data transfer.

Experimental results demonstrate that the robot responds accurately to user commands, enabling smooth navigation and precise motor control. The system operates consistently without significant errors, indicating its suitability for real-time surveillance applications.

However, system performance may vary depending on network conditions. Future improvements focus on optimizing video transmission, enhancing system security, and integrating AI-based automation like object detection, face recognition for improved functionality.

VII. CONCLUSION

This work presents the development of an intelligent robotic surveillance system capable of real-time monitoring using embedded platforms. By integrating Raspberry Pi 5, Arduino, and a camera module, the system enables efficient remote monitoring and control through a web-based interface, improving accessibility and operational flexibility.

Experimental evaluation confirms stable performance with low-latency video transmission and precise control of robot movement. The combination of mobility and live streaming enhances monitoring capabilities compared to conventional fixed surveillance systems. Moreover, the proposed solution is cost-effective, scalable, and well-suited for applications such as security monitoring, industrial inspection, and smart surveillance environments.

ACKNOWLEDGMENT

We would like to thank Design Space Lab, Department of Electronics and Communication Engineering at Nadimpalli Satyanarayana Raju Institute of Technology (NSRIT) for providing the resources and support required for this research work.

REFERENCES

- [1] M. A. Razaque, M. Milojevic-Jevric, A. Palade, and S. Clarke, "Middle-ware for Internet of Things: A survey," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4323–4350, 2020.
- [2] S. Li, L. Da Xu, and S. Zhao, "The Internet of Things: A survey," *Information Systems Frontiers*, vol. 22, no. 2, pp. 1–17, 2020.
- [3] R. Kumar and S. Tripathi, "Design of mobile robotic system for real-time surveillance," *International Journal of Advanced Research in Computer Engineering*, vol. 9, no. 3, pp. 45–50, 2021.
- [4] P. Sharma and A. Gupta, "IoT-based wireless surveillance robot with live video streaming," *IEEE Access*, vol. 9, pp. 102345–102356, 2021.
- [5] U. D. Bordoloi and A. K. Talukdar, "Raspberry Pi-based smart surveillance system," *Journal of Embedded Systems*, vol. 14, no. 2, pp. 78–85, 2022.
- [6] S. Monk, *Programming Arduino: Getting Started with Sketches*, 2nd ed. New York, NY, USA: McGraw-Hill, 2020.
- [7] H. K. Rathore *et al.*, "Real-time video streaming in IoT-based surveillance systems," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4501–4510, 2021.
- [8] OASIS, "MQTT Version 5.0," OASIS Standard, 2019.
- [9] A. Rosebrock, *Deep Learning for Computer Vision with Python*. PyImageSearch, 2020.
- [10] J. Redmon *et al.*, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [11] S. Wan *et al.*, "Cloud robotics: Current status and open issues," *IEEE Access*, vol. 8, pp. 109000–109015, 2020.
- [12] K. Lee and H. Kim, "Scalable cloud robotics architecture for intelligent systems," *IEEE Transactions on Cloud Computing*, vol. 10, no. 2, pp. 1234–1245, 2022.
- [13] A. Jolfaei *et al.*, "Security and privacy in IoT: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1204–1225, 2021.
- [14] M. Ammar *et al.*, "Internet of Things: A survey on security issues," *Future Generation Computer Systems*, vol. 102, pp. 792–807, 2020.
- [15] X. Zhang *et al.*, "Low-latency communication in IoT-enabled robotic systems," *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4567–4578, 2022.
- [16] M. Macenski *et al.*, "Robot Operating System 2 (ROS2): Design and architecture," *Science Robotics*, vol. 7, no. 66, 2022.
- [17] Y. Mao, C. You, J. Zhang, K. Huang, and K. Letaief, "Mobile edge computing: Survey and research outlook," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 1–25, 2021.
- [18] T. Taleb *et al.*, "On multi-access edge computing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 1–34, 2022.
- [19] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2021.
- [20] D. Boneh and V. Shoup, *A Graduate Course in Applied Cryptography*, 2020.