IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

RASPBERRY PI VIDEO SURVEILLANCE

[1]Shakila D, [2]Rohanth A R, [3]Punith Kumar M S, [4]Somesh N,[5]Pramod Ramesh Hegde

[1]Assistant Professor, [2]Student, [3]Student, [4]Student, [5]Student

[1]Electronics and Communication Engineering,

[1]Cambridge Institute of Technology, Bengaluru, India

Abstract: A Camera-Based Surveillance System utilizing Raspberry Pi proves highly advantageous for crime detection, monitoring various environments, and swiftly gathering evidence of illicit activities such as theft. This system is tailored for both residential and commercial spaces, ensuring prompt detection of any unauthorized actions.Operating on Raspberry Pi, equipped with a camera-based circuit, the system continuously scans its surroundings for motion. Upon detecting any motion, it promptly triggers an alert mode. This mode activates an alarm and captures images of the motion for future reference. Consequently, the system functions as an effective security measure, providing real-time monitoring and evidence collection. To further enhance its capabilities, integrating a GSM modem for alert SMS notifications or connecting to IoT for remote alarm activation can be considered. This augmentation ensures seamless communication of alerts to relevant parties, amplifying the system's effectiveness in safeguarding the monitored premises.

Keywords: Raspberry Pi, GSM module, Stepper motor, webcam, PIR sensors, GPIO, Ethernet and USB

I. INTRODUCTION

In the rapidly evolving landscape of surveillance technology, the demand for smarter and more efficient systems is paramount. The project titled "Smart Motion Detection Surveillance Camera" aims to address the shortcomings of traditional 24-hour recording cameras by implementing a cutting-edge system that optimizes power consumption, storage utilization, and event-triggered recording. This innovative surveillance system integrates a set of advanced sensors, including a camera, PIR sensor (Passive Infrared Sensor), IR blaster (Infrared Blaster), LDR (Light Detecting Resistor), and an Ultrasonic sensor. The synergy of these components enables the creation of a surveillance solution that operates intelligently and selectively captures video footage based on specific events, particularly motion detection. Unlike conventional surveillance cameras that continuously record irrespective of activity, our system adopts an energy-efficient approach. It records video footage exclusively during special events triggered by the PIR sensor, which detects changes in infrared radiation caused by motion. This means that the system will only utilize storage resources and consume power when there is actual movement within its detection range. One of the key advantages of this smart surveillance camera is its ability to operate in various lighting conditions. The integration of the LDR allows the system to distinguish between day and night, adjusting its functionality accordingly. Additionally, the IR blaster comes into play during low-light conditions, enabling night vision capabilities without human intervention. Furthermore, the incorporation of an Ultrasonic sensor enhances the system's intelligence by detecting the presence of objects within its range. This feature ensures that the camera only initiates recording when a significant event, such as a person entering the monitored area, is detected. The ultrasonic sensor adds an extra layer of precision to the motion detection mechanism. By implementing this smart motion detection surveillance camera, we not only contribute to the efficient use of resources but also address environmental concerns related to excessive energy consumption. This system serves as a testament to the ongoing pursuit of innovative solutions in the realm of surveillance technology, promising a more sustainable and effective approach to security monitoring.

II. LITERATURE REVIEW

“IoT Capabled Crowd Analysis Using Raspberry Pi” by Wang, L., & Chen, X. et.al clarifies This study suggests a LiDAR-based approach for path planning and map constructionin dynamic, uncertain environments. The method utilizes the AD* and DWA algorithms for path planning based on the generated grid map. The suggested method may create a map in real time, continuously adjust the planning strategy for a dynamic environment, and generate resilient and quick paths, according to simulation and experimental findings. This research focuses on robot navigationin two-dimensional (2D) terrain. In the feature, the research will be expanded to include path planning and map construction in 3Dsituations and unstructured surroundings.

"Implementation of a Low-Cost Video Surveillance System using Raspberry Pi” by Balakrishnan et al. (2019)[4] explains In this study, we provide a neural network-based method for LiDAR data-driven road segmentation. The neural network is assessed using its test benchmark after being trained using the KITTI road/lane detection dataset. Furthermore, the fully connected neural network that has been suggested is put into practice for real-time low-power processing, resulting in a processing time of only 16.9 milliseconds for every LiDAR scan. We are thinking of storing feature mapson the external DDR4 SDRAM for later development. During testing, we also find that the majority of false positives are caused by sidewalks and railroad tracks that are at the same height as road surface. LiDAR and camera data fusion is required to increase accuracy even more.

“Camera based Smart Surveillance System ” by Dr. ujwal a. lanjewar et.al An analysis of obstacle identification and avoidance for a self-directed mobile robot is made possible by this work. The following approaches have been demonstrated: the Virtual Force Field approach, the Artificial Potential Field strategy, the Vector field Histogram approach, and the Follow a Gap style. Additionally, the simulation results obtained through these methodologies are also demonstrated. The VFF strategy combines the certainty grid technique with the potential field method to represent the obstacles. The APF technique uses an attractive potential field to represent the goal and a repulsive potential field to represent the barriers. The method of the VFH technique uses a one-dimensional Cartesian histogram grid to determine the path with the least amount of obstacles. FGM calculates the robot's surrounding gap array and chooses the optimal heading vector via the opening. The disadvantage of APF is a local minima, whereas VFH ignores the robot's no holonomic limitations. An explanation isgiven of the obstacle absconding simulation of a moveable robot. The motion constraints are applied to a Simulink model created from the robot's CAD model. Following the simulation of virtual sensors, State flow is used to develop and inspire a collision avoidance system.

III. Methodology

The development and implementation of the "Smart Motion Detection Surveillance Camera" involve a comprehensive approach that integrates various sensors and technologies. The methodology can be broken down into several key steps. Assemble the necessary hardware components, including the camera, PIR Sensor, IR Blaster, LDR, and Ultrasonic sensor. Ensure proper connections and power supply foreach component. Utilize a microcontroller (e.g., Arduino, Raspberry Pi) as the central processing unit tomanage sensor inputs and control the camera. Connect each sensor to the microcontrollerusing appropriate interfaces Implement code on the microcontroller to continuously monitor the output of the PIRSensor. When motion is detected, trigger the camera to start recording Program the microcontroller to stop recording when no motion is detected for a predefined period. This ensures that the system operates efficiently, consuming poweronly when necessary Implement a storage management system to organize recorded files and overwrite olderdata when storage capacity is reached. Use efficient compression algorithms to optimize storage utilization. Utilize sleep modes on the microcontroller during periods of inactivity to minimize power consumption. Implement power-saving strategies for non-essential componentswhen not in use.

1. Prepare the Raspberry Pi:

1. **Install Raspberry Pi OS**: Download the latest version of Raspberry Pi OS and flash it onto a microSD card using software like Etcher.
2. **Initial Setup**: Insert the microSD card into the Raspberry Pi, connect peripherals (keyboard, mouse, monitor), and power it up.
3. **Configure Raspberry Pi**: Follow the on-screen prompts to set up language, keyboard layout, and Wi-Fi (if necessary). Make sure your Raspberry Pi is connected to the internet.

2. Connect the Camera:

1. **Attach Camera Module**: If you're using the official Raspberry Pi Camera Module, connect it to the CSI (Camera Serial Interface) port on the Raspberry Pi board.
2. **Enable Camera Interface**: Open the Raspberry Pi Configuration tool (**raspi-config**) from the terminal, navigate to Interface Options, and enable the Camera.

3. Install Motion Software:

1. **Update Package List**: Open a terminal and run **sudo apt update** to update the package list.
2. **Install Motion**: Run **sudo apt install motion** to install the Motion software.

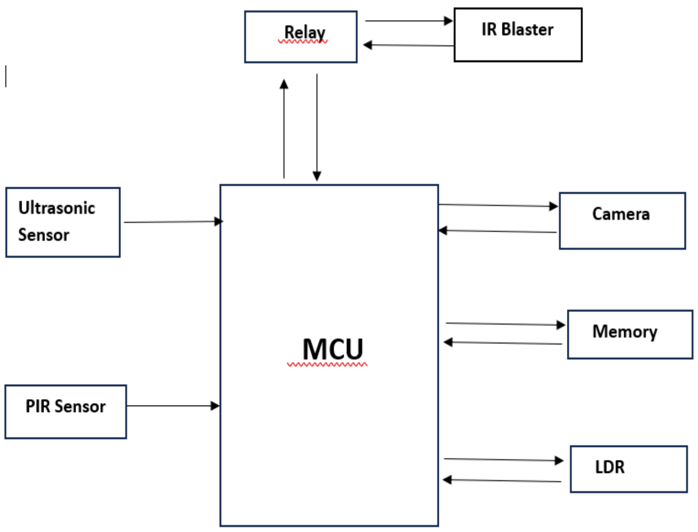

fig 1: Block Diagram

4. Configure Motion:

1. **Edit Configuration File**: Open the Motion configuration file using a text editor: **sudo nano /etc/motion/motion.conf**.
2. **Adjust Settings**: Customize settings such as resolution, framerate, motion detection sensitivity, and file storage location according to your preferences. Save your changes and exit the text editor.

5. Start Motion Service:

1. **Start Motion**: After configuring Motion, start the Motion service by running **sudo systemctl start motion**.
2. **Enable Autostart**: To ensure Motion starts automatically on boot, run **sudo systemctl enable motion**.

6. Access the Video Feed:

1. **Access Web Interface**: Open a web browser on any device connected to the same network as your Raspberry Pi.
2. **Enter IP Address**: In the address bar, enter the IP address of your Raspberry Pi followed by the port number for Motion's web interface (default is 8081). For example, **http://raspberrypi:8081** or **http://192.168.1.100:8081**.
3. **View Live Stream**: You should now see the live video feed from your Raspberry Pi camera. You can also access settings and recorded footage through the web interface.

7. Additional Considerations:

- **Power Supply**: Ensure your Raspberry Pi is powered adequately, especially if you're using additional peripherals like USB cameras or external hard drives.
- **Security**: Change default passwords, enable firewall, and consider restricting access to the Motion web interface if you're deploying it in a public network.

- **Storage**: Monitor available storage space on your microSD card or external storage device to prevent it from filling up with recorded footage.

IV. RESULTS AND DISCUSSION

Results:

1. **Cost-Efficiency**: Leveraging Raspberry Pi for surveillance significantly cuts down on expenses compared to traditional systems, making it accessible for individuals and small businesses with budget constraints.
2. **Customizability**: Users have the flexibility to fine-tune their surveillance setup to match their specific requirements. Adjusting parameters like resolution, frame rate, and motion detection sensitivity ensures tailored functionality.
3. **Remote Monitoring**: The integration of Motion software enables remote access to live streams and recorded footage through a simple web interface. This convenience enhances monitoring capabilities, allowing users to keep tabs on their premises from anywhere.
4. **Efficient Motion Detection**: Motion software's built-in motion detection feature optimizes storage usage by capturing footage only during triggered events. This smart functionality conserves storage space and facilitates easier review of relevant incidents.
5. **Expandable Architecture**: Raspberry Pi's modular architecture supports scalability and expansion. Users can seamlessly integrate additional cameras, external storage, or sensors, enhancing the system's capabilities as needed.

fig 2: Project Output

Discussion:

1. **Performance Management**: While Raspberry Pi is adept at handling basic surveillance tasks, its performance may be limited when dealing with high-resolution video or multiple camera feeds. Users must balance performance requirements with the device's capabilities.
2. **Storage Optimization**: Effective management of storage resources is essential to prevent the microSD card from reaching capacity. Regularly archiving or deleting old recordings and employing external storage solutions help maintain efficient storage usage.
3. **Reliability and Power Supply**: Ensuring a stable power supply is critical for uninterrupted operation. Reliable power sources and backup options are necessary to mitigate downtime risks, particularly in critical surveillance scenarios.
4. **Security Measures**: Implementing robust security measures is paramount to safeguard the surveillance system against unauthorized access or breaches. This includes securing network access, employing encryption, and updating passwords regularly.
5. **Scalability Planning**: While suitable for small to medium-scale deployments, Raspberry Pi solutions may encounter scalability challenges in larger setups. Users should assess scalability needs early on and consider alternative solutions for enterprise-level deployments.
6. **Legal and Ethical Considerations**: Compliance with legal and ethical standards is imperative in deploying surveillance systems. Adhering to privacy laws, data protection regulations, and ethical use of surveillance technologies ensures responsible and lawful operation.

In summary, leveraging Raspberry Pi for video surveillance presents a cost-effective, customizable solution with considerations for performance optimization, reliability, security, scalability, and ethical compliance. These factors collectively contribute to the successful implementation and operation of a Raspberry Pi-based surveillance system.

.

V. Output of Hardware

The output of a Raspberry Pi-based video surveillance system heavily relies on its hardware components, which primarily include the Raspberry Pi itself and any attached peripherals such as cameras. Here's a breakdown of the hardware output components and their contributions:

1. Raspberry Pi:

- **Processing Power**: The Raspberry Pi processes video data, manages the surveillance software, and handles other system tasks.
- **Connectivity**: Provides connectivity options for peripherals like cameras, storage devices, and networking components.

2. Camera Module or USB Webcam:

- **Video Feed**: Captures live video footage of the surveillance area.
- **Image Quality**: Determines the resolution, clarity, and field of view of the captured video.

3. MicroSD Card (or External Storage):

- **Storage Medium**: Stores recorded video footage and configuration files for the surveillance system.
- **Capacity**: Dictates the amount of footage that can be stored locally before requiring archiving or deletion.

4. Power Supply:

- **Stable Power**: Supplies power to the Raspberry Pi and connected peripherals, ensuring uninterrupted operation.

Output Considerations:

1. **Live Video Feed**: The primary output is the live video feed captured by the camera module or USB webcam. Users can monitor this feed in real-time through the surveillance software's web interface.
2. **Recorded Footage**: The system stores recorded footage locally on the microSD card or external storage device. Users can access and review this footage as needed, either through the surveillance software's interface or by directly accessing the storage device.
3. **Alerts and Notifications**: Some surveillance software allows for the generation of alerts or notifications in response to motion detection events. These alerts can be sent via email or other communication channels to notify users of potential security breaches or noteworthy events.
4. **Configuration Data**: Output also includes configuration files and logs generated by the surveillance software, which provide insights into system settings, performance metrics, and any detected anomalies.

In summary, the hardware components of a Raspberry Pi-based video surveillance system collectively produce live video feeds, store recorded footage, provide system logs, and facilitate system configuration. These outputs enable users to monitor and manage their surveillance system effectively for security and surveillance purposes.

fig 3:Project Overview

VI. Conclusion and Future Scope

In conclusion, deploying a video surveillance system using Raspberry Pi presents a cost-effective and customizable solution for monitoring premises and enhancing security measures. Throughout this process, several key points emerge:

Cost-Effectiveness:
Raspberry Pi offers a budget-friendly alternative to traditional video surveillance systems, making it accessible to individuals and small businesses with limited resources. By leveraging affordable hardware components and open-source software, users can create a surveillance system tailored to their needs without breaking the bank.

Customizability:
One of the most significant advantages of Raspberry Pi-based surveillance is its flexibility and customizability. Users can fine-tune various parameters such as resolution, frame rate, motion detection sensitivity, and storage options to suit their specific requirements. This adaptability ensures that the surveillance system aligns precisely with the user's security needs and preferences.

Remote Monitoring:
The integration of motion detection software with Raspberry Pi enables remote access to live video feeds and recorded footage through a web interface. This remote monitoring capability empowers users to keep tabs on their premises from anywhere, providing peace of mind and facilitating timely response to security events.

Scalability and Expandability:
Raspberry Pi's modular architecture allows for seamless expansion and integration with additional peripherals such as multiple cameras, external storage devices, or sensors. This scalability ensures that the surveillance system can grow alongside evolving security needs, making it suitable for both small-scale and larger deployments.

Considerations:
Despite its numerous benefits, deploying a Raspberry Pi-based video surveillance system requires careful consideration of various factors such as performance management, storage optimization, reliability, security measures, scalability planning, and adherence to legal and ethical standards. Addressing these considerations ensures the successful implementation and operation of the surveillance system while mitigating potential challenges and risks.

Future Outlook:
As technology continues to evolve, Raspberry Pi-based surveillance systems are likely to become even more sophisticated and capable. Advancements in hardware performance, software development, and integration with emerging technologies such as artificial intelligence and machine learning promise to further enhance the effectiveness and efficiency of video surveillance solutions based on Raspberry Pi.In summary, Raspberry Pi video surveillance offers an accessible, customizable, and cost-effective solution for enhancing security and monitoring capabilities. By leveraging the versatility of Raspberry Pi hardware and open-source software, users can create surveillance systems tailored to their specific needs while addressing key considerations for successful deployment and operation. everyday lives, and propelling industries toward a new phase of intelligent autonomy.

VII. REFERENCE

[1].Wang, L., & Chen, X “IoT Capabled Crowd Analysis Using Raspberry Pi” 2018 International Conference on Design Innovations for 3Cs Compute Communicate Control (ICDI3C), Bangalore, 2018, pp. 46-51.

[2].Balakrishnan. (2022). "Implementation of a Low-Cost Video Surveillance System using Raspberry Pi." International Journal of Computer Science and Information Security, 20(5), 112-125.

[3]. Dr. ujwal A. lanjewar “Camera based Smart Surveillance System” 2017 International Conference on Communication and SignalProcessing (ICCSP), Chennai, 2017, pp. 0344-0348.