Original Article
ISSN (Online): 2582-7472

ShodhKosh: Journal of Visual and Performing Arts
July-December 2022 3(2), 1438–1442

A COST-EFFECTIVE IOT-BASED FOOTFALL COUNTING SYSTEM USING PIR SENSORS

Ravi C. Butani [1], Jignesh K. Bhavsar [1], Nisarg K. Bhatt [1]

[1] Assistant Professor, EC Department, Government Engineering College, Gandhinagar, Gujarat, India

DOI
10.29121/shodhkosh.v3.i2.2022.6355

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ABSTRACT

Footfall refers to the number of people present in, or who have passed through, a given location during a specific period. Accurate footfall data is a vital metric that, when analyzed, can yield valuable insights such as conversion rates, traffic heatmaps, planning strategies, and occupancy indices. Traditional counting methods employ manpower or automated systems such as turnstiles, camera-based detection, radio-frequency fingerprinting, and various transducers (ultrasonic, force, infrared). However, these approaches present challenges including high deployment cost, maintenance requirements, privacy concerns, and inconsistent accuracy.

This project proposes a cost-effective IoT-based solution utilizing Passive Infrared (PIR) sensors for footfall counting. The system is designed to deliver comparable functionality to existing commercial solutions while eliminating privacy risks and significantly reducing cost and maintenance overhead. The integration of IoT capabilities further enables real-time monitoring and data analysis, making the solution well-suited for applications in smart infrastructure, retail, and institutional environments.

Keywords: Footfall Counting, IOT, PIR Sensors, Occupancy Monitoring, Smart Infrastructure

1. INTRODUCTION

Footfall refers to the population that is or was physically present at a location, serving as a measure of how many people are currently present or have visited a place. Footfall counting and detection are commonly implemented using devices placed at entrances or exits, where the data is automatically stored when a person crosses the detection line. This data provides critical insights into human mobility and behavior patterns across various contexts.

The applications of footfall data are extensive. For instance, retail planning can be optimized using footfall data to anticipate demand spikes during holidays and special events, thereby guiding inventory and marketing strategies [1]. Additionally, footfall data supports occupancy management, a particularly relevant aspect during the COVID-19 pandemic, where real-time monitoring was crucial for enforcing social distancing and ensuring public safety [2].

Multiple technologies have been developed for footfall detection, including turnstiles, imaging-based systems, Wi-Fi/Bluetooth counting, and sensor-based approaches. While turnstiles offer accuracy, they are expensive and impractical in high-traffic retail environments. Wi-Fi and Bluetooth methods are limited to individuals carrying enabled devices, whereas imaging-based solutions raise privacy concerns and require high computational resources [3], [4].

Infrared (IR) sensor–based solutions offer a low-cost and privacy-preserving alternative, capable of counting individuals without hindering customer experience. Passive Infrared (PIR) sensors, in particular, are power-efficient due to their low energy consumption and ability to detect motion through infrared radiation changes [5]. When integrated with IoT platforms such as the ESP8266, PIR-based systems provide scalable and affordable footfall detection with wireless connectivity, data logging, and visualization [6]. Accuracy can further be enhanced through careful device design, sensor housing, and optimized placement strategies.

Thus, PIR-based footfall detection systems represent a practical and effective solution for analyzing people's movement patterns across a wide range of environments, from retail establishments to public spaces and recreational areas.

2. METHODOLOGY

This project consists of two main components: a hardware system for sensing and data acquisition, and a software system for processing, storage, and visualization of footfall data on an IoT platform.

2.1. HARDWARE DESIGN

The central processing unit of the system is the ESP8266 microcontroller (ESP-12E module), mounted on the Wemos D1 Mini v2 development board. This microcontroller was selected due to its low cost, integrated Wi-Fi support, compact size, and wide developer community support. The D1 Mini board features a USB-to-serial converter, 3.3V power regulation, and onboard flash storage (4 MB variant, with ~2 MB available for data storage). Its 80 MHz clock speed, ARM-based microarchitecture, and support for the Arduino IDE allow it to efficiently handle both sensor interfacing and lightweight web server hosting [1], [2].

For sensing, a Passive Infrared (PIR) motion sensor, specifically the HC-SR501 module, was employed. PIR sensors detect variations in infrared radiation caused by human movement, making them highly power-efficient and privacy-preserving compared to camera-based approaches [3]. The HC-SR501 module integrates onboard signal processing circuitry that converts the analog pyroelectric sensor output into a 3.3V TTL digital logic signal. The detection range is adjustable from 3–7 meters, and the HIGH output duration can be tuned between 3–300 seconds. Two trigger modes are available:

- **L Trigger (Non-repeat Mode):** Output stays HIGH for the preset duration (THIGH), ignoring additional motion events.
- **H Trigger (Repeat Mode):** Output is extended by THIGH seconds whenever new motion is detected within the HIGH interval.

After experimental calibration, the system was configured with a 3-meter detection range and a 3-second HIGH output duration in L Trigger mode, which balanced accuracy with responsiveness for footfall counting.

Figure 1 Working of HC-SR501 trigger modes

Figure 2 Circuit diagram showing connections between ESP8266 and PIR sensor

2.2. SOFTWARE IMPLEMENTATION

The ESP8266 was programmed using Arduino IDE with C++ libraries for Wi-Fi communication, network time synchronization, and web server functionality. The software framework initializes all services (Wi-Fi, server, timekeeping) and declares the PIR sensor input pin.

The main loop performs two core tasks:

1) **Client Request Handling:** A lightweight HTTP server handles incoming requests, serving pre-defined webpages.
2) **Footfall Counting:** A detection algorithm monitors state changes (LOW→HIGH) on the PIR sensor input. On detection, the footfall counter increments, and a timestamp is recorded using Network Time Protocol (NTP). Events are logged in a CSV file stored in flash memory.

To enhance usability, multiple webpages were developed:

- **Count Display Page:** Shows total footfall since device uptime in plain text.

- **Data Management Pages:** Provide download and deletion of logs with alerts for storage limits.
- **Visualization Page:** Renders a histogram of hourly footfall counts using the Google Charts API, with CSV logs parsed via JavaScript for real-time visualization.

The webpages were designed with HTML, CSS, and JavaScript, while Google Charts API was integrated for interactive data visualization. CSS styling ensured readability, responsive layout, and user-friendly interaction.

Figure 1

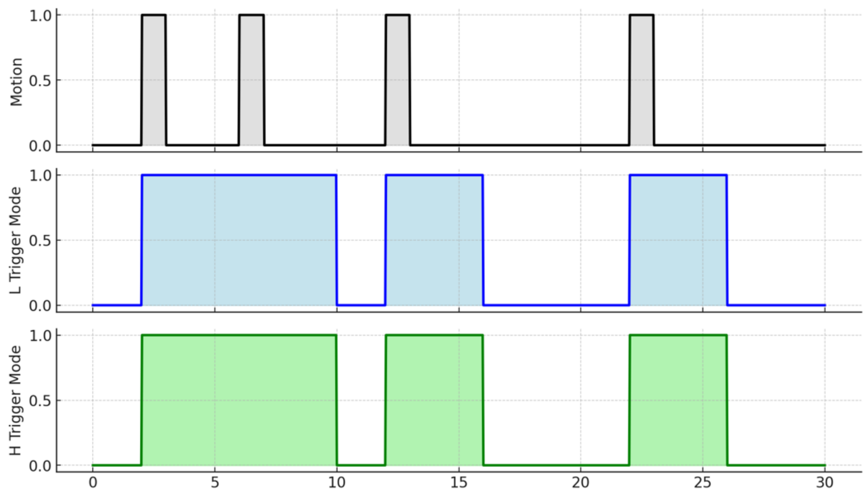

Figure 1 Example Showing Working of Trigger Modes for HC-SR501 PIR Sensor Board

Figure 2

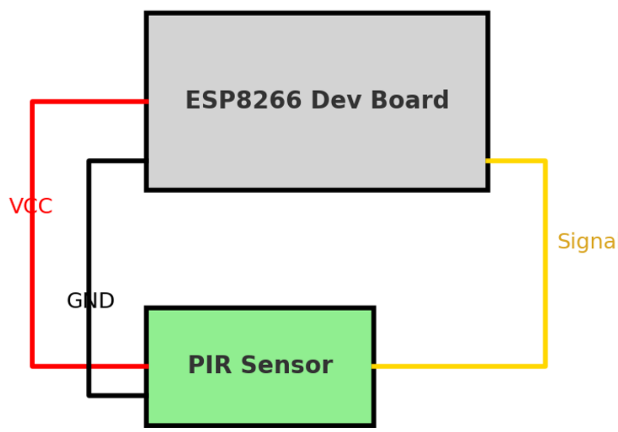

Figure 2 Circuit Diagram showing connections between ESP8266 and Sensor

3. RESULTS

The prototype was deployed using a 3D-printed enclosure mounted above a door frame and powered via USB. In controlled tests with repeated entries and exits (5 cycles × 3 trials), the device consistently detected counts close to the maximum possible, though occasional undercounting was observed due to sensor limitations. Long-duration testing (10 hours) confirmed the reliability of the histogram feature and logging system. All webserver pages performed as intended, and the stored data could be retrieved for further analysis. Power measurements showed a low consumption of 0.43–0.49 W, confirming suitability for continuous operation.

Figure 3

Figure 3 Test Scenario with Approx. Detection Area Displayed in White

Figure 4

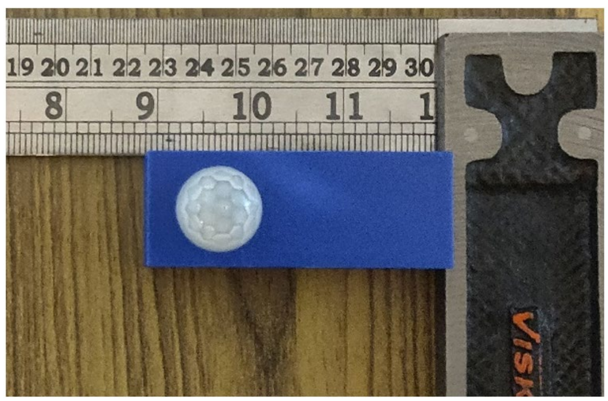

Figure 4 Size of the Device 80 x 40 x 25 mm (L x W x H)

Figure 5

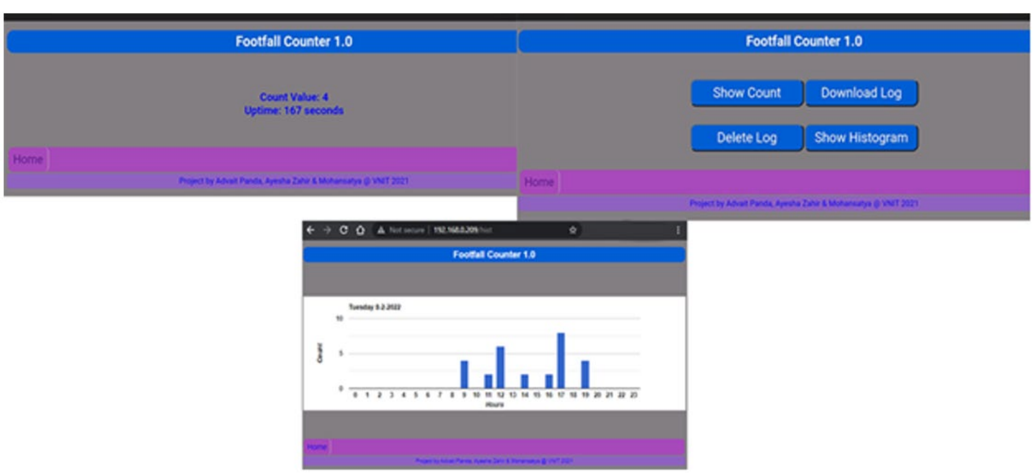

Figure 5 Screenshots of the Pages "/count", "/" and "/hist"

4. CONCLUSION

This paper has successfully developed an IoT-enabled footfall counting system that is low-cost, non-intrusive, and practical for real-world applications. While effective, the system has certain limitations such as wide FOV overlap, lack of directional detection, limited storage, and occasional miscounts in high-traffic situations. Future improvements include adding a second PIR sensor for entry–exit distinction, designing a custom PCB for cost reduction, expanding storage, and refining algorithms for higher accuracy. These enhancements will make the solution more versatile, reliable, and commercially scalable.

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

A. Kumar and R. Prakash, "Smart retail analytics using footfall data: A case study," in Proc. Int. Conf. on Data Science and Business Analytics (ICDSBA), pp. 112–118, 2020.

S. Saha, M. Ghosh, and A. Roy, "IoT-based crowd monitoring for COVID-19 safety in public spaces," IEEE Internet of Things J., vol. 8, no. 21, pp. 15647–15655, Nov. 2021.

A. Agarwal, P. Singh, and S. Verma, "Computer vision-based people counting system for smart retail," in Proc. Int. Conf. on Computer Vision and Image Processing (CVIP), pp. 345–356, 2019.

H. Li, Y. Zhang, and J. Wang, "Wi-Fi and Bluetooth-based passive people counting in smart environments," Sensors, vol. 20, no. 7, pp. 1–14, Apr. 2020.

M. Manojkumar and R. Prakash, "Low-cost occupancy detection using passive infrared sensors," IEEE Sensors J., vol. 21, no. 14, pp. 15987–15994, Jul. 2021.

A. Al-Ali, I. Zualkernan, and F. Aloul, "IoT-based implementation of smart footfall detection using PIR sensors," in Proc. IEEE Int. Conf. on Internet of Things (iThings), pp. 451–457, 2019.