International Transactions on Education Technology (ITEE)
Vol. 4, No. 2, May, 2026, pp. 149−161
E-ISSN: 2963-1947 | P-ISSN: 2963-6078, DOI:10.33050

IoT-Based Home Security System: ESP32-CAM Integration and Real-Time Notification

Qori Halimatul Hidayah[1*], Ryan Putra Laksana[2], Ary Prabowo[3], Dwi Santika Simatupang[4]
[1]Faculty of Communication, Universitas Esa Unggul, Indonesia
[2]Faculty of Information Systems, Universitas Esa Unggul, Indonesia
[3, 4]Faculty of Computer Science, Universitas Esa Unggul, Indonesia
[1]qori.halimatul@esaunggul.ac.id, [2]ryan.putra@esaunggul.ac.id, [3]ary.prabowo@esaunggul.ac.id, [4]dwi.sartika@esaunggul.ac.id
***Corresponding Author**

Article Info

Article history:

Submission October 30, 2025
Revised November 01, 2025
Accepted February 20, 2026
Published May 20, 2026

Keywords:

Internet of Things
ESP32-CAM
PIR Sensor
Telegram Bot
Home Security System

ABSTRACT

Conventional home security systems are generally unable to provide real-time monitoring accompanied by visual evidence when suspicious activity occurs, so the user response rate is still low. The development of the Internet of Things (IoT) allows the integration of sensors, cameras and communication platforms to produce more adaptive and responsive security systems. **This research aims** to develop a prototype IoT-based home security system using ESP32-CAM and PIR sensors integrated with the Telegram Bot API to provide image-based automatic notifications. **The research** was carried out through the stages of needs identification, literature study, design, prototype development, and functional testing. The system is designed to detect movement via a PIR sensor, take images using the ESP32-CAM, store them on MicroSD media, and send real-time notifications to users via the Telegram platform. **Test results** show that the system is able to detect movement accurately, take images with stable quality, and send notifications with a 100% success rate as long as a network connection is available. Moreover, there is also the provision of manually executing /takefoto commands for remotely monitoring the system. **This research** work contributes towards designing a home security system which is not only cost-effective but is also relatively simple to implement and highly adaptable in the community environment. Future development could involve cloud data storage and video recording capabilities.

DOI: https://doi.org/10.33050/itee.v4i2.956

1. INTRODUCTION

Home security plays a critical role in protecting individuals and their property from theft, vandalism, and unauthorized intrusion [1]. The increasing rate of criminal activity in residential environments has highlighted the need for reliable and responsive security systems capable of providing continuous monitoring and rapid alerts. With crime rates rising in so many neighborhoods as everything is all needs fast and profiling systems that can keep an eye on the whole thing [2].

What most people have is a standard alarm and CCTV setup but that isn't enough in many ways. However, there is often a lack of real time communication and limited range to monitor areas [3]. Then there's sort of a second-order problem, which is you don't get immediate visual evidence, because I think that compounds the difficulty for people to have to know what they do next. And, well that seems sort of obvious but still this lowers how effective those decisions are.

The IoT devices have been instrumental in developing advanced security systems due to integration of sensors and cameras with a variety of digital platforms operating in real-time [4, 5]. Devices such as ESP32-CAM and PIR sensors are popular for their ability to detect motion, take pictures and transmit information via the Internet at a relatively affordable price. It appears that affordability is the main reason behind this popularity [6].

There is plenty of literature related to systems implemented with the help of ESP32-CAM module connected to servers, or even to Telegram as a means of notification service, which, in my opinion, allows everyone to access all information easily [7]. However, there are also mobile applications available that enable one to remotely observe the performance of the camera. Nevertheless, there is something wrong about most of the existing constructions and approaches to their realization. Firstly, they are far from perfect regarding two-way communication, and, secondly, the integration between motion sensing and automatic photo capturing and sending the pictures within new media is not always the best solution to implement [8, 9].

Consequently, this research will focus on creating a convenient and innovative home security system using ESP32-CAM technology and the Telegram application as its main mode of communication [10]. By means of this system, users are able to use Telegram to issue commands for capturing images, and then the captured images would be returned to the user in real time [11]. Even though this method is not fully automated, the system is efficient enough to enable users to monitor their homes remotely. Thus, this system should become a useful tool for securing homes.

Several previous studies have developed IoT-based home security systems using ESP32-CAM technology for surveillance and motion detection purposes [12], [13], [14]. However, most existing systems mainly focus on basic monitoring or object detection functionalities without providing an integrated mechanism for real-time image transmission, lightweight remote interaction, and flexible notification management through a single communication platform. In addition, some systems still rely on more complex architectures or additional supporting devices, which may reduce implementation efficiency and accessibility for household-scale users.

Therefore, this study proposes an IoT-based home security system that integrates ESP32-CAM, PIR sensors, MicroSD storage, and Telegram Bot API into a unified and cost-effective architecture capable of automatic motion detection, real-time image transmission, and manual remote image acquisition through Telegram commands. The novelty of this research lies in the combination of autonomous image-based notification features and simplified remote interaction mechanisms within a lightweight and practical IoT security framework suitable for community and residential implementation.

2. RESEARCH METHOD

The research methodology use the rototyping-based System Development, as show Figure 1.

The prototyping process in this study consists of five interrelated phases: need assessment, prototype design, development, implementation and testing, and evaluation of effectiveness [15]. In the initial phase, a comprehensive identification of system requirements is conducted to ensure that the proposed solution aligns with real-world user needs. The system is designed to automatically detect motion and capture images instantly using the ESP32-CAM module, while simultaneously sending real-time notifications through Telegram services [16]. These requirements reflect the growing demand for intelligent security systems that emphasize responsiveness, automation, and remote accessibility [17]. Furthermore, the integration of image-based evidence with instant alerts enhances user awareness and decision-making capabilities. By defining these functional requirements early in the process, the study establishes a clear foundation for designing a system that not only detects intrusions but also provides meaningful and actionable information to users [18].

The second phase focuses on component selection and system workflow design, which are critical for ensuring system efficiency and reliability. In this stage, the ESP32-CAM is selected as the primary processing and imaging unit due to its compact design and built-in Wi-Fi capability, making it highly suitable for IoT-based applications [19]. The Passive Infrared (PIR) sensor is utilized for motion detection because of its sensitivity to infrared radiation changes, allowing it to effectively identify human movement. Additionally, the Telegram Bot API is integrated as a communication medium to facilitate real-time notification delivery [20, 21]. Beyond selecting components, the system workflow is carefully structured and represented through diagrams to illustrate the sequence of operations, from motion detection to image transmission [22]. This

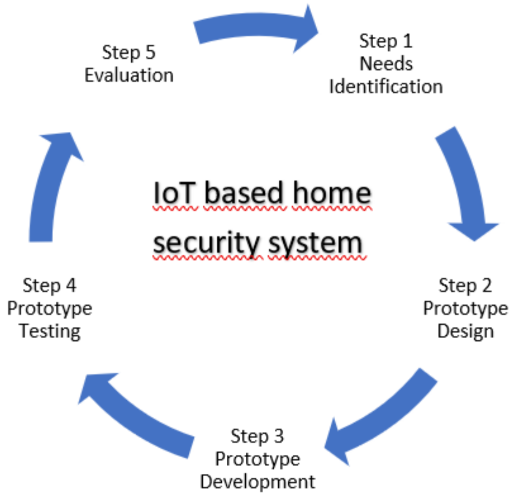

Figure 1. Prototyping-based System Development

structured approach ensures that each component interacts seamlessly, minimizing potential system conflicts. The thoughtful combination of these technologies results in a cohesive and efficient system architecture capable of achieving the intended security objectives [23].

The development phase represents a critical transition from conceptual design to a functional prototype through the integration of hardware and software components in a structured manner [24]. In this stage, the system architecture is implemented by combining the ESP32-CAM module, the PIR motion sensor, and the communication interface into a cohesive and coordinated unit [25, 26]. Programming is carried out using the Arduino IDE, where the core logic is developed to manage motion detection, automated image capture, and real-time wireless data transmission efficiently. Particular attention is given to the synchronization between the PIR sensor and the ESP32-CAM to ensure that image capture is triggered immediately upon motion detection, thereby minimizing latency and improving system responsiveness [27]. In addition to implementing core functionalities, this phase also involves optimizing system parameters such as detection sensitivity, image resolution, and transmission intervals to achieve a balance between performance, efficiency, and resource utilization [28]. Considerations related to memory usage, power consumption, and network stability are also addressed to enhance system reliability and consistency. The integration process requires iterative debugging and validation to ensure seamless communication between components and to eliminate potential errors or delays during operation [29]. The resulting system is a minimum viable prototype (MVP) that demonstrates the core capabilities of the proposed solution under initial conditions. However, the prototype may still have limitations in processing efficiency, environmental adaptability, and overall system stability, which provide valuable insights and opportunities for further refinement. Through continuous testing, evaluation, and incremental improvements, the system can evolve into a more robust, reliable, and scalable solution that is better suited for real-world applications and deployment scenarios [30].

The implementation and testing phase was conducted to validate the effectiveness, responsiveness, and operational reliability of the proposed IoT-based home security system. The evaluation process focused on several critical aspects, including the accuracy of the PIR sensor in detecting motion, the image capture performance of the ESP32-CAM module, and the reliability of real-time notification delivery through the Telegram platform [31]. Testing was performed systematically under various environmental conditions, such as differences in lighting intensity, object distance, and movement scenarios, to simulate real-world operational situations and evaluate overall system stability [32].

The final phase is evaluation, which aims to measure the overall effectiveness of the system in meeting predefined requirements and user expectations [33]. This phase is inherently iterative, as the system undergoes repeated cycles of testing and refinement until satisfactory performance is achieved. The evaluation process not only assesses technical performance but also considers usability and reliability in practical applications [34]. In addition, this study employs three primary data collection methods: literature review, observation, and experimentation. The literature review provides a solid theoretical foundation related to Internet of Things (IoT) technologies, ESP32-CAM modules, PIR sensors, and the Telegram Bot API, ensuring that the system design is grounded in established knowledge [35]. Observations and experiments are conducted to validate system performance, particularly in motion detection, image acquisition, and real-time notification delivery. Compared to conventional security systems such as CCTV and alarm systems, which often lack immediate notification and integrated visual evidence, the proposed system offers significant improvements in responsiveness and user interaction [36]. Ultimately, the iterative nature of this prototyping approach enables the development of a flexible, adaptive, and user-centered security system, as illustrated in Figure 2 and Figure 3, which present the development stages and operational workflow of the IoT-based home security system [37].

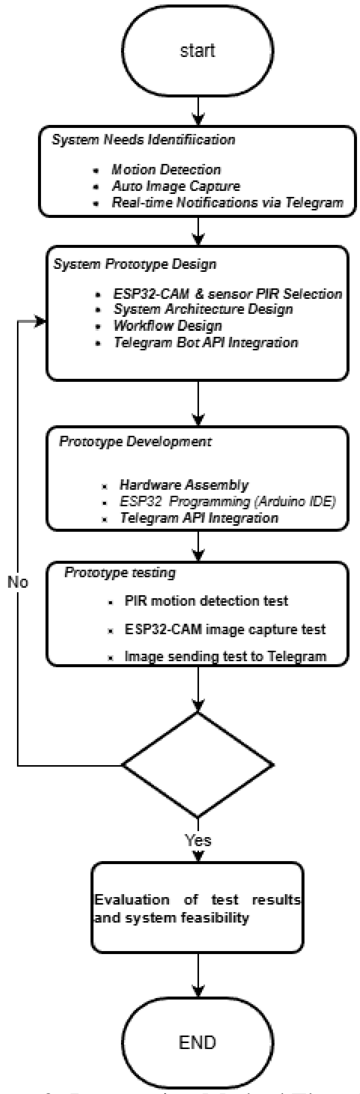

Figure 2. Prototyping Method Flowchart

Furthermore, the ESP32-CAM module was assessed in terms of image clarity, capture responsiveness, transmission speed, and wireless communication reliability to ensure that notifications and captured images could be delivered efficiently without significant delay [38], [39]. The testing process also examined the consistency of the PIR sensor in generating accurate motion detection signals and evaluated the responsiveness of the integrated Telegram Bot communication system [40], [41]. Through this evaluation process, potential technical limitations such as delayed notifications, reduced detection sensitivity, and communication instability were identified and addressed through iterative system refinement. This structured testing approach improved the readability of the methodology section while maintaining comprehensive technical evaluation of the developed system.

Figure 3 presents the workflow of the Internet of Things (IoT)-based home security system, illustrating the sequence of operations from motion detection to notification delivery. The flowchart demonstrates how the system integrates the ESP32-CAM, PIR sensor, and Telegram Bot API into a unified operational framework. The process begins with system activation, where all hardware components and software services are initialized to ensure proper functionality. Once activated, the PIR sensor continuously monitors the surrounding environment for any motion. When movement is detected, the system immediately triggers the ESP32-CAM to capture an image, which is then processed and transmitted via a Wi-Fi connection. The captured image, along with a notification message, is sent to the user through the Telegram platform, enabling real-time monitoring and rapid response. This workflow highlights the efficiency and automation of the system, demonstrating how multiple technologies can be seamlessly integrated to provide an effective and user-friendly home security solution.

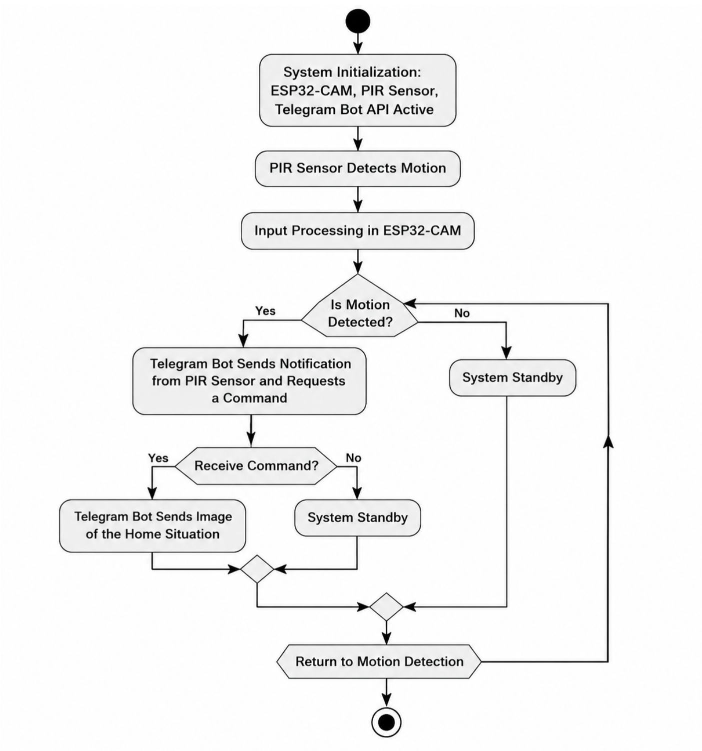

Figure 3. System workflow

Once the system is activated, the PIR sensor detects movement or suspicious activity in the monitored area. The detected data from the PIR sensor is then received by the ESP32-CAM for further processing. If no movement is detected, the system remains in standby mode and continues to monitor the environment continuously. Conversely, if the PIR sensor detects movement, the ESP32-CAM is activated to capture images. This image capture process can be performed automatically or through user command using the "/takefoto" command in Telegram Bot.

Upon successful capturing of the images, the ESP32-CAM then proceeds to transmit these images together with a message using the Telegram Bot API to the user's Telegram account. Through this system, the residents will be able to get live information about what is happening in their surroundings at home. Upon completing the sending process, the device resumes to its standby position, and this whole process starts anew.

3. FINDINGS

Figure 4 shows how the interaction works between a user and the home security system based on the Passive Infrared (PIR) motion sensor together with the Telegram application as a means of communication. When movement is detected by the PIR motion sensor, the home security system will automatically send the user the first message through Telegram Bot. Once this is done, the user can interact with the home security system using the /takefoto command in order to trigger the ESP32-CAM to capture the monitored area. The image captured by the device is then sent in real time throughThis approach enables the user to have complete control over the image capturing process, thereby ensuring efficiency in usage of storage space and bandwidth capacity of the system. image in real time through Telegram. This approach enables the user to have complete control over the image capturing process, thereby ensuring efficiency in usage of storage space and bandwidth capacity of the system.

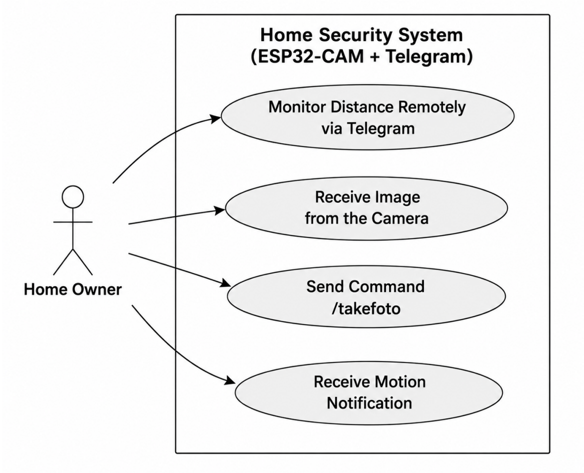

Figure 4. Usecase Diagram

3.1. Problem

The above components act as facilitators in collecting data, processing information, and transmitting the data through the internet. The choice of hardware is guided by functionality, economic viability, and compatibility of the devices. An explanation of the types of components used in this project is shown in Table 1.

Table 1. Tool Components

Component	Description
ESP32-CAM	A microcontroller board equipped with a camera module and Wi-Fi connectivity support.
ESP32-CAM MB (Module Base)	Understand formal practices
Camera OV2640	This camera is a built-in component of the ESP32-CAM module. It's used to capture images when motion is detected or when a user commands it via Telegram.
Sensor PIR (Passive Infrared Sensor)	This sensor is used to detect the movement of surrounding objects. When motion is detected, the sensor sends a signal to the ESP32-CAM to trigger image capture.
MicroSD Card	Functions as a storage medium for image files taken by the camera.
Male to Female Jumper Cable	Used to connect between electronic components, such as connecting a PIR sensor to the input pin on the ESP32-CAM.
USB Data Cable (Micro USB)	Used to connect the ESP32-CAM MB to a computer or external power source. This cable is also used to upload programs from the Arduino IDE to the microcontroller.

With the combination of hardware in Table 1, the developed Internet of Things (IoT) system is able to operate independently in detecting movement, taking images, storing data results, and sending real-time notifications to users via an integrated communication platform.

Developing a camera-based Internet of Things (IoT) system capable of detecting movement and transmitting images via the Telegram platform requires a number of supporting software components to ensure all system functions operate as intended. This software includes a development environment, supporting libraries, and board configurations used to support programming and hardware integration with digital communication services

3.2. Research Implementation

The software used in this research is as follows:

1. **Arduino**

   The Arduino IDE is used as the primary development environment for writing, editing, and uploading program code to the ESP32-CAM module.

2. **Board Manager: AI Thinker ESP32-CAM**

   The AI Thinker ESP32-CAM microcontroller board used in the programming process requires special configuration in the Board Manager in the Arduino IDE so that the system can recognize and support the board. This board selection ensures pin configuration compatibility, camera module support, and compatibility with the hardware features used in the system.

3. **Library**

   A number of libraries that provide functions outside of the project itself have been used to enhance the program's capabilities on the designed system. Such libraries are used for providing communication capabili- ties in the network, camera module control, and integration with the Telegram messaging platform. Libraries employed in this work are provided in Table 2.

4. **Connections and Communication Protocols**

   The created application relies on the use of Wi-Fi as the communication medium that connects the device to the Internet as well as to Telegram Bot API. Communication is carried out through the secure HTTPS protocol. As far as the creation of TLS connections is concerned, in the implementation of this system, the WiFiClientSecure class is employed. The connection procedure can be facilitated by not verifying the SSL certificate via the use of the setInsecure() method of the client.

Table 2. External Library

Library	Description
WiFi.h	Manages the wireless network connection between the ESP32-CAM and a Wi-Fi router.
WiFiClientSecure.h	Provides HTTPS connection support for secure communication with Telegram servers.
UniversalTelegramBot.h	Used to connect the system to Telegram bots, send messages, and receive commands.
FS.h and SD_MMC.h	Manages the file system and communication between the ESP32-CAM and a MicroSD memory card via the SD_MMC interface.
esp_camera.h	A dedicated library required for initializing and configuring the camera on the ESP32-CAM.
ArduinoJson.h	Used to process JSON data, which is essential for processing responses from the Telegram API.

5. **Main features of the ESP32-CAM module**

   The key characteristics of the developed system are motion sensing with a PIR sensor, automatic picture capturing or triggered by the user through Telegram, storage of captured images on MicroSD cards, and the automatic removal of older images when initializing the system. Furthermore, the system is capable of sending notifications and images via Telegram Bot and responding to /takefoto commands in real time. Proper software support and configuration enable the IoT system to operate autonomously, efficiently, and stably through the integration of sensors, cameras, storage media, and a digital communication platform.

The pin configuration on the ESP32-CAM is designed to ensure optimal communication between components without pin usage conflicts. Based on hardware and software analysis, the system's pin configuration is divided into two main sections: the pin configuration for the camera module and the pin configuration for the Passive Infrared (PIR) motion sensor.

1. **ESP32-CAM Camera Pin Configuration**
   All pins in table 3 are part of the internal connection between the ESP32-CAM module and the camera, and have been assigned according to the default configuration on the AI Thinker ESP32-CAM module.

Table 3. Camera Pin configuration

Pin Kamera ESP32-CAM
pin_d0 → GPIO 5
pin_d1 → GPIO 18
pin_d2 → GPIO 19
pin_d3 → GPIO 21
pin_d4 → GPIO 36
pin_d5 → GPIO 39
pin_d6 → GPIO 34
pin_d7 → GPIO 35
pin_xclk → GPIO 0 pin_pclk → GPIO 22
pin_vsync → GPIO 25
pin_href → GPIO 23
pin_sccb_sda → GPIO 26
pin_sccb_scl → GPIO 27
pin_reset → tidak digunakan (diset ke -1)

2. **PIR Sensor Pin Configuration**

   After analyzing the results obtained from both the hardware testing and code implementation, the pin configuration of the PIR sensor was carefully determined to ensure stable and accurate motion detection. The sensor's power supply is connected to the 5V pin of the ESP32-CAM module to provide sufficient operating voltage, while the OUT pin is connected to GPIO 12, which functions as a digital input to read signals generated by the sensor. The GND pin is connected to the ground of the ESP32-CAM to complete the electrical circuit and maintain signal stability. Through this configuration, the PIR sensor is capable of producing a high-level logic output whenever motion is detected within its sensing range, allowing the microcontroller to identify changes in the surrounding environment quickly and reliably. This setup is essential in minimizing false readings while ensuring that motion detection remains responsive under varying environmental conditions.

   The overall system implementation is divided into two main aspects, namely hardware and software, which are designed to work in an integrated manner to achieve optimal performance. The hardware aspect focuses on the proper arrangement and connectivity of components such as the ESP32-CAM module, PIR sensor, and supporting devices, while the software aspect manages data processing, decision-making, and communication processes. When motion is detected, the system automatically triggers the ESP32-CAM to capture an image and subsequently sends a notification via the Telegram platform, providing users with real-time updates along with visual evidence. This integration not only enhances the effectiveness of the motion detection system but also ensures that users can monitor their environment remotely in a more efficient, responsive, and practical way.

3. **Hardware Implementation**

   The hardware used consists of an ESP32-CAM module, a PIR sensor, a MicroSD, and several supporting components such as jumper cables and a USB-to-Serial module (ESP32-CAM MB). Implementation of tools into miniatures in the Figure 5.

Figure 5. Implementation tools into miniatures

4. **Implementation of system programs (software)**

Table 4. Software Installation Process

Process	Description
Network Initialization	The system connects the ESP32-CAM to a WiFi network to communicate with the Telegram API.
Camera and SD Card Configuration	The camera is configured to capture images in JPEG format, while the SD card is used to store the captured images before sending them.
PIR Sensor Integration	When the sensor detects motion, the system triggers a photo capture and sends it via a Telegram bot.
Telegram Bot	The system uses UniversalTelegramBot to receive user commands and send captured photos automatically or manually via commands such as /takefoto.

System programming was carried out using the Arduino IDE as the primary development platform, with the Board Manager configured to AI Thinker ESP32-CAM to ensure compatibility between the hardware and software components. The programming process utilized the C++ programming language because of its efficiency and suitability for embedded system development. The developed program consists of several interconnected main sections, including network initialization, camera and MicroSD card configuration, PIR sensor integration, and Telegram Bot communication, as presented in Table 4. Each section was designed to support the overall functionality of the system, enabling stable wireless communication, automatic motion detection, image capturing, temporary image storage, and real-time notification delivery through the Telegram platform. In addition, the integration between the ESP32-CAM and the PIR sensor allows the system to respond quickly whenever suspicious movement is detected, thereby improving system responsiveness and monitoring effectiveness. After the programming and integration stages were completed, comprehensive testing was conducted to evaluate the functionality, reliability, and responsiveness of the developed tool under several operational scenarios, such as powering on the device, detecting motion, and executing command-based image capture through Telegram. The testing results demonstrated that the developed IoT-based home security system was able to perform all intended operational functions effectively, including motion detection, image capture, local image storage, and real-time notification delivery through the Telegram platform. Based on repeated testing scenarios, the PIR sensor showed stable responsiveness in detecting human movement within the configured sensing range, while the ESP32-CAM module successfully captured and transmitted images with consistent quality under normal lighting conditions.

Furthermore, the system exhibited satisfactory communication performance in terms of response time and network reliability. The average delay between motion detection and notification delivery was approximately 2–4 seconds depending on Wi-Fi network stability and image transmission conditions. The integration between the ESP32-CAM and Telegram Bot API also demonstrated reliable real-time communication performance with minimal transmission failure during testing sessions. In addition, the use of MicroSD storage improved data handling efficiency by enabling temporary local storage before image transmission.

Although the system achieved stable overall performance, several operational limitations were identified during testing, including occasional notification delays under unstable network conditions and reduced motion detection sensitivity in low-temperature variation environments. Nevertheless, the overall testing results shown in Table 5 indicate that the proposed system provides reliable responsiveness, efficient real-time monitoring capability, and practical implementation potential for low-cost IoT-based residential security applications.

Table 5. Tool responsiveness results

Testing	Expected Results	Actual Results	Status
Powering On	Device Notifies that the device is ready for use.	IoT telebot, Class 11, sends a message that the device is ready for use	Pass
Moving in front of the PIR sensor	Notifies that suspicious movement is detected.	IoT telebot, Class 11, sends a message that the PIR sensor has detected movement.	Pass
Sending a command message to IoT telebot, Class 11	Taking a picture and saving it to a micro SD card, then sending the resulting photo to the ESP32-CAM.	The device successfully takes a picture and saves it to a micro SD card, then IoT telebot, Class 11 sends the image stored on the micro SD card. PASS	Pass

4. CONCLUSION

Based on the results of the design, implementation, and testing that have been carried out, an Internet of Things (IoT)-based home security system that utilizes ESP32-CAM, Passive Infrared (PIR) sensors, and Telegram Bot API has been successfully developed and is able to function according to the research objectives. The system is able to detect motion, notify users through Telegram in real time, and give visual proof in the form of pictures according to users' orders, which offers a more flexible and efficient way of controlling things in home surveillance systems. Although the proposed system successfully demonstrated real-time monitoring and notification capabilities, several limitations remain. The system performance is highly dependent on Wi-Fi network stability, which may affect notification delivery speed and image transmission reliability. In addition, the PIR sensor may produce false positive detections under certain environmental conditions, such as sudden temperature changes or unintended movement sources. Security aspects related to Telegram Bot API communication and HTTPS implementation also require further enhancement to minimize potential cybersecurity vulnerabilities. Future research may focus on improving network resilience, implementing artificial intelligence-based object recognition, strengthening communication security, and integrating cloud-based storage and video streaming capabilities.

5. DECLARATIONS

5.1. About Authors

Qori Halimatul Hidayah (QH) https://orcid.org/0009-0008-7567-4102

Ryan Putra Laksana (RP) https://orcid.org/0009-0008-5904-1321

Ary Prabowo (AP) https://orcid.org/0009-0004-5215-0497

Dwi Santika Simatupang (DS) https://orcid.org/0000-0002-4571-2939

5.2. Author Contributions

Validation was conducted by: QH Conceptualization was completed by: RP The methodology was developed by: DS Formal analysis was performed by: AP Writing, review, and editing were carried out by: RP Visualization was completed by: QH All authors, including: QH, RP, AP, and DS, have reviewed and approved the final version of the manuscript.

5.3. Data Availability Statement

The corresponding author may provide the data from this study upon request.

5.4. Funding

The research, writing, and/or publishing of this work were all done without financial assistance from the authors.

5.5. Declaration of Competing Interest

The authors state that none of their known conflicting financial interests or personal connections could have had an impact on the work that was published in this publication.

REFERENCES

[1] G. Jibin, H. Shihabudeen, S. Vignesh, A. Akhil, V. Deepak, and P. Juliemol, "Smartguard: An iot-based home surveillance system using esp32-cam and telegram," in *2025 8th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, 2025, pp. 458–463.

[2] J. Nannung, A. S. Miru *et al.*, "Development of an iot-based home security system prototype using multisensors and esp32-cam," *Journal of Embedded Systems, Security and Intelligent Systems*, pp. 168–177, 2025.

[3] R. R. PBV, V. S. Mandapati, S. L. Pilli, P. L. Manojna, T. H. Chandana, and V. Hemalatha, "Home security with iot and esp32 cam-ai thinker module," in *2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC-ROBINS)*. IEEE, 2024, pp. 710–714.

[4] M. D. T. P. Nasution, Y. Rossanty, R. Harahap, A. R. Tanjung, and T. A. M. Nasution, "Technology-driven resource utilization and integration to enhance firm performance," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 8, no. 1, pp. 268–283, 2026.

[5] K. Rambabu, K. Karthik, G. R. Sai, and K. H. Kumar, "Design and deploy an esp32-cam motion security system with telegram notifications," in *2025 6th International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, 2025, pp. 213–220.

[6] I. Yasri, A. Al-Alif, and F. Pirwanhadi, "A security home system feature as a part of home assistant through iot platform," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 17, no. 2, pp. 15–21, 2025.

[7] R. L. Avalareddy, B. P. Makala, P. Tokala, P. Maradani, M. Kumar *et al.*, "Esp32-cam-based wireless human detection system with pir sensor and instant alerts via gsm and blynk app," in *2025 5th International Conference on Trends in Material Science and Inventive Materials (ICTMIM)*. IEEE, 2025, pp. 784–788.

[8] S. Anggoro and A. Nuche, "Leadership configurations supporting togaf-based information system architecture at jenderal achmad yani university," *International Journal of Cyber and IT Service Management (IJCITSM)*, vol. 5, no. 2, pp. 134–143, 2025.

[9] A. Prakash, "Iot based anti-theft detection using esp32-cam," *Available at SSRN 5190363*, 2025.

[10] A. R. Bagaskara and A. Susanto, "Design of an internet of things (iot) based security system using esp32 cam and passive infrared receiver (pir) motion sensor in the home environment," *International Journal of Engineering Computing Advanced Research*, vol. 2, no. 1, pp. 11–18, 2025.

[11] F. Esparza-Segundo, S.-L. Enríquez-Arcadio, J. C. Gonzalez-Islas, A. Suarez-Navarrete, and M. de Jesus Gutiérrez-Sanchez, "Esp32-cam-based smart surveillance: Real-time transmission and alerts via telegram," *Programación matemática y software*, vol. 17, no. 2, pp. 48–60, 2025.

[12] N. Salsabila, A. Siswanto, and L. Bayuaji, "Design of a smart home door security system with face detection and smart bell using esp32-cam," in *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*. IEEE, 2025, pp. 124–129.

[13] A. A. Marathe, L. Sharma, L. Sailo, M. Limdiyawala, and A. Mate, "Motionguard: Esp32 cam-powered smart flooring system for enhanced security," in *2024 International Conference on Integration of Emerging Technologies for the Digital World (ICIETDW)*. IEEE, 2024, pp. 1–7.

[14] S. Adhya and G. Sen, "Iot based home security system," in *2023 7th International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech)*. IEEE, 2023, pp. 1–4.

[15] M. Gagan, K. R. Kaushik, K. F. Fernandes, and S. Madhu, "Intruder detection using surveillance cctv camera interfaced esp32 cam module," in *Computer Science Engineering*. CRC Press, 2024, pp. 402–406.

[16] R. K. Choudhary, P. Lohe, P. Gadekar, V. Sharma, S. Pahade, S. Tayde, and R. Nagpure, "Smart home security: An iot-driven facial recognition door lock using esp32-cam," *i-Manager's Journal on IoT and Smart Automation*, vol. 3, no. 1, 2025.

[17] A. Purnama, B. A. P. Indriani, and A. Nugraha, "Edge-based person detection using mobilenetv2 on esp32-cam for home surveillance system," *Jurnal Sistem Komputer dan Informatika (JSON) Hal*, vol. 1171, p. 1182, 2026.

[18] S. Owoeye, F. Durodola, A. Oyelami, R. Oladejo, S. Obasuyi, A. Qasim, and J. Ogundairo, "Implementation of a smart home intruder detection system using a vibrometer and esp 32 cam," *ABUAD Journal of Engineering Research and Development (AJERD)*, vol. 8, no. 1, pp. 14–20, 2025.

[19] S. Chopvitayakun, M. Rattanasiriwongwut, and P. Kadmateekarun, "Iot-based smart home integration: Innovative system architecture for intelligent monitoring and control," in *International Conference on e-Society, e-Learning and e-Technologies*. Springer, 2025, pp. 299–310.

[20] Y. D. Anna, H. Djajadikerta, and A. Setiawan, "Strengthening the foundations of socialpreneurship through integrated reporting a systematic bibliometric perspective," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 8, no. 1, pp. 296–309, 2026.

[21] K. Okokpujie, I. P. Okokpujie, F. T. Young, and R. E. Subair, "Development of an affordable real-time iot-based surveillance system using esp32 and t wilio api," *Int. J. Saf. Secur. Eng*, vol. 13, no. 2023, pp. 1069–1075, 2023.

[22] S. Vivek and G. Kaur, "Iot-based surveillance robot using esp-32 wi-fi cam," in *2025 IEEE International Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI)*, vol. 3. IEEE, 2025, pp. 1–6.

[23] K. Sangeetha, M. A. Karthikeyan, H. Ahamed, and S. Bharath, "Real-time classroom student monitoring framework using esp32-cam and cnn-based face recognition," in *2026 5th International Conference on Communication, Computing and Electronics Systems (ICCCES)*. IEEE, 2026, pp. 58–63.

[24] P. K. Goyal, M. Giri, and S. Verma, "Iot-based smart door lock system with face recognition using esp32 cam and android app," in *International Conference on Electrical and Electronics Engineering*. Springer, 2023, pp. 365–376.

[25] E. T. Rusmiati, L. Febrina, Y. Sari, and E. M. S. Sakti, "Adoption of ai driven ecological preaching systems using sem pls analysis," *Aptisi Transactions on Technopreneurship (ATT)*, vol. 8, no. 1, pp. 284–295, 2026.

[26] R. Syavira, "Implementation of a smart home doorbell system for guest identification using esp32-cam and iot-based telegram chatbot integration," *Journal of Technology and Computer*, vol. 2, no. 3, pp. 171–175, 2025.

[27] N. Angelika *et al.*, "Design and construction of an automatic door system based on iris detection using esp32-cam and opencv using the rapid application development (rad) method," *Jurnal Inotera*, vol. 11, no. 1, pp. 190–202, 2026.

[28] K. Purnima, B. Madhavi, D. Sukith, K. Rajesh, P. S. K. Reddy, and B. V. K. K. Reddy, "An iot-based smart wearable safety device using gsm, gps, and esp32-cam," in *2025 3rd International Conference on Sustainable Computing and Data Communication Systems (ICSCDS)*. IEEE, 2025, pp. 1883–1887.

[29] M. S. N. I. Marzuki, A. K. Z. M. Norfaizin, M. A. B. A. Razak, L. H. B. Zulkornain, M. F. M. Zin, and M. Z. Jusoh, "Energy-efficient home protection: A solar-powered, iot-enabled security solution," in *2025 IEEE International Conference on Applied Electronics and Engineering (ICAEE)*. IEEE, 2025, pp. 1–5.

[30] S. Gayathri, A. S. Prasad, H. Niteesh, K. T. Naik, S. Hurakadli, and R. Vishal, "Low cost iot based smart monitoring system," in *2024 International Conference on Recent Advances in Science and Engineering Technology (ICRASET)*. IEEE, 2024, pp. 1–6.

[31] T. Ivan Danu, A. A. Akhmad Ahfas, J. Jamaaluddin *et al.*, "Home surveillance monitoring with esp32-

cam and sd card for data storage," *Journal of Computer Networks, Architecture and High Performance Computing*, vol. 6, no. 1, pp. 419–429, 2024.
[32] R. M. Nor, L. C. Yaw, N. Shariffudin, S. Yahya, W. Nooriman, H. Abd Hamid, S. A. Shukor, M. Zuradzman, and E. Zainon, "Enhancing security through smart and low-cost iot-based surveillance system," in *2025 9th International Conference on Man-Machine Systems (ICoMMS)*. IEEE, 2025, pp. 1–5.
[33] A. R. Lellapati, B. K. Pidugu, M. S. R. Rangari, G. P. Acharya, and V. Jayaprakasan, "Development of advanced alerting system using arduino interfaced with esp32 cam," in *2022 IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA)*. IEEE, 2022, pp. 261–266.
[34] R. D. Handayani, Z. Jamal, M. Alkahfiansyah, L. Rosmalia, N. H. Sudibyo, and R. Herwanto, "Intelligent and secure package receiversystem utilizing internet of things (iot) technology," *Vokasi Unesa Bulletin of Engineering, Technology and Applied Science*, vol. 2, no. 3, pp. 581–592, 2025.
[35] M. I. Ismail, R. Elsayed, H. A. Sakr, M. M. Fouda, and A. F. Ashour, "An integrated iot smart home system with face and object detection using ssd for real-time security and control on esp32," in *2025 3rd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*. IEEE, 2025, pp. 1–6.
[36] N. Raju, A. Navya, N. Koteswaramma, B. Mounika, and T. Rajeshwari, "Iot-based door access control system using esp32 cam," *International Journal of Engineering Inventions*, vol. 11, no. 12, pp. 9–15, 2022.
[37] M. Ohemu and M. O. Idoko, "Development of a low-cost iot-enabled door access control system for smart home security," *Asian Journal of Computer Science and Technology*, vol. 14, no. 2, pp. 43–50, 2025.
[38] R. K. Choudhary, P. Lohe, V. Sharma, P. Gadekar, S. Pahade, S. Tayde, and R. Nagpure, "Innovative smart locking mechanisms using esp32-cam: Merging facial recognition, iot, and biometrics," *IoT, and Biometrics (April 26, 2025)*, 2025.
[39] S. Bousselmi, M. Gannouni, and K. Ouni, "An iot-based real-time fire safety monitoring system using esp32 for industrial applications," in *2025 IEEE International Multi-Conference on Smart Systems & Green Process (IMC-SSGP)*. IEEE, 2025, pp. 1–8.
[40] P. U. Rakhmawati, D. M. Syahnaryanti, R. Rizdania, and S. Sumantri, "Design and development of an internet of things (iot)-based smart bell using esp32-cam and telegram," *Smart Technology, Informatics, and Technopreneurship*, vol. 8, no. 1, pp. 121–131.
[41] W.-T. Sung, Y.-H. Lai, I. G. T. Isa, and S.-J. Hsiao, "Constructing an intelligent home system control using internet of things," in *2025 International Conference on Automation Technology (Automation)*. IEEE, 2025, pp. 1–6.