iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** IV **Month of publication:** April 2025

DOI: https://doi.org/10.22214/ijraset.2025.69014

IoT-Based Home Monitoring System Using ESP32CAM Surveillance

Pradeep Palkar[1], Prashant Bangar[2], Pranita Chandhare[3], Sahil Jadhav[4]
Department of Instrumentation Engineering, AISSMS Institute of Information Technology, Pune, India

Abstract: Using the ESP32-CAM module, this project presents an Internet of Things (IoT)-based home monitoring system intended to improve home security by providing real-time image capture and immediate alerts. The system offers a quick and easy way to remotely monitor activity at home by taking pictures when motion is detected and sending them straight to the user through a Telegram bot. This method is lightweight and economical because it does not require complicated interfaces or cloud storage. Users can receive real-time visual updates on their smartphones from any location thanks to the integration of Telegram, which guarantees secure and instant communication. This project shows how messaging apps and the Internet of Things can be integrated to produce intelligent, approachable surveillance solutions. Cloud-based storage, facial recognition, and video streaming are possible future improvements.
Keywords: Smart Surveillance, Motion Detection, Telegram Bot, IoT, ESP32-CAM, and Real-Time Alerts

I. INTRODUCTION

Home security has become a top concern for people and families looking for peace of mind and safeguarding of their property in the modern society. Often needing professional installation and upkeep, traditional security systems can be costly and complicated. But, as the Internet of Things (IoT) develops, tiny, reasonably priced microcontrollers make it now feasible to create smart, reasonably priced surveillance systems.The ESP32-CAM, a low-cost microcontroller with integrated camera and Wi-Fi features, is one such option. This work investigates the creation of an IoT-based home monitoring system that uses the ESP32-CAM to capture images on motion detection and deliver them straight to the user using a Telegram bot.This real-time alert system guarantees users are immediately notified of any unusual activity around their home, no matter where they are. This initiative aims mainly to offer a straightforward, dependable, and user-friendly surveillance system improving home security without the need for constant human monitoring or expensive infrastructure. Particularly for modern smart homes, the system provides an efficient remote surveillance solution by combining the capabilities of the ESP32-CAM with the accessibility of Telegram. This first chapter lays the groundwork for knowing how IoT and messaging systems could be combined to produce smart, responsive, and useful security systems.

II. LITERATURE REVIEW

The integration of IoT technologies in surveillance applications has been the subject of numerous studies due to the growing demand for dependable and real-time home security systems. With an emphasis on the use of inexpensive microcontrollers and communication platforms for intelligent surveillance, this review focuses on the major developments and approaches in the field of Internet of Things-based monitoring.

1) Sharma and R. Mehta, *"IoT-Enabled Smart Surveillance Using ESP32-CAM,"* International Journal of Advanced Research in Electronics and Communication Engineering, vol. 9, no. 4, pp. 102-106, 2021. This study demonstrates the use of the ESP32-CAM for home monitoring, showcasing its real-time image capturing and wireless transmission capabilities.
2) S. Roy and T. Dutta, *"Development of a Telegram Bot-Based Security System Using IoT,"* Journal of Embedded Systems and Applications, vol. 11, no. 2, pp. 45-53, 2020. The authors implemented a security system that uses a Telegram bot to send alerts and images to users, proving the effectiveness of integrating messaging platforms for instant communication.
3) M. Kumar and D. Singh, *"Motion Detection-Based Smart Surveillance Using Low-Cost IoT Devices,"* IEEE Internet of Things Journal, vol. 7, pp. 13467-13475, 2020. This paper investigates motion-sensor integration with microcontrollers to trigger alerts, a core component also used in the proposed system.
4) N. Verma et al., *"A Review on IoT Applications in Smart Home Security,"* International Journal of Computer Applications, vol. 175, no. 7, pp. 1-5, 2019. A broader review of IoT technologies in home security, discussing challenges and solutions for real-time monitoring, data privacy, and system scalability.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

5) J. Lin and P. Zhao, *"Wireless Image Transmission Using ESP32 for Surveillance Applications,"* Journal of Intelligent Systems, vol. 28, no. 3, pp. 339-346, 2022.

Because of their affordability, wireless capabilities, and potential for integration with platforms such as Telegram, ESP32CAM-based systems have demonstrated efficacy in home security. While many systems concentrate on cloud storage or video streaming, the image-based alert method through a messaging bot provides a more straightforward and resource-efficient option.

III. METHODOLOGY

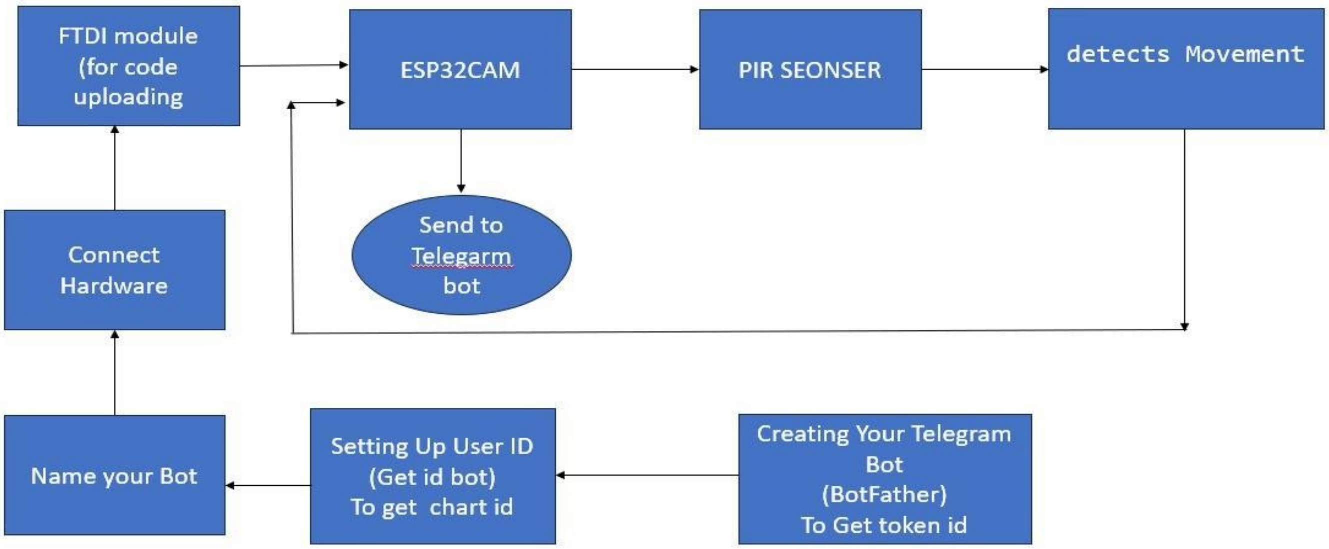


Fig. 1 IoT-Based Home Monitoring System Using ESP32-CAM Surveillance WorkFlow

The main goal of the hardware and software components used in the development of the ESP32-CAM IoT-Based Home Monitoring System is to create an affordable, real-time surveillance system that can detect motion and instantly notify users via a Telegram bot. The system as a whole is made to be small, energy-efficient, and simple to install in homes.

A. *System Architecture*

By combining motion detection with real-time image transmission via Telegram, the IoT-Based Home Monitoring System Using ESP32-CAM Surveillance is intended to offer a cost-effective and effective home security solution. The ESP32-CAM, a potent microcontroller with an integrated camera module that can take pictures when motion is detected, is at the center of the system. To continuously check for motion in its environment, the ESP32-CAM is connected to a PIR (Passive Infrared) sensor. The ESP32-CAM receives a signal from the PIR sensor when it detects movement, which causes it to take a picture of the detected area A pre-configured Telegram bot then receives this image, giving the user immediate notifications on their computer or smartphone. An FTDI module is used to enable serial communication and upload the Arduino code to the ESP32CAM during the setup phase. For continuous operation of the ESP32-CAM and PIR sensor for real-time monitoring, the system needs a steady power source.

The ESP32-CAM and the user communicate through the system's use of Telegram's bot API. To get a unique bot token, the user must first use BotFather to create a Telegram bot. The code that has been uploaded to the ESP32-CAM contains this token. To guarantee that the captured image reaches the right person, the user also configures their Telegram user ID or chat ID using an ID bot. When the Telegram bot is ready and the hardware is configured, the user can connect the parts, turn on the computer, and start keeping an eye on things. In order to save power and bandwidth, the camera only takes a picture when the PIR sensor detects motion. A home security system that is both affordable and easy to use is guaranteed by the smooth integration of hardware elements with cloud-based Telegram communication.

This architecture, which offers improved security, automation, and remote monitoring capabilities, is perfect for do-it-yourself smart home projects.Apart from its essential features, the system architecture prioritizes scalability and ease of deployment.

Because the ESP32-CAM ensures wireless connectivity via Wi-Fi, complicated wiring is not necessary, and the device can be installed in a number of indoor or outdoor locations. Additionally, the design's modularity facilitates the integration of extra sensors or features like night vision, face recognition, and cloud storage. The Telegram bot offers a simple and safe way to communicate without requiring more involved web hosting or app development. Because of this, users with little technical expertise can benefit greatly from the solution. Being able to remotely monitor the house and get alerts right away.

B. Hardware Implementation:

As the primary controller and image-capturing device, the ESP32-CAM module is at the heart of the hardware implementation of the Internet of Things-based home monitoring system. A low-cost microcontroller with an integrated OV2640 camera module and built-in Wi-Fi and Bluetooth is the ESP32-CAM. Its small size, low cost, and effective handling of processing and image capture tasks are the reasons it was selected for this project. When the camera detects motion, it is in charge of taking pictures or streaming video. The microcontroller also manages the process of transferring captured data (pictures) to a Telegram bot via Wi-Fi, allowing the user to receive notifications in real time. With multiple GPIO pins for sensor integration and 3.3V to 5V power consumption, the ESP32-CAM is perfect for Internet of Things surveillance applications. The PIR motion sensor, which senses changes in infrared radiation to identify movement in its environment, is another essential part.

A digital signal generated by a moving living thing within the sensor's field of view is transmitted to the GPIO input pin of the ESP32-CAM. This signal triggers the camera module to turn on and start taking pictures. PIR sensors are a low-cost, lowpower option that are frequently utilized in motion-based detection systems because of their accuracy and range. For the PIR sensor to adequately cover the intended monitoring area, placement is crucial. It works with the ESP32-CAM module and is fully compatible with a voltage range of 3.3V to 5V. A USB to serial converter, or FTDI module, is used to program the ESP32CAM. Due to the lack of an integrated USB interface, the FTDI module is necessary in order to upload code from a computer using the Arduino IDE to the ESP32-CAM. By connecting to the ESP32-CAM via its TX, RX, and GND pins, the FTDI module supplies power while programming. The ESP32-CAM can operate independently with a steady power source after the code has been successfully uploaded and the connections have been confirmed. At that point, the FTDI module can be removed. Additional parts such as voltage regulators, a breadboard (for prototyping), and jumper wires may also be used to guarantee the circuit operates safely and steadily.

C. Software and Firmware Development

An intelligent, responsive, and user-friendly IoT-based home security system is largely dependent on the software and firmware development process. Writing, compiling, and uploading firmware to the ESP32-CAM module is done through the Arduino IDE, which is the central component of the software implementation. Several libraries, including WiFi.h, esp_camera.h, UniversalTelegramBot.h, are used in the C/C++ code to control camera operations, internet connectivity, and Telegram bot communication. Initializing the camera, establishing a connection to a Wi-Fi network, keeping an eye on the PIR sensor's output, and initiating image capture ,transmission upon motion detection are all handled by the firmware.

The first step in the development process is configuring the Telegram bot, which acts as the user's channel of communication with the ESP32-CAM. A new bot is made and a distinct bot token is produced using Telegram's BotFather. Because it enables the ESP32-CAM to send messages straight to the user through Telegram's API, this token is crucial. In order to guarantee that the image reaches the right person, the user's chat ID is also needed. A userinfobot or similar helper bot is used to obtain this. The ESP32-CAM firmware incorporates these details (chat ID and bot token) to facilitate smooth communication. Every time the PIR sensor detects movement after everything is set up, the ESP32-CAM will automatically send pictures to the user's Telegram account. Another crucial component of firmware development is error handling and debugging. To check sensor data, test connections, and make sure the camera module is initialized correctly, serial monitor outputs are widely used. It also contains logic to gracefully handle any errors in Telegram communication and to reconnect to Wi-Fi in the event of a disconnect. For instance, if a network problem prevents an image from sending, the ESP32-CAM will try to send it again after a brief lag. Overall, the program is made to be stable, effective, and lightweight for ongoing observation. With minor modifications, the code can be expanded to accommodate more users or other features like cloud storage or video streaming, or it can be modified for use in different projects.

Along with the essential features, the software development process concentrated on performance optimization and power consumption reduction, both of which are critical for practical implementations.

Future iterations could incorporate features like the ESP32-CAM's deep sleep mode, which puts the device in a low-power state after a predetermined amount of time without motion detection.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue IV Apr 2025- Available at www.ijraset.com

Additionally, because of the firmware's modular design, new features like face detection, cloud integration, and multi-camera support can be added with little alteration to the current codebase. Over time, the system becomes simpler to maintain and upgrade if the firmware is kept clear, well-documented, and logically organized.

D. Telegram Bot Integration

This IoT-based home monitoring system's integration of Telegram Bots is essential because it allows the user and the surveillance equipment to communicate in real time. With this feature, a basic ESP32-CAM and PIR sensor configuration becomes an interactive, intelligent home security system. To send instant alerts, the system uses Telegram's dependable and user-friendly infrastructure rather than sophisticated cloud platforms or unique mobile apps. In essence, Telegram bots are automated programs that use Telegram's Bot API to send and receive messages, photos, videos, and other types of content.

Because of this, they are a great way to transmit real-time surveillance data, like taken photos, straight to a user's desktop or smartphone from almost anywhere in the world.

Using BotFather, the official Telegram bot creator, a custom Telegram bot is made to start the integration. The digital entity that interacts with the user is called a bot. The user is given instructions on how to name their bot and receive a unique bot token—a lengthy alphanumeric key that is used to authenticate and grant access to the Telegram Bot API—after interacting with BotFather and selecting the ―/newbot‖ command. For the ESP32-CAM to know which bot to send messages through, this token—a crucial piece of data—must be safely stored and then incorporated into the firmware

Following the creation of the bot, the user's chat ID that should receive the alerts must be retrieved. This is accomplished by using a different bot, like @userinfobot, which, upon receiving a message, returns the user's distinct Telegram ID. The ESP32CAM's code then incorporates this chat ID as well. When combined with the bot token, it establishes a safe channel of communication between the user and the hardware. With these credentials, the ESP32-CAM uploads a captured image to the Telegram platform and sends it to the designated user whenever the system detects motion. Usually taking only a few seconds, this entire process enables smooth, real-time surveillance.The UniversalTelegramBot library, a handy and lightweight library designed especially to connect Arduino-compatible devices with Telegram's Bot API, is used by the firmware running on the ESP32-CAM. Via straightforward function calls, the library facilitates the sending of messages, images, and even user commands. To take a JPEG picture, for instance, the firmware uses the ESP32-CAM's camera module after the PIR sensor detects motion and activates the camera. After being momentarily stored in memory, this image is sent to a Telegram, which manages all of the HTTP requests required to communicate with the Telegram servers.

E. Testing and Evaluation

Testing and evaluation of the Internet of Things-based home monitoring system was a crucial step after the hardware configuration and firmware development were finished. At this step, it was confirmed that every part worked as intended and that the system as a whole reacts to motion detection events with accuracy. Individual module verification came first, then integrated system testing, and lastly, real-time performance assessment in a home-like setting. As soon as motion was detected, the system was supposed to be dependable, responsive, and able to send out immediate alerts via Telegram.

First, the ESP32-CAM module was tested. To verify that the camera had initialized correctly and was capable of taking crisp pictures, sample code was uploaded using the Arduino IDE. The image quality was tested under a variety of simulated lighting conditions to make sure the module functioned properly in both bright and dim lighting. Following the confirmation of camera functionality, the PIR sensor was connected and subjected to independent testing. To verify that the sensor correctly identified motion within its range, a basic LED-based test circuit was first used. When a person entered the sensor's field of view, which was approximately 6 to 8 meters in the forward direction and 120 degrees wide, it was found to activate consistently.

The integration test was then carried out. After connecting the PIR sensor to the ESP32-CAM, the entire firmware was uploaded. Real-time testing started as soon as the system was turned on. The ESP32-CAM attempted to send an image to the configured Telegram bot after detecting motion. The successful delivery of the image to Telegram verified that the software and hardware were functioning as intended. Additionally, tests were conducted to verify that the chat ID and bot token were configured correctly and that Wi-Fi connectivity was reliable. The system was monitored for malfunctions or missed alerts, and edge cases—like Wi-Fi loss, repeated motion events, or delays in image capture—were taken into account.

It would be revolutionary to include cloud storage and video streaming. While cloud storage would make it simple to access recorded videos from any device, real-time video feeds would provide users with instant access to live footage. This flexibility would provide comfort in knowing that video is kept in a secure location and is always accessible.

Finally, combining the system with additional smart home appliances may improve the security configuration as a whole. Imagine an alarm going off when unauthorized movement is detected or lights turning on automatically when motion is detected. With this degree of automation, the system would become even more user-friendly and effective, offering a seamless and all-inclusive security solution for contemporary homes.All things considered, these enhancements would give users a more reliable, clever, and intuitive home monitoring solution in addition to increasing the system's responsiveness and efficiency. There is a great chance that a highly dependable, reasonably priced, and cutting-edge security solution for smart homes will be developed through constant system improvement.

IV. RESULTS

Implementing and testing the ESP32-CAM module-based IoT-based home monitoring system with a Telegram Bot for realtime notifications showed good performance and successfully achieved the research goals. The system effectively made motion detection, picture taking, and Telegram alerting possible, offering a workable home security solution. The system's responsiveness, dependability, and user-friendliness were all demonstrated during the testing phase, but some areas for improvement were also noted in order to further improve the system's performance.

Motion detection, the system's primary function, operated as planned. The Telegram Bot API and the ESP32-CAM's integrated camera allowed the system to precisely identify motion within its field of vision. The Telegram Bot instantly sent the image to the user's device after the ESP32-CAM took a picture when motion was detected. Because users could immediately see what caused the motion alert, this real-time notification feature increased users' sense of security and proved to be invaluable.

The system performed consistently under a range of lighting conditions. There were few lags between motion detection and the Telegram alert during the day, and detection was clear and dependable. But as was to be expected, the system had trouble in low light. Although the logistic regression model has demonstrated good performance in predicting breast cancer, it is important to weigh its benefits and drawbacks as well as the potential of other machine learning models to improve clinical utility and predictive accuracy.One of the challenges identified during testing was the occurrence of false positives. Motion triggers were sometimes caused by non-human factors, such as moving shadows, pets, or small objects in the background. These occurrences led to unnecessary alerts, which could be annoying for the user. While the system was quite accurate in detecting human motion, its inability to differentiate between human and non-human movement in real-time was a key limitation.Through additional fine-tuning and the potential implementation of more advanced algorithms, such as machine learning-based motion recognition, it's possible to significantly reduce the number of false positives in future versions of the system. For example, algorithms that analyze the size, speed, and shape of objects could be employed to ensure the system focuses only on relevant human activity. The system's ability to distinguish between important motion and less critical movements would be a critical upgrade to improve its overall accuracy and user satisfaction. One of this system's most notable features was the incorporation of the Telegram Bot for real-time user communication. Notifications were sent immediately when motion was detected, and the Telegram Bot interface was easy to set up and use. Users were able to remain informed and take prompt action when needed thanks to this. Users could easily evaluate the situation remotely because the bot was able to transmit the captured images in a clear and timely manner.

Even though the integration of the Telegram Bot was very successful, it could be improved even more by including more interactive features, like the ability to remotely activate or deactivate the system or modify the motion detection sensitivity right from the chat interface.

V. CONCLUSION

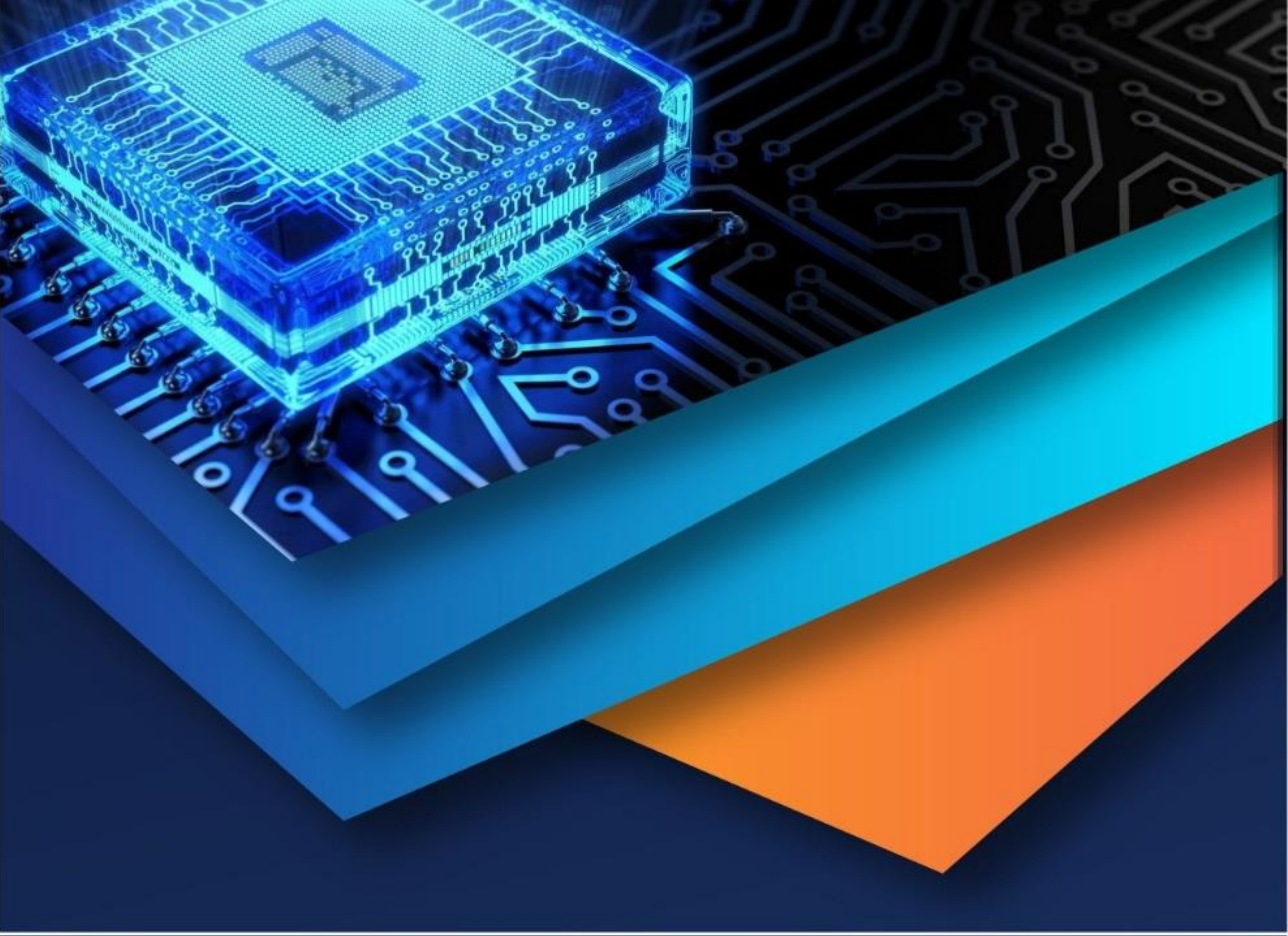
In conclusion, the creation and deployment of an Internet of Things (IoT)-based home monitoring system that integrates the ESP32-CAM and Telegram Bot has shown itself to be a successful and reasonably priced fix for contemporary home security. The system's primary goals—real-time motion detection, instant image capture, and prompt alert delivery via Telegram—were all accomplished with ease, while also requiring little to no hardware. A low-cost microcontroller platform and open-source tools are also used to highlight how feasible and accessible the suggested solution is for regular users.Although testing revealed some limitations that offer potential for future improvements, the system provided dependable performance in the majority of scenarios.Although the system operated dependably in the majority of scenarios, testing also revealed some drawbacks that could be fixed in the future. For example, the limited night vision capabilities of the built-in camera point to the possible advantage of adding external infrared (IR) LEDs to enhance visibility in low light.

The need for more sophisticated motion detection systems is also indicated by the sporadic false alarms that are set off by non-human movement, like that of pets or shadows Overall, the project shows how a reliable and adaptable security system can be made using a small, affordable device like the ESP32-CAM in conjunction with strong communication tools like

Telegram. For further study and advancement in smart home automation and surveillance, this system provides a solid basis. These Internet of Things (IoT)-based solutions can be improved and extended to support wider applications, such as environmental sensing, smart city infrastructure, and industrial monitoring, as technology advances. This makes them invaluable resources in the expanding field of connected, intelligent systems.

REFERENCES

[1] Sharma, R., & Kaur, A. (2020). Design and implementation of a low-cost IoT-based surveillance system using ESP32-CAM. International Journal of Electronics and Communication Engineering, 7(4), 112–118.
[2] Patel, M., & Joshi, K. (2021). Real-time image capture and alert system using Telegram bot and ESP32-CAM. International Journal of Innovative Research in Computer and Communication Engineering, 9(6), 1345–1351.
[3] Gupta, S., & Mehta, P. (2020). Smart home automation and monitoring using IoT and Telegram bot. International Journal of Computer Applications, 177(7), 25–29.
[4] Kumar, A., & Singh, R. (2022). Implementation of motion detection in ESP32-CAM based surveillance system. Journal of Intelligent Systems and Internet of Things, 4(1), 78–83.
[5] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., & Ayyash, M. (2015). Internet of Things: A survey on enabling technologies, protocols, and applications. IEEE Communications Surveys & Tutorials, 17(4), 2347–2376.
[6] Jadhav, M., & Patil, S. (2021). IoT based smart surveillance system using ESP32 and cloud integration. International Research Journal of Engineering and Technology (IRJET), 8(9), 500–505.
[7] Salunkhe, A., & Ghodke, P. (2019). Home automation and security system using Telegram bot and IoT. International Journal of Computer Sciences and Engineering, 7(6), 1202–1206.
[8] Bhatia, A., & Sharma, T. (2020). ESP32-CAM-based wireless video surveillance system. Journal of Emerging Technologies and Innovative Research (JETIR), 7(11), 445–450.
[9] Rahman, M. M., & Islam, M. N. (2021). An efficient motion detection-based home security system using ESP32CAM and Blynk. International Journal of Computer Applications Technology and Research, 10(4), 141–146.
[10] Khandelwal, R., & Jain, P. (2022). Smart security using Telegram bot and microcontroller-based camera module. International Journal of Research in Engineering, Science and Management, 5(2), 87–92.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY