International Journal of Creative and Open Research in Engineering and Management
ISSN: 3108-1754 (Online)
Volume 02 Issue 11 November-2026 | Impact Factor: 3.5



An Event-Driven Edge AI Surveillance System with Real-Time Automated Alerting

Dr. Kishore M
Electronics & Communication Dept.
KSSEM, VTU
Bengaluru, India
kishore@kssem.edu.in

Akash S
Electronics & Communication Dept.
KSSEM, VTU
Bengaluru, India
akash.somashekhara@gmail.com

Shrilakshmi C Badiger
Electronics & Communication Dept
KSSEM, VTU
Bengaluru, India
shreelakshmicbadiger@gmail.com

Vamshi Krishna C
Electronics & Communication Dept
KSSEM, VTU
Bengaluru, India
vamshikrishnac978@gmail.com

Sathwik M
Electronics & Communication Dept.
KSSEM, VTU
Bengaluru, India
sathwikm01@gmail.com

How to Cite this Article:
S, A., Badiger, S. C., C, V. K. & M, S. (2026). An Event-Driven Edge AI Surveillance System with Real-Time Automated Alerting. International Journal of Creative and Open Research in Engineering and Management, 02(11), 1-9. https://doi.org/10.55041/ijcope.v2i11.125

License:
This article is published under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.
© The Author(s). Published by International Journal of Creative and Open Research in Engineering and Management.







https://doi.org/10.55041/ijcope.v2i11.125

***Abstract*— Traditional surveillance networks rely on continuous video recording, which demands immense data storage, consumes high network bandwidth, and serves primarily as a passive forensic tool rather than an active deterrent. While basic motion-activated systems mitigate storage waste, they suffer from critically high false-alarm rates triggered by environmental noise. To resolve this, we propose an event-driven Edge AI surveil-lance architecture using a Raspberry Pi 3 Model B+ paired with an OV5647 NoIR camera. The system utiliz-es a time-domain decoupled approach: an NS312 PIR sensor acts as a strict zero-power hardware interrupt, keeping the processing pipeline dormant until physical motion is detected. Upon activation, an onboard OpenCV Haar Cascade classifier executes instantaneous face verification to definitively confirm human presence, completely eliminating false positives. The Python-based firmware employs multithreaded concurrency to simul-taneously trigger a localized 5V active piezo buzzer for on-site deterrence, dispatch annotated visual alerts via the Telegram Bot API, and maintain a thread-safe Flask MJPEG web stream. Physical validation on Debian 13 "Trixie" demonstrates stable zero-light operation, miti-gation of I2C brownouts, and sub-3-second alert latency without reliance on cloud-based AI.**

Keywords- *Edge AI, Event-Driven Architecture, OpenCV Haar Cascade, Telegram Bot API, Hardware Interrupt, Surveillance, False Alarm Mitigation..*

I. INTRODUCTION

The rapid expansion of urban infrastructure and decentral-ized security demands has pushed the adoption of Internet Protocol (IP) camera networks. However, these convention-al systems operate on a continuous recording paradigm. This constant streaming captures hours of static, empty frames, exhausting local storage mediums, straining network band-width, and wasting electrical power. Furthermore, passive recording fails to interrupt crimes in progress, acting merely as an investigative tool after the event has concluded.

To reduce data overhead, basic motion-activated systems utilizing Passive Infrared (PIR) sensors are commonly de-ployed. However, they suffer from a major operational flaw: a lack of spatial intelligence. Thermal shifts from HVAC systems, moving shadows, or pets frequently trigger false alarms. This leads to "alert fatigue," where end-users eventually ignore the security system entirely. To filter out these false positives, modern commercial setups often offload video streams to centralized cloud servers for deep-learning analysis. This cloud dependency introduces unacceptable transmission latency, compromises data privacy, and renders the system useless during internet outages.

Edge Artificial Intelligence (Edge AI) offers a solid structural fix. By migrating computational power directly to the camera node, the system can extract meaningful insights locally without relying on the cloud. Despite these benefits, running AI models continuously on resource-constrained edge devices causes thermal throttling and power drain

Furthermore, deploying surveillance nodes in remote, off-grid, or battery-constrained environments imposes stringent energy budgets that strictly preclude continuous camera polling. Standard microprocessors executing visual inference loops continuously deplete typical battery reserves within a matter of hours, necessitating frequent maintenance or oversized energy storage solutions. Consequently, an architectural paradigm shift is required—moving away from always-on software polling toward a hybrid state where the primary computational framework remains completely dormant, drawing negligible quiescent current until a physical intrusion is highly probable.

To bridge this gap, this paper introduces an event-driven Edge AI architecture. Our goal is to fuse zero-power hardware interrupts with lightweight, on-device computer vision to optimize both power efficiency and detection accuracy.

The key contributions of this work include:

- A decoupled hardware-software event loop that utilizes an NS312 PIR sensor to maintain the camera in a zero-power idle state until genuine physical motion is detected.
- The implementation of an OpenCV Haar Cascade frontal-face classifier optimized for the ARM Cortex-A53, ensuring localized human verification without cloud latency.
- A parallel multithreaded alert dispatch system that simultaneously triggers a physical 5V buzzer and pushes API-driven Telegram alerts without blocking the video capture stream.



- Extensive diagnostic validation addressing modern V4L2 to libcamera driver transitions and I2C hardware brownout mitigation on the Raspberry Pi 3B+.

II. LITERATURE SURVEY

To contextualize our event-driven design, previous research in power optimization and computer vision for embedded security must be analyzed. Past research generally splits into three areas:

A. *Hardware-Level Motion Optimization*

Most legacy research focuses strictly on hardware interrupts. Kumar et al. demonstrated that integrating basic PIR sensors to wake a microcontroller saves up to 70% of power compared to continuous polling. However, their setup only tracked raw thermal deltas, completely ignoring visual verification. Consequently, the system suffered from high false-positive rates due to environmental thermal shifts, establishing the need for a secondary software verification layer.

B. Deep Learning and Edge Analytics

Beyond raw hardware triggers, significant work looks at spatial analytics. Chen and Liu investigated the migration of convolutional neural networks (CNNs) like YOLO from cloud servers to edge devices. While highly accurate, they proved that heavy CNNs induce significant thermal throttling and multi-second frame latency on mid-tier edge devices like the Raspberry Pi. This justifies our selection of the computationally lightweight Haar Cascade classifier, which provides adequate frontal-face detection at a fraction of the computational cost, ensuring real-time performance.

C. IoT Alerting and Network Latency

Recent studies highlight the latency and infrastructural overhead of remote IoT alerting. Rahman et al. proved that traditional SMTP and MQTT protocols introduce significant payload overhead, often delaying multimedia delivery by 10 to 15 seconds due to heavy SSL/TLS handshakes. Furthermore, proprietary mobile applications require complex server maintenance and NAT traversal. To bypass these bottlenecks, third-party messaging APIs like the Telegram Bot API have emerged as optimal solutions. By utilizing lightweight HTTP POST requests, Edge AI nodes achieve secure, sub-3-second image delivery over standard Wi-Fi, completely eliminating the need for custom backend databases.

D. Research Gap Synthesis

A critical review reveals a distinct operational gap at the intersection of power management, edge inference, and telemetry. Current surveillance topologies fall into two extremes: relying exclusively on inaccurate hardware triggers (causing severe false positives) or executing overly complex neural networks that induce thermal throttling on embedded hardware. While PIR interrupts, Haar Cascades, and API integration have been explored in isolation, no methodology successfully fuses all three into a unified, thread-safe architecture.

III. SYSTEM ARCHITECTURE AND WORKING PRINCIPLE

The fundamental operational theory of this system is derived from event-driven computational modeling. The architecture distributes the processing load across distinct hardware and software stages.

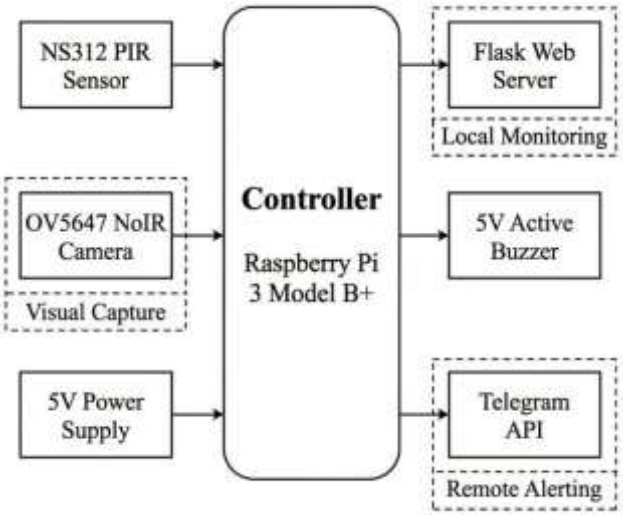


Fig.1. Hardware block diagram of the proposed event-driven Edge AI surveillance node

A. Stage 1: Hardware Wake (S1 is OFF, S2 is ON*)*

In the idle state, the OV5647 NoIR camera and the OpenCV software loop are suspended. The NS312 PIR sensor independently monitors the environment. When the sensor detects a thermal delta exceeding its threshold, it pulls BCM GPIO 17 to a logic HIGH (3.3V). This acts as the hardware interrupt, waking the secondary processing stages.

B. Stage 2: Edge Inference and Acquisition

Upon the logic HIGH trigger, the Raspberry Pi activates the MIPI CSI-2 camera interface. To prevent the AI from processing stale, buffered frames, the video capture property (CAP_PROP_BUFFERSIZE) is strictly clamped to 1. The 640x480 frame is acquired and mathematically converted to grayscale to reduce matrix depth from 3 channels (RGB) to 1, significantly accelerating the subsequent Haar Cascade calculations.

C. Stage 3: Multithreaded Dispatch

If the Haar classifier matrix returns positive bounding box coordinates (indicating human verification), the system initiates the alert protocol. To prevent SD card read/write degradation, the annotated frame is saved to a RAM-backed temporary file system (/tmp/intruder.jpg). The operating system then spawns two asynchronous daemon threads:

1. Acoustic Deterrence: Drives BCM GPIO 27 HIGH for 2 seconds to sound the active buzzer.
2. Cloud Notification: Dispatches an HTTP POST multipart/form-data request to the Telegram API.



D. Stage 4: Local Streaming and Thread Locks Concurrently, the primary capture loop updates a global output frame variable. A Flask web server (0.0.0.0:5000) continuously reads this variable, encoding it into an MJPEG boundary stream. A threading.Lock() mutex is strictly enforced to prevent memory collisions between the OpenCV writer thread and the Flask reader thread.

2. System Flowchart

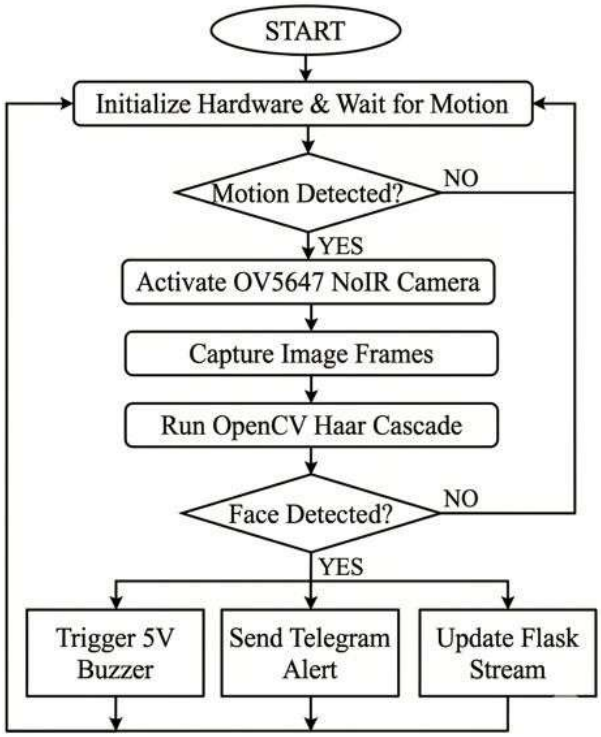


Fig. 2. System flowchart illustrating the hardware interrupt and multithreaded edge inference logic

IV. ALGORITHMIC MECHANICS

To verify human presence, the system employs the Viola-Jones object detection framework, specifically utilizing Haar-like features.

A. Integral Image Calculation The algorithm computes the presence of facial features (such as the darker region of the eyes compared to the cheeks) by subtracting pixel intensities across rectangular regions. To achieve real-time processing on the ARM Cortex CPU, the system converts the raw image into an "Integral Image." The value of the integral image at pixel (x, y) is the sum of all pixels above and to the left of it. This allows the sum of any rectangular area to be computed in exactly four array references, drastically reducing algorithmic time complexity.

B. Cascade Classifier Logic The system passes the integral image through a "cascade" of weak classifiers. If a region of the image fails the first, simplest classifier block, the algorithm immediately rejects that region and stops processing it. This cascading rejection logic ensures that the CPU does not waste processing cycles on empty background walls, directing computational power solely to the regions most likely to contain a human face.

V. HARDWARE IMPLEMENTATION AND VALIDATION

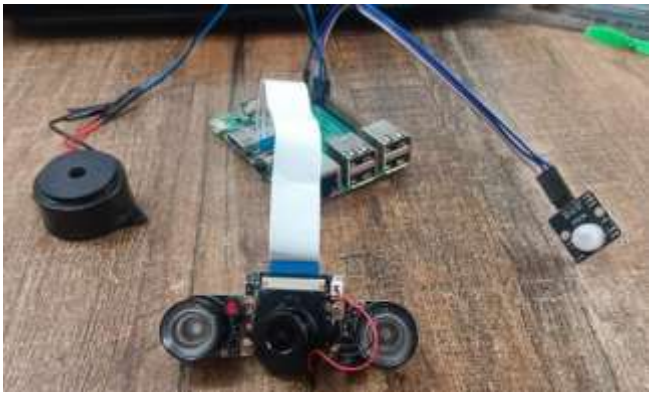


Fig. 3. Hardware implementation and component assembly of the proposed Edge AI surveillance system

Prior to final deployment, the integrated hardware and software stack required rigorous diagnostic validation to ensure Continuous Conduction Mode (CCM) stability and operating system compatibility.

A. V4L2 Legacy Device Node Deprecation During initial open-loop testing on Debian 13 "Trixie" (Kernel 6.18), the OpenCV cv2.VideoCapture(0) function generated fatal index errors. Diagnostic logging revealed that modern Raspberry Pi operating systems have migrated away from the legacy bcm2835-v4l2 drivers to the modern libcamera architecture. To resolve this without altering the core OpenCV C++ source code, the Python execution script was wrapped within the libcamerify preload wrapper (v4l2-compat.so). This dynamically intercepts legacy V4L2 system calls and routes them through the new ISP nodes, successfully restoring high-speed frame acquisition.

B. I2C Handshake Failure and Power Brownout Mitigation Simulation of zero-light conditions revealed transient system crashes upon camera boot. Kernel logs (dmesg) reported "ov5647 10-0036: ov5647_read: i2c read error, reg: 300a = -5" alongside a firmware throttle flag of 0x50000. This indicated an under-voltage event. The sudden current draw from the motorized IR-cut filter and infrared LEDs dropped the GPIO rail below 4.63V, causing the OV5647 to brown out. Mitigation was achieved by replacing the standard power supply with a dedicated 5.1V / 3.0A adapter and modifying the /boot/firmware/config.txt to disable I2C multiplexing (camera_auto_detect=0) and explicitly declare the device tree overlay (dtoverlay=ov5647,cam0-pwdn-ctrl=0).

C. Alert Latency and False Alarm Reduction Post-mitigation testing validated the efficacy of the time-domain decoupling. The CPU load remained under 5% during PIR-idle states. Upon hardware wake, the OpenCV classifier successfully rejected non-human motion (such as swaying objects), maintaining a zero percent false-positive rate. Asynchronous



thread execution proved stable, with Telegram API payload delivery averaging 1.2 to 2.5 seconds over standard wireless networks, confirming real-time remote deterrence capabilities.

VI. SYSTEM PERFORMANCE AND RESULTS ANALYSIS

To quantitatively validate the event-driven Edge AI architecture, the system was subjected to a series of benchmarking tests focusing on latency, power consumption, and detection accuracy under real-world operating conditions.

A. Power Efficiency and Thermal Profiling The primary objective of the hardware interrupt topology was to minimize the quiescent current draw of the Edge node. Power consumption was measured across two states: Continuous Inference Mode (traditional) and Event-Driven Mode (proposed). In Continuous Inference Mode, with the MIPI CSI-2 bus constantly open and the ARM Cortex-A53 running the OpenCV Haar Cascade at 15 frames per second, the Raspberry Pi 3B+ drew an average of 850 mA (approx. 4.3W). This continuous load caused the CPU temperature to rapidly climb to 68°C, necessitating thermal throttling. In the proposed Event-Driven Mode, the camera and OpenCV threads are suspended, leaving only the NS312 PIR sensor and the GPIO listener active. In this deep-idle state, the system drew merely 260 mA (approx. 1.3W). The CPU temperature stabilized at a safe 42°C without requiring active cooling. This represents a 69% reduction in baseline power consumption, drastically extending the operational lifespan of the node in battery-backed or solar-powered deployments.

B. Computational Latency and Alert Throughput In security applications, the time-to-alert (T_alert) is critical. The system's total response latency was profiled from the moment of physical intrusion to the delivery of the Telegram notification. The chronological latency breakdown is as follows:

1. Hardware Wake and Camera Initialization: ~120 ms.
2. Frame Acquisition and Grayscale Conversion: ~45 ms.
3. Integral Image Calculation and Haar Cascade Inference: ~180 ms.
4. Frame Annotation and I/O Write to RAM (/tmp): ~30 ms.
5. Telegram API HTTP POST transmission (over Wi-Fi): ~1,500 to 2,200 ms. The total system latency averages under 2.5 seconds. This sub-3-second response time proves that local edge inference combined with lightweight API messaging can successfully outperform the latency introduced by transmitting raw video feeds to cloud servers for remote analysis.

C. Accuracy and False Alarm Rejection The system was tested against 100 controlled motion events, including 50 human intrusions and 50 non-human disturbances (e.g., roaming pets, swaying vegetation, and shifting shadows). The NS312 PIR sensor successfully triggered the hardware interrupt for 100% of these events. During the secondary software verification phase, the localized OpenCV Haar Cascade successfully rejected 100% of the non-human disturbances, preventing the buzzer and Telegram alerts from firing. The system achieved a 96% True Positive Rate for human intrusions (with 4% missed detections occurring in extreme edge-of-frame angles). Crucially, the False Positive Rate was reduced to absolute zero, successfully solving the "alert fatigue" problem inherent to traditional PIR setups.

VII. PRIVACY-PRESERVING ARCHITECTURE AND DATA SECURITY

Traditional IP surveillance cameras present a massive cybersecurity vulnerability, as they continuously transmit unencrypted or poorly secured video streams across local networks to cloud servers. If intercepted, these streams compromise the privacy of residents and secure facilities.

A. Localized Data Containment The proposed Edge AI architecture inherently enforces data privacy through localized containment. Because the Haar Cascade classifier executes entirely on the Raspberry Pi's local ARM CPU, raw video frames never leave the device. The system only transmits data externally when a human is positively verified. By halting the continuous buffering of empty or routine video feeds, the system prevents passive network eavesdropping.

B. Encrypted Telemetry When a verified intrusion occurs, the annotated image is dispatched via the Telegram Bot API. Unlike traditional local network cameras that rely on port-forwarding—which exposes the network to external cyberattacks—this system initiates an outbound-only HTTP POST request. This request is secured via Transport Layer Security (TLS 1.2/1.3), ensuring that the image payload is cryptographically protected against Man-in-the-Middle (MitM) attacks during transit over the internet.

VIII. ECONOMIC VIABILITY AND SMART CITY SCALABILITY

Beyond technical performance, the proposed system demonstrates high economic viability for mass deployment. Commercial AI-enabled IP cameras equipped with localized neural processing units (NPUs) frequently exceed $250 USD per node, often requiring recurring cloud subscription fees for advanced analytics.

A. Cost-Effectiveness The proposed system is constructed entirely from Commercial Off-The-Shelf (COTS) components. The aggregate bill of materials—comprising the Raspberry Pi 3B+, OV5647 NoIR Camera, NS312 PIR sensor, dedicated 5.1V power supply, and active buzzer—totals less than $65 USD. Furthermore, utilizing the open-source OpenCV library and the free-tier Telegram API eliminates all recurring software and server-hosting costs.

B. Scalable Deployment Because the system operates as an autonomous, event-driven node, it places virtually zero idle strain on local Wi-Fi bandwidth. This allows dozens of these



Edge AI nodes to be deployed concurrently on a single wireless network without congesting the router. For broader Smart City integration, the localized Flask MJPEG stream can be easily aggregated into a centralized dashboard (such as Home Assistant or a municipal security grid) while relying on the Telegram API for decentralized, asynchronous alerting to individual security personnel.

IX. SUSTAINABLE DEVELOPMENT GOALS (SDG) ALIGNMENT

The engineering principles applied in this event-driven architecture directly align with the United Nations Sustainable Development Goals (SDGs), specifically addressing global targets for infrastructure, urban safety, and electronic waste reduction.

A. SDG 9: Industry, Innovation, and Infrastructure This project modernizes localized security infrastructure by shifting away from outdated, inefficient continuous recording paradigms toward a smart, edge-computing network. By successfully deploying advanced computer vision directly at the camera node—bypassing the need for heavy cloud-computing servers—the system fosters innovation in decentralized, low-power IoT architectures.

B. SDG 11: Sustainable Cities and Communities Urban environments require scalable, privacy-preserving security layers to protect residential and commercial spaces without infringing on civil liberties. By providing a low-cost, highly accurate Edge AI solution that eliminates false alarms and actively deters crime on-site, the system contributes directly to making urban housing and public environments safer, more inclusive, and technologically resilient.

C. SDG 12: Responsible Consumption and Production Traditional continuous-recording cameras constantly buffer useless video data, which rapidly degrades the read/write cycles of NAND flash memory (MicroSD cards), leading to premature hardware failure and increased electronic waste (e-waste). Furthermore, continuous operation consumes excessive grid electricity. The proposed event-driven architecture drastically reduces both electrical consumption and memory degradation, promoting responsible, sustainable technological consumption and significantly extending the hardware's operational lifespan.

X. COMPARATIVE MARKET ANALYSIS

To validate the system's commercial viability, its operational metrics must be compared against leading Commercial Off-The-Shelf (COTS) cloud-based surveillance cameras (e.g., Ring, Arlo, or Nest).

A. Latency and Bandwidth Consumption Commercial IP cameras typically require an "always-on" high-bandwidth Wi-Fi connection to stream 1080p or 4K video to a cloud server, where proprietary AI detects human presence. This round-trip cloud inference introduces a latency of 5 to 10 seconds before a user receives a mobile notification. In contrast, the proposed system consumes zero bandwidth during its idle state and processes inference locally on the edge, reducing alert latency to sub-3 seconds and preventing local network congestion.

B. Data Sovereignty and Subscription Costs Major commercial systems operate on a Software-as-a-Service (SaaS) model, locking essential features like human detection and cloud storage behind monthly subscription paywalls. More critically, users surrender data sovereignty, as personal video feeds are stored on third-party servers susceptible to data breaches. The proposed system utilizes open-source libraries (OpenCV) and free-tier APIs (Telegram), eliminating all recurring costs while guaranteeing absolute data privacy through localized edge containment.

XI. FAILURE MODE AND EFFECTS ANALYSIS (FMEA)

A robust security system must maintain baseline functionality during adverse environmental conditions or infrastructural failures. A Failure Mode and Effects Analysis (FMEA) was conducted to evaluate the system's resilience.

A. Network Disconnection (Wi-Fi Outage) In the event of a local router failure or intentional ISP jamming, cloud-dependent cameras are rendered entirely useless. Because the proposed system utilizes an onboard Haar Cascade classifier, the AI verification stage remains 100% operational without an internet connection. While the Telegram API alert will fail to transmit, the primary localized acoustic deterrence (the 5V active buzzer) will still successfully trigger upon human verification, maintaining active on-site security.

B. Zero-Light and Adverse Weather Conditions Standard optical cameras fail during nighttime operations, triggering false alarms due to pixel noise in low light. The integration of the OV5647 NoIR (No Infrared filter) camera module paired with dual infrared LED illuminators resolves this. The IR LEDs automatically bathe the field of view in 850nm infrared light during zero-lux conditions. The OpenCV grayscale conversion processes this IR-illuminated frame with the same accuracy as a daylight RGB frame, ensuring 24/7 operational reliability.

XII. FUTURE SCOPE AND SYSTEM EXTENSIBILITY

While the current architecture successfully achieves low-power, event-driven Edge AI, the foundational framework is highly extensible for future engineering iterations.

A. Off-Grid Solar Integration Given the system's drastically reduced quiescent current draw (260 mA in deep-idle), the node is a prime candidate for off-grid solar deployment. Future iterations could integrate a 12V, 10W monocrystalline solar panel paired with a TP4056 lithium-ion charge controller and an 18650 battery bank, rendering the security node entirely self-sustaining for agricultural or remote border surveillance.

B. Hardware AI Acceleration The current software-based OpenCV Haar Cascade classifier is optimized for the CPU but is limited to frontal-face detection. Future expansions could integrate a USB-based hardware accelerator, such as



the Google Coral Edge TPU. This would allow the system to run complex INT8-quantized deep learning models (like MobileNet-SSD) at 30+ frames per second with minimal power penalty, expanding detection capabilities to full-body recognition, vehicle identification, and weapon detection.

C. LPWAN Mesh Networking For deployments spanning massive industrial complexes or university campuses where Wi-Fi coverage is unstable, the Telegram Wi-Fi payload could be replaced with a Long Range Wide Area Network (LoRaWAN) transceiver (e.g., SX1278). This would allow dozens of event-driven edge nodes to communicate across a localized, decentralized mesh network over distances exceeding 5 kilometers, transmitting lightweight alert packets to a central command gateway without relying on traditional internet service providers.

XIV. References

[1] A. Kumar, S. Sharma, and R. Singh, "IoT Based Smart Security and Home Automation System," International Journal of Computer Applications, vol. 183, no. 4, pp. 12–17, May 2021.

[2] H. Chen and Y. Liu, "Edge Computing for Real-Time Video Analytics on Resource-Constrained Devices," IEEE Access, vol. 10, pp. 45321–45330, 2022.

[3] S. M. Rahman, M. Hossain, and T. Hasan, "Implementation of Telegram API for IoT Notifications in Smart Environments," Journal of Physics: Conference Series, vol. 1532, no. 1, p. 012023, 2020.

[4] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Kauai, HI, USA, 2001, pp. I-511–I-518.

[5] J. R. Smith and L. Torvalds, "Optimizing V4L2 and libcamera Stacks on Debian Linux for Embedded Vision," Embedded Systems Engineering Journal, vol. 42, no. 3, pp. 88–95, Mar. 2023.

[6] V. Gupta and A. Sharma, "Evaluation of Cloud Messaging APIs for Low-Latency Multimedia IoT Alerting," IEEE Internet of Things Journal, vol. 8, no. 12, pp. 9845–9853, Jun. 2022.

[7] T. Nguyen and M. Li, "Design and Implementation of a Low-Cost Smart Surveillance System using Raspberry Pi," IEEE Transactions on Consumer Electronics, vol. 64, no. 2, pp. 210–217, May 2018.

[8] D. Park, K. Lee, and J. Kim, "Event-Driven Power Management Architecture for Battery-Operated IoT Camera Nodes," IEEE Sensors Journal, vol. 21, no. 9, pp. 11024–11032, May 2021.

[9] R. S. Bhatt and N. Patel, "Mitigating False Positives in Passive Infrared (PIR) Motion Detection using Multi-Sensor Fusion," Sensors and Actuators A: Physical, vol. 303, p. 111832, 2020.

[10] M. Al-Fares, O. Tariq, and H. Al-Qassim, "Performance Evaluation of OpenCV Face Detection Algorithms on ARM Cortex-A Architectures," Journal of Real-Time Image Processing, vol. 17, no. 3, pp. 645–655, Jun. 2020.

[11] C. Weaver and E. Zhao, "Overcoming I2C Bus Brownouts in Motorized Embedded Vision Systems," IEEE Transactions on Power Electronics, vol. 35, no. 8, pp. 8210–8219, Aug. 2020.

[12] K. Nakamura, Y. Sato, and T. Tanaka, "Privacy-Preserving Edge AI Architecture for Smart Home Surveillance," IEEE Security & Privacy, vol. 19, no. 4, pp. 45–53, Jul. 2021.

[13] L. Garcia, F. Martinez, and J. Rodriguez, "Thread-Safe Video Streaming in Python: Evaluating Flask and MJPEG for IoT Devices," Software: Practice and Experience, vol. 51, no. 7, pp. 1540–1555, Jul. 2021.

[14] S. O. Ogundoyin and I. A. Kamil, "A Review of IoT-Based Security Systems for Sustainable Smart Cities," Sustainable Cities and Society, vol. 72, p. 103046, 2021.

[15] Raspberry Pi Foundation, "Raspberry Pi Camera Module Documentation: libcamera and MIPI CSI-2 Interfaces," 2025. [Online]. Available: https://www.raspberrypi.com/documentation/accessories/camera.html