Power-Efficient Surveillance Camera Using Sleep Mode and YOLOv3 Model-Based Edge Computing

Mhd. Idham Khalif[1*], Raden Deiny Mardian[2], Ade Faiz Kurnia Putra[3], M. Dhanu Wicaksono[4], Tirta Akdi Toma Mesoya Hulu[5], and Listyo Edi Prabowo[6]
[1−4]Electrical Engineering, Faculty of Industrial Technology, Universitas Trisakti
Jakarta, Indonesia 11440
[5]Master of Interaction Design, Faculty of Engineering, Architecture and Information Technology,
The University of Queensland
Brisbane, Australia 4072
[6]Electrical Engineering Department, Faculty of Engineering, Universitas Indonesia
Depok, Indonesia 16424
Email: [1]idham.khalif@trisakti.ac.id, [2]deiny_wp@trisakti.ac.id, [3]062002300007@std.trisakti.ac.id, [4]062002300004@std.trisakti.ac.id, [5]t.hulu@student.uq.edu.au, [6]listyo.edi@ui.ac.id

***Abstract*—Surveillance cameras play a vital role in a wide range of monitoring applications, particularly in ensuring real-time security and observation. However, conventional surveillance systems often face limitations in energy efficiency, especially when deployed in remote locations or powered by battery sources. Although many surveillance cameras offer high-resolution capabilities, only a few incorporate power management strategies to optimize energy usage. The research presents the design and implementation of a low-power surveillance camera system based on the ESP32-CAM platform, incorporating a sleep mode to enhance power efficiency. Two operational scenarios are tested: one with enabled sleep mode and one without. Experimental results show that the camera without sleep mode achieves a higher frame rate of up to 17.01 FPS than the sleep-enabled camera with a maximum of 3.53 FPS. Despite the reduced frame rate, the system successfully performs object detection using the YOLOv3 model processed via edge computing. Furthermore, the average wake-up time from sleep mode is 1.414 seconds, indicating a fast, responsive system suitable for low-power embedded applications. In terms of energy consumption, the sleep-enabled device consumes only 3475.543 mW over 2 hours of operation, compared to 5561.639 mW for the device without sleep mode, resulting in an energy saving of approximately 37.5%. These findings confirm that implementing sleep mode is effective in managing power consumption without compromising core surveillance functionality. The research contributes to the development of sustainable and energy-efficient monitoring solutions and highlights the potential for further enhancement through advanced edge computing platforms in future work.**

***Index Terms*—Power Efficient, Surveillance Camera, Sleep Mode, YOLOv3, Edge Computing**

Received: June 05, 2025; received in revised form: Sep. 25, 2025; accepted: Sep. 25, 2025; available online: March 05, 2026.
*Corresponding Author

I. Introduction

The rapid advancement of technology has significantly impacted various sectors. Innovations in Information and Communication Technology (ICT), such as Artificial Intelligence (AI), Machine Learning (ML), Internet of Things (IoT), and cloud computing, have accelerated digital transformation across key domains, including education, healthcare, industry, and government. These technologies not only enhance efficiency and productivity but also establish new paradigms in how people interact, communicate, and manage data. This shift is largely driven by the ongoing Fourth Industrial Revolution, characterized by the convergence of physical, digital, and biological systems. This era has triggered profound changes in business models and organizational structures, promoting broader automation and the integration of big data and AI into decision-making processes. A recent study has identified that, among numerous publications on technology adoption in Industry 4.0, nearly half utilize machine learning algorithms and neural networks. This finding indicates that AI, particularly through ML approaches, is playing an increasingly dominant role in supporting automation processes, enhancing quality control, and improving operational efficiency, especially in the manufacturing sector [1].

One of the most tangible implementations of recent technological advancements is the use of surveillance cameras or Closed-Circuit Television (CCTV). It has become increasingly sophisticated and integrated with IoT and AI technologies. Both of them rely heavily on Internet connectivity as their core foundation. A dominant trend in this transformation is the proliferation of electronics-based technologies. According to data from Statistics Indonesia (Badan Pusat Statistik (BPS)), by 2022, 67.88% of the Indonesian population owned mobile phones, and 66.48% of them had Internet access [2]. Modern CCTV systems are no longer merely passive visual recording devices. They have evolved into real-time analytics platforms which are capable of object detection, facial recognition, and people counting. They significantly support automated security systems and spatial management. The integration of the IoT allows CCTV cameras to connect and communicate via the Internet, enabling remote access and processing of surveillance data. Moreover, the incorporation of ML into surveillance systems further enhances adaptability and detection accuracy, thereby improving the effectiveness of environmental monitoring, crime prevention, and public facility management [3].

With the widespread adoption of modern surveillance cameras, these devices are fundamentally electronic equipment that require power sources to operate and function according to the applied technologies. According to the 2023 PLN Statistical Report, the total electricity sold in Indonesia reached 288.44 TWh, with the household sector accounting for 122.34 TWh, or approximately 42.41% of total national electricity consumption. The data highlight the household sector as the largest consumer of electricity, reflecting its significant reliance on electronic devices for daily activities [4].

On the other hand, the Indonesian government has initiated numerous energy-saving programs aiming to achieve sustainable energy. Various policies and strategic measures, such as the implementation of energy efficiency in the industrial and household sectors, the utilization of renewable energy sources, and the development of energy-saving technologies, are prioritized to reduce overall electricity consumption. Energy conservation is a systematic, integrated, and sustainable effort to preserve domestic energy resources and increase the efficiency of their utilization. The goal of optimizing energy use in energy systems is increasingly important as an effort to ensure sustainable energy availability in the future [5]. Every individual and business entity is obliged to carry out energy conservation. This regulation defines the specific responsibilities of the central government, regional governments, business actors, and the community in the implementation of energy conservation [6]. Therefore, the energy consumption of electronic devices, including surveillance cameras, is an important consideration in energy efficiency efforts.

In addition, with the increasing number and complexity of surveillance cameras equipped with advanced technologies, such as AI and IoT connectivity, power demand tends to increase. It is mainly due to complex computing operations and wireless communication, which inherently consume a considerable amount of power. Energy consumption in edge AI devices is generally dominated by computation, which typically ranges from a few to several tens of watts, while communication (e.g., Wi-Fi) consumes significantly less power, at around 32 mW [7].

The development of energy-efficient surveillance cameras is essential to support environmental sustainability and reduce national energy consumption. The implementation of energy-saving technologies not only has a positive economic impact but also aligns with global efforts to reduce carbon emissions and preserve natural resources. Approaches such as the use of ESP32-CAM–based camera modules, which can operate on battery power to support mobility, offer efficiency through low power consumption and the ability to perform basic monitoring functions independently without compromising camera functionality. These modules incorporate power-saving methods that can be conditionally activated and, when combined with machine learning algorithms to recognize specific situations for energy conservation, present a promising alternative solution to energy-saving challenges [8, 9].

The ESP32-CAM module is a device with high power consumption efficiency, making it highly suitable for IoT applications in remote monitoring. This device is equipped with a microcontroller integrated with Bluetooth and Wi-Fi connectivity, along with support for data storage via an Secure Digital (SD) card. Energy efficiency plays a crucial role in IoT, especially for battery-powered devices deployed in remote or hard-to-reach locations, where long operational lifespans of over 10 years are highly desirable [10].

To address the resource constraints of the ESP32-CAM, edge computing combined with lightweight models is an effective approach for object detection. Applying edge computing for object detection with You Only Look Once (YOLO) can improve runtime, memory consumption, and power usage by up to 445%, 69%, and 73%, respectively [11]. Previous research applying YOLOv3-Tiny for object detection has reported that the monitoring device consumes only 3.5 W of power [12]. Another study has demonstrated that applying edge computing with the YOLO model can significantly improve the inference speed of the network, with performance improvements of 61.88%, 69.1%, 59.36%, 64.07%, and 65.92%, thereby enhancing accelerator performance and surpassing existing methods [13]. Furthermore, the use of lightweight YOLO models optimized with edge computing demonstrates a balance between accuracy and efficiency while maintaining high inference speed [14]. These studies confirm that combining edge computing with YOLO variants can optimize object detection to achieve more accurate and real-time performance on low-power camera systems such as the ESP32-CAM.

Previous studies have demonstrated the feasibility of building a real-time object detection camera system using the YOLO model on the ESP32-CAM platform. Their results show that this platform is capable of running lightweight AI applications despite its limited resources [15]. With proper power management support, the operational efficiency of resource-constrained devices can be further improved. Another study has successfully implemented a surveillance camera capable of detecting humans using the YOLO model processed through edge computing, achieving a detection accuracy of 95.05% at 2 Frames per Second (FPS) [16]. These findings indicate that while the YOLO model provides high detection accuracy, it may lead to reduced frame rates.

More recent research has described an ESP32-CAM-based system performing real-time object detection using an AI model. The results are processed locally and transmitted to a server via Message Queuing Telemetry Transport (MQTT) while also being displayed through a local interface [17]. The system is further integrated with the Blynk application through an IoT network, enabling remote monitoring and control over the Internet. Additionally, another study has implemented a compressed YOLOv3 model on a resource-limited Unmanned Aerial Vehicle (UAV) for real-time detection, maintaining the original detection efficiency of the UAV system [18]. These results suggest that YOLOv3 model compression can be an effective solution to improve computational performance and achieve higher FPS, particularly in resource-constrained environments.

Although numerous studies have successfully developed ESP32-CAM-based surveillance camera systems with object detection and IoT integration capabilities, several limitations remain as key challenges. First, most existing systems lack effective power management mechanisms, resulting in inefficient energy utilization, particularly on resource-constrained devices. The need for effective power management is not only important but also urgent, especially for standalone IoT devices operating under strict energy constraints [19, 20]. Second, limited studies have explored the use of ML-based real-time person detection as a trigger for activating power-saving mechanisms, such as sleep mode, to reduce energy consumption without compromising the primary function of surveillance cameras. Therefore, in line with the concept of sustainable energy, electronic devices must be designed to manage power efficiently.

In this research, the reseachers design and implement an ESP32-CAM–based surveillance camera with minimal resources and specifications while maintaining its core functionality. The camera is capable of counting the number of people entering the image frame using the YOLOv3 object detection model implemented through edge computing. This capability serves as the basis for activating sleep mode, thereby enabling efficient and precise power management. Furthermore, the research compares the camera's power consumption with and without sleep mode to evaluate the effectiveness of energy savings. The proposed system is intelligent, accurate, and energy-efficient, making it suitable for long-term monitoring in resource-constrained environments while contributing to the development of low-specification, power-efficient surveillance camera solutions.

II. Research Method

A. Surveillance Camera Device Architecture

The surveillance camera device developed is designed for power efficiency and real-time object detection. It utilizes an ESP32-CAM module as the core component, along with a microwave sensor and battery components. Figure 1 shows a schematic diagram of the surveillance camera device. In general, the system architecture consists of several main components, namely the ESP32-CAM camera module, which also functions as the processing unit (microcontroller). This ESP32-CAM module is equipped with wireless communication capabilities (Wi-Fi), enabling it to transmit image data in real time to the edge computing unit without using cables.

In addition, the system is equipped with an HFS-DC06 microwave sensor. The HFS-DC06 is a microwave-based motion detection sensor that operates on the Doppler radar principle to detect object movement within its coverage area. This sensor emits high-frequency electromagnetic waves (approximately 5.8 GHz) and detects frequency shifts in the waves reflected by moving objects, such as humans or vehicles. The sensor functions as a wake-up trigger when the device is in deep sleep mode and detects the presence of a person near the camera.

Cite this article as: M. I. Khalif, R. D. Mardian, A. F. K. Putra, M. D. Wicaksono, T. A. T. M. Hulu, and L. E. Prabowo, “Power-efficient surveillance camera using sleep mode and YOLOv3 model-based edge computing”, CommIT Journal 20(1), 1–15, 2026.

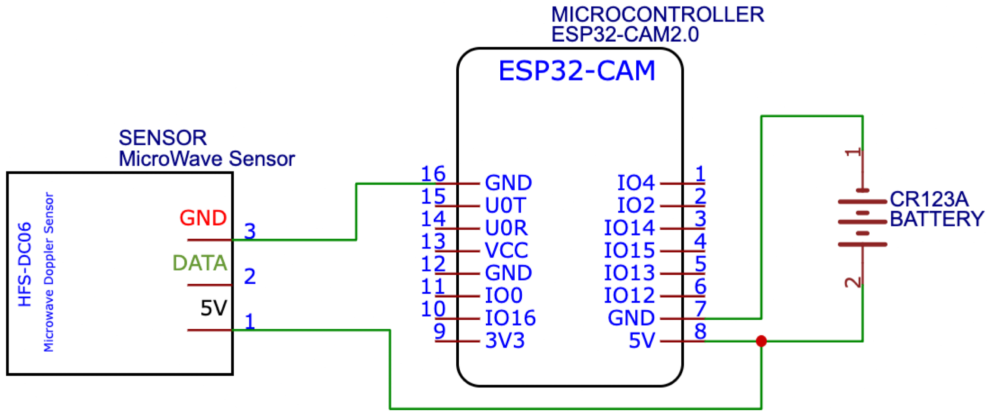

Fig. 1. Schematic diagram of surveillance camera device. Note: GND: Ground and VCC: Voltage Common Collector.

The HFS-DC06 microwave sensor is connected to digital pin 13. Based on the ESP32 datasheet, one of the pins that supports external wake-up functionality can be activated by setting a HIGH (logic 1) signal on pin 13 [21]. The selection of the HFS-DC06 sensor is based on its relatively wide detection range of up to 12 m, with an ideal installation height between 2 and 6 m. Microwave-based sensors utilizing the Doppler principle are widely used for motion detection due to their low power consumption, compact design, and cost-effective production. Compared to ultrasonic systems, this sensor offers a longer detection range and can be installed behind non-metallic materials that do not significantly reflect electromagnetic waves [22].

The device is powered by a compact CR123 battery (3.7 V, 2650 mWh, 34×17 mm), selected for its ability to supply sufficient energy while maintaining a small, portable, and easy-to-install design without requiring an external power source. This battery is capable of delivering sufficient power to support continuous operation of the embedded system, wireless communication module, and peripheral components while maintaining voltage stability. Its compact cylindrical form factor enables the overall device to remain small, lightweight, and portable, making it suitable for deployment in space-constrained or hard-to-access environments. Furthermore, the use of an integrated battery eliminates the need for an external power supply or complex wiring infrastructure, thereby simplifying installation, reducing maintenance requirements, and enhancing system flexibility. This self-contained power configuration also supports rapid deployment in field applications, particularly in remote or mobile monitoring scenarios where reliability and ease of installation are critical factors.

B. Overall Surveillance Camera Architecture

The overall architecture of the surveillance camera system integrates various hardware and software components to create a system capable of intelligently managing power usage. The system is designed not only to capture images and videos in real time but also to perform local data processing (edge computing) to detect objects, specifically the number of people [23–25]. Figure 2 illustrates the overall architecture of the surveillance camera system. The first block consists of the surveillance camera device, which captures images in real time and transmits them to the edge computing unit via a Local Area Network (LAN) connected to the same Wi-Fi router.

Both the surveillance camera device and the edge computing unit implement the Multicast Domain Name System (mDNS). The purpose of mDNS is not to assign static IP addresses, but to enable hostname-based service discovery within a local network without requiring manual IP configuration. Thus, even if the IP address of a device changes dynamically, communication can still be maintained through its hostname. The use of mDNS/Bonjour allows IoT devices, including cameras, to automatically discover and connect to edge nodes without relying on static IP addresses, even when IP changes occur [26].

Among various service discovery protocols, mDNS is a lightweight and standard-based solution widely adopted in smart home and IoT networks. Recent studies have proposed integrating spatial and location

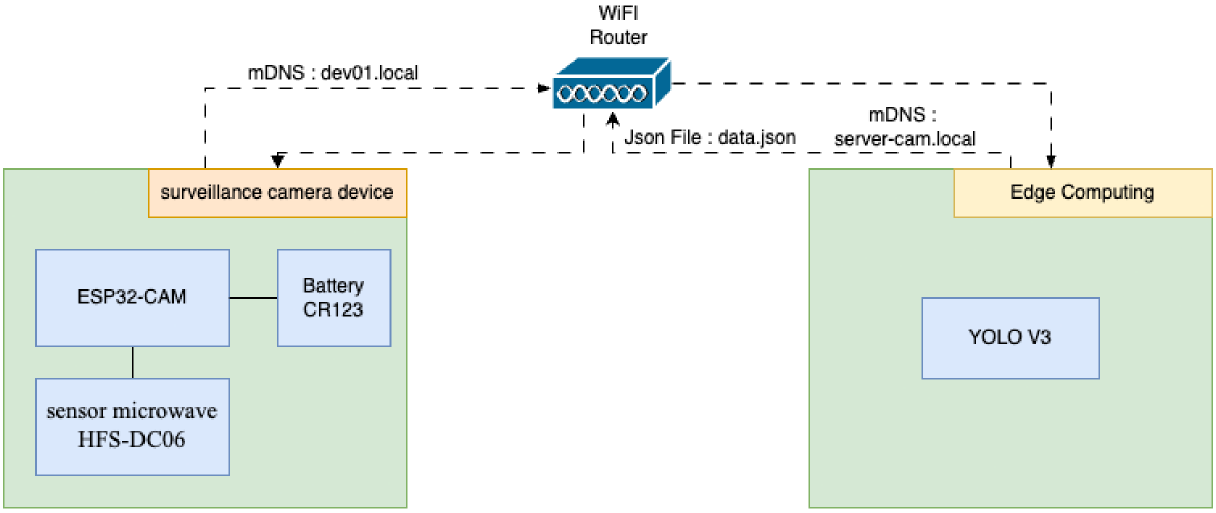

Fig. 2. Overall camera system architecture diagram. Note: Multicast Domain Name System (mDNS) and JavaScript Object Notation (JSON).

data into mDNS/DNS-SD frameworks to enhance service discovery performance in IoT environments [27]. Conventional mDNS/DNS-SD framework integrates spatial and location-aware information into the service discovery process. These improvements enable devices to not only identify available services but also evaluate their physical proximity or contextual relevance. By incorporating spatial metadata, discovery mechanisms can improve efficiency, reduce latency, and optimize network traffic, especially in dense IoT deployments where multiple devices provide similar services. The approach contributes to smarter, context-aware service selection and improves overall performance in distributed IoT systems.

After the image data are transmitted to and received by the edge computing unit, they are processed using the YOLOv3 machine learning model to recognize and count objects, specifically people. Once the YOLOv3 model successfully performs object detection and counting, the resulting data are parsed and formatted into a JavaScript Object Notation (JSON) file. This file can be accessed locally via mDNS from the edge computing unit and read by the surveillance camera device to trigger sleep mode activation.

In this system, the detection data from edge computing, particularly the number of detected individuals, are provided through a local network Hypertext Transfer Protocol (HTTP) endpoint in JSON format, which can be accessed using mDNS. The resulting data are structured in a lightweight and machine-readable JSON format, enabling seamless integration with external applications, dashboards, or monitoring systems. Access to this endpoint is facilitated through mDNS-based hostname resolution, allowing users or client devices to retrieve the data using a human-readable local domain name rather than a fixed IP address. This approach enhances system flexibility, simplifies deployment within local area networks, and supports plug-and-play functionality in dynamic IoT environments.

C. Power Management Workflow on Surveillance Camera Devices

Power management is a crucial aspect in designing surveillance camera devices, especially when they operate under limited energy resources such as batteries. The proposed system incorporates a power management scheme that integrates the deep sleep feature of the ESP32-CAM module with the use of the HFS-DC06 microwave motion sensor as a wake-up trigger to improve energy efficiency. To enhance energy efficiency, the proposed system integrates a structured power management scheme that combines the deep sleep feature of the ESP32-CAM module with the HFS-DC06 microwave motion sensor functioning as an external wake-up trigger. During idle periods, the ESP32-CAM enters deep sleep mode, significantly reducing current consumption by disabling non-essential subsystems while retaining minimal functionality required for wake-up detection. When motion is detected, the HFS-DC06 sensor generates a trigger signal that reactivates the microcontroller, enabling image capture and subsequent processing only when relevant activity occurs. This event-driven operation model minimizes unnecessary power usage, extends battery life, and ensures that the surveillance system remains responsive while maintaining high energy efficiency. Such an approach is particularly suitable for remote, portable, or

Cite this article as: M. I. Khalif, R. D. Mardian, A. F. K. Putra, M. D. Wicaksono, T. A. T. M. Hulu, and L. E. Prabowo, “Power-efficient surveillance camera using sleep mode and YOLOv3 model-based edge computing”, CommIT Journal 20(1), 1–15, 2026.

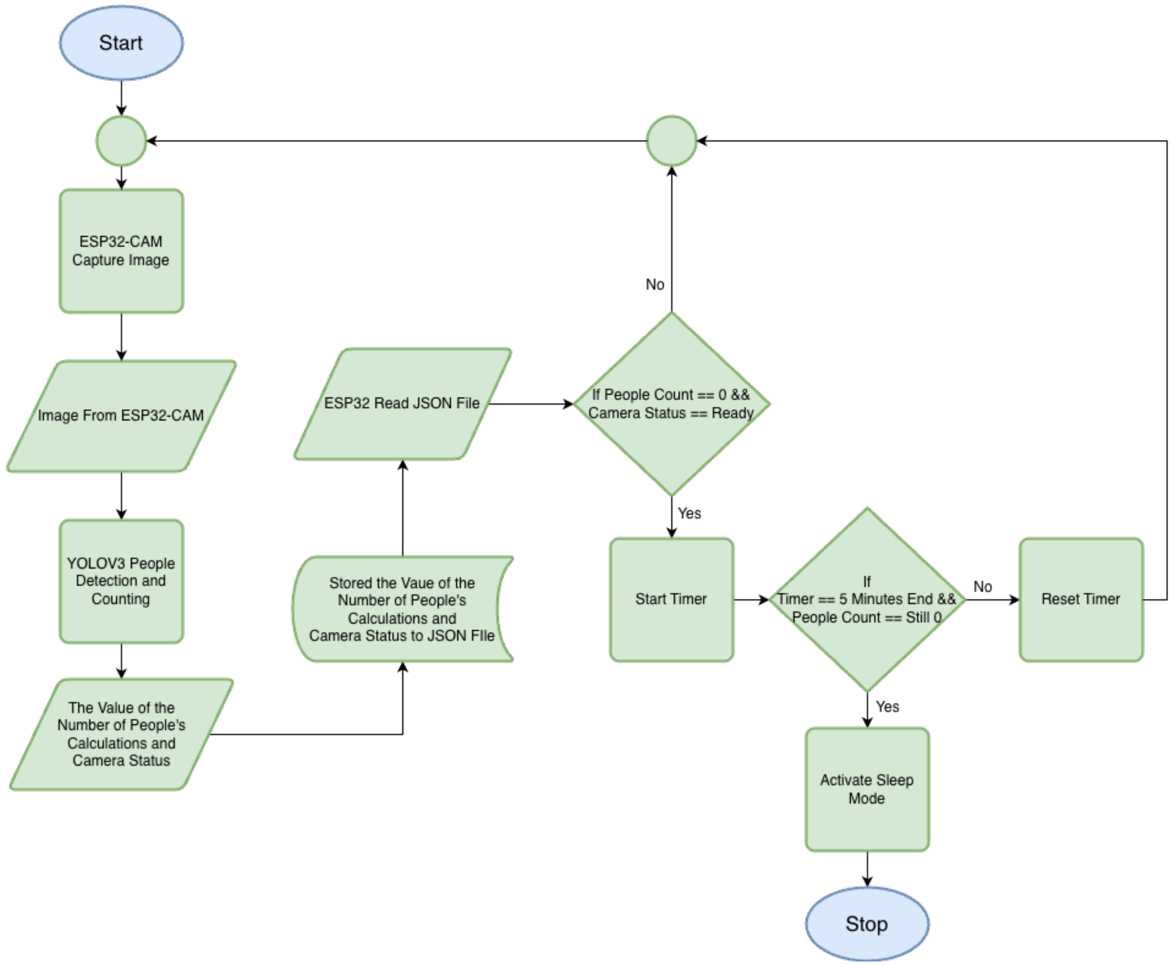

Fig. 3. Sleep mode implementation workflow for power management.

difficult-to-access installations where frequent battery replacement is impractical.

The sleep mode implemented in this system utilizes the deep sleep feature of the ESP32 module. According to the ESP32 datasheet, when operating in active mode, the ESP32 consumes an average current of approximately 115 mA, whereas in deep sleep mode, the average current is reduced to about 6μA [28]. Previous studies have evaluated various power modes of the ESP32 module, including deep sleep and reported that the average current during deep sleep reaching approximately 11 mA, even when accounting for active components such as voltage regulators, compared to about 50 mA in active mode [29].

Figure 3 illustrates the workflow of the sleep mode implementation for power management in the proposed surveillance camera system, demonstrating the operational logic of the power management mechanism. Image data captured by the ESP32-CAM device are transmitted to the edge computing unit and processed using the YOLOv3 model to detect and count individuals within the image frame. After successful detection and counting, the resulting data, along with the camera status, are formatted into a JSON file and made accessible through the local network for retrieval by the ESP32-CAM device.

Once the ESP32-CAM retrieves the number of detected individuals and the camera status, a timer is activated when the number of people is equal to zero and the camera status is set to ready. The timer operates for 5 minutes as a delay before entering sleep mode. If a person is detected during this interval, the timer is reset. When the timer reaches 5 minutes without detecting any individual, the camera device transitions into deep sleep mode and remains in this state until a motion signal is received from the HFS-DC06 microwave motion sensor. Upon detecting motion, the sensor sends a wake-up signal to the ESP32-CAM, causing the system to exit deep sleep mode and resume normal operation. The device then reinitializes the

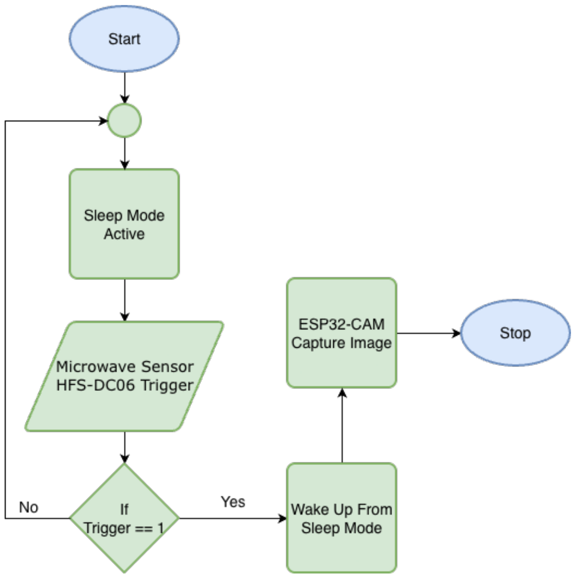

Fig. 4. Wake-up mode implementation workflow for camera reactivation.

camera module, reconnects to the local network, and continues the person detection process.

Figure 4 illustrates the workflow diagram of the wake-up mode activation process. When the device is in deep sleep mode, the HFS-DC06 microwave sensor detects human presence and triggers the system to wake up and resume operation. External wake-up trigger mechanisms can be implemented on the ESP32-CAM, where the image capture frequency is increased when a PIR sensor detects motion. When no motion is detected, the system automatically switches back to a low-frequency mode to conserve energy. In terms of detection range, the HFS-DC06 sensor provides wider coverage compared to PIR sensors.

D. Power Usage Measurement Architecture

The power usage measurement architecture is designed to monitor power consumption in real time to support comparative analysis of energy usage in the surveillance camera device. The power measurement unit is developed as a separate module using an independent microcontroller and a current sensor. It transmits the measurement data to a spreadsheet file, enabling time-series analysis of power consumption.

Figure 5 presents the schematic diagram of the power measurement system implemented for the surveillance camera device. The power measurement unit incorporates an ESP8266 WROOM microcontroller, which functions as the data acquisition and processing unit. This microcontroller is interfaced with the INA219 current sensor, which serves as the primary component for measuring electric current directly.

The INA219 current sensor measures electrical current by detecting the differential voltage across its internal shunt resistor. When current flows through the resistor, a small potential difference is generated, and the sensor converts this voltage drop into a digital current measurement value. The measurement data obtained from the INA219 are transmitted to the ESP8266 microcontroller via the I^2C communication protocol. The ESP8266 is connected to the edge computing unit, which provides JSON-formatted data over the Local Area Network (LAN). These data include the number of detected individuals, FPS, device active duration, wake-up time, and Wi-Fi connection time. After receiving the information, the ESP8266 processes the combined measurement and system performance data and uploads them to an online spreadsheet through an Internet connection. This mechanism enables structured time-series storage, remote monitoring, and further analysis of the recorded measurement results.

E. Comparative Evaluation of Object Detection Models

To identify the most suitable object detection model for a low-power surveillance system, YOLOv3, MobileNet-SSD, and EfficientDet are comparatively evaluated with a focus on human detection. Experiments conducted using real-time ESP32-CAM data at various resolutions show that, although lightweight models provided higher inference speeds, YOLOv3 achieves the most stable detection accuracy, making it the preferred choice for consistent performance under diverse conditions.

Table I presents a performance comparison of each machine learning model in detecting human objects. The evaluation is conducted by comparing the detection capability of each model over a two-hour observation period and calculating the detection success rate. This success rate is determined by dividing the number of frames in which human objects are successfully detected by the total number of analyzed frames and multiplied the result by 100%. This metric represents how consistently each model was able to accurately detect humans under real-time observation conditions using low-resolution ESP32-CAM data. YOLOv3 is selected as the most suitable model for the proposed power-efficient surveillance camera system due to its strong and consistent human detection performance across multiple resolutions. When integrated with the sleep mode mechanism, the system ensures reliable real-time monitoring while conserving energy, thereby providing an effective solution for edge computing-based surveillance under limited resource conditions.

Cite this article as: M. I. Khalif, R. D. Mardian, A. F. K. Putra, M. D. Wicaksono, T. A. T. M. Hulu, and L. E. Prabowo, “Power-efficient surveillance camera using sleep mode and YOLOv3 model-based edge computing”, CommIT Journal 20(1), 1–15, 2026.

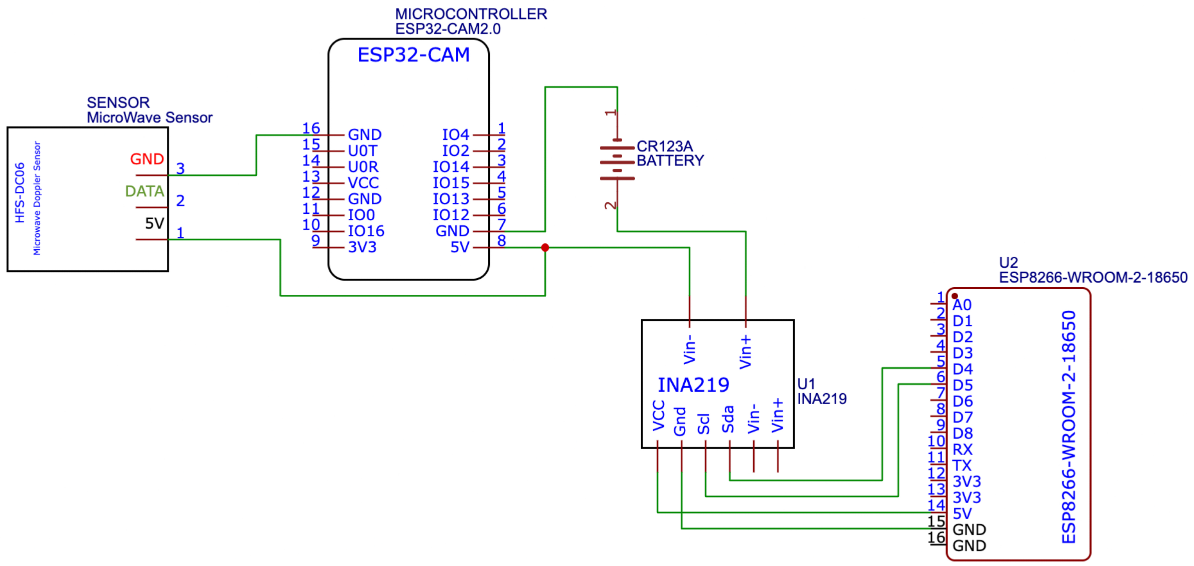

Fig. 5. Power usage measurement architecture. Note: GND: Ground and VCC: Voltage Common Collector.

TABLE I
RESULTS OF COMPARISON PERFORMANCE MACHINE LEARNING MODELS FOR HUMAN DETECTION.

Resolution	YOLOv3 (%)	MobileNet-SSD (%)	EfficientDet (%)
160×120	63.33	33.33	10.00
320×240	93.33	56.67	43.33
800×600	100.00	86.67	80.00

F. YOLOv3

YOLOv3 is a reliable real-time object detection model that balances accuracy and computational efficiency through its Darknet-53 backbone architecture. As a one-stage detector, it processes images in a single forward pass, making it suitable for low-power surveillance systems such as the ESP32-CAM. Previous studies have reported strong performance, with detection accuracy reaching 74.1% at a resolution of 512×512 [30, 31]. This balance between speed and accuracy makes YOLOv3 a practical choice for edge computing-based monitoring applications, where both efficiency and responsiveness are essential. In the research, the YOLOv3 model is employed using pre-trained weights based on the Common Objects in Context (COCO) dataset, without additional retraining on a custom dataset [32].

Figure 6 illustrates the YOLOv3 architecture and its implementation in the research for human detection, from image input to the final output. Input images with varying resolutions (160×120, 320×240, and 640×480) are first standardized to 608×608 pixels during preprocessing before being fed into the Darknet-53 backbone, which extracts hierarchical features through stacked 1×1 and 3×3 convolutional layers.

To improve detection across multiple object scales, the model applies a Feature Pyramid Network (FPN) strategy by combining low-resolution, context-rich feature maps from deeper layers with high-resolution, detail-rich maps from shallower layers. This fusion generates three detection scales (76×76, 38×38, and 19×19), each connected to a detection branch enhanced with a Dilated Attention Module (DAM). This module directs the model’s focus toward salient regions, thereby improving bounding box localization accuracy, confidence scores, and class predictions.

In the final stage, Non-Maximum Suppression (NMS) is applied to eliminate redundant bounding boxes and retain the most accurate detections. The resulting output provides visualized human bounding boxes along with confidence scores. Then, the system performance is evaluated using metrics such as confidence level [30].

III. RESULTS AND DISCUSSION

A. Implementation of Surveillance Camera Devices

The implementation of the surveillance camera device is based on the architecture described previously, with the main goal of providing a real-time and power-efficient visual surveillance solution. The device integrates several hardware components. Figure 7 shows

Cite this article as: M. I. Khalif, R. D. Mardian, A. F. K. Putra, M. D. Wicaksono, T. A. T. M. Hulu, and L. E. Prabowo, "Power-efficient surveillance camera using sleep mode and YOLOv3 model-based edge computing", CommIT Journal 20(1), 1–15, 2026.

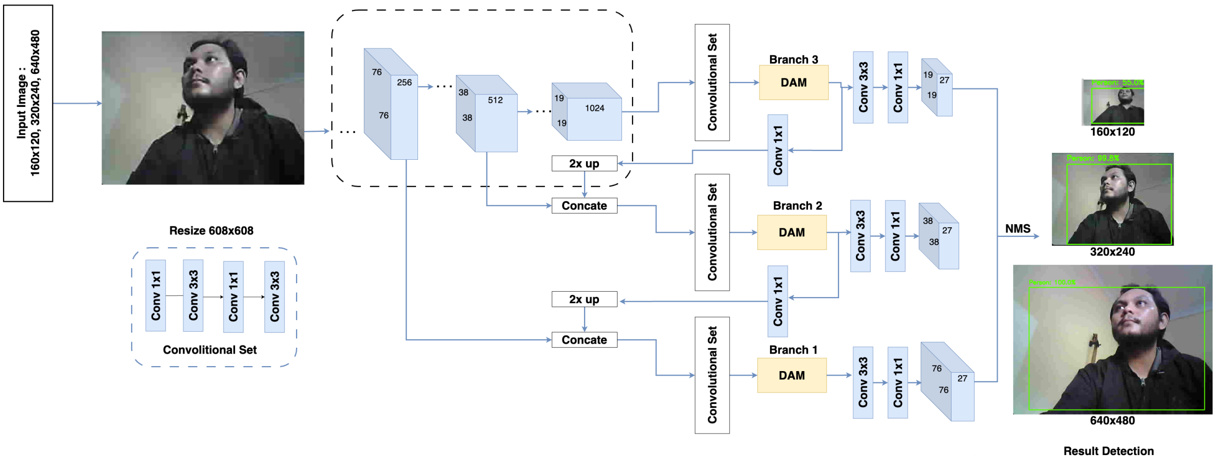

Fig. 6. Architecture and result of YOLOv3 for human detection. Note: Non-Maximum Suppression (NMS) and Dilated Attention Module (DAM).

the implementation and design results of an ESP32-CAM–based surveillance camera device. It is designed to operate independently with the support of an HFS-DC06 microwave motion sensor and a CR123 battery with a capacity of 2,650 mWh. This design focuses on energy efficiency and portability, making it suitable for use in areas that are difficult to reach with fixed electrical installations.

Figure 7(a) shows the front view of the surveillance camera device. In Fig. 7a, the ESP32-CAM module, which is the core of the system, is positioned at the top and directly connected to the HFS-DC06 microwave sensor located at the bottom. This sensor functions only when the surveillance camera device enters sleep mode, serving as a wake-up trigger.

Figure 7b shows the back of the surveillance camera device. In this section, the use of CR123 batteries as the main power source is visible. However, because the battery output voltage is only around 3.3 V, while the optimal operating voltage for the ESP32-CAM is 5 V, a step-up module is required to increase the voltage to 5 V so that it can operate optimally. In addition, this system is equipped with a TP4056 module, which functions as a battery charging circuit and allows the device to be recharged without having to replace the battery manually. This additional component is important for maintaining the sustainability of the device's operation in the long term.

Figure 7c shows the final form of the surveillance camera device after being fully assembled and fitted with a protective casing. This casing is designed not only to protect the internal components from dust, water, or physical impact, but also to make the device more compact and easier to install in various locations, such as walls, poles, or other surfaces. With a compact, modular design that supports power-saving operation, this device is ready to be used for environmental surveillance applications.

B. Evaluation of Detection and People-Counting Performance in the Surveillance Camera System

The implementation of the people detection and counting system is conducted to evaluate the performance of the ESP32-CAM-based surveillance camera in identifying human presence within the monitored area. The system employs the YOLOv3 model with the primary objective of automatically detecting and counting individuals in real time. The processing is performed on an edge computing platform running Ubuntu 24.04 on an ARM64 processor with 8 GB of RAM.

Figure 8a illustrates the application of people counting using the YOLOv3 model. The system successfully detects and counts individuals in video frames captured by a low-resolution camera (800×600 pixels) while maintaining its surveillance functionality. YOLOv3 operates as a one-stage object detection algorithm that divides the input image into grids and simultaneously predicts bounding boxes, class probabilities, and confidence scores in a single forward pass. The model employs the Darknet-53 backbone with 53 convolutional layers to extract hierarchical features and applies multi-scale detection to accurately recognize objects of varying sizes. This architecture enables YOLOv3 to achieve a balance between inference speed and detection accuracy, making it highly effective for real-time applications on resource-constrained devices.

In the research, the integration of YOLOv3 allows the ESP32-CAM not only to function as a basic video streaming device, as shown in Fig. 8b, but also to

Cite this article as: M. I. Khalif, R. D. Mardian, A. F. K. Putra, M. D. Wicaksono, T. A. T. M. Hulu, and L. E. Prabowo, “Power-efficient surveillance camera using sleep mode and YOLOv3 model-based edge computing”, CommIT Journal 20(1), 1–15, 2026.

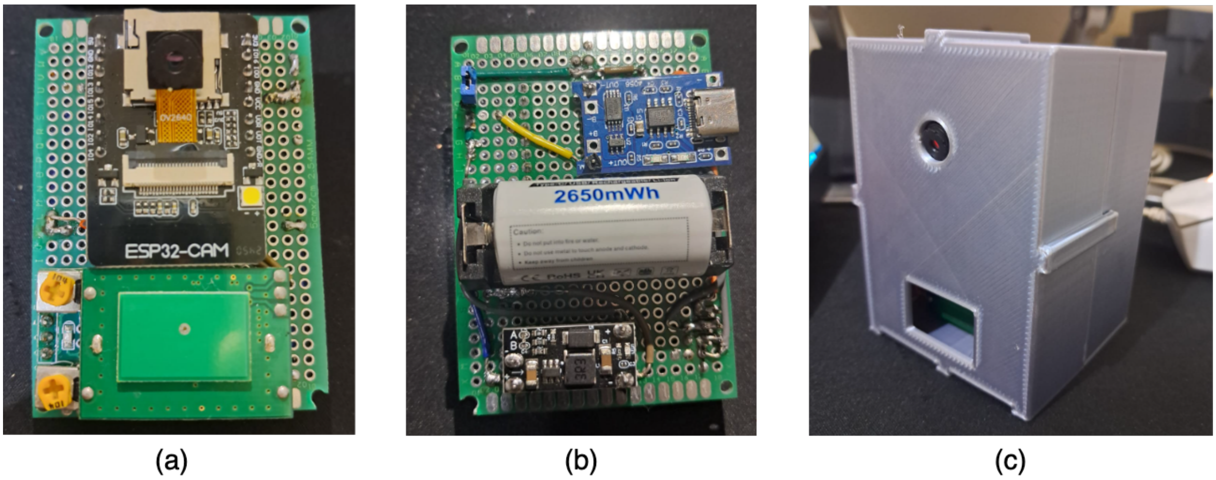

Fig. 7. Results of the implementation of surveillance camera devices.

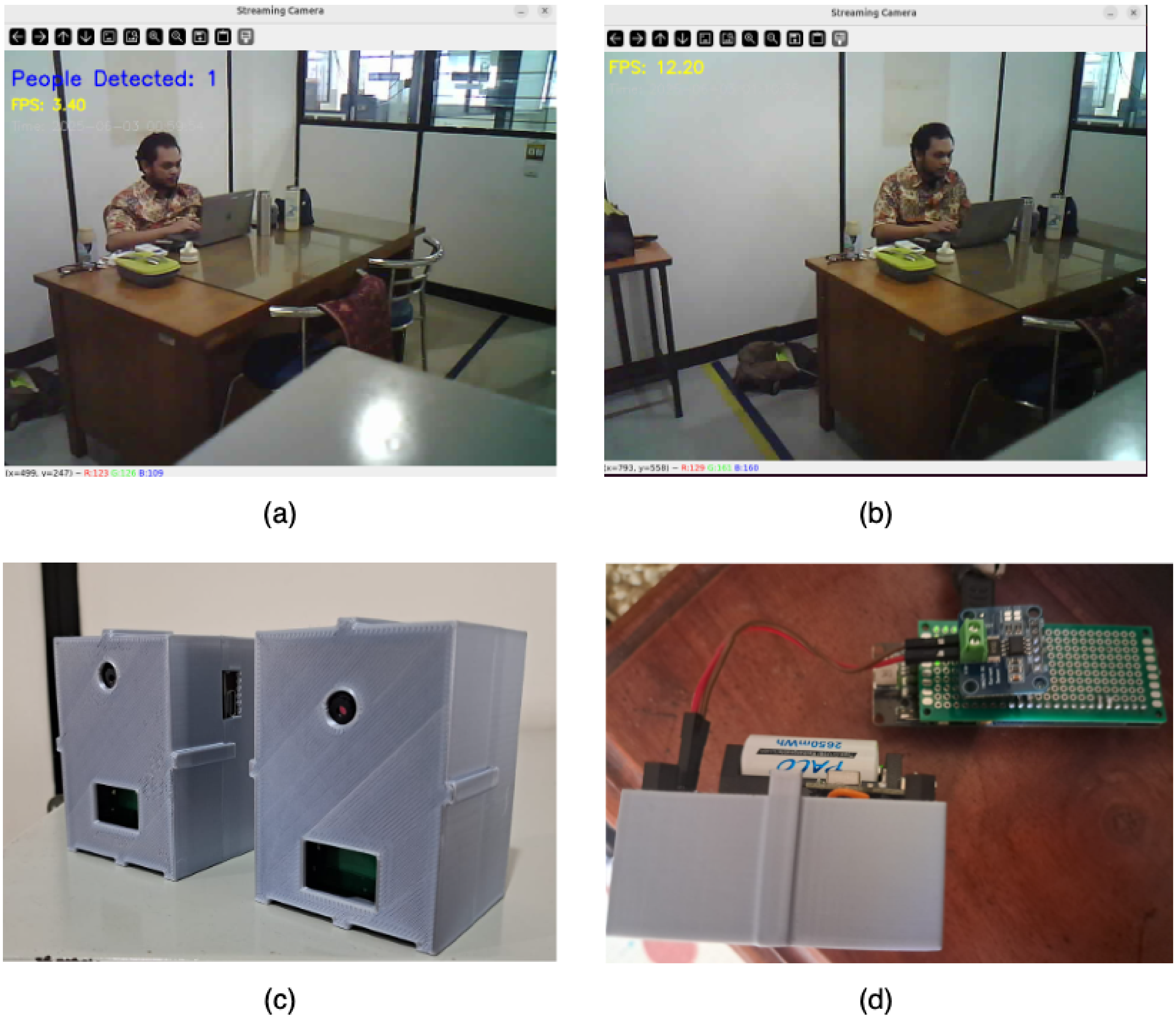

Fig. 8. Surveillance camera video streaming results and measurement configuration.

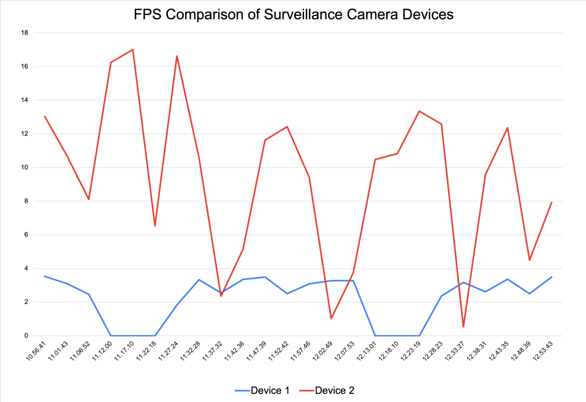

Fig. 9. FPS comparison of surveillance camera devices.

TABLE II
RESULTS OF FRAME PER SECOND (FPS) COMPARISON OF SURVEILLANCE CAMERA DEVICES.

Variable	Device 1	Device 2
Average	2.222	9.449
Max.	3.53	17.01
Min.	0	0.53

intelligently detect and count individuals, thereby directly supporting the implementation of sleep mode for energy saving. Figure 8c presents the device and testing configuration. Meanwhile, Fig. 8d shows the power measurement setup, in which a meter is connected in series to record current and voltage consumption during operation. Consequently, the application of YOLOv3 is central to this research, as it ensures reliable human detection while maintaining the system’s power-efficient performance.

Performance testing is carried out using two surveillance camera units that transmit image data to edge computing via a local network. The test is also configured in two different scenarios. The first camera is integrated with the YOLOv3 model to detect and count human objects, while the second camera performs only basic streaming functions without additional processing. Both cameras are tested under identical environmental conditions and hardware specifications. The purpose of this configuration is to conduct a comparative analysis of system performance, particularly in terms of FPS, to evaluate the impact of model integration on real-time surveillance performance.

Figure 9 shows the FPS measurement results for Device 1 and Device 2. The measurements are conducted for approximately two hours for each device, with sampling performing every five minutes. FPS is a parameter used to measure the number of frames or images displayed per second in a video or visual system, where a higher frame rate generally results in smoother motion and improved perceived visual quality, especially for dynamic content. Higher frame rates such as 60 FPS have been shown to produce better motion smoothness and viewer preference compared to lower rates like 30 FPS [33]. Additionally, frame rate has been identified as a key factor influencing the robustness of video-based human action recognition, highlighting its importance in real-time monitoring and analysis systems [34].

Table II shows the average, maximum, and minimum FPS values obtained from Device 1 and Device 2. For Device 1, the minimum FPS value is 0, which occurs when the device enters sleep mode and no image frames are processed. For Device 2, the minimum FPS value is 0.53, because Device 2 does not implement sleep mode. Therefore, the camera remains active and continues capturing images.

Based on the measurement results for Device 1, which detects and counts people using the YOLOv3 model, the average FPS obtained is 2.222, with a maximum FPS of 3.53. This result is much lower than Device 2, which performs only the streaming function, with an average FPS of 9.449 and a maximum of 17.01. The lower FPS on Device 1 is expected because it performs real-time object detection, which requires intensive computation for each video frame. This result highlights the trade-off between data analysis accuracy and streaming speed. In other words, the more complex the processing task (such as YOLOv3 for counting people), the higher the computational load on the device, resulting in a lower FPS.

Furthermore, this difference emphasizes the importance of optimization on the edge computing side. For example, using a lighter model (such as YOLOv3-tiny or YOLOv5-nano) or implementing selective frame processing can increase FPS without significantly sacrificing accuracy. This analysis also shows that devices performing only streaming are more efficient in real-time performance but do not provide additional information, such as the number of detected people, which is critical for energy-saving surveillance systems.

Nevertheless, in the context of power-efficient deployment, Device 1 demonstrates clear advantages, as the entire object detection and classification workflow is executed locally on the device, eliminating the need for continuous data transmission to external servers. This local processing approach significantly reduces both energy consumption and bandwidth usage, which is critical for surveillance camera systems deployed in resource-constrained or remote environments. Although Device 1 exhibits lower FPS performance, it remains functionally adequate for static or semi-real-time monitoring scenarios, where system efficiency and

Cite this article as: M. I. Khalif, R. D. Mardian, A. F. K. Putra, M. D. Wicaksono, T. A. T. M. Hulu, and L. E. Prabowo, "Power-efficient surveillance camera using sleep mode and YOLOv3 model-based edge computing", CommIT Journal 20(1), 1–15, 2026.

TABLE III
RESULTS OF CAMERA SURVEILLANCE RESPONSE TIME AFTER SLEEP MODE.

Variable	Connection Duration (ms)	Wi-Fi Connection Duration (ms)	Wake-Up Duration (ms)
Total	4,560,000	33,945	1,122
Average	190,000	1,414	46
Max	309,000	1,488	47
Min	0	1272	46

Note: All duration values are measured in milliseconds (ms).

sustainable operation take precedence over throughput.

C. Evaluation of Surveillance Cameras' Response Time After Sleep Mode

The measurement of response time in surveillance camera devices aims to determine the duration required for the system to wake from deep sleep mode and reconnect to a Wi-Fi network. This process begins by recording the system time using the `millis()` function immediately after the device exits deep sleep mode and ends when the Wi-Fi connection status indicates a successful connection. The difference between these two timestamps represents the connection duration, measured in milliseconds.

This duration, commonly referred to as wake-up time, is a critical performance metric, especially for energy-efficient IoT devices. A shorter wake-up time implies higher system responsiveness and reduced energy consumption. Therefore, accurately measuring this time is essential in designing low-power IoT-based monitoring systems that rely on sleep–wake cycles.

Table III presents the processed measurement results of the surveillance camera devices' readiness time after waking from sleep mode and becoming accessible again. Three performance parameters are evaluated. The first parameter is the duration during which the device remains active and connected to the Wi-Fi network. The total recorded duration is 4,560,000 ms, equivalent to approximately 1 hour and 16 minutes, with a minimum value of 0 ms corresponding to the period when the device is in sleep mode. The second parameter is the time required for the device to reconnect to the Wi-Fi network after waking from sleep mode. The average reconnection time is 1,414 ms, equivalent to approximately 1 second and 414 milliseconds. Although no strict standard governs Wi-Fi reconnection time, industry practices indicate that connection times below 2 seconds are generally considered efficient and responsive, particularly in scenarios requiring fast roaming and smooth user experience. The third parameter is the wake-up time from sleep mode. The average recorded wake-up time is 46.72 ms, indicating rapid system responsiveness after exiting sleep mode.

D. Comparison of Power Consumption on Surveillance Camera Devices

This section presents the results of an experimental evaluation comparing the power consumption of surveillance camera devices under two operating conditions: with and without the implementation of sleep mode. The primary objective of this evaluation is to examine the extent to which sleep mode contributes to energy savings and effective power management during video streaming, object detection, and person counting using the YOLOv3 algorithm. These findings underscore the importance of integrating sleep mode-based power management with optimized detection models to develop energy-efficient surveillance camera systems.

Figure 10 presents a comparison graph of power consumption for the two test scenarios of the surveillance camera devices. Each is measured for approximately two hours under identical environmental conditions. Device 1 is a surveillance camera device that implements sleep mode for power saving and management. Meanwhile, Device 2 operates without sleep mode. In the power usage test, three parameters are measured: current (mA), voltage (V), and power consumption (mW) for each measurement scenario. From Fig. 10c, it can be observed that Device 1 exhibits lower power consumption compared to Device 2. The test results demonstrate that the implementation of sleep mode effectively reduces and manages power usage compared to operation without sleep mode.

Table IV shows the results of processing the power consumption value during the measurement. On Device 1, the total current used for approximately two hours of measurement is 3,927.8 mA, which is smaller than the current usage of Device 2 of 5,766.2 mA. Hence, the total power consumption during the measurement for Device 1 is smaller by 3475.543 mW compared to Device 2 of 5561.639 mW. It can save power usage by approximately 37.5%. However, power savings in the application of sleep mode depend on the activity of the person recorded in the image frame. If the activity is always active 24 hours, the monitoring camera device remains active 24 hours too.

IV. CONCLUSION

The research develops a low-power surveillance camera system with a sleep mode mechanism that maintains monitoring functionality while improving energy efficiency. The system with sleep mode achieves 3.53 FPS, which is lower than the 17.01 FPS

Cite this article as: M. I. Khalif, R. D. Mardian, A. F. K. Putra, M. D. Wicaksono, T. A. T. M. Hulu, and L. E. Prabowo, “Power-efficient surveillance camera using sleep mode and YOLOv3 model-based edge computing”, CommIT Journal 20(1), 1–15, 2026.

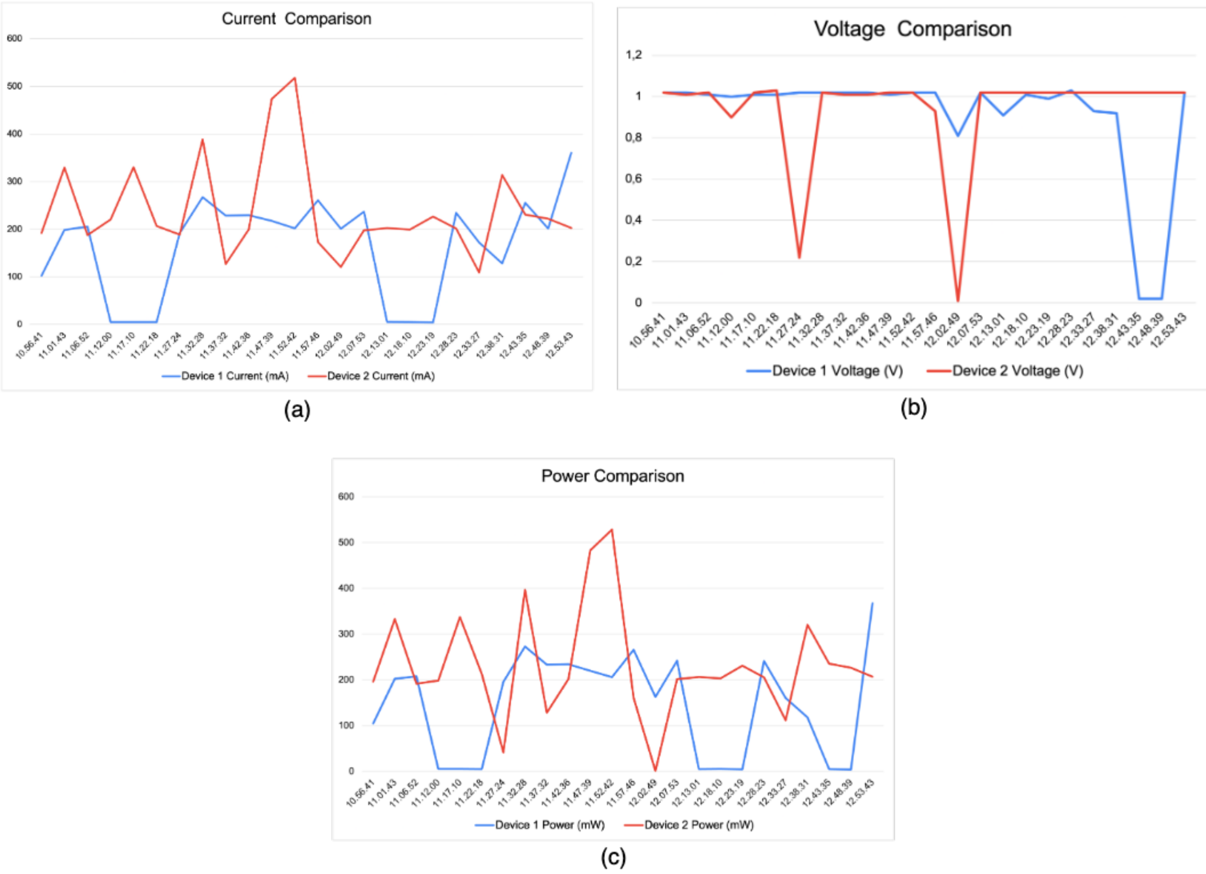

Fig. 10. Comparison of power consumption of surveillance camera devices.

TABLE IV
POWER CONSUMPTION COMPARISON OF SURVEILLANCE CAMERA DEVICES.

Variable	Device 1			Device 2		
	Current (mA)	Voltage (V)	Power (mW)	Current (mA)	Voltage (V)	Power (mW)
Total	3,927.8	21.88	3,475.543	5,766.2	22.44	5561.639
Average	163.658	0.9117	144.8143	240.258	0.935	231.735
Max.	360.5	1.03	367.71	518.1	1.03	528.462
Min.	4.7	0.02	4.036	109.6	0.01	1.21

of the non-sleep version, but it remains effective for real-time monitoring. The average wake-up time is 46 ms, with a maximum of 1.4 seconds, and power consumption is reduced by approximately 37.5% over a two-hour period.

The use of YOLOv3 enables real-time person detection and counting on low-resolution cameras with high accuracy, providing critical information for intelligent energy management, such as activating sleep mode. Without YOLOv3, the system can only stream video at a higher FPS, resulting in smoother playback but lacking the ability to monitor occupancy or optimize energy usage. This result highlights the trade-off between intelligent monitoring capabilities and raw streaming performance.

Several limitations should be noted. The experiments are conducted in a local network environment using ESP32-CAM hardware under relatively static indoor conditions for two hours. Moreover, the system is tested with a limited number of subjects and simple scenarios, which may not fully represent real-world environments with dynamic lighting, crowded scenes, or outdoor conditions. Furthermore, the YOLOv3 model is not retrained on datasets collected from the system itself, which may limit its detection performance in the specific monitored environments.

Future research should explore higher-specification camera modules (e.g., ESP-EYE or TTGO T-Camera

Plus) advanced edge computing platforms (e.g., NVIDIA Jetson), more realistic and diverse scenarios, longer-duration experiments, and model retraining on locally collected datasets. These improvements can enhance detection accuracy, maintain real-time performance, and further optimize energy efficiency. Hence, it can support continuous and power-efficient monitoring solutions.

ACKNOWLEDGEMENT

This research was supported by a grant from Universitas Trisakti through the Beginner Lecturer Research (PDP) scheme, from 2024 to 2025 (number: 0965/PDP/FTI/2024-2025). The authors would like to thank the Institute for Research and Community Service (LPPM) of Universitas Trisakti, the Faculty of Industrial Technology (DRPMF), and the Department of Electrical Engineering of Universitas Trisakti for providing grants and support to this research.

AUTHOR CONTRIBUTION

Conceived and designed the analysis with machine learning model testing, M. I. K.; Collected the data for accuracy and power consumption values, M. I. K.; Contributed data or analysis tools for accuracy and power consumption values, M. I. K.; Performed the analysis for accuracy and power consumption values, M. I. K. and R. D. M.; Wrote the paper, M. I. K. and R. D. M.; Trained and tested machine learning models, A. F. K. P. and M. D. W.; and Formatted the paper and checked the grammar and language, T. A. T. M. H. and L. E. P.

DATA AVAILABILITY

The measurement data supporting the findings of this study are available from the corresponding author (Mhd. Idham Khalif) upon reasonable request, as the raw experimental data contain device-specific configurations and system logs that are not publicly archived.

REFERENCES

[1] F. A. Alenizi, S. Abbasi, A. H. Mohammed, and A. M. Rahmani, "The artificial intelligence technologies in Industry 4.0: A taxonomy, approaches, and future directions," *Computers & Industrial Engineering*, vol. 185, 2023.

[2] BPS - Statistics Indonesia, "Statistical data stories for Indonesia – Digital transformation: Exploring the association of digitalization with education and health," 2025. [Online]. Available: https://bit.ly/3MnVInv

[3] K. Aminiyeganeh, R. W. L. Coutinho, and A. Boukerche, "IoT video analytics for surveillance-based systems in smart cities," *Computer Communications*, vol. 224, pp. 95–105, 2024.

[4] PT PLN (Persero), "Statistik PLN 2023," 2024. [Online]. Available: https://web.pln.co.id/statics/uploads/2024/07/Laporan-Statistik-2023-Ind.pdf

[5] M. I. Khalif and A. Muis, "Fuzzy logic optimization to control Air Conditioner (AC) conditions using rule-based algorithm," *ELKHA: Jurnal Teknik Elektro*, vol. 17, no. 1, pp. 38–45, 2025.

[6] President of the Republic of Indonesia, "Presidential regulation 70/2009 concerning energy conservation," 2009. [Online]. Available: https://bit.ly/4at3GF6

[7] C. L. Kok, J. B. Heng, Y. Y. Koh, and T. H. Teo, "Energy-, cost-, and resource-efficient IoT hazard detection system with adaptive monitoring," *Sensors*, vol. 25, no. 6, pp. 1–34, 2025.

[8] C. K. Rao, S. K. Sahoo, and F. F. Yanine, "An Internet of things–based intelligent smart energy monitoring system for solar photovoltaic applications," in *Performance enhancement and control of photovoltaic systems*. Elsevier, 2024, pp. 375–416.

[9] M. Giordano, N. Baumann, M. Crabolu, R. Fischer, G. Bellusci, and M. Magno, "Design and performance evaluation of an ultralow-power smart IoT device with embedded TinyML for asset activity monitoring," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[10] H. Rajab, H. Al-Amaireh, T. Bouguera, and T. Cinkler, "Evaluation of energy consumption of LPWAN technologies," *EURASIP Journal on Wireless Communications and Networking*, vol. 2023, pp. 1–26, 2023.

[11] S. Kim, C. Kim, and S. Kim, "Improving performance of real-time object detection in edge device through concurrent multi-frame processing," *IEEE Access*, vol. 13, pp. 1522–1533, 2024.

[12] R. Al Amin, M. Hasan, V. Wiese, and R. Obermaisser, "FPGA-based real-time object detection and classification system using YOLO for edge computing," *IEEE Access*, vol. 12, pp. 73 268–73 278, 2024.

[13] W. Qian, Z. Zhu, C. Zhu, and Y. Zhu, "FPGA-based accelerator for YOLOv5 object detection with optimized computation and data access for edge deployment," *Parallel Computing*, vol. 124, 2025.

[14] A. Upadhyay, G. Sunil, S. Das, J. Mettler, K. Howatt, and X. Sun, "Multiclass weed and

crop detection using optimized YOLO models on edge devices," *Journal of Agriculture and Food Research*, vol. 22, pp. 1–14, 2025.

[15] D. Syed, S. S. Refaat, and H. Abu-Rub, "Performance evaluation of distributed machine learning for load forecasting in smart grids," in *2020 Cybernetics & Informatics (K&I)*. Velke Karlovice, Czech Republic: IEEE, Jan. 29–Feb. 01, 2020, pp. 1–6.

[16] H. H. Nguyen, T. N. Ta, N. C. Nguyen, V. T. Bui, H. M. Pham, and D. M. Nguyen, "YOLO based real-time human detection for smart video surveillance at the edge," in *2020 IEEE Eighth International Conference on Communications and Electronics (ICCE)*. Phu Quoc Island, Vietnam: IEEE, Jan. 13–15, 2021, pp. 439–444.

[17] Y. H. Chang, F. C. Wu, and H. W. Lin, "Design and implementation of ESP32-based edge computing for object detection," *Sensors*, vol. 25, no. 6, pp. 1–26, 2025.

[18] F. Xu, L. Huang, X. Gao, T. Yu, and L. Zhang, "Research on YOLOv3 model compression strategy for UAV deployment," *Cognitive Robotics*, vol. 4, pp. 8–18, 2024.

[19] J. Kim and S. Ha, "Energy-aware scenario-based mapping of deep learning applications onto heterogeneous processors under real-time constraints," *IEEE Transactions on Computers*, vol. 72, no. 6, pp. 1666–1680, 2022.

[20] K. Dokic, "Microcontrollers on the edge–Is ESP32 with camera ready for machine learning?" in *International Conference on Image and Signal Processing*. Dubai, UAE: Springer International Publishing, June 20–21, 2020, pp. 213–220.

[21] Espressif Systems, "Sleep modes." [Online]. Available: https://docs.espressif.com/projects/esp-idf/en/stable/esp32/api-reference/system/sleep_modes.html

[22] P. Hobden, E. Nurellari, and S. Srivastava, "RF-SoC softwarisation of a 2.45 GHz doppler microwave radar motion sensor," *Journal of Sensor and Actuator Networks*, vol. 13, no. 5, pp. 1–27, 2024.

[23] K. Gebru, M. Rapelli, R. Rusca, C. Casetti, C. F. Chiasserini, and P. Giaccone, "Edge-based passive crowd monitoring through WiFi beacons," *Computer Communications*, vol. 192, pp. 163–170, 2022.

[24] T. C. Shen and E. T. H. Chu, "Edge-computing-based people-counting system for elevators using MobileNet–single-stage object detection," *Future Internet*, vol. 15, no. 10, pp. 1–21, 2023.

[25] M. A. Khan, H. Menouar, and R. Hamila, "LCDnet: A lightweight crowd density estimation model for real-time video surveillance," *Journal of Real-Time Image Processing*, vol. 20, no. 2, pp. 1–11, 2023.

[26] M. Katsigiannis and K. Mykoniatis, "Enhancing industrial IoT with edge computing and computer vision: An analog gauge visual digitization approach," *Manufacturing Letters*, vol. 41, pp. 1264–1273, 2024.

[27] C. F. Liao and Y. J. Weng, "Enabling space-aware service discovery model in home networks through a compatible extension to mDNS/DNS-SD," *Electronics*, vol. 12, no. 18, pp. 1–17, 2023.

[28] Espressif Systems, "ESP32 low-power management." [Online]. Available: https://docs.espressif.com/projects/esp-iot-solution/en/latest/low_power_solution/esp32_lowpower_solution.html

[29] H. Wu, C. Chen, and K. Weng, "An energy-efficient strategy for microcontrollers," *Applied Sciences*, vol. 11, no. 6, pp. 1–18, 2021.

[30] A. Vijayakumar and S. Vairavasundaram, "YOLO-based object detection models: A review and its applications," *Multimedia Tools and Applications*, vol. 83, no. 35, pp. 83 535–83 574, 2024.

[31] Y. Zhou, H. Qian, and P. Ding, "Lite-YOLOv3: A real-time object detector based on multi-scale slice depthwise convolution and lightweight attention mechanism," *Journal of Real-Time Image Processing*, vol. 20, 2023.

[32] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," in *European Conference on Computer Vision*. Zurich, Switzerland: Springer, Sep. 6–12, 2014, pp. 740–755.

[33] K. Debattista, K. Bugeja, S. Spina, T. Bashford-Rogers, and V. Hulusic, "Frame rate vs resolution: A subjective evaluation of spatiotemporal perceived quality under varying computational budgets," *Computer Graphics Forum*, vol. 37, no. 1, pp. 363–374, 2018.

[34] F. Harjanto, Z. Wang, S. Lu, A. C. Tsoi, and D. D. Feng, "Investigating the impact of frame rate towards robust human action recognition," *Signal Processing*, vol. 124, pp. 220–232, 2016.