Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 3, Maret 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i3.9619

Edge-Based Person Detection Using MobileNetV2 on ESP32-CAM for Home Surveillance System

Adi Purnama[1,*], Indriani[1], Bagus Alit Prasetyo[2], Agitama Nugraha[1]
[1] Faculty of Engineering, Department of Informatics Engineering, Widyatama University, Bandung, Indonesia
[2] Faculty of Engineering, Department of Electrical Engineering, Widyatama University, Bandung, Indonesia
Email: [1,*]adi.purnama@widyatama.ac.id, [2]indriani.st@widyatama.ac.id, [3]alit.prasetyo@widyatama.ac.id, [4]nugraha.agitama@widyatama.ac.id
Correspondence Author Email: adi.purnama@widyatama.ac.id[*]
Submitted: **25/02/2026**; Accepted: **30/03/2026**; Published: **31/03/2026**

Abstract− Affordable home surveillance systems increasingly require on-device intelligence to mitigate privacy risks, network dependency, and response latency associated with cloud-based video analytics. This research develops an edge-based person detection system utilizing a quantized MobileNetV2 architecture (α = 0.1) deployed on an ESP32-CAM module, integrated with the Message Queuing Telemetry Transport (MQTT) protocol for real-time alert delivery. To construct the dataset, image data were collected via a motion-triggered acquisition setup employing a passive infrared (PIR) sensor, resulting in a total of 500 manually labeled images categorized into person and non-person classes. The model was trained in over 40 epochs with a learning rate of 0.001, utilizing data augmentation and INT8 quantization to optimize embedded deployment. Performance was evaluated using a 23% testing split (116 unseen images), yielding an overall accuracy of 78.45%. For the person class, the model achieved 90.00% precision, 77.78% recall, and an 83.44% F1-score. On-device deployment via TensorFlow Lite required a peak RAM of 485.4 KB and 102.1 KB of flash memory. The average inference time was recorded at 1018 ms per frame, which limits continuous high-framerate processing but remains feasible for basic surveillance at approximately 1 frame per second. Finally, MQTT communication via an EMQX broker successfully transmitted detection alerts and image links to a mobile application for real-time monitoring and storage.

Keywords: MobileNetV2; ESP32-CAM; Person Detection; MQTT; Home Surveillance System

1. INTRODUCTION

Technological advancements in the modern era have significantly reshaped daily life, including how individuals and communities protect their homes and property. Despite these technological improvements, criminal activity in Indonesia continues to become more complex, demonstrating increasingly dynamic and unpredictable patterns. Numerous studies have identified strong correlations between socioeconomic pressures, such as income inequality, rapid urbanization, and unemployment, and the rising incidence of robbery, theft, and fraud in different regions of Indonesia [1], [2]. Meanwhile, the rapid proliferation of Internet of Things (IoT)-enabled devices and smart home ecosystems has introduced additional vulnerabilities into residential environments. These vulnerabilities include unauthorized access to connected systems, weaknesses in firmware security, insufficient encryption mechanisms, and fragile network interfaces that sophisticated attackers may exploit for malicious purposes [3], [4]. These conditions highlight the inadequacy of traditional security infrastructures, which often lack the intelligence, adaptability, and responsiveness required to counter modern threats. Consequently, there is an urgent and growing need for intelligent, affordable home security solutions that can provide continuous protection with minimal reliance on external, cloud-based infrastructures or human intervention [5].

Edge-based surveillance has emerged as a promising technological solution for modern home security needs. The increasing demand for cost-effective, privacy-preserving, low-latency surveillance systems has prompted researchers and practitioners to develop person detection models that operate directly on embedded hardware platforms. Unlike conventional camera systems, which continuously transmit video streams to remote servers for processing, edge-based systems offer clear advantages. Privacy risks are minimized, network congestion is reduced, and the need for stable internet connectivity is eliminated. Workflows for surveillance that are based in the cloud often result in significant delays and create bottlenecks, especially during periods of high traffic. This can delay or disrupt timely responses to security incidents [6]. Furthermore, many households and small-scale installations depend on low-cost devices with strict limitations in memory and processing power. Examples include the ESP32-CAM module. However, these installations still require reliable person-detection performance [7].

A person-detection framework based on the lightweight MobileNetV2 architecture and executed on an embedded platform, such as the ESP32-CAM, is a feasible and effective solution to these challenges. MobileNetV2 is designed to balance computational efficiency with recognition accuracy, making it particularly suitable for deployment in resource-constrained embedded systems such as the ESP32-CAM [8]. Theh proper optimization, the model can perform real-time or near-real-time inference while operating within the limited memory and computational budgets of ESP32-class hardware. Local execution of inference tasks strengthens data privacy by preventing unnecessary external data transmissions and substantially reduces bandwidth consumption, improving

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 3, Maret 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i3.9619

system responsiveness. The deployment process often incorporates model compression techniques and runtime optimizations, such as quantization, pruning, and architectural adaptations, to enhance performance under constrained operating conditions [9], [10].

Compared to other commonly used deep learning models for object detection, MobileNetV2 strikes a better balance between accuracy and computational efficiency for embedded systems. Heavier architectures, such as Faster R-CNN and YOLOv5, generally provide a higher level of detection accuracy, but they require substantial computational resources and memory. This makes them impractical for deployment on microcontroller-based platforms, such as the ESP32-CAM [11]. In contrast, lightweight models such as MobileNetV2 and Tiny-YOLO are designed specifically for edge AI applications with limited processing power and energy constraints. Recent studies indicate that MobileNetV2 achieves competitive performance while maintaining a significantly smaller model size and lower inference latency. These factors are critical for real-time surveillance systems with limited hardware resources [12], [13]. Studies in the domain of TinyML demonstrate that MobileNet-based frameworks are among the most effective solutions for implementing computer vision tasks on low-power devices, even with a minor compromise in precision [14], [15]. Therefore, MobileNetV2 was selected for this study as it strikes an optimal balance between performance and deployability for ESP32-CAM-based home surveillance systems [15], [16].

Recent studies have demonstrated the suitability of MobileNetV2-based detectors for low-power edge devices when supported by optimization techniques such as quantization. For instance, one study implemented an 8-bit quantized SSD MobileNetV2 model, demonstrating significant improvements in inference speed and memory efficiency on edge hardware. This model achieved real-time performance with latencies of 5 ms on Raspberry Pi 5 and 85 ms on Jetson Orin Nano while maintaining 80.65% accuracy and an F1 score of 0.92 for masked-face detection [17]. Additionally, a systematic review of TinyML-based person detection revealed that lightweight architectures, such as MobileNet and Tiny-YOLO, consistently offer the optimal balance of accuracy, latency, and energy usage for microcontrollers and compact edge platforms. The review also emphasized the importance of optimization methods, including quantization, pruning, and knowledge distillation, for enabling reliable, real-time detection on hardware such as ARM Cortex-M, ESP32, and STM32 devices [15]. These findings show that MobileNetV2 is a good model for person detection on home surveillance systems that don't have a lot of resources.

In modern surveillance architectures, communication protocols are essential for ensuring the timely and reliable delivery of security-related notifications, especially in systems operating on resource-constrained embedded devices. MQTT is one of the most efficient solutions for real-time monitoring applications due to its lightweight design, low bandwidth usage, and suitability for event-driven messaging in IoT environments. Leveraging these capabilities is essential in-home surveillance systems, where the rapid transmission of alerts can significantly enhance responsiveness and safety. This study presents a home surveillance system integrating the MobileNetV2 model for person detection with the ESP32-CAM platform. The system incorporates the MQTT protocol to enhance the ESP32-CAM's capabilities by enabling automatic delivery of detection-based notifications. MobileNetV2 is recognized as a compact yet reliable architecture that can perform high-accuracy image classification tasks on embedded and IoT-oriented platforms [8], [18], [19].

The proposed system integrates embedded computer vision with lightweight IoT communication, enabling a cohesive and efficient home-surveillance workflow. At the core of the system, MobileNetV2 is deployed on the ESP32-CAM to perform person detection directly on the device, ensuring that critical visual analysis occurs locally without relying on remote servers [20]. This local processing capability is complemented by the MQTT protocol, which functions as the system's communication backbone by transmitting detection events to subscriber clients in real time. MQTT's publish–subscribe mechanism allows alerts to be delivered quickly and reliably, even under low-bandwidth or unstable network conditions, making it well suited for residential environments where connectivity may vary [21]. Through this combination of embedded vision and lightweight messaging, the system is designed to provide continuous monitoring while maintaining efficiency, responsiveness, and ease of deployment.

This research aims to develop and evaluate an edge-based person detection system that leverages MobileNetV2's efficiency and the ESP32-CAM hardware platform's responsiveness to MQTT-driven notifications. The study will systematically assess the model's detection accuracy and demonstrate its integration into a fully functional home surveillance workflow. The goal is to deliver an effective, low-cost, privacy-preserving security solution suitable for widespread adoption. Ultimately, the findings are expected to contribute to the advancement of intelligent surveillance technologies, promote the broader utilization of edged AI methodologies, and support the development of real-time monitoring systems that are accessible and reliable in residential environments. By bridging the gap between sophisticated AI architectures and resource-constrained hardware, this work provides a scalable framework for localized security that prioritizes user data autonomy.

2. RESEARCH METHODOLOGY

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 3, Maret 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i3.9619

This research uses a combination of a quantitative experimental method and a system development approach. The system development approach emphasizes numerical measurement, controlled testing, and objective evaluation of system performance. This approach is commonly used in engineering and AI-based system validation [22], [23], [24]. The strategy is divided into multiple phases (see Figure 1), commencing with Research Planning, where the issue is pinpointed and system requirements are reviewed. The data and modeling stage involves collecting and labeling image data, training the MobileNetV2 model, and interpreting the results to verify its suitability for deployment. In the System Implementation stage, the optimized model is deployed to the ESP32-CAM device and integrated with MQTT to enable real-time event notifications. The final stage, Evaluation and Validation, includes evaluating model accuracy, conducting functional testing under real-world scenarios, and validating the system to ensure that the developed solution meets the intended objectives of an efficient, reliable home surveillance system.

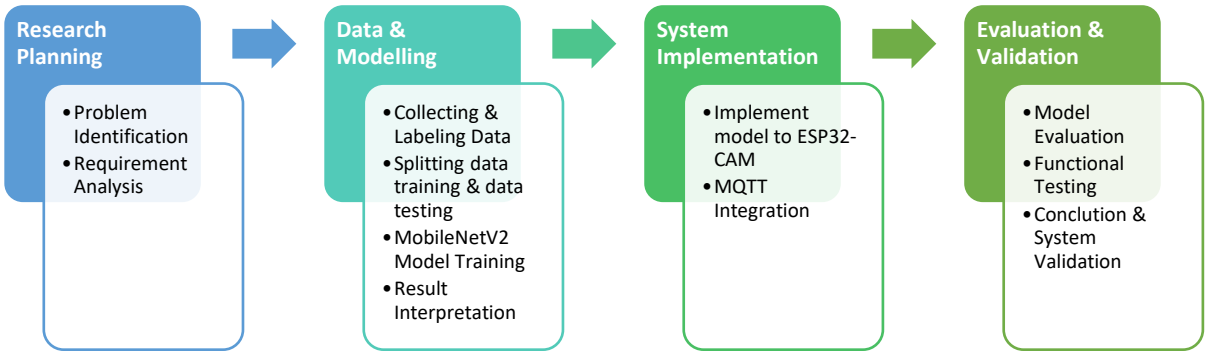

Figure 1. Research Stage

2.1 Research Planning

a. Problem Identification

The problem identification phase outlines the key issues that make the proposed home surveillance system necessary. Traditional security setups rely heavily on cloud-based processing, resulting in latency, increased bandwidth consumption, and limited real-time responsiveness. Also, continuous video transmission creates significant privacy risks for users [6]. Furthermore, low-cost devices such as the ESP32-CAM lack efficient built-in on-device person detection capabilities [20]. Meanwhile, criminal activity is becoming more complex, and the widespread adoption of IoT devices is creating new security vulnerabilities. These issues underscore the necessity of a lightweight, autonomous, and privacy-preserving person detection system that can operate effectively on resource-constrained hardware [5].

b. Requirement Analysis

The requirements analysis phase aims to identify the technical, functional, and operational needs for designing an ESP32-CAM–based intrusion detection system. This phase includes examining the ESP32-CAM's hardware limitations, selecting MobileNetV2 as the lightweight detection model, and adopting MQTT for real-time alert delivery. The detailed specifications of these components are summarized in Table 1. The system must continuously capture image frames from the ESP32-CAM, perform on-device person detection using the MobileNetV2 model, and send real-time alerts via MQTT whenever a person is detected. The system should process frames with minimal delay, maintain accurate classification under varying conditions, and ensure that users or servers subscribed to the MQTT broker receive instant notifications. These requirements establish a clear foundation to guide the development process toward achieving an efficient, reliable intrusion detection system.

Table 1. Technical Requirements

Component / Item	Specification	Function / Purpose
ESP32-CAM V3	32-bit MCU, ~520 KB RAM, limited storage, no GPU	Serves as the main processing unit responsible for running MobileNetV2 inference, capturing images, and transmitting detection results [25].
OV2640 Camera Module	Up to 2 MP resolution	Captures image frames used as input for the on-device person detection model [25].
Wi-Fi Connectivity	2.4 GHz Wi-Fi (built into ESP32)	Enables communication between the device and MQTT broker for sending real-time alerts [16].
HC-SR501 PIR Sensor	Detection Range 3m to 7m and Angle approximately 100-120 degrees cone	A motion sensor to detect objects and camera will capture images during the data collection phase [26].

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 3, Maret 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i3.9619

Component / Item	Specification	Function / Purpose
Power Supply (5V)	Stable 5V/2A recommended	Provides power to the ESP32-CAM and ensures consistent operation during image capture and inference [25].
MobileNetV2 Model	Lightweight CNN architecture	Performs low-latency, on-device person detection suitable for limited hardware resources [8].
MQTT Protocol	Lightweights publish-subscribe protocol	Facilitates efficient, low-bandwidth transmission of detection events to the server or user application [21].
MQTT Broker (EMQX)	Local or cloud-based broker	Receives published detection messages and distributes them to subscribed clients [21], [27].
MicroSD Card	FAT32 storage support	Stores captured images or logs locally into ESP32-CAM V3 [28].

2.2 Data & Modelling

a. Collecting and Labeling Data

Data collection is carried out automatically by designing a system that can detect moving objects. The system integrates an ESP32-CAM with an HC-SR501 PIR (passive infrared) motion sensor that detects changes in infrared radiation caused by human movement. Whenever motion is detected, the ESP32-CAM automatically captures and stores an image on an SD card. This ensures that the data is collected under real operational conditions that accurately represent "person" class scenarios [26]. The HC-SR501 module faces the target area and is connected to the ESP32-CAM via its digital output pin (see Figure 2). This connection enables the microcontroller to trigger the camera capture process based on sensor activation. This configuration enables the system to autonomously acquire images of human presence in a practical home surveillance setting.

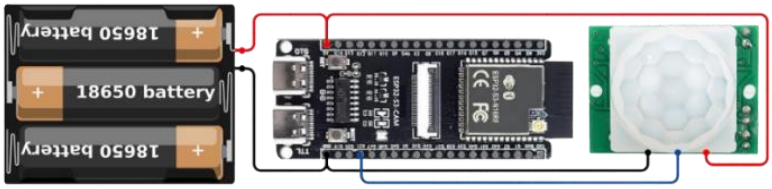

Figure 2. PIR Sensor Circuit Design for Collecting “Person” Image

After collection, all captured images are manually reviewed and labeled as either "person" or "non-person" to ensure correct ground-truth classification for training. Basic preprocessing steps, such as resizing into 64x 64 pixel and normalization, are applied to prepare the dataset for MobileNetV2. A total of 500 images were collected through this motion-triggered acquisition process.

b. Splitting data training & data testing

At this stage, a data disclosure strategy was implemented using the hold-out validation method to ensure the objectivity of the evaluation model [29]. The entire dataset (consisting of 500 samples) was divided into two main categories: the training set (77% of the dataset, or 384 samples) and the testing set (23% of the dataset, or 116 samples). This division was carried out at the beginning of the data acquisition process to ensure that the testing data remained stored and was never accessed by the model during the learning phase. Furthermore, during the training process, the training set was divided internally into a validation set using an 80:20 ratio. This was done to combine the loss and accuracy values at each epoch, thereby mitigating the risk of overfitting and enabling the model to generalize well to new data.

c. MobileNetV2 Model Training

MobileNetV2 is a lightweight convolutional neural network (CNN) architecture designed for efficient deployment on resource-constrained devices, such as microcontrollers and embedded vision platforms [30]. As seen in Figure 3, MobileNetV2 uses depthwise separable convolutions to reduce parameters and cost, enabling real-time inference with lower latency and memory usage. The architecture introduces two keys, first inverted residual blocks and second linear bottlenecks. These maintain representational power while minimizing tensor dimensionality and reducing information loss during feature extraction. In an inverted residual block, the input is first expanded to a higher-dimensional space. Then, it is passed through a depthwise convolution and projected back to a low-dimensional linear bottleneck layer. This process enhances efficiency without degrading accuracy [31]. Recent studies highlight the effectiveness of MobileNetV2 in embedded and IoT-based detection tasks due to its balance of computational complexity, accuracy, and model compactness [19], [32].

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 3, Maret 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i3.9619

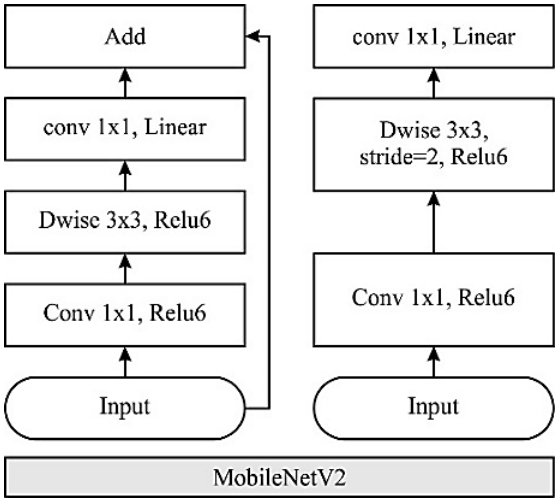

Figure 3. MobileNetV2 Architecture

MobileNetV2 model is trained or fine-tuned to classify images into person and non-person categories. The training process involves feeding manually labeled images into the network, adjusting parameters using the backpropagation algorithm, and monitoring performance metrics such as accuracy and loss throughout the training iterations. A portion of the dataset is allocated for validation to tune hyperparameters, evaluate generalization performance, and prevent overfitting [8]. The outcome of this stage is a trained MobileNetV2 model with satisfactory classification performance, making it suitable for deployment on the ESP32-CAM for edge-based person detection.

d. Result Interpretation

The interpretation of the model's output focuses on how MobileNetV2 differentiates "person" and "non-person" images captured by the ESP32-CAM. The model generates confidence scores indicating the likelihood of human presence, with clear human shapes and body outlines generally producing high-confidence predictions. Misclassifications often occur in low-light conditions, partially obstructed views, or backgrounds containing human-like patterns. These observations highlight the visual features the model relies on and reveal situations where its performance decreases, providing insight into whether further data refinement or model adjustment is needed before deployment.

2.3 System Implementation

a. Implementing the Model on ESP32-CAM

At this stage, the trained MobileNetV2 model is converted into a TensorFlow Lite Micro format that is compatible with the ESP32-CAM [33]. Then, the model is integrated into the device firmware, enabling the device to perform inference directly on the embedded hardware. This process includes configuring the camera module, setting up the inference loop, and ensuring the model operates within the ESP32-CAM's memory and computational constraints. Throughout the process, performance metrics such as inference time and memory usage are monitored to guarantee real-time capability.

b. MQTT Integration

The system incorporates the MQTT protocol to enable real-time notification capabilities. The ESP32-CAM acts as an MQTT publisher, sending alert messages when a person is detected. An EMQX MQTT broker manages message routes and ensures reliable delivery. Client applications on mobile devices or computers act as subscribers and receive these alerts instantly through the publish-subscribe mechanism. MQTT is a lightweight, bandwidth-efficient communication protocol designed for constrained IoT environments. It offers low latency and minimal overhead, making it highly suitable for real-time surveillance and event-driven notifications [34]. This integration ensures the system remains responsive under unstable or limited network conditions.

2.4 Evaluation & Validation

a. Model Evaluation

The performance of the MobileNetV2-based person detection system is evaluated after conversion and deployment on the ESP32-CAM. Standard classification metrics are used for the evaluation, including accuracy, precision, recall, and F1-score. These metrics measure how effectively the model distinguishes between "person" and "non-person" classes. These metrics are calculated using the following formulas [35]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 3, Maret 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i3.9619

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

TP (true positive) indicates correct person detections, TN (true negative) indicates correct non-person detections, FP (false positive) indicates incorrect person detections, and FN (false negative) indicates missed person detections. In addition to classification metrics, deployment performance indicators, such as inference speed and frame processing rate (FPS), are measured to determine the model's efficiency in real-time scenarios. These combined measurements verify that MobileNetV2 maintains reliable performance despite quantization and embedded deployment constraints on the ESP32-CAM.

b. Functional Testing

Functional testing evaluates how the system behaves in real-world environments. Tests are conducted under varying lighting conditions, distances, and subject movements to evaluate practical usability. During this stage, the system's ability to consistently detect "person" and send MQTT alerts within acceptable timeframes is verified. The stability of the detection and communication components is also evaluated.

c. Conclusion and System Validation

The final validation step confirms whether the developed system meets its intended objectives. This involves evaluating its real-time performance, reliability, accuracy, and suitability for home surveillance applications. Based on the evaluation results, conclusions are drawn about the system's strengths and limitations. Recommendations for improvement are also provided. Recommendations for enhancing the model or system architecture may also be provided.

3. RESULT AND DISCUSSION

This section presents the results from developing, implementing, and evaluating an edge-based person detection system that uses the MobileNetV2 model. This system is deployed on the ESP32-CAM and integrated with MQTT to deliver real-time notifications. The discussion covers all results in detail, including dataset characteristics, model training outcomes, evaluation metrics, and system behavior during deployment. The findings underscore the effectiveness of the proposed approach and the feasibility of using lightweight deep learning models on low-power embedded devices for home surveillance applications. Figure 4 illustrates the complete communication flow and component interaction, presenting the overall architecture of the developed system.

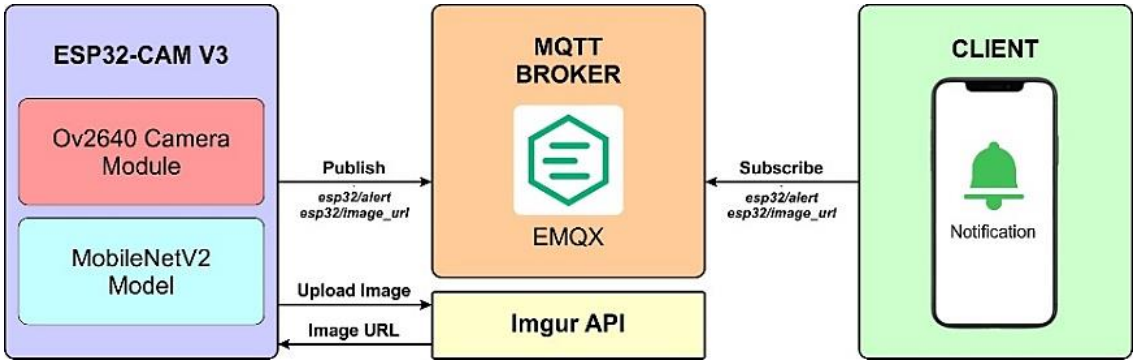

Figure 4. System Architecture

The ESP32-CAM V3 is equipped with an OV2640 camera module and a MobileNetV2 model that has been deployed, enabling the device to perform image acquisition and person detection directly. When motion is detected, the camera captures an image that is processed locally by the MobileNetV2 model to determine if a person is present. If a person is detected, the ESP32-CAM uploads the image to the Imgur API, which returns a publicly accessible URL for the image. This approach avoids transmitting raw image data through MQTT, significantly reducing communication overhead. After obtaining the image URL, the ESP32-CAM publishes two messages to the EMQX MQTT broker: a detection alert message under the esp32/alert topic and the corresponding image URL under the esp32/image_url topic. The MQTT broker routes these published messages to all subscribed clients using the publish–subscribe communication model. A mobile application subscribes to the same MQTT topics on the client side and receives the alert and image URL in real time. The application displays a notification to the user and retrieves the captured image using the provided URL. This allows for visual verification of the detected event and optional local storage of the image. This architecture enables efficient, low-bandwidth, privacy-preserving home surveillance by combining on-device intelligence with lightweight, message-based communication.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 3, Maret 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i3.9619

3.1 Dataset

A total of 500 images were collected through a motion-triggered acquisition process using an ESP32-CAM integrated with a PIR sensor, captured under real operational conditions with a top-down oblique camera angle to ensure optimal subject visibility. To enhance the model's robustness, the dataset encompasses various environmental conditions, including fluctuating light intensities and diverse background complexities, where the camera only recorded frames upon detecting motion within the monitored area. Following acquisition, all images were manually labeled as either "person" or "non-person" to generate precise ground-truth data for training the MobileNetV2 model. This manual annotation process, illustrated by the sample images in Figure 5, is essential to highlight visual differences across different scenarios and ensure the reliability of the classification model during training.

(a) (b)

Figure 5. (a) Labeled as “Person”, (b) Labeled as “non-Person”

3.2 MobileNetV2 Model Training Configuration

The MobileNetV2 model was trained to classify images into two categories, person and non-person, after the dataset was prepared and labeled manually. The model uses the MobileNetV2 backbone as a feature extractor composed of convolutional layers and lightweight inverted residual bottleneck blocks that efficiently learn visual patterns, such as human outlines, motion-related shapes, and background differences. To ensure low-latency inference on edge computing devices, the backbone was configured with a width multiplier (α) of 0.1. This architectural adjustment significantly reduces the number of channels and the overall parameter count, minimizing computational overhead while preserving essential feature extraction capabilities. These features are then passed through a classification head consisting of a global pooling layer and a fully connected output layer for binary classification. For this study, the model was trained for 40 cycles (epochs) with a learning rate of 0.001 using a GPU to speed up the training process. To improve the model's robustness and reduce overfitting due to the limited size of the dataset, data augmentation was enabled to allow the model to learn from varied image conditions. The training process used a batch size of 32 to support stable learning and computational efficiency. INT8 profiling was also activated to support quantized model deployment on embedded devices.

Table 2. MobileNetV2 Training Configuration

Parameter	Value
Number of training cycles (epochs)	40
Learning rate	0.001
Batch size	32
Width Multiplier (α)	0.1

3.3 Model Evaluation Metrics

A model evaluation was conducted to assess the ability of the trained MobileNetV2 model to classify surveillance images as either "person" or "non-person." The dataset, consisting of 500 images, was divided using a Hold-out Validation strategy into 384 training images (77%) and 116 testing images (23%). To monitor the learning process and mitigate overfitting, the training set was further partitioned automatically during the training phase, with 20% (approx. 77 samples) dedicated as a validation set. Based on the training curve (see Figure 6), both training and validation loss decreased significantly and converged after the initial epoch, indicating that the model successfully learned meaningful visual features. The validation loss remained stable and closely aligned with the training loss, demonstrating the model's strong generalization performance. The minor fluctuations observed in the validation curve are attributed to the natural variability of surveillance images, such as differing lighting intensities, background patterns, and partial occlusions.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 3, Maret 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i3.9619

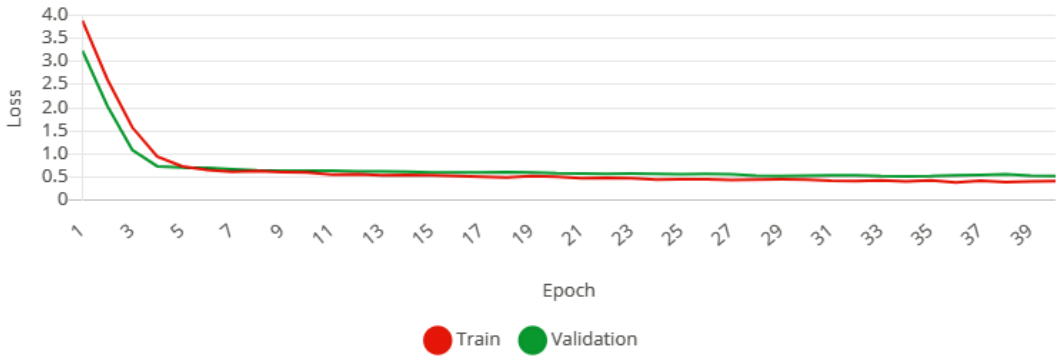

Figure 6. Loss Validation

We used confusion matrix and performance metrics, such as accuracy, precision, recall, and F1-score, to analyze the trained model and provide a detailed evaluation of its classification behavior. The testing dataset consisted of 81 images of people and 35 images of non-people. As shown in Table 3, the model correctly identified 63 images of people (true positives) but incorrectly classified 18 images of people as non-people (false negatives). This suggests that some instances of people were missing, which can occur when the human outline is unclear due to low lighting, obstruction, or when the subject appears too small within the captured frame. For the non-person class, the model correctly classified 28 images as non-people (true negatives); however, it misclassified 7 images of non-people as people (false positives). These false positives occur when background objects or patterns resemble human silhouettes, causing the model to mistakenly detect a person when none is present.

Table 3. Confusion Matrix

Actual \ Predicted	Non-person	Person	Total
Non-person	28	7	**35**
Person	18	63	**81**
Total	**46**	**70**	**116**

Based on 91 correct predictions out of 116 testing samples, the overall test accuracy was calculated as 78.45%, indicating that the model performed reliably for person detection under realistic surveillance conditions. The detailed metric evaluation results are presented in Table 4. When focusing on the person class, the model produced a high precision of 90.00%, meaning that most predictions labeled as “person” were correct and the number of false alarms remained low. This performance is highly beneficial for home surveillance applications because it reduces unnecessary notifications triggered by non-human objects. However, the recall for the person class was 77.78%, indicating that some person instances were still missing during inference. This confirms the presence of false negatives, which may be caused by challenging visual conditions such as low lighting, partial occlusion, motion blur, or cases where the person appears too small or unclear in the captured frame. As a result, the F1-score for the person class reached 83.44%, reflecting a balanced detection capability, with stronger reliability in positive predictions than sensitivity in capturing all person occurrences.

Table 4. Evaluation Results

Metric	Precision	Recall	F1-score	Accuracy
Non-Person	60.87%	80.00%	69.14%	78.45%
Person	90.00%	77.78%	83.44%	

For the non-person class, the model achieved a recall rate of 80.00%. This means that a large portion of non-person scenes was correctly identified, suggesting that the system can reliably recognize empty or non-human environments. However, the precision for the non-person class was lower at 60.87%, showing that a relatively notable portion of images predicted as "non-person" contained a person. This aligns directly with the false negatives observed in the person class, where human presence was incorrectly classified as background. Consequently, the F1-score for the non-person class was 69.14%, which is lower than the person class's F1-score. This confirms that the model's main weakness is misclassifying challenging person images as non-person rather than producing false person detections. Overall, these findings suggest that the model effectively detects people with high confidence when features are clear. However, performance could be further improved by enhancing the diversity of the dataset, particularly for challenging scenarios involving people, to increase recall and reduce missed detections in real-world surveillance conditions.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 3, Maret 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i3.9619

3.4 Inference Performance on ESP32-CAM

After training and converting the MobileNetV2 model into a quantized TensorFlow Lite format, we evaluated its on-device inference performance to determine its feasibility for implementation on the ESP32-CAM platform. The evaluation showed that the deployed model achieved an average inference time of 1018 ms per frame. This indicates that each prediction takes approximately one second, reflecting the inherent computational constraints of the low-cost ESP32-CAM microcontroller. In addition to latency, resource utilization was analyzed to assess the practicality of deployment. The model required a peak RAM usage of 485.4 KB, representing the memory footprint needed during runtime inference. Furthermore, the flash memory allocation was 102.1 KB, indicating that the stored model is highly compact and well-suited for embedded environments. These results confirm that the customized MobileNetV2 can be successfully executed on ESP32-class hardware with a manageable storage footprint. However, the inference speed remains a limiting factor for continuous, high-framerate processing. Operating at approximately 1 frame per second (FPS), the current setup is adequate for basic monitoring but may struggle to capture very fast-moving subjects.

Table 5. Performance Estimate for ESP32

	Image	Object Detection	Total
Latency	15 ms.	1018 ms.	1033 ms.
RAM	4.0 KB	485.4 KB	489.4 KB
Flash	-	102.1 KB	~102.1 KB

3.5 MQTT Communication and Alert Delivery

To enable real-time alerts in the proposed home surveillance system, MQTT communication was implemented with the EMQX broker acting as the message-routing server between the ESP32-CAM device and client applications. When motion is detected or a person-related event occurs, the ESP32-CAM publishes notification messages to specific MQTT topics. The broker log output shows that the ESP32-CAM successfully transmitted an alert message under the esp32/alert topic, indicating that a motion event was detected ("motion_detected"), and subsequently published the captured image reference as an accessible link (Imgur URL) under the esp32/image_url topic. This mechanism enables lightweight data delivery via MQTT, eliminating the need for the ESP32-CAM to directly stream high-bandwidth image data. Consequently, it is well-suited for resource-constrained and low-bandwidth IoT environments.

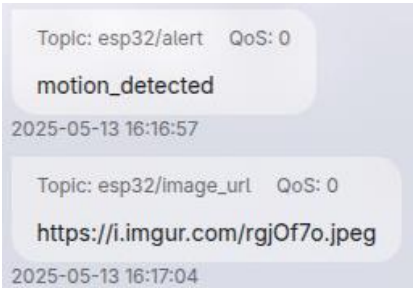

Figure 7. MQTT Broker (EMQX) Log

On the client side, a mobile application was developed to subscribe to these MQTT topics and display real-time notifications. The application displays an instant alert message ("Motion Detected!") when movement is detected, confirming that the MQTT publish-subscribe mechanism functions correctly and delivers timely updates to the user. In addition to receiving the notification, the application can load and display the corresponding captured image as soon as the URL message is received. This enables users to visually verify the detected event. Additionally, the application provides a feature that allows users to save the received image locally via the "Save Image" button. This supports evidence storage and enables further review if needed. Overall, the MQTT-based communication design effectively delivers real-time alerts and supports practical surveillance monitoring by combining event detection, lightweight message transmission, and user-friendly mobile access to captured image evidence.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 3, Maret 2026
e-ISSN 2685-998X
DOI 10.30865/json.v7i3.9619

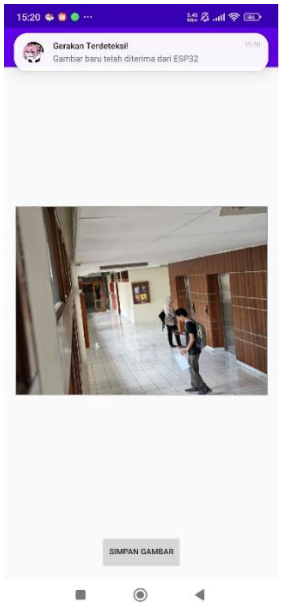

Figure 8. Android Application

3.6 Discussion

This study confirms that an edge-based home surveillance workflow combining MobileNetV2 inference on the ESP32-CAM with MQTT-based messaging is feasible and practical for low-cost residential monitoring. Using a motion-triggered (PIR) data acquisition pipeline, the system was trained and validated on real environment images, achieving 78.45% validation accuracy, with errors dominated by false negatives (missed persons) and a small number of false positives, indicating that challenging conditions such as low light, partial occlusion, and human-like background patterns can reduce detection reliability. On-device deployment shows that the model can run within the ESP32-CAM's storage and memory constraints, but the average inference time of 1018 ms/frame limits continuous real-time processing, making motion-triggered inference an important design choice to maintain responsiveness and efficiency. Finally, MQTT communication via EMQX successfully delivered detection alerts and image URLs to a mobile client in real time, enabling users to view and save captured evidence, thereby demonstrating an integrated, privacy-preserving surveillance system while highlighting the need for improved recall and faster inference in future enhancements.

4. CONCLUSION

This research concludes that an edge-based home surveillance system using a quantized MobileNetV2 on the ESP32-CAM, integrated with MQTT for real-time notification delivery, can be implemented effectively as a low-cost and privacy-preserving security solution. The system successfully supports the full workflow required for practical monitoring: motion-triggered image acquisition using a PIR sensor for dataset creation, manual labeling for supervised learning, MobileNetV2-based classification between person and non-person, on-device inference execution, and event-driven alert transmission through an EMQX MQTT broker to a mobile client application. The evaluation results obtained from the testing set indicate that the trained model achieved an overall accuracy of 78.45%, demonstrating that MobileNetV2 can effectively learn meaningful visual features from real surveillance data while maintaining a low false-alarm rate (90.00% precision). Furthermore, the MQTT implementation proved reliable in delivering notification messages and image references to users, enabling timely awareness and allowing captured evidence to be displayed and saved for review.

Despite the system's overall reliability, several limitations were observed that should be addressed in future work. The primary source of misclassification stemmed from false negatives, indicating that the model occasionally misses person instances under challenging visual conditions, such as low illumination, partial occlusion, motion blur, or when the subject occupies only a fraction of the frame. Furthermore, on-device inference performance remains a key constraint. With an average processing time of approximately 1018 ms per frame, continuous high-framerate video analysis is impractical. Operating at approximately 1 frame per second (FPS), the current continuous processing setup is adequate for basic monitoring but limits the ability to capture fast-moving subjects. Future research should expand dataset size and diversity to enhance model generalization. Additionally, exploring further optimization strategies such as more aggressive quantization, input resolution tuning, or adopting lighter detection architecture is highly recommended. To bypass the continuous processing limitation and optimize energy efficiency, integrating hardware-based motion triggers (e.g., utilizing the PIR sensor during the actual real-time inference phase) should also be explored. Finally, evaluating the system across

broader residential environments, strengthening IoT communication security (e.g., via authenticated topics and encrypted transport), and improving robustness under unstable network conditions would significantly enhance its readiness for real-world deployment.

ACKNOWLEDGMENT

This journal article was prepared by Adi Purnama, Indriani, Bagus Alit Prasetyo, and Agitama Nugraha. They are from the Faculty of Engineering at Widyatama University. The article draws upon the report titled "Edge-Based Person Detection Using MobileNetV2 on ESP32-CAM for Home Surveillance System." This study was funded by the Research, Community Service, and Intellectual Capital Bureau of Widyatama University in 2025. The views presented in this publication are those of the authors and do not necessarily reflect the perspectives of the funding agency.

REFERENCES

[1] L. Sugiharti, M. A. Esquivias, M. S. Shaari, L. Agustin, and H. Rohmawati, "Criminality and Income Inequality in Indonesia," *Soc. Sci.*, vol. 11, no. 3, p. 142, Mar. 2022, doi: 10.3390/socsci11030142.

[2] I. Ikhsan and A. Amri, "Exploration of macroeconomic effects on criminality in Indonesia," *Cogent Soc. Sci.*, vol. 9, no. 1, Dec. 2023, doi: 10.1080/23311886.2023.2206678.

[3] G. Vardakis, G. Hatzivasilis, E. Koutsaki, and N. Papadakis, "Review of Smart-Home Security Using the Internet of Things," Aug. 22, 2024, *Multidisciplinary Digital Publishing Institute*. doi: 10.3390/electronics13163343.

[4] A. A. Aldridge, "Examining the security essences of Internet of Things (IoT) devices in smart homes: challenges, vulnerabilities, and countermeasures," *Issues Inf. Syst.*, vol. 25, no. 1, pp. 279–292, Jan. 2024, doi: 10.48009/1_iis_2024_123.

[5] I. Cahyo Utomo, "Evaluasi Kerentanan Keamanan Pada Perangkat Iot: Studi Kasus Pada Smart home," *Indones. J. Comput. Sci.*, vol. 13, no. 3, pp. 4611–4625, Jun. 2024, doi: 10.33022/ijcs.v13i3.3994.

[6] A. C. Cob-Parro, C. Losada-Gutiérrez, M. Marrón-Romera, A. Gardel-Vicente, and I. Bravo-Muñoz, "Smart video surveillance system based on edge computing," *Sensors*, vol. 21, no. 9, p. 2958, Apr. 2021, doi: 10.3390/s21092958.

[7] I. Rodriguez-Conde, C. Campos, and F. Fdez-Riverola, "On-device object detection for more efficient and privacy-compliant visual perception in context-aware systems," Oct. 02, 2021, *Multidisciplinary Digital Publishing Institute*. doi: 10.3390/app11199173.

[8] K. Dong, C. Zhou, Y. Ruan, and Y. Li, "MobileNetV2 Model for Image Classification," in *Proceedings - 2020 2nd International Conference on Information Technology and Computer Application, ITCA 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 476–480. doi: 10.1109/ITCA52113.2020.00106.

[9] T. Singh, "Optimizing Neural Networks for Real-Time Object Detection Using Edge Computing and AI," Oct. 2024. doi: 10.2139/SSRN.5018415.

[10] Y. Xu, T. M. Khan, Y. Song, and E. Meijering, "Edge deep learning in computer vision and medical diagnostics: a comprehensive survey," *Artif. Intell. Rev.*, vol. 58, no. 93, pp. 1–78, Jan. 2025, doi: 10.1007/s10462-024-11033-5.

[11] R. Kaur and S. Singh, "A comprehensive review of object detection with deep learning," Jan. 01, 2022, *Academic Press*. doi: 10.1016/j.dsp.2022.103812.

[12] M. A. Abu Talib, S. Setumin, A. I. Che Ani, and S. J. Abu Bakar, "An Analysis of Lightweight Convolutional Neural Network Models for Image Classification Task on Edge Device," in *15th IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2025 - Conference Proceedings*, Institute of Electrical and Electronics Engineers Inc., 2025, pp. 175–180. doi: 10.1109/ICCSCE65566.2025.11182642.

[13] L. Zhao and L. Wang, "A new lightweight network based on MobileNetV3," *KSII Trans. Internet Inf. Syst.*, vol. 16, no. 1, pp. 1–15, 2022, doi: 10.3837/tiis.2022.01.001.

[14] R. Kallimani, K. Pai, P. Raghuwanshi, S. Iyer, and O. L. A. López, "TinyML: Tools, applications, challenges, and future research directions," *Multimed. Tools Appl.*, vol. 83, no. 10, pp. 29015–29045, Sep. 2024, doi: 10.1007/s11042-023-16740-9.

[15] Y. A. Soliman, A. S. Ghoneim, and M. M. Elkhouly, "A Comprehensive Systematic Review of TinyML for Person Detection Systems," *IAENG Int. J. Comput. Sci.*, vol. 52, no. 11, pp. 4074–4086, 2025.

[16] 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, p. 1656, 2025, doi: 10.3390/s25061656.

[17] H. Lokhande and S. R. Ganorkar, "Object detection in video surveillance using MobileNetV2 on resource-constrained low-power edge devices," *Bull. Electr. Eng. Informatics*, vol. 14, no. 1, pp. 357–365, Feb. 2025, doi: 10.11591/eei.v14i1.8131.

[18] D. Dovhal, N. Myronova, and A. Parkhomenko, "Research on Object Recognition Approaches for Mobile Platforms with Limited Resources," in *CEUR Workshop Proceedings*, 2025, pp. 142–154.

[19] A. Musa, H. A. Kakudi, M. Hassan, M. Hamada, U. Umar, and M. L. Salisu, "Lightweight Deep Learning Models For Edge Devices—A Survey," *Int. J. Comput. Inf. Syst. Ind. Manag. Appl.*, vol. 17, pp. 189–206, Jan. 2025, doi: 10.70917/2025014.

[20] K. Dokic, "Microcontrollers on the edge – is esp32 with camera ready for machine learning?," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer, 2020, pp. 213–220. doi: 10.1007/978-3-030-51935-3_23.

[21] B. Motamedi and B. Villányi, "A Reliable Publish–Subscribe Mechanism for Internet of Things-Enabled Smart Greenhouses," *Appl. Sci.*, vol. 14, no. 15, p. 6407, Jul. 2024, doi: 10.3390/app14156407.

[22] M. Colaço Júnior, R. Cruz, L. Araújo, A. Bliacheriene, and F. Nunes, "Evaluation of a process for the Experimental

Development of Data Mining, AI and Data Science applications aligned with the Strategic Planning," *J. Inf. Syst. Technol. Manag.*, vol. 19, pp. 1–31, Nov. 2022, doi: 10.4301/s1807-1775202219018.

[23] A. Ghanad, "An Overview of Quantitative Research Methods," *Int. J. Multidiscip. Res. Anal.*, vol. 06, no. 08, pp. 3794–3803, Aug. 2023, doi: 10.47191/ijmra/v6-i8-52.

[24] J. W. . Creswell and J. D. Creswell, *Research design : Qualitative, Quantitative, and Mixed Methods Approaches*, 6th ed. SAGE Publications, Inc., 2022.

[25] S. T. Nowroz, N. M. Saleh, S. Shakur, S. Banerjee, and F. Amsaad, "A Benchmark Reference for ESP32-CAM Module," May 2025, Accessed: Jan. 13, 2026. [Online]. Available: https://arxiv.org/pdf/2505.24081

[26] A. Atturoybi, M. F. Riadi, H. B. Jatiyoso, and A. Mardamsyah, "Development of PIR sensor-based security system and IoT-based esp-32 wrover cam module for monitoring military headquarters and vital objects," *J. Mandiri IT*, vol. 14, no. 1, pp. 191–197, Jul. 2025, doi: 10.35335/MANDIRI.V14I1.437.

[27] M. Kashyap, A. K. Dev, and V. Sharma, "Implementation and analysis of EMQX broker for MQTT protocol in the Internet of Things," *e-Prime - Adv. Electr. Eng. Electron. Energy*, vol. 10, p. 100846, Dec. 2024, doi: 10.1016/j.prime.2024.100846.

[28] I. D. Tirta, A. Wisaksono, A. Ahfas, and J. Jamaaluddin, "Home Surveillance Monitoring with Esp32-Cam and SD Card For Data Storage," *J. Comput. Networks, Archit. High Perform. Comput.*, vol. 6, no. 1, pp. 419–429, Jan. 2024, doi: 10.47709/cnahpc.v6i1.3498.

[29] G. Mezzadri, T. Laloë, F. Mathy, and P. Reynaud-Bouret, "Hold-out strategy for selecting learning models: Application to categorization subjected to presentation orders," *J. Math. Psychol.*, vol. 109, p. 102691, Aug. 2022, doi: 10.1016/j.jmp.2022.102691.

[30] S. Brockmann and T. Schlippe, "Optimizing Convolutional Neural Networks for Image Classification on Resource-Constrained Microcontroller Units," *Computers*, vol. 13, no. 7, p. 173, Jul. 2024, doi: 10.3390/computers13070173.

[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[32] Y. Zhou, S. Chen, Y. Wang, and W. Huan, "Review of research on lightweight convolutional neural networks," in *Proceedings of 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference, ITOEC 2020*, Institute of Electrical and Electronics Engineers Inc., Jun. 2020, pp. 1713–1720. doi: 10.1109/ITOEC49072.2020.9141847.

[33] G. N. Mamtha, S. Sharma, and N. Sing, "Embedded Machine Learning with Tensorflow Lite Micro," in *2023 International Conference on Power Energy, Environment and Intelligent Control, PEEIC 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 1480–1483. doi: 10.1109/PEEIC59336.2023.10451233.

[34] K. T. M. Tran, A. X. Pham, N. P. Nguyen, and P. T. Dang, "Analysis and Performance Comparison of IoT Message Transfer Protocols Applying in Real Photovoltaic System," *Int. J. Networked Distrib. Comput.*, vol. 12, no. 1, pp. 131–143, Jun. 2024, doi: 10.1007/s44227-024-00021-4.

[35] S. Raschka, Y. Liu, and V. Mirjalili, *Machine learning with PyTorch and Scikit-Learn*. Packt Publishing, Limited, 2022.