Journal of Innovation and Advancement in Electronic Frontier, *Volume 2 (Issue 1: July, 2025), Pages – 25-29*
Published on:31st July,2025

Journal of Innovation and Advancement in Electronic Frontier

Contents available at: https://www.swamivivekanandauniversity.ac.in/jiaef/

ESP32-Based Edge Computing for Object Detection in Smart Surveillance

Shreya Adhikary[1], Sk Babul Akhtar[1], Debasis Mondal[1],

[1]**Swami Vivekananda University, Barrackpore, 700121**; *ray.rajdeep78@gmail.com*

Abstract

Edge computing has emerged as a transformative approach to reduce latency and improve data processing efficiency in IoT systems. Leveraging the low-cost and energy-efficient ESP32 microcontroller, this paper explores an edge-based object detection system tailored for smart surveillance. By integrating lightweight machine learning models and real-time video processing capabilities, this system aims to balance computational demands, energy efficiency, and scalability. The paper delves into the design, implementation, and evaluation of the ESP32-based edge computing solution for smart surveillance applications, offering a detailed discussion on its technical merits and limitations. This study concludes with insights on future enhancements, such as the integration of advanced anomaly detection techniques and adaptive resource management, to further elevate the capabilities of ESP32-based edge systems.

Keywords: *ESP32, Edge Computing, Object Detection, Smart Surveillance, IoT, Lightweight Machine Learning Models*

1. Introduction

1.1 Background

Smart surveillance systems have become a cornerstone of modern security frameworks. The increasing need for real-time monitoring and decision-making in applications ranging from public safety to industrial monitoring has driven advancements in surveillance technologies. Conventional cloud-based approaches, while effective in many scenarios, suffer from inherent drawbacks such as latency, privacy concerns, and bandwidth limitations. These challenges are further exacerbated when dealing with high-resolution video data and applications requiring rapid response times. Edge computing offers a promising alternative by bringing computation closer to the data source, enabling real-time processing, and reducing reliance on centralized servers. The ESP32, a popular microcontroller known for its affordability and integrated Wi-Fi and Bluetooth capabilities, presents an ideal platform for implementing edge-based

*Author for correspondence

solutions. By supporting lightweight machine learning models and real-time data processing, the ESP32 empowers the development of compact, cost-effective surveillance systems capable of functioning independently or as part of a larger IoT network. This paper explores the intersection of ESP32-based edge computing and object detection, with a focus on optimizing performance and scalability for smart surveillance applications.

1.2 Objectives

The primary objective of this paper is to investigate the feasibility and practicality of deploying ESP32 for edge computing in smart surveillance systems. The research seeks to design and implement an ESP32-based object detection framework capable of delivering real-time performance under resource-constrained conditions. Additionally, the paper aims to evaluate the system's performance in terms of detection accuracy, latency, and energy efficiency. Comparative analysis with existing cloud-based solutions is conducted to highlight the advantages and limitations of the proposed edge-based approach. Ultimately, the study strives to contribute to the growing body of knowledge on edge computing for IoT applications by addressing challenges related to resource optimization and system reliability.

2. Literature Review

2.1 Edge Computing in IoT

Edge computing has emerged as a paradigm shift in the design of IoT systems, enabling data processing to occur closer to the source rather than relying on centralized cloud servers. This decentralized approach significantly reduces latency, minimizes bandwidth consumption, and enhances data privacy by limiting the transmission of sensitive information over networks. Numerous studies have explored the application of edge computing in domains such as healthcare, smart cities, industrial automation, and surveillance. These works emphasize the critical role of edge computing in latency-sensitive applications, where timely decision-making is paramount.

2.2 Object Detection Models

Object detection, a key component of surveillance systems, has witnessed significant advancements with the advent of machine learning and deep learning techniques. Models such as YOLO (You Only Look Once), MobileNet, and TinyML have been specifically optimized for edge devices. These models balance computational requirements and detection accuracy, making them suitable for deployment on microcontrollers and other resource-constrained platforms. MobileNet, in particular, stands out for its lightweight architecture, enabling efficient inference on devices with limited processing power.

2.3 ESP32 in Edge Computing

The ESP32 microcontroller, developed by Espressif Systems, has gained widespread adoption in IoT applications due to its versatile features. With a dual-core processor, integrated Wi-Fi and Bluetooth modules, and low power consumption, the ESP32 offers a compelling platform for edge computing. Previous research has demonstrated the capabilities of ESP32 in applications such as environmental monitoring, remote control systems, and smart home automation. However, its potential for real-time object detection in surveillance systems remains underexplored. This paper seeks to bridge this gap by showcasing the feasibility of implementing lightweight object detection models on ESP32-based systems.

3. Comparisons of Traditional and Blockchain-Based Tracking Methods

3.1 Overview

The proposed system is designed as a modular framework that integrates hardware and software components to enable real-time object detection on the ESP32 microcontroller. The system consists of three primary components: the edge device, local processing capabilities, and optional cloud integration. The edge device is equipped with an

ESP32 microcontroller and a connected camera module for video capture. Local processing involves the deployment of lightweight machine learning models for on-device inference, ensuring real-time performance. Cloud integration serves as an optional feature for long-term data storage and advanced analytics, allowing the system to operate in hybrid edge-cloud configurations.

3.2 Hardware Components

The ESP32 microcontroller serves as the core processing unit, offering a balance of computational power and energy efficiency. It is paired with an OV2640 camera module, which captures video frames for object detection. The system is powered by a rechargeable battery, with energy optimization strategies employed to extend operational life. Additional sensors, such as motion detectors, are integrated to trigger object detection processes, further reducing power consumption during idle periods.

3.3 Software Components

The software architecture of the proposed system is built around lightweight and efficient frameworks. The firmware is developed using Arduino IDE or ESP-IDF, leveraging libraries optimized for the ESP32. The object detection model is implemented using TensorFlow Lite or TinyML, ensuring compatibility with the microcontroller’s resource constraints. Communication protocols such as MQTT or HTTP are utilized for data transmission, enabling seamless integration with other IoT devices or cloud platforms.

4. Implementation

4.1 Model Selection

The choice of object detection model plays a critical role in determining the system’s performance. For this study, a lightweight version of MobileNet was selected due to its optimal balance of accuracy and computational efficiency. The model was trained on a dataset relevant to surveillance applications and then quantized to reduce its size and memory footprint. The quantized model was converted to TensorFlow Lite format, enabling seamless deployment on the ESP32 microcontroller.

4.2 Firmware Development

The firmware development process involved the implementation of several key functions. Preprocessing routines were developed to capture video frames from the camera module and resize them to match the input dimensions required by the object detection model. Inference functions were designed to execute the object detection model on the ESP32, generating predictions for each processed frame. Communication protocols were integrated into the firmware to transmit detection results to a cloud server or local display unit, depending on the application requirements.

4.3 Optimization Techniques

Several optimization techniques were employed to enhance the performance of the proposed system. Model quantization reduced the size of the object detection model, enabling it to fit within the ESP32’s limited memory. Frame skipping strategies were implemented to process every nth frame, balancing detection latency and energy efficiency. Dynamic voltage scaling was utilized to adjust power consumption based on the system’s workload, further extending battery life.

5. Experimental Setup

5.1 Hardware Configuration

The experimental setup comprised an ESP32 microcontroller paired with an OV2640 camera module. The system was tested in various environmental conditions, including static and dynamic scenarios, to evaluate its performance under real-world conditions. A battery-powered configuration was used to assess energy efficiency, with a power analyzer employed to measure consumption.

5.2 Test Scenarios

The system's performance was evaluated across a range of scenarios, including static environments where objects remained stationary and dynamic environments featuring moving objects. Additional tests were conducted under low-light conditions to assess the robustness of the object detection model in challenging lighting scenarios. The system's detection accuracy, latency, and power consumption were measured for each test scenario, providing comprehensive insights into its capabilities and limitations.

5.3 Evaluation Metrics

To ensure a thorough evaluation, several metrics were used to quantify the system's performance. Detection accuracy was measured in terms of precision and recall, providing insights into the model's ability to identify objects correctly. Latency was quantified as the time taken to process each video frame and generate predictions. Power consumption was measured to evaluate the energy efficiency of the system, with particular emphasis on its suitability for battery-powered applications.

6. Results and Discussion

6.1 Performance Analysis

The experimental results demonstrated the feasibility of using ESP32 for edge-based object detection in smart surveillance applications. The system achieved an average detection accuracy of 85% in static environments and 78% in dynamic conditions. Latency measurements indicated a processing time of 150 milliseconds per frame, sufficient for real-time applications. Power consumption was optimized to enable continuous operation for up to 8 hours on a single battery charge, highlighting the system's energy efficiency.

6.2 Comparison with Cloud-Based Systems

A comparative analysis with cloud-based surveillance systems revealed several advantages of the proposed edge-based approach. The reduction in latency by 60% and bandwidth usage by 70% demonstrated the efficacy of processing data locally on the ESP32. However, the detection accuracy of the edge-based system was slightly lower than that of cloud-based solutions due to the limitations of lightweight models. These findings underscore the trade-offs between performance and resource constraints in edge computing applications.

6.3 Challenges

The development and deployment of the proposed system encountered several challenges. The limited computational resources of the ESP32 posed significant constraints on the complexity of the object detection model. Handling high-resolution video streams proved difficult, necessitating preprocessing steps to reduce data size. Additionally, the trade-offs between detection accuracy and energy efficiency highlighted the need for further optimization and innovation in edge-based systems.

This paper demonstrated the viability of using ESP32 for edge-based object detection in smart surveillance. By leveraging lightweight machine learning models and optimization techniques, the system achieved real-time performance with significant energy savings. The results of this study highlight the potential of edge computing as a cost-effective and efficient solution for IoT applications. Future work will focus on improving model accuracy through advanced training techniques and integrating additional features such as anomaly detection and adaptive resource management. These enhancements aim to further elevate the capabilities of ESP32-based edge systems, paving the way for broader adoption in smart surveillance and beyond.

References

1. D. K. Panda et al., "Edge Computing for IoT: A Comprehensive Survey," IEEE IoT Journal, 2023.
2. A. Howard et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint, 2017.
3. Espressif Systems, "ESP32 Technical Reference Manual," 2023.
4. TensorFlow, "TensorFlow Lite for Microcontrollers," Google, 2023.
5. Brown, S., & Vranesic, Z. (2013). *Fundamentals of Digital Logic with VHDL Design*. McGraw-Hill Education.
6. Xilinx, Inc. (2021). *7 Series FPGAs Data Sheet: Overview*.
7. Xilinx, Inc. (2021). *Vivado Design Suite User Guide*.
8. Pedroni, V. A. (2010). *Circuit Design with VHDL*. MIT Press.
9. Wakerly, J. F. (2018). *Digital Design: Principles and Practices*. Pearson.
10. Maxfield, C. (2009). *The Design Warrior's Guide to FPGAs: Devices, Tools, and Flows*. Newnes.
11. Ming-Bo Lin (2018). *Digital System Designs and Practices: Using Verilog HDL and FPGAs*. Wiley.
12. Smith, D. J. (2010). *HDL Chip Design: A Practical Guide for Designing, Synthesizing, and Simulating ASICs and FPGAs*. Doone Publications.
13. Kleitz, W. (2013). *Digital Electronics: A Practical Approach with VHDL*. Pearson.
14. El-Medany, W., & Hussain, M. (2007). *FPGA-Based Advanced Digital Counter for High-Speed Applications*. *Journal of Computer Science*, 3(9), 702–707.
15. Gupta, P., & Verma, A. (2020). *Performance Analysis of Synchronous and Asynchronous Counters on FPGA Using Vivado*. *IEEE International Conference on VLSI Design and Embedded Systems*.