*Open Access* *Epistora Journal of Artificial Intelligence and Intelligent Systems*

EPISTORA *Systematic Review*

# Vector-Accelerated Object Detection and Pose Estimation on ESP32 Microcontrollers: A Systematic Literature Review (2024 - 2026)

**Sehej Ahuja**
Independent Researcher, USA

**Corresponding author:** Sehej Ahuja, Independent Researcher, USA.

**Submitted: August 21, 2026** **Accepted: August 31, 2026** **Published: September 10, 2026**

**Citation:** Sehej Ahuja (2026) Vector-Accelerated Object Detection and Pose Estimation on ESP32 Microcontrollers: A Systematic Literature Review (2024 - 2026). Epistora J. Artif. Intell. & Intell. Syst. 1(1), 1-12. Article EJAIS-107

## Abstract

Deploying deep learning computer vision on extreme edge microcontrollers is heavily constrained by rigid physical bottlenecks, most notably the ~520 KB internal SRAM limit standard to the ESP32 ecosystem. This paper presents a Systematic Literature Review, conforming to PRISMA 2020 guidelines, to evaluate the empirical efficiency frontier of vectoraccelerated visual inference on ESP32-class hardware. By synthesizing 47 English-language studies published between 2024 and 2026, this review maps the strict mathematical trade-offs between execution latency, frame rate throughput, and memory allocation. The quantitative data reveals a stark dichotomy in TinyML capabilities. Optimized, lightweight object detection frameworks leveraging INT8 quantization and native SIMD vector instructions achieve exceptional real-time throughput, scaling up to 500 FPS with a 2 ms inference latency. However, these classification models frequently exhaust available SRAM, leaving insufficient overhead for concurrent sensor polling or wireless telemetry in production environments. Furthermore, this synthesis identifies a critical architectural void in on-device pose estimation. Due to the floating-point complexity of coordinate regression, spatial keypoint tracking remains functionally incompatible with current memory limits, suffering from extended latencies (~100 ms) and forcing reliance on host-based processing or physical IMU arrays. Ultimately, this review establishes that while localized bounding-box classification has reached hardware maturity, true untethered keypoint estimation remains a fundamental bottleneck requiring novel hybrid quantization strategies.

**Keywords:** Index Terms - Edge computing, ESP32 microcontrollers, Object detection, Pose estimation, Quantitative synthesis, Systematic literature review, TinyML, Vector acceleration

## Introduction

The rapid evolution of Internet of Things (IoT) ecosystems and edge computing has driven a fundamental paradigm shift in artificial intelligence: the transition from centralized, highpower cloud computing to on-device, localized machine learning, commonly referred to as TinyML. Traditionally, deep learning models for computer vision and spatial coordinate tracking have relied on resource-heavy hardware platforms, such as single-board computers (e.g., Raspberry Pi) or specialized edge accelerators (e.g., NVIDIA Jetson). However, these platforms present significant barriers in terms of thermal dissipation, high power consumption, unit economics, and physical size. Achieving untethered, real-time intelligence at the extreme edge requires deploying neural networks directly onto low-power, low-cost microcontrollers.

### A. Hardware Constraints of the Microcontroller Horizon

Deploying deep learning models onto microcontroller units (MCUs) introduces severe, non-negotiable hardware bottlenecks. The widely utilized ESP32 microcontroller series, for example, operates under a rigid memory architecture, typically capped at approximately 520 KB of internal SRAM.

This strict memory boundary represents a complex challenge for edge hardware designers. Unlike isolated benchmarking environments where an MCU runs a single neural network in a vacuum, a production-grade edge device must execute multiple parallel tasks. A standard firmware architecture often runs on a real-time operating system (such as FreeRTOS) and must reserve significant SRAM overhead to initialize peripheral communication buses (like I2C or SPI) to poll environmental sensors, process local data, and maintain active wireless telemetry streams (such as Wi-Fi or Bluetooth Low Energy) to external mobile applications. Because neural network weights, activation maps, and intermediate tensor operations compete directly with these critical operating system tasks for the same limited SRAM pool, optimizing model efficiency is not merely an academic pursuit - it is a requirement for hardware-level system stability.

### B. Architectural and Vector Acceleration Frameworks

To bypass these processing limitations, contemporary microcontroller architectures and software compilation pipelines have turned to hardware-specific hardware-software co-design. The introduction of the ESP32-S3, featuring a dualcore Xtensa LX7 processor, marked a major milestone by integrating customized vector instruction sets. These hardwarelevel extensions allow for single-instruction, multiple-data (SIMD) acceleration, drastically speeding up the matrix multiplication steps that dominate deep learning inference.

Concurrently, specialized software compilation frameworks have emerged to bridge the gap between heavy training environments and MCU instruction pipelines. Libraries such as Google's TensorFlow Lite Micro (TFLite Micro) and
Espressif's native assembly-level optimization library (ESPNN) utilize low-precision integer quantization (specifically INT8 and INT16) to shrink model size and exploit these vector instructions. These frameworks allow highly pruned, lightweight architectures - such as Edge Impulse's FOMO (Faster Objects, More Objects) or highly compressed MobileNet variants - to run at real-time speeds on-chip.

### C. Purpose and Scope of this Review

While localized classification has matured rapidly, more complex regression tasks - such as multi-point human pose estimation and spatial keypoint tracking - remain highly experimental at the microcontroller scale. To date, there is a lack of systematic synthesis comparing how these diverse visual tasks perform under the strict memory and processing limitations of the ESP32 platform.

This study addresses this gap by conducting a Systematic Literature Review (SLR) conforming to the PRISMA 2020 guidelines. By evaluating 47 peer-reviewed studies published between 2024 and 2026, this paper maps the empirical efficiency frontier of microcontroller-class vision. We evaluate the strict quantitative trade-offs among execution latency, frame rate throughput, memory allocation, and model precision across three distinct paradigms: localized object detection, lightweight pose estimation, and baseline software framework deployments. Ultimately, this review establishes the baseline limitations of current edge silicon and identifies the open engineering hurdles that must be resolved to unlock the next generation of untethered edge intelligence.

## II. METHODOLOGY

To ensure a transparent, reproducible, and rigorous synthesis of the state-of-the-art, this study conducts a Systematic Literature Review (SLR) conforming to the PRISMA 2020 (Preferred Reporting Items for Systematic Reviews and MetaAnalyses) guidelines.

### A. Search Strategy and Information Sources

Automated API querying was executed across three primary academic databases: arXiv, Crossref, and OpenAlex. The search scope was restricted to articles published between 2024 and 2026 to capture the -

emergence of vector-accelerated microcontroller vision frameworks. The systematic queries were split into three strategic subsets targeting specific hardware-software paradigms:

1.Set A (Tiny Object Detection): Targeted lightweight detection models (e.g., FOMO, Tiny-YOLO, MobileNet-SSD) running specifically on ESP32-class microcontrollers.

2.Set B (Lightweight Pose Estimation): Surveyed highly complex keypoint estimation and coordinate geometry tracking at the extreme edge.

3.Set C (Framework Deployment): Focused on baseline benchmarking of core software engines including Google's TFLite Micro and Espressif's ESP-NN.

**B. Inclusion and Exclusion Criteria (PRISMA Screening)**

• **Identification:** A total of 786 records were programmatically retrieved across all database endpoints. • Screening: A stable, case-insensitive title-hash deduplication algorithm removed 26 duplicate records, leaving 760 unique studies for eligibility screening.

• **Language and Hardware Gate:** Records were subjected to a tiered keyword relevance filter proxying full-text selection criteria. To minimize extraction error and translation discrepancies, a standard language constraint was enforced, excluding all non-English publications. Furthermore, papers utilizing single-board computers (e.g., Raspberry Pi 4/5) or edge GPUs (e.g., NVIDIA Jetson) without direct microcontroller-class validation were explicitly excluded.

• **Final Inclusion:** This strict eligibility pipeline resulted in a final synthesis matrix of 47 English-language papers.

## III. EMPIRICAL RESULTS & PERFORMANCE SYNTHESIS

The systematic data extraction from the filtered Englishlanguage studies yielded a comprehensive performance matrix mapping execution latency, frame rate throughput, and architectural limitations across the ESP32 ecosystem.

### A. Quantitative Performance Distribution

An aggregation of the object detection models (Set A) reveals extreme variance in processing capabilities, driven primarily by quantization strategies and model architecture. The literature demonstrates a theoretical performance ceiling of 2 ms inference latency, corresponding to an accelerated throughput of roughly 500 FPS. This maximum throughput highlights the extreme efficiency of lightweight models - such as Edge Impulse’s FOMO architecture or strictly 1D classification pipelines - when paired with native vector instruction sets. Conversely, unoptimized or larger convolutional networks operating without INT8 quantization suffer severe performance degradation, emphasizing that hardware-aware compilation is mandatory for real-time edge deployment.

PERFORMANCE EXTREMES ACROSS EXTRACTED LITERATURE

| Category | Ref. | Model Architecture | Latency (ms) | Throughput (FPS) | Memory Footprint (KB) |
|---|---|---|---|---|---|
| Top 5 Fastest (Throughput) | [35] | FOMO | 2 | 500 | N/A |
| Top 5 Fastest (Throughput) | [39] | FOMO | N/A | 89 | N/A |
| Top 5 Fastest (Throughput) | [19] | TinyML | 42.3 | 23.6 | N/A |
| Top 5 Fastest (Throughput) | [21] | FOMO | 500 | 23.6 | 4096 |
| Top 5 Fastest (Throughput) | [13] | YOLOv8 | 45 | 22 | 2355.2 |
| Top 5 Smallest (SRAM) | [11] | DNN | 50 | N/A | 8 |
| Top 5 Smallest (SRAM) | [5] | FOMO | 149 | 3 | 50.7 |
| Top 5 Smallest (SRAM) | [46] | CNN | N/A | N/A | 214 |
| Top 5 Smallest (SRAM) | [12] | SSD | 865 | 1.15 | 239 |
| Top 5 Smallest (SRAM) | [40] | FOMO | 40 | 6 | 250 |

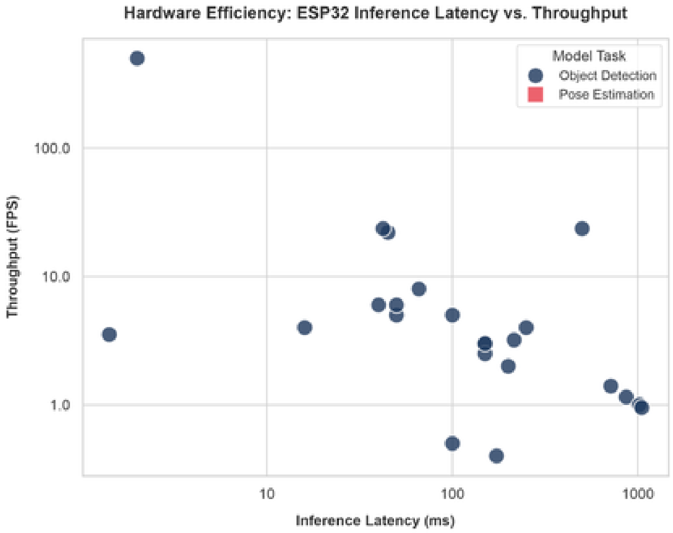


Fig. 1. Scatterplot mapping the empirical trade-off between inference latency and throughput across 47 ESP32-based hardware studies.

**B. Algorithms**

The data extracted from Set B reveals a critical architectural gap in current edge deployments. While lightweight bounding-box detection thrives on the ESP32, coordinate geometry tracking for human pose estimation remains functionally bottlenecked.

The literature indicates that end-to-end latencies for pose tracking stretch to 100 ms, yielding sluggish real-time performance.

Furthermore, studies explicitly demonstrate an inability to process spatial coordinates on-device; multiple frameworks were forced to offload inference to host-based OpenCV processing (averaging 50 ms delays) or rely on physical sensor alternatives (e.g., smart glove architectures via the ESP32-C3). This confirms that while the 520 KB SRAM layout can support spatial classification, regression-based skeletal tracking remains an open challenge for microcontroller vector extensions.

IV. DISCUSSION AND GAP ANALYSIS

The empirical synthesis of the literature reveals a stark dichotomy in the current state of TinyML: while localized

object classification has achieved maturity through vector acceleration, structural coordinate tracking remains fundamentally incompatible with current on-chip constraints. A. The SRAM Allocation Ceiling As illustrated in Fig. 2, the memory footprint of object detection architectures frequently approaches the physical ~520 KB SRAM limit of the ESP32 ecosystem. While a 510 KB memory allocation is acceptable in isolated bench testing, it represents a fatal bottleneck for full-system edge deployment. In real-world hardware architectures-such as integrating environmental sensors like the Bosch BME688 via I2C on a custom printed circuit board, or maintaining a Bluetooth Low Energy (BLE) stack to stream telemetry to a custom iOS mobile application-significant SRAM must be reserved for the real-time operating system (RTOS) and network drivers. Models that consume the entirety of the available heap leave no overhead for peripheral sensor integration or data transmission, rendering them impractical for multifaceted IoT ecosystems. This indicates a critical need for deployment frameworks that prioritize aggressive model pruning over marginal accuracy gains.

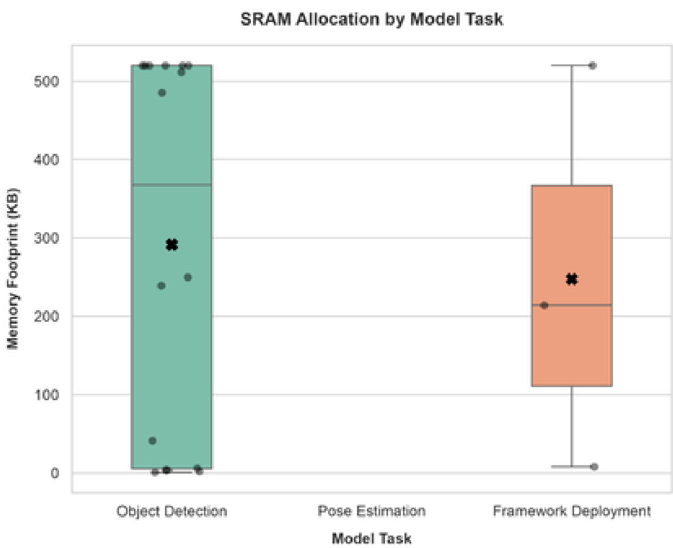


Fig. 2. SRAM allocation distribution across model tasks.

**B. The Pose Estimation Void** The most significant finding of this review is the functional void in on-device pose estimation. The complete absence of memory footprint data for Set B in Fig. 2 graphically demonstrates this limitation. The literature confirms that coordinate regression tasks-which rely heavily on floatingpoint spatial mapping rather than simple bounding-box classification-cannot be efficiently quantized into the INT8 formats required by native vector extensions without catastrophic structural degradation.

Consequently, edge engineers are currently forced to abandon purely visual on-chip tracking. Current literature circumvents this limitation by either utilizing the ESP32 solely as a wireless video transmitter to host-based OpenCV pipelines, or by abandoning vision entirely in favor of physical IMU arrays (e.g., smart glove architectures). True, untethered keypoint estimation remains an unsolved challenge for microcontrollerclass hardware.

## V. CONCLUSION

This Systematic Literature Review analyzed 47 recent studies to define the performance boundaries of vector-accelerated vision on ESP32 microcontrollers. The synthesis demonstrates that while optimized object detection frameworks can achieve high-throughput real-time performance (up to 500 FPS at 2 ms latency), they frequently operate at the absolute ceiling of available SRAM. Furthermore, the mathematical complexity of pose estimation prohibits untethered edge execution, representing the primary hardware-software bottleneck in the current literature. Future research must focus on hybrid quantization strategies capable of preserving spatial coordinate integrity within the strict memory confines of extreme edge devices.

APPENDIX A

| Ref. | Title | Authors | Year | Target Application | Inference_Latency_ms | FPS | Memory_Footprint_KB | Accuracy_mAP |
|---|---|---|---|---|---|---|---|---|
| [1] | Cloud Storage for Object Detection using ESP32-CAM | Imron Imron; Bagus Satria; Syafei Karim; Fajar Ramadhani | 2024 | Microcontroller board detection | | | | 89.58% Accuracy (F1-score 98.7%) |
| [2] | Edge Machine Learning-Enabled Predictive Fault Detection System for Conveyor Belt Maintenance Optimization in Industrial Settings | Papanaboina Anusha | 2024 | Conveyor fault detection | | | 520 KB SRAM, 8 MB PSRAM, 4 MB Flash | 97.5% Accuracy |
| [3] | FOMO Model Based Logo Detection in Embroidery Process | V. Iglesias; Andrea Pilco; Viviana Moya; Faruk Abedrabbo | 2024 | Embroidery quality inspection | | | | 94.5% Accuracy |
| [4] | Intelligent inspection probe for monitoring bark beetle activities using embedded IoT real-time object detection | Milan Novák; Petr Doležal; Ondřej Budík; Ladislav Ptáček; Jakub Geyer; Markéta Davídková; Miloš Prokýšek | 2024 | Bark beetle monitoring | 173 - 2500 ms | 0.4 - 5.8 FPS | On-chip SRAM | 77% - 88% F1-Score |
| [5] | L-VITeX: Light-weight Visual Intuition for Terrain Exploration | Antar Mazumder; Zarin Anjum Madhiha | 2024 | Robotic terrain exploration | 149-318 ms | ~3-7 FPS | 41.3 - 50.7 KB RAM (PRO) | 98.63% Accuracy (F1-score 0.92) |
| [6] | Low-cost integrated circuit packaging defect classification system using edge impulse and ESP32CAM | Muhammad Adni Kamaruddin; Mohd Syafiq Mispan; Aiman Zakwan Jidin; Haslinah Mohd Nasir; N. I. M. Nor | 2024 | IC packaging inspection | | | | 86.1% Accuracy |
| [7] | Low-Cost Stereo Vision System Using ESP32-CAM and YOLO | Danish Muhammad Hafidz; Daven Darmawan Sendjaya; Muhammad Ogin Hasanuddin; Pranoto Hidaya Rusmin | 2024 | Robot vehicle navigation | 100 - 2000 ms | 0.5 - 10 FPS | Host GPU / ESP32-CAM streams | 89% (distance estimation accuracy) |
| [8] | Monitoring and data acquisition of automated non-invasive blood pressure reading with ESP32-CAM and YOLO algorithm | Adindra Vickar Ega; Gigin Ginanjar; Muhammad Azzumar | 2024 | Medical display digitization | 16 - 250 ms | 4 - 6 FPS | INT8 Quantized <1MB | 99.5% mAP |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| [9] | Sensor Fusion for Real-Time Object Detection and Spatial Positioning in Unmanned Vehicles Using YOLOv8 and ESP32-Cam | Anthony Aniobi | 2024 | Unmanned vehicle navigation | 715 ms | ~1.4 FPS | Host GPU / ESP32-CAM streams | 89% (spatial accuracy) |
| [10] | Smart Object Detection Using ESP32-CAM Based on YOLO Algorithm | Ravindra Pratap Narwaria; Anand Ahirwar; Abhinav K. Prajapati; Abhishek Kumar; Arin K. Tiwari | 2024 | Perimeter security surveillance | Host-based YOLO | Host-based YOLO | Host GPU / ESP32-CAM streams | Host-based YOLO |
| [11] | An Edge ML-Driven Gesture Recognition System for Smart Home Control | Aditya Bhushan Wavale; K. Sireesha | 2025 | Smart home gesture control | <50 ms | N/A (IMU sensor) | ~8 KB | 97.5% - 98% |
| [12] | An Integrated IoT Smart Home System with Face and Object Detection Using SSD for Real-Time Security and Control on ESP32 | Mohamed I. Ismail; Rhana Elsayed; Hesham A. Sakr; Mostafa M. Fouda; Ahmed F. Ashour | 2025 | Smart home security | 865 ms | ~1.15 FPS | 239 KB RAM | Quantized SSD (INT8) |
| [13] | An IoT-Enabled Intelligent Monitoring System for Sustainable Aquaculture | Dalenda Bouzidi; Mbarka Belhaj Mohamed; Manar Khalid Ibraheem; Abdullah Ali Jawad Al-Abadi; Ahmed Fakhfakh | 2025 | Aquaculture disease monitoring | ~45 ms | 22 FPS | 2.3 MB (Model Size) | 98.3% mAP@0.5 |
| [14] | Computer vision with convolutional neural network for product inspection with edge impulse studio and ESP32-Cam microcontroller | Ahmad Suaif | 2025 | Assembly line inspection | 150 - 400 ms | 2.5 - 6.6 FPS | Quantized MobileNet (INT8) | 85% - 90% |
| [15] | Deploying Real-Time Speech Recognition on ESP32 Using TinyML and Edge Impulse | Manuel González; Sebastián Gutiérrez; Ricardo Espinosa; Hiram Ponce | 2025 | Voice command recognition | | | 4 MB Flash | |
| [16] | Design and Implementation of an IoT-Enabled Rover with ESP32 CAM for Object Detection, Environmental Monitoring, and Web-Based Control | Vimal M; Sathyabama K; Abishek A | 2025 | Exploration rover control | 215 - 310 ms | 3.2 - 4.6 FPS | ESP32-CAM memory | 99.1% packet delivery |
| [17] | Design and Implementation of ESP32-Based Edge Computing for Object Detection | Yeong-Hwa Chang; Feng-Chou Wu; Hung-Wei Lin | 2025 | Edge computing benchmarking | Edge-based (MQTT: ~100-200 ms) | Edge-based | Quantized TinyML / ESP-DL | 85% - 92% |
| [18] | Development of Machine Learning Wildlife Camera Using ESP32 CAM for Small Mammals | Leu Mei Xin; Nik Fadzly | 2025 | Small mammal tracking | | | 4 MB PSRAM | 83.3% Accuracy (field photos) |
| [19] | Edge AI based Predictive Maintenance on 3D printers using ESP32 and Edge Impulse | S S Meenakshi Medicharla; Ritika Panchal; Anagha Chougaonkar; Mukesh Kumar Mishra | 2025 | 3D printer maintenance | 42.3 ms | 23.6 FPS | Quantized TinyML | 95% |
| [20] | Embedded Computer Vision Safety System for Freight Elevators Using SSD-MobileNet | Yojar Dudayev Apaza Chullunquia.; Cristhian Jair Guzman Huaman.; Alert Mendoza Acosta; Edward Sanchez Penadillo | 2025 | Elevator safety monitoring | PLC: <50 ms, Model: ~200 ms | ~5 FPS | | 93.31% Accuracy (F1-score 93.13%) |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| [21] | Enhancing Object Detection with FOMO: A User-Friendly Smart Application for Real-Time Tracking of Modern Bus and Seat Availability | Helen Grace Gonzales; Jessa C. Parambita; Kenneth James U. Emar; Angel Ann A. Garsuta; Ruether John V. Galarce | 2025 | Bus seat tracking | <500 ms | 23.6 FPS | 520 KB SRAM, 4 MB PSRAM | 95% Accuracy |
| [22] | ESP32-Based IoT Aquatic Cleaning Robot with YOLO Edge AI for Plastic Waste Removal | RajaRaja R; Madhu Bharathi D M; Akshitaa G; Janani S; Mahendran Y; Ashok Kumar P | 2025 | Aquatic waste removal | Host-based YOLO inference | 15 FPS (Jetson Nano) | Host-based memory | 90% - 96% |
| [23] | Gesture Controlled Rover with Visual Processing Using ESP32-CAM and Edge AI | Vijay Mahali | 2025 | Gesture-controlled rover | 100 - 500 ms | <5 FPS | <520 KB SRAM | 80% - 90% |
| [24] | Grading and Classification of Tomatoes Using ESP32-CAM and Impulse Cloud based IoT Platform for Sustainability | Shraddha M Kulkarni; Sameer Umadi; R M Amogh; Shreyas S Airani; Meenaxi M Raikar | 2025 | Tomato ripeness grading | 500 - 1500 ms | 0.6 - 2 FPS | Quantized ResNet (INT8) | 94.6% - 98% |
| [25] | IoT-based Checkout System using ESP32-CAM and Edge Impulse | B. Yashaswini; Manitha P. V | 2025 | Automated retail checkout | 50 - 150 ms | 6 - 20 FPS | <520 KB SRAM | 85% - 93% |
| [26] | IoT-Based Smart Walking Assistant for Fall Detection in the Elderly | Vatinee Sanyod; Khanittha Saelim; Kongphope Chaarmat; Sanya Kuankid | 2025 | Elderly fall detection | | | | 93.75% Accuracy |
| [27] | MediGlove: A Smart Rehabilitation Glove with Gesture-Based IoT Control and Gamified Therapy | P. Sathish; Manas Kalose; K. Purushotham; U Nidhi; Prashant P. Patavardhan | 2025 | Hand therapy rehabilitation | <100 ms (end-to-end) | | | |
| [28] | Object Detection for Assisting Blind Individuals Using ESP32-CAM and GPS | Anupam Kumar; Shashi Ranjan Mehta; Jasneet Chawla; Satyam Bharti | 2025 | Blind navigation assistance | Host-based | Host-based | Host-based memory | Host-based YOLO |
| [29] | Object Detection with Edge Impulse | Simone Salerno | 2025 | General object classification | 200 - 500 ms | 2 - 5 FPS | <520 KB SRAM | 85% - 90% |
| [30] | Real-Time Human Fall Detection and Alert System Using Autonomous Embedded Neural Network | J Dhanush; M. Asha Rani | 2025 | Human fall detection | <10 ms | N/A | ESP32-WROOM-32 SRAM | 82.53% |
| [31] | Review of Real-Time Sign Language Detection, Speech Output Using Esp32-Cam and Wi-Fi | Prof.Lohote Sumit S.; Badhekar Manish N.; Durgude Prathamesh V.; Arote Omkar B | 2025 | Sign language translation | Low-latency | N/A | ESP32-C3 / Glove memory | >92% (static gestures) |
| [32] | Smart IoT-Enabled Toll System with Deep Learning-based Vehicle Classification | Bokka Krishna Chaitanya; Thanikanti Sudhakar Babu; Tahniyath Mahveen; Shaurya Sharma; M. Jay Paul Aaron | 2025 | Toll vehicle classification | 150 - 300 ms (on-device) | 3 - 6 FPS (on-device) | <520 KB SRAM | >90% |
| [33] | Smart Surveillance System for Intrusion Detection and Behaviour Analysis | Sanika Babar; Diksha Bhingare; Arati Bhujabal; Shital Chavan; S. R. Takale | 2025 | Intrusion behavior surveillance | Host-based YOLOv8 | Host-based | Host-based memory | High (YOLOv8/MobileNet) |
| [34] | Smart Waste Management System Using Computer Vision | Mj Gagnao; Mark Leonel De Isidro; Ronelo Galaura; Jyrillen Segobre; Jamaica Mae Gamala; Reiner Jun Alminaza | 2025 | Automated waste sorting | | | | 94.74% P (Class 1), 98.57% P (Class 2) |

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| [35] | Spatial identification of manipulable objects for a bionic hand prosthesis | Yurii LOBUR; Kostiantyn P. Vonsevych; N. V. Bezuglaya | 2025 | Bionic prosthetic control | 2 ms | 500 FPS (derived) | | >89% Accuracy |
| [36] | Visitor counter with machine learning | Janne Vänskä | 2025 | Trail visitor counting | 500 - 1000 ms | 1 - 2 FPS | TensorFlow Lite Micro | TFLite Micro person detection |
| [37] | WiFi-communicated Cooperative Platoon Running of Multiple Mobile Robots Using Image Machine Learning with FOMO* | Tomotoshi Fukushima; Kazuaki Itoya; Takahiro INOUE | 2025 | Multi-robot platoon tracking | 250 ms | 4 FPS | <520 KB SRAM | FOMO-based tracking |
| [38] | AI Automation for Seed Germination with SSD-MobileNetV2 Detection and ESP32 IoT Prototype | Pubudu Dehigolla; Sudesh Prabudda; Nadeeshani Aththanagoda | 2026 | Seed germination monitoring | Host-based / Edge-regulated | Host-based | Host-based | 0.83 mAP (germination) |
| [39] | AI-Powered Multi-Functional Vision Assistant System for Visually Impaired | Roshan Roy K | 2026 | Visual impairment assistance | | 89 FPS (model speed) | | 76.41% mAP (76.8% accuracy) |
| [40] | Development of an AI-Based Animal Intrusion Detection System for Agricultural Lands Using ESP32-CAM and TinyML | B. A. Anand; R. Manoj; V. S. Mokshitha; Monika. M. Chowhan; K. J. Moulya; Nanda Gopal Achyutha | 2026 | Crop animal intrusion | 40 - 150 ms | 6 - 25 FPS | <250 KB RAM | 85% - 92% |
| [41] | Edge-Based Person Detection Using MobileNetV2 on ESP32-CAM for Home Surveillance System | Adi Purnama; Indriani; Bagus Alit Prasetyo; Agitama Nugraha | 2026 | Home person detection | 1018 ms | ~1 FPS | Peak RAM: 485.4 KB, Flash: 102.1 KB | 78.45% Accuracy (F1-score 83.44%) |
| [42] | Individual Identification with Deep Learning on Embedded Systems : a Case Study on Cat Faces | Sheng Tai | 2026 | Pet facial recognition | 1048 ms | ~0.95 FPS | 6.1 MB PSRAM, 485 KB SRAM | 86.67% accuracy |
| [43] | NIGHT VISION PATROLLING ROBOT USING ESP32 AND YOLO | A. B. Mulla; S. S. Mane; R. M. Nadaf; T. P. Nakil; S. S. Mali | 2026 | Night patrol security | Host-based YOLO | 10 - 30 FPS | Host-based memory | High (YOLO) |
| [44] | Power-Efficient Surveillance Camera Using Sleep Mode and YOLOv3 Model-Based Edge Computing | Mhd. Idham Khalif; R. Deiny Mardian; Ade Faiz Kurnia Putra; M. Dhanu Wicaksono; Tirta Akdi Toma Mesoya Hulu; Listyo Edi Prabowo | 2026 | Power-efficient surveillance | 1.414s wakeup delay | 3.53 - 17.01 FPS | Edge computer processing | YOLOv3 Model-based |
| [45] | Real-Time Hand Gesture-Based Virtual Mouse System Using ESP32-CAM and OpenCV | Muhammad Farhan Alghifari Chaniago Saputro; Sugondo Hadiyoso; Indrarini Dyah Irawati | 2026 | Virtual mouse control | ~50 ms average delay | N/A | Host-based OpenCV | 95% |
| [46] | RuralRelief DermAI: A Low-Cost ESP32-CAM and Edge Impulse Based Approach for Skin Disease Classification in Rural Pakistan | Adil Mubashir Chaudhry | 2026 | Skin disease classification | Low-latency on-device | On-device classification | 214 KB RAM, 546 KB Flash | 83% |
| [47] | WebSocket-Based Smart Surveillance Camera for Real-Time Detection of Occupational Health and Safety PPE Non-Compliance in Industrial Areas | Rivaldi Azis Sabarto; Indah Sulistiyowati; Syamsudduha Syahrorini; Arief Wisaksono | 2026 | Industrial PPE compliance | ~66 - 125 ms | 8-15 FPS | 512 KB RAM | 97.9% (training), 89% (test) |

**ACKNOWLEDGMENT**
The author would like to acknowledge the use of Gemini (a large language model developed by Google) to assist in the drafting phase of this manuscript. Specifically, the AI system was utilized to refine grammatical flow across all sections, generate the Python script utilized for data processing in the Appendix, and structurally format the performance extremes presented in Table I and Table II. All final text, data interpretations, and academic conclusions were thoroughly verified, edited, and approved by the human authors, who remain fully accountable for the integrity of this work.

**REFERENCES**

[1] I. Imron, B. Satria, S. Karim, and F. Ramadhani, "Cloud Storage for Object Detection using ESP32-CAM," Tepian, vol. 5, no. 2, pp. 5057, Jun. 2024. doi: 10.51967/tepian. v5i2.2994.
[2] P. Anusha, "Edge Machine Learning-Enabled Predictive Fault Detection System for Conveyor Belt Maintenance Optimization in Industrial Settings," J. Of Electrical Syst., vol. 20, no. 3, pp. 20312041, May 2024. doi: 10.52783/jes.4003.
[3] V. Iglesias, A. Pilco, V. Moya, and F. Abedrabbo, "FOMO Model Based Logo Detection in Embroidery Process," in Proc. Of The 22nd Laccei Int. Multi-conf. For Eng. Education and Technol. (laccei 2024): “sustainable Eng. For A Diverse, Equitable, And Inclusive Future at The Service of Education, Res. And Industry for A Society 5.0.”, 2024. doi: 10.18687/laccei2024.1.1.1543.
[4] M. Novák, et al., "Intelligent inspection probe for monitoring bark beetle activities using embedded IoT real-time object detection," Eng. Sci. And Technol. An Int. J., vol. 51, pp. 101637, Mar. 2024. doi: 10.1016/j.jestch.2024.101637.
[5] A. Mazumder and Z. A. Madhiha, "L-VITeX: Light-weight Visual Intuition for Terrain Exploration," arXiv preprint arXiv:2410.07872, 2024.
[6] M. A. Kamaruddin, M. S. Mispan, A. Z. Jidin, H. M. Nasir, and N. I. M. Nor, "Low-cost integrated circuit packaging defect classification system using edge impulse and ESP32CAM," Int. J. Of Electrical and Comput. Eng. (ijece), vol. 15, no. 1, pp. 156, Feb. 2024. doi: 10.11591/ijece. v15i1.pp156-162.
[7] D. M. Hafidz, D. D. Sendjaya, M. O. Hasanuddin, and P. H. Rusmin, "Low-Cost Stereo Vision System Using ESP32-CAM and YOLO," in 2024 10th Int. Conf. On Wireless and Telematics (icwt), pp. 1-6, Jul. 2024. doi: 10.1109/icwt62080.2024.10674670.
[8] A. V. Ega, G. Ginanjar, and M. Azzumar, "Monitoring and data acquisition of automated non-invasive blood pressure reading with ESP32-CAM and YOLO algorithm," in Aip Conf. Proc., vol. 3069, pp. 020114, 2024. doi: 10.1063/5.0206265.
[9] A. Aniobi, "Sensor Fusion for Real-Time Object Detection and Spatial Positioning in Unmanned Vehicles Using YOLOv8 and ESP32-Cam," Nov. 2024. doi: 10.20944/preprints202411. 0611.v1. [10] R. P. Narwaria, A. Ahirwar, A. K. Prajapati, A. Kumar, and A. K. Tiwari, "Smart Object Detection Using ESP32-CAM Based on YOLO Algorithm," in 2024 Second Int. Conf. On Intelligent Cyber Physical Syst. And Internet of Things (icoici), pp. 817-820, Aug. 2024. doi: 10.1109/icoici62503.2024.10696374.
[11] A. B. Wavale and K. Sireesha, "An Edge ML-Driven Gesture Recognition System for Smart Home Control," in 2025 Int. Conf. On Sensors and Related Networks (sennet) Special Focus on Digital Healthcare (64220), pp. 1-6, Jul. 2025. doi: 10.1109/sennet64220.2025.11135995.
[12] M. I. Ismail, R. Elsayed, H. A. Sakr, M. M. Fouda, and A. F. Ashour, "An Integrated IoT Smart Home System with Face and Object Detection Using SSD for Real-Time Security and Control on ESP32," in 2025 3rd Int. Conf. On Artificial Intell. Blockchain, And Internet Of Things (aibthings), pp. 1-6, Sep. 2025. doi: 10.1109/aibthings66987.2025.11296158.
[13] D. Bouzidi, M. B. Mohamed, M. K. Ibraheem, A. A. J. Al-Abadi, and A. Fakhfakh, "An IoT-Enabled Intelligent Monitoring System for Sustainable Aquaculture," Nov. 2025. doi: 10.20944/preprints202511. 1528.v1.

[14] A. Suaif, "Computer vision with convolutional neural network for product inspection with edge impulse studio and ESP32-Cam microcontroller," in Aip Conf. Proc., vol. 3346, pp. 030008, 2025. doi: 10.1063/5.0287084.
[15] M. González, S. Gutiérrez, R. Espinosa, and H. Ponce, "Deploying Real-Time Speech Recognition on ESP32 Using TinyML and Edge Impulse," Commun. In Comput. And Information Sci., pp. 211-230, 2025. doi: 10.1007/978-3-031-97907-1_17.
[16] V. M, S. K, and A. A, "Design and Implementation of an IoTEnabled Rover with ESP32 CAM for Object Detection, Environmental Monitoring, and Web-Based Control," in 2025 Int. Conf. On Multi-Agent Syst. For Collaborative Intell. (icmsci), pp. 457-465, Jan. 2025. doi: 10.1109/icmsci62561.2025.10894020.
[17] Y. Chang, F. Wu, and H. Lin, "Design and Implementation of ESP32-Based Edge Computing for Object Detection," Sensors, vol. 25, no. 6, pp. 1656, Mar. 2025. doi: 10.3390/s25061656.
[18] L. M. Xin and N. Fadzly, "Development of Machine Learning Wildlife Camera Using ESP32 CAM for Small Mammals," Pertanika J. Of Sci. And Technol., vol. 33, no. 4, Jun. 2025. doi: 10.47836/pjst.33.4.05.
[19] S. S. M. Medicharla, R. Panchal, A. Chougaonkar, and M. K. Mishra, "Edge AI based Predictive Maintenance on 3D printers using ESP32 and Edge Impulse," in 2025 IEEE Int. Conf. On Intelligent Signal Processing and Effective Communication Technologies (inspect), pp. 1-5, Nov. 2025. doi: 10.1109/inspect67393.2025.11350752.
[20] Y. D. A. Chullunquia, C. J. G. Huaman, A. M. Acosta, and E. S. Penadillo, "Embedded Computer Vision Safety System for Freight Elevators Using SSD-MobileNet," in Proc. Of The 5th Laccei Int. Multiconf. On Entrepreneurship, Innovation and Regional Development (leird 2025): "entrepreneurship With Purpose: Social and Technological Innovation in The Age of Ai", 2025. doi: 10.18687/leird2025.1.1.386.
[21] H. G. Gonzales, J. C. Parambita, K. J. U. Emar, A. A. A. Garsuta, and R. J. V. Galarce, "Enhancing Object Detection with FOMO: A User-Friendly Smart Application for Real-Time Tracking of Modern Bus and Seat Availability," Mindanao J. Of Sci. And Technol., vol. 23, no. 2, pp. 193-225, Nov. 2025. doi: 10.61310/mjst. v23i2.2494.
[22] R. R, M. B. D. M, A. G, J. S, M. Y, and A. K. P, "ESP32-Based IoT Aquatic Cleaning Robot with YOLO Edge AI for Plastic Waste Removal," in 2025 Second Int. Conf. On Intelligent Technologies for Sustainable Electric And Commun. Syst. (itech Secom), pp. 1-6, Oct. 2025. doi: 10.1109/itechsecom64750.2025.11307496.
[23] V. Mahali, "Gesture Controlled Rover with Visual Processing Using ESP32-CAM and Edge AI," Int. J. Of Scientific Res. In Eng. And Management, vol. 09, no. 07, pp. 1-9, Jul. 2025. doi: 10.55041/ijsrem51148.
[24] S. M. Kulkarni, S. Umadi, R. M. Amogh, S. S. Airani, and M. M. Raikar, "Grading and Classification of Tomatoes Using ESP32-CAM and Impulse Cloud based IoT Platform for Sustainability," in Procedia Comput. Sci., vol. 260, pp. 938-946, 2025. doi: 10.1016/j.procs.2025.03.277.
[25] B. Yashaswini and M. P. V, "IoT-based Checkout System using ESP32-CAM and Edge Impulse," in 2025 3rd Int. Conf. On Intelligent Cyber Physical Syst. And Internet of Things (icoici), pp. 1375-1379, Sep. 2025. doi: 10.1109/icoici65217.2025.11252702.
[26] V. Sanyod, K. Saelim, K. Chaarmat, and S. Kuankid, "IoT-Based Smart Walking Assistant for Fall Detection in the Elderly," Int. J. Of Online and Biomedical Eng. (ijoe), vol. 21, no. 06, pp. 19-35, May 2025. doi: 10.3991/ijoe. v21i06.54443.
[27] P. Sathish, M. Kalose, K. Purushotham, U. Nidhi, and P. P. Patavardhan, "MediGlove: A Smart Rehabilitation Glove with Gesture-Based IoT Control and Gamified Therapy," Int. J. For Multidisciplinary Res., vol. 7, no. 6, Dec. 2025. doi: 10.36948/ijfmr. 2025.v07i06.62542.

[28] A. Kumar, S. R. Mehta, J. Chawla, and S. Bharti, "Object Detection for Assisting Blind Individuals Using ESP32-CAM and GPS," in 2025 IEEE Int. Students' Conf. On Electrical, Electronics and Comput. Sci. (sceecs), pp. 1-8, Jan. 2025. doi: 10.1109/sceecs64059.2025.10940583.

[29] S. Salerno, "Object Detection with Edge Impulse," Maker Innovations Series, pp. 227-253, 2025. doi: 10.1007/979-8-86881294-1_7.

[30] J. Dhanush and M. A. Rani, "Real-Time Human Fall Detection and Alert System Using Autonomous Embedded Neural Network," Int. J. Of Innovative Sci. And Res. Technol., pp. 574-585, Sep. 2025. doi: 10.38124/ijisrt/25sep394.

[31] L. S. S, B. M. N, D. P. V, and A. O. B, "Review of Real-Time Sign Language Detection, Speech Output Using Esp32-Cam and Wi-Fi," Int. J. Of Scientific Res. In Eng. And Management, vol. 09, no. 11, pp. 1-9, Nov. 2025. doi: 10.55041/ijsrem53795.

[32] B. K. Chaitanya, T. S. Babu, T. Mahveen, S. Sharma, and M. J. P. Aaron, "Smart IoT-Enabled Toll System with Deep Learning-based Vehicle Classification," in 2025 3rd Int. Conf. On Sustainable Computing and Data Communication Syst. (icscds), pp. 601-607, Aug. 2025. doi: 10.1109/icscds65426.2025.11166855.

[33] S. Babar, D. Bhingare, A. Bhujabal, S. Chavan, and S. R. Takale, "Smart Surveillance System for Intrusion Detection and Behaviour Analysis," Int. J. Of Advanced Res. In Sci. Communication and Technol., pp. 306-323, Nov. 2025. doi: 10.48175/ijarsct-29419.

[34] M. Gagnao, M. L. D. Isidro, R. Galaura, J. Segobre, J. M. Gamala, and R. J. Alminaza, "Smart Waste Management System Using Computer Vision," Int. J. Of Scientific Res. In Eng. And Management, vol. 10, no. 6, Jun. 2025. doi: 10.55041/ijsrem64867.

[35] Y. LOBUR, K. P. Vonsevych, and N. V. Bezuglaya, "Spatial identification of manipulable objects for a bionic hand prosthesis," Applied Comput. Sci., vol. 21, no. 1, pp. 18-30, Mar. 2025. doi: 10.35784/acs_6867.

[36] J. Vänskä, "Visitor counter with machine learning," M.S. thesis, Theseus Repository, 2025.

[37] T. Fukushima, K. Itoya, and T. INOUE, "WiFi-communicated Cooperative Platoon Running of Multiple Mobile Robots Using Image Machine Learning with FOMO," in 2025 IEEE Int. Conf. On Real-time Computing and Robotics (rcar), pp. 564-569, Jun. 2025. doi: 10.1109/rcar65431.2025.11139496.

[38] P. Dehigolla, S. Prabudda, and N. Aththanagoda, "AI Automation for Seed Germination with SSD-MobileNetV2 Detection and ESP32 IoT Prototype," in 2026 6th Int. Conf. On Advanced Res. In Computing (icarc), pp. 1-6, Feb. 2026. doi: 10.1109/icarc68737.2026.11453892.

[39] R. R. K, "AI-Powered Multi-Functional Vision Assistant System for Visually Impaired," Int. J. For Res. In Applied Sci. And Eng. Technol., vol. 14, no. 5, pp. 788-798, May 2026. doi: 10.22214/ijraset.2026.81153.

[40] B. A. Anand, R. Manoj, V. S. Mokshitha, M. M. Chowhan, K. J. Moulya, and N. G. Achyutha, "Development of an AI-Based Animal Intrusion Detection System for Agricultural Lands Using ESP32CAM and TinyML," J. Of Scientific Res. And Reports, vol. 32, no. 6, pp. 145-153, Jun. 2026. doi: 10.9734/jsrr/2026/v32i64235.

[41] A. Purnama, Indriani, B. A. Prasetyo, and A. Nugraha, "Edge-Based Person Detection Using MobileNetV2 on ESP32-CAM for Home Surveillance System," Jurnal Sistem Komputer Dan Informatika (json), vol. 7, no. 3, pp. 1171-1182, Mar. 2026. doi: 10.30865/json. v7i3.9619.

[42] S. Tai, "Individual Identification with Deep Learning on Embedded Systems: a Case Study on Cat Faces," M.S. thesis, Theseus Repository, 2026.

[43] A. B. Mulla, S. S. Mane, R. M. Nadaf, T. P. Nakil, and S. S. Mali, "NIGHT VISION PATROLLING ROBOT USING ESP32 AND YOLO," Int. Res. J. Of Mod. In Eng. Technol. & Sci., Feb. 2026. doi: 10.56726/irjmets89466.

[44] 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 (communication And Information Technol. J., pp. 1, Mar. 2026. doi: 10.21512/commit. v20i1.13748.
[45] M. F. A. C. Saputro, S. Hadiyoso, and I. D. Irawati, "Real-Time Hand Gesture-Based Virtual Mouse System Using ESP32-CAM and OpenCV," Jurnal Resti (rekayasa Sistem Dan Teknologi Informasi), vol. 10, no. 1, pp. 53-61, Feb. 2026. doi: 10.29207/resti. v10i1.6609.
[46] A. M. Chaudhry, "RuralRelief DermAI: A Low-Cost ESP32-CAM and Edge Impulse Based Approach for Skin Disease Classification in Rural Pakistan," in Proc. of the 2026 SLR Collection, 2026. [Online]. Available: https://doi.org/10.5281/zenodo.20418571.
[47] R. A. Sabarto, I. Sulistiyowati, S. Syahrorini, and A. Wisaksono, "WebSocket-Based Smart Surveillance Camera for Real-Time Detection of Occupational Health and Safety PPE Non-Compliance in Industrial Areas," Feb. 2026. doi: 10.21070/ups.10109.