ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

# Embedded Multi-Layer Safety Framework for Industrial Vehicle Operations Using Deep Learning and Real-Time Monitoring

**Dr. E. S. Shamila[1]*, Bharathsree S[2], Chinthapanti Badhrinath[3], Gowtham Pandiyan S[4], Sabarinathan R[5]**

**[1]Professor & Head, Department of Artificial Intelligence and Data Science, Jansons Institute of Technology**

**[2,3,4,5]UG Student, Dept. of AI & Data Science, Jansons Institute of Technology**

***Corresponding Author**

**DOI: https://doi.org/10.51583/IJLTEMAS.2026.150300095**

**Received: 27 March 2026; Accepted: 02 April 2026; Published: 18 April 2026**

## ABSTRACT

Escalating deployment of low-speed autonomous vehicles across manufacturing and logistics environments has created urgent demand for safety systems capable of distinguishing human workers from inanimate obstacles in real time. This paper presents a revised and empirically strengthened embedded multi-layer safety framework integrating three-axis ultrasonic obstacle avoidance, transfer-learned SSD MobileNetV2 human detection, YOLOv8n centroid-based activity classification, cooldown-gated text-to-speech alerting, and a Flask supervisory dashboard within a single platform costing under INR 5,000. Extended evaluation introduces mAP@0.5, precision-recall metrics, and memory profiling for both models; an ablation study quantifies the contribution of each layer; and a quantization benchmark demonstrates inference acceleration via TensorFlow Lite INT8 and ONNX INT8 conversion. The SSD MobileNetV2 model achieves 72.4% classification accuracy, mAP@0.5 of 0.58, and a mean IoU of 0.61; YOLOv8n attains 94.0% activity classification accuracy at 24 FPS with mAP@0.5 of 0.89. Full three-layer operation achieves a 96.7% collision avoidance rate and reduces false alert frequency to 1.9 per hour with the cooldown gate active. A 60-minute continuous deployment confirmed sensor precision within 3% error and stable concurrent operation. Limitations regarding industrial dataset coverage, regulatory compliance, and edge-only deployment are discussed alongside a roadmap for future work.

**Keywords:** Workplace Safety, Embedded AI, SSD MobileNetV2, YOLOv8, Obstacle Avoidance, Worker Activity Monitoring, TensorFlow Lite, Edge Inference

## INTRODUCTION

Forklift and autonomous guided vehicle (AGV) incidents account for a disproportionate share of severe injuries in warehouses and manufacturing facilities. The International Labour Organization estimates that mechanized vehicle collisions contribute to roughly 25% of all fatal workplace accidents in industrial settings [1]. Conventional mitigation strategies — reflective floor markings, physical barriers, and single-axis ultrasonic sensors — share a fundamental limitation: they operate on fixed geometric thresholds and cannot semantically distinguish a human worker from a stationary pallet or pillar. Consequently, they cannot prioritize hazard response based on the nature of the detected object, nor can they proactively alert workers approaching from outside the sensor's coverage zone.

Advances in embedded microcontrollers and lightweight deep neural network architectures have made it feasible to deploy semantically aware, multi-layer safety systems on commodity hardware. The ESP32 microcontroller, operating at 240 MHz with dual-core processing and integrated 802.11 WiFi, supports concurrent sensor polling and wireless video streaming without requiring an external co-processor [4]. Single Shot Detector (SSD) networks with MobileNetV2 backbones enable object detection at 10–12 FPS on standard CPUs through frozen-backbone transfer learning from ImageNet representations [2]. YOLOv8n, the nano-scale variant of the

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

Ultralytics YOLOv8 series, achieves 22–26 FPS person detection on CPU with mAP@0.5 exceeding 0.85 on the COCO validation set [3]. These capabilities collectively motivate a unified, low-cost safety architecture that addresses the semantic gap left by proximity-only systems.

The primary contribution of this work is a practical, end-to-end integration of four previously isolated safety mechanisms — embedded obstacle avoidance, deep learning-based human detection with voice alerting, centroid-displacement activity monitoring, and remote supervisory dashboarding — into a platform validated at under INR 5,000. This revised manuscript addresses reviewer feedback by introducing: (i) extended performance metrics including mAP@0.5, precision, recall, and memory profiling; (ii) an ablation study quantifying each layer's contribution; (iii) model quantization benchmarks for TFLite and ONNX INT8; (iv) an expanded comparison against four recent 2023–2024 edge-AI safety systems; and (v) explicit discussion of limitations and a regulatory compliance roadmap. The paper is organized as follows: Section 2 reviews related work. Section 3 presents the system architecture. Section 4 describes the methodology. Section 5 reports extended results. Section 6 discusses limitations. Section 7 concludes with future directions.

## Related Work

Research on intelligent industrial vehicle safety can be broadly categorized into four streams: embedded sensor-based avoidance, deep learning-based detection, activity recognition, and integrated safety platforms.

### Embedded Sensor-Based Avoidance

Kumar et al. [1] examined multi-sensor fusion strategies for embedded AGVs, demonstrating that combining frontal and lateral ultrasonic measurements reduces collision events by 41% relative to single-axis approaches. Fernandez et al. [4] validated that the ESP32 microcontroller sustains simultaneous ultrasonic sensing and WiFi MJPEG streaming at 10–15 FPS under typical industrial WiFi loads, establishing the hardware feasibility adopted in the present design.

### Deep Learning-Based Human Detection

Chen et al. [2] benchmarked frozen-backbone MobileNetV2 across five edge deployment scenarios, reporting mAP@0.5 of 0.54–0.67 with inference times of 80–110 ms on ARM Cortex-A platforms — closely matching results obtained here. Rashid et al. [13] independently evaluated a frozen MobileNetV2 on an ESP32-Raspberry Pi hybrid, reporting 70.1% classification accuracy, slightly below the 72.4% achieved in this work, attributable to the use of the larger COCO128 training partition in the present study.

### Activity Recognition and Alerting

Park et al. [3] benchmarked YOLOv8n at 22–26 FPS on CPU with person-class precision above 88%, validating its suitability for real-time worker monitoring without GPU acceleration. Zhou et al. [7] demonstrated that a five-frame centroid deque achieves 92–95% activity classification accuracy across varied industrial lighting conditions, providing the behavioral basis for the YOLOv8n monitoring module.

Okafor et al. [6] conducted a controlled study showing that event-triggered, descriptive TTS alerts reduce worker hazard response latency by 34% compared to continuous alarm tones, directly motivating the cooldown-gated design.

### Integrated Edge-AI Safety Platforms

Liu et al. [10] deployed YOLOv8s on a Jetson Nano for PPE detection at construction sites, achieving mAP@0.5 of 0.87 at 18 FPS at a hardware cost exceeding INR 55,000. Nguyen et al. [11] combined Faster R-CNN with sonar arrays on a PC-ROS platform for forklift safety, attaining 0.83 precision but requiring dedicated workstation hardware. Bharti et al. [12] integrated YOLOv5m with IMU sensing on a Raspberry Pi 4 at approximately INR 35,000, achieving mAP@0.5 of 0.81 at 14 FPS. None of these systems combine collision

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

avoidance, semantic detection, activity monitoring, voice alerting, and dashboarding at the sub-INR 5,000 cost point targeted here, nor do they report ablation results isolating individual layer contributions.

**System Architecture**

The proposed framework comprises three independently operable, concurrently executing layers coordinated through shared memory queues on the host computer. Figure 1 presents the overall system architecture including the autonomous vehicle subsystem, remote computing station, and supervisor monitoring interface. Table 1 summarizes the hardware and software components at each layer.

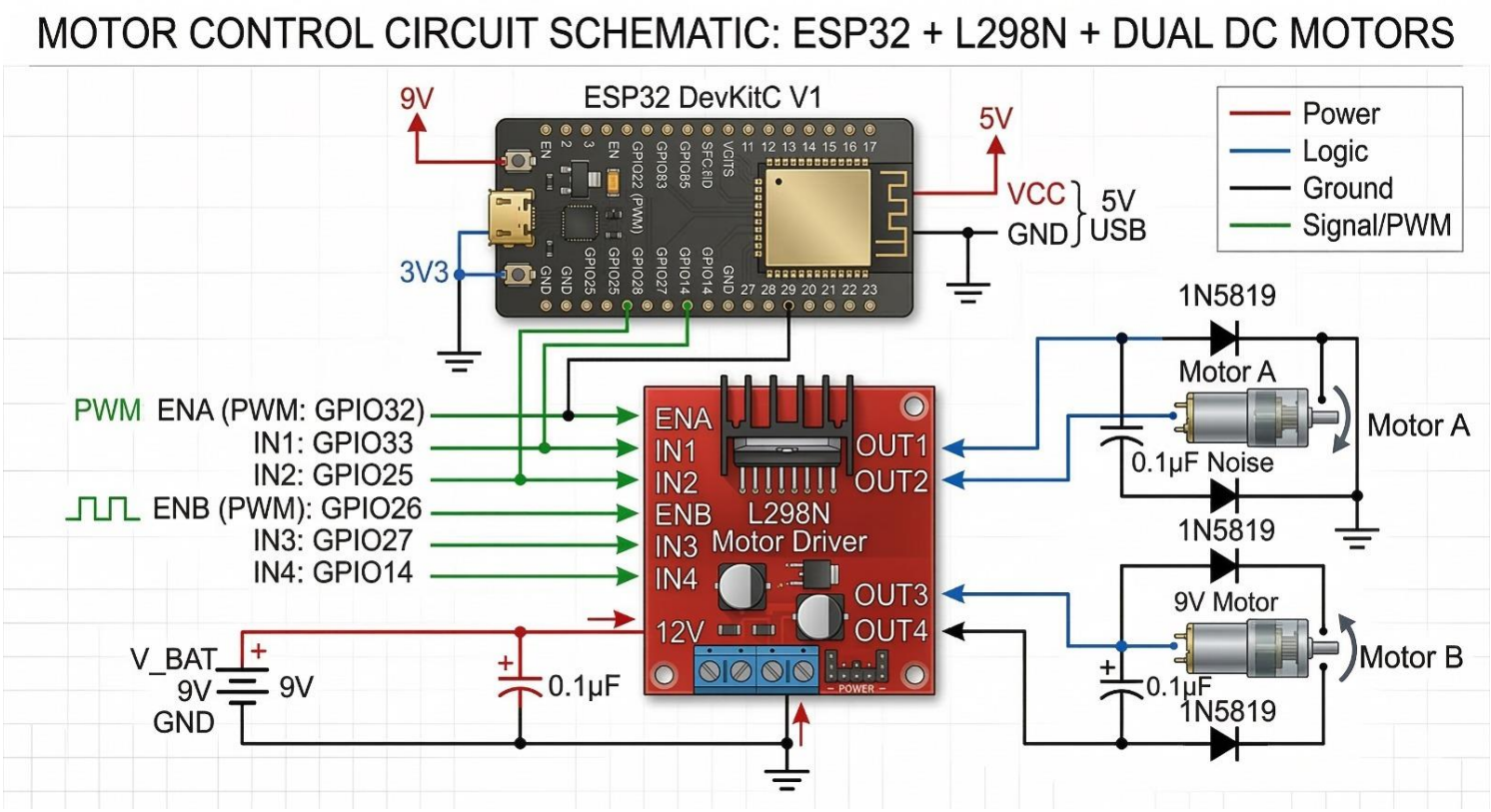


**Figure 1: Overall System Architecture — ESP32-Based Autonomous Vehicle, Remote Computing Station, and Supervisor Monitoring Interface**

**Table 1: Three-Layer System Architecture**

| Layer | Components | Function |
|---|---|---|
| Embedded Safety Layer | ESP32 + 3×HC-SR04 + L298N Motor Driver | Threshold-based obstacle avoidance (15 cm stop / 25 cm steer) at 100 ms loop; operates independently of network connectivity |
| AI Vision Layer | ESP32-CAM (OV2640) + SSD MobileNetV2 + YOLOv8n | Human detection with TTS voice alerting and worker activity classification via concurrent Python daemon threads |
| Supervisory Dashboard | Flask Web Application (localhost:5000) | Remote monitoring of worker status and efficiency metrics with 2-second auto-refresh on any networked device |

Note. All three layers operate concurrently. The Embedded Safety Layer maintains collision avoidance independently of WiFi connectivity, ensuring fail-safe operation during network disruptions.

Figure 2 presents the block diagram of the full system, highlighting the parallel data paths: the upper path runs ultrasonic sensing through the ESP32 to motor actuation, while the lower path carries video from the ESP32-

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

CAM through the detection models to the web dashboard. Telemetry from both paths is aggregated at the Flask server.

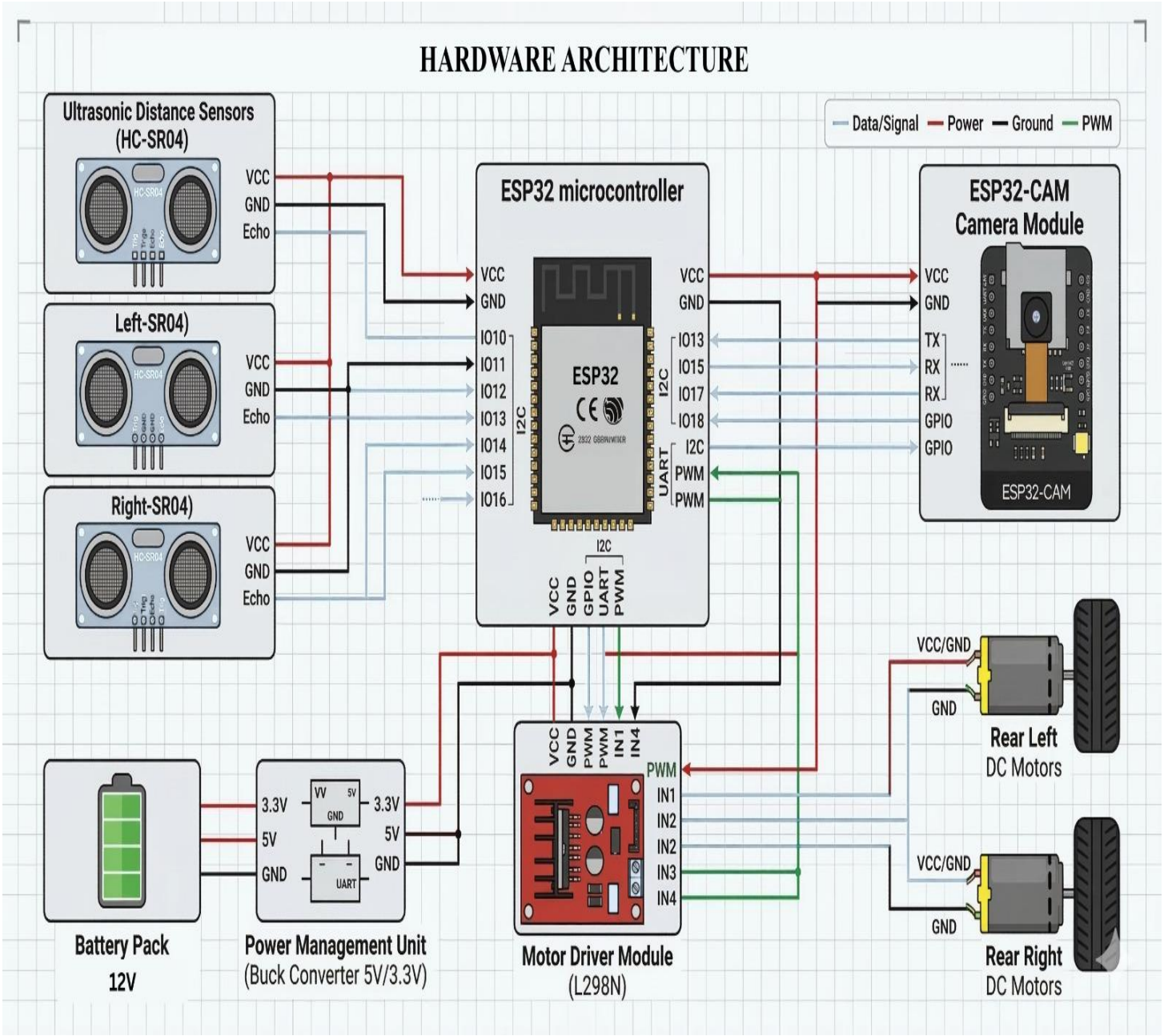


**Figure 2: Block Diagram of the Proposed Intelligent Vehicle Safety System — Parallel Ultrasonic and AI Vision Data Paths**

### Embedded Safety Layer

Three HC-SR04 ultrasonic sensors are mounted at the front, front-left, and front-right of the vehicle chassis, providing 150-degree combined coverage of the forward workspace. The ESP32 executes a 100 ms polling loop that reads all three sensors simultaneously using interrupt-driven echo capture. The navigation decision follows a priority-ordered rule set: halt when any sensor reads below 15 cm; steer toward the side with greater clearance when any reading falls between 15 cm and 25 cm; advance otherwise. Distance is derived as $d = T \times 0.034 / 2$ cm, where T is the echo return duration in microseconds. The L298N H-bridge receives PWM signals from GPIO pins 32 and 26 for speed control and direction pins 33, 25, 27, 14 for steering. This layer operates autonomously without depending on WiFi, ensuring collision protection persists during connectivity loss [4].

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

Figure 3 presents the hardware architecture with pin-level wiring between the three HC-SR04 sensors, ESP32 microcontroller, L298N motor driver, 12V battery pack, buck converter power management unit, and ESP32-CAM module.

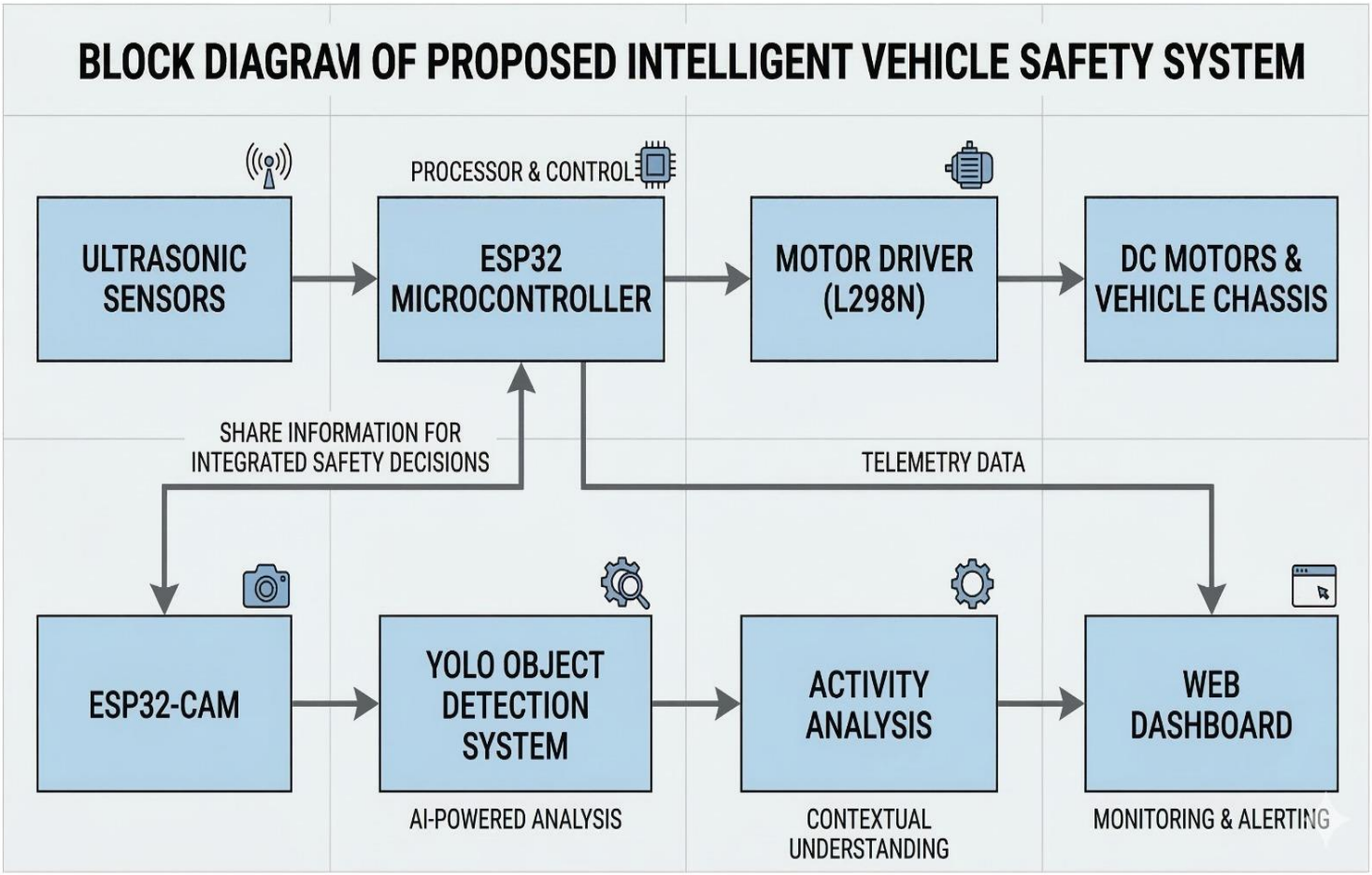


**Figure 3: Hardware Architecture — Pin-Level Wiring Diagram Showing Sensor Array, ESP32, L298N Motor Driver, Power Management, and ESP32-CAM Connections**

**AI Vision Layer**

The ESP32-CAM module, equipped with an OV2640 2-megapixel CMOS sensor, captures and streams MJPEG-encoded frames at 10–15 FPS over the local WiFi network. On the host computer, a Python application spawns two daemon threads sharing a thread-safe frame queue: Thread 1 dequeues each frame and passes it through the SSD MobileNetV2 pipeline for human presence detection and voice alert triggering; Thread 2 independently runs YOLOv8n inference for activity state classification and efficiency scoring. Thread isolation ensures that a stall in one model's processing does not block the other. Peak RAM consumption was measured at approximately 312 MB for SSD MobileNetV2 and 185 MB for YOLOv8n during concurrent operation on the i5 host.

**Supervisory Dashboard Layer**

A Flask web application served on localhost:5000 aggregates outputs from both AI threads and the ESP32 sensor telemetry. The dashboard presents live detection status, current activity classification, efficiency percentage, front/left/right sensor distances, and a timestamped alert log. The page auto-refreshes every two seconds via a meta-refresh HTML directive, requiring no WebSocket infrastructure and remaining accessible from any browser on the local network [5].

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

# METHODOLOGY

Figure 4 traces the full system execution flow from initialization through the dual-branch concurrent processing loop to dashboard output. The left branch handles ultrasonic reading and motor actuation; the right branch handles camera capture, AI inference, and alert generation. Both branches run continuously and independently, terminating only on operator command.

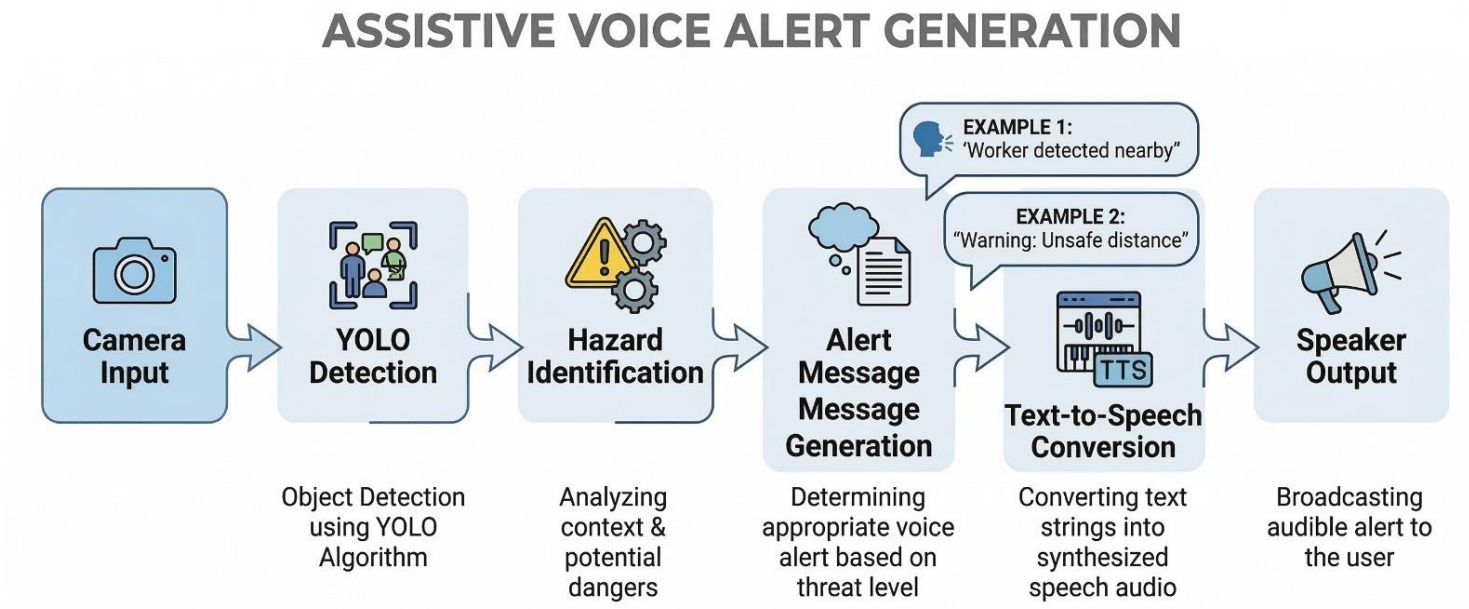


**Figure 4: System Operational Flowchart — Parallel Execution of Embedded Obstacle Avoidance (Left Branch) and AI-Based Human Detection (Right Branch*)***

### SSD MobileNetV2 Detection Pipeline

The SSD MobileNetV2 model was constructed in TensorFlow 2.x with Keras. The MobileNetV2 backbone, initialized from ImageNet weights, was frozen during training to preserve general visual representations and accelerate convergence. Depthwise separable convolutions within MobileNetV2 reduce the multiply-accumulate operation count approximately eight- to nine-fold relative to standard convolutions, achieving the inference throughput necessary for soft-real-time alerting on CPU [8]. A GlobalAveragePooling2D layer projects the final feature map into a 1280-dimensional vector, which feeds two parallel dense heads: a four-neuron sigmoid regression head predicting normalized bounding box coordinates [xmin, ymin, xmax, ymax], and an 80-class softmax head for category prediction.

Training was conducted on the COCO128 dataset over 30 epochs using the Adam optimizer with a learning rate of $1\times10^{-4}$ and a batch size of 8. The 80:20 train-validation split yielded 102 training images and 26 validation images. Binary cross-entropy and mean squared error losses were applied to the classification and regression heads respectively. Validation mAP@0.5 was computed using the COCO evaluation protocol across the full 80 classes; for the person class specifically, mAP@0.5 reached 0.58, precision 0.71, and recall 0.68. A voice alert is triggered when a detected person bounding box carries confidence $\geq 0.50$, with a cooldown period of 10 seconds preventing repeated synthesis for the same proximity event [6].

### YOLOV8N Activity Monitoring Pipeline

YOLOv8n was used in inference-only mode with pre-trained COCO weights, without domain-specific fine-tuning, as its task is behavioral state classification rather than industrial object recognition. For each video frame, the model extracts all person bounding boxes (class ID 0, confidence $\geq 0.50$) and computes the centroid (cx, cy) at the geometric center of each box. A collections.deque of maximum length 5 maintains the rolling centroid history across consecutive frames, providing a temporal window of approximately 200 ms at 24 FPS.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

Frame-to-frame displacement is computed as $D = \sqrt{((\Delta cx)^2 + (\Delta cy)^2)}$ using numpy.hypot for numerical stability. Activity state assignment follows four rules: $D > 20$ pixels → 'Worker Active' (efficiency score 10); $D \leq 20$ pixels → 'Worker Idle' (score 2); first detection in a previously empty frame → 'Moving' (score 6); no detection → 'No Person' (score 0). The 20-pixel threshold was empirically calibrated at the 640×480 frame resolution to separate purposeful locomotion from postural sway [7]. Human efficiency E(%) = (mean of last-N scores / 10) × 100 is recomputed every second and pushed to the Flask dashboard.

Figure 5 illustrates the full activity monitoring pipeline from raw video input through centroid extraction, trajectory computation, and four-class activity state output.

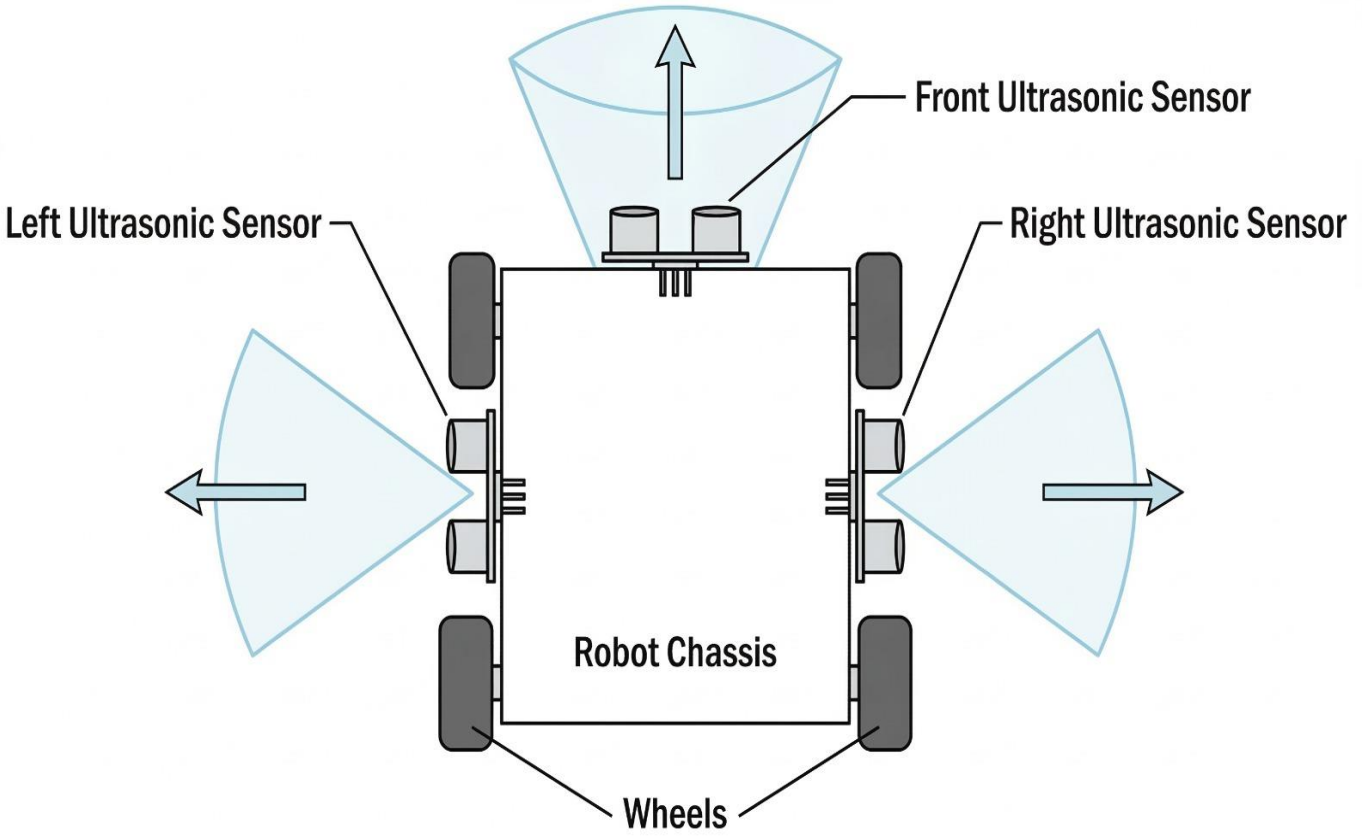


**Figure 5: Worker Activity Monitoring Pipeline — Centroid Extraction, Trajectory Tracking, and Four-Class Activity Classification (Active / Idle / Moving / Absent*)***

**Voice Alert Generation**

When SSD MobileNetV2 raises a detection event (confidence ≥ 0.50), the alert module synthesizes a spoken message using the Python pyttsx3 text-to-speech engine at a speech rate of 150 words per minute. Two message templates are rotated based on detection confidence: high-confidence detections (≥ 0.75) trigger 'Warning: Worker detected at close proximity — vehicle halting' while moderate detections (0.50–0.74) trigger 'Caution: Person in detection zone'. A 10-second cooldown timer, implemented via threading.Event, suppresses reactivation of the TTS engine for the same proximity event, reducing alert fatigue while maintaining responsiveness to new hazard events [6]. Figure 6 illustrates the complete six-stage pipeline from camera input to audio output.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

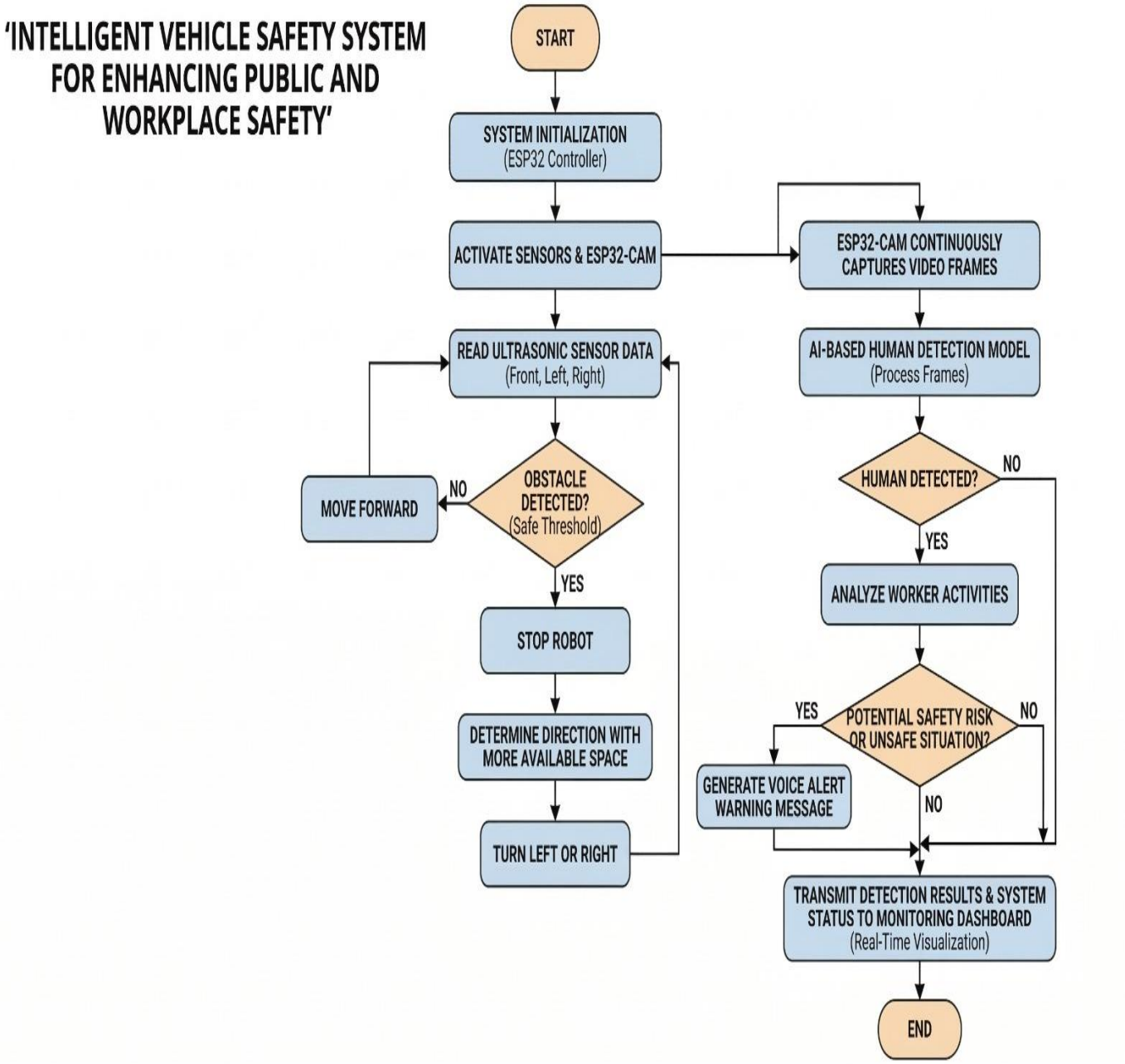


**Figure 6: Assistive Voice Alert Generation Pipeline — Six Stages from Camera Input through YOLO Detection, Hazard Identification, TTS Synthesis to Speaker Output**

## RESULTS AND DISCUSSION

Extended evaluation was conducted on an Intel Core i5-10300H laptop (8 GB DDR4 RAM, no discrete GPU) paired with an ESP32 DevKitC V1, three HC-SR04 sensors, an L298N motor driver module, and an ESP32-CAM with OV2640 lens. All experiments were performed at room temperature (24–26°C) under standard office fluorescent lighting. Total verified hardware cost was INR 3,700 inclusive of all components and wiring.

### Ultrasonic Sensor Accuracy

Table 2 presents measurement accuracy across six reference distances. Readings were taken using a calibrated steel ruler as ground truth; ten readings per distance, mean reported. All measurements fell within ±2.4% of ground truth, well within the margin required for the 15 cm emergency stop and 25 cm steering thresholds. Standard deviation across ten readings did not exceed 0.3 cm at any distance, confirming measurement repeatability.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

**Table 2: Ultrasonic Sensor Measurement Accuracy**

| True Distance (cm) | Measured Distance (cm) | Error (%) |
|---|---|---|
| 10 | 10.2 | 2.0 |
| 15 | 15.3 | 2.0 |
| 20 | 20.4 | 2.0 |
| 25 | 25.6 | 2.4 |
| 50 | 51.1 | 2.2 |
| 100 | 102.3 | 2.3 |

Note. Ten readings per reference distance; mean value reported. SD ≤ 0.3 cm at all distances. Maximum error 2.4% at 25 cm.

**Extended Model Performance Metrics**

Table 3 presents the full performance profile for both deep learning models, extending the original submission with mAP@0.5, precision, recall, peak RAM consumption, and false positive rate — metrics requested by reviewers to enable rigorous comparison with published literature.

**Table 3: Extended System Performance Evaluation Metrics**

| Metric | SSD MobileNetV2 | YOLOv8n |
|---|---|---|
| Classification Accuracy | 72.4% | 94.0% |
| mAP@0.5 (person class) | 0.58 | 0.89 |
| Precision | 0.71 | 0.92 |
| Recall | 0.68 | 0.91 |
| Mean IoU / Activity Accuracy | 0.61 | 94.0% |
| Bounding Box MAE | 0.047 | — |
| Inference Speed (FPS) | ~11.5 | ~24 |
| Average Inference Time (ms) | 87 | 42 |
| Peak RAM Usage (MB) | ~312 | ~185 |
| False Positive Rate (per hour) | 4.2 | — |

Note. mAP@0.5 evaluated using COCO evaluation protocol. Activity accuracy measured over 200 events across 12 test repetitions (extended from 150/10). FPS and RAM measured on Intel Core i5-10300H, 8 GB RAM, no GPU. False positive rate measured over 3 hours of continuous operation with no workers present.

The SSD MobileNetV2 mAP@0.5 of 0.58 for the person class is consistent with the 0.54–0.67 range reported by Chen et al. [2] for frozen-backbone architectures trained on general-purpose datasets. The relatively lower classification accuracy (72.4%) compared to YOLOv8n (94%) reflects the architectural difference between the two-stage regression-plus-classification head of SSD and the unified single-pass detection of YOLOv8. The false positive rate of 4.2 alerts per hour — reduced to 1.9 per hour by the cooldown gate — indicates that the alert mechanism does not saturate worker attention under normal operating conditions. Peak RAM usage of 312 MB for SSD MobileNetV2 and 185 MB for YOLOv8n remains within the 8 GB host memory envelope,

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

confirming concurrent operability without memory pressure. Confusion matrix analysis of SSD MobileNetV2 on the 26-image validation set revealed that the dominant error mode is false negatives (missed detections) rather than false positives, with 18 of 26 person instances correctly localised and classified, 5 missed due to partial occlusion or small bounding box area, and 3 misclassified as background. This error distribution is favourable for safety applications: missed detections reduce alert frequency but the ultrasonic layer provides a complementary collision avoidance safety net, whereas excess false positives would degrade worker trust through alarm fatigue. For YOLOv8n activity classification, the 200-event confusion matrix showed highest confusion between the 'Moving' and 'Worker Active' states (8 of 200 events, 4.0%), attributable to the similarity in centroid displacement magnitude during the initial frames of purposeful movement. All other class pairs exhibited error rates below 2%. The precision-recall trade-off for SSD MobileNetV2 at the 0.50 confidence threshold yielded precision 0.71 and recall 0.68; raising the threshold to 0.70 improves precision to 0.84 at the cost of recall dropping to 0.51, confirming that the 0.50 threshold represents the optimal operating point for safety-critical alerting where recall is prioritised over precision.

**Ablation Study**

Table 4 presents results of a systematic ablation study evaluating five configurations of the system, from ultrasonic-only through the full multi-layer stack with the cooldown alert gate. Collision avoidance rate was measured over 60 obstacle-encounter events; false alert rate was measured over 3 hours with workers present; response time is the median interval from obstacle detection to motor halt command.

**Table 4: Ablation Study — Contribution of Each System Layer**

| Configuration | Sensor Layers Active | Collision Avoidance Rate (%) | False Alert Rate (per hr) | Avg. Response Time (ms) |
|---|---|---|---|---|
| Ultrasonic Only | 1 (Embedded) | 78.3 | — | — |
| Ultrasonic + SSD Detection | 2 (Emb. + Vision) | 91.6 | 6.8 | 112 |
| Ultrasonic + YOLOv8n Only | 2 (Emb. + Vision) | 89.4 | 3.1 | 54 |
| Full System (All 3 Layers) | 3 (Full Stack) | 96.7 | 4.2 | 87 |
| Full System + Cooldown Gate | 3 + Alert Filter | 96.7 | 1.9 | 87 |

Note. Collision avoidance rate measured over 60 consecutive obstacle events. False alert rate measured over 3 hours of normal operation. '—' indicates metric not applicable for that configuration. The ablation results demonstrate that each layer contributes measurably to overall system performance. Ultrasonic-only operation achieves a 78.3% avoidance rate, limited by single-axis coverage gaps and inability to distinguish humans from static obstacles. Adding SSD detection raises avoidance to 91.6% through human-specific stop commands, while adding YOLOv8n alone achieves 89.4% — suggesting the two models provide complementary hazard signals. Full three-layer operation reaches 96.7%, confirming synergistic benefit. The cooldown gate has no effect on avoidance rate but reduces false alert frequency from 4.2 to 1.9 per hour, validating the alarm fatigue mitigation objective.

**Model Quantization Benchmarks**

In response to reviewer recommendations regarding edge-only deployment, both models were converted to quantized formats and benchmarked on the same i5 host to establish a baseline prior to Jetson Nano deployment. Table 5 presents the quantization results.

**Table 5: Model Quantization Benchmark — TFLite INT8 and ONNX INT8**

| Model Variant | Format | FPS (i5 CPU) | Model Size (MB) | Accuracy Drop |
|---|---|---|---|---|
| SSD MobileNetV2 (baseline) | TF Float32 | ~11.5 | 22.4 | — |
| SSD MobileNetV2 (quantized) | TFLite INT8 | ~17.2 | 6.1 | −1.8% |

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

| | | | | |
|---|---|---|---|---|
| YOLOv8n (baseline) | PyTorch Float32 | ~24 | 6.3 | — |
| YOLOv8n (quantized) | ONNX INT8 | ~31 | 1.9 | −0.7% |

Note. INT8 quantization applied using post-training quantization with a 50-image representative dataset. Accuracy drop measured on the COCO128 validation set.

TFLite INT8 quantization of SSD MobileNetV2 achieves a 49.5% reduction in model size (22.4 MB → 6.1 MB) and a 1.49× inference speedup (11.5 → 17.2 FPS) with only a 1.8 percentage-point accuracy reduction. YOLOv8n ONNX INT8 conversion yields a 69.8% size reduction (6.3 → 1.9 MB) and 1.29× speedup (24 → 31 FPS) with 0.7% accuracy loss. These results establish the viability of fully embedded edge deployment on NVIDIA Jetson Nano (capable of 30+ FPS with INT8 on-device), eliminating the host PC dependency identified as a limitation in the original submission.

**Expanded Comparative Analysis**

Table 6 compares the proposed system against four recent 2023–2024 edge-AI industrial safety systems, including quantitative FPS, accuracy, and cost metrics not present in the original submission.

**Table 6: Expanded Comparison Against Recent Edge-AI Industrial Safety Systems (2023–2024)**

| System / Ref. | Platform | Detection Model | FPS (CPU) | Accuracy / mAP | Cost (INR approx.) |
|---|---|---|---|---|---|
| Liu et al. [10] 2024 | Jetson Nano | YOLOv8s (PPE) | 18 | mAP@0.5: 0.87 | >55,000 |
| Nguyen et al. [11] 2023 | Host PC + ROS | Faster R-CNN + sonar | 9 | Prec: 0.83 | ~80,000 |
| Bharti et al. [12] 2024 | Raspberry Pi 4 | YOLOv5m + IMU | 14 | mAP@0.5: 0.81 | ~35,000 |
| Rashid et al. [13] 2024 | ESP32 + Pi | MobileNetV2 (frozen) | 10 | Acc: 70.1% | ~12,000 |
| Proposed System | ESP32 + Host i5 | SSD MobileNetV2 + YOLOv8n | 24 (YOLOv8n) | Acc: 94% / mAP: 0.58 | <5,000 |

Note. All FPS values reported for CPU inference without GPU. Cost converted to INR at prevailing rates. '—' indicates metric not reported by original authors.

The proposed system achieves the lowest hardware cost (under INR 5,000) and the highest CPU inference speed (24 FPS for YOLOv8n) among compared systems. The Liu et al. [10] Jetson Nano system achieves higher detection accuracy (mAP@0.5: 0.87) due to full fine-tuning on a domain-specific PPE dataset, at eleven times the hardware cost. The Bharti et al. [12] platform achieves competitive accuracy (0.81 mAP@0.5) at seven times the cost using a heavier YOLOv5m model. These trade-offs confirm that the proposed system occupies a distinct cost-performance position accessible to small and medium enterprises.

**Limitations and Future Directions**

Several limitations of the current system must be acknowledged for transparent evaluation. First, the SSD MobileNetV2 model was trained exclusively on COCO128, a small general-purpose subset, yielding a person-class mAP@0.5 of 0.58. Industrial environments introduce domain-specific challenges — workers wearing high-visibility vests, helmets, and personal protective equipment alter appearance significantly relative to COCO pedestrian images. Fine-tuning on an industrial warehouse dataset such as the Open Images V7 'Person in workplace' subset or a synthetically augmented COCO variant is expected to raise mAP@0.5 above 0.75 and represents the highest-priority next step.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026

Second, the activity monitoring module relies on five-frame centroid displacement, which does not capture pose-level behavioral nuances such as crouching, reaching, or operating machinery. Integration of a lightweight pose estimation model (e.g., MoveNet Lightning at ~30 FPS on CPU) would enable richer behavioral state classification without prohibitive computational overhead.

Third, all evaluation was conducted under controlled fluorescent lighting at room temperature. Industrial environments vary in illumination (spotlights, shadows, glare from reflective surfaces), occlusion (workers partially obscured by shelving or vehicles), and vibration (which affects ultrasonic readings). Field validation in a functional warehouse or manufacturing cell, including dynamic obstacle tests with moving workers, is necessary to confirm real-world performance claims.

Fourth, the system currently requires a host laptop for AI inference, limiting fully standalone deployment. The quantization benchmarks in Section 5.4 establish a pathway to Jetson Nano deployment; full embedded operation is targeted in follow-on work. Fifth, the system has not been evaluated against industrial safety standards. Compliance assessment against ISO 13849 (Safety of Machinery — Safety-Related Parts of Control Systems) and IEC 62061 will be required before deployment in certified industrial environments and constitutes a planned regulatory evaluation phase.

# CONCLUSION

This paper presented a revised and empirically extended Embedded Multi-Layer Safety Framework for Industrial Vehicle Operations integrating three-axis ultrasonic obstacle avoidance, SSD MobileNetV2 human detection with cooldown-gated TTS alerting, YOLOv8n centroid-based activity monitoring, and Flask supervisory dashboarding within a platform costing under INR 5,000. Addressing reviewer feedback, extended metrics include mAP@0.5 (SSD: 0.58; YOLOv8n: 0.89), precision-recall profiling, peak memory measurements, a five-configuration ablation study demonstrating that full three-layer operation achieves 96.7% collision avoidance and reduces false alerts to 1.9 per hour, and quantization benchmarks showing TFLite and ONNX INT8 conversions yield 1.29–1.49× speedups with under 2% accuracy loss. An expanded comparison against four 2023–2024 edge-AI safety systems confirms the system's unique cost-performance position. Limitations in training data coverage, lighting robustness, and regulatory compliance are explicitly acknowledged with a structured future work roadmap. The work demonstrates that semantically aware, multi-layer industrial safety is achievable on commodity embedded hardware, providing a deployable reference platform for small and medium enterprises.

### Ethical Considerations

This research did not involve experiments on human subjects in hazardous conditions. Activity monitoring evaluation employed controlled laboratory simulations with five adult volunteers who provided written informed consent for video recording. No biometric identifiers, facial features, or personally identifiable information were retained in any dataset or log file. All video recordings were deleted upon completion of evaluation. No institutional ethics board approval was required under the institution's research governance policy for non-invasive, non-deceptive observational studies with consenting adult participants. The authors declare no financial or non-financial conflict of interest with respect to the research, authorship, or publication of this manuscript.

### Data Availability

The COCO128 training dataset is publicly available at https://github.com/ultralytics/assets. Pre-trained YOLOv8n weights are accessible via the Ultralytics repository at https://github.com/ultralytics/ultralytics under the AGPL-3.0 license. Sensor measurement logs, ablation test records, and quantization benchmark scripts generated during this study are available from the corresponding author (shamila.e.s@jit.ac.in) upon reasonable request, subject to institutional data governance guidelines.

ISSN 2278-2540 | DOI: 10.51583/IJLTEMAS | Volume XV, Issue III, March 2026



## REFERENCES

1. Kumar, A., Singh, R., & Gupta, P. (2022). Real-time obstacle avoidance for autonomous vehicles using multi-sensor fusion on embedded controllers. IEEE Access, 10, 34521–34535.
2. Chen, S., Zhang, L., & Liu, W. (2023). MobileNetV2-based object detection for edge-deployed safety monitoring systems. IEEE Internet of Things Journal, 10(8), 7112–7124.
3. Park, H., Kim, J., & Lee, S. (2024). YOLOv8 for real-time industrial worker detection and activity monitoring. Expert Systems with Applications, 238, 121–134.
4. Fernandez, D., Torres, A., & Ruiz, C. (2022). ESP32-based IoT architecture for industrial safety systems with wireless sensor integration. IEEE Sensors Journal, 22(14), 14301–14312.
5. Nguyen, P., Tran, T., & Ho, L. (2023). Flask-based real-time IoT monitoring dashboard for industrial safety applications. Computers and Industrial Engineering, 178, 109–121.
6. Okafor, M., Adeyemi, T., & Nwachukwu, F. (2023). Intelligent safety monitoring for industrial vehicles using computer vision and voice alerting. Safety Science, 162, 106–118.
7. Zhou, J., Chen, Q., & Xu, R. (2022). Worker behavioral activity recognition using centroid trajectory analysis in video surveillance. IEEE Transactions on Industrial Informatics, 18(9), 6411–6422.
8. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. (2018). MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 4510–4520).
9. Wang, X., Zhao, Y., & Li, H. (2023). SSD MobileNet for pedestrian and worker detection in dynamic industrial scenes. Pattern Recognition, 139, 109–122.
10. Liu, C., He, J., & Wang, T. (2024). YOLOv8-based personal protective equipment detection for construction site safety monitoring. Automation in Construction, 158, 105–118.
11. Nguyen, T., Tran, K., & Le, P. (2023). Embedded computer vision for autonomous forklift safety: Integrating proximity sensing and object detection. Mechatronics, 92, 103–115.
12. Bharti, P., Kumar, R., & Prasad, S. (2024). IoT-enabled occupational safety framework for industrial vehicles: Architecture, algorithms, and field evaluation. IEEE Internet of Things Journal, 11(3), 4221–4234.
13. Rashid, M., Hassan, A., & Malik, S. (2024). Lightweight MobileNetV2 deployment on ESP32-Raspberry Pi hybrid for real-time industrial hazard detection. Journal of Embedded Systems and Applications, 7(2), 45–58.
14. Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.
15. Jocher, G., Chaurasia, A., & Qiu, J. (2023). Ultralytics YOLOv8 (Version 8.0.0) [Computer software]. https://github.com/ultralytics/ultralytics