International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2196-220
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0294

Rapidresq: AI-Based Traffic Accident Detection and IoT-Enabled Emergency Notification Grid

Dr. J. Nandhini[1], A Akalya [2], R Arya[3] , V Deeksha[4], N Kishorekumar[5]
[1]Professor, Dept. of ECE, Jai Shriram Engineering College., Tirupur, Tamilnadu, India,
[2,3,4,5] UG Scholar, Dept. of ECE, Jai Shriram Engineering College., Tirupur, Tamilnadu, India
Email ID: *aryaramesh0811@gmail.com[3]*

Abstract
Road traffic accidents continue to be a major global public safety concern, especially in areas with inadequate communications infrastructure. Conventional surveillance relies on manual observation, which causes response delays and human fatigue. RapidResQ, an autonomous real-time collision detection system utilizing edge computing on the ESP32-CAM platform, is presented in this paper. To minimize latency and eliminate the need for GPU-class edge hardware, cloud-assisted inference is made possible by a quantized Fast-CNN derived from MobileNetV2 that is deployed via Google Teachable Machine and accessed over local Wi-Fi.. False positives are decreased from 18.3% to 6.8% using a two-tier validation mechanism that combines AI visual classification with analog sensor threshold verification. A hybrid architecture is used for emergency alerting: GSM-enabled SMS with voice telephony for rural areas and IoT cloud messaging for urban areas. With an alert latency of less than five seconds, evaluation attains 90.5% classification accuracy.

Keywords: *Embedded AI, Traffic Accident Detection, ESP32-CAM, MobileNetV2, Fast-CNN, Edge Computing, IoT Emergency Alert, GSM Telephony, False Positive Suppression.*

1. Introduction
Clinical evidence consistently shows that victims who receive definitive medical intervention within the first sixty minutes of a traumatic event exhibit significantly improved survival rates and reduced long-term disability, which emphasizes the importance of the "golden hour" in trauma care. Any automated accident detection and alerting system meant for real-world deployment must adhere to a strict latency constraint due to this time-sensitive reality. In addition, even highly staffed monitoring centers are unable to ensure constant surveillance coverage across all active channels at any given time due to the sheer volume of simultaneous camera feeds in dense urban deployments surpassing practical human cognitive capacity. Another risk associated with vision-only approaches is that they are still vulnerable to environmental deterioration, such as dim lighting, motion blur, occlusion, and bad weather. These factors can lead to misclassification and false emergency alerts, which erode operator confidence and waste emergency resources. This problem has an equally important economic component; GPU-accelerated edge platforms that can run heavyweight inference models have unit costs that make widespread deployment unaffordable for developing-nation municipalities. Current IoT-based emergency notification systems exacerbate these constraints by sticking to a single communication paradigm without flexible fallback options, rendering them inoperable in places with spotty or nonexistent network coverage. As a result, there is a significant systems-level gap in the literature: no widely used solution covers the entire end-to-end pipeline from emergency notification to real-time accident detection in a single embedded, affordable, connectivity-resilient platform that can be deployed in both urban and rural areas. RapidResQ is presented in this paper with three contributions:

- Lightweight cloud-assisted inference: The ESP32-CAM uses local Wi-Fi to query a quantized MobileNetV2-derived Fast-CNN that has been trained and hosted via Google Teachable Machine. This allows for real-time classification without requiring GPU-class hardware or on-device TFLite execution.
- In contrast to traditional deployment pipelines that require specialized knowledge for model

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2196-220
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0294

retraining and redeployment, the Teachable Machine framework lowers the barrier to adaptation across a variety of traffic environments by allowing non-specialist operators to customize the system in the field.

- The system introduces a physically grounded validation layer that significantly lowers false activation rates a known drawback of purely vision-based systems operating under real-world environmental variability-by requiring corroborating evidence from vibration and impact transducers in addition to visual confirmation.
- Two-tier false-positive suppression: Before an alert is raised, AI visual classification and analog sensor verification are combined.
- Stratified communication using hybrid technology: GSM SMS/voice fallback for rural areas and adaptive IoT cloud messaging for urban settings.
- In previous embedded detection literature, single-paradigm communication stacks render systems inoperable in situations of reduced connectivity. This adaptive dual-mode architecture fills that gap.

The suggested system is built entirely on open and freely available platforms, such as the Google Teachable Machine, Arduino IDE, and commodity GSM modules. This ensures that independent research groups can replicate the system and that local governments with limited resources can deploy it without relying on proprietary ecosystems.

2. Related Work

Accelerometers and impact sensors were used in early methods. These systems show high false positive rates (~21.7%) from speed bumps, variations in the road surface, and braking events unrelated to collisions, despite being computationally lightweight[12][15][16]. Tan and colleagues [6] reported that Cloud CNN systems could achieve 92–94% accuracy with a latency of 10–15 s. Adewopo et al. [8] suggested ensemble deep learning, which produced excellent results at a high computational expense. Both cannot be used in rural areas since they require constant high-bandwidth connectivity[8][18][20]. YOLOv11-based detection with approximately 91% accuracy was demonstrated by Arefin et al. [7]. However, the memory requirements (~25 MB+) of YOLO variants surpass the capabilities of ESP32-class microcontrollers[7][17]. RapidResQ develops IoT-based accident detection on ESP32 by combining visual AI, multi-modal sensor fusion, and adaptive dual-mode communication into a single framework that covers

- microcontroller-class inference
- false positive suppression
- heterogeneous network communication.

Reddy et al. [12] first proposed IoT-based accident detection on ESP32[19][20].

3. System Architecture

RapidResQ serves as an inexpensive system for emergency alerts and collision detection. To minimize reliance on wide-area internet connectivity and maintain real-time responsiveness, all sensor fusion, decision logic, and alert dispatch are carried out locally on the NodeMCU, while AI inference is offloaded to a locally accessible cloud endpoint hosted via Google Teachable Machine.. Data flow pathways between components and the system architecture are depicted in Fig. 1.

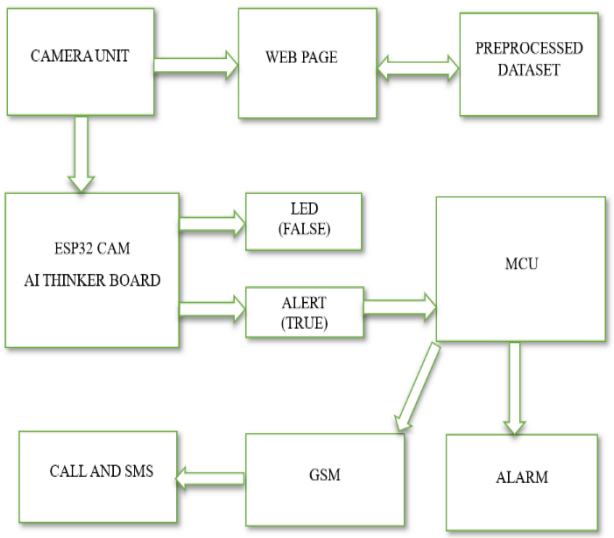

Figure 1 System Architecture Block Diagram

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2196-220
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0294

3.1. Hardware Design

There are four main parts of the hardware:

- ESP32-CAM (AI-Thinker): 240 MHz dual-core, OV2640 camera, 4 MB PSRAM, and built-in WiFi. main engine for AI inference.
- NodeMCU ESP8266: An additional MCU for GSM command sequencing, sensor integration, and alert orchestration. SIM800L GSM Module: Quad-band GSM/GPRS via hardware UART for SMS and voice calls at 9,600 band.
- Analog Sensor Array: GPIO D2, D5, D6, and D7 vibration/impact sensors that offer physical crash validation separate from the AI pipeline.

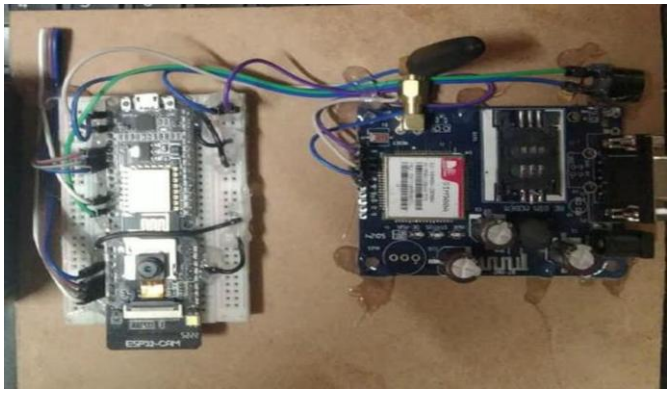

Figure. 2. Hardware Setup

Table 1 Hardware Component Specifications

Component	Specification	Function
ESP32-CAM	240 MHz, 4 MB PSRAM	AI Inference + Camera
NodeMCU ESP8266	80 MHz, 80 KB RAM	Sensor Fusion + Alerts
SIM800L GSM	Quad-band GSM/ GPRS	SMS + Voice Calls
Vibration Sensor	Analog 0–1023	Physical Crash Validation
Total System	5 V DC, ~500 mA pk	Complete Detection Unit

3.2.Software Design

Three layers make up the software stack:

- the inference layer, which uses Fast-CNN on ESP32-CAM.
- the coordination layer, which uses NodeMCU firmware for sensor fusion and decision logic.
- the communication layer, which uses GSM AT command interface and IoT cloud API.
- On the NodeMCU, all decision logic is executed locally.

4. Methodology

4.1. AI Model Architecture

MobileNetV2 [2], a depthwise separable convolutional architecture intended for resource-constrained inference, is the source of the Fast-CNN. A carefully selected dataset of traffic incidents was used for training using Google Teachable Machine. A cloud HTTP endpoint hosts the trained model, which is exported by Google Teachable Machine in TensorFlow.js format. The approximate size of the quantized model is 3.5 MB. The ESP32-CAM records QVGA frames and sends them over local Wi-Fi to this endpoint using HTTP POST. In return, it receives JSON confidence vectors. By avoiding on-device TFLite execution, this architecture circumvents the limitations of ESP32 PSRAM for MobileNetV2-class models. QVGA (320x240 pixels) is the input.

Table 2 Fast-CNN Architecture

Layer	Output Shape	Filters	Activation
Input	320×240×3	-	-
Conv2D Stem	160×120×32	32, 3×3, s=2	ReLU6
Depthwise Blocks	80→40→20	64→128→256	ReLU6
Global Avg Pool	1×1×256	-	-
Dense (Output)	5	5 classes	Softmax

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2196-220
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0294

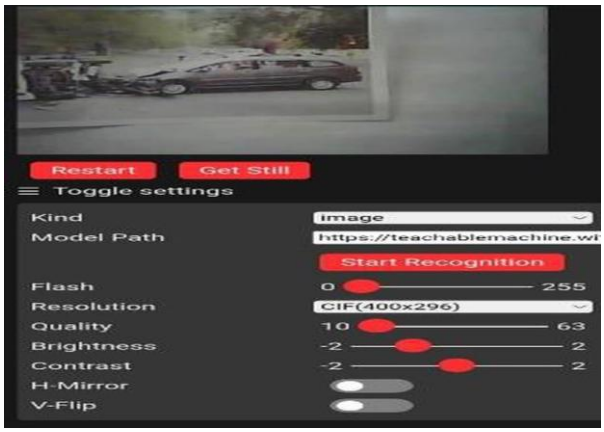

Figure 3. Real-Time Image Recognition Interface with Model Configuration

Loss Function: Categorical cross-entropy:

$$L = -\Sigma\ y_i \log(\hat{y})_i\ ,\ i = \{1, \ldots, C=5\}$$

where y is the ground-truth one-hot label and ŷ is the predicted class probability. Adam optimizer, η=0.001, batch=32.

4.2.Description of Dataset

The collection consists of 1,500 annotated traffic photos in five different categories that were taken from public accident databases and enhanced with original video. Gaussian noise, brightness jitter (±20%), rotation (±15°), and horizontal flipping are examples of augmentation. Divide: 80/10/10 train/val/test, with a ±10% class balance.

Table 3 Dataset Composition and Train/Val/Test

Category	Train	Val	Test	Total
Four-Wheeler Crashing	280	35	35	350
Two-Wheeler Crashing	240	30	30	300
Fire / Post-Accident	200	25	25	250
Rainy Night Crash	200	25	25	250
No Accident (Normal)	280	35	35	350
Total	1,200	150	150	1,500

4.3.Suppression of Two-Tier False Positives

False positives cause needless ambulance dispatches, squander emergency resources, and damage public confidence[6][21]. At one intersection, a system that generates one false alarm every hour results in 24 needless dispatches every day. Prior to sending out any alerts, RapidResQ uses a two-tier validation mechanism[20] Tier 1: Visual Confidence Gate: Only when the maximum CNN confidence score surpasses the threshold θ = 0.85 is a classification accepted: Decision = Accident if $\max(^{n}) \geq \theta = 0.85$ This gate eliminates erroneous detections caused by motion blur, partial occlusions, and illumination transients. The rate of false positives rises from 6.8% to more than 22% when θ is decreased below 0.70. Tier 2: Analog Sensor Confirmation At intervals of 250 ms, NodeMCU takes N=20 consecutive sensor readings. For confirmation, the mean reading must be greater than τ = 500 on an ADC scale of 0–1023:

$$\overline{S} = (1/N)\Sigma S$$

$$\text{Alert} = \text{Decision1}\ (\overline{S} \geq 500)\ (\text{flag} = 0)$$

A 10-second inter-alert suppression window is enforced by a flag variable for singular alert generation.

4.4.Types of False Positives:

- Type I: Visual misclassification: non-accident events with a confidence level of at least 0.85, such as emergency braking or construction vehicles. 18.3% is the frequency in visual-only mode.
- Type II: Sensor noise spikes: Variabilities in the road surface that result in brief readings above τ=500. reduced by averaging the means of N = 20 samples.
- Concurrent false conditions refer to Type III. Seldom do Type I and Type II events occur simultaneously (<1% of trials). suppressed by the 10-second window between alerts.

4.5.Hybrid Alert Distribution System

The system dynamically assesses network connectivity after incident confirmation in order to choose the best channel[18][20]:

- • IoT Cloud Mode (Urban): Structured

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2196-220
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0294

metadata, such as GPS coordinates, camera ID, traffic signal ID, sensor telemetry, timestamp, and incident type, can be transmitted to cloud platforms via Wi-Fi. enables forensic logging and real-time dashboard visualization[17].

- • GSM Fallback Mode (Rural): Automatic GSM-based alerting via SMS and voice calls sent using the AT command sequence-AT+CMGF=1, AT+CMGS, terminated with ASCII 26-is triggered by absent or deteriorated Wi-Fi.

5. IMPLEMENTATION

5.1.Configuration of the Firmware

The OV2640 is configured at QVGA with JPEG quality 10–12 by the ESP32-CAM firmware. Wireless: AP+STA mode simultaneously. Serial numbers are 9,600 baud (GSM) and 115,200 baud (camera). There is only one per-category alert dispatch enforced by flag variables c1–c4. Power: 5 V DC; brownout detection is turned off to allow for GSM transmission burst tolerance.

5.2. Model Installation

The Google Teachable Machine exports the Fast-CNN in TensorFlow.js format, which is stored at a cloud HTTP endpoint. Using the frames it has captured, the ESP32-CAM sends HTTP POST requests and gets JSON confidence vectors in return. This hybrid architecture delegates compute-intensive inference to the Teachable Machine cloud endpoint while retaining all time-critical decision logic — sensor polling, false positive validation, and alert dispatch — on the local NodeMCU microcontroller. The system requires only local Wi-Fi connectivity for inference, with GSM fallback activated automatically when Wi-Fi is unavailable. This design represents a deliberate trade-off: slightly higher inference latency (~850 ms HTTP round-trip) in exchange forsignificantly reduced hardware cost and complexity compared to on-device TFLite deployments on ESP32-S3 or NVIDIA Jetson-class devices

6. RESULTS AND DISCUSSIONS

6.1. Classification Outcomes

The Fast-CNN was tested on 150 held-out test samples in all five categories under a range of weather conditions, traffic densities, and illumination types (daylight, artificial, and low-light).

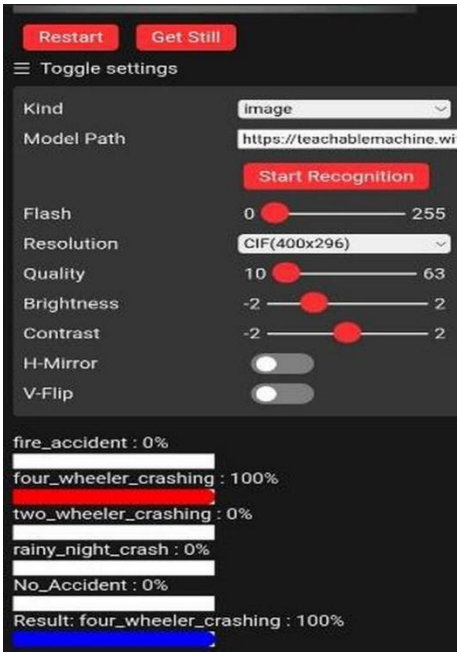

Figure 4 Accident Detection: Classification

Table 4. Per-Class Classification Performance Metrics

Category	Prec. (%)	Recall (%)	F1 (%)
Four-Wheeler Crash	92.3	91.4	91.8
Two-Wheeler Crash	89.7	90.0	89.8
Fire/Post-Accident	91.0	88.0	89.5
Rainy Night Crash	86.0	84.0	85.0
No Accident	94.2	95.7	94.9
Macro Avg	90.6	89.8	90.2

Macro F1: 90.2%; overall accuracy: 90.5%. In Rainy Night Crash (85.0% F1), performance deterioration is most noticeable because of decreased contrast and motion blur. The No Accident category, which is crucial for false positive suppression, attains the highest precision (94.2%).

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2196-220
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0294

6.2. False Positive Analysis

Table 5. False Positive Analysis Across Validation Modes

Validation Mode	FP Rate	TP Rate	Alert Acc.
Visual-Only (Baseline)	18.3 %	91.2 %	78.4%
Sensor-Only (No AI)	21.7 %	82.1 %	72.6%
Tier 1 Only (θ=0.85)	11.4 %	90.5 %	85.3%
Tier 1+2 (RapidResQ)	6.8%	90.5 %	93.1%

The two-tier validation lowers the false alert rate by 63%, from 18.3% to 6.8%. The sensor-only mode performs the worst (21.7% FP), indicating that neither modality is adequate on its own. The true positive rate is kept at 90.5% while the alert accuracy is attained at 93.1%.

6.3.Temporal Performance

Over 50 trials, end-to-end latency was measured. Mean: 4.7 s ($\sigma = 0.6$ s), which is significantly faster than cloud-dependent architectures (10–15 s) and manual surveillance (5–15 min) [6][8].

Table 6 Latency Breakdown By Processing Stage

Processing Stage	Mean (ms)	Std (ms)
Frame Capture + JPEG Encode	120	±15
HTTP Inference Request	850	±120
Sensor Polling (20×250 ms)	2,800	±50
Alert Decision Logic	30	±5
SMS / IoT Transmission	900	±200
Total End-to-End	~4,700	±390

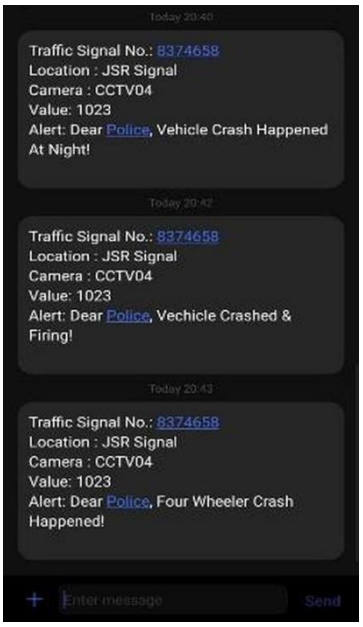

Figure 5 Emergency SMS Alert Interface with Structured Incident Payload

6.4.Comparative Evaluation

Table 7 Comparative Analysis Of Accident Detection Systems

System	Accuracy	Latency	Size	Rural
RapidResQ	90.5%	~5 s	3.5 MB	Yes
Cloud CNN [6]	92–94%	10–15 s	>200 MB	No
YOLOv11 [7]	~91%	~3 s	~25 MB	Partial
Sensor-Only [12]	~78%	~2 s	N/A	Yes
Manual [6]	Variable	5–15 min	N/A	Partial
Pathik et al. [19]	~89%	~8s	>100MB	No
Roy & Gupta [15]	~76%	~3s	N?A	Partial

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 04 April 2026
Page No: 2196-220
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0294

With a model size of only 3.5 MB and a hardware cost of about $13, RapidResQ achieves accuracy and latency competitive with cloud CNN baselines, offering complete support for rural GSM deployment. Cloud-based systems achieve slightly better accuracy at absolute broadband dependency and 2-3 times higher latency.

6.5. Power and Cost Analysis

6.6.

TABLE 8. Power Consumption Analysis

Operational Mode	Current Draw	Battery Life (3 Ah)
Idle / Monitoring	~180 mA @ 5 V	~16 hours
Active Inference	~320 mA @ 5 V	~9 hours
GSM TX Burst (peak)	~500 mA @ 5 V	Transient

At ~$13 per unit, RapidResQ offers a significant cost advantage over commercial systems ($500–$2,000) and GPU-accelerated edge devices (NVIDIA Jetson Nano: ~$100), enabling wide-area deployment in resource-constrained municipalities.

Conclusion and Future Work

RapidResQ, a unified framework for vehicle accident detection and emergency notification that combines edge-based visual AI with hierarchical communication channels, was introduced in this paper. With an end-to-end alert latency of less than five seconds, the deployed Fast-CNN achieves 90.5% overall accuracy and 90.2% macro F1-score across five incident categories.

Visual confidence gating (θ=0.85) with analog sensor mean verification (τ=500) is the two-tier false positive suppression that lowers the false alert rate from 18.3% to 6.8%, a 63% improvement that directly results in fewer needless emergency dispatches in actual deployment.

RapidResQ provides a cost-effective, infrastructure-aware basis for scalable traffic safety technology across diverse transportation environments at about $13 per unit, with complete GSM rural fallback support and minimal infrastructure requirements.

6.7.Integration of Thermal and Infrared Imaging

The system will incorporate thermal imaging or infrared illumination to address low-light deterioration and enhance rainy night crash detection performance.

6.8.Adaptive Thresholding and Temporal Averaging

Multi-frame temporal averaging and adaptive confidence threshold tuning will be used to improve detection robustness and lower the residual false positive rate.

6.9.Dataset Expansion

In order to increase diversity in terms of locations, vehicle types, weather conditions, and lighting scenarios, the dataset will be expanded from 1,500 images to over 10,000 images, which will enhance model generalization.

6.10. Migration to On-Device TFLite Inference

The transition from cloud-based HTTP inference to on-device TensorFlow Lite execution on the ESP32-S3 will be the main focus of future work. This will eliminate the need for Wi-Fi, lower latency variability, and allow for fully offline real-time operation with the device's hardware-accelerated inference capabilities and 8 MB PSRAM.

Reference

[1]. J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," Proc. IEEE CVPR, pp. 7263-7271, 2017.
[2]. A. Howard et al., "MobileNetV2: Inverted Residuals and
[3]. Linear Bottlenecks," Proc. IEEE CVPR, pp. 4510-4520, 2018.
[4]. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Proc. IEEE CVPR, pp. 770-778, 2016.
[5]. W. Liu et al., "SSD: Single Shot MultiBox Detector," Proc. ECCV,
[6]. pp. 21-37, 2016.
[7]. J. Redmon, S. Divvala, R. Girshick, and A.

Farhadi, "You Only Look Once," Proc. IEEE CVPR, pp. 779-788, 2016.

[8]. C. Tan, Z. Wang, Y. Li, and J. Chen, "A Survey on Deep Learning-Based Real-Time Traffic Accident Detection," IEEE Access, vol. 10, pp. 78456-78473, 2022.

[9]. M. S. Arefin, M. R. Islam, and K. Ahmed, "Real-Time Traffic Accident Detection Using YOLOv11," Heliyon, vol. 11, no. 1,e40123, 2025.

[10]. V. A. Adewopo, T. Chen, and R. Kumar, "Ensemble Deep Learning for Traffic Incident Detection," IEEE Trans. ITS, vol. 25, no. 4, pp. 4567-4579, 2024.

[11]. R. Sharma and P. Singh, "Enhanced Road Safety Through Deep Learning-Based Accident Detection," Int. J. Eng. Res. Technol., vol. 13, no. 8, pp. 2134-2142, 2024.

[12]. [K. Patel and M. Desai, "Comprehensive Review of CNN and YOLO Accident Detection," Int. J. Innovative Res. Technol., vol. 10, no. 3, pp. 567-575, 2023.

[13]. S. B. Sharmila, R. Kumar, and A. Verma, "Edge Computing for Accident Detection," Int. J. Scientific Res. Eng., vol. 9, no. 1, pp. 45-53, 2025.

[14]. K. P. Reddy, S. Rao, and M. Krishna, "IoT-Based Vehicle Accident Detection with ESP32," Int. J. Innovative Res. Physical Sci., vol. 10, no. 4, pp. 234-241, 2022.

[15]. A Review on YOLO and CNN Powered Real Time Accident Detection Systems, Int. Journal of Innovative Research in Technology (IJIRT), 2023.

[16]. Smart Traffic Accident Detection and Automated Response System using YOLO, Int. Journal of Research Publication and Reviews (IJRPR), 2024.

[17]. S. Roy and P. Gupta, “IoT-Based Accident Detection and Alert System,” International Journal of Emerging Technology, 2020.

[18]. K. P. et al., “Automatic vehicle accident detection and healthcare unit notification using IoT technology with ESP32,” Int. Journal of Innovative Research in Multidisciplinary Physical Sciences, vol. 10, no. 4, 2022.

[19]. .P. Devi et al., “IoT based accident and analysis system using edge computing,” Int. Journal of Innovative Research in Engineering, vol. 5, no. 2, 2024.

[20]. J. Tao, R. Ali, S. Ahmad and F. Ali, “IoT-Based Smart Accident Detection and Early Warning System for Emergency Response and Risk Management,” Int. J. of Advanced Computer Science and Applications (IJACSA), vol. 16, no. 3, 2025.

[21]. N. Pathik, R. K. Gupta, Y. Sahu, A. Sharma, M. Masud and M. Baz, “AI Enabled Accident Detection and Alert System Using IoT and Deep Learning for Smart Cities,” Sustainability, vol. 14, no. 13, 7701, 2022.E. Nasr, E. Kfoury and D. Khoury, “An IoT Approach to Vehicle Accident Detection, Reporting, and Navigation,” in Proc. 2016 IEEE Int. Multidisciplinary Conf. on Engineering Technology (IMCET), Beirut, 2016.

[22]. Kumar and A. Acharya, “An IoT-Based Vehicle Accident Detection and Classification System Using Sensor Fusion,” 2020 (smartphone and external sensors).