JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

AN IoT-ENABLED AND DEEP-LEARNING-ASSISTED FRAMEWORK FOR ROAD ACCIDENT DETECTION, VALIDATION AND EMERGENCY REPORTING

Madduri.Veera Venkata Satya Sri Tulasi[1], Nagulapalli. Divya Sri Durga Parvathi[2], Ramani. Pavani[3], M. krishnakartik[4*]

[1,2,3] UG Students, Department of CSE, Srinivasa Institute of Engineering & Technology (A), Cheyyeru, Amalapuram-533222, AP, India

[4] Associate Professor, Department of Chemistry, Srinivasa Institute of Engineering & Technology (A), Cheyyeru, Amalapuram-533222, AP, India

Abstract

Delayed notification and inaccurate identification of road traffic crashes can hinder the timely mobilisation of emergency assistance. This paper presents an integrated Internet of Things (IoT) and artificial intelligence (AI) framework for detecting possible vehicle accidents, validating visual evidence and communicating incident information to nearby emergency services. The framework combines a force-sensing unit, vehicle-speed information from a Global Positioning System (GPS) receiver, an ATmega328P microcontroller, an ESP8266-based communication unit, a Raspberry Pi single-board computer, a camera and a GSM communication module. A threshold-based sensor event initiates an audible cancellation interval and activates image/video acquisition. Captured frames are analysed using transfer-learning-based convolutional neural networks, with ResNet-50 and Inception-ResNet-v2 considered as candidate classifiers. Only an event that passes the visual validation stage is forwarded to the reporting module, which obtains geographic coordinates and identifies nearby hospitals and police stations for notification. The design is intended to reduce false alerts arising from routine braking, road irregularities and other non-collision events while shortening the time required to communicate an incident. The source study describes a prototype assembled on a toy vehicle and software-based accident/non-accident classification. However, the available documentation does not provide sufficient dataset details or independently reproducible performance measurements to establish real-road effectiveness. Accordingly, this paper presents the system architecture and prototype methodology without claiming new accuracy or field-performance results. A validation protocol and deployment requirements are outlined for future full-scale evaluation.

Keywords: Internet of Things; road accident detection; deep learning; ResNet-50; Inception-ResNet-v2; GPS; emergency reporting; intelligent transportation systems.

1. Introduction

Road traffic injuries remain a major public-health and transportation-safety concern. The World Health Organization (WHO) reported approximately 1.19 million road traffic deaths annually in its 2023 global status report, with road traffic injuries continuing to be a leading cause of death among people aged 5–29 years [6]. The consequences of a crash depend not only on the severity of the collision but also on the time taken to identify the event, determine its location and mobilise appropriate assistance. In locations with limited traffic surveillance or poor network coverage, an injured occupant may be unable to contact emergency services promptly.

Conventional vehicle-based warning approaches often rely on a single signal, such as abrupt speed variation or impact intensity. Such signals can also occur during emergency braking, passage over a pothole or speed breaker, and certain

mechanical disturbances. A single-threshold detector may therefore generate false alarms. Integrating multiple sensing modalities with image-based verification offers a way to improve the reliability of automatic incident reporting. IoT systems provide a means of connecting sensors, embedded controllers and communication services, while deep-learning models can learn visual patterns from images and video [2,5].

The present work develops a two-stage accident detection and reporting framework. In the first stage, embedded sensors identify a potential crash and activate a camera. A cloud-based visual classifier then determines whether the captured scene is consistent with an accident. In the second stage, the validated event is associated with GPS coordinates and communicated to emergency contacts and nearby response organisations. The system architecture follows the concepts reported in the supplied 2023 paper [1] and the related IoT/deep-learning study by Pathik et al. [2], while this manuscript reorganises the technical description and identifies the evidence required for a reproducible evaluation.

1.1 Objectives

- To describe an integrated sensor- and camera-assisted method for identifying possible road accidents.
- To use visual deep-learning validation to reduce false notifications caused by non-accident events.
- To obtain the accident location through GPS and support the identification of nearby emergency services.
- To define a transparent evaluation protocol for measuring detection performance, communication delay and system reliability.

2. Related Work

The Internet of Things connects physical devices through sensing, communication and software services. Atzori et al. describe IoT as an integration of identification and tracking technologies, sensor/actuator networks, communication protocols and distributed intelligence [5]. These capabilities are relevant to vehicle safety systems because measurements collected inside a vehicle can be transmitted to remote computing services for analysis and response.

Pathik et al. proposed an AI-enabled accident detection and alert system in which an IoT kit collects accident-related information, including position and impact-related measurements, and forwards the data for deep-learning validation [2]. Their work combines sensor triggering with visual classification and emergency-service notification. The supplied paper by More et al. describes a closely related arrangement using a force sensor, ATmega328P, ESP8266, Raspberry Pi, Pi camera, GPS and GSM module [1]. It also describes a toy-car prototype and software simulation of accident and non-accident classes.

Residual networks use shortcut connections to facilitate the optimisation of deep neural networks [3]. Inception-ResNet combines Inception-style multi-branch processing with residual connections; the original study reported that residual connections can accelerate training of Inception networks [4]. These architectures provide suitable transfer-learning backbones for visual classification, but their performance in road-accident recognition depends on the relevance, size, balance and diversity of the training dataset. General ImageNet performance cannot be treated as evidence of accident-detection accuracy.

Recent work on IoT-based vehicle accident detection has also explored visual situation reporting and sensor-based vehicle-state monitoring [7]. Collectively, the literature supports a multimodal architecture, but practical deployment still requires validation under different vehicle types, camera positions, lighting conditions, road environments and network conditions.

3. Proposed System Architecture

The proposed architecture consists of two connected functional stages: (i) accident sensing and AI-based verification, and (ii) location-based emergency reporting. The vehicle-mounted unit monitors impact-related and motion-related signals. When the configured trigger condition is met, the controller activates an alarm and starts the camera acquisition process. A short video sequence and still images are transmitted to a processing service. The classifier evaluates the visual evidence and returns an accident/non-accident decision. If the event is validated, the reporting module obtains the vehicle's coordinates and sends an alert to the relevant emergency contacts and response services.

3.1 Hardware Components

Component	Function in the framework
Force sensor	Measures an impact-related mechanical response and provides a signal to the microcontroller. Its installation position, range and calibration must be documented.
ATmega328P	Reads the sensor signal, compares it with the configured threshold and initiates the local alarm sequence.
NodeMCU ESP8266	Provides Wi-Fi connectivity and transfers event information between the embedded unit and the cloud/service layer.
Raspberry Pi 3B+	Coordinates camera capture and supports communication with the image/video processing service.
Pi camera and servo motor	Captures images/video of the surrounding scene; the servo arrangement can be used to orient the camera where implemented.
GPS receiver	Provides latitude and longitude and, where supported, vehicle-speed information.
GSM SIM900A	Provides cellular communication for sending incident notifications when configured network service is available.
Power supply	Supplies regulated power to the embedded components; backup power is desirable to maintain operation after vehicle power interruption.

3.2 Functional Sequence

1. The force sensor and vehicle-motion information are monitored continuously.
2. When the configured impact/motion trigger is crossed, the microcontroller activates a local alarm and starts a cancellation interval.
3. If the alert is cancelled by an occupant within the configured interval, the event is treated as an unconfirmed incident and the system may provide a non-emergency assistance message, subject to user configuration.
4. If the alert is not cancelled, the camera is activated and records a short video sequence and still images.
5. The captured frames are uploaded to the processing module, where a trained classifier evaluates whether the scene represents an accident.
6. If the visual result indicates an accident, GPS coordinates and vehicle/user identifiers are retrieved and the emergency-reporting module is activated.
7. The reporting module identifies nearby hospitals and police stations and transmits an alert containing the event type, location, timestamp and available supporting media.

4. Accident Detection and Validation Methodology

4.1 Sensor-Based Triggering

Let F(t) represent the calibrated force-sensor output at time t and S(t) represent the vehicle speed estimated from GPS. A candidate event is generated when the force signal exceeds a calibrated impact threshold, or when an abrupt speed change is detected within a defined time window. A general trigger expression is:

$$E(t) = [F(t) > T_F] \text{ OR } [|S(t) - S(t-\Delta t)| > T_S]$$

where T_F and T_S are experimentally calibrated thresholds and Δt is the observation interval. The supplied paper mentions a force threshold of 350 Pa and a 30-second alarm interval [1]. The 350 Pa value should be treated as a reported configuration, not a universal collision threshold: the sensor type, mounting arrangement, calibration procedure and relationship between its output and impact severity must be established before deployment. The source algorithm also contains both AND and OR trigger expressions; therefore, the exact Boolean logic must be fixed and documented in an implementation before reproducible testing.

The alarm cancellation interval is intended to give occupants an opportunity to suppress a false alert. The supplied design specifies 30 seconds [1]. This interval is a design parameter and should be assessed for usability and safety; it should not prevent an occupant from making an immediate manual emergency call.

4.2 Video Acquisition and Pre-processing

After a candidate event remains un-cancelled, the Raspberry Pi activates the camera. The supplied system describes acquisition of approximately 30 seconds of video together with still images [1]. The video is decomposed into frames, resized to the input dimensions required by the selected neural network, and normalised using the preprocessing procedure associated with the model. Frame sampling should be specified consistently so that redundant adjacent frames do not dominate the prediction. Data augmentation may be used during training to improve robustness to illumination, scale and viewpoint variation, provided that transformations do not change the semantic accident label.

4.3 Deep-Learning Classification

Two transfer-learning backbones are considered: ResNet-50 and Inception-ResNet-v2. ResNet introduces identity shortcut connections that support the training of deeper networks [3]. Inception-ResNet combines multi-branch Inception modules with residual connections [4]. For the target application, the classification head is adapted to two classes: accident and non-accident. The network may be fine-tuned using labelled frames or video clips from a road-scene dataset. The training corpus should include genuine collision scenes and hard negative examples such as speed breakers, potholes, sudden braking, vehicle stops, camera vibration and routine traffic congestion.

For a binary classifier, the predicted class can be selected from the model probability $p(y|x)$, where x is an input frame or sampled frame sequence. A confidence threshold τ may be used: an accident decision is returned when the estimated accident probability is greater than or equal to τ. For video-level decisions, frame predictions should be aggregated using a documented rule, such as a minimum proportion of positive frames or a temporal voting strategy. The threshold must be selected using validation data and then held fixed during final testing.

The source paper refers to ResNet-50 and Inception-ResNet-v2 and reports software classification of accident and non-accident inputs [1]. It does not provide enough information in the supplied copy to reproduce the dataset composition, frame-sampling strategy, training/validation split, hyperparameters or a complete confusion matrix. Consequently, no new numerical performance claim is made in this manuscript.

4.4 GPS-Based Location Estimation

Once an event is validated, the GPS receiver supplies latitude and longitude. The distance between the vehicle and a registered emergency facility can be estimated using the Haversine equation [7]:

$$a = \sin^2(\Delta\varphi/2) + \cos(\varphi_1)\cos(\varphi_2)\sin^2(\Delta\lambda/2)$$

$$d = 2R \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{(1-a)})$$

Here, φ_1 and φ_2 are the latitudes, λ_1 and λ_2 are the longitudes in radians, $\Delta\varphi$ and $\Delta\lambda$ are their differences, R is the mean Earth radius and d is the great-circle distance. The nearest facility by straight-line distance is not necessarily the fastest to reach; a production system should preferably use a verified road-network routing service and current facility availability.

5. Emergency Reporting Module

The reporting module maintains structured records for registered users, vehicles, hospitals and police stations. A user record may contain the vehicle owner's name, registered mobile number and emergency contacts. A vehicle record may contain registration number, vehicle category and associated user identifier. Hospital and police-station records should include verified names, geographic coordinates, telephone numbers and service coverage. Access to personal information must be restricted to authorised operations.

A validated alert should contain, at minimum, an incident identifier, timestamp, GPS coordinates, vehicle identifier, event classification, confidence score, callback number and a secure link to supporting media where available. The alert should be sent through the configured communication channel to the relevant emergency contacts and designated response services. Delivery acknowledgement, retry handling and an auditable event log are important because a transmitted message is not necessarily a received or acted-upon message.

6. Algorithm

1. Initialise sensor interface, GPS, camera, network connection and alarm controller.
2. Read force-sensor output F and GPS-derived speed S.
3. Evaluate the calibrated candidate-event condition E.
4. If E is false, continue monitoring.

5. If E is true, activate the audible alarm and start the cancellation timer.
6. If the occupant cancels the alert within the configured interval, log the cancelled event and return to monitoring.
7. Otherwise, capture the configured video segment and still images.
8. Pre-process sampled frames and submit them to the trained accident classifier.
9. Aggregate frame-level predictions to obtain the video-level decision.
10. If the decision is non-accident, log the event and return to monitoring.
11. If the decision is accident, retrieve GPS coordinates and registered vehicle/user details.
12. Identify relevant nearby emergency facilities and transmit the incident alert.
13. Record transmission status and continue retries/escalation according to the communication policy.

7. Prototype and Experimental Methodology

The supplied study describes an IoT kit installed on a toy automobile because the proposed hardware arrangement was not evaluated on a full-size road vehicle [1]. The described prototype includes sensors, an alarm, controller hardware, GPS and camera components. The software simulation uses Python and OpenCV to load image inputs and classify them into accident and non-accident categories [1]. These details support a proof-of-concept description, but they do not establish operational performance in real traffic.

7.1 Recommended Reproducible Evaluation

A complete experimental report should document the dataset source, number of unique videos, number of frames per class, class distribution, annotation procedure, exclusion criteria, train/validation/test split and whether frames from the same video or recording session are kept within a single split. Splitting adjacent frames randomly across training and testing can cause data leakage and inflate apparent performance. A video- or incident-level split is therefore recommended.

The following metrics should be calculated on an independent test set: accuracy, precision, recall (sensitivity), specificity, F1-score, false-positive rate and confusion matrix. System-level measurements should include sensor-trigger latency, visual verification latency, GPS fix availability, notification delivery time, end-to-end alert latency, packet delivery success and power consumption. Results should be reported separately for the sensor trigger, image classifier and complete integrated system.

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Precision = TP / (TP + FP); Recall = TP / (TP + FN)

Specificity = TN / (TN + FP); F1 = 2 × Precision × Recall / (Precision + Recall)

TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives, respectively. These metrics are essential because a high overall accuracy can conceal an unacceptable number of missed accidents or false emergency alerts.

8. Results and Discussion

The evidence available in the supplied paper supports a limited prototype-level observation: an IoT kit was assembled on a toy vehicle, and the software workflow produced accident and non-accident classification outputs for supplied image/video inputs [1]. The paper includes IDE screenshots illustrating the two prediction classes. However, the available report does not provide the underlying test-set size, class-wise sample counts, confusion matrix, confidence intervals or independently repeatable timing measurements. The screenshots therefore demonstrate that the software pipeline was exercised, but they are not sufficient to quantify generalisation performance.

The main architectural benefit is the use of two levels of decision-making. Sensor measurements provide an inexpensive trigger, while image-based verification offers an additional source of evidence before emergency reporting. This can potentially reduce false notifications compared with a single-sensor threshold. Nevertheless, visual verification can also fail when the camera is obstructed, poorly oriented, damaged, exposed to low light or unable to capture the collision. A system should therefore retain a manual emergency option and consider a fail-safe escalation policy when sensor evidence is severe but visual evidence is unavailable.

The use of cloud processing can support computationally intensive neural networks, but it creates a dependency on connectivity and server availability. A practical implementation should buffer event data locally, encrypt communications, authenticate devices, protect stored video and minimise the personal data transmitted. These safeguards are especially important because accident media may contain identifiable people, vehicle registration numbers and precise location information.

9. Limitations and Deployment Considerations

- The prototype described in the source is a toy-vehicle implementation; full-scale vehicle testing is not established by the available evidence.
- The force-sensor threshold and speed-change threshold require calibration for vehicle mass, sensor placement, mounting stiffness and road conditions.
- The source contains inconsistent AND/OR descriptions of the sensor trigger; the implemented decision logic must be specified unambiguously.
- The dataset composition and full model-training configuration are insufficiently documented for independent replication.
- A camera may not capture all collision directions or may be obscured, damaged or affected by lighting and weather.
- Cloud-based classification and GSM/Wi-Fi reporting depend on network coverage, service availability and power continuity.
- Security, privacy, consent, data retention and access-control mechanisms require explicit design and testing.
- Emergency-facility databases must be accurate and regularly updated; straight-line proximity alone does not establish response availability.

10. Future Scope

Future work should evaluate the system on full-size vehicles and controlled crash/near-crash scenarios, followed by carefully governed field trials. An expanded dataset should cover multiple vehicle categories, collision directions, weather and illumination conditions, road types and non-accident hard negatives. The performance of ResNet-50 and Inception-ResNet-v2 should be compared under identical data splits and training conditions. Further improvements may include temporal video models, inertial measurement units, edge inference, redundant communication channels, secure event authentication and automated acknowledgement from emergency services. A formal safety analysis should examine missed detections, false alerts, sensor failures and network outages before any public-road deployment.

11. Conclusion

This paper has presented an IoT-enabled and deep-learning-assisted framework for road accident detection, visual verification and emergency reporting. The architecture combines impact-related sensing, vehicle-speed information, embedded control, camera-based classification, GPS positioning and cellular communication. Its two-stage design is intended to distinguish potential collision events from non-accident disturbances before initiating emergency notifications. The supplied study demonstrates a toy-car prototype and software classification workflow, but the evidence available does not justify claims of validated real-road accuracy or a quantified reduction in fatalities. Reproducible dataset documentation, calibrated sensors, independent testing, full-scale vehicle trials, reliable communications and robust security controls are necessary next steps. The framework should therefore be regarded as a proof-of-concept architecture pending comprehensive validation.

References

[1] A. More, M. Mahajan, and H. Gholap, “AI and IoT Based Road Accident Detection and Reporting System,” International Journal for Research in Applied Science & Engineering Technology, vol. 11, no. III, pp. 319–329, Mar. 2023, doi: 10.22214/ijraset.2023.48904.

[2] 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, Art. no. 7701, 2022, doi: 10.3390/su14137701.

[3] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31, no. 1, 2017, doi: 10.1609/aaai.v31i1.11231.

[5] L. Atzori, A. Iera, and G. Morabito, “The Internet of Things: A survey,” Computer Networks, vol. 54, no. 15, pp. 2787–2805, 2010, doi: 10.1016/j.comnet.2010.05.010.

[6] World Health Organization, Global Status Report on Road Safety 2023. Geneva: WHO, 2023. ISBN: 978-92-4-008651-7.

[7] M. Aslam et al., “An IoT-Based Automatic Vehicle Accident Detection and Visual Situation Reporting System,” Journal of Advanced Transportation, 2024, doi: 10.1155/2024/4719669.