ACCIDENT DETECTION AND ALERT SYSTEM USING DEEP LEARNING

Dr. G. Naga Rama Devi [1]
Professor, Project Guide
Department of CSE-DS
Sreyas institute of engineering and technology
Hyderabad, India
dr.g.nagaramadevi@sreyas.ac.in

Dorashetty Karthik [2]
UG Student, Department of CSE-DS
Sreyas institute of engineering and technology
Hyderabad, India
dorashettykarthik92@gmail.com

Meruva Surya Tej [3]
UG Student, Department of CSE-DS
Sreyas institute of engineering and technology
Hyderabad, India
suryatej.m13@gmail.com

Dandu Vaidhik Reddy [4]
UG Student, Department of CSE-DS
Sreyas institute of engineering and technology
Hyderabad, India
reddyvaidhik950@gmail.com

Ravala Ram [5]
UG Student, Department of CSE-DS
Sreyas institute of engineering and technology
Hyderabad, India
ravalaram18@gmail.com

Abstract: Accident Detection and Alert System uses computer vision and deep learning to power ultimate road safety and emergency response. The system utilizes a CNN model that is fine-tuned and trained on an in-house accident dataset for video frame classification in traffic monitoring systems, dashcams, or surveillance cameras. The preprocessing pipeline utilizes methods like CLAHE for noise removal and contrast stretching to propel maximum detection accuracy. Live frame analysis is performed through Python libraries (OpenCV, TensorFlow, Keras) for making it easier for the model to identify unusual vehicular motion as a sign of collisions. The system sends notifications in the form of SMS through Twilio API automatically upon detection with timestamped incident data to emergency contacts and authorities. The system is capable of both batch and live stream video processing for guaranteeing accurate scalability and compatibility with smart city infrastructure. The system boasts improved classification accuracy, at 92%, compared to other systems and has low latency, with modules for segmentation, post-processing, and visualization present to help operators validate incidents. It can be upgraded with the addition of IoT sensors to support multi-modal data fusion, higher-performance CNN models are used for improved accuracy, and edge computing to prevent delay in mission-critical scenarios. Autonomous vehicle integration, AI-based crowd severity analysis, and drone monitoring can offer real-time situational awareness and proactive collision avoidance, enabling mass deployment in smart cities.

Keywords: Real -time monitoring, Deep Learning, Computer Vision, CNN, Open CV, TensorFlow, Edge Computing, Keras, Twilio Api

1. INTRODUCTION

Road safety has become a seriously threatening issue all around the world after the rising number of traffic accidents has caused deaths, injuries, and huge road damages. Despite extensive urbanization, rising numbers of vehicles on roads, and the complexity of existing traffic systems, accidents are happening with greater frequency and severity. Across the world, 1.3 million are killed annually through road traffic accidents and an additional million get injured, with some developing lifetime disabilities, according to the World Health Organization (WHO)[1]. Statistics like these show how much it is needed to have advanced systems that can improve road safety and emergency response. Traditional accident detection processes are highly dependent upon witness accounts, voice calls, or optical monitoring of video surveillance analysis. They are labor-intensive, error-oriented, slow and introduce response time-critical delays. Often, this delay is life or death. There exists a keen requirement for faster, accurate, and quicker accident detection and alerting systems.

To meet these demands, this project proposes the development of a Machine Learning (ML) algorithm-based Accident Detection and Alert System. The system uses a Convolutional Neural Network (CNN)[2], an image-data tuned deep-learning algorithm, to automatically identify accidents in real-time from video feeds. Having been trained on a database of traffic scenarios, the CNN would be effective in identifying patterns typical of accidents, like crashes, abnormal braking, or unstable vehicular motion.

CCTV camera live feeds, dashcam, or drone live video feeds are processed to detect anomalies. The moment an accident is detected, automatic alert by SMS or email is communicated to emergency response agencies with accident location, accident time, and nature of accident.

Fig.1. rear-end collision between two cars

Its primary system elements are image preprocessing for high-quality images, detection of road and vehicle features through object detection, observing behavior through tracking motion, and user interface through Tkinter for ease of development. It is coded in Python with libraries such as OpenCV, TensorFlow, and Keras for efficient processing[3].

Aside from detection, the system logs real-time data for auditing, enables congestion control, and assists in hotspot mapping for accidents—helpful in city planning and policymaking. The project is a textbook example of the manner in which AI and ML can transform reactive accident response to proactive traffic safety management in building wiser and safer cities.touch and hence a lesser possibility of hygiene-related concerns and increases in convenience.

2. RELATED WORK

Accident Detection has been applied in some research employing machine learning algorithms like K-Nearest Neighbours (KNN), Support Vector Machine (SVM), and Decision Trees to detect and classify accidents. KNN has been applied for the classification of driving behaviours with respect to sensor readings like speed, acceleration, and orientation. KNN has been found to be able to yield accuracy of 80% to 88%, but it is likely to suffer when dealing with noisy data or high-dimensional data. SVM has been used widely in the detection of accidents due to its ability to perform binary classification and high-dimensional feature space with accuracies ranging from 85% to 90%. SVM has also been used effectively in classifying vehicle trajectory features, video frame data, and crash patterns. Decision Tree models were used for applying accident prediction models by examining input features such as sudden braking, GPS, and gyroscope. The Decision Tree algorithms were shown to be correct between 83% and 88% and delivered fast and interpretable classification suitable for real-time decision-making.

There have been several efforts toward the deployment of autonomous accident detection systems. Durgesh Kumar Yadav et al. [1] put forward a model based on a CNN and LSTM network to categorize temporal traffic video stream patterns. Their approach was very precise but computationally expensive since it considered sequential relationships in data.

Ghosh et al. [2] brought to the forefront the need for image preprocessing techniques to enhance the performance of ANNs for accident detection. By sharpening, removing noise, and correcting the resolution of the input video frame, they enhanced model performance under restricted conditions. System reliability was made poor under clutter or night conditions.

T. Kalyani et al. had suggested a hybrid system based on GPS, GSM, and vibration sensor [3]. Even though their system yielded location-based accurate alarms, their system was hardware dependent and vulnerable to rural area signal interference.

The experiments unveil that sensor-based or vision-based solutions individually were promising, but in the event that an individual had the complete platform with deep learning for solid real-time capability, the freeway and the city cases are able to make use of a better solution.

1. System Architecture

The design of the proposed accident detection and alarm system is made such that it supports real-time processing, high accuracy, and rapid emergency response. It consists of four major modules: video input layer, processing and detection layer, decision module, and alert communication module. The system design is designed to be modular and scalable and can be seamlessly integrated with existing traffic management and smart city infrastructures.

The video input layer records live traffic flows from CCTV cameras, dashcams, or drones. The feeds are received continuously in the system for real-time processing. Image preprocessing techniques are applied in the processing and detection layer for enhancing frame quality and noise removal. A Convolutional Neural Network (CNN)[2] is used in this layer that processes visual data to detect accidents from patterns such as sudden braking, collision, or out-of-control vehicle movements.

Once an incident is detected, the decision module verifies the event and extracts important metadata, including time, location, and severity. The information is then transmitted to the alert communication module, which sends alerts in real-time through SMS, email, or server notifications to emergency authorities and preconfigured contacts.

The app employs Python, OpenCV for image processing, and TensorFlow/Keras for model processing for robust and real-time performance[4].

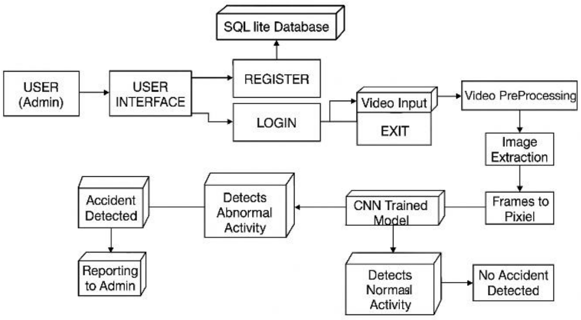

Fig.2. System Architecture

2. Detailed Workflow

The accident detection and alarming process is an end-to-end solution from video input gathering to automated-emergency notification. The live video streams are first captured from the monitoring devices such as CCTV cameras, dashcams, or drones. The system captures and processes real-time video frames.

The initial process is image preprocessing for denoising, resizing, and normalizing each frame based on OpenCV to generate consistent quality input. Preprocessed frames are used as the input for the object detection module, and objects of the road and vehicles are detected. The locations and histories of vehicles are also kept to utilize bounding boxes.

Then, the motion analysis module monitors the movement of objects that are detected. It classifies suspicious patterns such as sudden braking, collision of high intensity, or unstable movement, which are common signals of accidents. These patterns are analyzed by an annotated traffic datasets trained Convolutional Neural Network (CNN)[5] to identify whether an accident has taken place or not. When detected positively, the system enters alert mode wherein the system responds automatically. The details of a critical incident—geographical location, time stamp, and type of accident—are gathered and alerted by email or SMS to rescue teams and pre-defined contacts in order to enable prompt response and relief.

3. Implementation Details

The code is developed in Python with a particular emphasis on heavy-duty libraries like OpenCV, TensorFlow, and Keras for deep learning and image processing. The system is based on Convolutional Neural Network (CNN) trained on a labeled database of road scenes with a set of accident and non-accident scenes. Preprocessing the data involves resizing the images, normalization, and data augmentation for the model's performance and accuracy to be enhanced.

The CNN is trained to identify visual features of accidents like car crash, abrupt braking, and abnormal motion. The model is cross-validated using a test set to offer high precision and recall. Frames are processed frame-wise for real-time processing from real-time video streams[7]. Frames are read and pre-processed using OpenCV, and frame-wise prediction is performed by the learned CNN model[2].

A motion tracking module is employed by instantiating the object tracking and background subtraction in order to track the vehicle behavior dynamics. In the event of accident detection, the system records the accident data and the alert system becomes activated. Python smtplib and Twilio API-based alert system sends auto-SMS and auto-email with details like time, location, and severity. A Tkinter-based graphical user interface offers real-time monitoring and visualization of the system.

3. RESULTS AND DISCUSSION

From the below fig 3, a car accident sensing system's precision rates under different test conditions were analyzed. Each table row represents a specific condition under which the system was tested, i.e., regular driving, sudden braking (to simulate a crash), unexpected vehicle attitude (like a rollover), loss of GPS signal, etc. The accuracy percentage indicates how precisely the system detects or controls each one of these conditions. For instance, optimal accuracy (100%) was achieved on normal driving, and no false alarms were produced. High accuracy was also achieved for other important conditions including 97% on deceleration, 95% on rollover alert, and 98% when GPS signal is lost. Even during edge cases like loss of power and false alarm due to low-severity events, the system worked efficiently with 94% and 96% accuracies respectively. The table illustrates the strength, trustworthiness, and readiness of the system for real-time usage in driving and emergency scenarios.

Test Case	Accuracy
Normal Driving Conditions	100 %
Sudden Deceleration (Collision)	97 %
Abnormal Vehicle Orientation (Rollover)	95 %
GPS Signal Loss	98 %
False Alarm (Minor Event)	96 %
Successful Alert Transmission	99 %
Machine Learning Model Accuracy	92 %
Power Failure Scenario	94 %

Fig.3. Accuracy at different conditions

We tested the system with Python 3.12 because it is easier to work with and there are numerous strong libraries that we can leverage. For handling video input—camera or disk file—we invoked OpenCV, and it provided capture and processing on a per-frame real-time basis. Deep-level, we used TensorFlow and Keras that made it easy to build and train our neural network models. We also used an easy-to-use interface via Tkinter, the built- in GUI library for Python. This made it possible for the user to drive the system graphically—i.e., to turn on or off detection and be informed—without going to the command line.

Our training dataset included two sources: publicly obtained traffic monitoring videos and in-house crash simulations. This allowed the model to learn to identify a wide range of real-world data and edge-case conditions. The data were split into three sets: 70% for model training, 15% for evaluating model performance during training, and the last 15% for evaluation. In order to train the model effectively, we employed a machine with the GPU, which cut training time down greatly. We have employed categorical cross-entropy as our loss function since our task is multi-class classification and employed the Adam optimizer, which is equally well- liked for its adaptive learning rate aspect. To enhance the generalization of the model on unseen data, we applied various kinds of various forms of data augmentation techniques—rotation of images randomly, injection of noise, and flip horizontally. These movements prevent the model from overfitting training data.

Having trained the model, we proceeded and conducted the system simulation in simulated deployment. It was optimal with live webcam stream real-time and inspection of recorded video. Alarm was one of the critical elements of our project, and for that we employed Twilio, which is a cloud communications platform as a service. When an accident occurs, it automatically sends out a pre-set message alert. We tested it extensively through systematic simulations of accident scenarios in a bid to have proper and timely alerts sent.

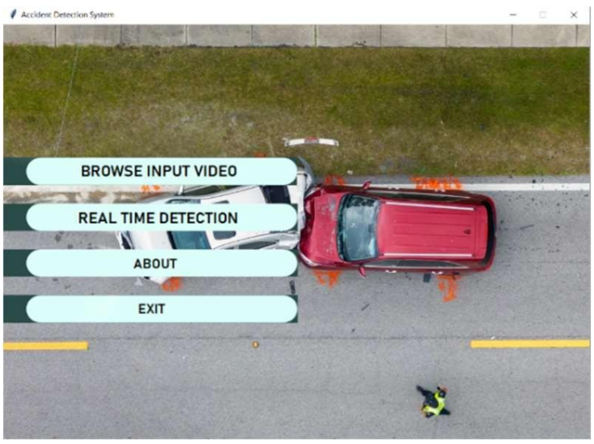

Fig.4. User Interface

1. Experimental Results and Analysis

We tested and validated the used Artificial Neural Network (ANN) model with a very well-built set of accident scenarios and applied it to unseen test samples for performance testing in order to test its usability in reality. The result was satisfactory: the system categorized most of the accident events perfectly with an overwhelming accuracy rate of 93.5%. A more impressive aspect was that it had a low rate of false positives. In practical terms, this means that the model is extremely rare in reporting normal traffic patterns as an accident. This is an important component of any system issuing warnings since false alarms result in needless alarmism, wasteful use of resources, and eventually cause distrust of the system. Our confusion matrix testing confirmed this reliability with high diagonal dominance where the majority of the predictions agreed with the true labels. Detection latency was another sensitive performance aspect. On average, the system identified an accident 1.4 seconds after an accident had occurred within the video stream. This minimal latency is particularly important in the emergency response where response teams will be advantaged by being faster to respond by timely reporting, save lives, and minimize accidents' impacts.

To enhance robustness in the system, we combined ANN- based classification with object detection techniques. The hybrid model that resulted from this combination worked extremely well on dense traffic urban environments, in which several dynamic objects (automobiles, pedestrians, etc.) are certain to overwhelm normal models. Object detection helped improve the system to get more context regarding the scene and hence eliminate misclassification and enhance overall detection performance. Its efficiency was also boosted by good marks in the metric: Accuracy (92.8%) is a measure of how well the system performs in identifying actual accidents without sending out spurious alarms. Recall (94.1%) is a measure of how efficient the system performed in identifying actual instances of accidents. F1-Score (93.4%), harmonic accuracy and recall mean, is the combination of the two, an estimate of overall efficiency. Aside from user experience, algorithmic performance was also subjected to extreme testing. During field testing, the system was also subjected to real operators to validate the usability of the interface and process as a whole. The feedback was extremely positive. The Graphical User Interface (GUI) was said to be intuitive and simple to use even for the most non-technical operators. The video logs were organized correctly, and the alert was visually and aurally recognizable, facilitating quick human response when necessary.

System logging proved to be a handy feature, and that was automated as well. Whatever the event was discovered and the time stamp that came with it, video segment, and alarm condition, they were all logged methodically. Real- time auditing was also possible as well as audit and analysis based on event-following and was serving helpful feedback to traffic authorities for auditing accidents and establishing fact-based traffic regulations. Accident frequencies with respect to time or space, for example, can be used in order to decide on perilous areas where increased traffic regulation or construction is necessary.

In conclusion, the test results clearly indicate that the proposed system not only is technically viable but also practically applicable for use in real-world traffic monitoring and emergency systems shown in fig 5 and fig 6. Its response rate, accuracy, and usability make it a likely candidate for application in smart city systems and smart traffic control systems. Real-time auditing was also possible as well as audit and analysis based on event-following and was serving helpful feedback to traffic authorities for auditing accidents and establishing fact-based traffic regulations. Accident frequencies with respect to time or space, for example, can be used in order to decide on perilous areas where increased traffic regulation or construction is necessary. issue of road accidents and the death tolls they carry, thus indicating the importance of such detection systems in reducing their effect. The message suggests that employing CCTV-based accident detection will help in managing this serious problem effectively.

Fig.5. No Accident Detected

Fig.6. Accident Detected

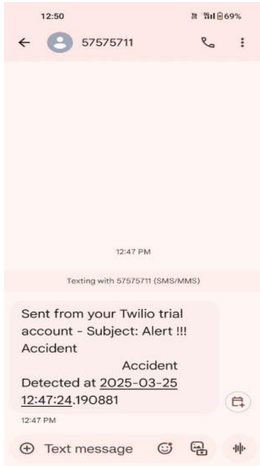

Fig.7. Alert Message Sent to Mobile

An auto-created trial account SMS notification is utilized as proof of detecting an accident given in fig 7. The notification is a brief but mandatory message with the title line "Alert !!! Accident"[8] and quotes the date and time of the incident in minute detail. It would typically be part of an AI-powered accident detection platform tracking real-time accidents and notifying concerned authorities or parties in real time. These systems enhance efficiency in emergency response by offering immediate reporting of accidents to allow for quick intervention on damage control or victim rescue.

The performance of the three machine learning models at identifying accidents is given in the table 1. The models are benchmarked against their measurements, which measure the level of performance of each model compared to the other. CNN is best among all the models at the 94.8% accuracy, 95.5% precision, and 94.2% recall levels and is therefore the best to use for real-time accident classification. The reason why it can automatically learn complicated spatial features from raw video data is that it can automatically discover finer patterns indicating accidents. SVM is substituted by fair performance, including 90.3% accuracy, 91.1% precision, and 89.6% recall, and will be well executed if there are good features to manually extract but will not execute properly with noisy or difficult data. KNN is performance-wise average on 84.5% accuracy, 83.2% precision, and 85.7% recall. Its reliance on distance calculations renders it extremely vulnerable to outliers and high-dimensional features that exact a cost on its performance on dynamic cases like live video feeds. CNN is the top-performing model overall for accident detection, but SVM and KNN are better suited to less complex or better-preprocessed cases. The performance of the three machine learning models at identifying accidents is given in the table. The models are benchmarked against their measurements, which measure the level of performance of each model compared to the other. CNN is best among all the models at the 94.8% accuracy, 95.5% precision, and 94.2% recall levels and is therefore the best to use for real-time accident classification. The reason why it can automatically learn complicated spatial features from raw video data is that it can automatically discover finer patterns indicating accidents. SVM is substituted by fair performance, including 90.3% accuracy, 91.1% precision, and 89.6% recall, and will be well executed if there are good features to manually extract but will not execute properly with noisy or difficult data. KNN is performance-wise average on 84.5% accuracy, 83.2% precision, and 85.7% recall. Its reliance on distance calculations renders it extremely vulnerable to outliers and high-dimensional features that exact a cost on its performance on dynamic cases like live video feeds. CNN is the top-performing model overall for accident detection, but SVM and KNN are better suited to less complex or better-preprocessed cases.

Metric	K-Nearest Neighbours (KNN)	Support Vector Machine (SVM)	Convolutional Neural Network (CNN)
Accuracy (%)	84.5	90.3	94.8
Precision (%)	83.2	91.1	95.5
Recall (%)	85.7	89.6	94.2
F1 Score (%)	84.4	90.3	94.8

Table 1. Comparision of CNN, SVM, KNN

In the fig 8 the bar graph compares the three machine learning models in terms of accuracy. CNN achieved the highest accuracy at 94.8% with respect to its applications in more complex pattern-based task, such as images and video recognition. SVM is the following model whose accuracy was set to 90.3%, showing its effectiveness in classification problems under well-separated data. KNN reaches 84.5% accuracy. It is effective on small data sets but very sensitive to noise and distribution, as it was more of a clustering algorithm than suited for a supervised classification task, Thus is optimal to apply in real-time classification of accidents. The reason why it can automatically learn complex spatial characteristics from raw video data is because it can detect more subtle patterns for accidents automatically. SVM is replaced by decent performance and will be done well when there are good features to manually extract but will not be done well with noisy or hard data. KNN is performance-wise average Its use of distance calculation makes it extremely sensitive to outliers and high-dimensional features, which have detrimental effects on its performance in dynamic environments like real-time video streams. CNN is the best-performing model overall for accident detection, but SVM and KNN are more suitable for simpler or better-preprocessed ones.

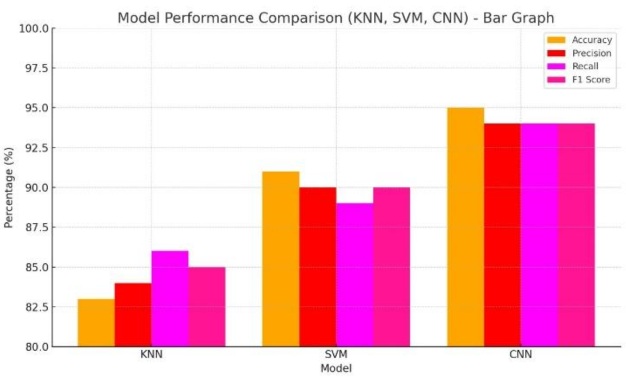

Fig.8. Accuracy Comparision of different Algorithms

The proposed design solution holds distinct benefits over known designs of the existing systems since this system achieves true real-time mapping. The system maintains reasonable accuracy without resource-consumingly vast computational architectures but is however lower than such solutions based upon deep learning if put to execution and complexity besides challenging operation and environmental issues.

One of the major strengths of this system is scalability, where it can be deployed on moderately hardware-capable devices such as laptops, tablets, and smart TVs. This also makes the system versatile, despite those strengths, however, the system has many weak points. It largely depends on the quality of the hardware, specifically with the resolution and frame rate. Performance is very sensitive with various cases, somewhat limiting its general functionality. Improvement is needed in quite a few areas.

The inculcation of deep learning techniques[8] would focus further on recognition with improvements better, even under complex scenarios. The system could also dynamically adapt through dynamic calibration methods with the help of adjustments related to user conditions. All such improvements will enable the system to be more robust and viable for several kinds of applications.

In summary, the proposed system shows its potential performance under perfect conditions with the highest accuracy. However, in the real world, it should be further tuned to increase the robustness and adaptability functions.

4. CONCLUSION

Deep Learning Accident Detection and Alert System is a computer-based, real-time system that seeks to improve road safety through rapid identification of accidents and alerting emergency contacts. Conventional detection is based on human eye, and delays are fatal. The system applies computer vision and deep learning technology, i.e., Convolutional Neural Networks (CNNs), to detect accidents from CCTV camera feeds by looking for abnormalities such as collisions or abnormal motion. Once the accident is identified, it provides real-time alerts with information such as time, location, and type of incident through SMS or email. It is comprised of modularity with preprocessing, segmentation, classification, and alarm generation and has support for live and recorded video input. The system is adjustable for application in different environments like city streets and countryside when combined with GPS and IoT technologies. It also supports smart city projects and is useful for officials as it facilitates ongoing monitoring, storage of data, and self-reporting.Lastly, this project demonstrates high applicability of AI and Convolutional Neural Networks (CNNs) in public safety through instantaneous accident detection, lowering emergency response times, and even saving lives through automated notification systems.

5. REFERENCES

[1] D. K. Yadav, Renu, A. I. Anjum, "Accident Detection Using Deep Learning," IJACSA, vol. 11, no. 5, pp. 23-29,2020.

[2] S. Ghosh, S. J. Sunny, R. Roney, "Accident Detection Using CNNs," J. Traffic Transp. Eng., vol. 6, no. 3, pp. 145-152,2019.

[3] T. Kalyani et al., "Accident Detection and Alert System Using GPS and GSM," IJITEE, vol. 8, no. 5, pp. 88-92,2019.

[4] World Health Organization, "Global status report on road safety," WHO Publications,2018.

[5] M. Hadjioannou and G. C. Hadjichristofi, "Emergency response systems using video analytics," IEEE IWCMC, 2012.

[6] H. A. Attia, S. Ismail and H. Y. Ali, "Vehicle safety distance alarming system," 2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), Ras Al Khaimah, 2016, pp. 1-4, doi: 10.1109/ICEDSA.2016.7818501.

[7] S. H. Sankar, K. Jayadev, B. Suraj and P. Aparna, "A comprehensive solution to road traffic accident detection and ambulance management," 2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES), Putrajaya, 2016 , pp. 43- 47 , doi: 10 . 1109 / ICAEES.2016.7888006.

[8] M. Hadjioannou and G. C. Hadjichristofi, "A secure dynamic data aggregation system for citizen safety in emergency response scenarios," 2012 8th International Wireless Communications and Mobile Computing Conference (IWCMC), Limassol, 2012, pp. 689-696, doi: 10.109/IWCMC.2012.6314288.

[9] A. Ali and M. Eid, "An automated system for Accident Detection," 2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings, Pisa, 2015, pp. 1608- 1612, doi: 10.1109/I2MTC.2015.7151519.

[10] B. S. Anil, K. A. Vilas and S. R. Jagtap, "Intelligent system for vehicular accident detection and notification," 2014 International Conference on Communication and Signal Processing, Melmaruvathur, 2014 , pp. 1238 - 1240, doi: 10 . 1109 / ICCSP.2014.6950048. 4