Global Journal of Engineering Innovations & Interdisciplinary Research

Original Article

Accident Detection and Alert System

T. Sai Lalith Pasad[1], Voruganti Lokesh[2], Bunga Sampath[2], Bathula Hariteja[2]

[1]Assistnat Professor, Department AI & DS Science, Vignan Institute of Technology and Science, Hyderabad, India

[2]UG Student, Department of AI&DS, Vignan Institute of Technology and Science, Hyderabad, India

Correspondence

DR.B. Ravikrishna

Associate Professor, Department of Artificial Intelligence and Data Science, Vignan Institute of Technology and Science, Hyderabad, India

- Received Date: 25 May 2025
- Accepted Date: 15 June 2025
- Publication Date: 27 June 2025

Keywords

Real-time Detection; Automated Alert System; Traffic Monitoring; Convolutional Neural Networks (CNNs); Computer Vision;Video Feed Analysis; Surveillance Cameras; Emergency; DeepLearning; OpenCV.

Abstract

Road safety is a critical concern, with traffic accidents posing significant risks to human life and infrastructure. Accidents are considered one of the most serious problems for human population and cause great disruption to traffic flow worldwide. The effects especially loss of life and disruption of traffic could be addressed by timely detection of these events in real time and notifying relevant authorities. The goal of this proposal is to research on how to design and implement Automated Accident Detection and Notification System by using modern day image analysis and deep learning tools like CNN. Using cameras as input, the system detects unusual vehicle dynamics such as sudden stops and irregular vehicle movements that are precursors of an impending accident. In the case where additional learning is required to classify events, the proposed solution incorporates the use of OpenCV to pre-process and retrieve motion information from video sequences and a set of CNN models to classify the learned features of the encapsuled event video related to an accident.

The system presented progresses the new studies in which, through the use of video, irregularities in traffic flows are quantified. Borrowing techniques of motion estimation like the Farneback Optical Flow and adaptive thresholding for anomaly detection, a solution to the problems is presented in the form of deep learning models. The objective is for accuracy though at low computation making it deployable in different settings like highways and urban intersections. The final goal of the project is improving road safety in general by fast-tracking responses during emergencies and limiting the aftermath of traffic accidents.

Introduction

Accidents have become a universal concern to various nations, resulting in numerous deaths, serious injuries, and damage to infrastructures. With the increasing complexity of urban roads and worsening automobile crowding, the accident incidence rate has worsened, thus making road safety a matter of concern to most governments, organizations, and communities worldwide. Traditional accident detection relies on either manual reporting or human observation. This causes a delay in responding to emergencies and does not help in the mitigation process of accidents. With new technological advancements in artificial intelligence and computer vision, there is now a chance for many automated systems to detect accidents and respond immediately.

This project will be based on designing an accident detection and alert system with the help of deep learning models with the aid of CNNs, popular because they are effective in image and video processing-based tasks. This use of real-time CCTV footages streaming from the traffic monitoring systems as well as surveillance cameras can be beneficial in enhancing the response of emergency services, thus reducing their reaction times after the accident. The integration of computer vision techniques with real-time video evaluation will allow the system to alert for abnormal events such as sudden vehicle collisions or unpredictable movement, which may indicate an accident.

The heart of the system is the deployment of a pre-trained CNN, which is fine-tuned with a custom dataset specifically curated for accident scenarios. The CNN takes in the video feed's individual frames and extracts features that enable it to classify the scene as either accident or non-accident. Using the high precision of deep learning models, the system ensures accuracy and reliability in accident detection, reducing false negatives and false positives.

The progress in this system has great potential for transforming the way traffic accidents are handled. This ensures that emergency responders are notified well ahead of time, which saves a lot of time to give medical care and clear traffic obstructions. The capability of the system to continuously function and work independently is an asset for urban planners, traffic managers, and local government institutions fighting to make the infrastructures for roadways safer.

This project also showcases the transformative nature of AI and deep learning in addressing fundamental real-world issues. Integration with the advanced technology of CNNs and computer vision proves the ways through which novel concepts can enhance

Citation: Pasad TSL, Voruganti L, Bunga S, Bathula H. Accident Detection and Alert System. GJEIIR. 2025;5(4):082.

quality life while ensuring public safety.

Given the expansion of cities, and rising automobile use, such systems will be integral in forming a transportation ecosystem that is both sustainable and secure.

In conclusion, the Accident Detection &Alert System Using CNN Feeding on Live CCTV Footage represents a giant leap forward in utilizing technology to address the pressing concern of road safety. The system is able to detect accidents in real time, give accurate alerts, and be integrated with existing infrastructure, offering a holistic solution that could save lives, minimize injuries, and reduce the societal impact of traffic accidents. It brings a more responsive, smarter, and resilient urban environment by filling in the gap between detection and response.

Related work

The Accident Detection and Alert Systemwasdevelopedby using machine learning, computer vision along with real-time processing technologies.

This section describes the step-by-step implementation process,the tools and frameworks used, and how these technologies work together to achieve the project's goals.

System Architecture

The system architecture consists of three major components: video input, accident detection, and alert generation. The input is a video feed, either pre-recorded or live feed from a camera. This feed is processed frame by frame to detect accidents using a pre-trained deep learning model. When an accident is detected, the alert system will be activated, which may include displaying the accident on screen and playing an audio notification to attract attention.

Technologies Used

1. **Machine Learning Frameworks:**

TensorFlow/Keras: This convolutional neural network model classifies video frames as "Accident" or "No Accident." It learns features like sudden motion changes, deformation of the vehicle, and collision.

2. **Numpy:**

Numpy is used for performing mathematical operations such as preparation of video frames to be passed on to the model and normalizing pixel values.

3. **Computer Vision:**

- ***OpenCV:*** Video feed processing was done by OpenCV, that included video frame capturing, resizing it for prediction, accident detection annotation, and printing the output in real time.
- ***Frame Preprocess:*** Frames are resized to a fixed size of (250x250) pixels and converted to RGB format before feeding to the model

4. **Audio Alerting Mechanism:**

- ***Winsound/Playsound:*** For Windows, the winsound library produces frequency-modulated beeps based on accident probability. For non-Windows, the playsound library plays a pre-recorded alert sound.

5. **Messaging Service:**

- ***Twilio API:*** The Twilio SMS API was integrated to send automated text messages to preconfigured emergency contacts when an accident is detected. The message includes details such as the detection timestamp and the location (if available).
- ***Message Format:*** Concise message format is used such as:" Accident detected at [location]. Probability: 95%. Immediate response required. "

This helps recipients receive action information in real time.

6. **Graphical User Interface:**

- ***Tkinter:*** Tkinteroffers a basic and user-friendly interface for the monitoring of video feeds and system controls. It offers real-time prediction displays and an option to begin or stop video feed.

Proposed approach

Our proposed approach is one for detecting highway accidents through patterns of traffic flow. A flow of vehicles in a highway normally represents a regular flow; every moving vehicle can be seen having a relatively constant value and direction in its velocity vector. As long as these are reasonable and comply with the typical settings of the highway and usual practices in driving, slight alterations can be accommodated.

However, if there is an accident where one or more of the vehicles are involved, then all of a sudden and significantly changes the velocity vectors of the involved vehicles. Such deviation can take a variety of forms: a huge difference in the magnitude or speed or both. A significant change, whether in speed or direction of the velocity vector, or possibly both, thus is an essential sign of an accident. In order to numerically quantify such changes, we employ mathematical expressions of equations (1) and (2) developed to sense the anomalies.

Our approach starts with preprocessing the data to remove noise, ensuring that extraneous fluctuations do not interfere with the analysis. We then construct a robust model that captures the dynamics of normal traffic flow. Comparing real-time traffic data against this baseline model will allow us to reliably identify instances where the movement deviates significantly, signaling a potential accident. This systematic approach helps to distinguish between normal fluctuations in traffic and critical events requiring immediate attention.

$$\|\vec{V}\| = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{1}$$

$$\theta = tan^{-1}\frac{(y_{i+1} - y_i)}{(x_{i+1} - x_i)}.\frac{180}{\pi} \tag{2}$$

It is also important to note that the parameters associatedwith velocity and orientation depend on the particular location of the highway and on the calibration of the camera used for monitoring. This means that accident identification cannot rely on generic parameters but rather requires the development of a location-specific traffic model. This can be done by retrieving the parameters which define normal motion flow of traffic from multiple scenes shot by the same camera. This enables us to set up a baseline or standard traffic model for the given monitoring area that will be especially specific to conditions.

The standard traffic model introduces several thresholds as reference points for the identification of irregular motion patterns. The thresholds are not fixed but change according to every newly received frame of video. It thus reflects the ongoing traffic situation aptly. Initially, one set of thresholds is obtained from analyzing the motion flow in the first Nnumber of frames of the video. These N number of frames are necessary to construct the basis understanding of normal traffic flow as they would be capturing a snapshot of typical motion dynamics in normal conditions. Selection and its significance, together with its effects on accuracy in the model, are further explained in the subsequent section.

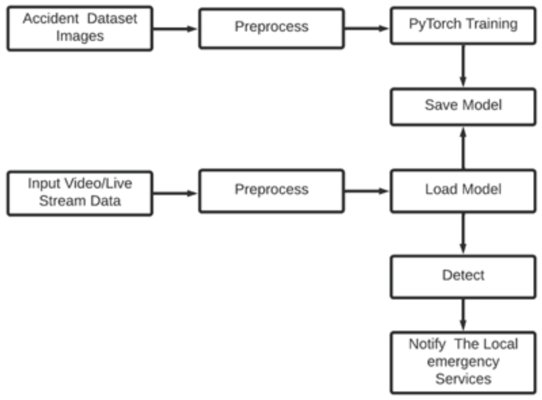

Figure 1: State diagram of model

Noise filtering and Velocities classification

The general traffic model was built by analyzing the traffic motion on field, which captures the dynamic movement of traffic within a specified area. This was achieved using the Farneback Optical Flow (OF) method, which calculates the displacement of every moving pixel (or particle) between successive frames within a defined Region of Interest (ROI). The ROI is confined with the highway road in order to only focus on vehicular motion for the purpose of accident detection on highways.

We identified only two categories of velocity vectors on closer observation: video artifacts such as dust, wind, or other environmental factors, and those which represent the movement of vehicles. The Optical Flow technique is sensitive to changes in lighting conditions along with noise, thus often producing incorrect vectors. Such vectors are mostly of small magnitudes and thus less relevant for analysis. To address this, a threshold is established to filter out vectors with magnitudes below a certain value.

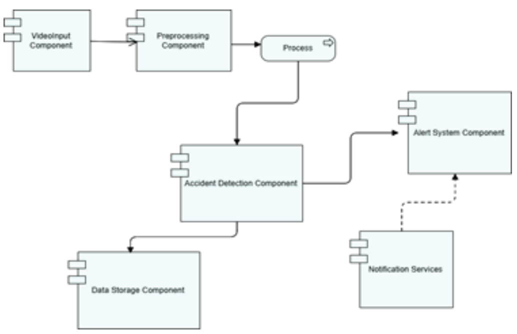

Figure 2: Component diagram of ADAS model

The threshold value is determined by analyzing Optical Flow across fixed number of successive frames which contain no traffic are present. For our model, we fixed this number at 10 frames. This threshold is normally computed once and depends on both the camera's location and camera's calibration settings. This ensures that only meaningful motion data is considered for further analysis.

Given an moving pixel 'P' with coordinates (xi,yi) in framei, OF computes its coordinate in the next framei+1 (xi+1 yi+1).

After eliminating the error vectors, the remaining velocity vectors, which relate to vehicular movement, fall into two groups:

- Vectors oriented along the normal flow of traffic – These describe typical vehicle movement along the highway.
- Distracting vectors with other orientations – These may result from various factors such as wheel movement, shadows on the road, or reflections from the vehicle's body due to changing light conditions.

This categorization will make the model distinguish and give priority to meaningful motion patterns while discarding noise and irrelevant data, thus making an accurate accident detection.

Motion and flow modeling

Our approach to modeling normal traffic motion flow is graphically depicted in a flowchart, designed for a one-way road. This step-by-step process involves several important steps that help analyze and model the traffic dynamics.

Step 1: Motion Detection

It starts by applying Optical Flow (OF) to every video frame i in the video sequence. Optical Flow will detect the pixel movement between consecutive frames to allow movement tracking within the region interest i.e. ROI. Subsequent to this is noise filtering to filter out spurious vectors generated from environmental artifacts or luminosity variation.

Step 2: Extraction of the Features

For every framei, we extract the feature through motion vectors being categorized into two separate categories:

- Those moving in line with the direction of normal traffic flow: SVA (these lie in a particular angle range; these are those in the expected vehicle motion directions. This range is preset as known and calibrated from the position of the camera).
- Distracting vectors (SVB): These are vectors with orientations outside the normal range, caused by factors such as wheel rotations, shadows, or light reflections on the road and vehicle surfaces.

The sum of these vectors (SVA and SVB) is computed for each frame i. Together, SVAi and SVBi form a sub-model that represents the traffic motion characteristics for that specific frame.

Step 3: Velocity Analysis

The SVA and SVB values vary over several frames in a sequence of normal traffic conditions. Analyzing the velocity trends, we notice that the closer the vehicles are to the camera, the larger their velocity vectors are. This relationship illustrates how perspective affects the motion.

Step 4: Statistical Modeling

To explain these fluctuations, we assume that SVA and SVB changes follow a binomial distribution. For N number of frames, we compute the mean values of SVA and SVB along with their standard deviations ($\sigma 1$,$\sigma 2$). These statistical metrics are preserved as a list LN that consists of N sub-models representing particular frames that make up the sequence.

Step 5: Threshold Calculation

Using the gathered data, thresholds THA and THB are calculated based on equations (3) and (4). These are the boundaries of what constitutes normal traffic motion. They are the base of the normal traffic model.

To improve the model's accuracy, we incorporate a tunable constant α that may be set experimentally for maximum performance. Analogously, parameters α and N are obtained from experimental verification and are set in the final design.

$$THA = \frac{1}{N}\Sigma_1^N (SVA)_i + \lambda\sigma_{1,i} \qquad (3)$$

$$THB = \frac{1}{N}\Sigma_1^N (SVB)_i + \lambda\sigma_{2,i} \qquad (4)$$

This structured methodology ensuresthe traffic model is accurate and adaptable enough to capture the nuances of normal traffic flow, filtering out noise and distractions. The resulting thresholds enable reliable detection of deviations that may indicate potential accidents or anomalies.

Implementation

The Accident Detection and Alert System relies on efficient model training and real-time video processing, and a robust alert mechanism. A CNN was trained on a labelled dataset of both accident along with non-accident frames. Important steps included preprocessing the frames, defining the CNN architecture, and optimizing the model for accurate and fast inference. The trained model, saved as model.json and model_weights.keras, ensures portability and is loaded during runtime. OpenCV is used to capture video frames from live camera feeds or pre-recorded videos. The system resizes each frame and feeds it through the model for accident prediction. If the prediction confidence exceeds a predefined threshold (e.g., 90%), the system overlays the detection results on the video feed, which gives real-time visual feedback.

The alert mechanism includes both audio alerts and a messaging system to respond in real time. A dynamic beep sound notifies the local observer of the occurrence; the frequency and duration reflect the model's confidence level. Meanwhile, Twilio API triggers SMS alerts for predefined emergency contacts. It will outline the type of accident detected, confidence score, and optional location information if that information is available. The system runs at a seamless 30 FPS, with error-handling capabilities for its reliability during an interruption. Messaging can be scalable to extend its capabilities for sending email notifications or integrating into emergency response networks for more versatile usage.

Implementation Workflow

Model Training:

The accident detection model is trained on a labelled dataset of accident as well as non-accident frames. Major steps included preprocessing frames, defining a CNN architecture, and optimizing the model for realtime inference.

The model trained, saved as model.json and model_weights.keras, was loaded during runtime to ensure portability.

Video Processing:

The video frames are grabbed from either the video capture object representing an open camera or a file. To process, each frame is captured one after another to detect accidents.

Frames are resized and passed to the model for prediction. For a probability higher than a given threshold about "Accident" (e.g., 90%), the system overlays the prediction result over the video feed.

Alert Mechanism:

Audio Alarms: It emits a beep sound if it has sensed an accident. The rate and period of the beep depends on the confidence score; this provides feedback to the user intuitively.

Messaging Service: In parallel, it sends an SMS through Twilio API to predefined emergency contact numbers. It includes the following information:

Short description of the accident that is detected, The probability score which will tell about the model's confidence.

Optional details such as location if GPS data is available.

Real-Time Processing and Error Handling:

The system processes video feeds at approximately 30 frames per second (FPS), which means that it can run seamlessly in real time.

Built-in error handling will ensure that the system continues to run smoothly in case the video source is interrupted or if the model is experiencing issues.

Messaging Integration:

Emergency contacts are predefined in the system. Upon attaining the detection threshold, the system will automatically send text messages using the Twilio API within a fraction of a second.

The messaging can be further expanded to include emails or even be integrated with an emergency response system.

Learning

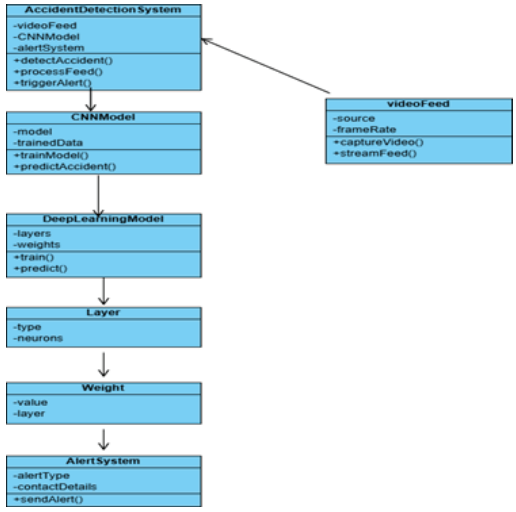

Figure 3. Object diagram of model

Data Set

The dataset which is used to train and test the accident detection model containsa total of approximately 1,603 videos where each one has a duration of 3 seconds and contains 100 frames,comprising920 positive videos where accidents occur within the last 15 to 20 frames and 683 negative videos without accidents. All videos are recorded in 720x1280 resolution so that it is ensured that model training has high-quality input. The data, which was collected from six major cities in Taiwan, provides a diverse range of accident scenarios and environmental conditions to improve the generalizability and effectiveness of the model.

For training and testing the model, the dataset is divided into 77% training data i.e.(1,234 videos) and 23% testing data i.e.(369 videos). The dataset covers different types of accidents, such as 52.6% motorbiketocar collisions, 13.7% cartocar collisions, 15.4% motorbiketomotorbikecollisions,and 18.3% involving other types of accidents. This detailed categorization and labeling of accident frames, combined with the variety of collision types, provide a comprehensive foundation for the training an accurate and reliable accident detection system suitable for realworld scenarios.

Table I. Distrubution of data set

	POSTIVE	NEGATIVE	TOTAL
TRAINING SET	708	525	1233
TESTING SET	212	158	370
TOTAL	920	683	1603

Figure 4: *Model generated testing split image*

Implementation

The Accident Detection & Alert System will use video feeds to accurately and timely detect accidents, and the alert for-the emergency services in real time. Measures the efficiency ofthe system in identifying the occurrence of accidents using real-time video data that might be indicative of a sudden vehicle stop, collision, or swerving. The accuracyof the system is determined bycomparing the results obtained from the system with ground truth data, in order to minimize false positives (non-accidents mistakenly identified as accidents) and false negatives (missed accidents). A low false positive rate ensures that the emergency services were not unnecessarily alerted while minimizing false negatives is essential in detecting all accidents.

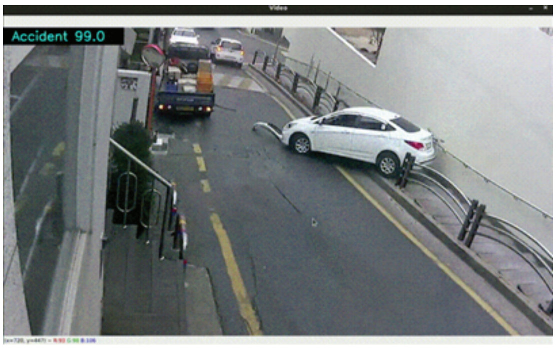

Figure 5: *Real-time demonstration on CCTV*

Validation

Training Loss

Training loss measures how bestone's model is performing on the training data provided. Training loss will be calculated by the evaluation of the difference of predicted output and the actual target values while training.This Training Loss indicates whether the model's learning from the data provided or not. A decrease in training loss typically shows that the model is improving its performance and giving better predictions for the training set.

Training Accuracy

Training accuracy is the percentage of correctly predicted outcomes to that of the total predictions for the training data. Training Accuracy gives the measure on how well the model is fitting the training data. High training accuracy implies the model'sidentifying the patterns present in the training Dataset.

Validation Loss

Validation loss refers to the error calculated on validation dataset; a separate subset of data is not used for training. It assesses how goodone's model is able to generalize to unknown data. If validation loss decreases, it indicates the model is

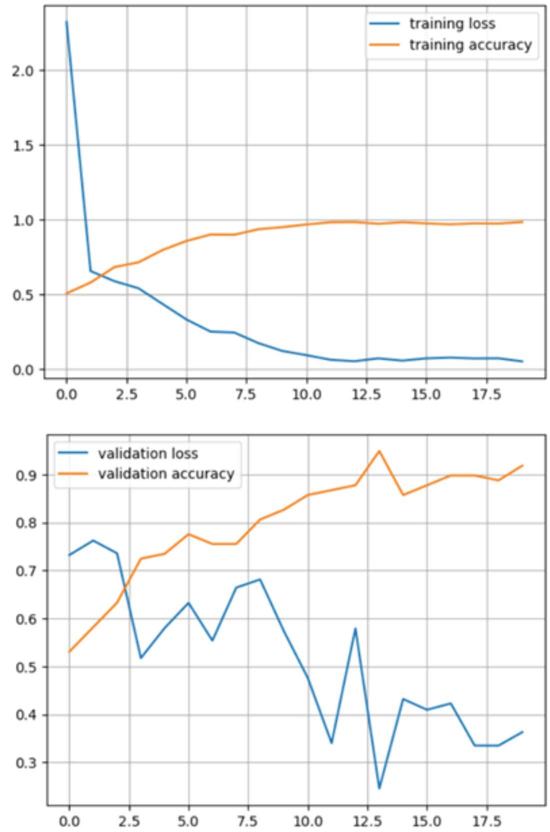

Figure 6:Validation loss v/s accuracy and Training loss v/sAccuracy

learning patterns which can generalize beyond the given training data.

Validation Accuracy

Validation accuracy is the measure of the percentage of the accurately predicted outcomes on the validation dataset.It's an indicator of how good the model performs on unknown data. High validation accuracy with low training accuracyimpliesunderfitting, while the reverse might indicate overfitting.

References

1. N. Buch, S. A. Velastin, and J. Orwell, "A review of computer vision techniques for the analysis of urban traffic," IEEE Transactions on In telligent Transportation Systems, vol. 12, no. 3, pp. 920–939, September 2001.
2. S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 4, pp. 1773–1795, Dec 2013.
3. E. B. Lerner, MS, EMT-P, R. M. Moscati, and MD, "The golden hour: Scientific fact or medical "urban legend"?" Official Journal of the Society for Academic Emergency Medecine, vol. 8, no. 3, pp. 758–760, July 2001.
4. "National observatory on road safety road accidentality 2015-estimates as of january 26, 2016." [Online]. Available: http://www.securite routiere.gouv.fr/
5. G. Farneb¨ack, Two-Frame Motion Estimation Based on Polynomial Expansion. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 363–370.
6. B.Maaloul, A. Ahmed, S.Niar, C.Valderrama, F. Derraz, and N. Harb:, "Vision-based traffic accident detection techniques

opportunities and challenges," in International Conference on Advanced Communication Systems and Signal Processing, November 2015.

7. Y. K. Ki and D. Y. Lee, "A traffic accident recording and reporting model at intersections," IEEE Transactions on Intelligent Transportation Systems, vol. 8, no. 2, pp. 188–194, June 2007. "Okcars: Oklahoma collision analysis and response system." [Online]. Available: http://www.trb.org/
8. Z. hui, X. yaohua, M. lu, and F. Jiansheng, "Vision-based real-time traffic accident detection," in Intelligent Control and Automation (WCICA), 2014 11th World Congress on, June 2014, pp. 1035–1038.
9. O. Akooz and M. E. Karsligil, "Severity detection of traffic accidents at intersections based on vehicle motion analysis and multiphase linear regression," in Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on, September 2010, pp. 474–479.
10. Renu, Durgesh Kumar Yadav, Iftisham Anjum, Ankita, "Accident Detection Using Deep Learning: A Brief Survey", International Journal of Electronics Communication and Computer Engineering (IJECCE), vol. 11, no. 3, pp. 16-25, May 2020.
11. Ghosh, S., Sunny, S.J. and Roney, R., 2019, March. Accident detection using Convolutional Neural Networks. In 2019 International Conference on Data Science and Communication (Icon DSC) (pp. 1 6). IEEE.
12. R. Bhallamudi et al., "Deep Learning Model for Resolution Enhancement of Biomedical Images for Biometrics," in Generative Artificial Intelligence for Biomedical and Smart Health Informatics, Wiley Online Library, pp. 321–341, 2025.
13. R. Bhallamudi et al., "Artificial Intelligence Probabilities Scheme for Disease Prevention Data Set Construction in Intelligent Smart Healthcare Scenario," SLAS Technology, vol. 29, pp. 2472–6303, 2024, Elsevier.
14. R. Bhallamudi, "Improved Selection Method for Evolutionary Artificial Neural Network Design," Pakistan Heart Journal, vol. 56, pp. 985–992, 2023.
15. R. Bhallamudi et al., "Time and Statistical Complexity of Proposed Evolutionary Algorithm in Artificial Neural Networks," Pakistan Heart Journal, vol. 56, pp. 1014–1019, 2023.
16. R. Krishna et al., "Smart Governance in Public Agencies Using Big Data," The International Journal of Analytical and Experimental Modal Analysis (IJAEMA), vol. 7, pp. 1082–1095, 2020.
17. N. M. Krishna, "Object Detection and Tracking Using YOLO," in 3rd International Conference on Inventive Research in Computing Applications (ICIRCA-2021), IEEE, Sept. 2021, ISBN: 978-0-7381-4627-0.
18. N. M. Krishna, "Deep Learning Convolutional Neural Network (CNN) with Gaussian Mixture Model for Predicting Pancreatic Cancer," Springer US, vol. 1380-7501, pp. 1–15, Feb. 2019.
19. N. M. Krishna, "Emotion Recognition Using Skew Gaussian Mixture Model for Brain–Computer Interaction," in SCDA-2018, Textbook Chapter, ISBN: 978-981-13-0514, pp. 297–305, Springer, 2018.
20. N. M. Krishna, "A Novel Approach for Effective Emotion Recognition Using Double Truncated Gaussian Mixture Model and EEG," I.J. Intelligent Systems and Applications, vol. 6, pp. 33–42, 2017.
21. N. M. Krishna, "Object Detection and Tracking Using YOLO," in 3rd International Conference on Inventive Research in Computing Applications (ICIRCA-2021), IEEE, Sept. 2021, ISBN: 978-0-7381-4627-0.
22. T. S. L. Prasad, K. B. Manikandan, and J. Vinoj, "Shielding NLP Systems: An In-depth Survey on Advanced AI Techniques for Adversarial Attack Detection in Cyber Security," in 2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS), IEEE, 2024.
23. S. Sowjanya et al., "Bioacoustics Signal Authentication for E-Medical Records Using Blockchain," in 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS), vol. 1, IEEE, 2024.
24. N. V. N. Sowjanya, G. Swetha, and T. S. L. Prasad, "AI Based Improved Vehicle Detection and Classification in Patterns Using Deep Learning," in Disruptive Technologies in Computing and Communication Systems: Proceedings of the 1st International Conference on Disruptive Technologies in Computing and Communication Systems, CRC Press, 2024.
25. C. V. P. Krishna and T. S. L. Prasad, "Weapon Detection Using Deep Learning," Journal of Optoelectronics Laser, vol. 41, no. 7, pp. 557–567, 2022.
26. T. S. L. Prasad et al., "Deep Learning Based Crowd Counting Using Image and Video," 2024.
27. François Chollet https://keras.io/about/ (2020), Keras [Online], Available.