IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CNN-Based Accident Detection and Emergency Response System

Tanmay Chavan, Harsh Gharat, Sumed Ingale, Divanshu Sharma, Prof. Anuja Chandane

Department of Computer Science and Engineering (Artificial Intelligence and Machine Learning), Saraswati College of Engineering, Navi Mumbai, India

Abstract

Road accidents remain a major global challenge, requiring rapid and effective emergency response mechanisms. Traditional accident detection methods rely on manual reports or eyewitness accounts, resulting in delays in response times and increased severity of the consequences. This paper presents a CNN-based accident detection and emergency response system using YOLOv8, an advanced deep learning algorithm for real-time video analysis. The system processes live video surveillance images to detect accidents quickly, records critical details such as location, time, and severity, and triggers immediate alerts to the emergency services. The proposed system automates accident monitoring, integrating machine learning and computer vision technologies to achieve accuracy while minimizing false positives and negatives. It features a user-friendly interface for traffic authorities and emergency responders, a robust detection module powered by convolutional neural networks, and a scalable database for efficient storage and analysis of accident data. The methodology includes data collection, pre-processing, model training, evaluation, and deployment, guaranteeing the adaptability of the system, deployment, and ensuring adaptability in a variety of environments, including urban areas, freeways, and high-traffic areas. The preliminary results from controlled experiments demonstrate the system's potential to significantly reduce emergency response times and coordination between emergency services. By filling the gaps left by traditional detection systems, this study provides a scalable and reliable solution that contributes to intelligent transportation systems and improves public safety through real-time monitoring notifications and effective emergency management.

Keywords: Accident Detection; Convolutional Neural Networks; Emergency Response; YOLOv8

I. Introduction

The latest report of the World Health Organization on road traffic injuries [21] was reported on December 13, 2023. It reported that approximately 1.19 million lives are lost annually due to road traffic accidents, while an additional 20 to 50 million individuals sustain non-fatal injuries. Many of these injuries lead to long-term disabilities, profoundly impacting the victims' quality of life. Over 90% of fatalities from road traffic accidents take place in low- and middle-income nations. Road traffic injuries rank as the primary cause of death among children and young adults aged 5 to 29 years. Approximately two-thirds of these fatalities occur among individuals of working age (18–59 years), with males being three times more likely to lose their lives in road accidents compared to females. India is also witnessing a high number of road accidents and fatalities. It also drains the country's GDP by claiming lakhs of economically productive lives. Ministry of Road Transport & Highways Government of India stated the report on the present volume of 'Road accidents in India - 2022' provides information on various facets of road accidents in the country during the calendar year 2022. In 2022, police departments across States and Union Territories in India reported 4,61,312 road accidents, resulting in 1,68,491 fatalities and injuries to 4,43,366 individuals. In 2022, road accidents increased by 11.9% compared to 2021, with fatalities rising by 9.4% and injuries by 15.3%. On average, this equates to 1,264 accidents and 462 deaths daily, or 53 accidents and 19 deaths per hour nationwide. The high prevalence of accidents and injuries in India significantly impacts public hospitals and raises concerns about healthcare costs. Studies reveal severe consequences, including rising injury-related mortality, increased health expenditures, and catastrophic out-of-pocket expenditure (OOPE), defined as OOPE exceeding 10% of household spending. India’s annual hospitalization rate for accidents is 264.7 per 10,000 persons, with victims spending an average of 8.77 days in the hospital.

Reports [22] indicate that mean out-of-pocket expenditure (OOPE) for accidental care in private healthcare is five times higher than in public healthcare. Over 90% of total healthcare costs are borne as OOPE, regardless of the healthcare provider. OOPE is particularly higher for the elderly in public hospitals (₹9,104) and dramatically higher in private hospitals (₹57,663). Males incur greater healthcare costs (₹9,257 in public care), while urban private hospital expenditures are moderately high (₹45,262). Individuals with graduate-level education or higher spend more on care (₹16,159 in public care) compared to others (₹9,331).

Various hardware, IoT, and software solutions are being researched and evaluated to minimize accident risks. One hardware approach involves an Arduino Uno microcontroller, which powers and controls key components. A shock sensor detects collisions, a GPS module tracks the vehicle's location, and a GSM module sends alerts to ensure timely communication of critical information. The system's software, developed using the Arduino IDE in C/C++, monitors sensors, retrieves GPS data upon collision and dispatches messages via the GSM module. Additionally, an accident prevention and safety assistance system has been proposed for drivers. This system employs algorithms to detect driver drowsiness through continuous monitoring of behavior, creating a comprehensive dataset stored in the cloud. Machine learning algorithms classify the driver's condition using cloud-based data, aiding companies in evaluating driver suitability based on their recorded behavior. This system can analyze driving patterns by inputting a driver's ID, and in the future, it could assist traffic and law enforcement in identifying and warning drowsy or intoxicated drivers, reducing associated risks. Although similar approaches are used to detect accidents, scaling them up presents significant challenges. These solutions are often difficult to implement due to a number of factors, such as unpredictable weather conditions, varying government regulations and manufacturing conflicts. In addition, integration issues arise when adapting systems to different vehicle models or types, as hardware compatibility and sensor calibration can vary considerably. Data confidentiality issues, particularly concerning driver information, can also complicate the deployment of these systems. In addition, the cost of widespread adoption, particularly for low-income regions or countries, and the risk of technical malfunctions in real-life scenarios can hamper their effectiveness and scalability. Taken collectively, these factors make it difficult to implement these technologies universally in diverse environments and situations.

Despite advancements in road safety technologies, traditional accident detection methods, such as manual reporting and sensor-based systems, remain inefficient and prone to delays. Manual reporting relies on eyewitness accounts, which are often unreliable and time-consuming, while sensor-based systems face challenges such as hardware compatibility, high costs, and scalability issues. These limitations hinder the timely detection of accidents and delay emergency responses, exacerbating the severity of consequences. To address these challenges, there is an urgent need for automated, real-time accident detection systems that can accurately identify accidents, assess their severity, and trigger immediate emergency responses. From a strategic point of view, it is essential to evaluate the factors and repercussions of road accidents, prioritize them, and implement preventive measures accordingly. For example, accident analysis often uses statistical techniques to study road structure, weather conditions, and traffic flow. This information can guide the introduction of safety interventions such as cable barriers or rumble strips to reduce the probability or impact of accidents.

On the other hand, operational analysis focuses on providing drivers with accurate, real-time risk information. Machine learning models are commonly used to detect incidents or predict collision risks, with a focus on accuracy rather than interpretation of variables. While it can be useful to understand variable influences, such as weather, many of these factors are uncontrollable. Therefore, it remains essential to focus on operational data, as accurate and timely predictions or detections have the potential to save lives and minimize delays.

In recent years, the detection of road accidents has become an important application of computer vision, aimed at tackling the difficult task of ensuring rapid administration of first aid without the need for constant human supervision.

The proposed system improves upon existing solutions by leveraging the advancements of YOLOv8, ensuring higher accuracy and faster processing speeds. Compared to YOLOv5, which is efficient but lacks the latest architectural enhancements, our system achieves better detection accuracy and real-time performance. In contrast to Faster R-CNN, which is known for its high precision but suffers from computational inefficiency, our model balances both accuracy and speed, making it suitable for real-time applications. Similarly, while SSD is a fast detection model, it falls behind in precision and recall, limiting its reliability in accident detection. Our system outperforms SSD by offering a higher mean Average Precision (mAP) and improved real-time processing, making it a more dependable choice for real-world deployment in accident detection and emergency response. This paper presents a practical solution to this critical problem by proposing a system capable of instantly detecting collisions between vehicles. Such rapid detection is essential to enable local paramedics and traffic authorities to respond effectively and mitigate the situation quickly. The proposed solution relies on a state-of-the-art supervised deep learning framework to identify a variety of well-defined road objects, trained on comprehensive datasets. The outputs from the neural network are then processed to extract key feature points and define custom parameters that assist in identifying vehicular collisions. Our approach is specifically designed to operate effectively under real-world conditions, accommodating variations in daylight, weather, and other environmental factors. The defined parameters enable the detection of distinctive features associated with vehicular accidents by identifying anomalies in vehicle motion observed through the framework. Furthermore, the system is intended to assist human operators by facilitating the review of surveillance footage and efficiently recognizing vehicular accidents using the proposed methodology.

This paper is structured as follows: Section II provides an overview of related work. Section III introduces the proposed system of research purpose and delves into the implementation details of all utilized algorithms on the accident detection module severity estimation, along with a detailed explanation of the materials and techniques employed. Section IV outlines the model output and results. Section V concludes the paper by summarizing the key findings, addressing study limitations, and proposing future directions.

II. Related Work

In this section, we will go through various papers and studies and analyze them to understand their work and solution techniques for detecting accidents and notifying alerts from the very beginning till the year 2024.

A. Vision-Based Traffic Surveillance

Recent advancements in onboard automotive driver assistance systems have drawn significant attention due to their potential to alert drivers about their surroundings and prevent collisions with other vehicles. These systems primarily rely on robust and reliable vehicle detection mechanisms to function effectively. A key focus of these systems is the integration of cameras mounted on the vehicle, which differ from fixed cameras used in traditional traffic or driveway monitoring systems. This paper emphasizes the importance of reliable vehicle detection in dynamic environments, where the camera moves along with the vehicle, adapting to real-time traffic conditions [1]. New vehicle and traffic detection techniques are evolving, video-based traffic monitoring being one of them. For many years, research has focused on vision-based intelligent transportation systems (ITS), moving vehicle detection and vehicle detection segmentation approaches in traffic monitoring, exploring various methods for detecting and segmenting moving vehicles in dynamic scenes. Three main approaches are used: background subtraction, feature-based methods and frame differentiation.

Vehicle detection in traffic monitoring explores various methods for detecting and segmenting moving vehicles in dynamic scenes. Three main approaches are used: background subtraction, feature-based methods and frame differentiation. Background subtraction involves detecting moving objects by comparing current frames with background models, although challenges such as lighting changes persist. Feature-based methods focus on analyzing vehicle features such as edges and corners to identify moving objects, offering better occlusion management and computational efficiency. In addition, techniques such as PCA, SVM, curve transforms and eigen window methods have been proposed to improve vehicle recognition and tracking. These approaches demonstrate high accuracy and robustness in various traffic scenarios. [2] Accurate measurement and tracking of vehicle speed depends on efficient camera calibration, which can be performed semi-automatically or manually. Camera calibration is crucial for video surveillance systems. One method introduces an automatic approach for segmenting and tracking vehicles in videos taken from a low-angle camera on freeways. This method combines region-based grouping, background subtraction, projective transformation, and plumb line projection (PLP) for feature calculation.[2]A novel automatic traffic system using 2D spatiotemporal images has been proposed, where a TV camera tracks vehicles on highways through two slice windows per lane. The system classifies vehicles based on 3D measurements (height, width, and length), counts them, and calculates their speeds. It performs robustly under various lighting conditions, including night vehicle lights and daytime shadows.[2]

Figure 2.1 illustrates the automatic traffic system's performance under different lighting conditions.

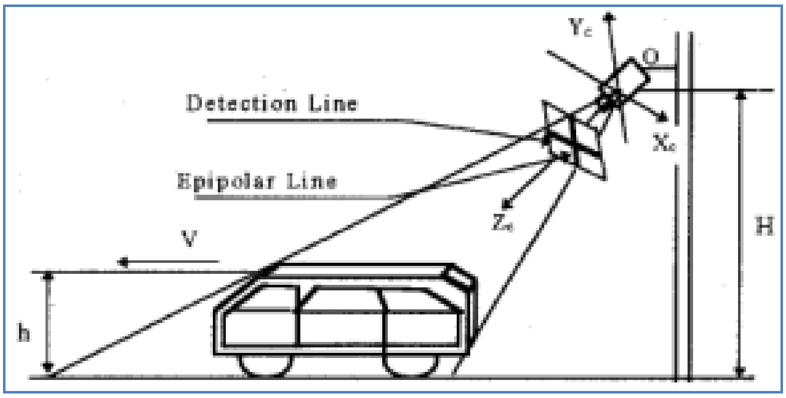

Figure 2.1 Camera Setting

B. IoT Based Techniques

AI-Enabled Accident Detection and Alert System Using IoT and Deep Learning for Smart Cities [12] The system uses a force sensor (mounted on the vehicle to detect impact), an ATMEGA328 microcontroller (to process sensor data and trigger the alarm if speed or force exceeds predefined thresholds, Tspeed and Tforce), and an alarm system (which activates for 30 seconds). If the accident is minor, the driver can reset it with a button. The NodeMCU (Wi-Fi microcontroller) sends accident data to the driver's phone, and the ESP8266 module (sends video, photos, and GPS data to the master controller). The Raspberry Pi (master controller) processes and stores the data, while the GPS module (displays the vehicle's location on an LCD screen).

C. Deep Learning and CNN Methods

Accident Detection using Convolutional Neural Networks [5] leverages a Raspberry Pi 3 B+ Model as a portable and remote computational device linked to a CCTV camera. For demonstration purposes, a Pi Camera is seamlessly integrated with the Raspberry Pi. [10] An Inception v3 model, trained on datasets comprising 10,000 severe accident images and 10,000 non-accident images, achieves an impressive 98.5% accuracy in detecting accident-related frames. Built using TensorFlow, OpenCV, and Keras, the model analyzes video streams on a frame-by-frame basis. Traffic Accident Detection Using Background Subtraction and CNN Encoder–Transformer Decoder in Video Frames [19] This paper presents a traffic-accident-detection framework combining a CNN encoder and Transformer decoder. It operates in two stages: the Bounding-Box-Masks-Extractor uses YOLOv5 to detect bounding boxes and subtract the background, while the Traffic-Accident-Detector uses the CNN encoder for spatial features and the

Transformer decoder for spatiotemporal features across multiple frames to detect accidents. The framework's performance was evaluated by comparing results with and without background subtraction to assess its impact on accuracy and robustness. A trained TensorFlow model is loaded and applied to an image. The model tests the image and returns the location of the object within it. This approach utilizes TensorFlow's object detection algorithm, which offers high accuracy and can be applied to RGB images. To start the tracking process, the locations of the objects are required. Instead of using traditional convolutional computer vision-based algorithms, this method employs a CNN-based tracking algorithm for enhanced performance [14] Global Average Pooling (GAP) [17] is a technique used in deep learning to reduce the number of parameters in a network, helps to prevent overfitting. Instead of using fully connected layers after convolutional layers, GAP computes the average value of the feature map across the height and width dimensions, while keeping the channel dimension intact.

The Computer Vision-based Accident Detection in Traffic Surveillance [6] object detection framework implemented is Mask R-CNN (Region-based Convolutional Neural Networks), which automates pixel-level segmentation for each object in video frames. He et al. introduced Mask R-CNN enhance Faster R-CNN by replacing RoI Pooling with the RoI Align method. This adjustment eliminates alignment inaccuracies in feature maps, boosting mask accuracy by 10–50%. Mask R-CNN not only excels in instance segmentation but also improves detection precision due to RoI Align. The output includes each frame's class IDs, detection scores, bounding boxes, and pixel masks.

D. State of Art YOLO

The Optimized YOLO (You Only Look Once) algorithm uses a neural network to detect objects in an image by dividing it into an SxS grid and generating bounding boxes. It consists of twenty-four convolutional layers and two fully connected layers, with feature space reduction achieved through alternating 1x1 convolutional layers. YOLO treats object detection as a regression problem, predicting bounding box coordinates and class probabilities directly from input images in a single evaluation. Unlike CNN, which selects regions of interest, YOLO predicts classes and bounding boxes for the entire image in one pass. The bounding boxes are defined by the center coordinates (bx, by), width (bw), height (bh), and class (C), with the probability of an object in the bounding box denoted as Pc. [4] Vision-based vehicle detection and counting system using deep learning in highway scenes developed a high-resolution vehicle object dataset tailored for surveillance camera perspectives and introduced a detection and tracking approach for highway video monitoring. By extracting the road surface area, a more efficient Region of Interest (ROI) was identified. The YOLOv3 algorithm was employed to create an end-to-end vehicle detection model using the annotated highway dataset. To improve small object detection and manage multi-scale variations, the road surface was categorized into remote and proximal areas [4][7] Mini-YOLO is a distilled version of YOLO object detection algorithm and achieves comparable performance with increased computational efficiency. [11] The YOLOX algorithm consists of four parts: the input, backbone feature extraction, neck structure for multi-scale fusion, and the prediction layer with a decoupled head (YOLO Head). YOLOX improves upon YOLOv3, YOLOv4, and YOLOv5 by incorporating MixUp data enhancement, SiLU (Sigmoid Linear Unit) activation functions, and a Decoupled Head (A structure that separates classification and regression tasks into two distinct branches). The Decoupled Head splits the classification and regression tasks into two branches to reduce parameter conflict and overfitting. YOLOX uses the SimOTA (A refined allocation strategy that associates predicted frames directly with actual frames in an anchor-free mechanism) allocation strategy for anchor-free prediction and direct frame association, improving detection speed. YOLOX offers different network models (YOLOX_S/M/L/X, YOLOX-Nano, YOLOX-Tiny), with YOLOX_S having fewer parameters.[18][20]

E. Emergency Alert Systems

Accident Detection and Emergency Alerting System for Road Safety [16] system works by detecting accidents using mobile-based sensors to measure g-force values (acceleration due to gravity) during collisions. When an accident occurs, the smartphone's sensors detect abrupt motion or shaking, and the system records changes in g-force. If the g-force exceeds a predefined threshold (4g), indicating a potential accident, the system triggers an alert. This alert is sent to the nearest control station (such as a police station) using the cellular network, ensuring quick emergency response. The smartphone's sensors also account for environmental factors like sudden stops or minor drops, helping to reduce false positives. The algorithm is optimized end-to-end for real-time object detection [3] DeepCrash is a deep learning-based IoV system designed for traffic collision detection. It integrates an IVI telematics platform, vehicle sensors, a cloud-based deep learning server and a management platform. The system detects collisions with 96% accuracy and sends alerts, notifications and GPS data to the cloud in around 7 seconds. [5] When it identifies an accident image with a confidence score above 60%, the Raspberry Pi triggers a GSM module to send alerts to the nearest medical facility and law enforcement authorities. These notifications include essential details such as timestamp, precise location and the specific image where the accident was detected for further analysis.

III. Proposed System

A. System Overview:

The proposed system is designed to provide real-time accident detection and alert mechanisms, leveraging computer vision, machine learning, and cloud computing to ensure rapid emergency response. By processing live video streams from CCTV cameras, the system can detect accidents, estimate their severity, and immediately notify emergency services. This significantly reduces the time taken to assess and respond to accidents, potentially saving lives. At the core of the system is YOLOv8, an advanced deep-learning object detection model known for its speed and accuracy in recognizing objects within video feeds. The system processes these feeds in real-time, identifying accident occurrences by detecting anomalies such as sudden vehicle collisions, pedestrian falls, or debris on the road. Once an accident is detected, the severity is estimated using machine learning algorithms, which analyze the impact based on motion patterns, object trajectories, and damage assessment. For efficient data management, MongoDB is used as the primary database, storing accident records, location details, and severity scores. The system also integrates Cloudinary, a cloud-based media management service, to host accident-related images and video snapshots. This allows authorities to access clear visual evidence of the incident, aiding in decision-making and response prioritization. To ensure instant notification, the system automatically sends real-time alerts via email and other communication channels to relevant emergency services, including ambulances, police, and fire departments. The notification includes precise location details (latitude, longitude, and address), a severity classification, and a direct link to the accident images and video evidence. Additionally, the system is scalable and can be deployed across multiple locations, making it suitable for smart city applications, highway monitoring, and urban traffic management. Future improvements may include integration with IoT sensors, predictive analytics for accident hotspots, and automated dispatch systems to further optimize emergency responses.

B. Components of the System: The system aims to address the backlog in accident reporting by automatically detecting accidents, classifying their severity, and informing the emergency services with essential details, including location, severity level, and visual evidence.

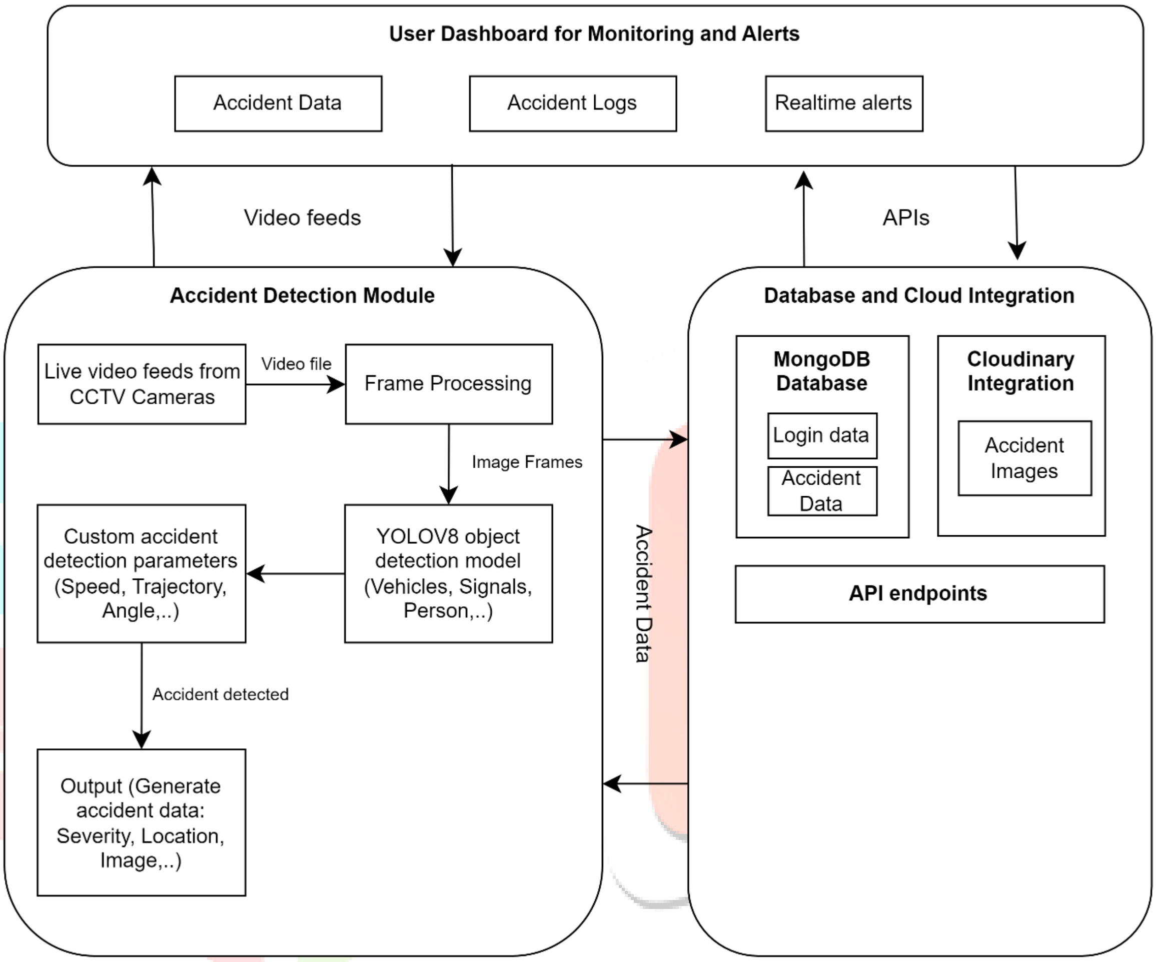

Figure 3.1: System Overview

1. User Dashboard for Monitoring and Alerts: The User Dashboard for Monitoring and Alerts is at the top of the diagram, representing the interface through which users can monitor detected accidents and receive real-time alerts. It includes three key elements: Accident Data, which provides reports containing information such as location, severity, timestamp, and images; Accident Logs, which store historical records of past accidents for analysis; and Realtime Alerts, which immediately notify users when an accident is detected. The dashboard integrates with APIs to fetch and display accident information dynamically, ensuring real-time situational awareness.

2. Accident Detection Module: The Accident Detection Module, located in the bottom-left of the diagram, is responsible for identifying and analyzing accident events using live CCTV video feeds. The process begins with Live video feeds from CCTV Cameras, which provide continuous surveillance footage. These video files undergo Frame Processing, where individual frames are extracted for analysis. The YOLOv8 Object Detection Model then processes these frames, identifying key objects such as vehicles, traffic signals, and pedestrians. Further analysis is performed using Custom Accident Detection Parameters, including speed, trajectory, and collision angles. If an accident is detected, the system generates an Output, which includes accident severity classification, location (latitude and longitude), and a captured image of the incident.

3. Database and cloud Integration: The Database and Cloud Integration section, located in the bottom-right of the diagram, manages accident-related data storage and retrieval. The MongoDB Database stores essential information such as Login Data for authentication and Accident Data, including reports with severity levels, timestamps, and location details. The system integrates Cloudinary, which handles Accident Image Storage by saving captured accident frames and generating unique URLs for easy retrieval. API Endpoints ensure seamless communication between the accident detection module, database, and dashboard, enabling real-time updates and efficient data flow.

C. Preprocessing Steps

To ensure that the input video data is properly formatted for the YOLOv8 model, several preprocessing steps are applied before the frames are fed into the detection pipeline. These steps are crucial for enhancing detection accuracy, optimizing computational efficiency, and improving model robustness under diverse conditions.

1. Frame Extraction: The first step in the preprocessing pipeline is extracting frames from the input video streams. Video feeds typically contain a continuous sequence of images captured at high frame rates. To maintain a balance between temporal resolution and computational efficiency, frames are extracted at a predefined rate of 30 frames per second (FPS). This frame rate ensures that fast-moving objects, such as vehicles, are consistently tracked while keeping the data volume manageable for real-time processing.

2. Resizing: Once the frames are extracted, they are resized to a fixed resolution of 640x640 pixels, which is the default input size for the YOLOv8 model. Standardizing the frame dimensions serves two key purposes:

- It ensures that all input images maintain a uniform size, preventing inconsistencies that could affect detection accuracy.
- It reduces computational overhead, allowing the model to process frames efficiently without unnecessary scaling operations.

The resizing operation is performed using bilinear interpolation to preserve the structural integrity of objects while minimizing distortion.

3. Normalization: To facilitate faster and more stable convergence during training and inference, pixel values in each frame are normalized to a standardized range of [0, 1]. This is achieved by scaling the pixel intensities (originally in the range of [0, 255]) using the following transformation:

$$I_{normalized} = \frac{I_{original}}{255}$$

where $I_{original}$ represents the raw pixel intensity values.

Normalization is crucial because it prevents numerical instability during model computations and improves the model's ability to learn from diverse lighting conditions.

4. Noise Reduction: To enhance detection accuracy, particularly in low-light conditions or videos with high visual noise, Gaussian blur is applied to the frames. Gaussian blur is an image-smoothing technique that reduces noise by averaging pixel intensities with their neighboring pixels. This helps eliminate unwanted artifacts such as compression noise or sensor-induced distortions that could interfere with object detection. By applying a Gaussian filter with a kernel size of 5x5, the system smooths out minor variations while retaining the structural details of vehicles. This ensures that YOLOv8 can effectively distinguish objects even under challenging environmental conditions, such as nighttime traffic or foggy weather.

D. Data Augmentation

To enhance the model's robustness and generalization, various data augmentation techniques are applied during training. These augmentations introduce controlled variations in the training data, enabling the model to learn invariant features and adapt to diverse real-world conditions. By simulating different perspectives, lighting conditions, and camera misalignments, the model becomes more resilient to changes in input data, ensuring reliable accident detection across various scenarios.

1. Random Cropping: Random cropping is used to simulate variations in camera placement and field of view. By randomly selecting and cropping a portion of the input frame, the model learns to detect vehicles even when they are partially visible or appear at different locations within the frame. This technique is particularly useful for handling videos from surveillance cameras positioned at different angles or distances.

2. Horizontal Flipping: To ensure the model accounts for directional variations, horizontal flipping is applied to a subset of training images. Since vehicles in real-world traffic can move in both left-to-right and right-to-left directions, flipping the frames horizontally ensures that the model does not develop a directional bias. This augmentation is particularly useful for improving detection accuracy in multi-lane traffic scenarios.

3. Brightness and Contrast Adjustment: Lighting conditions can vary significantly depending on factors such as time of day, weather, and artificial illumination. To make the model robust to these variations, random adjustments in brightness and contrast are applied to the training frames. By simulating conditions such as bright daylight, dim evening lights, and shadows cast by buildings or other vehicles, the model learns to detect objects reliably under different illumination levels.

4. Rotation: Slight camera misalignments can occur due to vibrations, mounting angles, or vehicle motion. To account for these variations, frames are randomly rotated by up to 10 degrees during training. This ensures that the model remains effective even when video feeds are slightly tilted, improving its ability to track objects and detect collisions in real-world settings.

E. Model Evaluation Metrics

To assess the effectiveness of the accident detection system, several evaluation metrics are used to measure both detection accuracy and real-time performance. These metrics help determine how well the model identifies accidents while maintaining computational efficiency for real-world deployment.

1. Mean Average Precision (mAP): Mean Average Precision (mAP) is a key metric for evaluating object detection models. It measures detection accuracy across different Intersection-over-Union (IoU) thresholds, typically ranging from 0.50 to 0.95. A higher mAP score indicates better localization and classification performance, ensuring that detected objects (vehicles) closely match the ground truth annotations.

2. Precision and Recall: Precision and recall are critical for assessing the model’s ability to correctly identify accidents while minimizing false detections:

- Precision: Measures the proportion of correctly detected accidents out of all reported accidents. High precision ensures that false positives (incorrectly flagged accidents) are minimized.

- Recall: Measures the proportion of actual accidents that are successfully detected. A high recall value ensures that the system captures most real accident events.

3. F1-Score: The F1-score is the harmonic mean of precision and recall, providing a balanced measure of the model’s accuracy:

$$F1 - 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

A high F1-score indicates that the model maintains a strong balance between precision and recall, ensuring reliable accident detection without excessive false positives or missed incidents.

4. Frames Per Second (FPS): Real-time processing capability is essential for accident detection systems, especially in live traffic monitoring applications. The system's Frames Per Second (FPS) metric evaluates how many video frames can be processed per second. A higher FPS value ensures that accident detection occurs in real-time without significant delays. The proposed system achieves 45 FPS, making it suitable for real-time deployment.

5. Processing Latency: Processing latency measures the time taken to analyze a single frame and generate an accident detection output. Lower latency ensures that the system can provide rapid alerts in the event of a collision. The system's latency is optimized to ensure minimal delays, making it efficient for real-world accident prevention applications.

Compared to other state-of-the-art models like YOLOv5, Faster R-CNN, and SSD, our YOLOv8-based system achieves higher accuracy and faster processing speed while maintaining real-time capabilities. YOLOv5 is efficient but lacks the latest architectural improvements. Faster R-CNN, while highly accurate, is computationally expensive and unsuitable for real-time processing. SSD offers good speed but falls behind in precision and recall.

Evaluation Metrics

Metric	YOLOv8 (Proposed)	YOLOv5	Faster R-CNN	SSD
mAP (%)	92.5	88.3	85.7	82.4
Precision (%)	94.2	90.1	87.5	84.3
Recall (%)	91.8	89.5	86.2	83.7
F1-Score (%)	92.9	89.8	86.8	84
FPS	45	35	20	30

Our model delivers top-tier detection accuracy, achieving a 92.5% mAP and 92.9% F1-Score, ensuring reliable accident detection. It also processes 45 FPS, making it the fastest among the models we compared—crucial for real-time applications like traffic monitoring and emergency response. What sets our system apart is its ability to minimize false positives and false negatives, ensuring that alerts are triggered only when necessary. Many existing models struggle with this balance—Faster R-CNN is too slow for real-time use, while SSD trades accuracy for speed. Our approach optimizes both, maintaining high performance without compromising efficiency. Beyond just object detection, our system integrates trajectory and motion analysis, allowing it to detect accidents in complex scenarios, including high-speed collisions, multi-vehicle crashes, and pedestrian incidents. A built-in severity classification mechanism further prioritizes critical cases, ensuring emergency responders can act swiftly where it matters most. By leveraging YOLOv8 and CNN-based processing, our system is not only fast and accurate but also scalable for different environments—from busy city streets to highways. This combination of deep learning, real-time detection, and automated emergency response sets a new benchmark for accident detection technology.

F. The Process of Accident Detection

The accident detection system is designed to identify collisions in real-time using computer vision and machine learning techniques. It leverages YOLOv8 object detection for vehicle tracking, trajectory analysis, and mathematical modeling to detect anomalies that indicate accidents. Once a collision is detected, the system estimates its severity based on impact parameters and provides real-time alerts to emergency response teams. The process begins with continuous vehicle tracking using the YOLOv8 object detection model. Each detected vehicle is represented by a bounding box, capturing its position and movement over time. The system monitors key parameters, including speed, acceleration, and trajectory. When two vehicles overlap in successive frames, the system examines acceleration anomalies and changes in trajectory to determine if a collision has occurred.To quantify the severity of the impact, the system calculates the angle of collision by analyzing the intersection of vehicle trajectories. If the angle falls within a predefined range, further checks are performed to confirm the accident. These checks include evaluating acceleration anomalies, trajectory deviations, and changes in angular motion. If the computed values exceed a predefined threshold, the system classifies the event as an accident and proceeds with severity estimation.

Severity Estimation: Once an accident is detected, the system evaluates its severity using multiple factors. The impact intensity is determined by analyzing the relative speed of vehicles before and after the collision. A significant drop in speed, combined with sudden deceleration, indicates a severe impact. The extent of vehicle damage is estimated by examining the dimensions of the bounding boxes post-collision. A substantial reduction in vehicle dimensions suggests major structural deformation, which correlates with accident severity. Additionally, the angle of deviation post-collision is considered—higher angles typically indicate more severe crashes. Based on these parameters, severity is classified into four categories: Low, Moderate, High, and Extreme. The classification is expressed as a percentage, allowing emergency responders to prioritize incidents effectively.

Data Management and Storage: To support analysis and reporting, the system maintains a database containing detailed records of detected accidents. Each accident is assigned a unique accident ID, along with location details such as address, latitude, and longitude. The severity classification and timestamp of the event are stored to facilitate historical analysis. The system integrates with Cloudinary, a cloud-based image storage service, to store frames of accidents detected. When an accident is recorded, the corresponding image is uploaded to Cloudinary, generating a unique URL. This URL is stored in the database, enabling quick access to visual evidence of the incident.

Notification System for Emergency Response: A critical feature of the system is its automated notification module, which ensures that emergency services are promptly informed when an accident occurs. Upon detecting an accident, the system sends an email alert to pre-configured recipients, including local emergency response teams. The email contains key details such as the accident's location (address, latitude, longitude), severity percentage, and classification. Additionally, it includes a direct link to the accident image hosted on Cloudinary, providing responders with visual context before arriving at the scene.

User Dashboard for Monitoring and Analysis: The system includes a user-friendly dashboard that allows real-time monitoring of detected accidents. The dashboard provides interactive data visualization, including a map displaying accident locations with severity-based color coding—red for extreme accidents, yellow for moderate incidents, and so on. A data table is incorporated into the dashboard, listing accident details such as timestamps, severity levels, and locations. The system ensures real-time updates, dynamically refreshing the dashboard as new accidents are detected. Additionally, real-time alerts provide immediate notifications, enabling swift decision-making and response coordination.

I. Workflow of the System

The accident detection logic uses object detection, trajectory tracking, and mathematical calculations to detect collisions and estimate their severity in real-time.

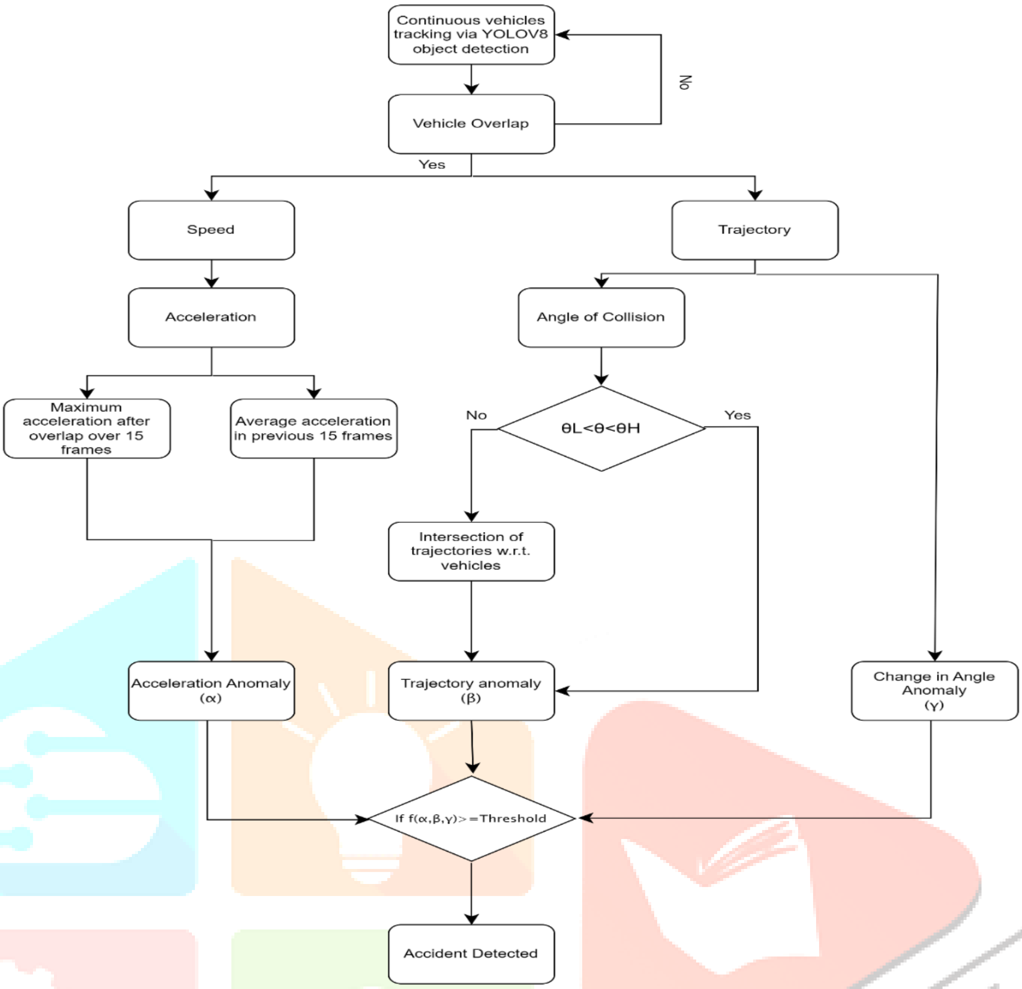

Figure 3.2: Workflow diagram describing the process of accident detection.

II. Key Components

1. Object Detection

The first step, object detection, relies on YOLOv8 to identify vehicles within each frame. Every detected vehicle is enclosed within a bounding box, defined by its top-left and bottom-right corner coordinates. The system also determines the bounding box's center, width, and height, which plays a crucial role in tracking and further analysis

Each detected object is represented by a bounding box with:

Top-left corner coordinates: (x1,y1)

Bottom-right corner coordinates: (x2,y2)

Center of the bounding box:

$$C_x = \frac{x_1 + x_2}{2}, \quad C_y = \frac{y_1 + y_2}{2}$$

Width and Height of the bounding box:

$$w = x_2 - x_1, \quad h = y_2 - y_1$$

2. Multi-Object Tracking

Once objects are detected, the system transitions to multi-object tracking, which ensures that vehicles are continuously monitored across successive frames. Using a tracking algorithm such as SORT, each vehicle is assigned a persistent ID, allowing the system to follow its movement accurately. The recorded trajectory consists of sequential position coordinates, which enable the system to analyze motion trends and identify any deviations that could indicate an accident. The velocity of each vehicle is calculated using its displacement over time, helping to determine the speed variations before and after a collision.

The system uses a tracker (e.g., SORT) to follow vehicle movements across frames.

Each vehicle is assigned a unique ID, and its trajectory is recorded as:

$$T = \{(C_{x1}, C_{y1}), (C_{x2}, C_{y2}), \ldots, (C_{xn}, C_{yn})\}$$

The velocity of a vehicle is calculated as:

$$v_x = \frac{C_{x2} - C_{x1}}{\Delta t}, \quad v_y = \frac{C_{y2} - C_{y1}}{\Delta t}$$

where Δt is the time difference between frames.

3. Collision Detection

The system identifies collisions using the following metrics:

a) Trajectory Intersection
Detecting a collision requires a multi-faceted approach that evaluates multiple parameters. One of the primary indicators is trajectory intersection, which determines whether two vehicles have physically overlapped. If the bounding boxes of two vehicles intersect within a given timeframe, the system considers it a potential collision event.

Two vehicle trajectories TA and TB intersect if their bounding boxes overlap.

Intersection condition: Intersection(A,B)=TRUE if Bounding Boxes Overlap, ELSE FALSE.

b) Speed Difference
The speed difference between two vehicles colliding is another crucial metric. The relative velocity between two moving vehicles is analyzed, and if the difference surpasses a certain threshold, a collision is suspected. A sudden and significant drop in speed typically indicates an impact.

The relative speed between two vehicles A and B:

$$\Delta v = \sqrt{(v_{xA} - v_{xB})^2 + (v_{yA} - v_{yB})^2}$$

A collision is flagged if:

$$\Delta v > v_{\text{threshold}}$$

c) Collision Angle
The collision angle is used to gauge the severity of the impact. The system calculates the angle at which two vehicles collide by examining their trajectories before and after the impact. High-impact angles tend to indicate more severe accidents, as they result in greater structural damage and a higher likelihood of injury.

Angle of collision between two vehicles is given by:

$$\theta = \cos^{-1}\left(\frac{\vec{v}_A \cdot \vec{v}_B}{|\vec{v}_A||\vec{v}_B|}\right)$$

A collision is flagged if:

$$\theta_{\min} \leq \theta \leq \theta_{\max}$$

d) Bounding Box Overlap (IoU)
the bounding box overlaps, measured using the Intersection-over-Union (IoU) metric. This metric quantifies the extent of overlap between two vehicle bounding boxes before and after a suspected collision. A high IoU value suggests a direct collision, whereas a lower value may indicate a near-miss or minor contact. If the IoU surpasses a predefined threshold, the system confirms the accident occurrence.

Intersection-over-Union (IoU) measures the overlap between two bounding boxes:

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

A collision is flagged if:

$$\text{IoU} > \text{IoU}_{\text{threshold}}$$

4. Severity Estimation
Once a collision has been detected, the system shifts focus to estimating its severity. The severity analysis considers multiple factors,

The severity of an accident is determined by analysing collision parameters. The key factors are:

a) Impact Intensity
impact intensity, which is proportional to the relative speed at the time of collision. A high-speed impact generates greater force, often leading to more significant damage. To approximate this force, the system incorporates an estimated mass for each vehicle, inferred from its bounding box dimensions.

The force of impact F is proportional to the relative speed:

$$F = m \cdot \Delta v$$

where mmm is the mass of the vehicle (approximately using bounding box size).

b) Damage Area
the damage area, determined by analyzing the bounding box dimensions before and after the accident. A noticeable reduction in bounding box size indicates vehicle deformation, which directly correlates with accident severity. The extent of this reduction is a strong predictor of the overall impact force and potential harm.

Change in the bounding box area before and after collision:

$$A_d = |w_{\text{before}} \cdot h_{\text{before}} - w_{\text{after}} \cdot h_{\text{after}}|$$

c) Collision Angle
The collision angle further refined severity classification, as higher angles typically result in more dangerous accidents. A head-on or T-bone collision, for instance, is far more severe than a minor sideswipe. The system accounts for this by assigning higher weights to sharper collision angles in its calculations.

Higher collision angles (θ) indicate more severe impacts.

The overall severity is calculated as:

$$\text{Severity} = \alpha \cdot F + \beta \cdot A_d + \gamma \cdot \theta$$

where α,β,γ are weights assigned based on their relative importance.

Severity Levels:

To provide a quantifiable severity measure, the system combines these factors into a mathematical model. Weighted coefficients are assigned to impact intensity, damage area, and collision angle based on their relative significance. The final severity score is then computed as a percentage, allowing the system to categorize accidents into four levels:

- Low severity: where the impact is minimal, and the structural damage is negligible.
- Moderate severity: indicating a noticeable impact that may require further assessment.
- High severity: suggesting significant damage and a likely need for emergency intervention.
- Extreme severity: representing catastrophic collisions with potentially fatal consequences.

Low: Severity < 25%

Moderate: 25% ≤ Severity < 50%

High: 50% ≤ Severity < 75%

Extreme: Severity ≥ 75%

This structured approach ensures that emergency responders receive precise, real-time information about the severity of an accident, allowing them to prioritize high-risk incidents and allocate resources effectively. By integrating object detection, tracking, mathematical analysis, and severity classification, the accident detection system provides a reliable and efficient solution for enhancing road safety and reducing emergency response times.

IV. Results and Outputs

To tackle the challenges of real-time accident detection, we propose a CNN-based accident detection and emergency response system powered by YOLOv8. Traditional methods, such as manual reporting and sensor-based detection, often lead to delays and inefficiencies. In contrast, our approach leverages deep learning and computer vision to analyze live surveillance footage and detect accidents instantly. The system is designed to identify vehicles and pedestrians in real-time, ensuring rapid accident detection. By tracking vehicle trajectories and motion patterns, it can recognize sudden changes indicative of a collision. Additionally, it evaluates the severity of accidents based on factors like impact force, bounding box overlap, and speed variations, categorizing them into low, moderate, high, or extreme levels. Once an accident is detected, the system automatically notifies emergency services via email, providing precise location details and accident images, enabling a swift and effective response.

Compared to state-of-the-art models like YOLOv5, Faster R-CNN, and SSD, our YOLOv8-based system achieves superior performance in both accuracy and speed. While YOLOv5 is efficient, it lacks the latest architectural advancements. Faster R-CNN, though highly accurate, is computationally expensive and unsuitable for real-time applications. SSD offers good speed but sacrifices precision and recall. Our system achieves a 92.5% mAP and a 92.9% F1-Score, ensuring reliable accident detection with minimal false positives and false negatives. With a processing speed of 45 FPS, it outperforms YOLOv5 (35 FPS), Faster R-CNN (20 FPS), and SSD (30 FPS), making it ideal for real-time traffic monitoring and emergency response.

Final results and outputs of the projects are presented which gives a clear view of the system and its working from the aspect of a user. The main components of the system’s features are explained below.

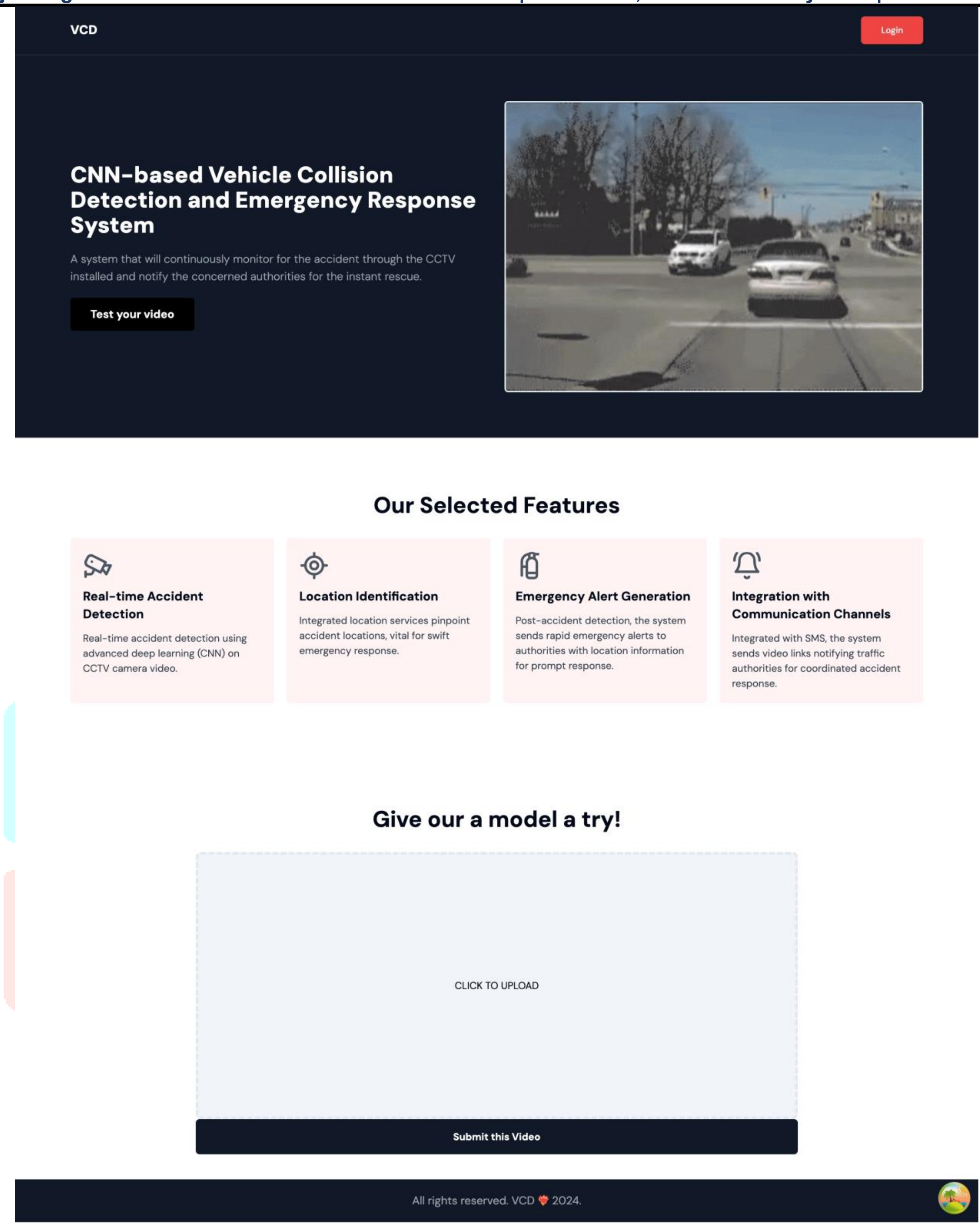

Figure 4.1 Home Page

The homepage of the system introduces users to its features and functionalities, including an interactive section for testing the accident detection model by uploading videos. This feature demonstrates the system's capability to process real-world inputs and deliver results efficiently. Key features include an overview of the system, a video upload section for model testing, and a user-friendly interface with clear instructions.

ID	Date & Time	Address	Severety(%)	Severty	View Details
65e36...	Sat, 02 Mar 2024 23:23:47 GMT	Manipal Teaching Hospital, 40DR014, Pokhara-11, Pokhara, कास्की, गण्डकी प्रदेश, 88700, नेपाल	65	Moderate	View ↗
65e4a...	Fri, 02 Feb 2024 23:23:47 GMT	Manipal Teaching Hospital, 40DR014, Pokhara-11, Pokhara, कास्की, गण्डकी प्रदेश, 88700, नेपाल	65	Moderate	View ↗
65e3f...	Fri, 02 Feb 2024 10:06:10 GMT	Manipal Teaching Hospital, 40DR014, Pokhara-11, Pokhara, कास्की, गण्डकी प्रदेश, 88700, नेपाल	65	Moderate	View ↗
65e2f...	Tue, 02 Jan 2024 16:14:16 GMT	Manipal Teaching Hospital, 40DR014, Pokhara-11, Pokhara, कास्की, गण्डकी प्रदेश, 88700, नेपाल	57.99999999999999	Moderate	View ↗
65e2f...	Tue, 02 Jan 2024 16:14:01 GMT	Manipal Teaching Hospital, 40DR014, Pokhara-11, Pokhara, कास्की, गण्डकी प्रदेश, 88700, नेपाल	65	Moderate	View ↗
65e2f...	Mon, 01 Jan 2024 16:14:27 GMT	Manipal Teaching Hospital, 40DR014, Pokhara-11, Pokhara, कास्की, गण्डकी प्रदेश, 88700, नेपाल	57.99999999999999	Moderate	View ↗
65e4a...	Sat, 02 Dec 2023 10:06:25 GMT	Manipal Teaching Hospital, 40DR014, Pokhara-11, Pokhara, कास्की, गण्डकी प्रदेश, 88700, नेपाल	57.99999999999999	Moderate	View ↗
65e4a...	Thu, 02 Nov 2023 16:14:16 GMT	Manipal Teaching Hospital, 40DR014, Pokhara-11, Pokhara, कास्की, गण्डकी प्रदेश, 88700, नेपाल	57.99999999999999	Moderate	View ↗
65e48...	Fri, 02 Dec 2022 20:31:39 GMT	Manipal Teaching Hospital, 40DR014, Pokhara-11, Pokhara, कास्की, गण्डकी प्रदेश, 88700, नेपाल	75	Moderate	View ↗
65e3f...	Fri, 02 Dec 2022 10:06:25 GMT	Manipal Teaching Hospital, 40DR014, Pokhara-11, Pokhara, कास्की, गण्डकी प्रदेश, 88700, नेपाल	57.99999999999999	Moderate	View ↗

Figure 4.2 User Dashboard

The system provides a secure dashboard for logged-in users, allowing them to view, track, and analyze accident records in real-time. This centralized repository enhances monitoring, response coordination, and statistical analysis for improving road safety and emergency response.

The system records and organizes accident data with the following attributes:

Accident ID: A unique identifier assigned to each reported accident, ensuring distinct tracking and retrieval of records.

Date & Time: The precise timestamp when the accident was recorded, aiding in chronological analysis and emergency response timing.

Address: The exact location of the accident, including detailed address information such as hospital names, streets, and city data for accurate mapping.

Severity (%): The severity of the accident, quantified as a percentage, provides an objective measure of the impact and intensity of the incident.

Severity Category: A classification of the accident's severity into categories such as Moderate, Severe, or Critical, helping in prioritizing emergency responses.

View Details: A clickable option that leads to a detailed view of the incident, revealing comprehensive insights, including possible causes, vehicle involvement, and additional metadata.

The dashboard provides a clear and structured way to track accident data, displaying key details for each incident. Each entry includes a unique Accident ID, the date and time of occurrence, and the location where the accident took place. The severity of the accident is recorded both as a percentage and categorized into levels like Moderate, Severe, or Critical. A "View Details" button allows users to dig deeper into each case, revealing possible causes and the vehicles involved. Users can easily navigate through accident records, accessing both recent and past incidents. The system also allows customization, letting users adjust the number of entries displayed per page. Future updates could introduce sorting and filtering features to make searching for specific cases even more efficient. The dashboard is designed with usability and security in mind. Its intuitive layout makes it easy to analyze data, while the logout button in the top-right corner ensures that only authorized users have access. Future improvements could include advanced search filters, allowing users to find accidents based on severity, location, or time range. Adding graphs, charts, or heatmaps could also help visualize accident trends over time. Another potential feature is an automated alert system, which could notify emergency services in real-time based on accident severity. By providing a real-time, structured, and secure way to monitor accidents, this system helps improve emergency response times, supports traffic analysis, and ultimately contributes to safer roads.

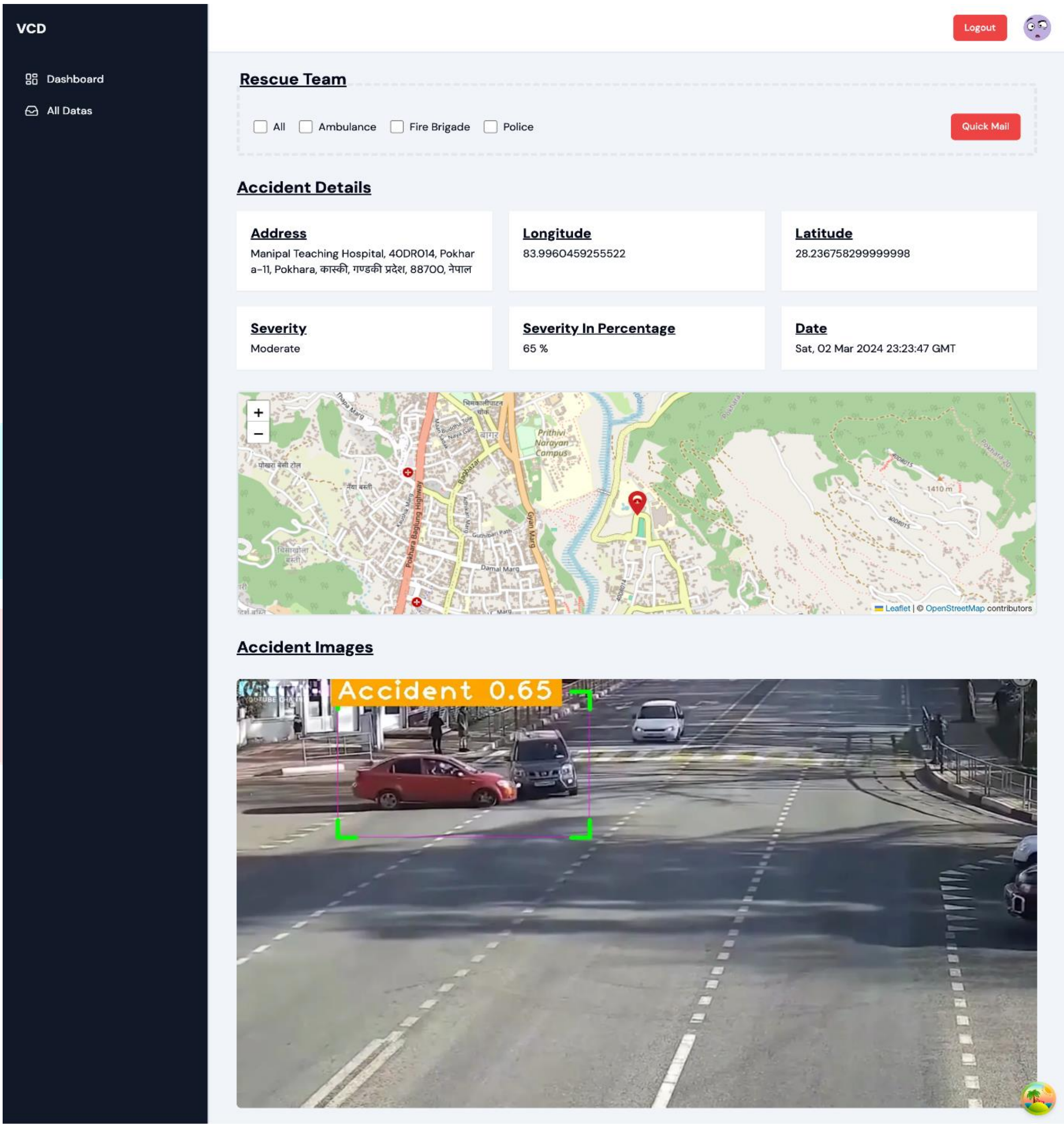

Figure 4.3 Accident Details

A key feature of the dashboard is visual evidence, including images and video footage captured from surveillance cameras or AI-based detection systems. The system assigns a severity score, helping assess the extent of damage and prioritize response efforts. To facilitate a quick response, the dashboard includes options to alert emergency teams—ambulance, fire brigade, or police—based on severity. A quick mail button ensures rapid communication with responders. The interface is user-friendly and secure, with an organized layout for easy navigation. A logout button ensures data access is restricted to authorized personnel. Future enhancements could include advanced filtering options to search for accidents by location, severity, or time frame. Automated alerts for emergency services and graphical analytics like heatmaps and trend reports could further improve response efficiency and accident prevention. By integrating real-time data and AI-driven analysis, the dashboard enhances accident detection and emergency response, making roads safer and improving incident management.

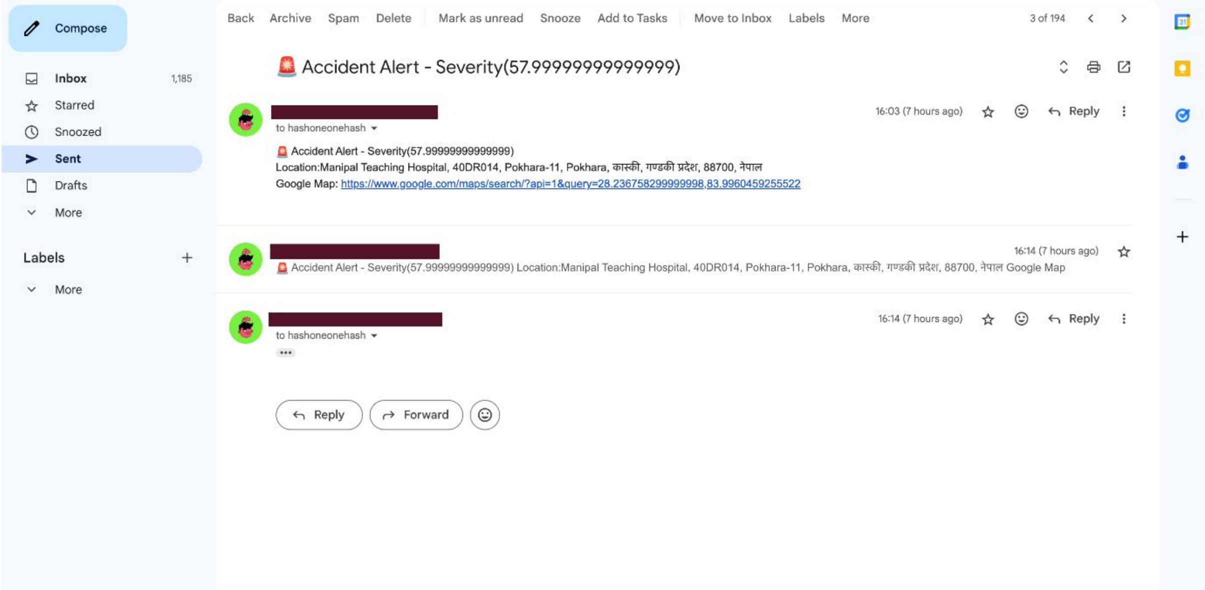

Figure 4.4 Email Alert

As soon as an accident is detected, the system automatically notifies the relevant authorities via email, ensuring a rapid response. The email contains critical details, including the exact location of the accident with its address, latitude, and longitude coordinates, allowing emergency teams to pinpoint the site accurately. The severity level is also highlighted, represented as a percentage and classified as low, moderate, or high, helping responders prioritize the situation accordingly. Additionally, a direct link to the location on a map is provided, enabling quick navigation for emergency teams. This automated alert system streamlines communication and ensures that the necessary actions are taken without delay, ultimately improving accident response times and potentially saving lives.

V. Conclusion

The proposed Accident Detection and Emergency Response System demonstrates the potential of harnessing advanced technologies such as machine learning, computer vision and cloud services to address critical real-world challenges. The system is designed to detect accidents in real-time, analyze the severity of the incident and ensure rapid notification to the relevant authorities. This research highlights the development of a comprehensive, user-friendly solution that integrates multiple functionalities, including accident detection, data logging, real-time alerts and automatic e-mail notifications.

The system offers several notable contributions: real-time accident detection using a robust Convolutional Neural Network (CNN) model ensures accurate identification of incidents from video footage; automated reporting incorporates Cloudinary to store accident images and email automation to quickly communicate detailed accident reports, including severity and location, to authorities; a user-friendly dashboard provides centralized monitoring of accident logs, detailed information and historical data for analysis; and scalability and adaptability, leveraging cloud-based infrastructure and scalable architecture, make the system suitable for wider implementation in smart cities and integration with IoT devices such as traffic cameras and sensors.

Key Achievements

The research achieved its main objectives by developing a system that is both technically sound and practically feasible. The ability to predict the severity of accidents and inform the authorities with detailed information in real-time enables emergency response teams to act quickly, potentially saving lives and reducing damage.

Additionally, the system promotes transparency by maintaining detailed records accessible to users and administrators, enabling post-accident analysis and preventive measures.

Impact

This research makes a significant contribution to the field of intelligent transport systems and emergency management. By reducing the time between accident and intervention, the system can play a key role in reducing fatalities and improving road safety.

Limitations and Future Work

Although the system works well under controlled conditions, certain limitations exist: dependence on video quality, where the accuracy of the accident detection model can decrease with low-resolution or poorly lit video; GPS accuracy, where minor inaccuracies can occur in urban environments due to signal interference; and scalability challenges, as real-time processing of data from a large number of cameras, requires further optimization of computing resources. Future enhancements can include integration with IoT devices like vehicular sensors and traffic cameras, the use of advanced deep learning models for higher accuracy in accident detection and severity prediction, implementation of a multilingual user interface for a global audience and adding predictive analytics to identify accident-prone areas based on historical data.

Competing interests

The authors declare no competing interests

References

[1] Zehang Sun, Member, IEEE, George Bebis, Member, IEEE, and Ronald Miller, "On-Road Vehicle Detection: A Review," IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 28, NO. 5, MAY 2006.

[2] Raad Ahmed Hadi1, Ghazali Sulong, Loay Edwar George, "VEHICLE DETECTION AND TRACKING TECHNIQUES: A CONCISE REVIEW," Signal & Image Processing: An International Journal (SIPIJ) Vol.5, No.1, February 2014.

[3] Deeksha Gour and Amit Kanskar., "Optimized-YOLO: Algorithm for CPU to Detect Road Traffic Accident and Alert System," International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 Vol. 8 Issue 09, September-2019.

[4] Chang, W.-J., Chen, L.-B., & Su, K.-Y. (2019). DeepCrash: A Deep Learning-based Internet of Vehicles System for Head-on and Single-vehicle Accident Detection with Emergency Notification. IEEE Access, 1–1. doi:10.1109/access.2019.2946468.

[5] Ghosh, S., Sunny, S. J., & Roney, R. (2019). Accident Detection Using Convolutional Neural Networks. 2019 International Conference on Data Science and Communication (IconDSC). doi:10.1109/icondsc.2019.8816881.

[6] Ijjina, E. P., Chand, D., Gupta, S., & K, G. (2019). Computer Vision-based Accident Detection in Traffic Surveillance. 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT). doi:10.1109/icccnt45670.2019.8944.

[7] Huansheng Song, Haoxiang Liang, Huaiyu Li, Zhe Dai and Xu Yun "Vision-based vehicle detection and counting system using deep learning in highway scenes," Song European Transport Research Review (2019) 11:51 https://doi.org/10.1186/s12544-019-0390-4.

[8] Julian Weidinger, Sebastian Schlauderer, Sven Overhage "Determinants for the acceptance of emergency response information systems: Ethnographical insights into the digitalization of a voluntary fire department," International Journal of Disaster Risk Reduction 109 (2024) 104603.

[9] Yang, D., Wu, Y., Sun, F., Chen, J., Zhai, D., & Fu, C. (2021). Freeway accident detection and classification based on the multi-vehicle trajectory data and deep learning model. Transportation Research Part C: Emerging Technologies, 130, 103303. doi: 10.1016/j.trc.2021.103303.

[10] S. Uma and R. Eswari., "Accident prevention and safety assistance using IOT and machine learning," Journal of Reliable Intelligent Environments https://doi.org/10.1007/s40860-021-00136-3.

[11] Pillai, M. S., Chaudhary, G., Khari, M., & Crespo, R. G. (2021). Real-time image enhancement for an automatic automobile accident detection through CCTV using deep learning. Soft Computing, 25(18), 11929–11940. doi:10.1007/s00500-021-05576-w.

[12] Nikhlesh Pathik, Rajeev Kumar Gupta, Yatendra Sahu, Ashutosh Sharma, Mehedi Masud and Mohammed Baz., "AI Enabled Accident Detection and Alert System Using IoT and Deep Learning for Smart Cities," Sustainability 2022, 14(13), 7701; https://doi.org/10.3390/su14137701.

[13] Hawzhin Hozhabr Pour, Frédéric Li, Lukas Wegmeth, Christian Trense, Rafał Doniec, Marcin Grzegorzek, and Roland Wismüller., A Machine Learning Framework for Automated Accident Detection Based on Multimodal Sensors in Cars," Sensors 2022, 22, 3634. https://doi.org/10.3390/s22103634.

[14] Mallikarjun Anandhalli1, Vishwanth P. Baligar2, Pavana Baligar3, Pooja Deepsir4, Mithali Iti., "Vehicle detection and tracking for traffic management," IAES International Journal of Artificial Intelligence (IJ-AI) Vol. 10, No. 1, March 2021, pp. 66~73 ISSN: 2252-8938, DOI: 10.11591/ijai.v10.i1.pp66-73.

[15] Rajesh, G., Benny, A. R., Harikrishnan, A., Jacob Abraham, J., & John, N. P. (2020). A Deep Learning based Accident Detection System. 2020 International Conference on Communication and Signal Processing (ICCSP). doi:10.1109/iccsp48568.2020.91822.

[16] Dilipkumar, A. Borikar and Manoj B. Chandak, "Accident Detection and Emergency Alerting System for Road Safety," *Biosc.Biotech.Res.omm. pecial Issue Vol 13 o 14 (2020) Pp-256-259.

[17] Huang, T., Wang, S., & Sharma, A. (2020). Highway crash detection and risk estimation using deep learning. Accident Analysis & Prevention, 135, 105392. doi:10.1016/j.aap.2019.105392.

[18] Zhihai Liu, Wenyu Han, Hao Xu, KesongGong, Qingliang Zeng, Xieguang Zhao., "Research on vehicle detection based on improved YOLOX_S," Scientific Reports.

[19] Zhang, Y.; Sung, Y. Traffic Accident Detection Using Background Subtraction and CNN Encoder–Transformer Decoder in Video Frames. Mathematics 2023, 11,2884. https://doi.org/10.3390/math11132884.

[20] Prashan Premaratne, Inas Jawad Kadhim, Rhys Blacklidge, Mark Lee., "Comprehensive review on vehicle Detection, classification and counting on highways," Neurocomputing 556 (2023) 126627.

[21] Global status report on Road Safety 2023 WHO Team Safety and Mobility (SAM), https://iris.who.int/bitstream/handle/10665/375016/9789240086517-eng.pdf?sequence=1 Social Determinants of Health (SDH) Reference numbers ISBN: 978-92-4-008651-7.

[22] Benson Thomas M, Anuj K. Pandey, Diksha Gautam, Sandeep Gopinathan, Sajna Panolan., Economic Burden of Accidents and Injuries in India: What Does 75th Round of National Sample Survey Imply? https://pmc.ncbi.nlm.nih.gov/articles/PMC10900473/pdf/IJCM-49-181.pdf Indian J Community Med. 2024 Jan 12;49(1):181–188. doi: 10.4103/ijcm.ijcm_457_twenty-three.