# AI-Based Accident Detection Set with Limited Force and Automatic 112 Call System

[1]Mohammed Firhaz P M, [2]Muhammed Anas A, [3]P Suraj, [4]P Aswin, [5]Sajitha

*[1]Student, [2]Student, [3]Student, [4]Student, [5]Assistant Professor (CSE) Computer Science and Engineering Department, Nehru College of Engineering and Research Centre (NCERC), Thrissur, India*

---------------------------------------------------------------------***---------------------------------------------------------------------

**Abstract**
The AI-Based Accident Detection with Limited Force and Automatic 911 Call System is a mini-project designed to identify minor vehicle collisions or incidents where limited force is applied, which might otherwise go unnoticed. Using artificial intelligence (AI) and machine learning algorithms, the system analyzes data from vehicle sensors (such as accelerometers, gyroscopes, and impact sensors) to detect even low-impact accidents. Upon detection, the system automatically initiates a call to emergency services and shares the vehicle's precise GPS location, ensuring prompt assistance. This project aims to improve road safety by addressing minor accidents that could still pose risks to drivers, passengers, or pedestrians.
Example

Keywords:AI,IoT in vehicles,Intelligent Transport System, Real-Time Montoring

## 1. INTRODUCTION

Road accidents are a major global concern, leading to significant loss of life and property. The delay in emergency response further increases the severity of injuries and fatalities. To address this issue, AI-based accident detection systems are being developed to ensure faster response times and improve survival rates. This project focuses on designing an "AI-Based Accident Detection System with Limited Force and Automatic 112 Call System." The system is designed to detect accidents based on impact force and other critical parameters. Unlike conventional systems that trigger alerts at high impact forces, this model incorporates limited force detection, ensuring even minor accidents are identified. Upon detecting an accident, the system automatically initiates an emergency call to 112 (the universal emergency number in Europe and other regions), providing real-time location details to first responders. The implementation of artificial intelligence (AI) and sensor-based detection enhances accuracy, reducing false alarms while ensuring prompt action in critical situations. This project aims to contribute to road safety by minimizing response time, improving post-accident assistance, and potentially saving lives.

The primary objective of the project is to develop an AI-based accident detection system capable of accurately identifying accidents using real-time sensor data. The system will be designed to intelligently process accelerometer, gyroscope, and GPS data through machine learning and sensor fusion techniques to ensure precise accident detection. A key feature of the system is the integration of a limited force threshold mechanism, which sets a predefined impact value to distinguish between minor disturbances and severe accidents, thereby enhancing detection accuracy. To further improve real-time responsiveness, the system will be equipped with an automated emergency call functionality that immediately contacts emergency services (112) upon detecting a confirmed accident, ensuring prompt assistance to the victim. Additionally, the system aims to minimize false positives by refining detection algorithms and validating performance against real-world accident scenarios, ultimately enhancing reliability and efficiency. A user-friendly interface will also be developed, allowing users and emergency responders to access accident reports and location details effortlessly. To make the system suitable for continuous operation in vehicles or mobile devices, it will be optimized for low power consumption, ensuring minimal energy usage without compromising performance. GPS-based location tracking will be implemented to accurately determine the exact accident site and transmit this information to emergency responders for quicker intervention. Finally, rigorous testing and validation under various conditions will be conducted to assess the system's real-time effectiveness, ensuring it meets the highest standards of accuracy, reliability, and efficiency in accident detection.

## 2. LITERATURE SURVEY

A literature review compiles the existing scholarship on a given topic, drawing from the work of recognized scholars and researchers. It encompasses the latest understanding, comprising significant discoveries, theoretical insights, and methodological advancements within the subject matter. Utilizing secondary sources, literature reviews don't present fresh experimental endeavors. Rather, they allow us to cultivate and showcase proficiency in information retrieval and critical analysis.

Accident detection and emergency response systems have been an active area of research in recent years. Various approaches have been explored to enhance the efficiency and accuracy of accident detection using artificial intelligence (AI), Internet of Things (IoT), and sensor-based systems.

### 2.1 Traditional Accident Detection Systems

Earlier accident detection relied on manual reporting bystanders or victims, which often led to delayed response times. Later, On-Board Diagnostics (OBD) and GPS-based systems were introduced, which could send location data upon impact. However, these systems primarily responded to high-impact collisions, making them ineffective for lower-force accidents that still require medical attention.

### 2.2 AI-Based Accident Detection

Recent studies have incorporated machine learning and AI models for accident detection. S. Ghose proposed a convolutional neural networks (CNN)-based accident detection system. This model has two classes, named accident and non-accident. These models improve detection accuracy but may require high computational power.

.

### 2.3 Sensor-Based Detection

Many researchers have explored the use of accelerometers, gyroscopes, and vibration sensors for accident detection. Isha Khot developed an Android app that uses only the accelerometer sensor and GPS module of a smartphone for accident detection and notification. One of the main features of this app is that it also notifies the nearby user about the accident location and the emergency services

### 2.4 IoT and Automated Emergency Call Systems

With the advent of IoT-enabled smart vehicles, real-time data transmission has improved accident reporting. Systems like eCall in Europe automatically dial emergency services upon detecting a crash. However, research has identified a gap in detecting lower-force accidents, which is where this project aims to contribute..

### 2.5 Proposed System Contribution

Our project builds upon existing research by incorporating:

Limited force detection to address low-impact accidents AI-based classification to reduce false positives.

Automatic 112 emergency call system for immediate response. By integrating these feastures,the system aims to improve accident detection efficiency and enhance the system

## 3. SYSTEM ANALYSIS AND DESIGN

System analysis provides more emphasis on understanding the details of an existing system or a proposed one and then deciding whether the proposed system is desirable or not and whether the existing system needs improvement. Thus, system analysis is the process of investigating a system, identifying problems, and using the information to recommend improvements to the system. The chapter deals with the detailed description of each module, the problems faced by the current System and how these problems are rectified by proposed system.

The new system represents a significant improvement over its predecessor, effectively tackling many of the limitations encountered with the old system and enhancing overall efficiency. By automating tasks that were previously done manually, such as calculating monthly income, the new system not only improves accuracy but also saves valuable time. Moreover, the development of modules has been streamlined, resulting in a more visually appealing interface that enhances the user experience. In terms of functionality, the system delivers reliable and comprehensive information, effectively meeting the needs of its users.

### 3.1 Module Description

#### 3.1.1 AI-based Accident Detection Module

The accident detection system is designed to identify accidents accurately using AI models trained on sensor and camera data. It leverages advanced machine learning techniques, including Convolutional Neural Networks (CNN) for image analysis and Recurrent Neural Networks (RNN) for processing sensor data, enabling it to recognize complex patterns associated with accidents. To enhance detection accuracy, the system employs data fusion techniques that integrate readings from multiple sensors, ensuring a more comprehensive analysis. By extracting motion patterns from accelerometer and gyroscope data, the system can detect sudden changes in movement and orientation that indicate a potential accident. Additionally, it analyzes the force impact to determine the likelihood of an accident occurring, distinguishing between minor disturbances and serious incidents. To minimize false alarms, the system incorporates AI-based pattern recognition, which helps in filtering out non-accident events that may otherwise trigger unnecessary alerts. This intelligent processing not only improves the reliability of accident detection but also ensures timely and accurate notifications, making the system a valuable tool for real-time accident monitoring and response.

#### 3.1.2 Limited Force Consideration Module

Limited Force Consideration Module is designed to ensure that only accidents surpassing a predefined force threshold are reported, thereby minimizing false positives and enhancing the reliability of the accident detection system. It achieves this by utilizing force measurement algorithms that accurately calculate impact strength, enabling the system to distinguish between minor disturbances and significant collisions. To further refine its accuracy, the module incorporates adaptive thresholding, which is derived from real-world accident datasets and dynamically adjusts based on various environmental factors such as road conditions, vehicle type, and driving speed. During processing, the system continuously evaluates detected force values against established thresholds, allowing it to ignore events that fall below the limit and thus do not require emergency intervention. By dynamically modifying these thresholds based on situational factors, the module ensures optimal sensitivity and responsiveness in different accident scenarios. This intelligent approach not only prevents unnecessary alerts but also guarantees that severe accidents are promptly and accurately identified, making the system more efficient and reliable in real-world applications

**3.1.3 Data Acquisition Module**

The Data Acquisition Module is responsible for collecting real-time data from various sensors and cameras to ensure accurate accident detection. It integrates multiple components, including accelerometers and gyroscopes, which capture motion-related data, allowing the system to monitor changes in speed, direction, and orientation. Additionally, pressure sensors are utilized to measure the force of impact, providing critical information for assessing the severity of an accident. Cameras further enhance detection accuracy by capturing visual evidence, enabling video-based analysis to complement sensor readings. During processing, the module converts raw sensor signals into structured data, ensuring that the information is organized and ready for further analysis by the AI-driven accident detection system. This seamless integration of multiple data sources enhances the reliability of accident detection, enabling the system to make precise and informed decisions while reducing false positives and improving overall efficiency.

**3.1.4 4 Preprocessing Module**

The Preprocessing Module plays a crucial role in cleaning and normalizing sensor data to eliminate noise, ensuring that the accident detection system operates with high accuracy and reliability. It employs advanced data filtering algorithms such as the Kalman filter and moving average techniques to smooth out irregularities and enhance data quality. Additionally, it utilizes feature extraction techniques to prepare the refined data for AI-driven analysis, allowing machine learning models to efficiently identify accident patterns. Closely integrated with preprocessing, the Threshold Analysis Module is responsible for determining whether a detected event surpasses the predefined force limit. This module incorporates dynamic threshold calculation based on impact force, ensuring that only significant accidents are flagged while filtering out minor disturbances to minimize false positives. It also incorporates adaptive learning mechanisms that continuously refine threshold values over time, improving detection accuracy by accounting for varying environmental and contextual factors. By seamlessly working together, the Preprocessing and Threshold Analysis Modules enhance the system's ability to distinguish between minor incidents and severe accidents, ultimately contributing to a more robust and intelligent accident detection framework.

**3.1.5 Alert System Module**

The Alert System Module is designed to promptly notify emergency contacts when a confirmed accident occurs, ensuring a swift response to critical situations. It utilizes multiple communication channels, including SMS, email, and mobile app notifications, to deliver alerts efficiently. Additionally, it integrates with emergency services, enabling automatic reporting to first responders for immediate assistance. During processing, the module compiles accident details, such as the time, severity, and exact location of the incident, before transmitting this information to designated contacts. By providing realtime notifications with precise accident data, the system enhances emergency response efficiency, potentially reducing the time taken for medical aid and intervention. This proactive approach ensures that help reaches the affected individuals as quickly as possible, making the accident detection system highly effective in real-world applications.

**3.1.6 Data Logging Module**

The Data Logging Module is responsible for storing accident data to facilitate future analysis and continuous system improvements. It utilizes a cloud-based database, such as Firebase or PostgreSQL, to securely store and manage accident-related information, ensuring accessibility and scalability. Additionally, a data visualization dashboard is integrated into the system, allowing administrators to monitor, analyse, and review accident trends and system performance over time. During processing, the module logs crucial details, including timestamps, impact force, and sensor data, which can be used to refine detection algorithms and enhance overall system accuracy. By maintaining a structured record of accident events, this module supports long-term data analysis, helping researchers and developers optimize the system's reliability while also enabling stakeholders to assess accident patterns and improve road safety measures.

**3.2 System Analysis**

**3.2.1 Problem Statement**

Current accident detection systems often either over-detect minor incidents or fail to detect significant accidents due to limitations in data analysis. This system aims to address theseissues by integrating force measurement and AI-driven analysis.

**3.2.2 Functional Requirements**

1. Accident Detection: Identify accidents based on sensor data.
2. Force Measurement: Differentiate between major and minor incidents.
3. AI Analysis: Use machine learning to analyze patterns and validate accidents.
4. Alert System: Notify relevant personnel via SMS, email, or mobile apps.
5. Data Logging: Store event data for future analysis.

**3.2.2 Non-Functional Requirements**

The system is designed with several key non-functional requirements to ensure its effectiveness and reliability in real-world applications. Real-time processing is a critical feature, enabling the system to detect accidents quickly and trigger an immediate response. High accuracy is another essential aspect, as the system is optimized to minimize false positives and negatives through advanced AI algorithms and refined detection techniques, ensuring that only genuine accidents are identified. Scalability is a core consideration, allowing the system to adapt seamlessly to different environments, whether in urban settings, highways, or remote locations, making it suitable for various vehicle types and road conditions. Reliability is also a fundamental requirement, ensuring consistent performance across diverse conditions, including varying weather, road surfaces, and driving patterns. By integrating these nonfunctional attributes, the system guarantees efficiency, robustness, and dependability, making it a highly effective solution for accident detection and emergency response.

### 3.3 System Design

#### 3.3.1 System Architecture

The system architecture is structured into three main layers: the input layer, processing layer, and output layer, each playing a vital role in ensuring accurate accident detection and response. The input layer consists of various sensors, including accelerometers, gyroscopes, and pressure sensors, which continuously monitor motion and impact data. Additionally, cameras such as surveillance systems or invehicle dashcams capture visual evidence, enhancing the system's ability to detect accidents effectively. The processing layer is responsible for analyzing the collected data, beginning with the preprocessing module, which filters and normalizes sensor readings to remove noise and enhance accuracy. The AI model then utilizes advanced machine learning algorithms, including Convolutional Neural Networks (CNN) for image processing and Recurrent Neural Networks (RNN) for sensor data analysis, to detect potential accidents. A critical component of this layer is the threshold analysis module, which determines whether the detected event surpasses the predefined force limit, ensuring that only significant incidents are flagged. Once an accident is confirmed, the output layer takes action by notifying emergency contacts through the alert system, providing real-time details via SMS, email, or mobile apps. Simultaneously, accident data is stored in a secure database for future analysis and system improvements. Additionally, a dashboard is available for administrators to visualize incidents, track trends, and optimize the system's performance over time. This multi-layered architecture ensures a seamless and efficient process from data acquisition to accident detection, notification, and logging, making the system highly reliable for real-world applications.

#### 3.3.2 Data Flow Diagram (DFD)

1. Sensors and cameras capture data →
2. Preprocessing filters noise →
3. AI Model analyzes data →
4. Threshold Check determines accident status →
5. Alerts are sent if an accident is confirmed →
6. Data is logged for future analysis.

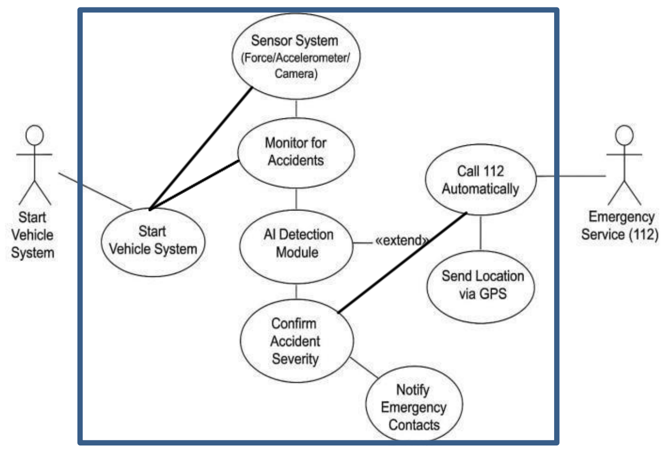
#### 3.3.3 Technology Stack

The system is built on a robust technology stack that ensures efficient accident detection, processing, and communication. At the hardware level, the system incorporates sensors such as the MPU6050 accelerometer for motion detection and impact analysis, along with a Raspberry Pi Camera for capturing visual data. These sensors provide essential real-time inputs for accident detection. For processing, the system leverages Python as the primary programming language, utilizing OpenCV for image processing and TensorFlow/Keras for machine learning-based accident detection. These frameworks enable the AI model to accurately analyse sensor and video data, distinguishing between minor incidents and serious accidents. For seamless communication between system components, MQTT and WebSockets are employed, ensuring efficient data transfer with minimal latency. The system's backend relies on cloudbased databases such as Firebase and PostgreSQL, which store accident data, sensor logs, and emergency alerts securely while allowing for scalability and remote access. On the frontend, the system is designed using ReactJS for an intuitive and user-friendly interface, while Node.js handles backend logic and API interactions. This technology stack ensures that the system operates in real-time, accurately detects accidents, and provides quick emergency response, making it a reliable and scalable solution for accident detection and reporting.

## 4. USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

**Use case diagram**



The AI-based vehicle accident detection and emergency notification system is designed to enhance road safety by detecting accidents in real-time and ensuring a swift emergency response. The system consists of several key components, including actors, use cases, relationships, and system boundaries, each playing a crucial role in its functionality.

The actors in the system include the User/Driver, who initiates the vehicle system, and Emergency Services (112), which receives alerts and location data when an accident occurs. The system is structured around multiple use cases, starting with the Start Vehicle System, which is triggered by the driver and activates all subsequent monitoring and AI modules. The Sensor System, comprising force sensors, accelerometers, and cameras, begins collecting real-time data as soon as the vehicle is started, feeding this information into the Monitor for Accidents module. This module continuously analyzes sensor data to identify potential accidents, passing the information to the AI Detection Module, which utilizes advanced artificial intelligence techniques to determine whether an accident has occurred.

Upon detecting an accident, the system proceeds to Confirm Accident Severity, assessing the impact's magnitude and determining the level of emergency response required. If the accident is deemed significant, the system initiates the Notify Emergency Contacts use case, sending alerts to the user's pre-registered contacts. In severe cases, the system extends into the Call 112 Automatically use case, ensuring that emergency responders are alerted immediately. Simultaneously, the system triggers Send Location via GPS, providing precise location data to emergency services to facilitate a quick response.

The system's relationships are structured to ensure seamless operation. The extend relationship between the AI Detection Module and Call 112 Automatically signifies that emergency calls are only placed when an accident is confirmed as severe. Various associations link actors to their respective use cases, such as the connection between Emergency Services and both Call 112 Automatically and Send Location via GPS, ensuring a direct response to critical situations.

All Internal processes are enclosed within the System Boundary, visually distinguishing the core functionalities of the system from external actors. This structured approach ensures a comprehensive, reliable, and automated accident detection and response mechanism, significantly improving road safety and emergency response times.

## 5. SCREENSHOTS

```
accident_detection_project/
│
├── data/
│   ├── sensor_data.csv     # Manual sensor data
│
├── models/
│   ├── accident_model.pth  # Saved PyTorch model
│
├── src/
│   ├── train_model.py      # For training AI model
│   ├── detect_accident.py  # For real-time/manual detection
│   ├── call_service.py     # Simulated emergency call module
│
└── README.md
```

**Figure 1 Project Directory**

Sample `sensor_data.csv`:

| X_axis | Y_axis | Z_axis | Gyro_X | Gyro_Y | Gyro_Z | Accident |
|---|---|---|---|---|---|---|
| 0.5 | 0.3 | 9.8 | 0.02 | 0.01 | 0.00 | 0 |
| 5.0 | 4.2 | 3.0 | 3.50 | 2.80 | 1.20 | 1 |
| 0.6 | 0.2 | 9.7 | 0.01 | 0.02 | 0.00 | 0 |

- **Accident** = 1 (detected), 0 (normal).

**Figure 2 Data of sensors**

```python
import pandas as pd
import torch
from torch.utils.data import DataLoader, TensorDataset
from sklearn.model_selection import train_test_split

data = pd.read_csv('../data/sensor_data.csv')
X = data.drop('Accident', axis=1).values
y = data['Accident'].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

train_dataset = TensorDataset(torch.tensor(X_train, dtype=torch.float32), torch.tensor(y_train, dtype=torch.long))
train_loader = DataLoader(train_dataset, batch_size=4, shuffle=True)
|
```

**Figure 3 Model development To load data**

```
import torch.nn as nn

class AccidentDetector(nn.Module):
    def __init__(self):
        super(AccidentDetector, self).__init__()
        self.fc1 = nn.Linear(6, 16)
        self.fc2 = nn.Linear(16, 8)
        self.fc3 = nn.Linear(8, 2)  # Binary output

    def forward(self, x):
        x = torch.relu(self.fc1(x))
        x = torch.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

**Figure 4 Model Development To define model**

```
model = AccidentDetector()
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=0.001)

for epoch in range(50):
    for inputs, labels in train_loader:
        outputs = model(inputs)
        loss = criterion(outputs, labels)
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
    print(f'Epoch {epoch+1}, Loss: {loss.item()}')

# Save model
torch.save(model.state_dict(), '../models/accident_model.pth')
```

**Figure 5 Model Development To train model**

```
import torch

model = AccidentDetector()
model.load_state_dict(torch.load('../models/accident_model.pth'))
model.eval()
```

**Figure 6 Real time accident detection load trained model**

```
def manual_input():
    x = float(input("Enter X-axis: "))
    y = float(input("Enter Y-axis: "))
    z = float(input("Enter Z-axis: "))
    gx = float(input("Enter Gyro-X: "))
    gy = float(input("Enter Gyro-Y: "))
    gz = float(input("Enter Gyro-Z: "))
    return torch.tensor([[x, y, z, gx, gy, gz dtype=torch.float32)
```

**Figure 7 Real time accident detection Manual data input**

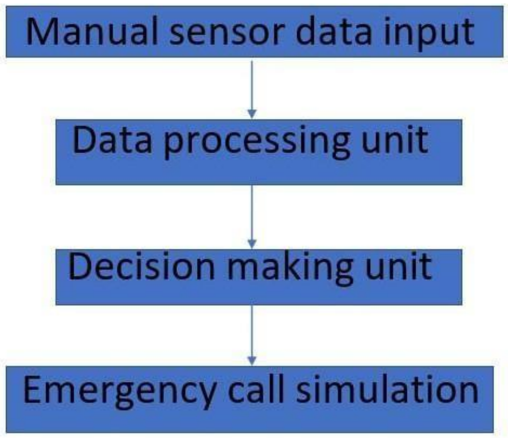
```
def predict(data):
    output = model(data)
    _, predicted = torch.max(output, 1)
    return predicted.item()

data = manual_input()
result = predict(data)
if result == 1:
    print("Accident Detected! Calling 112...")
    from call_service import make_call
    make_call()
else:
    print("No accident detected.")
```

**Figure 8 Real time accident detection Predict accident**

```
def make_call():
    print("Simulating call to 112... [CONNECTED]")
```

**Figure 9 Emergency response activation**

## 6. Block Diagram




## 7. CONCLUSION AND FUTURE WORK

CONCLUSION

AI-based accident detection systems with limited force impact assessment and automatic emergency call functionality play a crucial role in enhancing road safety. By utilizing advanced sensors, machine learning algorithms, and real-time data processing, these systems can detect accidents accurately and promptly alert emergency responders. This reduces response time, potentially saving lives and minimizing the severity of injuries. The integration of such technology into vehicles contributes to intelligent transportation systems, ensuring safer roads and better accident management.

FUTURE SCOPE

The future of AI-based accident detection systems holds immense potential, with advancements aimed at increasing accuracy, efficiency, and integration with modern technologies. One key area of improvement is the enhancement of **AI algorithms**, where deep learning models will evolve to detect a broader range of accident scenarios with greater precision, reducing false positives and improving reliability. Additionally, **IoT and smart infrastructure integration** will enable these systems to connect with smart city infrastructure and traffic management networks, facilitating real-time accident prevention and faster emergency response. The adoption of **5G and cloud computing** will further accelerate communication and data processing, significantly reducing latency and ensuring that emergency services receive alerts instantaneously.
Another promising development is the **integration of wearable and mobile-based accident detection**, extending protection to cyclists, motorcyclists, and pedestrians through wearable devices and smartphone applications, enhancing road safety beyond vehicles. In the realm of **autonomous vehicle safety**, AI-driven accident detection will play a crucial role in improving collision avoidance and emergency response systems in self-driving cars, making them safer and more responsive in unpredictable situations. Furthermore, **multi-modal communication** will enhance emergency alert mechanisms by incorporating various channels such as SMS, automated voice calls, and push notifications, ensuring timely and efficient communication with first responders and emergency contacts. These advancements will collectively transform accident detection systems, making them more intelligent, proactive, and capable of significantly reducing accident-related casualties and response times.

## ACKNOWLEDGEMENT

We express our sincere gratitude to **Ms.Sajitha**, Assistant Professor, Department of Computer Science and Engineering, NCERC, for his valuable guidance, support, and encouragement throughout the course of this project. His mentorship played a vital role in shaping our research and ensuring its successful completion. We also thank Nehru College of Engineering and Research Centre (NCERC) for providing the infrastructure and resources necessary to carry out this project effectively. We sincerely appreciate the efforts of all our project team members for their dedication, cooperation, and hard work at every stage of development.We are especially grateful to the volunteers who contributed fingerprint data during the testing phase, as well as to our parents and peers for their continuous support and motivation throughout this journey.

## REFERENCES

1. Kumar, S., & Gupta, R. (2022). "Emergency alert system for road accidents using GPS and GSM modules." International Journal of Engineering Trends and Technology (IJETT), 70(1), 54-59.

2. Sharma, N., & Patel, M. (2022). "Emergency Response System using GSM and GPS." Journal of Advances in Science and Technology, 15(2), 80-84.

3. World Health Organization (WHO). (2022). "Global status report

4. Khandelwal, M., & Jain, A. (2021). "Force sensors in accident detection: A review." Journal of Sensor Technology, 11(1), 13-19.

5. Al-Suwaidi, A., & Ismail, N. (2021). "IoT-based smart accident detection and rescue system using cloud technology." Sensors, 21(10), 3556.

6. Roy, A., & Das, S. (2021). "Wireless accident alert system using GSM and GPS modules." International Journal of Engineering and Advanced Technology (IJEAT), 10(5), 45-49.

7. Ramesh, R., & Deepa, T. (2020). "Smart Helmet with Accident Detection and Automatic Alert System." International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(3), 82-85.

8. Ahmed, M., & Singh, R. (2020). "AI and Machine Learning Approaches for Road Accident Detection." IEEE Access, 8, 123456–123469.

9. Rani, D., & Arora, M. (2020). "Accident detection using CNN and edge computing." Journal of Ambient Intelligence and Humanized Computing, 11, 4033–4040.

10. Singh, A., & Thomas, J. (2020). "Low-cost accident detection system using Arduino and sensors." International Journal of Research in Electronics and Computer Engineering (IJRECE), 8(2), 237-242.

11. Yadav, P., & Verma, S. (2020). "Design and implementation of smart accident detection system using force sensor." International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, 9(3), 181-186.

12. Ramesh, R., & Deepa, T. (2020). "Smart Helmet with Accident Detection and Automatic Alert System." International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(3), 82-85.

13. Zhang, Y., & Liu, X. (2019). "Deep learning-based accident detection from surveillance videos." Pattern Recognition Letters, 125, 533-540.

14. Dey, S., & Roy, M. (2019). "Integration of AI in emergency response for road safety." International Journal of Emerging Trends in Engineering Research (IJETER), 7(11), 540-545.

15. Karthik, S., & Prabhu, E. (2019). "Smart vehicle accident detection system using accelerometer and GPS." International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 4(1), 245-250.

16. Chavan, A., & Patil, D. (2018). "Automatic accident detection and rescue system using GSM and GPS module." International Journal of Electrical and Computer Engineering, 8(3), 1451-1458.

17. Sahoo, P. K., & Kim, H. (2018). "Personalized accident detection and alert system using mobile sensors." IEEE Transactions on Mobile Computing, 17(5), 1031–1044.

18. AI Enabled Accident Detection and Alert System by Rajeev Kumar Gupta Computer Science and Engineering, Pandit Deendayal Petroleum University, Gandhinagar 382007, India

19. Chat GPT