IoT-Enabled Driver Safety & Crash Detection System

Sahana Narasipura Vasudevarao
Electrical Engineering and Computer Science Department
Syracuse University
Syracuse, NY, United States
snarsipu@syr.edu

Sahil Chaubal
Electrical Engineering and Computer Science Department
Syracuse University
Syracuse, NY, United States
sachauba@syr.edu

Shrinidhi Kulkarni
Electrical Engineering and Computer Science Department
Syracuse University
Syracuse, NY, United States
skulka18@syr.edu

Siddhesh Save
Electrical Engineering and Computer Science Department
Syracuse University
Syracuse, NY, United States
ssave01@syr.edu

Abstract*—Ensuring driver safety continues to be a pressing issue, with fatigue, negligence, and slow emergency response times playing a major role in road accidents. Most current solutions emphasize post-incident tracking and data logging, offering little in terms of real-time accident prevention. This project introduces the design and simulation of an "IoT-Enabled Driver Safety & Crash Detection System," built around a hybrid framework combining the ESP32 microcontroller with Python-based AI modules. The system features proactive safety measures including seat belt usage verification, alcohol level monitoring, crash detection through accelerometer data, GPS-enabled emergency alerting, and real-time drowsiness monitoring using computer vision techniques. The ESP32-based module, developed using the Wokwi simulation platform, replicates key vehicular conditions through various sensors: a slide switch for seat belt status, a potentiometer to simulate alcohol levels, an MPU6050 accelerometer for crash detection, and an OLED display for in-vehicle feedback. Sensor data is logged to the ThingSpeak cloud for real-time visualization, while Twilio integration enables automated SMS alerts in the event of a crash. A simulated GPS module dynamically generates location data, which is included in emergency reports to assist with rapid response. To further improve driver monitoring, a Python-based computer vision module leverages OpenCV, dlib, and facial landmark detection to track the Eye Aspect Ratio (EAR). By monitoring prolonged eye closure, a strong indicator of fatigue the system can detect signs of drowsiness and issue timely alerts. Though it operates independently from the ESP32 setup, this vision module complements it by focusing on accident prevention through behavioral analysis. ThingSpeak outputs confirm accurate sensor data logging across key parameters, including seatbelt status, alcohol level, acceleration, crash detection, and GPS-based location. The Wokwi simulation validates the system's real-time responsiveness, while the drowsiness detection module effectively identifies signs of driver fatigue through live webcam input. This integrated solution addresses the critical aspects of prevention, detection, and emergency response offering a scalable prototype aimed at improving road safety. By leveraging open-source platforms and cost-effective hardware, the system demonstrates strong potential for real-world adaptation. It highlights how the convergence of IoT and AI can drive the development of intelligent transportation systems focused on proactive and personalized driver assistance.

Keywords—IoT, Driver Safety, Crash Detection, ESP32, Drowsiness Detection, Computer Vision, Eye Aspect Ratio (EAR), OpenCV, dlib, ThingSpeak, Twilio, Accelerometer, Intelligent Transportation Systems

I. Introduction

The exponential growth in vehicle ownership and road traffic has been accompanied by a significant rise in traffic accidents globally. According to international road safety statistics, a large percentage of these incidents are caused by preventable human errors such as driver fatigue, alcohol impairment, inattention, and delayed emergency response. While traditional vehicle safety systems have made strides in addressing post-collision response often through GPS- or GSM-based alert mechanisms [1][2] they fall short in detecting and responding to the risk factors before an incident occurs. These reactive systems are critical for mitigation but insufficient for prevention, which is the more effective long-term strategy.

To address these gaps, the IoT-Enabled Driver Safety & Crash Detection System is proposed as a holistic, proactive safety solution. This system is built around the ESP32 microcontroller, chosen for its connectivity, versatility, and cost-efficiency, and integrates a suite of simulated sensors. These include a digital slide switch to emulate seatbelt usage, a potentiometer to simulate varying alcohol concentration levels, and an MPU6050 accelerometer for detecting sudden motion indicative of a crash. Data collected from these sensors are displayed to the driver in real time via an OLED screen and are also continuously transmitted to ThingSpeak, a cloud platform used for visualizing and logging sensor data.

Beyond these hardware-level safeguards, the system integrates an intelligent computer vision module developed in Python using OpenCV, dlib, and imutils. This module analyzes the driver's facial landmarks to calculate the Eye Aspect Ratio (EAR), a reliable metric for detecting prolonged eye closure indicative of drowsiness a major factor in road accidents [8]. When drowsiness is detected, immediate visual and terminal-based alerts are triggered, providing a critical early warning against fatigue-induced incidents. Developed and thoroughly tested within virtual simulation platforms like Wokwi and Proteus [2][3], the integrated solution combines IoT-based sensor monitoring with AI-driven behavioral analysis into a unified, proactive safety framework. Its modular, real-time, and cost-effective design positions it as a scalable model capable of actively preventing accidents and advancing intelligent transportation systems.

II. Literature Review & Significance

Existing literature in the domain of driver safety has largely been directed towards post-incident detection and notification. Prior solutions have relied heavily on GPS and GSM modules to alert emergency contacts once an accident occurs [1][2]. There are also studies that focus on individual components, such as alcohol detection using gas sensors [3] or crash detection via vibration sensing methods [4]. More recent research has explored using AI-based approaches for detecting driver drowsiness through facial and behavioral cues [5][6]; however, these systems generally operate in silos and do not provide a unified solution.

This project stands apart by combining multiple safety features into a single framework. The system not only monitors seatbelt compliance and alcohol levels but also detects sudden accelerations indicative of crashes. Meanwhile, the drowsiness detection module adds an additional layer of prevention, ensuring that driver inattention is recognized early [8]. By integrating all these elements with real-time cloud analytics on ThingSpeak and rapid emergency response via Twilio SMS alerts, the solution bridges the gap between isolated systems. This holistic approach is significant because it offers a scalable, cost-effective, and proactive alternative to traditional safety systems.

The importance of such a unified system is further underscored by its potential for widespread adoption in varied mobility contexts from urban centers to rural areas where timely intervention can significantly reduce accident rates.

III. Problem Analysis

Despite the growing adoption of vehicle safety technologies, many modern systems remain fundamentally reactive in nature. Features such as GPS-based crash alerts and post-incident emergency notifications, while valuable for improving response times, do little to prevent the occurrence of accidents in the first place. These systems are typically designed to activate after a collision has occurred, rather than identifying and addressing the risky behaviors or environmental conditions that may lead to it. As a result, they fail to provide the proactive intervention necessary to reduce accident rates effectively.

In addition, existing solutions tend to focus on isolated safety parameters such as seatbelt usage, alcohol consumption, or drowsiness without integrating these into a broader, real-time monitoring framework. This compartmentalized approach does not reflect the complexity of real-world driving risks, which often arise from a combination of human error, environmental factors, and vehicle conditions. A driver may be wearing a seatbelt, for instance, but still be intoxicated or drowsy, rendering the presence of a single safety check insufficient.

Another key limitation lies in the cost and complexity of deploying comprehensive safety systems. High-end commercial solutions often involve sophisticated sensor arrays, proprietary software, and custom hardware configurations that are expensive to implement and maintain. These systems are not only inaccessible in cost-sensitive environments such as public transport or developing regions, but they also lack the flexibility needed for rapid prototyping, upgrades, or integration into diverse vehicle platforms.

There is, therefore, a clear need for a modular, cost-effective, and scalable system that can combine various safety detection methods into a unified platform. Such a system should be able to proactively detect early indicators of danger like elevated alcohol levels, excessive speed, seatbelt violations, or signs of fatigue and trigger immediate alerts or corrective actions before a critical event occurs. Additionally, it should be capable of logging data to the cloud for remote monitoring and historical analysis, enabling broader applications such as fleet management, insurance validation, and predictive maintenance.

The IoT-Enabled Driver Safety & Crash Detection System proposed in this project addresses these gaps by integrating hardware-based sensing with cloud connectivity and computer vision–based behavioral analysis. It brings together multiple detection layers physical, cognitive, and situational into one cohesive ecosystem. The system is designed to minimize false positives through calibrated thresholds, reduce dependency on proprietary hardware, and offer flexible deployment options. In doing so, it provides a more comprehensive, proactive, and inclusive solution to the modern challenges of road safety.

IV. Methodology

To address the above challenges, this system was developed as a hybrid model using both embedded simulation and external AI modules. The methodology was divided into two main parts:

A. *Embedded IoT Subsystem (ESP32 + Wokwi)*

The embedded portion was built on the ESP32 microcontroller and simulated using the Wokwi platform to emulate real-world driving conditions without physical hardware:

- Seat Belt Monitoring: Simulated using a digital slide switch connected to the microcontroller. The system restricts all operations unless the seatbelt is detected as fastened, enforcing basic safety compliance before vehicle movement.
- Alcohol Detection: A potentiometer simulates varying alcohol concentration levels by producing analog input values. The system evaluates these values and triggers warnings if the level crosses a predefined intoxication threshold, modeling real-world breathalyzer behavior.
- Crash Detection: An accelerometer module measures motion and sudden changes in velocity. The system interprets rapid deceleration or free fall as crash scenarios, enabling early crash detection even without direct impact sensors.
- OLED Display: A compact OLED screen displays real-time system feedback, including seatbelt status, alcohol levels, and vehicle speed (inferred from accelerometer data), ensuring the driver is constantly informed of their safety status.
- Twilio Integration: In crash situations, the system invokes the Twilio API to send emergency SMS alerts to a predefined contact. This mechanism ensures critical information including crash confirmation and location is communicated immediately.
- ThingSpeak Logging: All sensor readings are transmitted to the ThingSpeak cloud platform,

allowing centralized storage and visualization of parameters such as seatbelt state, alcohol level, acceleration, crash detection status, and geographic coordinates.

- GPS Simulation: Instead of using physical GPS hardware, the system generates randomized but realistic latitude and longitude coordinates to simulate the vehicle's position during a crash. These coordinates are included in cloud logs and SMS alerts to mimic location tracking.

B. *Drowsiness Detection Module (Python + OpenCV + Dlib)*

To enhance the system's preventive capabilities, a standalone Python-based computer vision module was developed, focusing on monitoring the driver's facial behavior:

- Live Video Feed Capture: The system captures continuous video from the user's webcam, providing a live stream of the driver's face to analyze their eye activity in real time without requiring external sensors.
- Facial Landmark Detection: A pretrained shape predictor model from the dlib library is used to detect 68 facial landmarks, with specific emphasis on the regions surrounding the eyes. These landmarks allow precise tracking of subtle facial movements.
- Eye Aspect Ratio (EAR) Calculation: Using six eye-specific landmarks, the system computes the Eye Aspect Ratio (EAR), a mathematical metric that indicates whether eyes are open or closed. This value is central to determining driver fatigue.
- Drowsiness Condition Detection: If the EAR falls below a threshold for more than 2 seconds, the system interprets it as prolonged eye closure and triggers a drowsiness alert, distinguishing between normal blinking and actual fatigue.
- Real-Time Visual Feedback: The module overlays real-time data on the video feed, including EAR values and bounding contours around the eyes. When drowsiness is detected, a prominent warning is displayed to notify the driver immediately.

This modular integration allows the IoT system to handle hardware-based monitoring and alerts, while the AI module handles visual fatigue detection. Both systems operate independently but in parallel, offering a scalable and layered safety framework that targets both environmental and human risk factors.

V. Implementation

The IoT-Enabled Driver Safety & Crash Detection System is developed through two parallel but complementary subsystems: an embedded IoT-based simulation utilizing an ESP32 microcontroller and a computer vision–based module using Python for drowsiness detection.

A. *Embedded IoT System (ESP32 on Wokwi)*

The embedded IoT subsystem leverages the ESP32 DevKit microcontroller, renowned for its versatility and cost-effectiveness, making it ideal for IoT applications. The entire system was simulated using the Wokwi online platform, chosen for its capability to accurately replicate the behavior of physical hardware components and sensors virtually.

- Seatbelt Compliance Monitoring (Slide Switch): The seatbelt compliance system is simulated using a digital slide switch interfaced with GPIO pin 2 of the ESP32. The microcontroller firmware constantly monitors the state of this switch, ensuring that vehicle operations can proceed only when the seatbelt is securely fastened. If the seatbelt remains unfastened, the system triggers an alert mechanism designed to encourage compliance.
- Alcohol Intoxication Detection (Potentiometer Simulation): Alcohol detection within the system is represented using a potentiometer connected to analog pin 34. This device simulates a real-world breathalyzer by providing analog voltage values corresponding to different levels of alcohol concentration. The ESP32 firmware processes these analog signals, comparing them against a predefined threshold (0.08). If this threshold is exceeded, indicating potential driver impairment, the system activates immediate visual and auditory warnings to prevent unsafe driving conditions.
- Crash Detection Mechanism (MPU6050 Accelerometer): A simulated MPU6050 accelerometer module serves as the core sensor for collision detection. Integrated via the I2C communication protocol, this sensor continuously measures acceleration values along multiple axes. Sudden changes in these values, specifically rapid deceleration, or near-zero acceleration indicative of a collision, trigger the crash detection protocol. Upon detecting such abnormal motion, the system automatically activates safety response procedures, including visual and auditory alerts and emergency notifications.
- Crash Event Trigger (Pushbutton Simulation): To robustly test crash detection algorithms, a pushbutton connected to GPIO pin 23 simulates crash scenarios. When pressed, this digital input signals a simulated crash event to the firmware, prompting the immediate execution of emergency response measures. This simulated scenario ensures the system's reliability under potential real-world conditions.
- Real-time Status Feedback (OLED Display): An integrated SSD1306 OLED display, communicating through the I2C protocol, provides continuous visual feedback to the user. Displayed information includes critical safety parameters such as seatbelt status, current alcohol levels, and instantaneous acceleration readings. This real-time feedback mechanism is crucial for informing the driver of their current safety status, allowing prompt corrective actions if necessary.
- Immediate Alerting Mechanisms (LED & Buzzer): Instantaneous alerts are essential for effective driver feedback. To achieve this, the system incorporates an LED and a piezoelectric buzzer connected to GPIO pins. These components promptly activate whenever safety violations, such as unfastened seatbelts or excessive alcohol levels, are detected. The audible and

visual signals ensure quick recognition of critical safety issues, prompting rapid corrective measures from the driver.

The ESP32's built-in Wi-Fi capability allows the system to connect to internet networks seamlessly. Within the simulation, this connectivity emulates real-world conditions, enabling continuous data transmission to cloud-based analytics services. Real-time sensor data from the ESP32 is transmitted to the ThingSpeak cloud platform, a robust IoT analytics service. Organized into structured data fields, the sensor values include:

- Field 1: Seatbelt Status
- Field 2: Alcohol Level
- Field 3: Acceleration Data
- Field 4: Crash Status
- Field 5: Latitude
- Field 6: Longitude

This centralized logging facilitates comprehensive analysis and visualization of the system's safety parameters, enabling remote monitoring and future predictive analyses.

In critical scenarios, such as a detected crash, the system employs the Twilio SMS API for instantaneous emergency communications. Upon triggering the crash detection algorithm, the ESP32 sends an automated SMS alert containing simulated GPS coordinates to predefined emergency contacts. This rapid-response functionality significantly enhances the efficacy of emergency interventions following an accident.

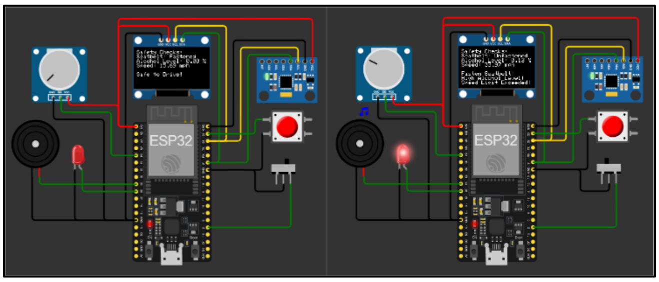

Fig. 1. IoT system displays for safe and violation conditions.

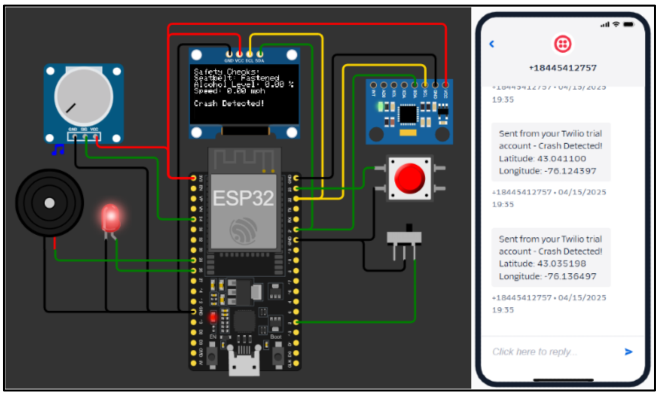

Fig. 2. Crash detection simulation with Twilio SMS emergency alerts.

The ESP32 firmware serves as the central intelligence of the embedded system, executing continuous sensor monitoring and safety protocols. The core functions of the firmware include:

- Real-time Monitoring: Sensor inputs are continuously polled and evaluated against predefined safety thresholds, ensuring prompt detection of potential safety violations.
- Threshold-based Decision Making: Upon identifying violations (e.g., alcohol levels exceeding 0.08 BAC, seatbelt unfastened, or accelerometer indicating crash), the firmware instantly activates relevant alert mechanisms (LED and buzzer) and prepares data packets for immediate cloud transmission.
- Emergency Event Processing: Detected crash events trigger specific emergency-response routines, involving both local alerts and remote SMS notifications. The firmware simulates GPS coordinates for emergency messaging, providing crucial location information to emergency responders.

Through meticulous design, calibration, and testing within the simulated environment, the embedded IoT subsystem demonstrates robust performance, reliable integration of hardware and software components, and effective real-time safety management. The system's modular and scalable nature ensures it can readily transition from a simulated prototype to a real-world hardware implementation with minimal adjustments.

B. *Drowsiness Detection Module (Python Script)*

To complement the embedded IoT subsystem and enhance overall driver safety, a dedicated drowsiness detection module was developed using Python, leveraging popular computer vision libraries such as OpenCV, dlib, and imutils. This module aims to proactively identify driver fatigue through real-time analysis of visual indicators, specifically eye behavior.

The module integrates the following technologies and libraries to achieve robust drowsiness monitoring:

- OpenCV: OpenCV (Open Source Computer Vision Library) serves as the core framework for real-time image processing and video analysis. It provides efficient and robust functionality for video capture from the driver-facing webcam, preprocessing video frames, and rendering visual feedback with real-time annotations.
- dlib: The dlib library supplies advanced machine learning and data analysis tools, notably a Histogram of Oriented Gradients (HOG)-based frontal face detector. This powerful face detection approach reliably identifies the driver's face within video frames, providing accurate bounding boxes that serve as the starting point for further analysis.
- Facial Landmark Detection: Utilizing dlib's pretrained facial landmark predictor, the system identifies and localizes 68 distinct facial landmarks. These landmarks accurately pinpoint key facial structures, including the contours around the eyes, mouth, eyebrows, and nose. Precise localization of these landmarks is crucial for subsequent eye-specific computations.

The following sequential steps outline the real-time analysis pipeline for driver fatigue detection:

- Real-Time Video Capture: The system captures continuous live video from a webcam positioned to clearly capture the driver's face. Each incoming frame undergoes preprocessing steps, including color-to-grayscale conversion, which significantly optimizes computational efficiency and detection accuracy.

- Face Detection: For every processed frame, dlib's face detector scans and accurately identifies the driver's face, generating rectangular bounding boxes around detected faces. This step provides the spatial context needed for precise facial landmark detection.

- Facial Landmark Localization: Within each detected face region, the landmark predictor identifies 68 distinct facial landmarks. These landmarks are points representing crucial facial anatomy, and specifically, six key points around each eye are isolated for detailed eye behavior analysis.

- Eye Region Extraction: Using the landmark indices provided by dlib, the module isolates the coordinates corresponding to the left and right eye contours. These coordinates are essential for subsequent computation of eye openness through the Eye Aspect Ratio (EAR).

- Eye Aspect Ratio (EAR) Computation: EAR is computed based on six points surrounding each eye. The mathematical formula for EAR calculation is:

$$EAR = \frac{\| p_2 - p_6 \| + \| p_3 - p_5 \|}{2 \| p_1 - p_4 \|}$$

  Here, points p1 and p4 represent horizontal eye landmarks (eye corners), while points p2, p3, p5, and p6 represent vertical eye landmarks. Specifically, the numerator calculates the vertical eye-opening distance at two different points and the denominator measures the horizontal width of the eye. The EAR value provides a normalized metric indicating the openness or closure state of the eyes.

  Under normal conditions (eyes open), the EAR remains relatively stable at a higher value. However, when the driver's eyes begin closing due to drowsiness, the EAR decreases significantly.

- Continuous Monitoring and Thresholding: EAR values are monitored over consecutive frames to assess eye closure patterns. The system defines critical thresholds and temporal criteria to determine drowsiness. EAR Threshold is set empirically at **0.25**, indicating substantial eye closure when values fall below this threshold. To distinguish between normal blinking and drowsiness, the system checks whether EAR remains below the defined threshold for a set number of consecutive frames. In this implementation, prolonged eye closure lasting more than 48 consecutive frames is considered indicative of driver fatigue.

- Alerting and Feedback Mechanisms: Upon detecting sustained low EAR values (indicative of prolonged eye closure), the module immediately activates alert mechanisms. A prominent visual warning, "DROWSINESS ALERT!", appears directly on the real-time video feed, clearly visible to the driver. Concurrently, an audible or textual wake-up alert is triggered through the terminal or integrated audio system, providing additional immediate feedback to the driver.

The module significantly enhances driver safety by providing the following preventive benefits:

- Proactive Safety Measure: By continuously analyzing driver behavior, specifically through eye closure patterns, the system proactively identifies early signs of fatigue. Early detection and intervention help significantly reduce the risk of fatigue-related road accidents.

- Real-Time Performance: The module operates effectively under real-time constraints, processing video frames quickly enough to provide instant feedback without perceptible delays. Such real-time responsiveness is critical for timely intervention in fatigue situations.

- Accessibility and Cost-effectiveness: By leveraging open-source technologies such as OpenCV and dlib, the module remains accessible, easy to implement, and cost-effective. This enhances the scalability and potential for broad adoption in both commercial and personal vehicle safety systems.

Future work could expand the module's capabilities through enhancements such as:

- Adaptive Thresholding: Employing adaptive or machine learning-based EAR thresholds to dynamically adjust detection sensitivity based on environmental conditions, lighting variability, and individual driver behaviors.

- Integration with IoT Subsystem: Seamless real-time communication between the drowsiness detection module and the embedded IoT subsystem can allow for synchronized safety responses, such as automatic vehicle speed reduction or emergency services notifications in severe fatigue cases.

- Enhanced Feedback Channels: Integration of auditory or haptic feedback mechanisms to ensure quicker and more effective driver alerts, especially during critical situations where visual attention may already be compromised.

This comprehensive computer vision-based drowsiness detection module significantly enriches the IoT-Enabled Driver Safety & Crash Detection System, emphasizing proactive accident prevention and greatly enhancing overall driver safety.

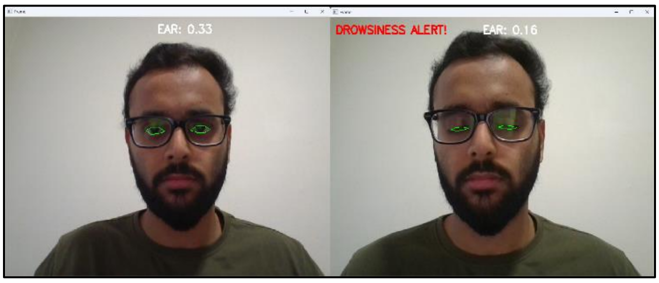

Fig. 3. Real-time EAR-based driver drowsiness alert system.

VI. Results

The IoT-Enabled Driver Safety & Crash Detection System underwent rigorous evaluation through detailed simulation testing for the IoT subsystem and real-time testing for the drowsiness detection module. The outcomes from both evaluation scenarios demonstrated that the integrated system performs robustly and accurately under conditions closely mimicking real-world driving environments.

- ThingSpeak Cloud Analytics: The data logging capability was successfully verified through real-time updates transmitted to the ThingSpeak cloud platform. All six monitored fields—seatbelt compliance, alcohol level, acceleration data, crash detection status, and simulated GPS coordinates (latitude and longitude)—consistently updated without interruption or latency issues. The graphical visualization clearly illustrated variations corresponding to different simulated scenarios, such as seatbelt violations, high alcohol detection events, and collision occurrences. These cloud-based analytics validate the system's capability to reliably log and present critical safety data in a user-friendly, actionable format.

- Crash Detection Performance: The crash detection component exhibited exceptional reliability in accurately identifying collision events during simulation tests. Simulated crash events, generated by manually triggering a pushbutton mechanism, were consistently detected by the accelerometer subsystem. Immediately upon detection, the system correctly activated visual (OLED display and LED indicators) and auditory alerts (buzzer), providing instant feedback. Furthermore, crash events were reliably recorded, confirming the robustness of the collision detection algorithm and sensor calibration.

- Emergency SMS Alerts via Twilio: The system's capability to promptly inform emergency contacts upon detecting a crash was rigorously tested. SMS alerts were successfully dispatched through the Twilio API, consistently arriving at designated emergency contacts within seconds of the simulated crash event. These emergency notifications included dynamically generated GPS coordinates, ensuring accurate and timely location data. The immediate receipt of these SMS alerts underscores the reliability and efficacy of the system's emergency communication protocol.

- Real-Time Drowsiness Detection: The computer vision-based drowsiness detection module demonstrated precise real-time performance. During live testing with webcam input, the Eye Aspect Ratio (EAR) accurately reflected driver eye closure patterns, consistently dropping below the critical threshold (EAR < 0.25) during simulated prolonged eye closure. The system reliably distinguished between normal blinking and fatigue-induced eye closure, triggering a prominent visual "DROWSINESS ALERT!" message precisely after the defined threshold duration (48 consecutive frames). This visual alert effectively gained the driver's attention, validating the module's capability to identify and respond promptly to signs of driver fatigue.

VII. Outcome

The successful integration of the embedded IoT subsystem with the computer vision–based drowsiness detection module resulted in the development of a comprehensive, multifunctional prototype dedicated to enhancing driver safety. The combined approach effectively demonstrated capabilities in both proactive accident prevention and rapid accident detection, delivering continuous real-time monitoring of crucial safety parameters such as seatbelt usage, alcohol intoxication levels, sudden vehicular impacts, and driver fatigue.

Real-time sensor data logging and visualization through cloud analytics services allowed for continuous remote monitoring and provided insights into driver behavior and vehicle condition trends. Additionally, prompt alert mechanisms—including visual, auditory, and emergency SMS notifications—were reliably triggered during testing, validating the system's immediate responsiveness in critical safety scenarios.

The modular and scalable architecture of the system enables straightforward integration of future enhancements. Potential expansions include transitioning from simulated environments to physical hardware deployments, integrating mobile applications for remote monitoring and driver notifications, and leveraging advanced cloud-based analytics or AI-driven predictive maintenance and risk assessment services. This inherent scalability ensures the prototype's relevance and adaptability in addressing evolving road safety challenges.

VIII. Discussion and Conclusion

This project aimed to develop a unified system that not only detects accidents but actively prevents them through real-time driver monitoring and automated alerting. By combining embedded IoT sensors with AI-driven visual analysis, the system effectively addresses key safety concerns, including seatbelt noncompliance, alcohol intoxication, driver fatigue, and crash detection. The successful integration of cloud platforms (ThingSpeak), messaging APIs (Twilio), and computer vision models (dlib with OpenCV) highlights the potential of an interdisciplinary approach to enhance vehicle safety.

The implementation process provided several valuable insights. First, cloud-based platforms such as ThingSpeak proved to be highly effective for real-time monitoring, particularly in visualizing time-series data from multiple safety sensors. The integration of Twilio's SMS functionality added an extra layer of responsiveness, ensuring that crash alerts were promptly sent to emergency contacts. Second, utilizing Wokwi for simulation offered a safe and cost-effective environment for developing and testing the embedded system, eliminating the need for physical hardware. This facilitated a more efficient iterative design and debugging process.

The Python-based drowsiness detection module demonstrated the practical application of AI in enhancing driver safety. By utilizing the Eye Aspect Ratio (EAR) as a reliable metric, the system effectively identified signs of fatigue and triggered visual alerts. The use of open-source libraries ensured that the solution was both accessible and adaptable, allowing for future enhancements, such as integrating sound-based alarms or implementing automatic vehicle halting mechanisms.

Despite its success, several challenges were encountered. Fine-tuning the crash detection threshold with accelerometer data required careful calibration to minimize false positives. Similarly, adjusting the EAR threshold and frame count for drowsiness detection was essential, as environmental factors such as lighting conditions and camera quality influenced the system's accuracy.

In conclusion, this project illustrates the potential of combining IoT and AI to develop an intelligent, proactive driver safety system. The modular architecture ensures scalability for real-world applications, while the use of accessible tools facilitates adaptability in both academic and commercial settings. Future work could focus on deploying the system on physical hardware, incorporating GPS modules for enhanced location tracking, and adding voice-based feedback or mobile app integration to improve usability and expand the system's reach.

REFERENCES

[1] Aarya, D. S., C. K. Athulya, P. Anas, Basil Kuriakose, Jerin Susan Joy, and Leena Thomas. "Accident alert and tracking using Arduino." Int J Adv Res Electr Electron Instrument Eng 7, no. 4 (2018).

[2] IBRAHEM, WALEED MOHY ELDEEN, MOMEN YASEEN FADEL ALMOLA AHMED, MUSAB YAHIA MERGHANI, and HASSAN TAWFIQ HASSAN. "Accident Detection and Reporting System Using GPS and GSM Module." PhD diss., AlMughtaribeen university, 2017.

[3] Kumar, A. Ajith, V. Jaganivasan, T. Sathish, and S. Mohanram. "Accident detection and alerting system using GPS & GSM." *International Journal of Pure and Applied Mathematics* 119, no. 15 (2018): 885-891.

[4] Moorthy, CM Dakshina, and J. N. Reddy. "Recovery of interlaminar stresses and strain energy release rates in composite laminates." *Finite elements in analysis and design* 33, no. 1 (1999): 1-27.

[5] Kumar, Rajeev, and Harish Kumar. "Availability and handling of data recived through GPS device: In tracking a vehicle." In *2014 IEEE International Advance Computing Conference (IACC)*, pp. 245-249. IEEE, 2014.

[6] Lee, SeokJu, Girma Tewolde, and Jaerock Kwon. "Design and implementation of vehicle tracking system using GPS/GSM/GPRS technology and smartphone application." In *2014 IEEE world forum on internet of things (WF-IoT)*, pp. 353-358. IEEE, 2014.

[7] Kluga, Janis, and Ansis Kluga. "Driver's professional skills evaluation with modified GPS vehicle tracking system." In *2017 Electronics*, pp. 1-4. IEEE, 2017.

[8] Soukupova, Tereza, and Jan Cech. "Eye blink detection using facial landmarks." In *21st computer vision winter workshop, Rimske Toplice, Slovenia*, vol. 2, p. 4. 2016.