AI-Powered Real-Time Drowsiness Detection and Auto Braking

Ms B. Rekha[1], Midde Nandini[2], Mohammed Shoeb[3], Maradani Priyanka[4], Kukkala Anudeep[5]
[1]*Assistant Professor, Dept, of Electronics and Communication Engineering (ECE), Teegala Krishna Reddy Engineering College, Hyderabad, Telangana, India*
[2,3,4,5]*Students, Dept, of Electronics and Communication Engineering (ECE), Teegala Krishna Reddy Engineering College, Hyderabad, Telangana, India*

***Abstract*—Driver fatigue and drowsiness remain leading contributors to global road accidents, necessitating the development of intelligent in-vehicle safety systems. This research presents the design and implementation of an AI-powered real-time drowsiness detection and autonomous braking system designed to mitigate collision risks. The proposed system utilizes a hybrid approach, integrating computer vision algorithms with a hardware-based intervention mechanism. The software layer, developed in Python, leverages the OpenCV library for real-time video processing, while NumPy and SciPy are employed for mathematical modeling and calculating the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR). By utilizing a 68-point facial landmark detector, the system identifies micro-sleep patterns and frequent yawning with high precision.**

Upon detecting a drowsiness threshold, the software communicates via serial interface with an Arduino Nano microcontroller to initiate a multi-staged alert and safety protocol. The hardware response includes immediate auditory and haptic feedback through a buzzer and vibration motor, alongside visual warnings via LEDs. Simultaneously, a relay module is triggered to control a DC motor, simulating an automatic braking sequence by reducing vehicle speed autonomously. This integration of high-level machine learning with low-latency embedded hardware provides a robust fail-safe mechanism. Experimental results demonstrate that the system achieves high accuracy in varied lighting conditions, offering a scalable and cost-effective solution for enhancing automotive safety and reducing fatigue-induced fatalities.

***Index Terms*—Arduino nano, LED, buzzer, vibration motor, dc motor, relay, buzzer, open cv, scipy, nump, pycharm**

I. INTRODUCTION

The global escalation in road traffic accidents has identified driver fatigue and microsleep as leading contributors to fatal vehicular collisions. Drowsiness significantly impairs cognitive functions, reducing reaction times and situational awareness, which necessitates the development of robust, real-time monitoring systems. This research presents an integrated AI-Powered Real-Time Drowsiness Detection and Auto-Braking System, designed to bridge the gap between software-based facial analysis and physical vehicular intervention. The system utilizes a high-definition camera interface coupled with sophisticated computer vision algorithms to monitor the physiological state of the driver. By leveraging Python as the primary computational backbone, the system employs OpenCV for real-time image acquisition and pre-processing, while NumPy and SciPy are utilized for high-speed mathematical operations and the calculation of the Ear Aspect Ratio (EAR). This software stack allows for the precise detection of eye closure durations and yawning patterns, distinguishing between natural blinking and the onset of fatigue.

To translate digital detection into physical safety measures, the software communicates with an Arduino Nano microcontroller, which serves as the hardware's central processing unit. Upon the detection of a drowsiness threshold, the system initiates a multi-layered alert and intervention protocol. Immediate sensory feedback is provided through a high-decibel Buzzer and a Vibration Motor integrated into the steering or seat, designed to startle the driver back into an alert state. Simultaneously, visual indicators via

LEDs notify the cabin of the system's status. If the driver remains unresponsive, the system engages an Auto-Braking mechanism; the Arduino triggers a Relay module to modulate the power to the DC Motors, simulating a controlled deceleration to bring the vehicle to a safe halt. This hybrid approach combining deep-learning-based facial landmarks with an automated hardware response offers a comprehensive fail-safe mechanism that moves beyond mere warning systems to active collision avoidance.

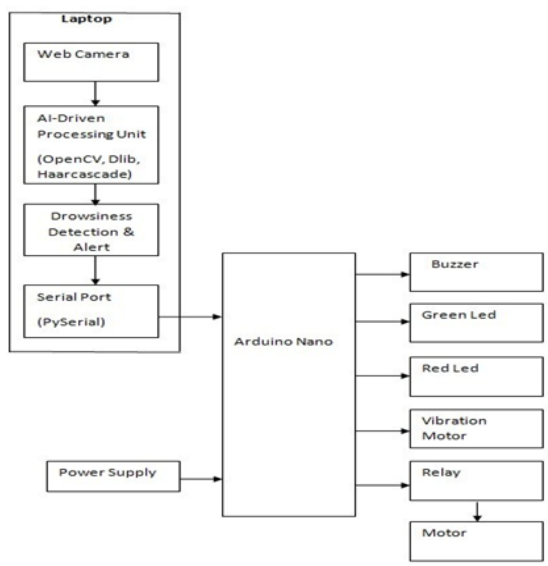

Figure 1: Block Diagram

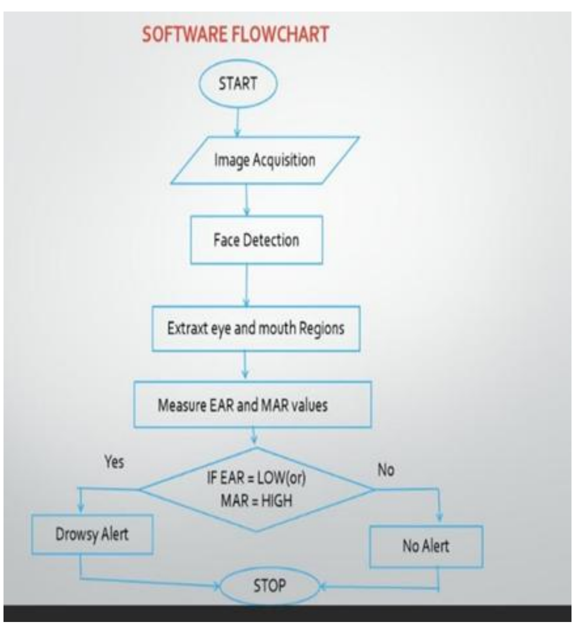

Figure 2: flow chart

Arduino Nano:

The Arduino Nano is a compact, versatile, and breadboard-friendly microcontroller board based on the ATmega328P (for version 3.x). It offers the same connectivity and processing power as the Arduino Uno but in a significantly smaller form factor, making it ideal for space-constrained projects. The board lacks a DC power jack, instead drawing power via a Mini-B USB connection or an unregulated 6-20V external power supply through the V_{in} pin. It features 14 digital input/output pins, of which 6 can provide PWM (Pulse Width Modulation) output to control motors or LED brightness.

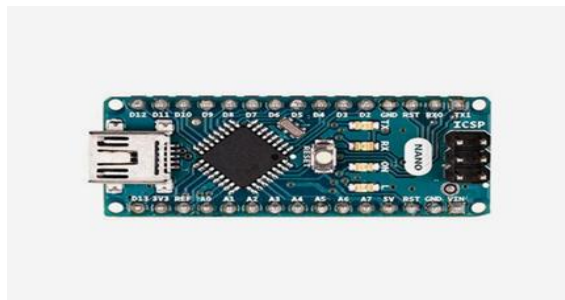

Figure 3: Arduino Nano

Vibration motor:

A vibration motor is a compact electromechanical device designed to generate haptic feedback or physical movement through rapid mechanical oscillations. These motors are ubiquitous in modern technology, primarily serving as the "silent" notification system in smartphones, pagers, and wearable devices. The fundamental operating principle relies on the concept of an unbalanced mass rotating around a shaft, commonly referred to as an Eccentric Rotating Mass (ERM). When the motor's internal rotor spins at high speeds, the off-centre weight creates a centrifugal force that shifts the motor's center of gravity, resulting in the characteristic buzzing sensation.

Figure 4: Vibration motor

DC motor:

A Direct Current (DC) motor is an electromechanical device that converts electrical energy into mechanical energy through the interaction of magnetic fields and conductors. The fundamental operating principle is based on Lorentz's Law, which states that a current-carrying conductor placed within a magnetic field experiences a physical force. DC motors are highly valued for their excellent speed control and high starting torque, making them ideal for applications requiring precision or heavy lifting. There are several types of DC motors, including brushed, brushless (BLDC), series-wound, and shunt-wound, each offering different performance characteristics. For instance, series motors are known for massive starting power, while shunt motors provide constant speed regardless of the load.

Figure 5: DC motor

Relay:

A relay is an electrically operated switch that allows a low-power signal to control a much higher-power circuit. It acts as a critical bridge between electronic control units and heavy-duty electrical loads, providing essential galvanic isolation to protect sensitive components from high-voltage surges. The core mechanism consists of an electromagnetic coil, an armature, and a set of metallic contacts. When current flows through the coil, it generates a magnetic field that pulls the armature toward it, physically moving the contacts to either complete or break an external circuit. This "electromechanical" action is what distinguishes traditional relays from solid-state versions. Relays are categorized by their contact configurations, such as Single Pole Single Throw (SPST) or Double Pole Double Throw (DPDT), which determine how many circuits they can control simultaneously.

Figure 6: Relay

LED:

Light Emitting Diodes, commonly known as LEDs, are semiconductor devices that convert electrical energy directly into light through a process called electroluminescence. Unlike traditional incandescent bulbs that rely on heating a filament, LEDs produce light when electrons move within a semiconductor structure. They consist of a P-N junction made from specific materials like Gallium Arsenide or Gallium Nitride; when a forward voltage is applied, electrons recombine with holes, releasing energy in the form of photons. The specific color of the light is determined by the energy band gap of the semiconductor material used. LEDs are incredibly energy-efficient, consuming significantly less power than halogen or fluorescent lamps while offering a much longer operational lifespan, often exceeding 50,000 hours. They are highly durable because they lack fragile filaments or glass enclosures, making them resistant to shock and vibration.

Figure 7: LED

Buzzer:

A buzzer is a compact yet essential electronic signaling device designed to convert an electrical signal into audible sound. At its core, the buzzer serves as a bridge between a machine's internal logic and human perception, providing immediate feedback that something requires attention. Whether it is the soft "beep" of a microwave finishing its cycle, the sharp "chirp" of a smoke detector, or the rhythmic pulsing of a truck reversing, buzzers are the voice of our hardware. They are primarily categorized into two types: electromagnetic and piezoelectric. Electromagnetic buzzers operate via an internal solenoid and a vibrating disk, while piezoelectric buzzers utilize the unique properties of ceramic materials that deform when subjected to an electric field. This deformation creates pressure waves in the air, resulting in a clear, high-pitched tone.

Figure 8: Buzzer

PyCharm:

PyCharm provides a robust ecosystem tailored for Python development, which is critical for managing the high-computational demands of real-time image processing. In this project, PyCharm acts as the central orchestrator, managing the integration of the OpenCV library for frame acquisition and facial landmark detection, alongside NumPy and SciPy for performing the mathematical calculations involved in the Eye Aspect Ratio (EAR) and Mouth Aspect Ratio (MAR) algorithms. The choice of PyCharm for this research is justified by its advanced debugging tools and dedicated virtual environment management. These features ensure that the specific versions of dependencies—essential for the consistent performance of the AI model—remain isolated and stable. Furthermore, PyCharm's seamless integration with serial communication libraries (such as pyserial) allows for the real-time transmission of trigger signals to the Arduino Nano. When the software detects a drowsiness threshold violation, PyCharm executes the logic that sends a command byte over the USB serial port to the microcontroller, which subsequently activates the relay for auto-braking, the vibration motor, and the buzzer.

NumPy:

NumPy (Numerical Python) acts as the fundamental computational engine required to handle the massive arrays of data generated during real-time image processing. While OpenCV captures the video frames, NumPy is utilized to transform these frames into multi-dimensional arrays (tensors), allowing for efficient mathematical operations on pixel intensities. Specifically, in this project, NumPy is critical for calculating the Ear Aspect Ratio (EAR), a geometric metric used to determine if a driver's eyes are closed. NumPy's ability to handle temporal sliding windows is vital for the "Real-Time" aspect of the system. To avoid false positives caused by natural blinking, the system stores a sequence of EAR values in a NumPy array and calculates the mean or rolling average to distinguish between a blink and a prolonged eye closure (drowsiness). Once a threshold violation is detected, the processed data is converted into a format compatible with the serial communication protocol for the Arduino Nano. This ensures that the hardware components including the buzzer for alerts, the vibration motor for tactile feedback, and the relay for the DC motor's auto-braking are triggered with minimal latency.

Scipy:

SciPy's most critical contribution is the implementation of the Euclidean Distance formula via its spatial module. To detect drowsiness, the system identifies specific facial landmarks specifically the corners and lids of the eyes. SciPy is then used to calculate the Eye Aspect Ratio (EAR). By using scipy. spatial. distance. Euclidean, the system can compute these distances between vertical and horizontal eye coordinates with extreme precision and minimal computational overhead. This is vital for real-time applications where latency can result in an accident. When the EAR value drops below a statistically significant threshold for a specific number of frames, SciPy's logic triggers the alert state.

Open cv:

OpenCV acts as the primary computational engine for high-speed image processing and facial feature extraction. Operating within a Python environment, the library facilitates the capture of real-time video streams from the vehicle's internal camera, processing each frame at a high temporal resolution to ensure zero-latency detection. The workflow begins with Grayscale Conversion to reduce computational load, followed by the implementation of the Haar Cascade

Classifier or Dlib's frontal face detector to isolate the driver's facial region from the background noise of the vehicle cabin. Once the face is localized, OpenCV maps specific facial landmarks—primarily the coordinates corresponding to the eyelids and pupils. OpenCV monitors the EAR across successive frames. When the EAR falls below a predefined threshold for a specific number of frames (indicating a blink duration exceeding the physiological norm for an awake driver), OpenCV identifies a drowsiness event.

II. CONCLUSION

The development and implementation of the AI-Powered Real-Time Drowsiness Detection and Auto Braking System represent a significant advancement in the field of Intelligent Transportation Systems (ITS). This research successfully integrated high-level computational libraries, including OpenCV, NumPy, and SciPy, with the low-latency processing capabilities of the Arduino Nano to create a robust fail-safe mechanism for modern vehicles. The software architecture, centered on the Eye Aspect Ratio (EAR) algorithm, demonstrated a high degree of accuracy in distinguishing between natural ocular movements and signs of fatigue. By utilizing SciPy for Euclidean distance calculations and NumPy for optimized array processing, the system achieved a real-time frame rate capable of identifying micro-sleep events within milliseconds, a critical threshold for high-speed vehicular safety.

The hardware integration proved equally vital, transforming digital detection into physical intervention. The Arduino Nano served as the central nervous system, coordinating a multi-staged alert and response protocol. Upon detecting a breach in the EAR threshold, the system initiated an immediate sensory warning via a high-frequency Buzzer and a high-intensity LED, followed by tactile feedback from a Vibration Motor to alert the driver's physical senses. Most importantly, the transition from detection to mitigation was achieved through a Relay-controlled DC Motor circuit, simulating an Auto-Braking maneuver. This automated intervention ensures that even in cases of total driver incapacitation, the vehicle can reduce kinetic energy and mitigate the severity of potential collisions.

REFRENCES

[1] T. Soukupová and J. Čech, "Real-Time Eye Blink Detection using Facial Landmarks," in Proc. 21st Computer Vision Winter Workshop (CVWW), Feb. 2016.

[2] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1867–1874.

[3] S. Al-Waisy et al., "A multi-level fusion-based drowsiness detection system using deep learning," IEEE Access, vol. 9, pp. 110214–110232, 2021.

[4] . M. S. Wibowo and R. Ritzkal, "Automation of vehicle braking systems with ultrasonic sensors and Arduino," Journal of Robotics and Control (JRC), vol. 2, no. 4, pp. 312–317, 2021.

[5] B. R. S. Reddy and K. S. Rao, "Driver fatigue detection and alert system using vibration and sound feedback," International Journal of Engineering and Advanced Technology, vol. 10, no. 1, pp. 158–163, 2020.

[6] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library. Sebastopol, CA: O'Reilly Media, 2008.

[7] K. K. Sharma and S. Bhargava, "IoT and AI-based intelligent braking system for smart vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 3, pp. 2890–2901, 2023.