IJARSCT

International Journal of Advanced Research in Science, Communication and Technology

International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal

ISSN: 2581-9429

Volume 5, Issue 2, November 2025

Impact Factor: 7.67

Voice Automated Smart Headgear

Dr. Suma[1], Ananya P S[2], Meghashree C Shekhar[3], Karthik Matale[4], Ramesh R[5]

Professor, HOD, Department of ECE[1]

Students, Department of ECE[2-5]

Vidya Vikas Institute of Engineering and Technology, Mysore

sumavvietec@gmail.com, ananyaps58@gmail.com, meghashreecs98@gmail.com

karthiksmatale@gmail.com, rameshrnayak31@gmail.com

Abstract: *A smart headgear is an advanced piece of headgear designed to enhance safety and functionality for various activities, including motorcycling, cycling, and industrial work. Utilizing modern technology, smart helmets integrate features such as communication systems, sensors, and health monitoring to provide a comprehensive safety solution. The core of a smart helmet typically includes a micro-controller connected to various sensors, such as accelerometer, gyroscopes, and heart rate monitors. Our project focuses on reducing the manual interruptions for a safe ride by installing voice automation for visor movement in the rider's headgear as well as send an SOS or emergency call on detecting an accident through fall detection sensor this will include components such as AI voice modulator, ESP 32, stepper motor, microphone GSM module & Bluetooth chip in industrial applications, smart headgear can enhance worker safety by monitoring environmental conditions, such as toxic gas levels, and alerting wearers to potential hazards.*

Keywords: Smart helmet, safety, voice automation, accident detection, IoT, wearable technology, rider safety, industrial safety, sensors, AI, ESP32

I. INTRODUCTION

A smart headgear integrates advanced technology and traditional safety features to enhance user experience and safety. Equipped with sensors, communication system, and another electronics component.

These helmets are designed not only to provide the traditional protective functions but also to incorporate features such as:

1. Safety enhancement: smart helmet often includes sensors that detect impacts, falls, and abnormal movements, sending alerts in case of accidents. Some models have integrated cameras for recording rides or monitoring surroundings

2. Communication: equipped with Bluetooth connectivity, smart helmets enable emergency phone calls. They may also have built-in speakers and microphones for ease of use

3. The hand free movement of the visor eliminates the manual intervention for the movement of visor due to dust or wind via voice command Our project is inspired by these advancements have come up with an idea to implement voice automation in helmets for better safety and less manual involvement in the head gear, therefore, controlling the movement of the visor in the helmet via voice command and also by controlling basic application such as emergency or SOS call through GSM module Bluetooth chip as well as AI voice module.

Voice-automated smart helmets represent a significant leap forward in personal protective equipment, merging advanced technology with the fundamental need for safety. Traditionally, helmets have served as passive protective gear, offering physical barriers against impact. However, the modern era demands more than just passive protection; it calls for active safety features that enhance awareness, improve communication, and provide instant access to crucial information, all without compromising the user's focus on their primary task. This is precisely where voice automation transforms the conventional helmet into an intelligent, interactive device.

The concept of a "smart" helmet has been evolving for some time, incorporating features like integrated cameras, GPS navigation, and Bluetooth connectivity for communication. Yet, the true potential of these features has often been hampered by cumbersome controls. Operating small buttons or touchscreens while engaged in activities like riding a motorcycle, cycling, or working in hazardous industrial environments can be distracting and, in some cases, even dangerous.

The need for hands-free operation became increasingly apparent, paving the way for voice automation to become the central pillar of next-generation helmet design.

Voice automation allows users to control various functionalities of the helmet through simple verbal commands. This hands-free interaction minimizes distractions, enabling the user to maintain full concentration on their surroundings and tasks.

Imagine a motorcyclist navigating unfamiliar roads; instead of fumbling with a phone or a separate GPS device, they can simply say, "Navigate to the nearest gas station," and the directions are seamlessly relayed through the helmet's integrated audio system. Similarly, a construction worker could inquire about a blueprint detail or a safety protocol without removing their gloves or shifting their focus from heavy machinery.

Beyond mere convenience, the integration of voice automation significantly enhances safety. By eliminating the need for manual interaction with controls, the risk of accidents caused by distraction is drastically reduced. Furthermore, voice- activated features can include emergency response systems, where a simple voice command, or even an automatic trigger based on impact detection, can alert emergency services with the user's location. This immediate and effortless communication can be life-saving in critical situations.

The applications of voice-automated smart helmets are incredibly diverse, spanning across various sectors. For motorcyclists and cyclists, these helmets offer integrated navigation, music control, communication with fellow riders, and even real-time weather updates, all accessible through voice.

In industrial settings, they can provide instant access to technical manuals, safety checklists, and communication with remote teams, enhancing efficiency and adherence to safety regulations. First responders, firefighters, and law enforcement personnel can benefit immensely from hands-free communication, tactical information access, and the ability to record incidents without breaking their operational stride. Even in recreational activities like skiing or mountain biking, these helmets can provide turn-by-turn directions, track performance metrics, and allow seamless communication within a group.

The underlying technology for these helmets typically involves miniature microphones with advanced noise cancellation to accurately capture voice commands even in noisy environments, sophisticated speech recognition software, and integrated processors to handle the commands and execute the desired functions. Powering these features requires efficient battery management systems, designed to last for extended periods.

In conclusion, voice-automated smart helmets represent a paradigm shift in personal protective equipment. They move beyond passive protection to offer an active, intelligent, and interconnected safety solution. By integrating intuitive voice control, these helmets not only enhance user convenience and productivity but, most importantly, significantly elevate safety standards across a multitude of applications. As technology continues to advance, we can anticipate even more sophisticated features and widespread adoption of these innovative helmets, making our world a safer and more connected place.

II. OBJECTIVES:

The primary objectives of this project are:

1. 100 % hand free ride without any human intervention in the head gear.
2. Voice automation command for emergency call.
3. Fall detection sensor when triggers make an emergency or SOS call without any voice command in case the rider gets unconscious right after the accident.
4. Voice automated command for head lights.
5. Voice automated left & right indicator ON & OFF function.

III. METHODOLOGY

The development of a voice-automated smart helmet hinges on a meticulous and iterative methodology, ensuring a product that is not only technologically advanced but also safe, reliable, and user-centric. The process begins with thorough requirements gathering and analysis, where extensive market research, user surveys, and stakeholder interviews are conducted to understand the diverse needs of target users— from motorcyclists to industrial workers. This initial phase also involves a crucial feasibility study and a comprehensive regulatory compliance review to define precise technical specifications and adhere to safety standards like DOT, ECE, or ANSI.

Once requirements are clear, the next step is system architecture design, a critical stage where hardware components like high-fidelity microphones with advanced noise cancellation, powerful microcontrollers, various communication modules (Bluetooth, Wi-Fi, 4G/5G), and robust battery management systems are meticulously chosen. Simultaneously, the software stack—including the operating system, speech recognition engine, and natural language processing (NLP) components—is carefully planned, with a strong emphasis on safety integration for features like emergency alerts.

The methodology then moves into iterative prototype development, starting with individual component testing and progressing to the creation of a Minimum Viable Product (MVP) that integrates core voice automation functionalities. This stage involves the careful integration of electronic components into a standard helmet shell, considering ergonomics and weight distribution, often utilizing custom 3D- printed solutions. Crucially, software development and integration run in parallel, focusing on training the speech recognition model for diverse environmental conditions, developing robust NLP algorithms, and implementing all defined features such as navigation, communication, and emergency systems. Each prototype undergoes rigorous internal testing, leading to continuous refinement and optimization.

The culmination of the development process is comprehensive testing and validation, which includes bench testing of individual components, controlled environment testing, and extensive field testing. User Acceptance Testing (UAT) with real-world users is paramount, providing invaluable feedback on usability and performance in actual scenarios. This phase also includes rigorous performance testing for metrics like voice command accuracy and battery life, along with durability and environmental testing to ensure the helmet withstands extreme conditions. Finally, safety compliance testing by certified laboratories verifies adherence to all relevant standards, ensuring the helmet's readiness for eventual manufacturing and deployment.

IV. HARDWARE REQUIREMENTS

1. ESP32 microcontroller: A versatile, low-power microcontroller with integrated Wi-Fi and Bluetooth, serving as the brain of the smart helmet for processing commands and controlling peripherals.
2. GSM module: Enables cellular connectivity for the helmet, allowing for long-range communication, emergency calls, and data transmission without relying on Wi-Fi.
3. Servo gear/Motor: Provides precise rotational movement, potentially used for automated visor control, adjusting integrated components, or haptic feedback mechanisms within the helmet.
4. Accelerometer MPU 6040/6050: Detects motion, orientation, and impact, crucial for features like crash detection and automatic emergency alerts.
5. Relay: An electrically operated switch used to control higher-power circuits or devices within the helmet system, such as turning on/off external lights or other motors.
6. Head gear / Helmet: The foundational protective shell that houses all the electronic components and provides physical safety to the wearer.

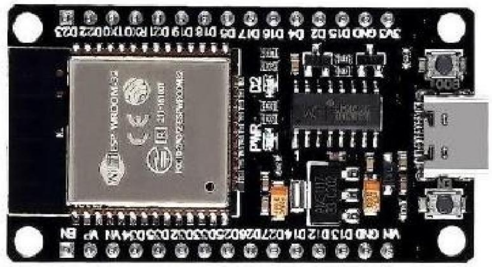
V. SOFTWARE REQUIREMENTS:

1. Visual Studio Code: A lightweight, yet powerful and highly customizable code editor used for developing software, including embedded systems and web applications, offering extensive extensions for various programming languages and platforms.
2. Arduino IDE: A simple, cross-platform integrated development environment specifically designed for writing, compiling, and uploading code to Arduino- compatible microcontrollers.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 5, Issue 2, November 2025 Impact Factor: 7.67

Hardware Requirements:

a. ESP 32 microcontroller:



ESP32 is a family of low-cost, energy-efficient microcontrollers that integrate both Wi- Fi and Bluetoothcapabilities. These chips feature a variety of processing options, including the Tensilica Xtensa LX6 microprocessor available in both dual-core and singlecore variants, the Xtensa LX7 dual-core processor, or a single-core RISC-V microprocessor. In addition, the ESP32 incorporates components essential for wireless data communication such as built-in antenna switches, an RF balun, power amplifiers, low- noise receivers, filters, and power- management modules.

Typically, the ESP32 is embedded on device-specific printed circuit boards or offered as part of development kits that include a variety of GPIO pins and connectors, with configurations varying by model and manufacturer. The ESP32 was designed by Espressif Systems and is manufactured by TSMC using their 40 nm process. It is a successor to the ESP8266 micro- controller.

b. GSM Module:

A GSM modem is a specialized type wireless modem that works with a GSM wireless network uses TDMA technology for technology for transmitting signals. It accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone. Mixture Models (GMM) or deep learning- based methods can be used.

A GSM modem can be an external device or a PC Card / PCMCIA Card. An external GSM modem is connected to a computer through a serial cable or a USB cable. When a GSM modem is connected to a computer, this allows the computer to communicate over the mobile network. While these GSM modems are most frequently used to provide mobile internet connectivity, many of them can also be used for sending and receiving SMS and MMS message.

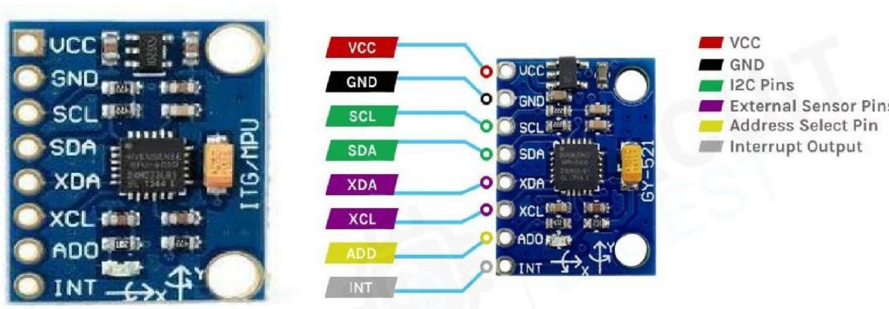
GSM Modem sends and receives data through radio waves.

c. Servo gear/Motor:

A servo gear, often referred to as a servo gearbox, is a precision-engineered gear system designed to work with servo motors. It's used to increase torque output, reduce load on the servo motor, and enhance overall motion control system performance. Servo gearboxes are commonly used in applications requiring high precision, speed, and dynamic performance.

d. Accelerometer MPU 6040/6050:

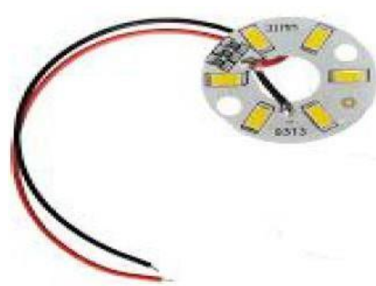
The MPU-6050 is a widely used 6-axis motion tracking sensor developed by Invent Sense (now part of TDK), combining a 3- axis accelerometer and a 3-axis gyroscope on a single chip. It's popular in applications like robotics, drones, wearable, and motion tracking systems.The MPU- 6050 is a 6-axis(combines 3-axis Gyroscope, 3-axis Accelerometer) motion tracking devices. Changes in motion, acceleration and rotation can be detected.It is commonly used in robotics, gaming controllers, and other electronic devices that require motion detection.



e. Relay:

A relay is an electrically operated switch that controls a high- power circuit with a low-power signal. It's essentially a switch that's activated by an electrical current, allowing a small current to control a larger one. This is useful for situations where you need to switch a high-power device, like a motor or a light, with a low-power control signal, such as a button press.

f. SMD 5730 Brightness SMD Light board:



IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, November 2025
Impact Factor: 7.67

Its 5730 0.5 Watt WHITE high-quality SMD LED. These LED's consume very low power and emit much intense bright light, many times more than a normal LED, and generally produce a wide angle of light disbursement which can be narrowed through the use of an external lenses.

SOFTWARE DESCRIPTION:

The Voice-Automated Smart Helmet is an innovative, integrated safety and communication solution for riders, leveraging advanced software to provide a hands-free, intuitive, and enhanced riding experience. Its open-source compatible software core allows for future expandability and community-driven improvements.

• Software Core and Architecture

The smart helmet's software is built upon a robust embedded system architecture, primarily programmed using Embedded C/C++. This choice prioritizes low-level hardware interaction, memory efficiency, and real-time performance crucial for safety-critical applications. The core processing unit, often a microcontroller (e.g., Arduino-compatible ATmega series, ESP32, or Raspberry Pi for more complex systems), executes the helmet's functionalities.

The software architecture can be broadly categorized into several interconnected modules:

Voice Recognition & Processing Module: This is the heart of the voice automation, responsible for capturing audio input, processing it to identify predefined voice commands, and translating them into actionable signals. It employs sophisticated algorithms for noise cancellation to ensure accurate recognition even in noisy environments.

Sensor Data Acquisition & Processing Module: Manages input from various onboard sensors (e.g., accelerometer, gyroscope, GPS, alcohol sensor, IR proximity sensor, environmental sensors). It continuously monitors data for anomalies (e.g., sudden impacts, high alcohol levels, unusual head movements) and processes it for alerts, navigation, or operational control.

Communication Module: Handles wireless communication protocols such as Bluetooth for smartphone connectivity, Mesh Intercom for rider-to-rider communication, and potentially GSM/GPRS for accident alerts with location data.

Display & Feedback Module: Manages audio outputs (via integrated speakers for voice prompts, music, calls) and visual feedback (e.g., LED indicators for turn signals, battery status, or a heads-up display if present).

Control Logic Module: Implements the core "if-then" logic, taking processed sensor data and voice commands as inputs to trigger specific outputs (e.g., adjusting the visor, initiating a call, sending an alert, controlling vehicle ignition).

• Procedure to Initialize and Operate Smart Helmet Software

Step 1 − Hardware Prerequisites:

Ensure the smart helmet is charged or connected to a power source. Confirm all integrated sensors (e.g., voice microphone, accelerometer, GPS) and communication modules (e.g., Bluetooth) are functional.

Step 2 − Power Up the Helmet:

Activate the helmet's power switch. A confirmation audio cue or LED indicator will signal that the helmet's internal software has initiated.

Step 3 − Establish Smartphone Connection (Optional, for advanced features):

For features like GPS navigation, calls, or music streaming, pair the helmet with your smartphone via Bluetooth.

On your smartphone: Go to Bluetooth settings and select the smart helmet from the list of available devices.

On the helmet: The helmet may provide audio prompts or LED indicators to confirm successful pairing.

Step 4 − Voice Command Calibration (Initial Setup):

While the core voice recognition is pre-configured, some helmets may offer optional voice calibration for personalized accuracy. This usually involves speaking a series of prompts. Refer to the helmet's user manual for specific calibration procedures.

Step 5 − Begin Operation:

Once powered on and optionally connected, the helmet's software continuously monitors sensor data and listens for voice commands.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 2, November 2025
Impact Factor: 7.67

• Program Structure (Conceptual)

Similar to Arduino sketches, the smart helmet's embedded software adheres to a structured approach:
setup() function:
Executed once upon power-up or reset.
Initializes all hardware components (sensors, communication modules, audio/visual outputs).
Sets initial states for variables and pin modes.
Configures communication protocols (e.g., Bluetooth pairing modes).
Initializes voice recognition engine.
loop() function:
Continuously executes after setup() completes.
Input: Reads data from all sensors (accelerometer, GPS, environmental, etc.) and continuously listens for voice commands from the microphone.
Process Data: Analyzes sensor data for predefined conditions (e.g., accident detection, alcohol levels, helmet worn status). Processes voice input against a library of recognized commands.
Logic & Decision Making: Based on processed inputs and voice commands, the software executes specific actions. This includes:
Triggering accident alerts (with GPS coordinates) via GSM. Controlling helmet features (e.g., visor up/down, turn indicators) based on voice commands.
Managing audio playback and call handling.
Issuing warnings or preventing vehicle ignition (if integrated). Output: Provides feedback to the user through audio cues, spoken instructions (e.g., navigation prompts), or visual indicators.

• Programming Using Embedded C/C++

The primary language for developing the smart helmet's firmware is Embedded C/C++. It offers the low-level control and efficiency required for real-time operation on resource- constrained microcontrollers. Key aspects include:
Header Files (.h): Define the structure and declarations of functions and variables for various modules (e.g., voice_recognition.h, sensor_interface.h).
Source Files (.cpp or .c): Contain the actual implementation of the functions declared in the header files.
Libraries: Extensive use of specialized libraries for hardware interaction (e.g., sensor drivers, communication protocols like Bluetooth stacks, GPS parsing libraries, audio codecs, voice recognition libraries).
Interrupt Service Routines (ISRs): Used for time-critical responses to external events, such as a sudden impact detected by an accelerometer, ensuring immediate action.

• Arduino Code Libraries (for Prototyping/Simpler Implementations)

While a full-fledged smart helmet might use more powerful platforms, Arduino IDE and its libraries are excellent for prototyping and even simpler production models.
Library Structure: Similar to the Arduino IDE description, libraries consist of C++ files (.cpp) for implementation and header files (.h) for declarations.
Importing Libraries: Developers would use the Arduino IDE's "Sketch > Include Library > Manage Libraries" feature to add specialized libraries for voice recognition (e.g., Voice Recognition Module Library), GPS modules, Bluetooth modules, and sensor interfaces.

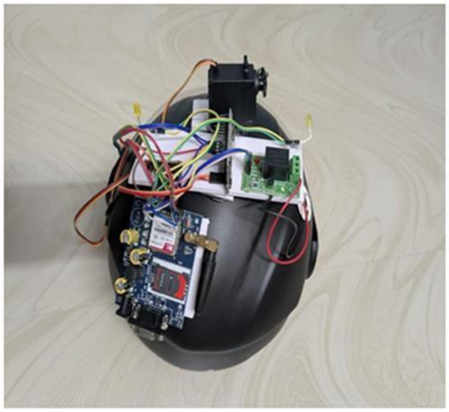
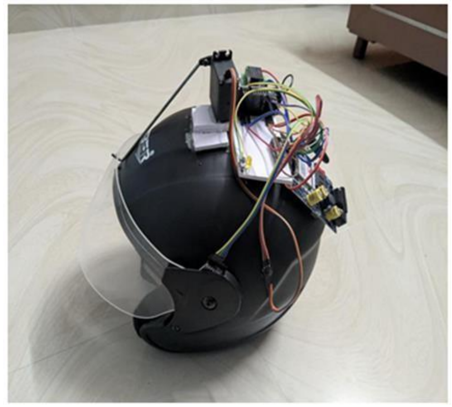
• Code Explanation (High-Level Concepts)

Libraries: Provide pre-written functions for common hardware interactions (e.g., reading from an accelerometer, sending data over Bluetooth) and complex algorithms (e.g., voice processing).
Pin Definitions: Use #define or const int to assign meaningful names to hardware pins, improving code readability (e.g., #define LED_PIN 13, #define MIC_INPUT_PIN A0).

Variables: Store data acquired from sensors (e.g., float currentSpeed;, int alcoholLevel;), processed information, and system states.
Instances: Create objects of classes defined in libraries to interact with specific hardware components or software services (e.g., BluetoothSerial bt;, GPS_Module gps;).
setup(): The mandatory initialization block.
loop(): The perpetual execution block containing the main logic for continuous monitoring, processing, and response.
Code Logic: Typically follows an input-process-output model, often incorporating if-then-else structures for decision-making based on sensor readings and voice commands.

- **From Software to Hardware: The Deployment Process Code → Compile → Upload → Run**

Code: Developers write the Embedded C/C++ code for the
helmet's functionalities in a suitable IDE (e.g., Arduino IDE, PlatformIO, or a more specialized embedded IDE like Keil for certain microcontrollers).
Compile: The IDE's compiler translates the human-readable code into machine-executable instructions (binary code). This step checks for syntax errors and ensures the code is valid.
Upload: The compiled binary is then transferred ("uploaded") to the helmet's microcontroller via a USB connection or a wireless programming interface.
Run: Once the upload is complete, the microcontroller starts executing the new program automatically, bringing the smart helmet's features to life.
Merits of Voice-Automated Smart Helmet Software
Hands-Free Operation: Maximizes rider safety by eliminating the need to take hands off handlebars for controls.
Enhanced Connectivity: Seamless integration with smartphones for calls, music, and navigation.
Real-Time Safety Alerts: Immediate notification of potential hazards (e.g., alcohol detection, accident).
Situational Awareness: Provides critical information (e.g., GPS directions, environmental data) directly to the rider.
Personalization: Adaptable through voice command customization and potential future software updates.
Modular and Scalable: Software architecture allows for adding new features and sensors.
Demerits of Voice-Automated Smart Helmet Software Voice Recognition Accuracy: Can be affected by external noise (wind, traffic), rider's accent, or speech impediments.
Battery Consumption: Advanced software features and continuous sensor monitoring can drain battery quickly.
Complexity: Developing and debugging complex embedded software requires specialized skills.
Latency: Slight delays in voice command processing or data transmission can occur.
Cybersecurity Risks: If connected to external networks, potential vulnerabilities to hacking or data breaches (though less critical for basic helmet functions).
User Adoption: Riders may need time to adapt to new voice control paradigms.
Applications of Voice-Automated Smart Helmet Software Motorcycle and Cycling Safety: Core application for enhanced rider protection and communication.
Industrial Safety: Helmets for construction, mining, or hazardous environments with voice-activated alerts, communication, and environmental monitoring.
Emergency Services: First responder helmets with integrated communication, navigation, and vital sign monitoring.
Adventure Sports: Helmets for skiing, climbing, or other sports offering communication, GPS tracking, and safety features.
Logistics and Delivery: Hands-free communication and navigation for delivery personnel.

VI. CONCLUSION

This project aims at reducing the hand movement by giving currency in return of plastic wastes. Today the plastic waste is increasing and is non-biodegradable. So this project aims at reducing the plastic waste by installing such vending machines at public places where people can dispose into such machines in exchange of coins. The capacity of the machine can also be increased by various process such as increasing the container volume , increasing the number of intake ports .

SCOPE FOR FUTURE WORK

1. AI Integration: Artificial Intelligence will enable helmets to learn rider preferences, predict needs, and offer personalized suggestions, enhancing the overall riding experience.
2. Augmented Reality (AR) Enhancements: AR can overlay critical information, such as speed and navigation, directly onto the rider's field of view, improving situational awareness.
3. Health Monitoring: Incorporating sensors to monitor vital signs like heart rate and fatigue levels can alert riders to potential health issues, promoting safety.
4. Extended Connectivity: Integration with smart vehicles and IoT devices will allow helmets to communicate with surrounding infrastructure, providing real time data and alerts.
5. Sustainability Features: Solar-powered charging and energy- efficient components will make helmets more eco-friendly and reduce dependency on external power sources.

REFERENCES

[1] C Impana, M Hamsaveni and H T Chethana, A Review on smart helmet for accident detection using IOT, Vol-5, 30 OCTOBER 2019.

[2] Agung Pangsetu, M N Mohammed, Al-Zubaidi, Internet of things toward a novel smart helmet for motorcycle: review, Vol 1, 2 march 2021.

[3] Tarek Eldemerdash, Raed Abdulla, Vikneswary Jayapal, Chandrshekharan Natraj, Maythem K.Abbas, IOT Based smart helmet for Mining industry application,Vol 29, 2020

[4] Natraja N, Mamatha K S, DR Keshavamurthy, DR Shivashankar, SMART HELMET, 18 may 2018

[5] Mohammed Ehsanul Alim, Sarosh Ahmed, Marzieh Naghdi Dorabati, Ihab Hassoun, Design& Implimentation of IOT Based Smart Helmet for Road Accident Detection, 18 M

[6] Prashanna Rangan1 , M. Sangameshwaran2 , V. Poovendan3 , G. Pavanpranesh4 and C. Naveen, Voice Controlled Smart Helmet Nov 2018

[7] MSamrajyam1 , U Dhanasri2 , L Guruvardini3 , V Sharon Angel 4 , CH Yamini Priya, Automatic Smart Helmet Using IoT, April 2023

[8] D.Pravin Kumar, K.M.Saran, K.R.Surya, V.Uthamaseelan, S.Vishwa Karthik, Smart Helmet With Voice Control Using IoT, October 2024

[9] Santhosh Rao, Voice Controled Wiper for Smart Helmet Using IoT, october 2019