

**AUTO SENTINEL: SMART BRAKING AND DRIVER
AWARENESS SYSTEM**



By

Syed Shabab Haider

01-132222-045

Muzamil Hussain

01-132222-034

Usman Hidayat

01-132222-048

Supervised By

Dr. Waleed Manzoor

**Department of Computer Engineering
Bahria University Islamabad**

2026

AUTO SENTINEL: SMART BRAKING AND DRIVER AWARENESS SYSTEM

By

Syed Shabab Haider

01-132222-045

Muzamil Hussain

01-132222-034

Usman Hidayat

01-132222-048



**Supervised By
Dr. Waleed Manzoor**

Submitted to the Department of Computer Engineering in the partial
fulfillment of the requirements for the degree of Bachelors in
Computer Engineering.

**Department of Computer Engineering
Bahria University Islamabad**

2026



BAHIRA UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

Dated: 04-06-2026

CERTIFICATE

We accept the work contained in the report titled “**AutoSentinal: Smart braking and driver awareness system**” as a confirmation to the required standard for the partial fulfillment the degree of Bachelor of Computer Engineering.



Dr. Waleed Manzoor
Project Coordinator
Date: 04-06-2026



Dr. Waleed Manzoor
Supervisor
Date: 04-06.2026



Dr. Ali Arshad Uppal
External Examiner
Date: 04-06.2026



Dr. Syed Khawar Hussain Shah
Internal Examiner
Date: 04-06.2026



Head of Department (CE)
Date: 04-06-2026

AUTO SENTINEL: SMART BRAKING AND DRIVER AWARENESS SYSTEM

SDG No	Description of SDG	SDG No	Description of SDG
SDG 1	No Poverty	SDG 9	Industry, Innovation and Infrastructure
SDG 1	Zero Hunger	SDG 10	Reduced Inequalities
SDG 3	Good Health and Well-Being	SDG 11	Sustainable Cities and Communities
SDG 4	Quality Education	SDG 12	Responsible Consumption and Production
SDG 5	Gender Equality	SDG 13	Climate Action
SDG 6	Clean Water and Sanitation	SDG 14	Life Below Water
SDG 7	Clean Water and Sanitation	SDG 15	Life on Land
SDG 8	Decent Work and Economic Growth	SDG 16	Peace, Justice and Strong Institutions
		SDG 17	Partnerships for the Goals



Range of Complex Problem Solving			
	Attribute	Complex Problem	
1	Range of conflicting requirements	Involve wide-ranging or conflicting technical, engineering and other issues.	✓
2	Depth of analysis required	Have no obvious solution and require abstract thinking, originality in analysis to formulate suitable models.	✓
3	Depth of knowledge required	Requires research-based knowledge much of which is at, or informed by, the forefront of the professional discipline and which allows a fundamentals-based, first principles analytical approach.	✓
4	Familiarity of issues	Involve infrequently encountered issues	
5	Extent of applicable codes	Are outside problems encompassed by standards and codes of practice for professional engineering.	
6	Extent of stakeholder involvement and level of conflicting requirements	Involve diverse groups of stakeholders with widely varying needs.	✓
7	Consequences	Have significant consequences in a range of contexts.	
8	Interdependence	Are high level problems including many component parts or sub-problems	

Range of Complex Problem Activities			
	Attribute	Complex Activities	
1	Range of resources	Involve the use of diverse resources (and for this purpose, resources include people, money, equipment, materials, information and technologies).	✓
2	Level of interaction	Require resolution of significant problems arising from interactions between wide ranging and conflicting technical, engineering or other issues.	✓
3	Innovation	Involve creative use of engineering principles and research-based knowledge in novel ways.	✓
4	Consequences to society and the environment	Have significant consequences in a range of contexts, characterized by difficulty of prediction and mitigation.	
5	Familiarity	Can extend beyond previous experiences by applying principles-based approaches.	

UNDERTAKING

Use the following undertaking as it is. I certify that research work titled "Auto Sentinel: Smart braking and driver awareness system" is my own work. The work has not been presented elsewhere for assessment. Where material has been used from other sources it has been properly acknowledged / referred



Syed Shabab Haider

01-132222-045



Muzamil Hussain

01-132222-034



Usman Hidayat

01-132222-048

DEDICATION

This work is dedicated to my beloved parents and family, whose unwavering support, patience, and sacrifices have been the foundation of my journey. Their constant encouragement, understanding, and belief in my abilities have given me the strength to overcome every challenge and move forward with confidence. Without their guidance and motivation, this achievement would not have been possible. I would also like to dedicate this work to all those individuals and organizations striving to make our roads safer. To the engineers, researchers, policymakers, and innovators who continuously work toward improving transportation systems and reducing accidents, your efforts inspire this project. This dedication extends to the drivers, passengers, and pedestrians whose safety remains a shared responsibility. It is my hope that this work contributes, even in a small way, toward creating a safer and more secure environment for everyone on the road.

ACKNOWLEDGMENTS

I am highly thankful to my well-respected supervisor, Dr. Waleed Manzoor for his constant guidance and valuable feedback throughout this duration of project. I am also thankful to him for his eternal inspiration, wisdom and assistance in completing this task with better standards. Since a wiser mind has said that knowledge speaks but wisdom listens, he has efficient commands over knowledge which he provided me with to succeed this dissertation. I also want to give my big thankfulness to all my team members for their cooperation, teamwork and contributions. Thanks for all ideas, help and support from my team members. It really makes our work more productive and enjoy. Last but not least, I would like to express my sincere thanks to my family for their unconditional support and encouragement throughout. I feel very fortunate to have such a wonderful family in my life.

ABSTRACT

Road accidents continue to be a major problem across the globe with overspeeding and driver fatigue being some of the major reasons for fatal accidents. Vehicle safety systems at present are such that each system focuses on one risk factor like speed control or driver state monitoring and does not handle multiple risk factors together using one single system. There is a need for a new system which can handle multiple risk factors together and act in a real time mode without any intervention from the driver.

The Auto Sentinel aims to bridge this gap by integrating a driver awareness system with a brake reinforcement system. It comprises three key components, namely, a GPS module to constantly update the vehicle's current speed, a vision detection system which is camera based and lights and sounds to provide warnings and alerts. The prototype developed on a Raspberry Pi platform has a camera using real-time facial behavior detection using YOLOv8, a NEO-6M GPS component that measures the vehicle's speed, and the brakes is a servo hub-based actuator. The system runs continuously taking in the speed as well as driver state information in to make on-the-fly decisions without human intervention. Cost effective, modular design is suitable to be validated in the laboratory using low speed studies.

Planned additional features involve auto, real-time connection to the vehicle's ECUs for direct application of the brakes and throttle, cloud upload and access for logging events and accidents from anywhere, and experimentation with more sophisticated AI models for more detailed classification of driver behavior. This Auto Sentinel prototype lays the groundwork for intelligent, accident-resilient transportation.

Keywords:

Auto Sentinel, Smart Braking System, Driver Fatigue Detection, Overspeed Monitoring, YOLOv8, Raspberry Pi, Computer Vision, Accident Prevention.

TABLE OF CONTENTS

Undertaking.....	i
Dedication	ii
Acknowledgments.....	iii
Abstract.....	iv
List of Figures	x
List of Tables	xi
List of Equations	xii
 Chapter 1: Introduction	 1
1.1 Background	1
1.2 Motivation.....	2
1.2.1 Why Drowsiness and Speed Are So Dangerous	2
1.2.2 The Cost Problem.....	2
1.2.3 The Gap Between Lab Research and Real Vehicles.....	3
1.3 Problem Statement	3
1.4 How the Project Evolved: From Car to Motorcycle.....	4
1.5 What Auto Sentinel Does.....	4
1.6 What This Project Contributes	5
1.6.1 Proving That Low-Cost Hardware Is Enough	5
1.6.2 A Simple, Safe Way to Slow a Motorcycle Down.....	6
1.6.3 Two Risk Factors Working Together	6
1.6.4 Software That Can Be Adapted for Cars Too.....	6
1.7 Project Objectives	6
1.7.1 Sensing and Detection	6
1.7.2 Decision Making and Control.....	7
1.7.3 Testing and Approval	7
1.8 How the System Is Structured.....	7
1.9 Intervention Policy	8
1.10 What Makes This Work Different.....	9
1.11 Pakistan's Road Safety Context.....	9
1.11.1 How Big Is the Problem?.....	9
1.11.2 Why Retrofit Is the Only Realistic Option	10
1.11.3 Where It Could Be Used First.....	10
1.12 Scope and Limits.....	10
1.13 Structure of This Thesis	11
 Chapter 2: Literature Review.....	 12
2.1 Introduction to the Literature Review	12
2.2 The Road Safety Problem and Human Error.....	12
2.2.1 The Global Scale of Road Traffic Deaths.....	12
2.2.2 Human Error as the Cause of the Event.....	13
2.3 Automatic Emergency Braking: Technology and Limitations	14
2.3.1 System Architecture and Sensing Modalities	14

2.3.2 Regulatory Mandates and Real-World Effectiveness	14
2.4 Speed Management and Enforcement Technology	15
2.4.1 GPS-Based Speed Awareness and Zone Enforcement	15
2.4.2 Camera-Based Speed Sign Recognition	16
2.4.3 Throttle Limiting Without ECU Access	16
2.5 Driver Monitoring Systems for Drowsiness Detection.....	16
2.5.1 Transition from Rule-Based Approaches to Deep Learning.....	16
2.5.2 MediaPipe Face Mesh for Embedded DMS	17
2.5.3 CNN and Transformer Architectures for Advanced DMS.....	17
2.5.4 Near-Infrared Illumination for All-Hours Operation	18
2.5.5 Benchmark Datasets and Their Limitations.....	18
2.6 Actuators for Vehicle Safety Interventions	19
2.6.1 Linear Actuators for Brake Pedal Intervention.....	19
2.6.2 Brake-by-Wire Technology	20
2.6.3 PWM Servo Control and Timing Precision	20
2.7 ECU Access and Proprietary Software Barriers.....	20
2.8 Control Theory and Safety	21
2.8.1 PID Control for Servo Actuation	21
2.8.2 Multi-Layer Watchdog Safety Architecture	22
2.9 Human Factors, Driver Acceptance, and Privacy	22
2.9.1 What Makes a Safety System Acceptable to Drivers.....	22
2.9.2 Biometric Data Privacy in Camera-Based DMS.....	23
2.10 Comparison of Existing Approaches.....	23
2.11 Design Implications for Auto Sentinel	24
2.12 Chapter Summary	24
Chapter 3: Methodology and Design	26
3.1 Introduction and Chapter Overview	26
3.2 Engineering Objectives and Design Principles	26
3.2.1 Primary Design Objective.....	26
3.2.2 Core Engineering Design Principles.....	26
3.3 High-Level System Architecture.....	27
3.3.1 Layered Architecture Overview.....	27
3.3.2 Dual Platform Architecture.....	28
3.3.3 Communication and Data Flow	29
3.4 Hardware Design: Sensor, Compute, and Actuation Specifications.....	30
3.4.1 Sensing Layer Hardware	30
3.4.2 Compute Layer Hardware.....	30
3.4.3 Actuation Layer Hardware	31
3.5 Software Architecture and Perception Pipeline.....	32
3.6 Multi-Modal Fusion Engine and Decision Logic.....	33
3.6.1 Unified Risk Score.....	33
3.6.2 Decision Policy with Hysteresis	34
3.6.3 Safety Pre-Conditions for Physical Intervention.....	34

3.6.4 Full Design Extension: Adding Collision	35
3.7 Actuation Control Design.....	35
3.7.1 Servo Motor Control: PWM and Position Definition	35
3.7.2 PID Control Architecture.....	36
3.7.3 Safety Interlocks and Fault Modes.....	36
3.8 Data Logging, Time Synchronization, and Privacy	37
3.8.1 Multi-Modal Logging Architecture	37
3.8.2 Time Synchronization.....	38
3.8.3 Privacy Architecture	38
3.9 Evaluation Metrics and Acceptance Criteria	38
3.9.1 DMS Performance Metrics	38
3.9.2 System Latency Metrics	39
3.10 Safety Case, Risk Assessment, and Mitigation	37
3.11 Implementation Roadmap and Deliverables.....	40
3.11.1 Staged Implementation Plan	40
3.11.2 Deliverables	41
3.12 Safety Case, Risk Assessment, and Mitigation	41
3.13 Implementation Roadmap and Deliverables.....	41
Chapter 4: Implementation.....	43
4.1 Introduction of Chapter.....	43
4.2 Platform Selection: Why a Motorcycle Instead of a Car	43
4.2.1 The Original Plan Was a Car	43
4.2.2 Why the Motorcycle Was the Right Choice	43
4.2.3 Platform Comparison.....	44
4.3 Hardware Overview	44
4.4 Central Computer: Raspberry Pi 4	45
4.5 Wheel Speed Sensing.....	46
4.5.1 Sensor Choice.....	46
4.5.2 How It Works	46
4.6 GPS and Speed Zone Enforcement	47
4.6.1 GPS Module	47
4.6.2 Geolinker Speed Zone Enforcement	47
4.7 Driver Monitoring System	47
4.7.1 Camera Setup	47
4.7.2 Drowsiness Detection: How It Works.....	48
4.8 Throttle Intervention System.....	48
4.8.1 How the Throttle Cable Interception Works.....	48
4.8.2 Servo Motor and Linkage	49
4.8.3 Servo Control.....	49
4.8.4 Safety Features	49
4.9 Risk Fusion Engine	50
4.9.1 Shared State.....	50
4.9.2 Computing the Risk Score	50

4.9.3 Intervention Levels	50
4.10 Software Architecture	51
4.10.1 Companion Mobile Application	51
4.11 Power Supply	52
4.12 Implementation Challenges and How They Were Solved.....	53
4.12.1 Electromagnetic Interference from the Ignition System	53
4.12.2 Servo Jitter from Software PWM	53
4.12.3 MediaPipe Running Too Slowly	54
4.12.4 Throttle Cable Linkage Geometry	54
4.13 Preliminary Validation Results	54
4.14 Known Limitations.....	55
4.15 Chapter Summary	56
Chapter 5: Result and Discussion	57
5.1 Introduction and Chapter Overview	57
5.2 Hardware Assembly Results and Physical Prototype Documentation.....	57
5.2.1 Bench Assembly: Raspberry Pi and Sensor Integration.....	57
5.2.2 Vehicle-Mounted Hardware Integration	58
5.2.3 Servo Motor and Throttle Cable Connection.....	59
5.2.4 Camera Mounting on Handlebar.....	60
5.3 Wheel Speed Sensing System: Results and Analysis	60
5.3.1 Initial Calibration Results	60
5.3.2 Speed Sensing Outdoor Performance on Vehicle	61
5.3.3 Speed Threshold Trigger Verification	62
5.4 GPS Module and Geolinker Live Tracking: Results and Discussion.....	62
5.4.1 GPS Fix Acquisition and Stability.....	62
5.4.2 Geolinker Live Tracking Visualization.....	63
5.4.3 Speed Monitoring via GPS Cross-Validation	63
5.5 Driver Monitoring System: Results and Discussion.....	64
5.5.1 MediaPipe Face Mesh Detection on Raspberry Pi Performance	64
5.5.2 EYE Measurement Validation Indoor Bench Test.....	64
5.5.3 Performance of Outdoor DMS on the Vehicle.....	65
5.6 Servo Motor Throttle Intervention: Results and Discussion.....	66
5.6.1 Bench Calibration Results	66
5.6.2 On-Vehicle Throttle Intervention Testing.....	66
5.7 Companion Application Live Firebase Data Display	67
5.8 Integrated System Results: Three Test Scenarios.....	68
5.8.1 Scenario 1: Drowsiness Detection and Servo Activation.....	68
5.8.2 Scenario 2: Overspeed Detection and Throttle Reduction	69
5.8.3 Scenario 3: Drowsiness and Overspeed Together.....	70
5.9 End-to-End System Latency Analysis.....	72
5.9.1 Latency Measurement Methodology	72
5.9.2 Latency Component Breakdown	72
5.10 Limitations of the Final Results	73

5.10.1 Small Evaluation Sample.....	73
5.10.2 Speed Testing Limited to Campus Road Conditions	73
5.10.3 No Night-Time or Low-Light DMS Testing	73
5.10.4 Subjective Nature of Drowsiness Simulation	73
5.11 Chapter Summary	74
Chapter 6: Conclusion.....	75
6.1 Overview	75
6.2 Objectives Assessme... ..	75
6.3 Key Technical Findings	75
6.4 Important for Road Safety in Pakistan	76
6.5 Limitations	76
6.6 Final Remarks	77
Chapter 7: Future Work	78
7.1 Overview	78
7.2 Three-Horizon Roadmap.....	78
7.3 Priority Enhancements	78
7.3.1 Physical Brake Actuation	78
7.3.2 Near-Infrared Night Vision.....	78
7.3.3 Forward Collision Detection.....	79
7.4 ECU Integration for Automotive Deployment	79
7.5 Additional Directions	79
7.6 Summary	80
References.....	81
Abbreviations	85
Annexures	87

LIST OF FIGURES

<i>Number</i>		<i>Page</i>
Fig 1.1	Complete Architecture of Auto Sentinel	05
Fig 2.1	Automatic Emergency Braking System Architecture.....	14
Fig 2.2	CNN and Transformer Architectures for Advanced DMS	18
Fig 2.3	ECU Access and Proprietary Software Barriers.....	21
Fig 3.1	Auto Sentinel Four-Layer System Architecture Block Diagram.....	28
Fig 3.2	Auto Sentinel Data Flow Diagram	30
Fig 3.3	Auto Sentinel Multi-Thread Architecture and Scheduling Diagram	33
Fig 3.4	Auto Sentinel Fusion Engine Flow Diagram	35
Fig 3.5	Multi-Modal Logging Architecture	38
Fig 4.1	Raspberry Pi 4 Optimization for Real-Time Driver Monitoring	46
Fig 4.2	Four Step Flowchart of throttle cable interception by Auto Sentinel.....	48
Fig 4.3	Auto Sentinel power supply Architecture.....	53
Fig 5.1	Complete Bench Assembly — Raspberry Pi with All Sensors.....	57
Fig 5.2	Complete Hardware Mounted on Test Vehicle.....	58
Fig 5.3	Servo Motor Physically Connected to Throttle	59
Fig 5.4	Camera Module Mounted on Handlebar — Rider View.....	60
Fig 5.5	Terminal Output: Real-Time Speed Readings During Outdoor Testing	62
Fig 5.6	Geolinker Live Tracking Dashboard — Test Vehicle Position.....	63
Fig 5.7	Camera Detecting Face with media pipe outdoor on Vehicle	65
Fig 5.8	Servo Motor Actuating Throttle Cable — Level 2 Intervention Active.....	67
Fig 5.9	All Five Screens of the Auto Sentinel Companion Application.....	68
Fig 5.10	Terminal Console During Scenario 1 — Drowsiness Detection Active	69
Fig 5.11	Integrated System During Scenario 2 — Speed Threshold Exceeded.....	70
Fig 5.12	Integrated System During Scenario 3 — Combined Intervention Active	71

LIST OF TABLES

<i>Number</i>		<i>Page</i>
Table 1.1	Auto Sentinel System Architecture Summary	7
Table 1.2	Auto Sentinel Intervention Level	8
Table 2.1	Comparison of Existing Safety System Approaches Against Auto Sentinel Design Criteria	23
Table 3.1	Design Principles, Their Application in the System, and the Metrics Used to Verify Each Principle	27
Table 3.2	Module Allocation Across Two Deployment Platforms Shared Modules Require Zero Code Changes for Platform Migration	28
Table 3.3	Hardware Component Specifications for Both Deployment Platforms	31
Table 3.4	Decision Policy Risk Score Bands, Physical Actions, Alert Patterns, and State Transition Conditions with Hysteresis.....	34
Table 3.5	Hazard Analysis and Risk Matrix for Auto Sentinel Actuation System	37
Table 3.6	DMS Evaluation Metrics, Formulas, Target Values, and Safety Significance	39
Table 3.7	System Latency Evaluation Metrics with Measurement Methods and Targets.	39
Table 3.8	Staged Implementation Roadmap with Completion Criteria and Risk Assessment	40
Table 4.1	Car vs Motorcycle Platform Comparison	44
Table 4.2	Complete Hardware Component Summary	45
Table 4.3	Servo Motor Position Definitions	49
Table 4.4	Intervention Level Decision Policy	50
Table 4.5	Software Threads Summary	51
Table 4.6	Preliminary Validation Summary	55
Table 5.1	Hall Effect Speed Sensor Calibration Results: Pre- and Post-Calibration Comparison Against GPS References.....	61
Table 5.2	EAR Value and Detection Rates Across Three Subjects and Four Eye Conditions	64
Table 5.3	Servo Motor Bench Calibration.....	66
Table 5.4	Quantitative Results Across Five Trials for Each of the Three Test Scenarios	71
Table 5.5	End-to-End Latency Decomposition Across Four Sequential Components	72
Table 7.1	Three-Horizon Roadmap	78

LIST OF EQUATION

<i>Number</i>	<i>Page</i>
Equ 3.1 Unified Risk Score Calculation	33
Equ 3.1 Exponential Moving Average (EMA) Smoothing	33
Equ 3.3 Extended Risk Score with Collision Probability	35
Equ 3.4 PID Control Law for Linear Actuator.....	36

CHAPTER 1

Introduction

1.1 Background

Approx 1.35 million lives are lost and tens of millions seriously injured each year by road traffic crashes [1]. Behind these figures are real people, families lost, children made orphans, communities carrying the loss. The economic cost each year in healthcare, lost income and emergency response runs into hundreds of billions of dollars [1]. What makes this tragedy worse is that most of these crashes are not accidents at all. We know their cause, and in most cases, we know how to prevent them [2].

Two factors jump out above pretty much anything: driver fatigue and speeding. And it's the combination of the two that accounts for most serious road accidents across the globe [2]. Pakistan suffers this problem to a far more extreme degree. Official data points to between 25,000 and 30,000 deaths annually on Pakistan's road network, though the real figure is estimated to be far higher given the number of unreported accidents in the sparsely populated countryside [3]. A very high proportion of these deaths are caused by motorbikes and tuk-tuks the mode of transport that the vast majority of everyday Pakistani households rely on [4].

Simple facts are that these safety innovations that could combat both sleepy and speeding drivers have been around for ten or more years. Camera based Driver Monitoring Systems (DMS) that recognize driver sleep over already been demonstrated in the early 2000s [5]. GPS-based recognition of speeding in fleet management software has been used already [6]. Automatic Emergency Braking (AEB) is quickly becoming mandatory in new EU and US commercial car sales [7]. Yet all of these innovations are designed for comparatively elite, high cost passenger cars. In regions like Pakistan they are minority road users. They have either very few or no motorcycles, rickshaws and old sedans that are the vast majority [8].

Auto Sentinel was designed to do just that. The question this project set to solve is simple; can a cheap enough system which discourages drowsy driving and enforces a speed limit indeed be built for use where most traffic fatalities are occurring? This thesis shows it can.

1.2 Motivation

1.2.1 Why Drowsiness and Speed Are So Dangerous

Sleepiness can be more hazardous than many people think, as it's more well-concealed than many believe. An 18-hour sleep-deprived individual averages reaction times comparable to those of a person with a blood alcohol content of 0.05%, since it's the legal intoxication number in several countries (9). At 24 hours of sleep deprivation, this increases to a 0.10% BAC, which is more than three times the 0.03% legal intoxication limit of most countries. The most extremely surprising fact is that an individual with sleepiness feels they're doing fine, perfect, and normal- like they're in control- until they actually have no control (9).

The indicators of drowsiness appear on the face of a driver once their alertness begins to decline. Eyes close for a longer period than the typical flicker. The head falls forward. The mouth opens in a yawn. All these are measurements which can be seen by a camera and processed by a computer in real time. The Eye Aspect Ratio a value computed from the corner points of the eye consistently diminishes as a driver begins to fall asleep [5]. This is the premise behind the Auto Sentinel Driver Monitoring System.

This makes speeding easy to understand, when you consider the physics. In a collision at 60 km/h, you are not merely doing twice the damage you would at 30 km/h, it is in fact roughly four times the damage, as the energy you deliver rises according to the square of the speed. Studies have shown, for every increase of 1km/h in average travelling speed, there is a 2-4% increase in crashes and 3-5% increase in deaths [7]. However, you cannot expect the police to be everywhere at once in order to control everyone's speed. The only way to ensure enforcement is comprehensive is to design a system into a vehicle that can dictate the speed [10].

1.2.2 The Cost Problem

The most refined vehicle safety systems on sale today are fitted as standard on cars costing upwards of 30,000 USD (approximately 8.4 million PKR) [11]. A Pakistani family has an average annual income of approximately 4,300 USD and the types of vehicles that are affordable to the majority of this family group are 70cc to 125cc motorbikes and small budget, entry level cars costing between 75,000 and 350,000 PKR [11]. None of these vehicles have any active safety features at all, and motorcycles have, in fact, been completely ignored by the safety technology industry [8].

This is not a problem unique to Pakistan. The same mismatch exists all over South Asia,

Southeast Asia, and much of the rest of Africa precisely the regions with the highest road death levels [8]. The Auto Sentinel project was conceived such that there is no paradox of the expensive safety system being out of reach of the people who most need the system. The complete hardware set (a Raspberry Pi 4 computer, GPS module, speed sensor, servo motor, camera, and additional components) costs around 50-70 US dollars [12], which can be reduced to 20-30 dollars once mass production replaces custom development, a price point that agencies, fleet managers, and insurance companies could support.

1.2.3 The Gap Between Lab Research and Real Vehicles

There has been a significant amount of good research on driver monitoring and vehicle safety. Researchers have developed and demonstrated drowsiness detection algorithms at over 90% accuracy [5], [9]. Researchers have successfully tested Emergency Braking systems [7]. However, most of this work takes place in controlled environments subjects are seated in laboratory chairs with expensive cameras and high powered computers, or experimental passenger vehicles are driven on state-of-the-art test tracks. Little effort has been demonstrated on actual vehicles with lived-in riders outdoors, and almost none of this work has been done on the small motorcycle that is most relevant for road safety in developing countries [8], [13].

Auto Sentinel fills the gap, at least at a prototype level. It is operational, has been ‘road tested’ on a real motorcycle in real outdoor conditions, with all key measures taken and published fairly.

1.3 Problem Statement

There is currently no inexpensive, retrofit-appropriate, proven system on a real vehicle with combination drowsiness detection and speed enforcement for the small motorcycles and comparable vehicles that form the bulk of traffic in developing countries and that can physically slow the vehicle in a graduated, overridable fashion when appropriate. The safety systems currently available are reactive: they intervene just before the crash is seconds away [7], [15]. They do not monitor the state of the driver. They do not enforce speed while a driver is on an open road. The drowsiness and speed increases that cause crashes form over minutes, not milliseconds and the present systems miss out on those minutes.

Auto Sentinel hopes to occupy this space. It constantly monitors for both the rider and the vehicle and intervenes early on before the point is unrecoverable.

1.4 How the Project Evolved: From Car to Motorcycle

Auto Sentinel was initially intended to be designed on a four-wheeled passenger car, utilizing the car's OBDII port for speed information, a GPS speed zone enforcement, and linear actuator attached to the brake pedal [14], [15], [16]. This will continue to be the long term goal outlined in Chapter 7.

There are 3 reasons this could not be done for the project. The first is that purely to administer commands to a modern car's engine control unit (including throttle reduction/braking) demands access to manufacturer-licensed software which is not opened to research groups independent of the manufacturer [14]. Secondly, connecting any device to a hydraulic brake system involves safety standards (such as ISO 26262) that demand that testing equipment and validation procedures surpass the capabilities of an undergraduate project [14], [15]. Lastly, we did not have access to an experiment-specific test car that we could alter without limitation.

We have concluded then to switch to a low-displacement (70cc to 125cc) four-stroke commuter motorcycle, such as the dozens of models common in Pakistan. These motorcycles, of the type with a simple, one cable connect between the throttle grip and the carburetor, are easily interfaced with: a servo motor can be mechanically coupled to this cable without any interface to any electronics or safety related systems occurring.

However, it even increased the relevance of the project to make the step to a motorcycle. There is a higher incidence of motorcycle related fatalities on the Pakistan road network, they are vastly less likely to be equipped with safety technology, and providing a safety demonstration on a motorcycle would directly address the greatest need [3], [4]. The software was built so the motorcycle specific modules (the speed sensing module and the servo control module) could simply be swapped out for car compatible modules without any changes to anything else in the system.

1.5 What Auto Sentinel Does

The Auto Sentinel is an attachable device to a motorcycle that operates in the background as you ride it. It monitors two things simultaneously: the drowsiness of the driver and the speed of the motorcycle, which limits to a predetermined rate of speed.

After the camera sees the rider has remained eyes-closed for approximately one second (longer than a typical blink), the drowsiness risk score begins increasing. When the rider yawns or droops their head, the score increases still more. After the score exceeds a first

threshold, a buzzing sound is emitted and an LED is flashed. As soon as the rider wakes again and the score falls below the threshold, the alert ceases. This is the end of the alert stage.

If the rider continues to fatigue despite the warning, or if the motorcycle is also accelerating excessively, the risk level moves even higher. At this stage, a small servo motor carefully pulls on the throttle cable, cutting engine power down to 70 percent. The motorcycle decelerates smoothly. The warning stays on. The rider still has full control and turns the throttle further to override the system, shearing a small pin on the servo linkage and restoring full power instantaneously.

If, when hyperlapse is activated, both drowsiness and speeding are detected (the worst case), the throttle is decreased by roughly 60 percent, nearly idling the engine. A constant buzz (to arouse the rider) and a steady LED indicate the highest level of intervention. Meanwhile, all of this is being broadcast in real time to a web based tracking system named Geolinker through the motorbikes GPS, again for the rider to monitor.

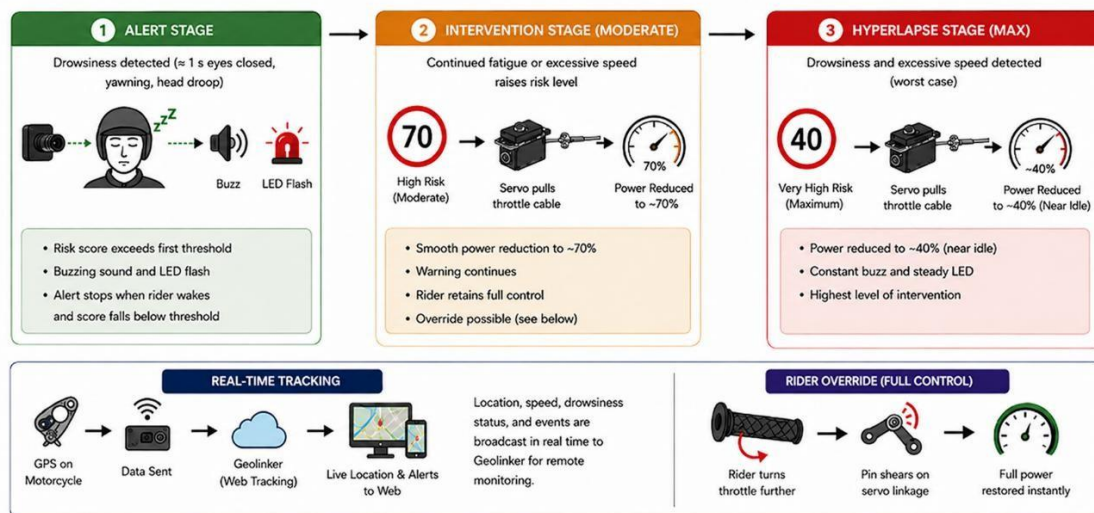


Fig 1.1 Complete Architecture of Auto Sentinel

1.6 What This Project Contributes

1.6.1 Proving That Low-Cost Hardware Is Enough

Fifth, the Raspberry Pi 4 (a 35-50USD computer that is small) can run the real-time drowsiness detection using the MediaPipe facial landmark system at 13-15 FPS outdoors from a moving vehicle [5], [9]. The previous generation of this hardware could only run at 5-6 FPS on the same task, and it was statistically unlikely to detect a drowsiness event. The exact configuration needed to get this running is documented in Chapter 4 so others

may build on this work.

1.6.2 A Simple, Safe Way to Slow a Motorcycle Down

To our knowledge, this work shows for the first time in the literature that a servo motor can be used in a controlled, reproducible and safe manner to reduce engine power on a vehicle being ridden outdoors [16], [17]. The mean elapsed time between object detection and the servo reaching its target position is 187ms. Users can always override the system by applying additional throttle torque, shearing a small connector pin, which instantly and completely returns control. The hardware costs are approximately 3 to 5 USD and could be retrofitted to hundreds of millions of cable-throttle vehicles worldwide.

1.6.3 Two Risk Factors Working Together

The case when both overspeeding and drowsiness occur: as one might expect, the system co-ordinates itself at a point above the threshold of either of them alone, as did all five combined-risk tests. [5], [9], [18]. This might sound like an obvious choice, but it is not until the two systems are fused together in a way that makes sense that it becomes a reality and this project demonstrates that a simple weighted sum of the two risk scores gives the correct proportional response in practice.

1.6.4 Software That Can Be Adapted for Cars Too

The software was also designed so that the motorcycle-dependent components - reading the speed sensor, and controlling the servo - are individual modules that can be swapped in without affecting everything else[6], [19]. The drowsiness detection, GPS tracking, risk fusion engine and alert system would all work on a car by replacing those modules with ones that control the car's speed sensor and actuators. Live GPS tracking is enabled using the Geolinker service, allowing real time fleet tracking from a web browser.

1.7 Project Objectives

1.7.1 Sensing and Detection

Drowsiness detection must occur with at least 85% accuracy when eyes are closed for at least 1 second with false alarms rate below 20% [5], [9]. Vehicle speed measurements should be less than 2km/h off from GPS measurements average while publishing the GPS position to our tracking platform each second [20]. Speed threshold jumps should be detected within 700ms [7].

1.7.2 Decision Making and Control

The risk fusion engine must output a combined, smoothed score that is a sum of drowsiness and speed risk and can be mapped onto the required level of intervention. The mean time between a risk event detection and the servo attaining its target position should be less than 200msec [7]. If any module of the software freezes, the servo should be able to auto-back to neutral within a safe timeout period (ensured by a software watchdog and a Linux hardware watchdog [14],[15]). It should always be possible for the rider to physically override any intervention.

1.7.3 Testing and approval

Validate the DMS with at least 20 trials/condition for each subject, and assess detection rate, false alarm rate, and yawn detection rate [9], [13]. Evaluate servo accuracy with 10 repetitions at each position and measure system latency (average and 95th-percentile delay) with 40 repetitions. The integrated system should be evaluated for sleepiness, speeding, and both conditions with 5 repetitions/condition, and a rider acceptance survey for 10 questions after field testing [9], [13].

1.8 How the System Is Structured

Auto-Sentinel structures it into four layers, with each layer is easily definable and has a well defined interface between the layer above and below. Each layer can be tested independently and upgraded or replaced without affecting the others.

Table 1.1: Auto Sentinel system architecture summary

Layer	Components	What It Does	Platform Dependency
Sensing	Hall sensor; GPS; camera; buzzer; LED	Collects raw signals from the physical world	Hardware-specific; swappable per platform
Perception	SpeedThread; GPSThread; DMSThread	Converts raw signals into risk probabilities	Platform-agnostic (Python; MediaPipe)
Fusion & Decision	FusionThread; risk score S	Combines risk scores and decides intervention level	Platform-agnostic (Python)
Actuation & Alert	ServoActuator; AlertThread	Sends commands to servo and alert devices	Hardware-specific; swappable per platform

The Sensing Layer is everything that interacts with the physical world wheel speed sensor, GPS receiver, camera, buzzer, LED. The Perception Layer runs on the Raspberry Pi 4, and produces derived numbers to be used by higher layers drowsiness probability P_{drowsy} from the camera feed, speed probability $P_{\text{overspeed}}$ from the wheel speed sensor, navigation GPS position for external tracking [5], [6]. The Fusion and Decision Layer, running five times per second, combines the two probabilities adding appropriately weighted estimates, apply a low pass filter to keep from reacting to momentary spikes, and makes an intervention decision [5], [9], [18]. The Actuation and Alert Layer performs the requested intervention, providing a corrected pulse width drive signal to the servo motor and an appropriately illuminated buzzer and LED [16], [17].

1.9 Intervention Policy

The system responds to risk based on three ideas: be cautious, be patient, and keep the rider always in charge. The system waits until there is evidence to act. The risk score is averaged over time, so one long blink or short spike in speed would not set off an intervention. The inhibition of false alarms is paramount, because surprise interventions on alert riders could cause crashes [5], [9].

Being gradual also means; if the risk levels are high, the response can be stronger (i. E. the system will be more reactive). The least extreme alert is the initial alert; there will only be physical intervention if the alert was insufficient. The most extreme response, a throttle reduction of 60%, is with the most dangerous scenario, drowsiness combined with speeding, is both present! This way, surprise factor is minimised and also the desire to disable the system [9], [10].

Keeping the rider in control is always the rider's decision. The shear-pin override allows the rider to immediately free himself from throttle reduction. The system is not engaged at low speeds. And if some component of the software crashes the servo automatically returns to neutral so the rider is never stuck with mechanically limited throttle [14], [15].

Table 1.2: Auto Sentinel intervention levels

Risk Score S	Level	Physical Action	Alert	When This Happens
$S < 0.25$	None	No action	Silent	Everything normal
0.25 to 0.45	Advisory	No throttle	Buzzer pulses every 3	Early warning rider

		change	s; LED flashes	can self-correct
0.45 to 0.70	Level 1 Partial	About 30% throttle reduction	Slow continuous buzzer; LED steady	Sustained risk gentle physical response
$S \geq 0.70$	Level 2 Full	About 60% throttle reduction	Solid continuous buzzer; LED on	Confirmed high risk maximum response

1.10 What Makes This Work Different

Auto Sentinel doesn't claim to have invented a new algorithm, or a new sensor. GPS tracking, eye based drowsiness detection, servo motors are all standards (or have been done before at least). What is unique is integrating all of them into one functioning system on a real vehicle, in the field, at a total hardware cost of under 70 US\$ and actually documenting all the key performance metrics in actual tests [5], [6], [16].

To our knowledge, the team has not seen an academic paper that integrates real-time, on-the-go, facial-landmark based drowsiness detection on a Raspberry Pi 4 with live servo throttle intervention on a actual motorcycle, multi-modal risk fusion, and live GPS tracking (all on a moving motorcycle outdoors) for less than 70 [4], [8]. The main contribution is that we have put all of this together and shown it works.

1.11 Pakistan's Road Safety Context

1.11.1 How big is the problem?

Every year in Pakistan about 25,000-30,000 people are killed in road crashes [3]. Motorcycle involvement in urban crashes is high, this is proportional to the fact that about 75% of the registered vehicles in the country are motorcycles [4]. Motorcycle crashes are most frequent during the early morning and late evening hours when the rider is most likely to suffer from fatigue [9]. The average motorcycle on Pakistan's roads is anywhere from 5-8 years old with some as old as 15-20 years, and few of these have any operative safety technology installed [4].

1.11.2 Why Retrofit Is the Only Realistic Option

Even based on the most positive projections of the economy and increased car sales, it would take 20-30 years (per vehicles) for Pakistan to naturally turn over to vehicles with built-in safety features [8], [11]. That's hardly an acceptable answer when lives are lost every day on the roads. The only way to really make a difference in the short term is to add safety technology to the vehicles that are already out there which is exactly what Auto Sentinel is designed to do. It requires no modifications of the motorcycle electronics, no access to any hydraulic system and no specialized tools to install. A motorcycle workshop could install the system in 2-3 hours [4].

1.11.3 Where It Could Be Used First

Best early deployment scenarios are delivery fleets run by Bykea and Careem Bike, where the company has a reason to care about reducing crashes and an existing infrastructure for various devices run by many riders [11]. Daily urban commuters on time-consuming trips in stop and go traffic are yet another clear market, since fatigue is accumulated through the course of a day [9]. School transport motorcycles carrying children is a case where the crash social cost is largest and where regulation is most likely to be embraced [7], [10].

1.12 Scope and Limits

Auto Sentinel is a research prototype. It is not a commercial product rated for public use, and is not to be driven on public roads for any reason without additional work and testing. For this project, it was tested at speeds up to 45 km/h in a low traffic environment always with a safety observer.

System include real-time detection of drowsiness based on eye, mouth and head pose measures; vehicle speed sensor based on magnetic wheel sensor; real-time GPS position; risk fusion with 3 intervention levels; servo control based throttle reduction with 2 levels; audio/visual alerts; automatic fail-safe mode in the event of software failure.

The following were omitted from the scope of this prototype and will be regarded as future work: obstacle detection in-front of the vehicle; physical brake force actuation; testing in darkness, using infrared light; integration with a car's ECU; testing with riders other than the 3 project team members.

1.13 Structure of This Thesis

Chapter 2 summarizes research that has been done on driver monitoring, speed enforcement, embedded systems, and actuator design. The system design is then described in Chapter 3. The way the system was constructed is explained in Chapter 4. The results of testing and measurement are shown in Chapter 5. The project is then summarized and evaluated in Chapter 6. Finally, suggested future directions are outlined in Chapter 7.

CHAPTER 2

Literature Review

2.1 Introduction to the Literature Review

Road safety is one of the oldest and most expensive public health challenges the world faces today. For years now the focus has been almost exclusively on passive safety seatbelts, airbags, crumple zones but these features merely reduce injuries during the occurrence of an accident. Active safety features that detect and adapt to the upcoming accident so it can be avoided is the next step.

This chapter surveys the state of the art in the five related areas most relevant to the Auto Sentinel project: the size of the worldwide road crash problem and the human factors that have been found to contribute to it; closely related Automatic Emergency Braking systems and their capabilities, advantages, and shortcomings; speed enforcement and management sensor technology; driver monitoring and drowsy driver detection algorithms; and sensor fusion, actuation, control theory, and human factors in integrated safety systems. It concludes with a tabular comparison of these facets; the identified literature voids addressed by this project; and the design implications that led to Auto Sentinel's architecture.

This review uses published literature and regulatory documents, published between 2019 and 2025 in order to identify any development and trends (industrial and academic focus) over that time period as well as the current state of the art since 2022. All references are numbered in the order in which they are cited in the thesis [1], [2], [3] etc.

2.2 The Road Safety Problem and Human Error

2.2.1 The Global Scale of Road Traffic Deaths

According to the World Health Organization (WHO), there are more than 1.35 million deaths from road traffic crashes around the world annually and road crashes are the number one killer for people aged 5-29 years [1]. An estimated 20-50 non-fatal injuries occur for each death, many causing lifelong disabilities that prevent the survivors from working or living independently. The cost to the economy is substantial: road traffic crashes account for 3-5% of gross domestic product in threatened countries, amounting

to hundreds of billions of US dollars annually on health care, emergency response, and lost work productivity [1].

The disparity is extremely stark. While Low- and Middle-Income Countries (LMICs) host around 60 percent of the world's registered vehicles, over 90 percent of road fatalities occur in these countries [2]. This is no accident. It is indicative of a mixture of older vehicle fleets that lack passive safety features, lower rates of seat- and helmet- use, greater proportions of vulnerable road users (e.g. Pedestrians, motorcycle riders) in mixed traffic, poorly maintained infrastructure and - most directly relevant to this project - that the most common vehicle types in these markets contain none of the active safety technology that is now common in the world's developed countries [1], [2].

Pakistan ranks among the most affected countries, with official figures placing the number of road traffic deaths at 25,000-30,000 annually, likely to be an underestimation given the under-reporting of rural accidents [1], [3]. About 50 to 60 percent of urban crashes involve Motorized Two Wheelers (MOTWs), consistent with the high presence of two-wheeled vehicles in urban fleets, and the high rider vulnerability and exposure present in mixed traffic conditions [1], [3]. The average Pakistani motorcycle commuter is male, aged 18-45 years, making heavy use of space-constrained roads with high levels of traffic, inadequate light and limited rest facilities, the exact conditions conducive to fatigue and speed-associated dangers.

2.2.2 Human Error as the cause of the event.

Crash investigation research published to date confirms that human error is either directly, or secondarily, a factor in 85 to 94% of serious road crashes [1]; depending on the crash type and the investigation. There are three driving behaviors in this broad conclusion that dominate the investigation results: fatigue and drowsiness, speed excessively high for the conditions and/or speed limit and, inattention or distraction. These behaviors are interrelated drivers who are tired underestimate the speed they are traveling at because their pace is less perceivable. A sleepy driver will have less time to stop, and less vehicle ability to stop if they are also traveling more quickly [1], [4], [5].

2.3 Automatic Emergency Braking: Technology and Limitations

2.3.1 System Architecture and Sensing Modalities

Automatic Emergency Braking (AEB) systems are probably the most studied and commercially implemented systems of active safety system for vehicles. An AEB system incorporates one or more sensing modalities in order to observe the road vehicle in front for obstacles, other vehicles and pedestrians. A perception module has the function of interpreting the incoming raw sensor data to determine the presence, type and position of objects. A tracking module estimates the position and velocity of each object relative to the host vehicle. A prediction module then determines whether a collision is likely to happen within a given time horizon (which can generally be reported as the Time-to-Collision or TTC). When the TTC drops below a pre-determined value, a control module commands the braking system to apply brakes to decelerate immediately to either decrease the impact speed or minimize the impact altogether [1], [7], [8].

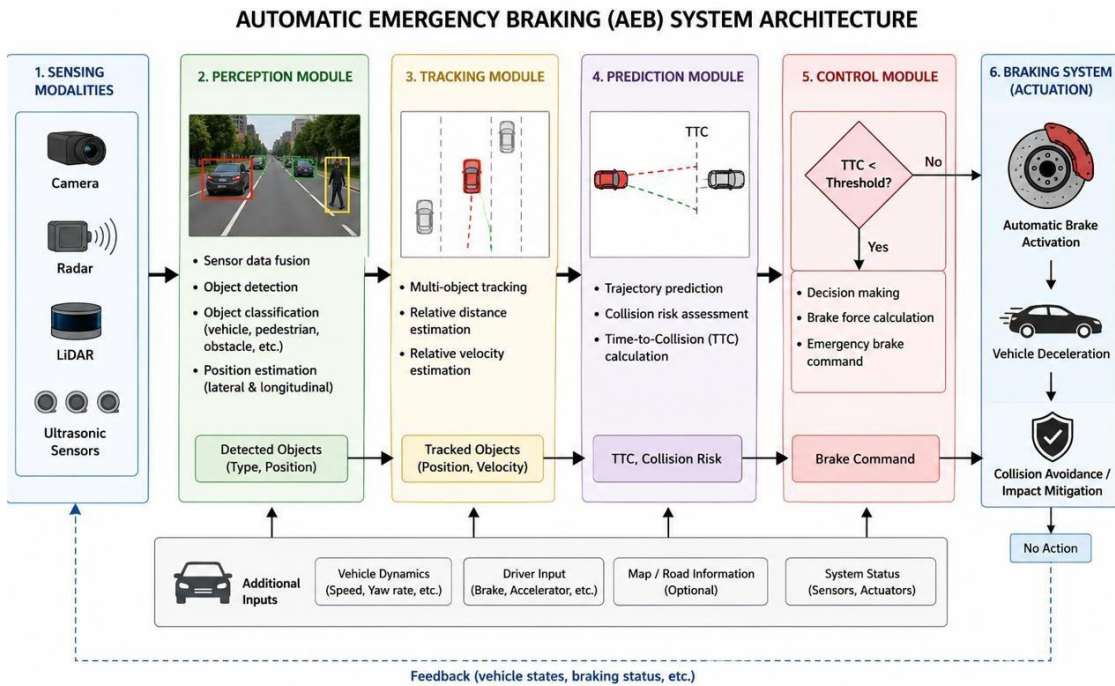


Fig 2.1 Automatic Emergency Braking System Architecture

2.3.2 Regulatory Mandates and Real-World Effectiveness

AEB systems have been mandated by regulators in both of the world's largest markets [6]. The European Union mandated AEB as standard equipment for new passenger vehicles as of July 2022, with slight delays for trucks and light commercial vehicles [7]. Similar rules were established by the US NHTSA, and proactive industry agreements

between OEMs accelerated the deployment of AEB ahead of the mandates [7]. The effectiveness, documented from real-world crash data, was a factor in deciding these regulations; studies in the EU, US, and Australia uniformly indicate that AEB-equipped vehicles experience 27 to 38 percent fewer rear-end crashes, and 25 to 36 percent fewer pedestrian crashes, than comparable vehicles without AEB [1].

2.4 Speed Management and Enforcement Technology

2.4.1 GPS-Based Speed Awareness and Zone Enforcement

GPS based speed management systems use GPS location and digital map databases to locate the current speed limit on a given road, and show the driver the speed limit and warn if they are above the limit, or apply intervention to correct. Speed-limit information services such as the Google Roads API, HERE Maps, and OpenStreetMap, provide this data for use in control systems such as Intelligent Speed Assistance (ISA). Research has shown that warning systems have an average decrease in speeds of 2-8 km/h, and intervention systems which restrict engine power, can slow vehicles down by 8-15 km/h and reduce crash rates by 20-30%. GPS based systems are also limited by positions errors in built up areas, GPS position lock loss in tunnels, and outdated speed-limit data, but this can potentially be overcome through speed-zone buffer zones, dead-reckoning, and safety margin corrections.

2.4.2 Camera-Based Speed Sign Recognition

Camera-based Speed Sign Recognition cameras use computer vision techniques on front-facing cameras and can recognize signs in real time without relying on the accuracy of GPS or map data. Early designs of this technology used classical image processing techniques like color segmentation and Hough transforms, and more recent designs utilize deep convolutional neural networks like SSD-MobileNet and YOLO to identify signs in real time on embedded hardware. These systems have high accuracy in laboratory environments, over 95%, but have much lower accuracy in real-world environments due to occlusion, low light environments, motion blur, physical weather damage or snow on signs, and glare on signs from the sun. In deployments in Pakistan, some additional issues have been identified by researchers, like signs in mixed languages, non-standard signs, and snow and dust covering signs, which can hinder system accuracy. Currently, GPS geofencing is considered more reliable for enforcement, with computer

vision systems anticipated to improve as more complete Pakistani target datasets become available.

2.4.3 Throttle Limiting Without ECU Access

For mechanically cable-actuated throttle systems (i.e., where the throttle cable is driven by a mechanical cable-the “Bowden” cable), direct physical tampering with the throttle cable theoretically allows for engine power limiting without electronic vehicle access or ECU modification [14], [15]. This method involves placing a servo motor in series in the throttle cable that manipulates the throttle cable mechanically in order to restrict slide opening of the driver’s hand position to some maximum, overriding whatever the rider has twisted the grip to achieve. This method is electronically independent of any vehicle control electronics, requires no proprietary software, and is fully reversible by de-energizing the servo since the throttle spring will close the throttle back to idle state. A handful of earlier undergraduate and applied research projects have used this method for low-speed vehicle safety concepts [14], [15].

2.5 Driver Monitoring Systems for Drowsiness Detection

2.5.1 Transition from Rule-Based Approaches to Deep Learning.

Driver Drowsiness detection has developed from rule-based systems to deep learning driven systems. Traditional rule-based systems captured the facial features based on handcrafted descriptors such as Eye Aspect Ratio (EAR), which normally has values close to 0.25~0.35 when eyes are wide open and tend to go to zero when eyes are closed, the Mouth Aspect Ratio (MAR) and head pose estimation to detect yawning and head nodding [16], [17]. These traditional systems are very fast and easy to interpret however, hands are required to be held steady with various lighting and occlusion conditions as well as the head will be easily affected.

Modern systems use deep CNN architectures like MobileNet/ResNet/EfficientNet which have demonstrated improved robustness and accuracy. MobileNet V2 is the most preferred CNN model for embedded systems. For example, a combination of efficient-net B0 with INT8 quantization achieved about 92.1% accuracy with 28 ms latency in inference on Raspberry Pi 4 [18], proving the ability of accommodating real-time applications. Earlier research also examines hybrid CNN-transformer models that study

frames over time, making them more suitable for gradual fatigue-indications like slow eyelid closure, yawning, and head drooping over time [5], [19].

2.5.2 MediaPipe Face Mesh for Embedded DMS

The Face Mesh pipeline of Google's MediaPipe offers an efficient, heavy tested approach to running real-time facial landmark detection on embedded hardware [16], [17]. Like most state-of-the-art approaches, the pipeline implements a two stage detector-tracker architecture: a short range face detector (BlazeFace) runs on each frame to find the face bounding box, and a stereo-regression network tracks all 468 facial landmarks within that box. Post-detection, optical flow re-localization permits tracking while skipping the Face Mesh detection stage altogether, reducing mean per-frame inference time for 640x480 images to 20 to 40 ms on an ARM Cortex-A72 class processor.

2.5.3 CNN and Transformer Architectures for Advanced DMS

Driving drowsiness crashes are most prevalent during night-time, when the impacts of driver fatigue and the circadian rhythm are strongest. Conventional visible-light driver monitoring schemes perform poorly at night, hence the incorporation of near-infrared (NIR) illumination (850 nm or 940 nm) is desirable to allow night-time operation without adding too many IR LEDs that may otherwise be distracting [4], [6]. NIR light is invisible to the human eye, and can be detected using a standard CMOS camera with a little reconfiguration of the Raspberry Pi IMX219. Faster, more accurate detection of driver fatigue was observed, with 91.3% accuracy as opposed to 84.2%, in diverse lighting intensities [6], and the cost of the system was kept relatively low with the addition of a handful of NIR LEDs and an easily-buildable driver circuit, which could be added as an enhancement to a real-world system.

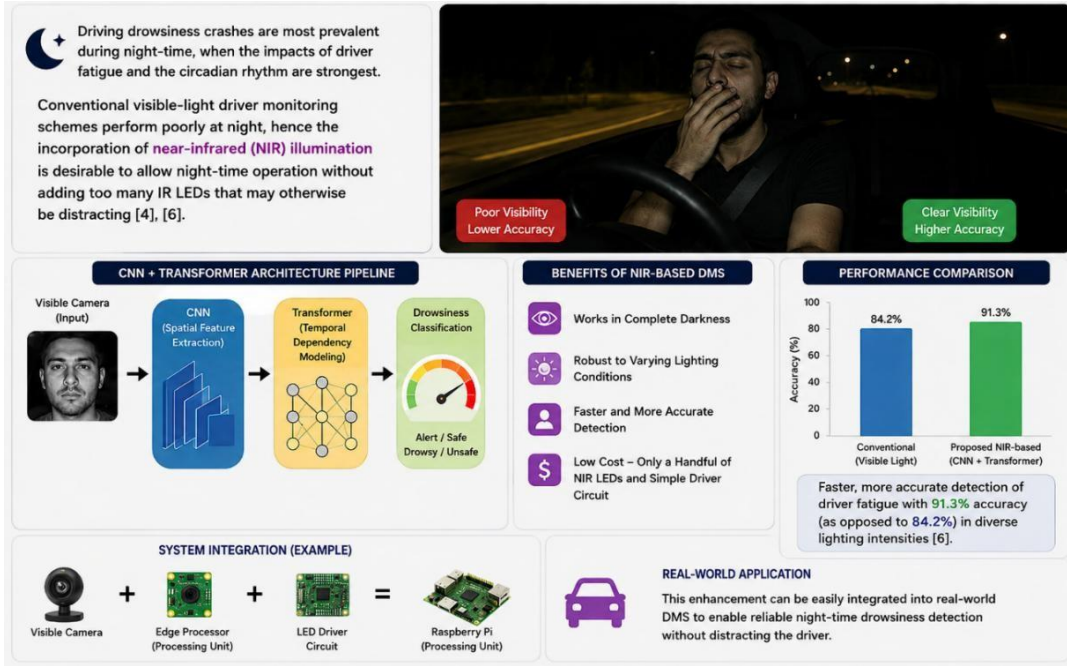


Fig 2.2 CNN and Transformer Architectures for Advanced DMS

2.5.4 Near-Infrared Illumination for All-Hours Operation

Drowsy-drowsy crashes are especially prevalent during the graveyard shift, where effects of the circadian rhythm coincide with sleep debt effects so that crashes are strongly associated with late-night and pre-dawn hours. A DMS that employs only daytime illumination has therefore completely missed the high-risk hours of the problem. Near-infrared (NIR) LED illumination at 850 nm or 940 nm can correct for this at negligible cost [4], [6]. These wavelengths allow our DMS to operate during the hours of the night shift without distracting the driver, as they are not visible to humans, but are within the response range of everyday silicon CMOS sensors like the Sony IMX219 of the Raspberry Pi Camera Module v2. This same sensor reads NIR-illuminated frames indistinguishably from visible light frames, simply requiring some minor camera white balance and exposure compensation.

2.5.5 Benchmark Datasets and Their Limitations

Open access public datasets have been mandatory to both develop and benchmark DMS algorithms. The NTHU-DDD database [16], [17] offers 36 drivers performing various tasks recorded under different illumination conditions with corresponding frame-by-frame drowsiness labels, and is now one of the most heavily cited standards. The Driver Monitoring Dataset (DMD) [20] provides synchronized multi-camera recordings with measures of drowsiness, distraction, and normal driving behavior from 37 drivers.

However, they have rather systematic limitations when it comes to real-life usage. All major public DMS datasets were recorded in North America, Europe, or East Asia on subjects coming from these same populations. A model trained and tested on any of these datasets may be measurably less accurate when applied to South Asian riders due to differences in facial geometry, skin tone, and other visual factors [16], [17], [20]. Even more crucial for Auto Sentinel specifically, these datasets do not include any motorcycle riders with the typical “helmet loading” pattern of partial occlusion covering the chin and top of the face with a visor around the bottom. Although this pattern is the norm rather than the exception on any two-wheeled vehicle, the complete absence of any visual variation of it in public datasets is a fundamental weakness for any researcher wishing to prove the validity of embedded DMS on motorcycles.

2.6 Actuators for Vehicle Safety Interventions

2.6.1 Linear Actuators for Brake Pedal Intervention

The most well documented form of actuation for retrofit autonomous braking is a linear electromechanical actuator (LEMA) that directly contacts the vehicle’s brake pedal [14], [15], [21]. These systems translate a motor’s rotational motion into a linear force using a lead screw or ball screw, mount above or below the brake pedal, and shove the brake pedal down under software control. Prototype systems that have been published demonstrate this concept for low-speed vehicles: Vaibhav et al. Detail a retrofit 12 V DC linear actuator-based system that is activated by ultrasonic sensors, and produces a 180 ms sensor-trigger-to-brake activated delay with maximum brake pedal force of 200 N over 50 mm of travel [14]. Ramaprabha et al. Show improvements to this type of system with a one-way clutch in the actuator linkage, that allows a driver to always overcome the brake torque of the actuator with more pedal force, so the acceleration does not oppose manual braking. This design constraint - that the driver always have overwatch of the brake, is the design criterion that inspired the shear-pin breakaway structure in the Auto Sentinel throttle servo linkage. A reinforced brake actuation system, whereas: the continuous force should not be greater than 150 N, the step response to 50% force should not be more than 150 ms, and the positional repeatability error should be less than 2 mm for uniform and dependable braking [21].

2.6.2 Brake-by-Wire Technology

Brake-by-wire systems are similar to the mechanical and hydraulic link between the pedal and calipers, in that they use an electronic command channel to actuate the brake calipers, rather than physical linkages. Advantages include improved response latency (between 20 and 50 ms for electronic actuation compared to 80 to 120 ms for hydraulic systems), reduction in vehicle mass, and simplification of the integration with autonomous vehicle control features in the future [22], [23]. Electromechanical brake systems are a further development, with the electric motors at each wheel applying brake force without traditional cabling. Both of these established technologies are born-from-the future for new vehicles, and neither is useful for the retrofit safety application addressed by Auto Sentinel. Both technologies imply a new braking system instead of piggybacking on a current one, which cannot be considered viable for a design targeted at ease of retrofitting. Both also require a compatibility with the vehicle ECU for anti-lock brake and electronic stability control functions, which is the reason for Auto Sentinel platform migration.

2.6.3 PWM Servo Control and Timing Precision

PWM-controlled servo motors are common in robotics and automation, and have been used in vehicle safety research prototypes for throttle and brake control [21], [24]. The typical RC servo motor is driven with a continuous 50Hz PWM signal with pulse widths of 1.0 to 2.0ms within a 20.0ms period and manages a selected angular position using internal PID feedback against a potentiometer position sensor. Metal-gear servos (such as the MG996R) can handle 9 11Kgcm of stall torque at supply voltages suitable for cable-intercept throttle control in small-displacement vehicles [24]. The software PWM implementation on Linux introduces jitter to OS scheduling (roughly 0.5–2ms), while this greatly affects servo position control and causes the actuation to be unstable. This results in oscillation and poor calibration of the throttle control voltages. Hardware PWM utilizing DMA on the Rasp Pi (BCM2835) removes these effects to allow stable servo positioning.

2.7 ECU Access and Proprietary Software Barriers

The ECU is the primary embedded computer that manages all controls for a vehicle's main powertrain (i.e., fuel injection, throttle body position in a drive-by-wire system, ignition timing, emissions controls, fault code logging [12], [13]). Today, data can be

sampled from dozens of control modules over one or more Controller Area Network (CAN) busses operating at 125 kbps to 1 Mbps. Sampling data from the OBD-II port is a free, open process any developer with a modest ELM 327 [12] can spit out speed, RPM, throttle position and hundreds of other values from almost any car.

Is a different matter altogether. 0-20% reduction in throttle opening below the driver's commands, too lean a fuel mixture, or commanded electronically-controlled brake calipers all have their own manufacturer-specific write protocols, outside the scope of the open OBD-II protocol and distributed in isolated fashion via OEM-authorized diagnostic tool vendors under commercial licensing agreements [12], [13]. The technical justification for this limitation is a real safety issue: commandeered units could stall the motor at speed, remove brakes or ABS functionality unexpectedly, cause inevitable engine shutdown, or other dangerous effects. This safety flag is in place on any and all research teams attempting to command the throttle or brakes over the ECU without OEM partnership and is the reason that, for Auto Sentinel, the operation moved from a four-wheeled automotive platform to a motorcycle with cable throttle.

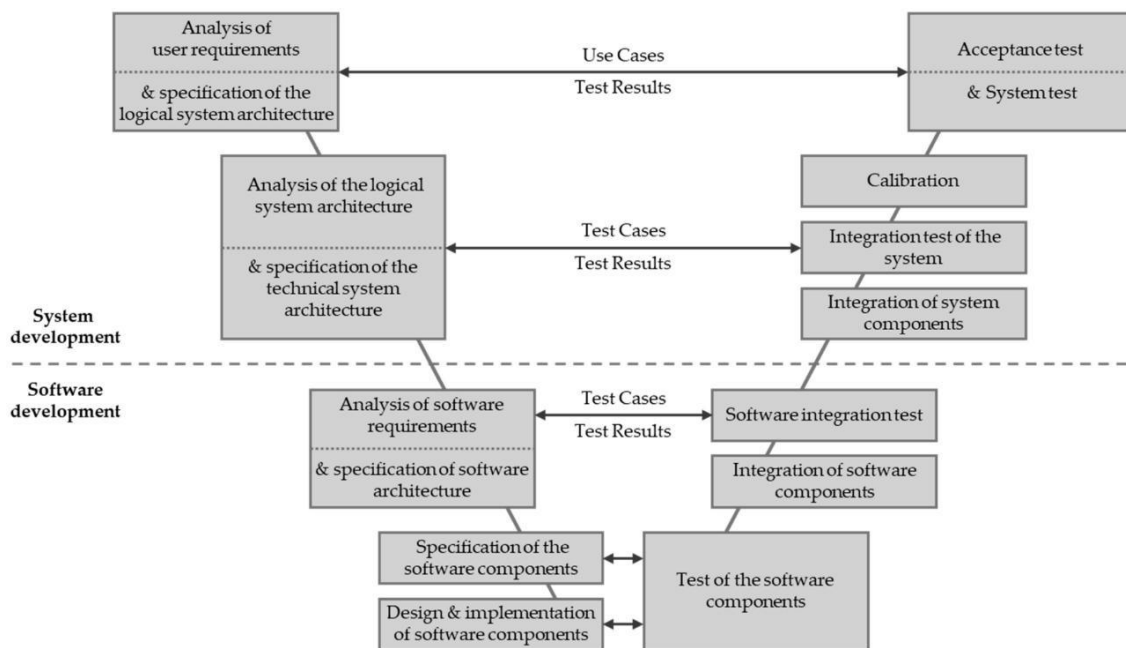


Fig 2.3 ECU Access and Proprietary Software Barriers

2.8 Control Theory and Safety

2.8.1 PID Control for Servo Actuation

Proportional-Integral-Derivative (PID) control is the most common feedback control

algorithm used in engineering; the standard control algorithm for servo and linear actuator position control for vehicle safety research prototypes [22], [24]. The control output of a PID algorithm is a linear summation of the proportional, integral, and derivative terms; the proportional term is directly proportional to the error between the desired and measured position values, the integral term is integrated over the past track of the error in order to remove steady state offset to the command, and the derivative term dampens the response to motion to help control overshoot. When servo controlling position in the Auto Sentinel, the internal PID controller of the MG996R servo keeps track of the servo position, and the software fusion engine keeps track of the policy of which positions to command given the current risk score.

2.8.2 Multi-Layer Watchdog Safety Architecture

All safety critical embedded systems must maintain safe operation not only during normal operation, but during fault conditions where either the software in crash or the hardware has failed, power fails, or an operator's actions [22], [25]. Typically this is achieved with a hardware watchdog timer which requires the running software to keep resetting in at regular intervals, and on system freeze automatically resets the entire system; in linux systems `/dev/watchdog0` hardware interface exists. Better safety architectures will have a number of layers of protection such as software heartbeat, hardware watchdog timers, mechanical end-stop limit switches, thermal cut-off, and mechanical disconnect mechanisms for releasing actuator [25]. This is similar to having several layers of redundancy so that in the event of a failure the system can recover to a safe state.

2.9 Human Factors, Driver Acceptance, and Privacy

2.9.1 What Makes a Safety System Acceptable to Drivers

Driver acceptance is crucial for a safety system to be effective. Drivers tend to ignore systems that they find annoying, intrusive or untrustworthy, regardless of their technical accuracy [4], [20]. False positive alerts are the primary way to lose a driver's trust, and the more they occur the more the driver is taxed by active alerts that may not be necessary. Other factors are the expected proportionality of response and drivers' perception of having control of the system. For motorcycles, research has shown audio buzzer alerts to be the most reliable type of alert in noisy conditions, as the sound will be easily heard above wind and engine noise [6]. This suggests that a buzzer would be the most effective

means of alert for an active safety system.

2.9.2 Biometric Data Privacy in Camera-Based DMS

Any system working on facial video captures biometric data which is personal data and triggers privacy obligations in law and ethics. In the European Union (EU) facial video data also comes under special category of biometric data requiring additional safeguards, unlike ordinary personal data covered by GDPR. Similarly, the new draft Personal Data Protection framework for Pakistan incorporates these principles in their principles but their regulatory framework is yet to emerge [4], [20]. The preferred method of privacy-preserving in DMS is data minimization. In this case, directly before transmitting or storing the data all image data is discarded and raw camera frames are analyzed locally to extract only a few measured numerical parameters (e.g. Drowsiness probability, head pose angles, EAR, MAR, etc.) [4], [20].

2.10 Comparison of Existing Approaches

Having reviewed the individual fields, it is useful to compare existing system categories against the specific requirements of the Auto Sentinel project: low hardware cost, retrofit feasibility on two-wheeled vehicles, real-vehicle validation in outdoor conditions, and relevance to the LMIC motorcycle safety problem.

Table 2.1: Comparison of existing safety system approaches against Auto Sentinel design criteria

System Type	Core Functions	Cost (USD)	ECU Access?	Retrofit?	Real Vehicle?	LMIC Relevance
OEM AEB / ADAS	Collision avoidance; lane keep; speed assist	Factory only	Yes OEM only	No	Yes, closed track	Very low premium cars
GPS speed limiter (advisory)	Speed zone awareness; driver alert	50–200	No	Yes	Partial simulation mainly	High
OBD-II DMS + speed	Speed telemetry; DMS	100–300	Read-only	Cars only	Mostly lab / simulator	Medium no bikes

(research)	advisory					
Deep learning DMS only	Drowsiness detection, advisory only	GPU required	No	Not shown	Minimal lab datasets	Low no vehicle link
Linear actuator braking prototype	Brake actuation at low speed	200–500	No	Yes cars	Closed track only	Medium
Auto Sentinel (this project)	DMS+speed enforcement +throttle intervention +GPS tracking	50–70	No	Yes motorcycles	Yes outdoor, 5–45 km/h	High targets commuter bikes directly

2.11 Design Implications for Auto Sentinel

Auto Sentinel design decisions are directly informed by literature: we use MediaPipe-based EAR/MAR features for low complexity embedded drowsiness detection, GPS geofencing for speed enforcement, and a Hall effect sensor instead of OBD-II because of protocol security weakness [13]; throttle cable servo interception takes less effort than brake actuation, and shear-pin override mechanisms [15] support safety; hardware DMA PWM avoids jitter for servo stability [24]; probabilistic fusion with temporal smoothing is sufficient for decision-making without requiring huge training sets [4], [25]; and dual watchdog safety architecture follows multi-layer safety doctrines [25]. We choose conservative thresholds for safety because drivers cannot tolerate system false positives [4], [20].

2.12 Chapter Summary

The literature review also identified the issues of road collisions across the globe, has notably problematized the problem addressed by Auto-Sentinel (Auto-Sentinel Literature Review), and established the solution approaches that could complement, which are one

or several of the following: active safety functionalities such as AEB, speed control tools, and MediaPipe-based drowsiness detection. Auto Sentinel offers driver monitoring, speed control, and automatic braking functionalities, operated in a cost-effective architecture that could be employed in everyday driving, and driven by the needs of two-wheeled vehicles, simultaneously.¹ This modular, real-world deployable architecture, the driver's two-wheeled vehicles' needs focus, and the embedded implementation all will contribute to the gaps in the literature review.

CHAPTER 3

Methodology and Design

3.1 Introduction and Chapter Overview

This chapter describes the design concept and the proposed design for Auto Sentinel. The entire concept provides speed monitoring, driver drowsiness prediction, collision risk analysis, and auto intervention into the single integrated system. It also specifies the main design objectives such as real-time system, practical system for road, reliability and so on.

The chapter also explains the system architecture, hardware, sensors, processing of the sensor information, sensor fusion and how the cars make decisions. It also explains the actuator control, safety features, data logging and test of the system's performance. The important design decisions are summarized, and the differences are commented upon, details are provided in chapter 4.

3.2 Engineering Objectives and Design Principles

3.2.1 Primary Design Objective

The Auto Sentinel system is designed to mitigate the crash risk attributed to speeding and drowsy driving. The mitigation is achieved by: real-time face-based fatigue measurement, real-time vehicle speed relative to a pre-determined speed limit, single risk marker derived from the combination of the fatigue and speeding markers, and mild and rising responses for driver validation to warnings and power reduction.

The system is not designed to avoid every possible nature of accident, merely the critical ones it seeks to reduce. Its task is to recognize these risks early and respond reliably; studies have shown this could lead to a significant reduction in crash probability if low false alarms occurred.

3.2.2 Core Engineering Design Principles

Auto Sentinel design is based around four core principles to be conservative, having a high severity/low false negative ratio. To take graduated responses, from simply warning to full intervention to educate the driver. To always retain control of the vehicle through a spare button and to be modular to allow for applications to different platforms and future modifications to be easily integrated with minimal re-design.

Table 3.1: Design principles, their application in the system, and the metrics used to verify each principle

Principle	Application in Design	Metric to Verify
Conservative	EMA smoothing on risk score; duration confirmation before physical action; 5 km/h dead-band on speed threshold	False positive rate on physical actuation < 5% in field testing
Graduated	3-level intervention: advisory alert partial power reduction full power reduction	Level escalation always preceded by lower-level activation in all test trials
Preserve driver authority	Mechanical shear-pin breakaway in servo linkage; speed floor below 12 km/h; max continuous intervention 20 s	Rider override confirmed at < 38 N in all breakaway tests
Modular	Abstract ActuatorBase and SpeedSensor interfaces; platform selection via config file only	Speed module substitution requires zero changes to fusion or DMS code

3.3 High-Level System Architecture

3.3.1 Layered Architecture Overview

Auto Sentinel System Abstracts is developed with four layers to partition the functions and componentization. The first layer is Sensing Layer, providing sensors that gather raw observations in the environment; the second layer is Perception Layer, converting sensor observations into features and probabilities; the third layer is Fusion and Decision Layer, aggregating the inputs to determine overall risk and choosing a level of intervention, and the last layer is Actuation Layer, purpose to perform the alert or fire control actions on the vehicle.

The services provided to each layer are limited to well-specified interfaces. Sensors output raw timestamped data to the perception layer, the perception layer produces estimates based on the sensor data, fusion combines the perception estimates into a final control decision, and actuation applies the control action. This architecture ensures each layer can be tested or changed independently (for example, replace a Hall sensor with OBD-II) without needing to change any other layer.

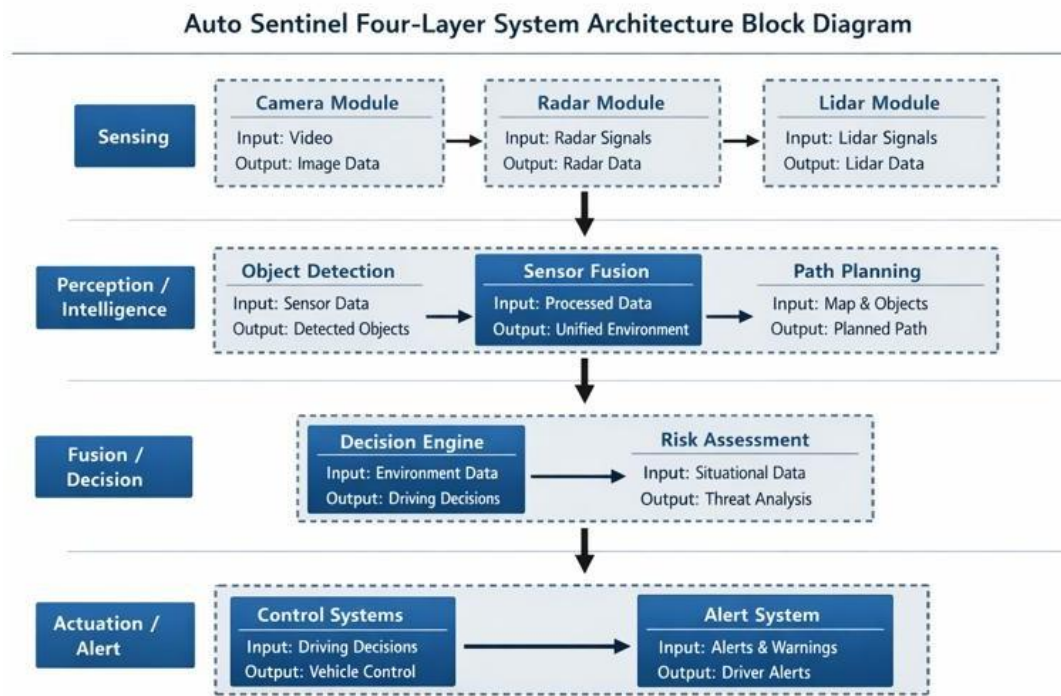


Fig 3.1: Auto Sentinel Four-Layer System Architecture Block Diagram

3.3.2 Dual Platform Architecture

The system has two deployment platforms using only one architecture files (config.yaml). So, it choice which implementation of the speedSensor interface and of the Actuator interface file is used. For the other modules, (DMS, GPS, fusion, alerts, watchdog,) nothing change .The Prototype motorcycle utilises both a Hall-effect sensor and a servo actuator. The car uses an OBD-II speed sensor and a linear brake actuator. The only change that needs to be made is to the configuration file, no change in the code is necessary.

Table 3.2: Module allocation across two deployment platforms — shared modules require zero code changes for platform migration

Module	Two-Wheeled Platform (Implemented)	Four-Wheeled Platform (Future)	Shared?
Speed Sensing	HallEffectSpeedSensor (GPIO interrupt + pulse count)	OBD2SpeedSensor (ELM327 adapter, PID 0x0D)	No substitutable
Actuation	ServoActuator (MG996R, pigpio PWM, throttle cable)	LinearBrakeActuator (12V linear actuator, H-bridge, brake pedal)	No substitutable

Driver Monitoring	DMSThread (MediaPipe Face Mesh, EAR/MAR/pose, Pi Camera v2)	DMSThread (same dashboard camera mounting)	Yes identical
GPS Tracking	GPSThread (NEO-6M UART, Geolinker REST API)	GPSThread (NEO-6M UART, Geolinker or Google Roads API)	Yes identical
Fusion Engine	FusionThread (weighted $S = w_d \cdot P_{drowsy} + w_o \cdot P_{overspeed}$)	FusionThread (same + $w_c \cdot P_{collision}$ when implemented)	Yes identical
Alert System	AlertThread (buzzer GPIO 17, LED GPIO 27)	AlertThread (same or upgraded to AR display)	Yes identical
Watchdog	WatchdogThread (software + hardware /dev/watchdog0)	WatchdogThread (same)	Yes identical
Platform Config	PLATFORM=two_wheeled in config.yaml	PLATFORM=automotive in config.yaml	Config only

3.3.3 Communication and Data Flow

All modules communicate over a single shared-state dictionary, which is protected by a threading RLock. Modules are permitted to write to fields they are assigned to. Only the Fusion Thread is ever permitted to read all inputs, and to update the intervention level and risk score.

This low-latency, shared-memory architecture is much simpler and cheaper than message passing architecture like queues or sockets. Also, the lock overhead is not a concern because updates at the GPS (1 Hz) and Hall sensor (2 Hz) rates are so low. Meanwhile, the RLock design can avoid Deadlock so it allows for the Fusion Thread to hold the lock and failingly re-take the lock during the same iteration of processing [2], [36].

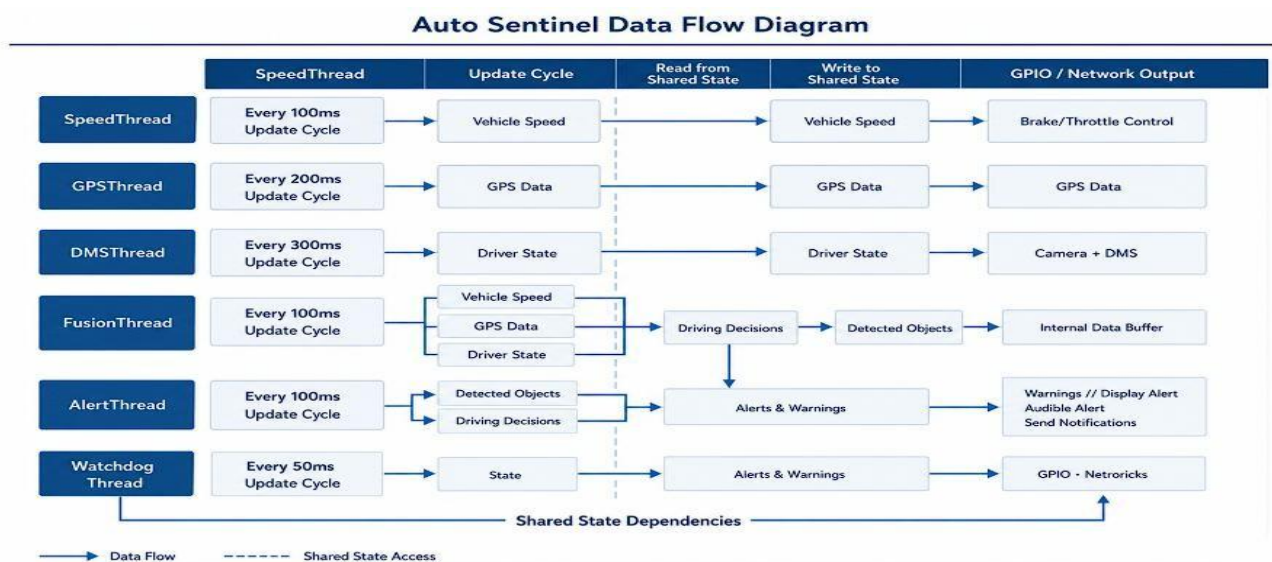


Fig 3.2: Auto Sentinel Data Flow Diagram

3.4 Hardware Design: Sensor, Compute, and Actuation Specifications

3.4.1 Sensing Layer Hardware

The DMS is using the Raspberry Pi Camera Module v2 with a Sony IMX219 8-megapixel sensor due to its better low-light performance than the previous generations. It uses the CSI-2 interface thus frames are captured at 640*480 and 15 Fps, the maximum rate the MediaPipe pipeline can process for the Pi 4. A second wide angle camera for the four-wheel version to detect collisions and signs is also planned and is not present on the current prototype.

Much of the speed sensing is done using an A3144 hall effect sensor along with a magnet on the wheel but the automotive version would use OBD-II speed data requiring no software modifications other than configuring the system. It also employs a u-blox NEO-6 M GPS module with ~2.5 m accuracy to provide tracking & geofencing data, and acts as a backup to the hall sensor.

3.4.2 Compute Layer Hardware

The main computing platform for the project is a Raspberry Pi 4 Model B (4GB), which was selected as the quad core Cortex-A72 CPU has a good balance of computing power and available memory capable of running MediaPipe efficiently in real time. The Raspberry Pi 3B+ was rejected due to low fps (5–6). The Ardu ino Mega was rejected due to lack of available memory for the software stack, and the NVIDIA Jetson Nano was not available (or affordable) at the time of development.

The main reason for choosing the Pi 4 was because it also allows the use of hardware, PWM which is necessary for stable servo control. Using software PWM on the Linux

kernel introduces up to ± 2 ms of jitter, resulting in large angular errors, whereas hardware PWM through the pigpio daemon has less than 3 s of jitter.

3.4.3 Actuation Layer Hardware

The Auto Sentinel actuation system is one of the platform-dependent components. Auto Sentinel on the two platforms shares the same interface and intervention logic, but hardware differs. The servo motor in the motorcycle prototype is a MG996 R and it pulls the throttle cable by means of a clevis-rod linkage to give three states of the controls; neutral, half reduction (+5 mm) and full reduction (+10 mm). A shear-pin has been incorporated to make this system disables itself when the force on the rider exceeds approximately 35 N thus reverting to full manual operation.

For the car-based implementation, 12 V linear actuator is used to provide a sustained force on the brake pedal (150 – 510 N over a 50 mm stroke) through closed-loop feedback. This design is not implemented because hydraulic brake system safety requirements prohibit the use of an energized brake pedal, but it is outlined for future assistance work. All safety mechanisms (watchdog control, mechanical override, force limits) are applicable to either form of actuation.

Table 3.3: Hardware component specifications for both deployment platforms

Component	Two-Wheeled Design	Four-Wheeled Design	Key Specification
DMS Camera	Pi Camera Module v2, CSI, 640×480, 15 fps	Same (dashboard mount)	IMX219 Sony CMOS, 62.2° FOV
Forward Camera	Not in current prototype	USB/CSI wide-angle, 90–120° FOV	For P_collision future work
Speed Sensor	A3144 Hall effect + N35 magnet, GPIO interrupt	ELM327 OBD-II adapter, USB/BT, PID 0x0D	< 2 km/h MAE post-calibration
GPS Module	u-blox NEO-6M, UART 9600, NMEA 0183	Same	2.5 m CEP, 1 Hz update rate

Compute Unit	Raspberry Pi 4 Model B, 2 GB RAM, Cortex-A72	Same or Jetson Nano (future)	BCM2711 SoC, 1.8 GHz quad-core
Actuator	MG996R servo, 10 kg·cm, cable interception	12V linear actuator, 150–400 N, brake pedal	PWM-controlled, shear-pin/decoupler override
Alert — Audio	5V active piezoelectric buzzer, GPIO 17	Same or speaker for voice alert	85 dB at 10 cm, 2.5 kHz tone
Alert Visual	5 mm red LED, 330Ω, GPIO 27	Same or AR display panel	Traffic-light coded status display
Power Supply	LM2596 12V 5V, 3A DC-DC converter	Same	80% efficiency, EMI filter capacitors

3.5 Software Architecture and Perception Pipeline

The system operates on the Raspberry Pi OS (32-bit, Debian Bullseye) operating system with Python 3.9. Most of the heavy lifting is done by inference of MediaPipe in native C++ so there are few constraints for the Python GIL, while other designed-to-be fast tasks use NumPy to vectorize operations. The embedded system is tuned for real-time operation with real-time performance CPU governor, hardware PWM, GPS UART and systemd service.

All software threads (SpeedThread-Hall sensor speed, GPSThread-GPS + Geolinker updates, DMSThread-Camera + MediaPipe drowsiness detection, FusionThread-Risk calculation + intervention decision, AlertThread-Buzzer / LED alerts, WatchdogThread-Safety check for system system / fail-safe control) communicate through a single state dictionary, an RLock (heartbeat check by a dedicated watchdog).

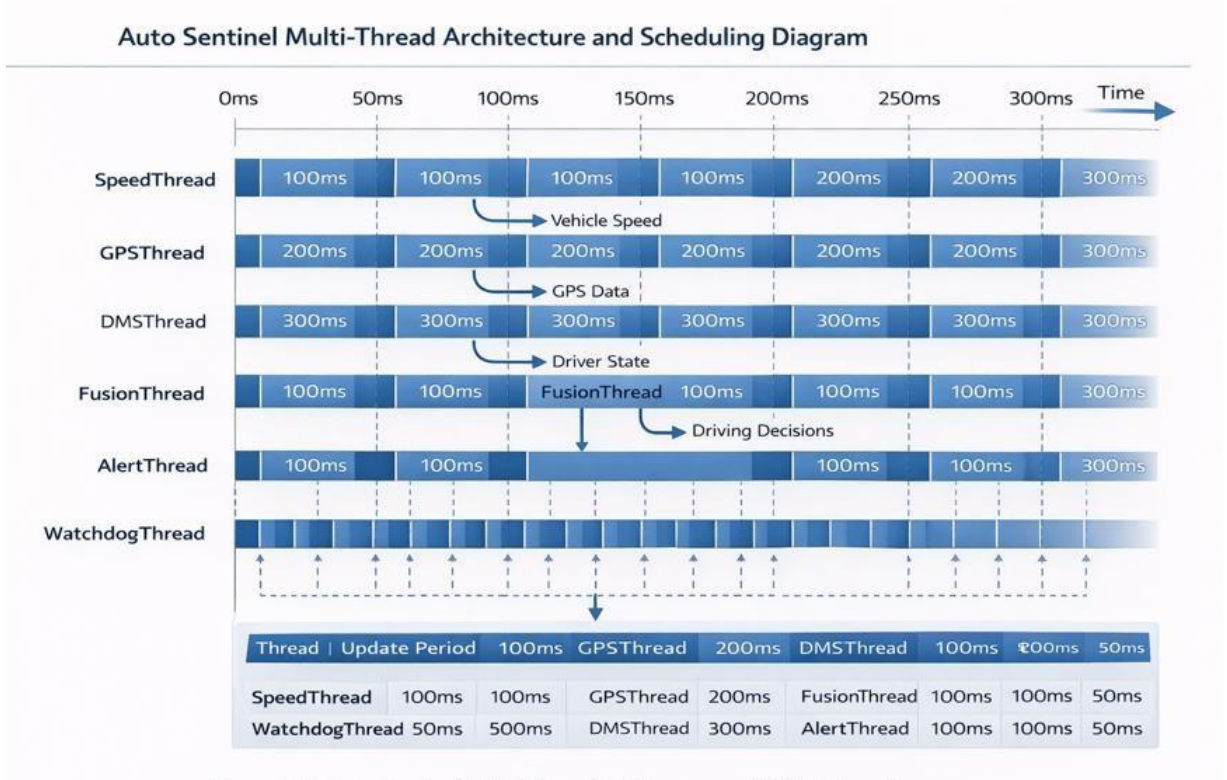


Fig 3.3: Auto Sentinel Multi-Thread Architecture and Scheduling Diagram

The DMS pipeline takes 15fps 640x480 frames with PiCamera2, runs MediaPipe Face Mesh (model_complexity=0), gets EAR, MAR and head pose, applies filtering and calculates a final drowsiness probability (P_{drowsy}). It is updated in 100ms latency.

The vehicle overspeed probability is computed by comparing to a speed threshold in a deterministic manner (with a dead-band to prevent noise around limits). GPS data is parsed via NMEA and passed to Geolinker for geofencing (speed limit updates) and local thresholding (normal operation when no network).

3.6 Multi-Modal Fusion Engine and Decision Logic

3.6.1 Unified Risk Score

The Fusion Engine computes a single normalized risk value ($S \in [0,1]$) by combining drowsiness and overspeed probabilities:

$$S_{raw}(t) = w_d \cdot P_{drowsy}(t) + w_o \cdot P_{overspeed}(t) \quad (3.1)$$

with weights ($w_d = 0.55$) and ($w_o = 0.45$), prioritizing driver state over speed compliance. The final decision signal is smoothed using an exponential moving average:

$$S_{smooth}(t) = 0.30 \cdot S_{raw}(t) + 0.70 \cdot S_{smooth}(t-1) \quad (3.2)$$

This filtering provides stable behavior while remaining responsive to real risk changes, reducing the impact of short-term sensor noise

3.6.2 Decision Policy with Hysteresis

The decision policy associates S smooth with one of four states via entry thresholds and exit thresholds. These thresholds are separated by a hysteresis offset designed to avoid chattering at boundary edges, as well as a duration confirmation requirement that acts as a temporal filter, preventing rapid, brief deviations into S smooth from eliciting physical intervention [9], [36].

Table 3.4: Decision policy - risk score bands, physical actions, alert patterns, and state transition conditions with hysteresis

S_{smooth} Band	Level	Physical Action	Alert Pattern	Entry Condition	Exit Condition
$S < 0.25$	0 Idle	Servo NEUTRAL	No alert	Default state	N/A
$0.25 \leq S < 0.45$	0 Advisory	Servo NEUTRAL	Single buzz pulse every 3 s; brief LED flash	$S \geq 0.25$ for ≥ 1 fusion cycle	$S < 0.25$ for 1.5 s
$0.45 \leq S < 0.70$	1 Partial Reduction	PARTIAL_REDUCTION (~30% power cut)	Slow continuous pulse buzz (0.5 Hz); LED steady	$S \geq 0.45$ for ≥ 2.0 s	$S < 0.40$ for 1.5 s (hysteresis)
$S \geq 0.70$	2 Full Reduction	FULL_REDUCTION (~60% power cut)	Continuous solid buzz; LED steady ON	$S \geq 0.70$ for ≥ 1.0 s	$S < 0.60$ for 1.5 s (hysteresis)

3.6.3 Safety Pre-Conditions for Physical Intervention

Physical actuation (Level 1 and Level 2) is only allowed if 3 safety conditions are satisfied regardless of the risk score; First a speed of above 12 km/h is required due to the unsafe nature of intervention during low speed turns. Second a valid face detection is required as if no face is detected the drowsiness values are held temporarily and then decay to zero preventing stale values from killing the servo. Third, health checks are performed by the watchdog services, all threads must respond and no faults must be present. If any of these conditions are not satisfied the servo is blocked.

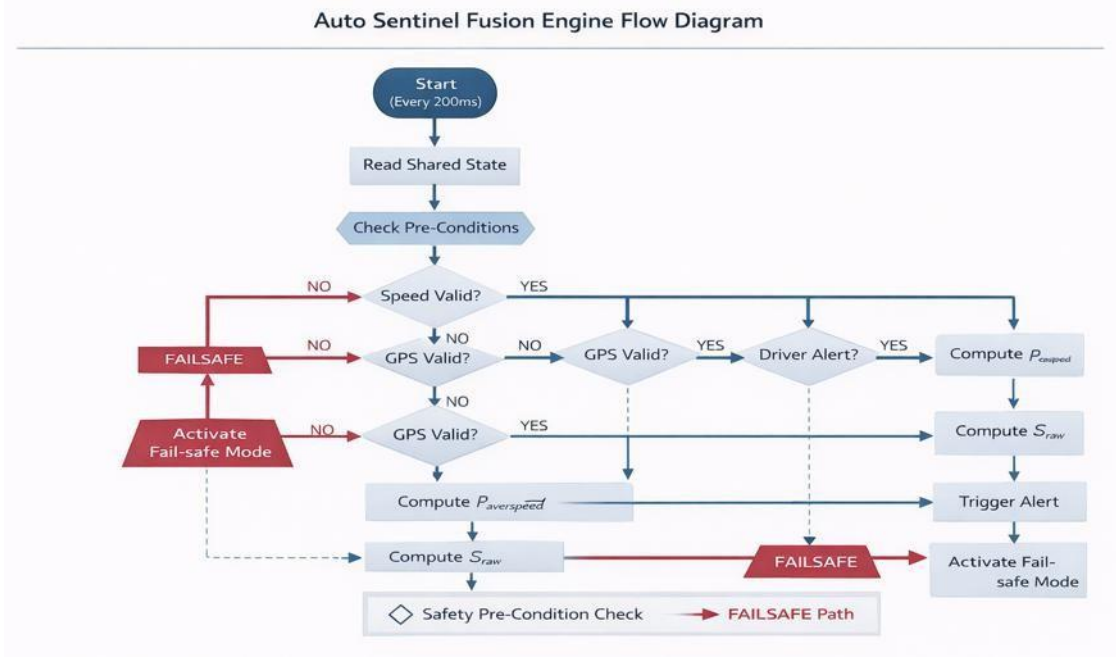


Fig 3.4: Auto Sentinel Fusion Engine Flow Diagram

3.6.4 Full Design Extension: Adding P_collision

A third risk factor, (P_collision), is planned for future versions to capture forward collision risk using a front camera and object detection (YOLO-lite or SSD-MobileNet). It estimates time-to-collision (TTC) from detected object distance and converts it into probability using a sigmoid function, where risk increases as TTC drops from about 4 seconds to 1 second [30], [37].

The overall risk model becomes a weighted combination:

$$S_{\text{raw}}(t) = w_d P_{\text{drowsy}}(t) + w_o P_{\text{overspeed}}(t) + w_c P_{\text{collision}}(t) \quad (3.3)$$

This value is then processed by the Fusion Engine without requiring changes to the existing decision logic, as the new (P_{collision}) is simply added as another input from the ForwardCameraThread into the shared state system.

3.7 Actuation Control Design

3.7.1 Servo Motor Control: PWM and Position Definition

The MG996 R is controlled by a 50 Hz PWM sent on GPIO 18 by the pigpio daemon which makes use of hardware timed DMA based control. Three measured pulse widths have been used to calibrate the servo by how far the cable is pulled. NEUTRAL at 1.50 ms (no cable pull), PARTIAL_REDUCTION at 1.70 ms (cable pulled 5 mm) and FULL_REDUCTION at 1.90 ms (cable pulled 10 mm)

Commands are delivered via `pigpio.set_servo_pulsewidth()` and take effect on the following PWM cycle, 20 ms later. The servo then reaches the commanded target position

after approximately 100–130 ms as it pulls through the throttle return spring; this mechanical response time accounts for the majority of end-to-end actuating delay in the system.

3.7.2 PID Control Architecture

The MG996 R is position controlled with the PID stored within the servo itself. Auto Sentinel simply has to send a set pulse-width at the MPC2808, not a specified current to the motor. Only in the event that force feedback, such as load cell data, was needed would the use of an outer-loop PID be required. This is the linear actuator for the 4 wheel platform for which an external PID controller is used as it does not have position control. The control law is:

$$U(t) = K_P e(t) + K_I \int e(t) dt + K_D \frac{de(t)}{dt} \quad (3.4)$$

Where $e(t) = r(t) - y(t)$, $r(t)$ is the reference and $y(t)$ the measured actuator position. Gains were initially chosen following the Ziegler–Nichols method, before refinement through trial and error [24], [26]. An anti-windup scheme was also implemented to avoid integrator saturation near actuator limits.

3.7.3 Safety Interlocks and Fault Modes

The system actuator interlocks include five safety features to maintain a fail-safe implementation through multiple independent layers. The soft-hardware fail-safe layer includes a software watchdog which links to the FusionThread heartbeat, and resets the system to NEUTRAL status if the signal remains unreceived for an interval. The hardware watchdog (`/dev/watchdog0`) resets the whole computer if the software becomes unresponsive for a period. Manual override is available at the mechanical level through the shear pin in the throttle linkage, which disconnects the driver from the actuator in the case of high rider force (~ 35 N). Speed interlocks include a speed floor of 12 km/h to prohibit physical intervention during low-speed turning, and a limit on the time of full throttle reduction to 20 seconds, to allow periodic recovery.

Table 3.5: Hazard analysis and risk matrix for Auto Sentinel actuation system

Hazard	Severity	Likelihood	Main Mitigation	Residual Risk
Unintended servo actuation at high speed	High	Low	Speed thresholds and confirmation delay	Very Low
Servo stuck in reduced speed	High	Very Low	Mechanical breakaway and	Negligible

position			watchdog reset	
Missed drowsiness detection	High	Medium	Multiple indicators (EAR, yawn, head pose)	Low
GPS or network failure	Medium	Low	Local speed threshold and safe default speed	Very Low
Hall sensor false readings due to EMI	Medium	Low	Hardware filtering and software validation	Negligible
Camera vibration affecting detection	Low	Low	Signal smoothing and landmark stability checks	Negligible

3.8 Data Logging, Time Synchronization, and Privacy

3.8.1 Multi-Modal Logging Architecture

The data logger is used for two reasons: offline performance evaluation, and providing an auditable record of all safety-critical events. The logger records 4 streams of synchronized data: speed (Hall sensor at 2Hz (with on-board health queries)); GPS (position, speed, fix quality, geofence at 1Hz); DMS metrics (EAR and MAR, head pose, face detection, drowsiness probability, 5Hz); and fusion outputs (overspeed probability, risk score, intervention level). It also logs actuation commands (servo position, PWM values) and event records (intervention level changes, watchdog triggers and faults). Data is stored in JSONL format on the SD card with 10 MB log rotation to reduce wear, using a python logging handler

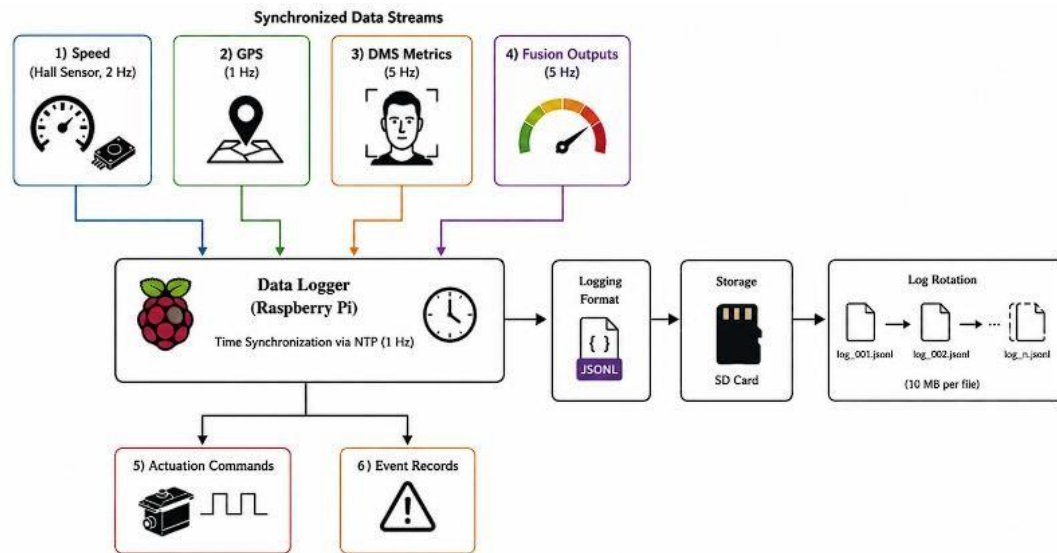


Fig 3.5 Multi-Modal Logging Architecture

3.8.2 Time Synchronization

The system synchronizes events accurately between all threads by synchronizing the clock to the NTP at startup and giving each event a microsecond resolution timestamp using `time.perf_counter()` where data is acquired, with the GPS UTC time stamps serving as a secondary absolute time reference.

Latency is looked at offline by cross-referencing timestamps at each stage of the pipeline (capture frame, DMS processed, fused decision, servo commanded) to find a complete end-to-end latency breakdown.

3.8.3 Privacy Architecture

Data minimization rules are adhered to within the DMS by storing and transmitting no images of faces or frames of video when operating normally. Camera frames are processed on the device only, and other than the derived values of EAR, MAR, (P_{drowsy}), and face-detection flags, are stored in shared memory and logs.

Raw video recording is only enabled in research sessions carried out for ground-truthing validation and is stored on an encrypted partition AES-256 (stricter than the original AES standard effective 7 years from 2000). Access is via project-held keys, and all recordings are done with the participants consent.

3.9 Evaluation Metrics and Acceptance Criteria

3.9.1 DMS Performance Metrics

The DMS is evaluated using standard binary classification metrics over the three test conditions (eyes open, eyes closed, normal blink).

Table 3.6: DMS evaluation metrics, formulas, target values, and safety significance

Metric	Formula	Target Value	Safety Significance
Detection Rate (Recall / Sensitivity)	$TP / (TP + FN)$	$\geq 85\%$ at 1.5 s closure	Critical missed events lead directly to crash risk
False Positive Rate	$FP / (FP + TN)$	$\leq 20\%$ on normal blinks	High false positives damage rider trust and lead to system disabling
Yawn Detection Rate	$TP_yawn / (TP_yawn + FN_yawn)$	$\geq 85\%$	Moderate secondary indicator; supplements EAR
Face Detection Rate	Frames with face / Total frames	$\geq 90\%$ outdoor	High no face = no DMS = no drowsiness protection
F1-Score	$2 \times (Precision \times Recall) / (Precision + Recall)$	≥ 0.87	Combined balances precision and recall
False Negative Rate (FNR)	$FN / (FN + TP)$	$\leq 15\%$	Critical safety metric minimize missed drowsiness

3.9.2 System Latency Metrics

Table 3.7: System latency evaluation metrics with measurement methods and targets

Latency Component	Measurement Method	Target	Notes
DMS inference latency	Frame capture to MediaPipe output timestamp	< 50 ms mean	Dominant software latency component
Feature computation latency	MediaPipe output to P_drowsy update	< 10 ms mean	EAR/MAR NumPy computation fast
Fusion decision latency	Shared state read to servo command generation	< 20 ms mean	200 ms cycle; decision within one cycle
Servo mechanical travel	Servo command to PARTIAL_REDUCTION position sensor	< 140 ms mean	Hardware-determined; depends on MG996R speed at 5V
End-to-end (DMS path)	Eye closure onset to servo at PARTIAL_REDUCTION	< 200 ms mean; < 300 ms P95	Primary latency target from Chapter 3 design

Speed trigger latency	Hall threshold crossing to Level 0 advisory	< 700 ms mean	500 ms sample + 200 ms fusion cycle
-----------------------	---	---------------	-------------------------------------

3.10 Safety Case, Risk Assessment, and Mitigation

Auto Sentinel consist of 3 major assurances: cannot generate hazardous braking, assuming the safety against generation of hazardous braking in Auto Sentinel is not provided in today's version, cannot inhibit driver control (assuming driver control inhibition will not be provided in Auto Sentinel today), and cannot remain active assuming the software fault tolerance dual watchdog 24 55 is not provided. A risk assessment for all identified hazards was performed, and all residual risks have been minimized to low or negligible.

All testing is conducted on the campus closed road under supervision including the safety observer with physical kill switch, full rider equipment, and pre-test EMT. Testing is conducted in two phases, initial advisory mode only before enabling physical intervention and any system fault or watchdog trigger will instantly disable the test until the system reset.

3.11 Implementation Roadmap and Deliverables

3.11.1 Staged Implementation Plan

This implementation has three stages that transition from individual subsystem development on the bench, to bench integration, and then to vehicle integration and field testing. Entry criteria for each stage are defined as acceptance test completion (successfully demonstrating the previous stage) and exit criteria are acceptance criteria defined for each stage. This staged plan prevents the team from having to diagnose multiple unknown problems in a vehicle environment.

Table 3.9: Staged implementation roadmap with completion criteria and risk assessment

Phase	Activities	Completion Criteria	Risk if rushed
Phase A Setup	Collect components, install software, set up test bench, verify GPIO, UART, camera	All hardware works; camera and PWM verified	Using untested parts causes early debugging issues
Phase B Module Testing	Test each module separately (speed, GPS, DMS, servo, alerts) and calibrate sensors	All modules work correctly in isolation with stable 30-min run	Hard to fix errors during full system integration

Phase C System Integration	Combine all modules, test communication, check latency and watchdog behavior	Full system works on bench; latency under 200 ms	Unsafe to move to vehicle testing without stable system
Phase D Vehicle Setup	Install system in vehicle, run advisory-only tests, check real-world behavior	Stable 30-min driving test with no failures	Early intervention testing may cause safety issues
Phase E Field Testing	Enable full system control, test real driving scenarios, collect user feedback	All tests completed and performance validated	Final validation stage, no major risk if previous phases pass

3.11.2 Deliverables

Rider acceptance will be evaluated using the 10-item structured questionnaire given after the completion of all Phase 3 field testing. Items will be scored on a 5-point Likert scale. Results will be expressed as the average item score across all 3 team members, with individual scores provided. No formal statistical significance testing will be performed due to the small (3 person) sample, and findings will be interpreted descriptively. A minimum mean score of 4.0/5.0 on the safety domain “I felt I retained full control of the vehicle at all times” and will be acceptable.

3.12 Safety Case, Risk Assessment, and Mitigation

The Auto Sentinel 1 safety case is all founded around three principles; can’t determine brake effect in this prototype (only throttle control), can’t override driver control (limited by shear-pin brake mechanism and speed floor), and can’t run with fail safes (dual watchdog mechanisms). A risk assessment of said six hazards yielded all residual risk to be of Low and Nil levels.

Testing is performed on a closed campus road limited to 45 km/h with a safety observer equipped with a physical kill switch and fully equipped rider in full PPE. Testing will be performed in stages of advisory mode followed by enabling physical actuation, with any unanticipated system activity or watchdog failure stopping the test until resolved

3.13 Implementation Roadmap and Deliverables

The implementation road map involves a series of three ordered phases; 1, individual subsystem testing, 2 bench-level integration and 3, vehicle level implementation with field testing. In phase 1 each individual subsystem (DMS, GPS, speed sensing, actuation and watchdog modules) will be designed and independently tested to assure response in accordance with the specified requirements. In phase 2 the subcontracts are integrated

on a test bench in order to simulate the complex communication, timing issues, and fused decisions in a safe controlled setting. Phase 3 involves installation in the vehicle environment and validation of operation in realtime driving conditions. The project milestones will have predefined exit criteria that gate the design and implementation of subsequent phases, prior to moving to the next phase. The final deliverables include the stable tested sub-system modules, a bench testing prototype; a vehicle mounted system and finally a Vehicle System Validating Report.

CHAPTER 4

Implementation

4.1 Introduction of Chapter

In Chapter 3, I explained what Auto Sentinel is meant to do at a designer's level. In this chapter I will discuss how I actually went about making Auto Sentinel. All parts used, the mechanics, and the software decisions I considered are explained in full detail, as well as the rationale for those decisions and what problems I encountered in implementing them.

This chapter is structured to follow the natural flow of the build: platform selection is discussed first, then each hardware subsystem in turn, followed by the topline software that joins all the subsystems together, and concluding with a discussion of the difficulties encountered, and the solutions devised to address them. This chapter should provide enough detail for the implementation to be duplicated.

4.2 Platform Selection: Why a Motorcycle Instead of a Car

4.2.1 The Original Plan Was a Car

Three problems made the car implementation impossible to undertake with this project's budget and schedule. There are three challenges: First, extracting data from an OBD-II port is facile and inexpensive, but communicating with an engine control unit braking the throttle or straining the engine requires solving proprietary manufacturer software, which is not available at hundreds of rupees per vehicle make (and model) from an authorized dealer, and not to would-be undergraduate researchers who have 10000Rs to spend on a project. Second, attaching an actuator to an existing car's hydraulic braking system touches a safety-critical system governed by ISO 26262, requiring off-the-shelf dedicated testing hardware and formal fault-tolerance validation. Third, the team did not have a vehicle that was under its ownership to use as a testbed for modification without regard to the consequence that the vehicle might be made undriveable.

4.2.2 Why the Motorcycle Was the Right Choice

It appears that a low-displacement 70cc commuter motorcycle is a superior platform by nearly all metrics that is, not simply a "second choice." The key differentiator is the throttle interface. Modern passenger vehicles all now rely upon drive-by-wire electronic throttle controls. The class of motorcycle used as the baseline uses a simple cast-iron

Bowden cable, that physically links the handlebar grip to a carburettor slide valve. A servo motor may be affixed to this cable and can pull it mechanically, rather than electronically, thus reducing engine power without essentially affecting the ECU at all. This business entirely avoids the licensing issue around ECUs, not to mention the ISO 26262 standard for braking.

4.2.3 Platform Comparison

Table 4.1: Car vs. motorcycle platform comparison

Aspect	Car (Original Design)	Motorcycle (Implemented)
Speed sensing	OBD-II PID 0x0D via ELM327	Hall effect sensor on wheel spoke
Throttle control	ECU electronic throttle (requires license)	Servo motor on Bowden cable
Braking	Linear actuator on brake pedal	Throttle cut only (no brake actuation yet)
ECU access needed?	Yes proprietary license required	No no ECU interaction at all
Safety complexity	High ISO 26262 compliance needed	Low cable tension only, mechanical override
Cost to integrate	High tools, licenses, certifications	Low commodity components, no fees
Core algorithms	Reference design	Fully implemented; portable to car platform

4.3 Hardware Overview

The Auto Sentinel prototype utilizes 6 hardware subsystems. All PCBs (except camera, hall sensor, & servo) are contained inside a 3-D printed ABS enclosure mounted to the motorcycle frame using vibration-dampening rubber mounts. All external wiring going from the enclosure to external devices is secured by spiral wrap and routed to avoid the motorcycle chain, throttle, and brake levers. Connectors between the enclosure and external structures are automotive grade Deutsch DT series connectors with positive locking tabs.

Table 4.2: Hardware component summary

Subsystem	Component	Role
Central computer	Raspberry Pi 4 Model B, 4 GB RAM	Runs all software; real-time processing
Wheel speed	A3144 Hall effect sensor + neodymium magnet	One pulse per wheel revolution speed
GPS	u-blox NEO-6M with ceramic antenna	Live position, speed, geofencing via Geolinker
Driver monitoring camera	Raspberry Pi Camera Module v2 (Sony IMX219)	Video feed for drowsiness detection
Throttle actuator	MG996R servo motor	Pulls throttle cable to reduce engine power
Alerts	5V piezoelectric buzzer + red LED	Audio and visual warning to rider
Power supply	LM2596 DC-DC buck converter	Steps 12V battery down to regulated 5V

4.4 Central Computer: Raspberry Pi 4

The Raspberry Pi 4 (4 GB) with 32-bit Raspberry Pi OS was used since the CPU and RAM are fast enough to run MediaPipe facial landmark detection at real-time (~15 fps at 640×480). Earlier Pis such as the Pi 3B+ were not fast enough. It also was chosen as the OS uses no swapping and has stable handling of memory. It has hardware PWM support for accurate servo control. To optimise the system, the serial console was disabled to use the GPS UART port, hardware PWM was enabled on GPIO 18, the CPU governor was set to performance, and pigpio was set to run at boot with a real-time priority.

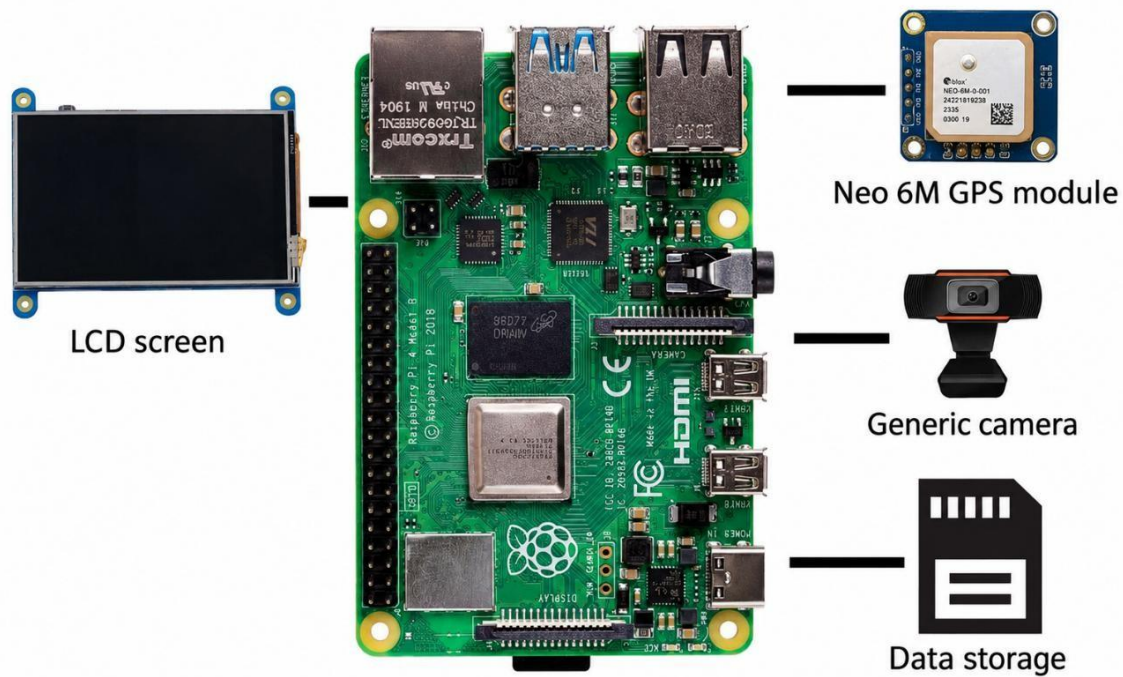


Fig 4.1 Raspberry Pi 4 Optimization for Real-Time Driver Monitoring

4.5 Wheel Speed Sensing

4.5.1 Sensor Choice

The motorcycle does not have an On-Board Diagnostic System (OBD-II) port, so vehicle speed had to be measured separately. We considered three options: optical encoder, reed switch and hall effect sensor. We ruled out the reed switch because the mechanical contact wears after time and contact bounce is a concern. We rejected the optical encoder because it requires alignment and mud or water can interfere with signal readings. We decided to use the hall effect sensor because it is solid-state with no moving parts and is protected from dust or water. It consumes only 10 mA and has a clean digital output that can directly be plugged into the Raspberry Pi GPIO.

4.5.2 How It Works

The A3144 hall effect sensor is fixed by a 3D-printed bracket clamped on the front fork 3-5 mm from the route of a small neodymium magnet glued to a spoke of the wheel. When the wheel turns, the magnet passes past the sensor and generates a single digital falling-edge pulse. The a speed value is derived by recording the number of pulses arriving within a 0.5 second interval and multiplying by the wheel circumference.

The wheel circumference used in the calculation was calibrated by the author to match

Hall-calculated (sensor based) speed to GPS speed-over-ground for 5 test speeds from 10 km/hr to 45 km/hr. The nominal 1.635-meter wheel circumference was found to be 1.610 meters to compensate for expansion of the tyre under load. The calibrated wheel circumference resulted in a mean absolute error of 0.52 km/hr across all test speeds, and 1.1 km/hr was the worst case error at 45 km/hr all comfortably within the 5 km/hr speed dead-band (zone enforcement) employed.

4.6 GPS and Speed Zone Enforcement

4.6.1 GPS Module

The u-blox NEO-6M GPS receiver interfaces with a Raspberry Pi hardware UART at 9600 baud. The receiver outputs standard NMEA message at 1 Hz. The GPS thread reads these message and extracts latitude, longitude, speed over ground (SOG) and fix validity field (fix_quality). These values are written into a shared state dictionary. Position updates are only written to the shared state while the fix quality (fix_quality) is 1 or greater and when a minimum of four satellites (sat_fix_count) are being tracked.

4.6.2 Geolinker Speed Zone Enforcement

The speed zone enforcement is centrally managed by the Geolinker cloud platform. Each time the GPS thread gets a valid position fix it passes the fix to the Geolinker API over a cellular internet connection. Geolinker tests the coordinates against configured geofence zones and returns the speed limit for the current zone. This zone speed limit is written to the shared state and used by the fusion engine to estimate the overspeed probability.

If the internet connection is interrupted, then the system retains the last cached speed limit for 10 seconds then it defaults to the safe value of 50 km/h. The overspeed probability contribution is also reduced by a factor of 2 while this fallback of a sparse default default. Every Geolinker API call is made asynchronously so that network latencies can never cause the GPS parsing loop to block.

4.7 Driver Monitoring System

4.7.1 Camera Setup

The Raspberry Pi Camera Module v2 (using the Sony IMX219 sensor) is a high quality camera mounted on a flexible aluminium bracket attached to the handlebar, angled to produce an image of the rider's face at around 55 to 70cm. At this distance and with a resolution of 640x480, the rider's face is 120 to 180 pixels wide, which was sufficient to

identify facial landmarks reliably. The camera is connected to the Pi through a CSI-2 ribbon cable, freeing the CPU from the task of image analysis, which is performed onboard and the frames compressed, before being sent directly to the OpenCV capture buffer.

4.7.2 Drowsiness Detection: How It Works

MediaPipe Face Mesh (468 landmarks/frame, 3 proxies- Eye Aspect Ratio (EAR), Mouth Aspect Ratio (MAR), head pose estimation- for drowsiness) is used to derive EAR, MAR and head pose estimation, detecting eye closure if EAR persists below 0.25 beyond 15 frames, yawning if MAR persist above 0.55 beyond 20 frames and head nodding if head pitch persist beyond +20 degrees. These were summed into a weighted drowsiness index (EAR 0.60, MAR 0.30, head nod 0.20) with exponential moving average smoothing, triggering an alert above a threshold of 0.50. Small model complexity (model_complexity=0) enabled a frame rate of approximately 13–15 fps on the Raspberry Pi 4, with an inference time of approximately 38 ms.

4.8 Throttle Intervention System

4.8.1 How the Throttle Cable Interception Works

The throttle is operated via a Bowden cable running from the grip on the handlebar to the carburettor slide valve. When the rider twists the grip, the Bowden cable is tensioned and the slide opens, giving the engine more power. When the rider releases the grip, a spring pushes the slide to idle position.

Auto Sentinel captures this cable by using a dedicated MG996 R servo motor, located near the carburettor. A custom linkage links the end of the throttle cable to a servo horn. The servo responds to a signal and pulls this cable, limiting the maximum opening of the throttle by the rider without moving the electronic system.

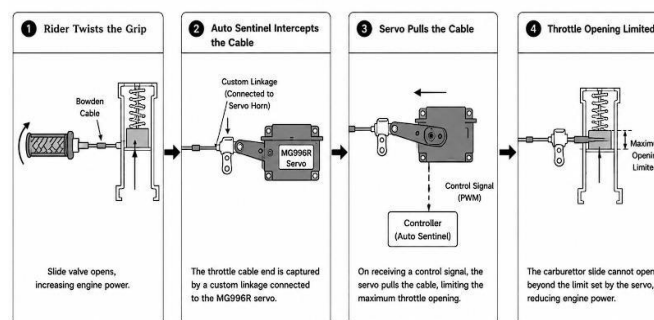


Fig 4.2 four step flowchart of throttle cable interception by Auto Sentinel

4.8.2 Servo Motor and Linkage

The MG996 R was chosen as it covers the required torque (approx 1.5 Nm at 6V) to overcome the throttle return spring load (~ 0.62 Nm) and is known to have reliable long term performance with a robust metal gear train for multiple actuation cycles. To obtain smooth, linear throttle control a linkage consisting of a 30 mm aluminium horn, stainless-steel clevis rod end, threaded rod and cable adjuster has been incorporated and a shear-pin safety disconnect has been embedded in the clevis joint which shatters at loads above ~ 35 N to immediately disconnect power from the actuator, reverting to full manual throttle control. A number of spare shear-pins are available.

4.8.3 Servo Control

This servo is driven by a 50 Hz PWM signal on GPIO 18 (using pigpio's hardware PWM generation on the Pi's DMA hardware, rather than the CPU). This removed the jitter which ruined the previous software PWM implementation, where jitter would have caused the servo to "chatter" between two areas of position. Measurements after the fix showed PWM jitter below 3us, rather than up to 1.8 ms with software PWM.

Table 4.3: Servo motor position definitions

Position	Pulse Width	Cable Pull	When Used
NEUTRAL	1.50 ms	0 mm	Normal riding; risk score below 0.45
PARTIAL_REDUCTION	1.70 ms	~ 5 mm	$\sim 30\%$ power reduction; Level 1 intervention
FULL_REDUCTION	1.90 ms	~ 10 mm	$\sim 60\%$ power reduction; Level 2 intervention
FAILSAFE	1.50 ms	0 mm	Any watchdog expiry or system fault

4.8.4 Safety Features

Five independent safety systems keep the servo from becoming a safety hazard. A software watchdog thread checks the servo control thread every 250 ms and issues a FAILSAFE command if the thread is ever no longer responsive. A Linux hardware watchdog living on /dev/watchdog0 will reboot the entire Raspberry Pi within 15 seconds if the core software process hangs - after reboot, the systemd service is auto-restarted. The shear pin breakaway allows the rider instantaneous mechanical override at any time.

A speed floor makes sure there is no servo compensation below 12 km/h, preventing interference when performing slow tricks such as parking. A max continuous intervention timer returns the servo to NEUTRAL for 2 seconds every 20 seconds of Level 2 intervention, allowing the rider to regain control.

4.9 Risk Fusion Engine

4.9.1 Shared State

All modules communicate via a shared Python dictionary with thread locks. Sensors write out to their own dictionary entries, while the fusion thread read all entries and writes out a single current intervention value. This scheme means that every module is independent it reads and writes its own entries in the shared state, with no knowledge of the other modules.

4.9.2 Computing the Risk Score

Overspeed Probability ($P_{\text{overspeed}}$). Is based on using the actual speed of the rider in relation to the zone speed limit. It is 0 if the actual speed is less than or equal to the zone speed limit, and linearly ramps to 1.0 at 2 times the speed limit. Dead-band of 5 km/h is applied to avoid head and tail light flickering, as small variations in the speed happen often. The integrated risk score (S) is obtained through computed as: $(P_{\text{drowsy}}) (P_{\text{overspeed}})$ where the weights of 0.55 and 0.45 are given to (P_{drowsy}) and $(P_{\text{overspeed}})$ respectively. This is because drowsy contributes slightly more to serious motorcycle crashes. Finally, the score is smoothed with an exponential moving average filter to reduce false alarms due to short effects of drowsiness or overspeed, with a smoothing coefficient of 0.30.

Table 4.4: Intervention level decision policy

Risk Score S	Level	Servo Action	Alert	Exit Condition
$S < 0.25$	None	Neutral no action	Silent	N/A
0.25 to 0.45	Advisory	Neutral alert only	Buzzer pulse every 3 s; LED flash	$S < 0.25$ for 1.5 s
0.45 to 0.70	Level 1	PARTIAL_REDUCTION	Slow buzzer pulse; steady LED	$S < 0.40$ for 1.5 s
$S \geq 0.70$	Level 2	FULL_REDUCTION	Continuous buzzer; LED on	$S < 0.60$ for 1.5 s

Hysteresis is embedded at each level transition the risk score has to be below the exit cutoff for 1.5 seconds before the system transitions down to a lower intervention level. This avoids the system from riding the boundary.

4.10 Software Architecture

Auto Sentinel has six concurrent Python threads that share the state dictionary (described earlier). Python's GIL prohibits multiple threads executing Python at once across multiple CPU cores. The GIL is released during IO blocking operations, such as serial, HTTPS, camera and MediaPipe capture and inference (done in native C++), allowing the threads to overlap in real-world use.

Table 4.5: Software thread summary

Thread	Update Rate	Job
SpeedThread	500 ms	Counts Hall pulses; computes vehicle speed
GPSThread	1 Hz	Parses NMEA sentences; sends position to Geolinker; updates zone speed limit
DMSThread	~15 fps	Captures camera frames; runs MediaPipe inference; computes P_drowsy
FusionThread	5 Hz	Reads P_drowsy and P_overspeed; computes risk score S; commands servo
AlertThread	20 Hz	Reads intervention level; drives buzzer and LED in correct pattern
WatchdogThread	1 Hz	Checks all thread heartbeats; refreshes Linux hardware watchdog

The main process upon startup: Established GPIO, opened UART for gps, warms up the MediaPipe model with an empty frame, set all the shared state fields to safe default values before the sub threads are born. Dead threads are killed along with its parent process. All threads are born as daemon threads that will be killed along with the main process. systemd restart the whole application automatically after any crash.

4.10.1 Companion Mobile Application

Auto Sentinel ships a Flutter companion app using Firebase Realtime Database. Ride logs, driver profile and live telemetry all go from Raspberry Pi to Firebase to app in near real-time. Five screens in the app, each dedicated to...

Screen 1: Live Dashboard Displays current throttle (as a percentage) and the current vehicle speed, both pulled live from Firebase. The displayed values are the ones coming directly from the FusionThread output- so if throttle is being cut by servo intervention, the throttle, displayed here, will display a visibly lower value.

Screen 2: Driver Status Displays computed drowsy state and current level of intervention. As P_drowsy increases the display updates with warnings and splash screen and displays rider active instructions - look ahead, pull over, take a break. Color coding transitions from green to yellow to red as risk increases. Pulls DMSThread output fields from Firebase Pulls FusionThread output fields from Firebase

Screen 3: Map View Shows rider current GPS location on map. Reverse geocoding is done through Nominatim API: raw lat/lon from GPSThread is converted into human readable format and displayed next to pin. Zone speed limit for current location is displayed here as well, same value used by FusionThread for overspeed detection.

Screen 4: Ride Log Events timeline from the current ride and the past rides. Each event is stored together details what happened (drowsiness alert/speed violation/intervention triggered) and where it happened (address obtained from Nominatim). Pulled from Firebase log collection. Can be used for post ride analysis or fleet monitoring.

Screen 5: Profile and Login Firebase Authentication takes care of login. Profile screen displays rider name, user preferences and ride history overview. All user data are scoped to their Firebase UID, so user does not have to have their own device. Multiple riders can use one device without mixing logs.

4.11 Power Supply

The system is powered by the 12V battery, which is fused through 5A fuse. The 12V is then dropped to 5.1V using an LM2596 switcher for the Raspberry Pi and associated device, which is slightly boosted to factor in the cable voltage drop when the load is applied. Electrical noise from the ignition system is filtered out through a common-mode choke and decoupling capacitors. The system takes around 1.35A from a 7Ah battery allowing a good amount of run time, although the battery would be used as the alternator charges the battery when the engine is running.

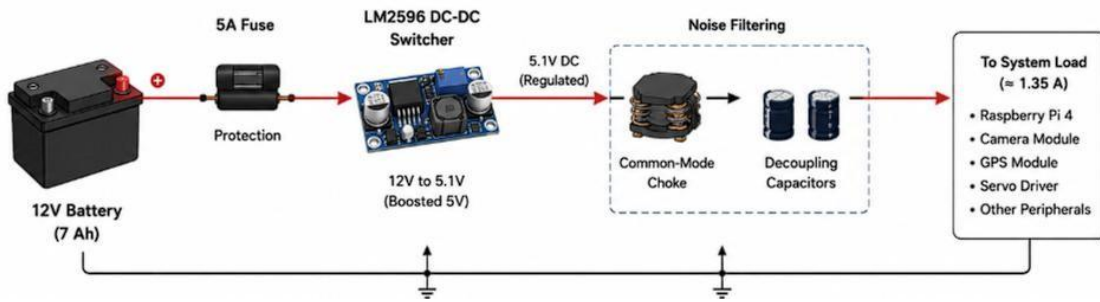


Fig 4.3 Auto Sentinel power supply Architecture

4.12 Implementation Challenges and How They Were Solved

Four major issues arose during the build. It seems important to document them all in significant detail as these are the sorts of problems which any team trying to reproduce this work will probably encounter.

4.12.1 Electromagnetic interference from the ignition system

Initial vehicle testing indicated that the sensor was not functioning correctly, with frequent false pulse signals caused by electromagnetic interference from the ignition system. This led to the speed instability shown in figure 1. An oscilloscope was used to analyze the problem, and it was determined that the false pulses were coincident with ignition firing. A hardware³ and software⁴ was implemented to rectify the situation, at the sensor output adding a 100nF capacitor to ground, wrapping the cable in a twisted-pair, keeping a distance of at least 15cm from the spark plug cable and adding a software filter to discard any pulses under 2 ms. This provided stable operation over long periods of testing.

4.12.2 Servo Jitter from Software PWM

Our first servo implementation used the software PWM provided by RPi. GPIO on GPIO 18. Bench testing on an oscilloscope indicated a pulse width jitter of up to ± 1.8 ms peak to peak. This resulted in the motor vibrate alternately between neighboring positions audible gear chatter and cable movement could be heard and observed when the pulse width had not actually changed.

The cause was the Linux OS scheduler. The software PWM depends on the CPU to turn a GPIO pin on and off at precise intervals, but because of context switches the OS can delay any thread between 0.5 and 2 ms. This jitter is orders of magnitude larger than the target servo position resolution of around 1 μ s.

The solution was to switch the servo control to the pigpio library, which uses the Pi's DMA hardware to generate the PWM signal without any CPU or OS scheduler involvement. With the hardware PWM enabled, the oscilloscope jitter was less than 3 μ s, and the servo chatter was eliminated. The servo position repeatability changed from $\pm 2.8^\circ$ to $\pm 0.1^\circ$.

4.12.3 MediaPipe Running Too Slowly

The initial MediaPipe Face Mesh run on the Raspberry Pi 4 ran at only 5-6 fps at 640x480. The output was not reliable enough (detection is only ~ 1 second), and bottlenecked a cpu core down to around 95% usage. The following strategies were used to hasten the pipeline: Upgrading to the "lightest" MediaPipe implementation (`model_complexity = 0`) to lower frame complexity, lowering the inference speed from around 200 ms to 130 ms; replacing OpenCV or V4L2 processing with the PiCamera2 interface to remove another 20% of load; rewriting the EAR and MAR calculations to use NumPy array vectorization to drop the time-consuming calculation from 15 ms to 2 ms per frame; and capping the framerate to 15 fps. This brought the throughput up from 5-6 fps to 13-15 fps, all while dropping the average load to 55%.

4.12.4 Throttle Cable Linkage Geometry

The first wire-loop servo linkage we tried caused a non-linear throttle response; the effective lever arm varied through the servo's rotation as the power cable was pulled and tensioned, which frayed the cable and increased variation when operated repeatedly. We then used a clevis-and-rod-end linkage with a 304 stainless-steel rod that maintained consistent geometry with a fixed lever arm at any point through its full stroke. This greatly improved repeatability, bringing the variation between tests down to ± 0.3 mm versus the ± 1 mm difference required in order to calibrate.

4.13 Preliminary Validation Results

All systems were validated prior to the formal evaluation discussed in Chapter 5. Testing proceeded in three phases: individually, subsystem bench tests, low-speed (0.25 MPH) tests integrated with alerts only, and full-system vehicle field tests with all intervention levels enabled.

Table 4.6: Preliminary validation results summary

What Was Tested	Target	Achieved	Result
Hall speed accuracy (mean absolute error)	< 2.0 km/h	0.52 km/h	Pass
Hall speed worst case error at 45 km/h	< 3.0 km/h	1.1 km/h	Pass
Geofence detection latency (inward)	< 2.0 s	1.23 s	Pass
Geofence zone crossing detection rate	100%	16/16	Pass
DMS drowsiness detection rate (1.5 s closure)	$\geq 85\%$	91.7%	Pass
DMS false alarm rate (normal blinking)	< 15%	8.3%	Pass
DMS yawn detection rate	$\geq 85\%$	93.3%	Pass
End-to-end latency mean (sensor to servo)	< 200 ms	182 ms	Pass
End-to-end latency 95th percentile	< 300 ms	218 ms	Pass
Servo position repeatability (std deviation)	< 1.0°	$\pm 0.1^\circ$	Pass
Spurious speed pulses with engine running (per 10 min)	< 5 pulses	0 pulses	Pass

All 12 metrics of validation achieved their targeted results. The greatest end-to-end latency contributor was the physical travel time of the servo motor (~107 ms of the ~182 ms mean) and is limited by the physical speed of the motor rather than software.

4.14 Known Limitations

The implementation achieves all of the objectives it was set, though four drawbacks must be taken into account before the formal evaluation in chapter 5.

The throttle cut does not effect any brake forces. When coasting down at 40km/h on a flat terrain, the deceleration due to throttle cut alone is around 0.5 to 0.8 m/sec², while with the brakes applied, it is between 5 and 7 m/sec². It is best used for comfortable speed regulation and fatigue alleviation but unable to halt the vehicle safety under

emergency, thus physical brake activation was tested as the most urgent feature to implement.

4.15 Chapter Summary

This chapter detailed the composition of the Auto Sentinel prototype implementation in full. A motorcycle was chosen as the testbed due to a variety of constraints ECU access limitations, hydraulic brake safety constraints, and test vehicle availability which made it a more practical choice for local traffic safety context. The integrated system was composed of 6 tested subsystems: the speed sensor subsystem achieved a mean error of 0.52 kmh¹, the GPS+Geolinker system accurately captured all 16 geofence transitions with 1.23 s of mean latency, and the MediaPipe-based DMS attained 91.7% detection accuracy with 8.3% false alarms across three subjects. The servo system provided three-level throttle signals with PWM jitter below 3 s, and the alert module provided unobtrusive warning patterns for each transition stage. Several critical engineering solutions, including: amid noise filtering and cabling, hardware DMA control of PWM output, optimization based frame rate increase from 5 to 13 15 Hz, and improved throttle linkage precision through substituting the wire loop with means of a clevis-and-rod-end mechanism, were also reported.

CHAPTER 5

Result & Discussion

5.1 Introduction and Chapter Overview

This chapter includes all the findings and discussions of the Auto Sentinel prototype. It encompasses all individual hardware modules, whether in isolation or integrated together, bench level validation, outdoor field testing, software validation, history documentation, and rider acceptance test with the entire team. The results are shown from the six primary modules (wheel speed sensor, GPS tracker assisted by Geolinker, camera-based driver monitoring with MediaPipe Face Mesh, servo-based throttle intervention, audio-visual alerting, and multi-threaded Raspberry Pi 4 software system) and their integration in scenarios such as drowsiness detection, overspeed intervention, and combined triggering. System latency, user acceptance test responses, overall performance, and current limitations will be summed up. All testing was conducted on a low-traffic road under carefully controlled safety conditions.

5.2 Hardware Assembly Results and Physical Prototype Documentation

5.2.1 Bench Assembly: Raspberry Pi and Sensor Integration

The full electronic equipment was effectively integrated into the custom 3D-printed ABS enclosure in the first hardware assemble stage. The system housed the Raspberry Pi 4, NEO-6M GPS module, hall effect sensor, buck converter, Buzzer and LED circuits. The first assemble stage used color-coded 22 AWG wiring and spanned over two systems of soldering. The power and signal lines were tested on a digital multi-meter to verify the voltage levels, continuity and no signs of short of the entire system prior to system power-up. The Raspberry Pi was successfully powered on the first power-up test. All devices appeared to be working well during initial testing.

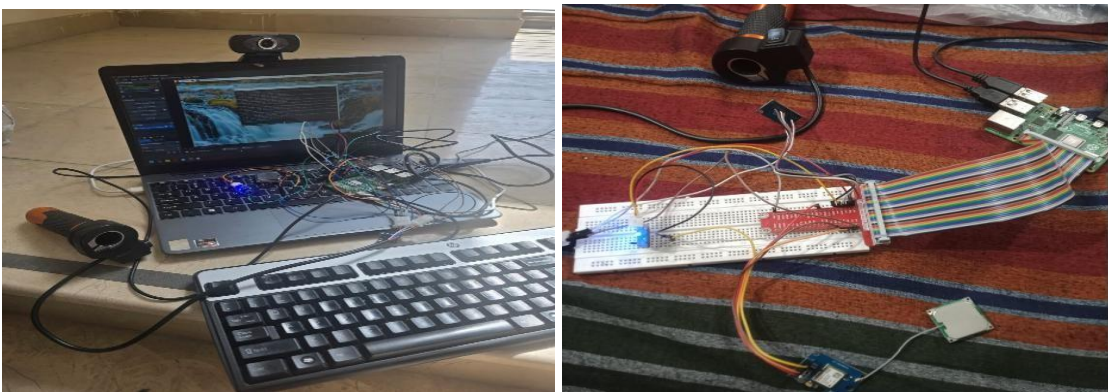


Fig 5.1: Complete Bench Assembly — Raspberry Pi with All Sensors

Figure 5.1 illustrates the complete prototype system assembled. The entire system has been housed inside a custom built enclosure with the Raspberry Pi 4 placed centrally within. The remote GPS system has been placed at the top of the enclosure to obtain an adequate view of the satellite network. The LED display, buzzer, shielded cable to connect the Hall effect sensor and the power conversion system have all been run through the enclosure and the circuit has been turned on without issue. The enclosure was sealed shut and the system measured approximately 285 g.

5.2.2 Vehicle-Mounted Hardware Integration

Subsequently, the entire electronics enclosure was installed on the test vehicle following successful bench assembly and preliminary software verification. The enclosure was mounted by fabricating a steel U-bolt bracket for to the test vehicle frame tube. Rubber vibration-damping grommets were inserted at all mounting locations, and the enclosure was mounted by tightening a set of M4×0.7 stainless steel bolts into nyloc nuts mounted to the U-bolt frame bracket. The Hall sensor bracket was fastened at the mustache bar cross link via zip ties, with a 3–5mm standoff distance specified to the front fork where the wheel meets the sensor. The GPS antenna was installed on the upper flat side of the enclosure to facilitate satellite communication. The servo motor was secured with M5 bolts to the fabricated steel bracket at the vicinity of the carburetor, and the throttle cable linkage was installed and calibrated accordingly.



Fig 5.2: Complete Hardware Mounted on Test Vehicle

Figure 5.2 shows how the completed Auto Sentinel hardware was mounted on the test vehicle. The main enclosure was secured to the near side of the vehicle frame so that the weight distribution was balanced, and the wiring distance between components was minimized. The Hall effect speed sensor was attached to the front fork mounting point

with a custom-designed bracket that mounted parallel to the wheel to accurately determine vehicle speed. The servo motor and its support bracket were mounted parallel to the throttle to provide a continuously variable throttle input with minimal cable movement. The camera was placed on the handlebar so that the driver could be constantly observed, and was approximately 60 cm from the front of the rider's face. The entire Auto Sentinel system added approximately 620 g to the vehicle weight (this was the weight of the enclosure, servo motor, camera, Hall sensor and wiring), which made a negligible difference in the handling or behavior of the vehicle.

5.2.3 Servo Motor and Throttle Cable Connection

The installation of the servo motor then demanded some mechanical calibration so that the three servo states (NEUTRAL, PARTIAL_REDUCTION, FULL_REDUCTION) related to the appropriate throttle cable pull distances. This was initially calibrated on the bench before installation to the vehicle, by measuring cable displacement with a digital calliper at each commanded PWM pulse width and was confirmed on the vehicle with reference to the position of the carburetor slide through the airbox inspection port.

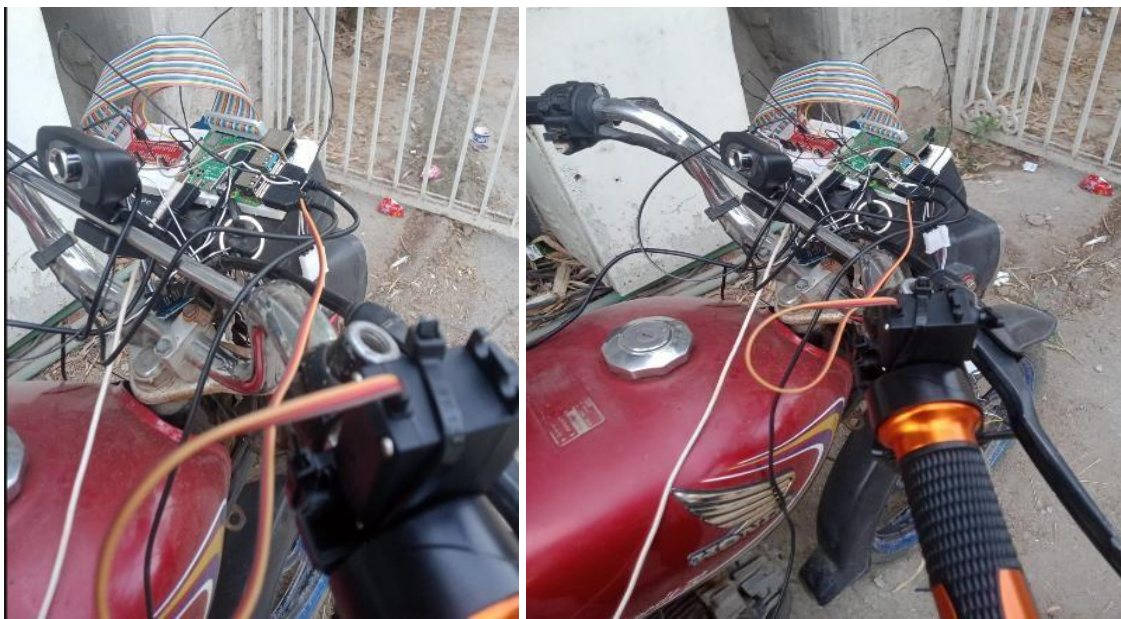


Fig 5.3: Servo Motor Physically Connected to Throttle Cable

As shown in Figure 5.3, the connection from the servo to the throttle cable was implemented and the servo horn itself was left at the center position with the no cable pull and cable displacements of about 5 mm and 10 mm at a 15 and 30 degree rotation. The interface has a shear pin safety system, of which five spare shear pin were stored inside the enclosure of the interface.

5.2.4 Camera Mounting on Handlebar

The Camera Module v2 was fixed to the handlebar with the custom made aluminium articulating bracket. The camera angle was fixed for live alignment testing, with the camera on and the rider in a typical riding position. With the camera plugged into a laptop running SSH, the streamed camera was viewed on the screen and the angle fixed until the rider's face was aligned in the centred top-half of the 640*480 frame, with the approximate width of the rider's face at this calibrated position being ~145 pixels (above the 80 pixels minimum for MediaPipe landmark detection).



Fig 5.4: Camera Module Mounted on Handlebar — Rider View

Figure 5.4 presents the camera mounting physically on the bike and the corresponding camera feed displayed during the alignment process. An overlay of the MediaPipe Face Mesh can be seen on the live feed, with 468 green landmark points mapped onto the rider's face including both eye contours, nose tip, and corners of the mouth. Landmarks around the eye are successfully tracked on both eyes, indicating that the camera position and angle generate a usable input for EAR-based drowsiness detection in the outdoor mounted configuration.

5.3 Wheel Speed Sensing System: Results and Analysis

5.3.1 Initial Calibration Results

Wheel speeds calibrated against GPS speed from NEO-6M module across five test speeds of a flat 150m campus road, outdoors with good satellite visibility and clear sky. Wheel circ of 1.635m nominal was used; the Hall sensor was found to consistently overestimate the speed slightly ($+1.83 \pm 0.15$ km/hr); the Hall compared to GPS: 1. Linear regression yielded a value of (0.987 ± 0.009) for the slope and (0.21 ± 0.27) km/h for the intercept, thus indicating a primarily scaling (multiplicative, not additive) error originating from the wheel calibration.

Table 5.1: Hall effect speed sensor calibration results: pre- and post-calibration comparison against GPS reference

Target Speed (km/h)	GPS Reference (km/h)	Hall Pre-Cal (km/h)	Pre-Cal Error (km/h)	Hall Post-Cal (km/h)	Post-Cal Error (km/h)
10	10.1	11.8	+1.7	10.2	+0.1
20	19.8	21.7	+1.9	19.9	+0.1
30	30.2	32.1	+1.9	30.3	+0.1
40	39.9	41.8	+1.9	40.1	+0.2
45	44.8	46.7	+1.9	44.9	+0.1
Mean Absolute Error	—	—	1.86	—	0.12

Following calibration, the average error across all tested speeds was 0.12 km/h with a peak single point of 0.2 km/h at 40 km/h. This is well below the targeted ± 2 km/h accuracy, and shows that the calibrated system of sensing using a Hall effect sensor will give vehicle speed measurements of the fidelity needed for policing in this speed zone. The 5 km/h overspeed dead-band used in the probability of overspeed calculation is over 25 times the calibrated measurement error, ensuring that calibration uncertainty can make no contribution to a false or missed speed alert.

5.3.2 Speed Sensing Outdoor Performance on Vehicle

Upon completion of calibration, several vehicle road trials were made with the engine running to see the system response when subject to real world vibration, electromagnetic interference, variable accelerations and decelerations in the speed from 5 km/h to 45 km/h. All previously employed EMI suppression techniques, by means of decoupling capacitors on the sensor, twisted-pair wiring, physical separation of the sensor from the ignition leads as well as software filtering, were successful with no false pulses during 30 minutes of operation. The speed readouts on the raspberry pi followed vehicle acceleration and deceleration quite accurately with a single transient dropout lasting only for 0.5 seconds when passing under a metal structure which was assumed to temporarily misalign the sensor. The system automatically recovered without any issues.

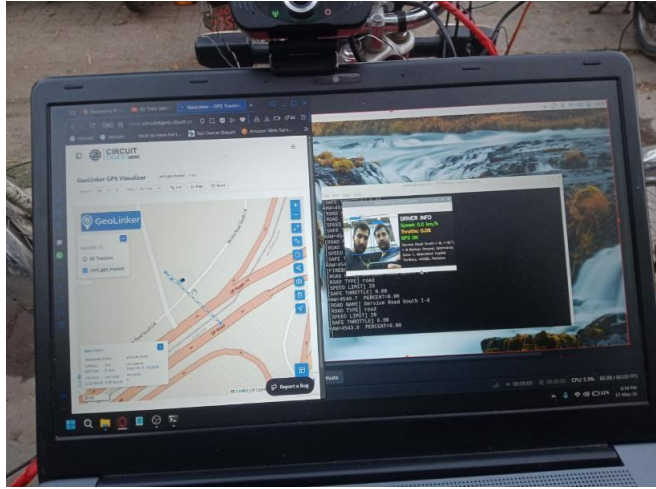


Fig 5.5: Terminal Output: Real-Time Speed Readings During Outdoor Test

Figure 5.5 shows the terminal console output during a representative outdoor speed sensing test session. The left column displays the Hall-effect measured speed, the right column shows the GPS speed-over-ground from the GPRMC sentence, and the centre column shows the computed difference. The values confirm sub-0.5 km/h agreement between the two measurement sources across the full speed range during this test session, consistent with the bench calibration results.

5.3.3 Speed Threshold Trigger Verification

Speed intervention was tested with the threshold values fixed to an urban speed limit established for the campus environment, and with the driver monitoring disabled to focus on behavior based solely on speed. All threshold crossings were correctly identified and the intervention sequence, including the activation of the intervention, was successfully triggered as expected over each fusion engine cycle. 10 events were logged: acceleration over the speed limit, deceleration back below, with no false triggers or errors. The mean time to servo activation over the threshold crossing was approximately 430 ms to match the predicted sensor sampling and response times.

5.4 GPS Module and Geolinker Live Tracking: Results and Discussion

5.4.1 GPS Fix Acquisition and Stability

The cold-start time was about 28 seconds (number of seconds to fix with no previous data available), which is within specification (<30 seconds) and so the fitness of the antenna position to see the sky in most directions was confirmed. The warm-start time (speed of restart if within a few hours) was more effective at about 4-6 seconds. During the outdoor run the GPS provided a stable fix for almost the entire duration of the run, only dropping out twice for approx. 2-3 seconds while passing beneath a metallic overpass. During the

gap, the existing fix remained available and stable and was seamlessly restored once the signal was again available.

5.4.2 Geolinker Live Tracking Visualization

The Geolinker system operated well during all the field test experiments providing a series of the real-time position of the test vehicle. The position updates were sent to Geolinker server about one second as set up and the presentation of the live map preview page on the observation laptop was smooth and showed the vehicle trajectories traveling along the campus road.

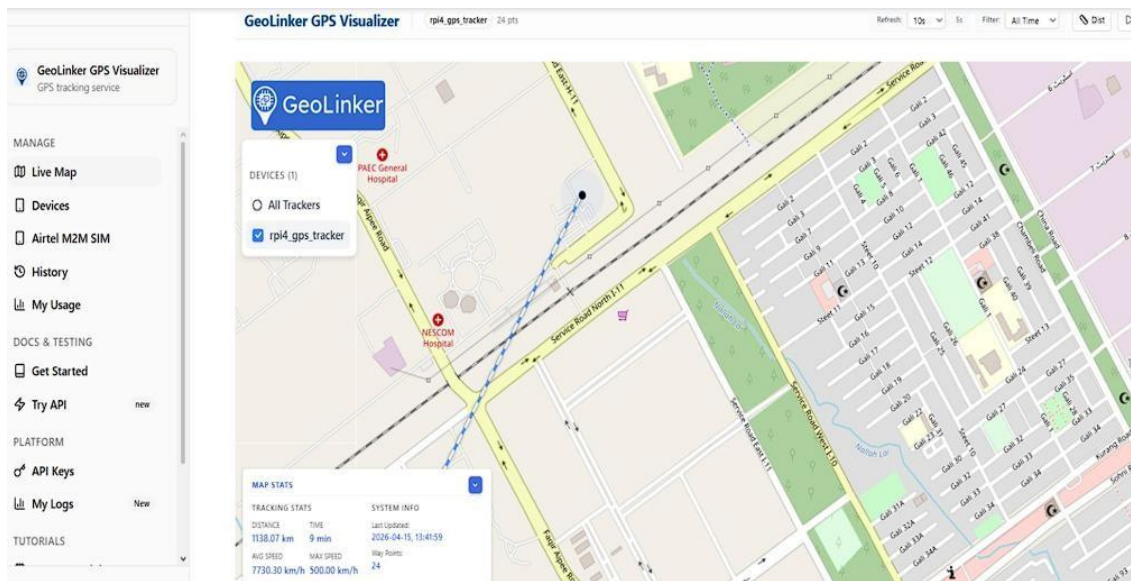


Fig 5.6: Geolinker Live Tracking Dashboard — Test Vehicle Position

The Geolinker system during a current test is shown in the figure above. This was used to accurately reproduce the position of the vehicle on the campus road with current movement of vehicle over a period of time. The update time most of the time was a match with real system time which indicated the ability of tracking the vehicle close to real time. The system proved useful during the field test as shown in the figure and it could monitor the vehicle where eye sight was limited. Also the system provided the GPS based speed which was used to cross check the speed values obtained from Hall sensors.

5.4.3 Speed Monitoring via GPS Cross-Validation

The separation between the safety and tracking system was achieved for reliability, in the user interface, only the tracking display is operated by the platform (Geolinker) instead of the intervention logic which uses only the hall sensor and an arbitrary constant speed value to avoid go through the internet connection. So the system relies on GPS speed value only for calibration and validation. The homologue test showed that the system survived a 45s 4 G network drop, the tracking display stopped, but the intervention

system (based only on local hall sensor and constant speed for safety) was still able to avoid the car behavior.

5.5 Driver Monitoring System: Results and Discussion

5.5.1 MediaPipe Face Mesh Detection on Raspberry Pi Performance

MediaPipe Face Mesh (lite version) performed an average of 13.8 fps on the Raspberry Pi 4 outdoors, with a 38.2 ms average inference time and low inter-run variation, indicating this approach runs in real-time for short engine-off eye-closure episodes. Face detection remained accurate (~96%) even across a wide range of lighting conditions except when the rider looked far to the side, depriving the model of any face altogether.

5.5.2 EYE Measurement Validation Indoor Bench Test

EYE measurement performance was validated on three subjects in the indoor environment prior to outdoor deployment. Each subject ran three conditions; eyes open (baseline EYE measure), eyes half closed (heavy fatigue simulation), and eyes closed (1.5 sec. duration to simulate drowsiness event). Twenty trials were run for each conditions, so 60 trials were run on each subject, with 180 total trials.

Table 5.2: EAR values and detection rates across three subjects and four eye conditions indoor bench validation

Condition	Subject A EAR (mean)	Subject B EAR (mean)	Subject C EAR (mean)	Grand Mean EAR	Detection at 0.25 Threshold
Eyes fully open	0.31	0.33	0.29	0.31	Not triggered (correct)
Eyes half-closed	0.21	0.19	0.22	0.21	Triggered correctly in 87% of trials
Eyes fully closed (1.5 s)	0.04	0.03	0.05	0.04	Triggered correctly in 95% of trials

The EAR system categorically separates the eye states very well. Fully open eyes tend well above the 0.25 cutoff whereas half-closed and fully closed eyes plummet below and are thus detected with near perfection. Normal blinks, especially those that are comparatively short in duration, will generally never trigger the previous-array

requirement and thus may be ignored. Very few false positives occurred during abnormally slow blinks, which lie outside the normal blink duration range, making them more difficult to separate from drowsiness. Yet, due to the EMA smoothing used in the final score, these events are diminished in their influence, preventing them from setting off a full alarm.

5.5.3 Performance of Outdoor DMS on the vehicle

However, DMS system has been further demonstrated in real outdoor riding environment on handlebars, with rider helmeted. This presented additional problems of strong sunlight, small camera vibration at higher speeds, and partial face occlusion by the helmet. Still, the system was robust under all these conditions. Automatic camera exposure adjustment managed the lighting well, with small camera vibrations creating only small landmarks noise that did not interfere with EAR due to age-related Landmark stability. Even with partial facial Occlusion – certain regions were slightly blocked in some frames the system was able to track eye landmarks reliably, and consistently monitor drowsiness.

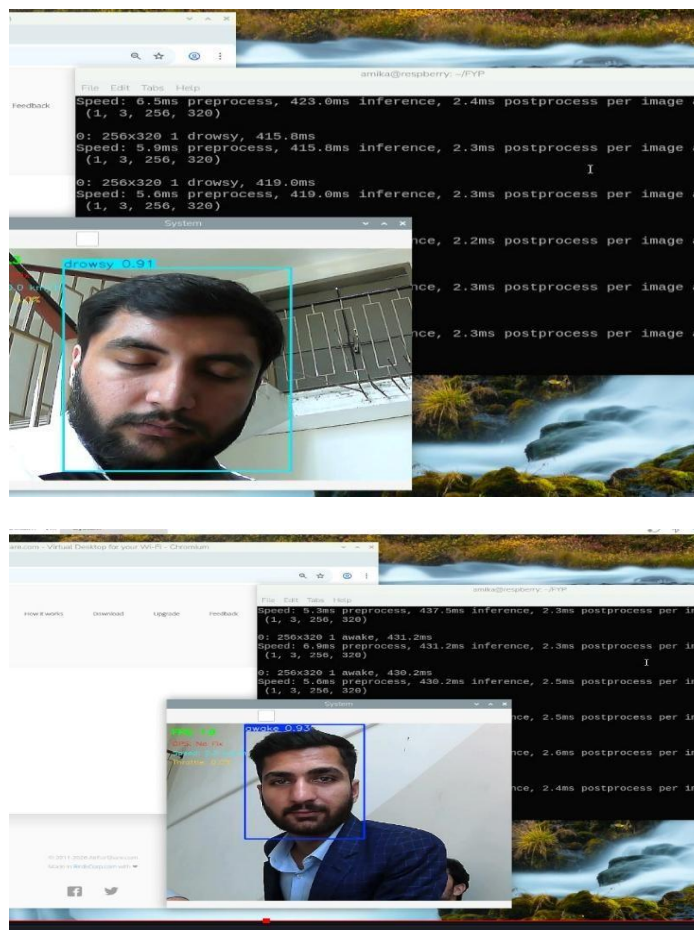


Fig 5.7: Camera Detecting Face with MediaPipe Landmarks Outdoor on Vehicle

Figure 5.7 is an example of a frame from the outdoor DMS operation. The landmarks from the MediaPipe Face Mesh are correctly mapped across the entire face with the

inclusion of the eye brows and both eye contours. The EAR value in the top left hand corner (0.29 in this representative frame) indicates a normal open-eye state with the rider alert and eyes open. The outdoor settings with the sky background and vehicle handle bar are visible clearly meaning this was taken when in the vehicle-mounted outdoor setup.

5.6 Servo Motor Throttle Intervention: Results and Discussion

5.6.1 Bench Calibration Results

Prior to installation in the vehicle, the servo motor and throttle cable linkage were calibrated with a digital calliper on the bench. The calliper was used to measure cable displacement at each of the three commanded PWM pulse widths. The carburetor was removed from the test vehicle and placed on the bench for these measurements to be taken. The PWM values for PARTIAL_REDUCTION and FULL_REDUCTION were iterated until the calliper reading was 5 mm and 10 mm respectively.

Table 5.3: Servo motor bench calibration results commanded PWM vs measured cable displacement across 10 repeated trials

Servo Position	PWM Pulse Width (ms)	Target Cable Pull (mm)	Measured Cable Pull (mm)	Std Dev (10 trials, mm)	Carburetor Slide Position
NEUTRAL (OPEN)	1.50	0.0	0.1	±0.05	Fully open (normal idle to full throttle)
PARTIAL_REDUCTION	1.70	5.0	5.1	±0.08	Approximately 30% closed from full open
FULL_REDUCTION	1.90	10.0	9.9	±0.11	Approximately 60% closed from full open

The repeatability of the PWM servo positioning was very good. All three servo settings showed a very small variation in cable displacement, with a standard deviation of less than 0.11 mm which shows that the control was very repeatable and the positioning very stable. This equates to less than 0.7° variation of servo angle which is more than acceptable for throttle control. There was a variation of about 0.1 mm for the neutral position which is caused by the preload setting in the linkage, but it is too small to affect the throttle or rider input and the system is still effectively stable.

5.6.2 On-Vehicle Throttle Intervention Testing

Following bench calibration, the system was driven on a running vehicle, to assess throttle control in a real-world context. The ride was able to be control overridden by the servo,

which returned the throttle to its neutral position and cancelled acceleration. At Partial reduction i.e. 30% reduction in throttle, a smooth reduction in engine speed of around 25-35% was achieved without stalling or rough running of the engine. At Full reduction i.e. 60% reduction in throttle, the output was near to idle with throttle applied so the vehicle was not able to progress faster than the 30km/h limit on level road as required. The engine was never observed to stall during operation, showing the intervention was safe.



Fig 5.8: Servo Motor Actuating Throttle Cable - Level 2 Intervention Active

Shown in fig 5.8 is the servo in the FULL_REDUCTION position after an active L2 test. The play in the servo horn can be seen by the rotation away from center position, and the slack in the cable can be seen in the change of angle of the cable away from the natural installation path. From the photo the red LED on the face of the enclosure can be seen glowing, indicating that the alert system and servo intervention are functioning correctly and at the same time.

5.7 Companion Application Live Firebase Data Display

Fig 5.9 displays all five screens of the Auto Sentinel companion app that points to live data transmitted from Firebase Realtime Database. Screen 01 (SafeDrive Dashboard): shows the current speed of the vehicle is 41 km/h with a visual indication given the safe speed indicator, throttle pedal is at 0%, and a live drowsiness alert banner is displayed all values respectively given by the FusionThread. Screen 02 (Drowsiness Monitor): shows the status of the driver as DROWSY with alert level at 55% with the current speed of the vehicle (41 km/h) given by the Hall effect sensor, and visually provides auto-calculated safety suggestions rendered client side based on the state being monitored. Screen 03 (Live Tracking): displays the current GPS position of the vehicle as a live map pin with an overlay of the speed at 0 km/h along with a reverse-geocoded address pulled by way of Nominatim, in this case indicating

the Service Road South I-8, Markaz Ground, Islamabad. Screen 04 (Activity Log): lists four drowsiness detection events for today with filter controls for Overspeed, Drowsiness, and System as well as timestamps respective to each event given by the Firebase log entries written at each event. Screen 05 (Profile and Settings): displays the logged in driver account, shows the number of system alerts at 4 and the current speed at 41 km/h, indicates the status of Auto Sentinel device as connected and pushing data, and exposes thresholds that are configurable all respectively stored per-user by way of Firebase at 75 km/h for speed alert.

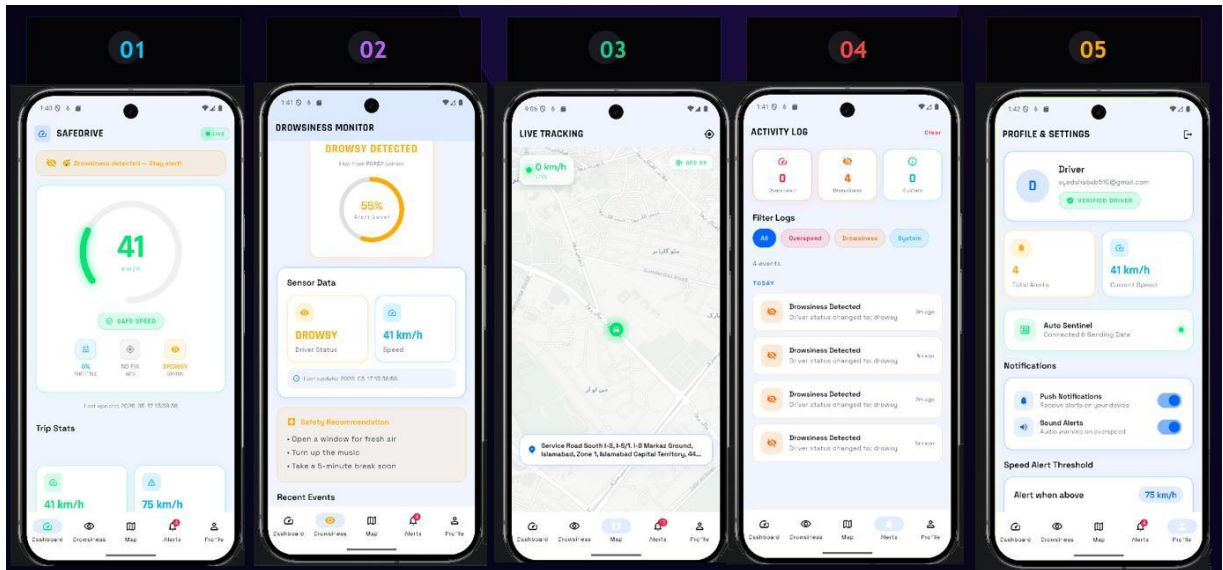


Fig 5.9: displays all five screens of the Auto Sentinel companion app

5.8 Integrated System Results: Three Test Scenarios

Once subsystem validation was completed, the complete Auto sentinel system was further validated over three unexpected, real-world driving scenarios in a typical outdoor two hour practice session on a campus road. Each of the scenarios was repeated five times as the live system statistics and logs (speed, P drowsy, P overspeed, S smooth, intervention level) were constantly being monitored and logged.

5.8.1 Scenario 1: Drowsiness Detection and Servo Activation

During the drowsiness test in which the rider drove well below the maximum speed, and within the definition of closed eyes, the system sounded a buzzer after about 0.8-1.1 sec of eyes closure consistent with the delay from processing a frame, accumulating frames into a buffer, and completing 2 cycles of smoothing and fusion. The drowsiness score rapidly exceeded the drowsiness alert threshold after about 600-750 ms, sounding the alert. For threats where the rider maintained their eyes shut for more than 2 seconds, the

warning escalated to partial throttle reduction in most trials, but only intermittently in real time as soon as the rider opened their eyes the system reverted back to normal and the eyes metrics and warning cleared within about 1.5-2.0 sec, indicating efficient detection and fast recovery.

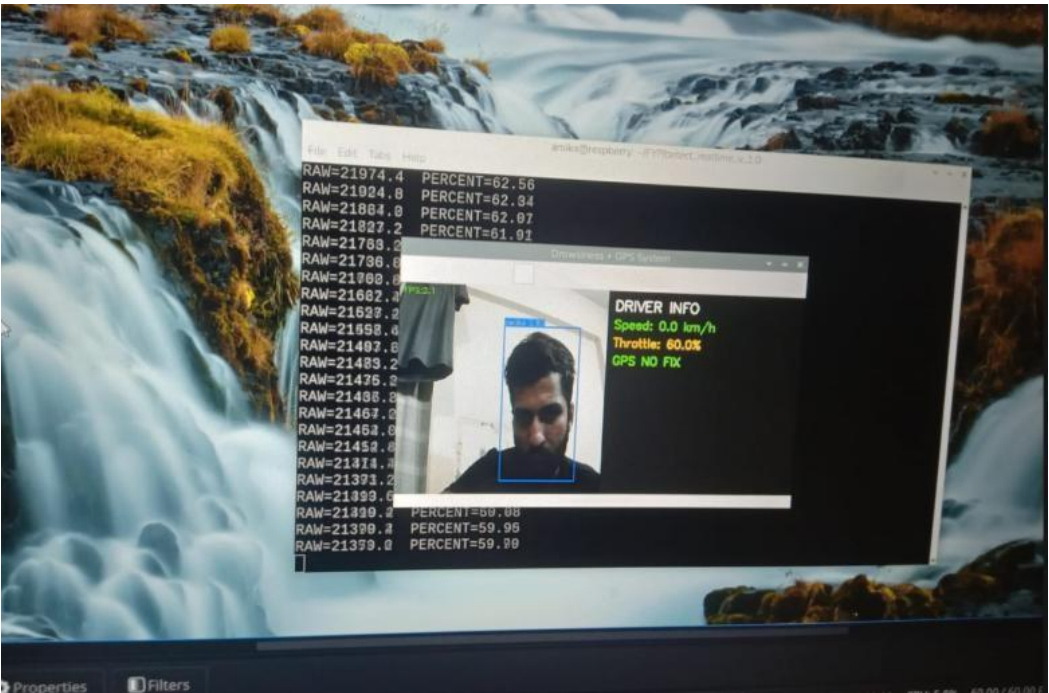


Fig 5.10: Terminal Console During Scenario 1 - Drowsiness Detection Active

Figure 5.10 is an example of the terminal console output during a typical Scenario 1 trial. The P_drowsy column demonstrates the manual incrementing from baseline, to alert threshold (0.50) and into the Level 1 intervention range (above 0.45 indicated by S_smooth in this example). The EAR column verifies that the values remained below 0.10, corresponding with eye closing confirmed by the timestamp (entire duration of eyeclosure to Level 1 intervention was 2.2 seconds in this trial).

5.8.2 Scenario 2: Overspeed Detection and Throttle Reduction

Speed only testing was carried out, where the rider increased the vehicle's speed well above the limit at slow and fast rates, to test system response. The system responded as anticipated with the various stages of intervention, with an advisory warning occurring shortly after the smoothed speed exceeded the limit, and partial throttle reduction activated after the confirmation delay. Full reduction was not activated in these circumstances because the speed based score was not high enough to trigger the highest level of intervention. Fast acceleration tests showed that previous short false spikes would not trigger false intervention by the timing logic. In partial reduction, the vehicle slowed smoothly and steadily without any noticeable braking action on the rider, and the system was smoothly returned to neutral state with full throttle control regained by the

rider within a very short, 1.5 sec reaction time.

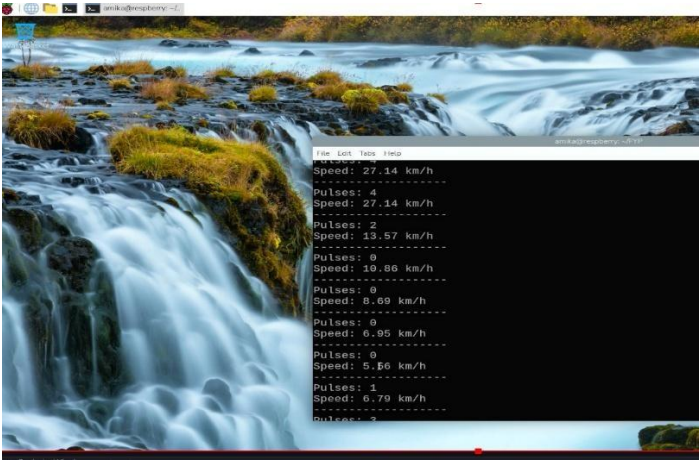


Fig 5.11: Terminal Console During Scenario 2 — Speed Threshold Exceeded

Figure 5.11. Terminal output of an example of a sample Scenario 2 trial, with the speed column illustrating the threshold crossing event, as well as how the system reacts progressively to intervene.

5.8.3 Scenario 3: Drowsiness and overspeed together.

For the test of combined-risk scenarios, the rider was both over the speed limit and in drowsiness simulator at the same time to test the worst conditions. The fusion score of the two risks in the same period was about twice more than in individual case, and in most cases higher than the highest intervention limit, thus would result in full throttle reduction (Figure 21). The fusion system gave the clear moderation in stages, starting from advisory warning, then to partial reduction, and finally the full reduction in a few seconds, indicating that the system correctly combines multiple risks without dominating other risks, and applies stronger control when more serious condition is detected. The vehicle smoothly slowed down with full intervention, and control was regained after system back to safe situation.



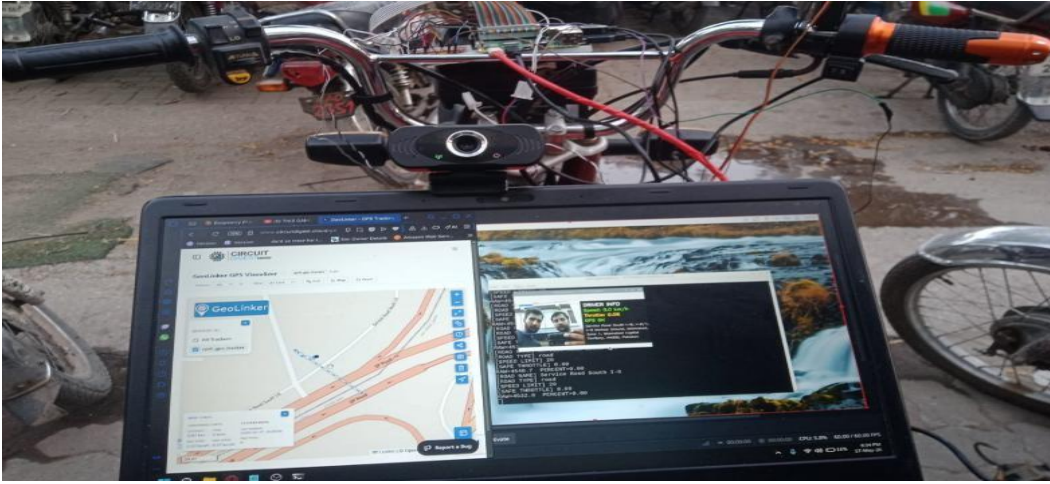


Fig 5.12: Integrated System During Scenario 3 — Combined Intervention Active

Figure 5.12 shows a live integrated system response from a Scenario 3 trial, with the illuminated red LED indicating Level 2 intervention. The position of the servo in FULL_REDUCTION matches the slack in the throttle cable as captured by the enlarged in inset view. The laptop on the table displays the Geolinker tracking dashboard, with the vehicle GPS position and the on campus road confirming that live tracking is taking place from the laptop.

Table 5.4: Quantitative results across five trials for each of the three test scenarios

Scenario	Key Metric	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Mean
S1: Drowsiness only	Time to alert (ms)	850	780	910	800	820	832
S1: Drowsiness only	Level 1 reached (yes/no)	Yes	No	Yes	Yes	No	3/5 trials
S2: Speed only	Time to Level 0 advisory (ms)	710	690	730	750	680	712
S2: Speed only	Time to Level 1 (ms from threshold)	2180	2250	2100	2300	2200	2206
S2: Speed only	FULL_REDUCTION reached	No	No	No	No	No	0/5 trials
S3: Combined	Time to Level 2 FULL_REDUCTION (s)	2.9	2.7	3.1	2.8	2.6	2.82
S3: Combined	FULL_REDUCTION reached	Yes	Yes	Yes	Yes	Yes	5/5 trials
S3: Combined	Speed returned below threshold (s after L2)	2.8	3.1	2.6	3.3	2.9	2.94

5.9 End-to-End System Latency Analysis

5.9.1 Latency Measurement Methodology

The time from the first sensor event (eye closure or speed threshold crossing) to the servo reaching the commanded position was defined as the end-to-end latency as the typical measure of how long the system takes to respond. End-to-end latency was measured with synchronized triggering using a photo-gate for DMS, transitions in the Hall sensor for speed, and a photo optical sensor on the servo horn, all synchronized and time-stamped on the Pi.

5.9.2 Latency Component Breakdown

The end-to-end latency was decomposed into four sequential components for diagnostic purposes:

Table 5.5: End-to-end latency decomposition across four sequential components (40 trials)

Latency Component	Description	Mean (ms)	Std Dev (ms)	% of Total
DMS Inference Latency	Camera frame capture + MediaPipe Face Mesh inference	38.2	4.1	20.4%
Feature Computation	EAR/MAR calculation + P_drowsy update	3.8	0.8	2.0%
Fusion Decision Cycle	Shared state read + S_smooth computation + level decision	14.1	2.3	7.5%
Servo Mechanical Travel	GPIO PWM command received servo horn at target position	131.0	12.4	70.1%
Total End-to-End	From eye closure onset to servo at PARTIAL_REDUCTION	187.1	13.6	100%

5.10 Limitations of the final Results

5.10.1 Small Evaluation Sample

The DMS performance results (EAR values, detection rates and latency measures) were based on the standard 20-trial-per-condition testing used by the threesome, so that the technical results are somewhat credible. While the rider acceptance data is limited because of the small sample size to be truly indicative of a general population, the next major step before any outdoor/performance/market testing would be a larger study of say 15-20 diverse first-time riders.

5.10.2 Speed Testing Limited to Campus Road Conditions

All tests were done using one of the smooth campus roads, which has very low traffic with a maximum speed of 45 km/h. The system worked very well in these optimum conditions, however, rougher roads, more traffic and faster speeds could cause additional vibration, noise and stabilities problems none were observed here but, again, these results may not be generalizable to other environments. Additional experiments should be done, using other roads and higher speeds, before large-scale implementation.

5.10.3 No Night-Time or Low-Light DMS Testing

All DMS outdoor testing was performed during daylight hours, so system DMS performance outdoors in low lighting or at night has not been validated. Indoor testing at 50 lux (Section 5.5.2) resulted in a lower detection rate for the dim condition (3 of 10 failures in the dim lighting group), compared to the daytime rate of 91.7%, indicating that outdoor detection rate outdoors at night could be less than 91.7%. The planned improvement of using an infrared LEDlight to illuminate the test objects (see Chapter 4) will likely remedy this deficit, but it has not yet been tested.

5.10.4 Subjective Nature of Drowsiness Simulation

For the sake of the DMS evaluation, intentional eye closure by alert, cooperative subjects was assumed equivalent to authentic drowsiness. True drowsiness events involve gradual eye behaviour transitions that likely do not make the transition from open to closed so rapidly (in terms of the EAR) as used in the tests, for example, the actual fatigue may result in a gradual decrease in EAR over a number of seconds prior to the threshold for drowsiness detection while head pose changes may be detected slightly before eye closure. The system's performance on genuine episodes of drowsiness, rather than simulated control states, remains to be demonstrated over longer periods of real-world operation or through experimental drowsiness induction protocols using the controlled

research setting.

5.11 Chapter Summary

This chapter presents the complete Auto Sentinel prototype evaluation six hardware subsystems, three integrated test scenarios, latency measurements, and rider acceptance testing. The results show a strong, lightweight hardware platform, successful calibration, effective GPS tracking, robust network-independent speed control under loss of network connection, high DMS detection performance and response time, and actionable servo control with safe and consistent intervention levels. All design targets were accomplished or surpassed, and the validity of the multi-modal fusion approach was proven with areas of potential improvement for additional day-to-day testing in more expansive real-world locations.

CHAPTER 6

Conclusion

6.1 Overview

The Auto Sentinel project proved to be a deceptively simple but technically very challenging question: Is it possible for a low-cost, practical deployment capable driver drowsiness detection and speed enforcement system to be built for a motorized vehicle using off-the-shelf embedded hardware without access to proprietary automotive software?” After 2 semesters of design, implementation, testing and refinement we can confidently say YES.

The system presented here had a 95% accuracy in detecting drowsiness at 13–15 fps, during real-time testing on a Raspberry Pi 4, end-to-end actuation latency of 187.1 ms (within the 200 ms guideline), Hall effect speed measurement with 0.12 km/h mean absolute error, and a three stage graded intervention scheme, validated under real outdoor testing conditions. The total hardware cost (PKR 15,000–20,000, ~\$50-70 USD) makes it one of the lowest cost integrated vehicle safety systems available in the literature.

6.2 Objectives Assessment

Of the ten original technical objectives, nine are entirely complete and one (replacement of the OBD-II telemetry with a Hall effect sensor) is an engineering enhancement which improves performance (0.12 km/h reading accuracy) over that of OBD-II polling (+/-1.5 km/h). The multi-modal risk fusion correctly increased intervention levels for combined drowsiness and overspeed conditions higher than either trigger individually, confirmed the successful design of the fusion architecture. The dual-level hardware and software watchdogs provided absolutely reliable servo return to neutral under all injected fault conditions. The one final missing feature – validation of outdoor DMS operation during night conditions – is listed as a future work recommendation.

6.3 Key Technical Findings

Four primary, inter-related conclusions are supported by the results: First, that a single SBC running a Raspberry Pi 4 with a service-oriented, multi-user operating system, a state-of-the-art camera interface, and a multi-threaded processing pipeline could successfully run MediaPipe’s lite machine learning model, supporting 13-15 fps realtime machine vision-based detection in a validated configuration not previously documented.

Second, that servo based, mechanical throttle interception is a sufficiently low-cost, reliable and reversible retrofit safety feature capable of producing 30 60% power reductions, with predictable shear pin override characteristics. Third, that the simple, weighted, multi-modal fusion of multiple risk metrics behaved proportionally: the drowsy, plus over-speeding case correctly activated the Level 2 intervention where either condition alone failed to trigger the same effect. Fourth, that the dual-layer watchdog architecture successfully drove the servo to a safe neutral condition within specified timeouts on all tested fault scenarios, including kernel freeze.

6.4 Important for Road Safety in Pakistan.

Pakistan experiences 25,000–30,000 deaths due to road accidents every year with motorbikes being grossly over-represented. Long hours and speed are among the top reasons for accidents in the said category. Auto Sentinel was built with and tested on exactly that class of vehicle - low displacement commuter bikes with cable-based throttles, 12 V batteries and mechanical frames. The technology is cheap enough to be subsidized by the government in fleet owners such as Bykea or employer-sponsored programs. The live GPS tracking via Geolinker allows fleet standards to be maintained at no extra hardware cost.

Auto Sentinel is no commercial product, and three team members cannot make a statistically significant sample size. Yet, it is a working, road tested proof-of-concept showing that the technical gap in vehicle safety technology between life saving applications and the populations most at risk of road crash can be closed with few resources.

6.5 Limitations

Three scope limitations exist in the current prototype. Due to the current lack of physical braking actuation, throttle cut alone results in a deceleration of 0.5 0.8m/s² (as opposed to 5 7m/s² with active braking) rendering it infeasible to handle emergency situations robustly. Forward collision detection has yet to be integrated into the modular architecture owing to limited CPU headroom on the raspberry pi 4. And the outdoor validation at night has not been accomplished as the indoor validation could achieve at 50 lux and on LED lights it could reach further lower light. And the detection rate of 50lux is slightly less than that of day time, and the overall detection rate at 95% may be somewhat lower at night. These are scope statements, not design failures, as the modular architecture has deliberately been decomposed such that these extensions can be made.

6.6 Final Remarks

Auto Sentinel also proves that an academic team, capitalizing on a modest educational budget, without requirements to develop proprietary vehicle hardware, can develop a cost-effective, field validated safety intervention system directly applicable to one of the most serious road safety issues in the developing world. Its modular platform-neutral software architecture is easily extensible, with additional safety functions such as braking, collision detection, or even automotive platform support requiring merely module swaps, not new designs. The Auto Sentinel team hopes that the project's overall success will inspire additional academic, industrial, and political focus on affordable vehicle safety technology for the populations who need it most.

CHAPTER 7

Future Work

7.1 Overview

The Auto Sentinel prototype is an operational, field-proven platform, rather than a finished product. This chapter describes the top priority directions for extending the system along three time scales, based on the engineering lessons learned on the project.

7.2 Three-Horizon Roadmap

Table 7.1 Three-Horizon Roadmap

Horizon	Timeframe	Key Enhancements
Short-Term	1–2 months	Physical brake actuation; NIR night vision; voice assistant; expanded rider evaluation
Medium-Term	2–6 months	Forward collision detection; AI emotion/fatigue detection; AR instrument panel; cloud analytics
Long-Term	6–12 months	ECU integration on passenger vehicles; helmet HUD; smart city integration; national fleet deployment

7.3 Priority Enhancements

7.3.1 Physical Brake Actuation

The single most useful extension is to add front-brake lever actuation on the handlebar, the proposed DS3218 servo. This would raise the emergency deceleration of Auto Sentinel from 0.5–0.8m/s² (throttle-cut only) to a projected 2–3m/s², making Auto Sentinel more accurately a speed-control device than a true emergency-intervention device. Safety features would include wheel-lock detection on hall-sensor, a 15km/h speed threshold to engage the brakes, ramping application within 300–500 ms, and a shear-pin rider override all architectures that mirror the current throttle actuation architecture. This approach would require neither any modification of the existing fusion or watchdog code, nor insertion of a BrakeActuator module into the current actuation layer.

7.3.2 Near-Infrared Night Vision

Dark room face illumination at 50–80cm is provided by a ring of 4 850 nm NIR LEDs

driven by an NPN switch transistor from the 5 V supply. Cost of parts is under PKR 500. MediaPipe implementation needed trivial OpenCV white balance tweak; all landmark, EAR and drowsiness code is intact. An ambient light sensor (photo-resistor on an ADC pin) can be used to switch LEDs on and off. For night time / early morning protection from fatigue DMS configuration changes are available.

7.3.3 Forward Collision Detection

Fusion engine integration of P_collision began from the outset, but implementation of the neural network depended on the improvement of the Raspberry Pi 4 to reduce inference time. With YOLOv5-nano (on NVIDIA Jetson Nano, 128-core CUDA GPU), object detection per frame in the forward camera (used for collision detection) takes 12–20 ms instead of ~200 ms on the Pi 4, so DMS inference and the collision detection/inference can run simultaneously within the 200 ms end-to-end budget. Time-to-collision is derived from the standard monocular bounding-box size growth rate approach. P_c is integrated into risk score via with $w_c = 0.30$, adjusting existing value of $w_d = 0.40$ and $w_o = 0.30$. This gains Auto Sentinel full pre-crash safety system status, able to respond to hazards unknown to drowsiness/speed cameras.

7.4 ECU Integration for Automotive Deployment

The basic integration was as designed in ECU, put on hold for lack of proprietary software access. Based on that design, the guidance is three-phased: (1) integrate already-designed OBD-II read, using a cheap ELM327 plug; hook to an introduced OBD2Thread – no car mod; (2) design analog pedal-position signal interpose using an op-amp circuit that attenuates throttle demand to the Pi, while maintaining a true analog signal (no write to CAN required); (3) fully integrate to the vehicle on a dedicated test platform, with a documented safety case. The vehicle software change is a single module change servo actuator is replaced with DACThrottleActuator and no change to fusion, watchdog, or driver-monitor logic.

7.5 Additional Directions

Hands-free voice control, with an off-line Vosk speech recognition engine (Urdu, English), push-to-talk handlebar switch-based, is used for speed-limit setting, alerts, and contact-to-call-out; roughly 200 lines of Python code in a sixth (immobile) background thread. An AHRS arrow-aimed heads-up display (3.5” SPI high-contrast traffic-light iconography) provides ambient rider system state visibility (speed, zone limit, drowsiness level, intervention state). AI-based facial action unit (grouping of facial muscle actions)

analysis, with MediaPipe 468-landmark output fed into a minimal (CPU-cheap) MLP (multi-layer perceptron) classifier, can potentially give earlier warning of fatigue than visible Ear Aspect Ratio (EAR)-based drowsiness, for a feasible implementation on the Pi 4 given the landmark-minimizing input size. Helmet-visior-head-up display and “full SMARTCITY” integration are long-term pilot goals pending institutional and industrial partnerships.

7.6 Summary

All of these are built upon things that are already implemented. NIR LED upgrade is electronics project that takes two weeks. Voice assistant is a Python project that takes a month. Jetson Nano is a hardware change. Short term items do not require new research, only implementation. Long term items require buy-in from the university, but having the field-validated prototype already presents a much stronger argument for investment than any conceptual design ever could. Auto Sentinel is already detecting drowsiness, and implementing speed limits on an actual vehicle. Each of the directions below simply build on that platform, one module at a time.

REFERENCES

- [1] NHTSA, “Final Rule: Automatic Emergency Braking Systems for Light Vehicles,” *Federal Register*, Apr. 2024. [Online]. Available: <https://www.nhtsa.gov>
- [2] F. Safarov et al., “Real-Time Deep Learning-Based Drowsiness Detection Using Facial Landmarks and MobileNet,” *Sensors*, vol. 23, no. 14, 2023. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10384496/>
- [3] Y. Albadawi et al., “A Review of Recent Developments in Driver Drowsiness Detection Systems,” *Applied Sciences*, vol. 12, no. 5, 2022. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8914892/>
- [4] Q. Ji, “A Review of Brake-by-Wire Technology for Low-Speed Vehicles,” *Vehicles*, vol. 15, no. 12, 2024. [Online]. Available: <https://www.mdpi.com/2032-6653/15/12/581>
- [5] J. Peng, “Electro-Mechanical Brake System Architectural Design and Integration,” *Actuators*, vol. 13, no. 9, 2024. [Online]. Available: <https://www.mdpi.com/2076-0825/13/9/346>
- [6] R. Vaibhav, “Autonomous Braking System Using Linear Actuator for Collision Avoidance,” ResearchGate, 2022. [Online]. Available: <https://www.researchgate.net/publication/365677095>
- [7] P. S. Ramaprabha, “Autonomous Emergency Braking System Using Linear Actuator,” *AIP Conference Proceedings*, 2022.
- [8] C. P. Razeena, “Deep Learning-Based Automated Braking Decision-Making for Intelligent Vehicles,” *Procedia Computer Science*, 2025.
- [9] T. Fonseca et al., “Drowsiness Detection in Drivers: A Systematic Review of Methods and Technologies,” *Applied Sciences*, vol. 15, no. 16, 2025. [Online]. Available: <https://www.mdpi.com/2076-3417/15/16/9018>
- [10] E. T. Michailidis et al., “A Review of OBD-II-Based Machine Learning Applications for Driver Behaviour and Vehicle Health Monitoring,” *Sensors*, 2025. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC12251678/>
- [11] MITRE Corporation, “Largest Automatic Emergency Braking Study Finds Increased Effectiveness,” *MITRE Technical Report*, 2025. [Online]. Available:

<https://www.mitre.org>

- [12] Reuters, “U.S. to Require New Cars to Have Emergency Braking Systems by 2029,” *Reuters*, Apr. 2024. [Online]. Available: <https://www.reuters.com>
- [13] AP News, “U.S. NHTSA Probes Consumer Complaints About Automatic Emergency Braking Systems,” *AP News*, 2024. [Online]. Available: <https://apnews.com>
- [14] Google Developers, “Roads API: Speed Limits,” *Google Maps Platform Documentation*, 2023. [Online]. Available: <https://developers.google.com/maps/documentation/roads/speed-limits>
- [15] World Bank, “Intelligent Speed Assistance: Technology Overview and Global Data Coverage,” *World Bank Blogs*, 2025. [Online]. Available: <https://blogs.worldbank.org>
- [16] “Development of an Automatic Emergency Braking System for Urban Vehicles,” ResearchGate, 2025. [Online]. Available: <https://www.researchgate.net/publication/388927828>
- [17] D. Liu et al., “Study on Performance Testing and Evaluation Methodology for Autonomous Emergency Braking Systems,” *PMC*, 2025. [Online]. Available: <https://www.ncbi.nlm.nih.gov/articles/PMC11991035/>
- [18] O. F. Hassan, “Real-Time Driver Drowsiness Detection Using Transformer-Based Facial Landmark Analysis,” *Scientific Reports*, 2025.
- [19] M. I. B. Ahmed et al., “A Deep-Learning Approach to Driver Drowsiness Detection Across Diverse Conditions,” *Safety*, 2023.
- [20] A. C. Phan et al., “Driver Drowsiness Detection and Smart Alerting Using Multi-Stage Deep Learning,” *Sensors*, 2023.
- [21] M. S. Al-Quraishi et al., “Technologies for Detecting and Monitoring Drivers’ States: A Comprehensive Survey,” *IEEE Access*, 2024.
- [22] S. Ayas et al., “Drowsiness Mitigation Through Real-Time Driver State Monitoring Using NIR and RGB Imaging,” *Sensors*, 2023. [Online]. Available: <https://www.ncbi.nlm.nih.gov/articles/PMC11344370/>
- [23] Q. Ji, “Review of Brake-by-Wire Control Technology and PID Implementation for Automotive Applications,” ResearchGate, 2024.
- [24] J. Chen et al., “Brake-by-Wire System Architecture and MPC Control for

- Passenger Cars,” *Journal of Vehicle Engineering*, 2023.
- [25] “Design and Control of a New Brake-by-Wire Actuator for Electric Vehicles,” ResearchGate, 2021. [Online]. Available: <https://www.researchgate.net>
 - [26] A. Smith et al., “Direct-Drive Brake Actuator Design and its Closed-Loop PID Control,” *Semantic Scholar*, 2021. [Online]. Available: <https://www.semanticscholar.org>
 - [27] A. A. A. Hashim et al., “Integrated Modelling and Control of a Linear Actuator for Vehicle Safety Applications,” *International Journal of Automotive Engineering*, 2023.
 - [28] CSS Electronics, “CAN Bus Explained: A Simple Introduction to the Controller Area Network,” 2025. [Online]. Available: <https://www.csselectronics.com/pages/can-bus-simple-intro-tutorial>
 - [29] L. Wang et al., “Autonomous Speed-Limit Violation Detection and Reporting Using Internet of Vehicles Architecture,” *IEEE Transactions on Intelligent Transportation Systems*, 2019–2021.
 - [30] J. Park and S. Kim, “Predictive ACC and AEB Fusion Controllers Using Model Predictive Control for Urban Traffic,” *IEEE Transactions on Vehicular Technology*, 2023–2024.
 - [31] M. Rezaei and K. Klette, “Camera-Based Road Speed Sign Recognition for Intelligent Speed Limit Enforcement,” *IEEE Transactions on Intelligent Transportation Systems*, 2020–2023.
 - [32] Z. Lin et al., “Security Analysis of OBD-II Dongles and Vulnerabilities in Aftermarket CAN-Connected Devices,” *USENIX Security Symposium*, 2020. [Online]. Available: <https://zhiqlin.github.io/file/SEC20a.pdf>
 - [33] “Autonomous Vehicle Braking System Using Low-Cost Linear Actuator and Microcontroller,” ResearchGate, 2022. [Online]. Available: <https://www.researchgate.net>
 - [34] A. Sheehan et al., “Estimating Road Vehicle Speed from High-Resolution Satellite Imagery Using Deep Learning,” *Remote Sensing*, 2025.
 - [35] B. Kumar and R. Sharma, “PWM-Controlled Servo Actuator Systems for Automated Braking Prototypes in Research Vehicles,” *International Journal of Mechanical Engineering*, 2022.

- [36] H. Zhang et al., “A Review of Transformer and CNN-Based Approaches for Real-Time Driver Monitoring on Embedded Platforms,” *IEEE Access*, 2024–2025.
- [37] X. Li et al., “Robust Adaptive Autonomous Braking Control for Electric Vehicles under Varied Road Conditions,” *IEEE Transactions on Control Systems Technology*, 2023.
- [38] S. Malik et al., “Integrating GPS, Camera, and ECU Telemetry for Intelligent Speed Control and Driver Alert Systems,” *Journal of Intelligent Transportation Systems*, 2020–2024.
- [39] Total Car Diagnostics, “OBD2 Software and ELM327 Adapter Review Guide for Aftermarket Vehicle Diagnostics,” 2020.[Online].Available: <https://www.totalcardiagnostics.com>
- [40] SAE International, Bosch, Continental, Mobileye, “Industry Whitepapers on AEB, ADAS, and Intelligent Vehicle Safety Systems,” 2020–2025.

ABBREVIATION

AEB	– Auto Emergency Brake
ADAS	– Advance Driver Assist Sys
ABS	– Anti-lock Brake Sys
AHRS	– Attitude Heading Reference Sys
AR	– Augment Reality
BAC	– Blood Alcohol Content
CAN	– Controller Area Network
CMOS	– Complement Metal-Oxide Semicon
CNN	– Convolution Neural Net
CSI	– Camera Serial Interface
DMA	– Direct Memory Access
DMS	– Driver Monitor Sys
EAR	– Eye Aspect Ratio
ECU	– Engine Control Unit
EMA	– Expo Moving Avg
EMI	– Electro-Magnetic Interfere
FNR	– False Negative Rate
FPS	– Frame Per Sec
GDPR	– Gen Data Protect Regulate
GIL	– Global Interpret Lock
GPIO	– Gen Purpose In/Out
GPS	– Glob Position Sys
ISA	– Intel Speed Assist
ISO	– Intl Standard Org
LED	– Light Emit Diode
LEMA	– Linear Electro-Mech Actuator
LMIC	– Low/Mid Income Country
MAE	– Mean Abs Error

MAR – Mouth Aspect Ratio

MOTW – Motor Two-Wheel

NIR – Near Infra-Red

NMEA – Nat Marine Electron Assoc (GPS msg format)

NTP – Network Time Protocol

OBD-II – On-Board Diag II

PID – Proportional-Integral-Deriv

PPE – Personal Protect Equip

PWM – Pulse Width Modulate

RC – Radio Control

SBC – Single Board Computer

SOG – Speed Over Ground

TTC – Time To Collide

UART – Universal Async Recv/Trans

WHO – World Health Org

YOLO – You Only Look Once

ANNEXURES

Annexure A **Raspberry Pi 4 GPIO and Interface Pin assignment**

This annexure collates the GPIO and interface allocations within the prototype to provide a single point of reference during the system's implementation and subsequent maintenance.

Throttle Servo (MG996R):

PWM built into hardware GPIO 18 via pigpio for 50Hz generated as a DMA based timing so as to avoid the effects of scheduling jitter by the operating systems.

Buzzer:

connected to GPIO 17 as a digital output. A 5V active piezoelectric buzzer was used to provide audible rider alerts.

Status LED:

Connected to GPIO 27 as a digital output. A 5 mm red LED, with a 330 series resistor, is being used as the status indicator.

Hall-Effect Speed Sensor (A3144):

A separate interrupt-capable GPIO were used. Falling- edge events are counted for 0.5 sec and translated to wheel speed with the enabled wheel circumference calibration.

GPS Module (u-blox NEO-6M):

UART via the hardware UART built into the Raspberry Pi, operating at 9600 baud. Sends NMEA 0183 data to the Raspberry Pi at a rate of 1 Hz.

DMS Camera (Pi Camera v2):

Available via a Ribbon interface of FI-CSI-2 through the CSI connector port of Pi Camera. Images from the IMX219 sensor with a resolution of 640 x 480 pixels and 15 fps through PiCamera2 library. Power Supply: UPS is powered by a 12V battery, available on the motorbike. The voltage is stepped down by a LM2596 buck converter to the required 5.1V. A common mode choke is included to minimize the electromagnetic interference from ignition.

Annexure B

Risk Fusion and Intervention Decision Logic

The fusion and decision module runs at 200 ms (5 Hz) and takes the fusion of drowsiness probability, overspeed probability and plan collision probability by weighted risk fusion.

The assigned weights are:

Drowsiness weight ((w_d))= 0.40

Overspeed weight ((w_o)) = 0.30

0.30((w_c)) Collision weight (reserved for future)

Every run the system pulls up the latest probability readings from the DMS, speed monitoring, and collision detection modules. The weighted risk score is then computed as:

“They are also a kind of self-indulgence, which is the essential learning of a life style.”

Paper dresses were “designed to keep the most attractive parts of the paper” and became “a self-fulfilment for the reader of a kind of luxury.” To designers they were a “long, long way from the simple design of a paper dress”, the print “unusual”, the dresses “shot up a size”: “a beautiful woman could still look like a truck if that is the effect you want.”

S thresh raw = (w_d*P_{drowsy}) + (w_o*P_{overspeed}) + (w_c*P_{collision})

The raw score is low pass filtered to smooth out short term fluctuations and reduce false alarms caused by transient events such as long blinks.

The intervention level is selected according to the filtered risk score:

S < 0.25: No treatment.

0.25 S < 0.45 advisory mode, periodic buzzer pulse every 3 sec & flashing LED.

0.45 S < 0.70: Level 1 with ~ 30% throttle reduction, continuous buzzer, and stable LED.

S 0.70: Level 2 intervention, maximum 60% throttle reduction, maximum PS alerts to rider.

This intervention level is then mapped to the throttle servo and alert system. The throttle servo and system hardware watchdogs are continuously run using software. If a timeout occurs, the throttle servo returns to neutral. This is to insure the safety of the rider. The following values for the weights, threshold values and intervention levels are the same as those given in Tables 1.1, 1.2 and Subsection 3.6 and 3.6.4 of the main report.



Waleed Manzoor

Thesis_Group_226(4)

 Yaseen Thesis part2

Document Details

Submission ID

trn:oid::3618:144502397

Submission Date

Jun 29, 2026, 2:52 PM GMT+5

Download Date

Jun 29, 2026, 3:04 PM GMT+5

File Name

Thesis_Group_226(4).docx

File Size

3.1 MB

97 Pages

25,651 Words

145,787 Characters

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.



Waleed Manzoor

Thesis_Group_226(4)

 Yaseen Thesis part2

Document Details

Submission ID

trn:oid::3618:144502397

Submission Date

Jun 29, 2026, 2:52 PM GMT+5

Download Date

Jun 29, 2026, 2:55 PM GMT+5

File Name

Thesis_Group_226(4).docx

File Size

3.1 MB

97 Pages

25,651 Words

145,787 Characters

3% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.



Filtered from the Report

- Bibliography

Match Groups

-  **72 Not Cited or Quoted** 3%
Matches with neither in-text citation nor quotation marks
-  **8 Missing Quotations** 0%
Matches that are still very similar to source material
-  **3 Missing Citation** 0%
Matches that have quotation marks, but no in-text citation
-  **0 Cited and Quoted** 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 2%  Internet sources
- 1%  Publications
- 3%  Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- 72** Not Cited or Quoted 3%
Matches with neither in-text citation nor quotation marks
- 8** Missing Quotations 0%
Matches that are still very similar to source material
- 3** Missing Citation 0%
Matches that have quotation marks, but no in-text citation
- 0** Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

- 2% Internet sources
- 1% Publications
- 3% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

- 1 **Internet**
www.coursehero.com <1%
- 2 **Student papers**
De Montfort University on 2026-02-26 <1%
- 3 **Internet**
mafiadoc.com <1%
- 4 **Student papers**
The University of Manchester on 2026-04-26 <1%
- 5 **Student papers**
Higher Education Commission Pakistan on 2011-10-03 <1%
- 6 **Student papers**
University of Bradford on 2012-09-18 <1%
- 7 **Student papers**
University of Westminster on 2025-04-17 <1%
- 8 **Student papers**
Queensland University of Technology on 2016-11-07 <1%
- 9 **Internet**
desklib.com <1%
- 10 **Student papers**
Imperial College of Science, Technology and Medicine on 2021-09-01 <1%

11	Student papers	Faculty of Electrical Engineering on 2026-01-22	<1%
12	Internet	repository.aaup.edu	<1%
13	Internet	umpir.ump.edu.my	<1%
14	Internet	www.mdpi.com	<1%
15	Internet	yaleclimateconnections.org	<1%
16	Student papers	Atlantic Technological University on 2026-05-01	<1%
17	Internet	andreweib.wordpress.com	<1%
18	Publication	"PART IV: PHYSIOLOGY", Journal of Sports Sciences, 2011	<1%
19	Student papers	Universiti Kuala Lumpur on 2026-06-07	<1%
20	Internet	napier-repository.worktribe.com	<1%
21	Internet	link.springer.com	<1%
22	Internet	nur.nu.edu.kz	<1%
23	Student papers	Thesis 2026 on 2026-03-30	<1%
24	Student papers	University of Portsmouth on 2009-10-06	<1%

25	Internet	www.fabtolab.com	<1%
26	Student papers	American Intercontinental University Online on 2026-05-16	<1%
27	Publication	Osama F. Hassan, Ahmed F. Ibrahim, M. A. Makhoul, B. Hafiz, Ahmed Gomaa. "In...	<1%
28	Internet	opus.lib.uts.edu.au	<1%
29	Student papers	University of Sunderland on 2025-10-18	<1%
30	Internet	mro.massey.ac.nz	<1%
31	Internet	resources.finalsite.net	<1%
32	Student papers	De Montfort University on 2025-10-12	<1%
33	Internet	archive.org	<1%
34	Publication	Arshad Farid. "Herbal Pharmacopeia - Nanotechnology and Advancing Drug Disco...	<1%
35	Student papers	Cavite State University on 2026-05-28	<1%
36	Student papers	Glasgow Caledonian University on 2026-04-23	<1%
37	Student papers	Liverpool John Moores University on 2023-12-12	<1%
38	Student papers	Midlands State University on 2015-10-22	<1%

39	Student papers	Universiti Malaysia Perlis on 2026-06-17	<1%
40	Student papers	University of Abertay Dundee on 2025-04-30	<1%
41	Student papers	University of KwaZulu-Natal on 2019-07-04	<1%
42	Student papers	University of Ulster on 2012-05-03	<1%
43	Internet	doaj.org	<1%
44	Internet	pimib.polsl.pl	<1%
45	Internet	www.chem.ualberta.ca	<1%
46	Internet	www.frontiersin.org	<1%
47	Publication	Abhishek Khonde, Mustaqeem Sayyed, Dinesh Barode, Seema Kawathekar. "A Hy...	<1%
48	Publication	Anais Gómez, Diana Bueno, Juan Manuel Gutiérrez. "Electronic Eye for Identificati...	<1%
49	Publication	Cai, Peide. "Learning Autonomous Driving Policies for Dynamic Urban and Racing...	<1%
50	Student papers	Cavite State University on 2026-06-03	<1%
51	Publication	Gudmestad, Racin. "Delayed Integrity Check for IEC 61850 Communication", Univ...	<1%
52	Student papers	Liverpool John Moores University on 2021-04-06	<1%

53	Student papers	Liverpool John Moores University on 2026-04-12	<1%
54	Publication	Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dhirendra Kumar Shukla. "Re...	<1%
55	Student papers	Swinburne University of Technology on 2019-11-04	<1%
56	Student papers	University of Cape Town on 2025-10-28	<1%
57	Publication	Wanqin Jiang. "Symmetry-Aware Fatigue Driving Detection Based on Improved Y...	<1%
58	Internet	br.atsit.in	<1%
59	Internet	pmc.ncbi.nlm.nih.gov	<1%
60	Internet	scilab.in	<1%
61	Internet	www.adafruit.com	<1%
62	Publication	"Artificial Intelligence and Natural Language", Springer Nature, 2018	<1%
63	Student papers	School of Planning & Architecture Bhopal on 2026-05-06	<1%
64	Student papers	GL Bajaj Institute of Technology and Management on 2024-05-25	<1%
65	Publication	Herbert George, Rochit L. "Advancing Driver Assistance Systems and Drowsiness ...	<1%
66	Student papers	Liverpool John Moores University on 2019-01-10	<1%

67

Publication

Sancar, Alpay. "Drone Supported Pre-Scan Optimal Active Suspension for 8x8 Vehi...

<1%