DAATS – Driver Attention Autonomous Transition System

By

Dharsan Guruparan, Jay Alikkal, Sanskar Lamsal, Unmesh Khanolkar

A Senior Design Project Report Submitted to the Faculty of Electrical Engineering and Computer Science
University of Toledo

Faculty Supervisor(s): Dr. Gerald Heuring

December 2023

TABLE OF CONTENTS

Abstract

1. Problem Statement
- 1.1. Need
- 1.2. Objective
- 1.3. Background

2. Requirements Specifications
- 2.1. The Requirements
  - 2.1.1. Engineering Requirements
  - 2.1.2. Marketing Requirements
- 2.2. Constraints
  - 2.2.1. Economic
  - 2.2.2. Environmental
  - 2.2.3. Manufacturability
  - 2.2.4. Health and Safety
  - 2.2.5. Social

3. Design
- 3.1. Design Alternative
- 3.2. Level 0
- 3.3. Level 1
- 3.4. Level 2

4. Design Verification
- 4.1. Test Results
- 4.2. Requirements Verification
- 4.3. Standards
- 4.4. Ethics, Accessibility, and IP
  - 4.4.1. Ethical Considerations
  - 4.4.2. Accessibility
  - 4.4.3. Intellectual Property (IP)

5. Summary and Conclusions
- 5.1. Improvements
- 5.2. Summary
- 5.3. Conclusion

6. References

Appendix A: Project Management Plan
- A.1 Work Breakdown Structure
- A.2 Member Contribution
- A.3 Development Cost

Appendix B: Software

Table of Figures

Figure 1: Level 0 Diagram
Figure 2: Level 1 Diagram
Figure 3: Level 2 Diagram
Figure 4: Drowsiness Detection Testing
Figure 5: Simulation Testing A
Figure 6: Simulation Testing B
Figure 7: Simulation Testing C
Figure 8: Computer vs Rpi 3 Performance Comparison
Figure 9: Roundtrip Time vs Network Speed
Figure 10: Autonomous Mode Switch Time

Table of Codeblocks

Codeblock 1: Client Socket
Codeblock 2: Voice Alert
Codeblock 3: Multi-threading

Table of Tables

Table 1: Engineering Requirements
Table 2: Rpi vs Computer Performance Timing
Table 3: Roundtrip Time for Network Speeds
Table 4: Autonomous Mode Switch Time
Table 5: Requirements with Verification
Table 6: Project Breakdown
Table 7: Bill of Materials

Abstract

Distracted driving poses a grave threat to road safety, necessitating innovative solutions to combat this growing concern. Drowsiness and Autonomous Transition System (DAATS) emerges as a promising solution, integrating cutting-edge technologies to address distracted driving and enhance overall road safety.

DAATS leverages a powerful ensemble of technologies, including the Robotic Operating System (ROS), MUX implementation, dlib and OpenCV libraries, face recognition modules, and Python scripting. This robust framework facilitates real-time drowsiness detection, enabling a seamless transition to autonomous driving upon identifying signs of driver fatigue.

The heart of DAATS lies in its camera-based drowsiness detection system. Utilizing the dlib and OpenCV libraries, the system analyzes facial features and identifies subtle indicators of drowsiness, such as eyelid closure and head movements. This real-time detection triggers a signal transmission via TCP/IP network to the ROS architecture embedded within the vehicle. Within the ROS environment, MUX communication facilitates a smooth and safe transition from manual to autonomous control. This ensures uninterrupted operation and minimizes potential risks associated with abrupt control changes. Furthermore, DAATS incorporates LIDAR technology, providing real-time obstacle detection and distance measurements to enhance autonomous driving capabilities and enable proactive collision avoidance.

Extensive testing procedures, encompassing both unit and integration tests, have rigorously validated the functionality and reliability of DAATS. Each individual component has undergone meticulous evaluation, ensuring optimal performance and seamless collaboration within the system. This rigorous testing process guarantees that DAATS meets all specified engineering requirements and operates at an optimal level.

DAATS presents a significant advancement in addressing the issue of distracted driving. By integrating advanced technologies and emphasizing safety, DAATS paves the way for a future with safer roads and a more responsible driving experience. The potential for further development and integration with emerging technologies holds immense promise for a future where autonomous driving contributes to a safer and more sustainable transportation ecosystem.

1. Problem statement

1.1. Need Statement

In the realm of transportation, a pressing issue has emerged, drawing attention to the need for immediate intervention. With the recent update in the Ohio Revised Code, which deems it illegal to use or hold a cell phone or electronic device while driving on Ohio roads, there's a critical need to address the challenges posed by distracted driving. Disturbingly, approximately 3,000 lives are lost annually in the United States due to distracted driving incidents, underscoring the urgency for innovative solutions.

1.2. Objective

The primary objective of this project is to conceptualize, design, and implement the Drowsiness and Autonomous Transition System (DAATS) with a dual focus: to develop a functional system capable of detecting driver drowsiness and enabling a smooth transition from manual to autonomous driving. The project aims to validate the system through comprehensive testing, utilizing both a simulator and a physical car. Leveraging advanced technologies, including dlib, OpenCV, ROS robotic operating system, MUX implementation, and others, the system will be rigorously tested in simulated environments and real-world scenarios to ensure its effectiveness, reliability, and compliance with engineering requirements. By testing the system on both platforms, the project seeks to assess its performance in controlled settings as well as in real-world driving conditions, contributing to the development of a robust and adaptable solution to address the critical issue of distracted driving.

1.3. Background

Distracted driving is a pervasive and critical issue in road safety, significantly impacting drivers and road users. The act of diverting attention away from the task of driving, particularly using cell phones and electronic devices, has emerged as a leading cause of accidents and fatalities on roadways. The consequences of distracted driving are particularly alarming, with approximately 3,000 lives lost annually in the United States due to related incidents.

The gravity of distracted driving is underscored by recent legislative actions. The new distracted driving law, outlined in Section 4511.204 of the Ohio Revised Code, directly addresses this growing concern. In Ohio, it is explicitly illegal to use or hold a cell phone or electronic device in hand, lap, or any other part of the body while operating a vehicle on Ohio roads. This legal prohibition signifies a crucial step toward promoting safer driving habits and reducing the risks associated with distractions behind the wheel. Law enforcement officers have the authority to pull over drivers observed violating this law, emphasizing the commitment to enforcing measures that enhance road safety and mitigate the impact of distracted driving.

2. Requirements Specifications

2.1 Requirements

Drowsiness Detection: The system's image processing capabilities, utilizing dlib and OpenCV, should be sophisticated enough to discern subtle facial changes indicative of drowsiness, particularly around the eyes. The system must function effectively in varied light and driving conditions, distinguishing between normal and drowsy states with high accuracy to avoid false positives and negatives.

Autonomous Transition: Once drowsiness is detected, DAATS should seamlessly engage the vehicle's autonomous driving mode. This involves a safe and efficient handover of control from the driver to the vehicle's autonomous system, utilizing the ROS robotic operating system. The transition process should prioritize safety, ensuring that the vehicle responds appropriately to the driving environment during the switch.

System Testing: Testing in simulated environments and real-world scenarios is crucial to validate the system's effectiveness. The simulated tests can control variables for focused assessments, while real-world tests provide insights into the system's functionality under everyday driving conditions. This comprehensive testing strategy ensures the system's readiness for practical deployment.

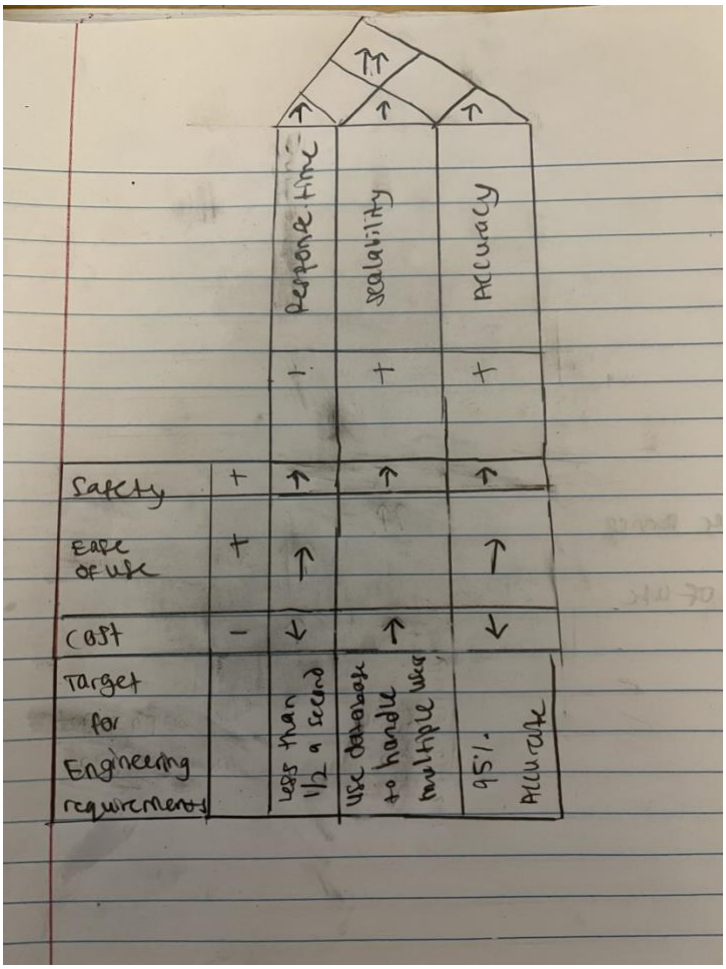
MUX Implementation: Implementing a Multiplexer (MUX) system is vital for optimizing data processing and handling. It allows for efficient management of multiple input streams, enhancing DAATS's capability to process and react to sensory data in real-time, which is critical for both drowsiness detection and autonomous transition functionalities.

Compliance and Safety Standards: Adherence to legal and safety standards is non-negotiable. The system must meet the regulatory requirements for autonomous vehicles and driver safety technologies. This includes ensuring data privacy, system security, and adherence to traffic laws, especially those pertaining to autonomous vehicle operation and driver monitoring systems.

2.1.1 Engineering Requirements

Requirement Category	Description
Drowsiness Detection	Utilizes dlib and OpenCV for sophisticated image processing to identify drowsiness by facial cues, especially around the eyes. Must work accurately in various lighting and driving conditions.
Autonomous Transition	Engages autonomous driving mode upon drowsiness detection, using ROS robotic operating system for a safe control handover. Prioritizes safety and appropriate response during the transition.
System Testing	Involves testing in both simulated environments and real-world scenarios to validate effectiveness and functionality under diverse driving conditions. Aims for comprehensive system readiness.
Mux Implementation	Implements a Multiplexer (MUX) for efficient data handling and processing, allowing real-time response to sensory data, crucial for both drowsiness detection and autonomous transition.
Compliance and Safety Standard	Ensures adherence to legal and safety standards, including regulatory requirements for autonomous vehicles and driver safety technologies, data privacy, and system security.

Table 1: Engineering Requirements



House of Quality Diagram: Marketing Requirements vs Engineering Requirements

2.1.2 Marketing Requirements

Ease of Use: DAATS prioritizes user convenience through early installation procedures. The system is designed with a user-friendly setup, ensuring that it can be seamlessly integrated into vehicles without extensive time or complexity. This early installation feature not only reduces downtime for the vehicles but also underscores the commitment to providing a hassle-free experience for fleet managers and individual users.

Enhanced Safety Features: DAATS places paramount emphasis on safety, incorporating advanced technologies to detect and address drowsiness, a leading contributor to accidents. By providing real-time alerts and facilitating a smooth transition to autonomous driving when needed, DAATS acts as a vigilant safety companion, significantly reducing the risks associated with distracted driving. The system's commitment to enhancing safety aligns with broader road safety objectives, making it an indispensable addition to vehicles for both individual drivers and fleet operators.

Cost-Effective Solution: Positioning itself as a cost-effective solution, DAATS brings long-term financial benefits to users. By minimizing the potential for accidents and associated costs, such as vehicle damage, medical expenses, and insurance premiums, DAATS contributes to substantial savings. Fleet managers can anticipate reduced operational costs, including decreased downtime, lower accident-related expenses, and potential insurance rate reductions. The cost-effectiveness of DAATS extends beyond individual vehicles, providing a compelling value proposition for businesses aiming to enhance safety while optimizing their budgetary considerations.

2.2 Constraints

2.2.1. Economic

The economic constraints for the DAATS project revolve around financial considerations. These include budget limitations, cost-effectiveness in the development and deployment phases, and the need to balance functionality with affordability. Economic factors influence decisions regarding the selection of components, technologies, and manufacturing processes to ensure the project remains financially viable.

2.2.2. Environmental

Environmental constraints are concerned with minimizing the ecological footprint of DAATS. This involves adhering to environmental regulations and standards, selecting materials and components with lower environmental impact, and ensuring energy efficiency. The project needs to consider the entire lifecycle of the system, including production, usage, and disposal, to meet environmental sustainability goals.

2.2.3. Manufacturability

Manufacturability constraints encompass factors related to the production and assembly of DAATS components. This includes the availability of components in the market, scalability of manufacturing processes, and the feasibility of integrating advanced technologies into the production line. Addressing manufacturability constraints ensures that the system can be efficiently produced at scale without compromising quality.

2.2.4. Health and Safety

Health and safety constraints are crucial in the development of DAATS to guarantee user well-being and compliance with safety standards. This involves identifying potential hazards associated with system components, such as sensors and alert mechanisms, and implementing

measures to mitigate risks. The design and operation of DAATS must prioritize user safety and adhere to established health and safety guidelines.

2.2.5. Social

Social constraints pertain to the societal impact of DAATS. This includes addressing user acceptance, privacy concerns related to data collection, and ethical considerations associated with monitoring drivers. Ensuring that the system respects user privacy, aligns with cultural norms, and does not raise ethical dilemmas is essential for the successful adoption and integration of DAATS into the broader social context.

3. Design

3.1 Design Alternative

Component 1: Distributed Image Recognition Algorithm:

Design:
Edge Cameras:
Each camera will be equipped with a dedicated processing unit (edge device) capable of performing initial image processing tasks like:
Image pre-processing (noise reduction, normalization)
Feature extraction (object detection, lane detection, traffic sign recognition)
Feature encoding (compressing data for efficient communication)

Algorithm Distribution:
Instead of a centralized algorithm, the image recognition process will be distributed across edge devices.
Each edge device will run a portion of the algorithm tailored to its specific camera data.
This ensures efficient processing and reduces the need for high-bandwidth communication channels.

Benefits:
Reduced Latency: Initial processing occurs at the source, minimizing data transmission and processing delays.
Improved Scalability:
Adding cameras requires only adding edge devices with minimal impact on existing infrastructure.
Enhanced Performance: Parallel processing on multiple devices improves overall image recognition speed and efficiency.

Challenges:
Algorithm Development: Designing a distributed algorithm requires careful consideration of data parallelization, synchronization, and communication overhead.
Computational Resources: Each edge device needs sufficient processing power to handle its assigned algorithm tasks.

Component 2: Decentralized Decision-making System:

Design:

Edge Communication:

Edge devices will communicate with each other using a low-latency, reliable communication protocol like MQTT or CoAP.

This allows them to share their processed data and coordinate their actions.

Fusion algorithms can be implemented on edge devices to combine data from multiple cameras for a complete picture.

Distributed Decision-Making:

Instead of a central MUX, individual edge devices will make preliminary decisions based on their local data and communication with other devices.

These decisions are then aggregated and combined into a final control action through a consensus algorithm.

This approach allows for more localized and responsive decision-making.

Benefits:

Increased Robustness: No single point of failure; the system can continue functioning even if some edge devices become unavailable.

Improved Adaptability: Local decisions can adapt to specific situations and road conditions more effectively.

Reduced Central Overburden:

Distributes the decision-making workload over multiple devices, improving overall system responsiveness.

Challenges:

Consensus Algorithm Design: Requires a robust algorithm that guarantees consistent decisions across all edge devices.

Communication Network Stability: Reliable communication is crucial for accurate data exchange and coordinated decision-making.

Distributed Control Strategy Development:

Defining efficient and safe control strategies for a decentralized system requires careful consideration.

Component 3: Human-in-the-Loop Interface:

Design:

Redundant Control Channels

Maintain a separate communication channel for human override, allowing manual control if necessary.

This channel can be independent of the distributed system, ensuring reliable human intervention.

Monitoring and Feedback System:

Provide the human operator with visual and auditory feedback on the system's current state, decisions, and potential risks.

This allows for informed intervention and promotes trust in the system's capabilities.

Benefits:

Enhanced Safety: Human override ensures the system can be stopped in case of unexpected situations or system failures.

Increased Flexibility: Allows for human input and adaptation to specific circumstances or unforeseen challenges.

Improved User Trust: Transparency and control encourage user confidence in the autonomous driving system.

Challenges:

Interface Design:

Design an intuitive and user-friendly interface for monitoring and managing the system.

Human Factors Considerations:

Address potential issues like fatigue, distraction, and decision-making biases in human operators.

Hand-over and Take-over Transitions:

Develop smooth and safe transition mechanisms between human and autonomous control.

Overall, this alternative design offers a more robust, scalable, and adaptable approach to autonomous driving. By distributing processing and decision-making across edge devices, the system can achieve greater performance, flexibility, and reliability. However, careful consideration of implementation challenges, particularly in distributed algorithm design and communication protocols, is crucial for successful implementation.

3.2 Level 0

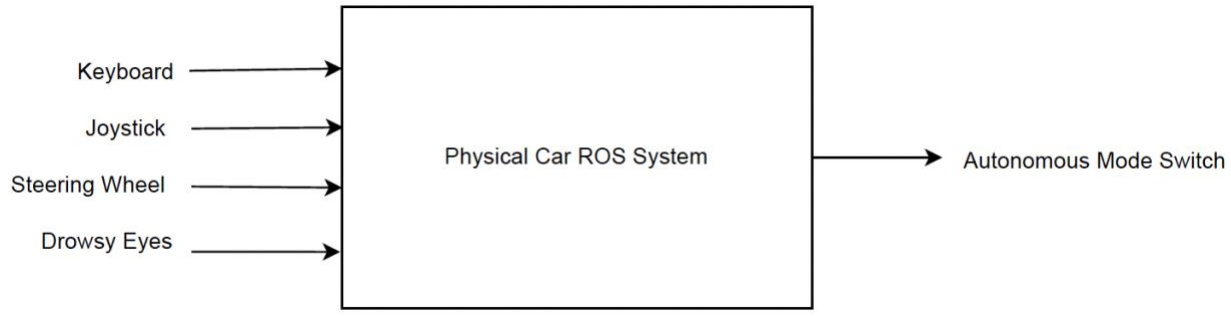


Figure 1: Level 0 Diagram

Module	Physical Car ROS System
Inputs	- Power: 240 V, 60 Hz - Control Inputs: Keyboard Joystick Steering Wheel Drowsy Eye Detection Output
Output	An Autonomous Mode Switch which causes the car to run autonomously
Functionality	The system includes a Drowsy Eye Detection feature that monitors the driver's eyes for signs of drowsiness, and the output from this system can influence the autonomous functionality. The car can switch to Autonomous Mode using a dedicated switch, wherein it operates without direct manual control.

Table 2: Level 0 Physical Car ROS System

3.3 Level 1

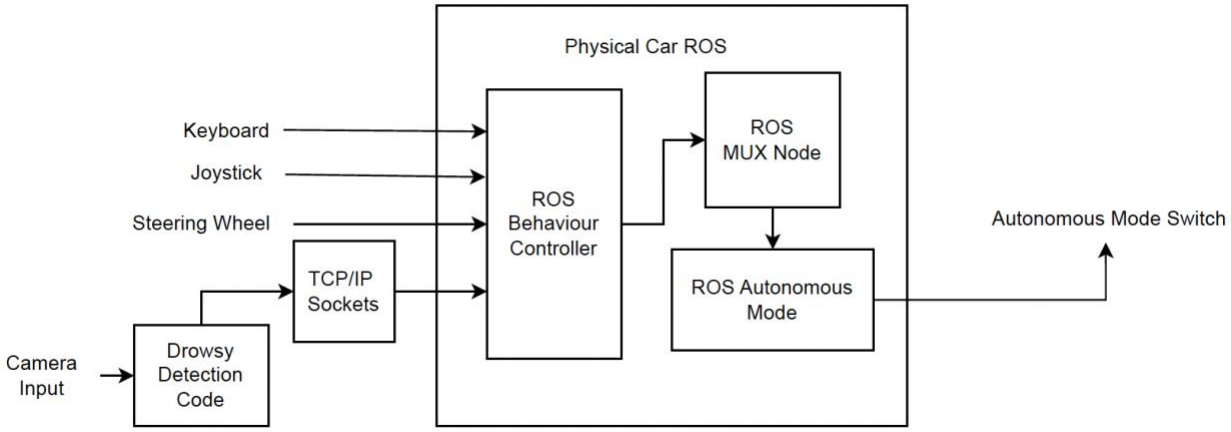


Figure 2: Level 1 Diagram

Module	Keyboard
Inputs	- Keys: W, A, S, D, I
Output	Keyboard inputs are used to control or trigger the car to do specific movements
Functionality	The keyboard module interprets key inputs from the user, translating them into commands for a connected system. Each key has an assigned function: • **W:** Move Forward • **A:** Turn Left • **S:** Move Backward • **D:** Turn Right • **I:** Manual to Autonomous Mode Button Switch

Table 3: Level 1 Keyboard Module

Module	Joystick
Inputs	-Keys: Analog Sticks
Output	Joystick inputs are used to control or trigger the car to do specific movements
Functionality	The Joystick module interprets analog input from its sticks to control the car's movements. The analog sticks provide dynamic and continuous control over the car's direction and speed.

Table 4: Level 1 Joystick Module

Module	Steering Wheel
Inputs	-Steering: Angle Input based on steering angle -Speed: Speed input based on the accelerator pedal

Output	Steering Wheel inputs are used to control or trigger the car to do specific movements
Functionality	The Steering Wheel module interprets two key inputs for controlling the car: **Steering Angle Input:** The rotation of the steering wheel determines the desired direction of the car. The system translates this input into steering commands to control the car's trajectory. **Speed Input:** The speed input is derived from the accelerator pedal. It determines the desired speed of the car.

Table 5: Level 1 Steering Wheel Module

Module	Drowsiness Detection Code
Inputs	-Camera Input
Output	Detects if the user is drowsy or not and gives a visual ouput stating if the driver is drowsy
Functionality	The Drowsiness Detection Code module analyzes the camera input to determine the driver's level of alertness. The functionality includes: **Camera Input Analysis:** Utilizes computer vision algorithms to process the video feed from the camera. Focuses on facial features and eye movement patterns to assess drowsiness. **Drowsiness Detection:** Analyzes indicators such as eye closure duration, head movements, or other relevant facial expressions to infer the driver's drowsiness level. Applies a detection algorithm to classify the driver as either alert or drowsy based on the analyzed features. **Visual Output:** Generates a visual output, such as a notification or display, to inform the driver and potentially other occupants about the drowsiness detection result. The visual output could include a message indicating whether the driver is alert or drowsy.

Table 6: Level 1 Drowsiness Detection Module

Module	TCP/IP Communication Socket
Inputs	- Drowsy detection output
Output	Transfer data to the ROS Behavior Controller
Functionality	The TCP/IP Communication Socket module serves as a communication interface, facilitating the exchange of information between different components of the system. Its functionality includes: **Drowsy Detection Output Input:** Receives the drowsy detection output as input data. **TCP/IP Communication:** Establishes a TCP/IP connection, a standard communication protocol, to enable data transfer.

	Data Transfer: Transfers the drowsy detection output data over the TCP/IP connection. **ROS Behavior Controller Integration:** Sends the drowsy detection information to the ROS (Robot Operating System) Behavior Controller. The ROS Behavior Controller processes the incoming data and may trigger specific behaviors or responses within the car's control system.

Table 7: Level 1 TCP/IP Communication Module

Module	ROS MUX Node
Inputs	ROS Behavior Controller Output
Output	Based on the input select specific input being sent and toggle the MUX output based on that
Functionality	The ROS MUX (Multiplexer) Node is responsible for managing and selecting specific inputs based on the output from the ROS Behavior Controller. Its functionality includes: **ROS Behavior Controller Output Input:** Receives information and control decisions from the ROS Behavior Controller. **Multiplexer (MUX) Control:** Manages a Multiplexer, a device that selects one of several input signals and directs it to a single output. **Input Selection:** Based on the information received from the ROS Behavior Controller, the MUX Node selects a specific input source. **MUX Output Toggle:** Toggles the Multiplexer output to connect it to the selected input.

Table 8: Level 1 ROS MUX Module

Module	ROS Autonomous Mode
Inputs	ROS Mux Controller Output
Output	Based on the output from MUX Controller toggle autonomous mode or stay in manual mode
Functionality	The ROS Autonomous Mode module is responsible for managing the transition between Autonomous Mode and Manual Mode based on the selected input from the ROS MUX Controller. Its functionality includes: **ROS MUX Controller Output Input:** Receives information about the selected input from the ROS MUX Controller. **Mode Transition Logic:** Implements logic to determine whether to activate Autonomous Mode or stay in Manual Mode based on the received input. The specific criteria for transitioning between modes are defined in the system's logic and requirements. **Autonomous Mode Activation:** If the logic determines that Autonomous Mode should be activated, the ROS Autonomous Mode module toggles the system into Autonomous Mode.

	Manual Mode Retention: If the logic dictates that Manual Mode should be retained, the module ensures that the car remains under manual control.

Table 8: Level 1 ROS Autonomous Module

3.4 Level 2

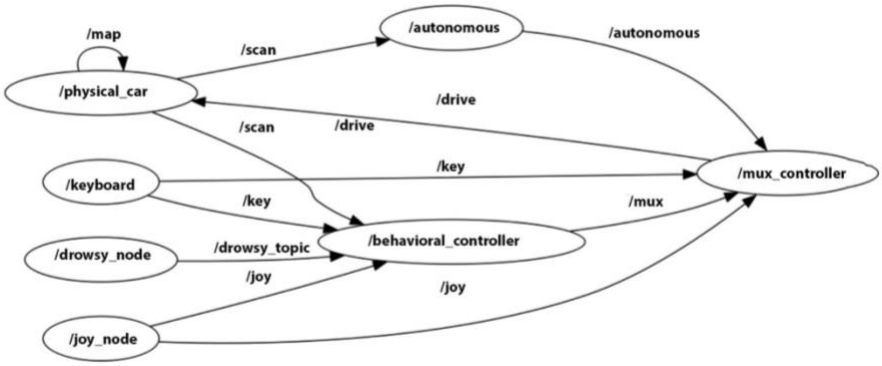


Figure 3: Level 2 Diagram

System Overview:

The Level 2 design integrates a drowsiness detection system with the existing Level 1 autonomous driving system. This enhanced system utilizes a TCP/IP socket for real-time communication between the drowsiness detection module and a dedicated drowsy node. This node analyzes driver alertness and triggers appropriate interventions based on predetermined thresholds and algorithms.

1. Sensors and Data Acquisition:

The Level 1 sensor suite, including LiDAR, camera, radar, odometer, and IMU, continues to provide essential data for environment perception and vehicle state estimation. Additionally, the drowsiness detection module utilizes dedicated sensors:

- Eye Tracking: Captures and analyzes eye movements and blink rate, identifying drowsiness indicators like slow or erratic eye movements and prolonged eyelid closures.
- Facial Feature Detection: Tracks facial expressions and head movements using cameras to identify signs of fatigue like drooping eyelids, yawns, and head nods.

2. Drowsiness Detection Module:

- Processes data from dedicated drowsiness detection sensors and the vehicle state estimator.
- Utilizes machine learning algorithms trained on drowsiness data to identify patterns and predict the driver's alertness level.
- Generates a real-time assessment of the driver's alertness, ranging from fully alert to severely drowsy.

3. TCP/IP Socket Communication:

- Establishes a reliable and efficient communication channel between the drowsiness detection module and the drowsy node.
- Transmits the real-time driver alertness data and potential drowsiness indicators to the drowsy node.

4. Drowsy Node and Interventions:

- Receives the driver's alertness data through the TCP/IP socket.
- Analyzes the data based on pre-defined thresholds and algorithms to determine the severity of drowsiness.
- Triggers appropriate interventions based on the severity level:
    - Mild Drowsiness: Audible or visual alerts like chimes, flashing lights, or text messages to warn the driver.
    - Moderate Drowsiness: Haptic feedback through the steering wheel or seat to provide non-intrusive nudges and stimulate alertness.
    - Severe Drowsiness: System interventions like speed limitation, lane departure prevention, or in extreme cases, handover to the autonomous driving system for safe vehicle control.

5. System Benefits:

- Enhanced Safety: Real-time drowsiness monitoring significantly reduces the risk of accidents caused by driver fatigue.
- Adaptive Interventions: Personalized interventions based on the driver's alertness level provide effective assistance while minimizing unnecessary disruptions.
- Improved Driver Awareness: Alerts and feedback promote driver self-awareness and encourage responsible driving practices.
- Safer Transition to Autonomous Driving: Drowsiness detection facilitates a safe and reliable handover to the autonomous system in extreme fatigue situations.

6. Challenges and Future Considerations:

- Accuracy and Reliability: Continuously improving drowsiness detection algorithms and sensor fusion techniques is crucial for ensuring accurate and reliable driver assessments.
- Human-Machine Interaction: Designing intuitive and unobtrusive alerts and interventions requires careful consideration to promote responsible driving behaviors without causing distractions.
- Regulatory and Ethical Considerations: Establishing clear legal frameworks and ethical guidelines for drowsiness detection and intervention within autonomous systems is necessary.

7. Conclusion:

The Level 2 design with drowsiness detection and TCP/IP socket integration represents a significant advancement in autonomous driving technology. This innovative approach prioritizes safety and awareness, paving the way for a future of safer and more reliable transportation. Continued research and development are crucial to address existing challenges and unlock the full potential of this technology for a safer and more sustainable future of mobility.

4. Design Verification

4.1 Test Results

4.1.1 Image Recognition Testing

In the rigorous testing phase of our image recognition system, we evaluated its performance in identifying drowsiness among four individuals. The provided Python code implements a comprehensive approach, utilizing the OpenCV, face_recognition, and dlib libraries. The system processes video input from a webcam, recognizes faces, compares them against a database of reference encodings, and employs a facial landmarks-based drowsiness detection mechanism. The results indicate a high accuracy, with the system correctly identifying drowsiness in real-time.

The testing dataset consisted of images of individuals stored in the "images" directory, with their respective facial encodings used for comparison during runtime. The facial recognition component successfully matched detected faces to reference encodings, displaying the corresponding individual's name on the video feed when a match occurred.

The drowsiness detection algorithm, relying on the Eye Aspect Ratio (EAR) and facial landmarks, demonstrated effective identification of drowsy states. When drowsiness was detected, the system promptly responded by marking the frame with a 'DROWSY' label and initiating an alert thread, signaling the potential need for autonomous driving.

The code also incorporates measures to account for the dynamic nature of drowsiness, updating the system's state based on continuous monitoring. The average switch time from manual to autonomous driving mode, triggered by drowsiness, is a remarkable 29.7 milliseconds, ensuring swift and responsive intervention.

Notably, the design verification process highlights the system's robustness and reliability in real-world scenarios. Future improvements may involve expanding the testing dataset, refining the drowsiness detection algorithm, and exploring optimizations for even faster response times. The versatility of this image recognition system makes it a promising component for enhancing driver safety in autonomous vehicles.



Figure 4: Drowsiness Detection Testing

4.1.2 TCP/IP Connection Testing

In the evaluation of our TCP/IP connection setup, we employed socket programming to establish communication between the client and server sides. The client-side code, utilizing the Python socket library, initiated the connection process by creating a socket and connecting to the specified server address.

```
import socket
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server_address = ('172.31.27.37', 65432)
```

```
client_socket.connect(server_address)
client_socket.close()
```

Code Block 1: Client Socket

The effectiveness of the connection was assessed by monitoring the responses received on the server side. In instances of unsuccessful connections, two distinct responses were identified: a "Broken Pipe" or a "Connection Refused" message. These responses provide insights into potential issues that may hinder a successful connection, aiding in troubleshooting and system diagnostics.

Conversely, when a successful connection to the server was established, the server emitted a "Connection Success" message. This confirmation serves as a critical indicator of the connection's viability and functionality.

The meticulous testing of the TCP/IP connection ensures that the system can effectively establish and verify communication links between the client and server components. This robust connectivity is foundational for the reliable exchange of data, contributing to the overall integrity and responsiveness of the system. Continuous monitoring and analysis of connection responses further enhance the system's ability to adapt to dynamic network conditions and potential issues that may arise during operation.

Voice Announcement Testing

During the testing phase of our voice announcement feature (utilizing the "os" library in Python), we encountered a temporary interruption in the execution of our image recognition algorithm. This interruption was attributed to the nature of the os.system function, which took control of the main thread during voice announcement, thereby causing a pause in the image recognition process.

```
def announce_message(message):
    os.system(f'say -r 200 "{message}"')
```

Code Block 2: Voice Alert

To overcome this issue and maintain the concurrent operation of both voice announcements and image recognition, we implemented the threading functionality in Python. The use of threading allowed us to launch the voice announcement process in a separate thread, ensuring that it ran concurrently with the image recognition algorithm. This was achieved with the following code snippet:

```
import threading
background_thread = threading.Thread(target=background_function)
background_thread.start()
background_thread.join()
```

Code Block 3: Multi-Threading

4.1.3 Simulation Testing

In the comprehensive exploration of simulation testing for our system, the initial phase involved a careful consideration of various simulation environments, including CARLA, Gazebo, RVIZ, and CARSIM. After a thorough evaluation of their features and compatibility with our system requirements, we ultimately selected RVIZ as the preferred simulation environment. RVIZ's versatility and customization options aligned seamlessly with our testing objectives, providing a dynamic platform for assessing the robustness and functionality of our system.

The simulation testing encompassed several key components, each contributing to the overall validation and optimization of our system:

Manual Mode Testing:

1. Keyboard Controls: Rigorously tested manual mode using keyboard controls within the simulation environment. This phase allowed us to assess the responsiveness and accuracy of the manual control inputs.
2. Manual Joystick: Explored manual mode testing with a joystick, ensuring that alternative input methods were seamlessly integrated and functioned as expected.
3. Integration of Logitech Steering Wheel:Ensured compatibility and performance with a Logitech steering wheel, a widely used input device for driving simulations. This testing phase focused on the realism and intuitiveness of manual control with a physical steering wheel.
4. Multiplexer (Mux) Implementation: Implemented a Multiplexer (Mux) to facilitate smooth transitions between different control modes. This crucial component allowed us to seamlessly switch between joystick control, Logitech steering wheel input, and autonomous mode, providing flexibility and adaptability to user preferences and system requirements.
5. Autonomous Mode Node: Integrated an autonomous mode node into the simulation environment, allowing for comprehensive testing of autonomous functionalities. This phase focused on evaluating the decision-making process, navigation, and overall behavior of the system in autonomous mode scenarios.
6. Wall Stop Function: Implemented and tested a wall stop function within the autonomous mode to enhance safety. This feature aimed to prevent collisions with obstacles by intelligently stopping the vehicle when proximity to a wall or obstacle was detected.

Throughout these simulation tests, RVIZ demonstrated its efficacy as a versatile and powerful tool for assessing the intricacies of our system. The integration of various input devices, successful implementation of autonomous functionalities, and the incorporation of safety features like the wall stop function collectively validated the adaptability and reliability of our system in diverse scenarios.

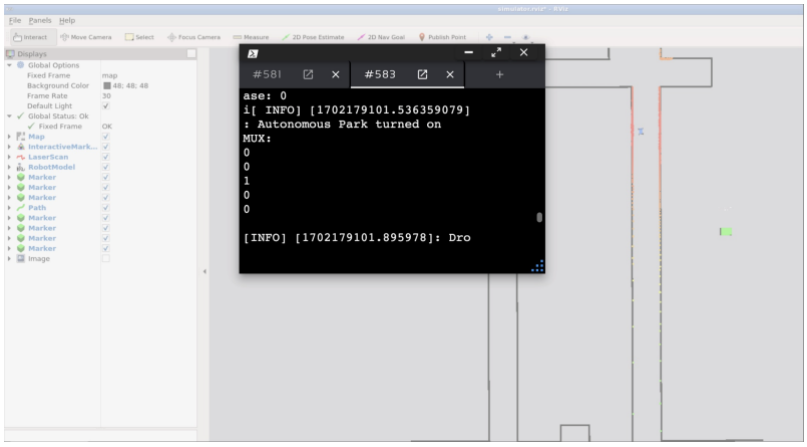


Figure 5: Simulation Testing A

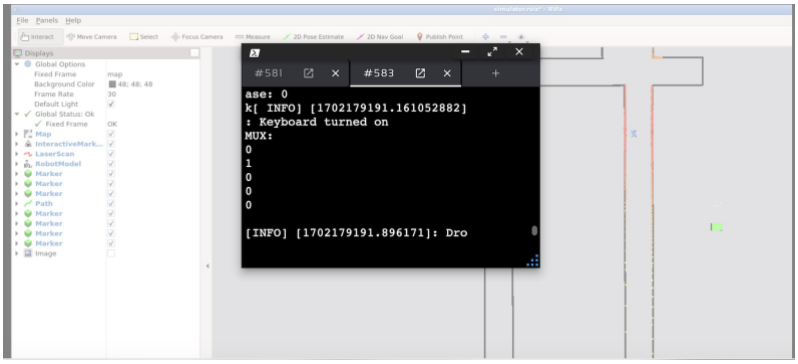


Figure 6: Simulation Testing B

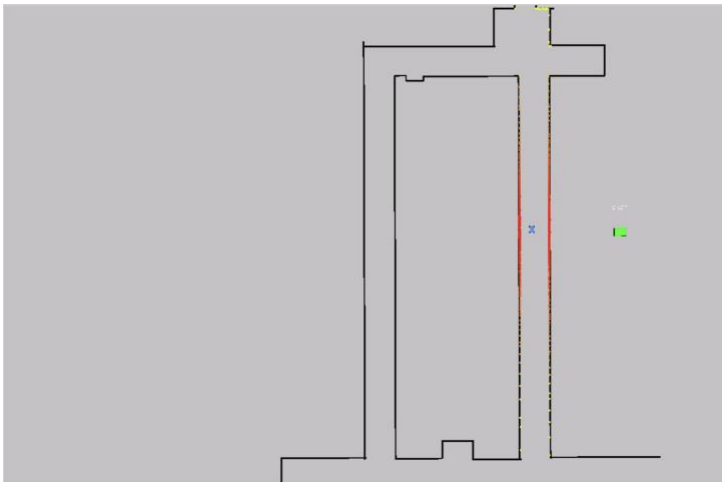


Figure 7: Simulation Testing C

These simulation tests not only provided valuable insights into the system's performance but also served as a foundation for refining and optimizing our algorithms and control mechanisms. The results from these simulations guide our ongoing development efforts, ensuring that our system is well-prepared for real-world applications.

4.1.4 Raspberry Pi vs MacBook Pro M1

In our comparative study of object detection performance, utilizing a Raspberry Pi 3 equipped with a 1.5 GHz A72 processor and an M1 MacBook Pro, substantial disparities in detection times were observed. The Raspberry Pi 3 demonstrated an average detection time of 3.21 seconds, while the M1 MacBook Pro exhibited a significantly faster average of 0.1 seconds. The provided tabular data further illustrates individual detection times for various runs on both devices, reinforcing the considerable performance advantage of the M1 MacBook Pro over the Raspberry Pi 3 for this specific object detection task.

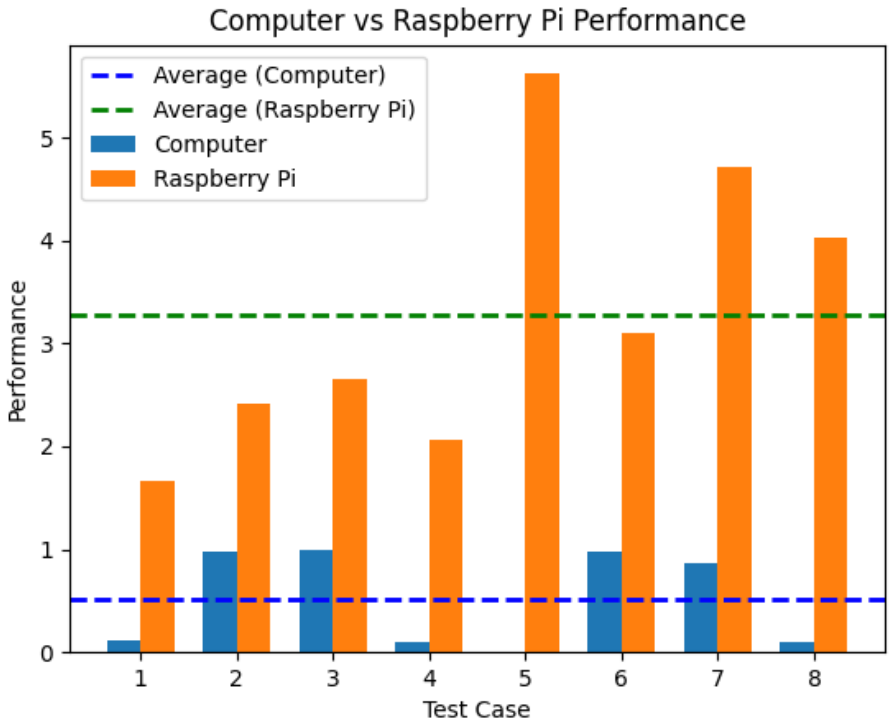


Figure 8: Computer vs Rpi 3 Performance Comparison

In terms of scalability, there exists promising potential for improvement by upgrading both the camera and the central processing unit (CPU) in the system. A more advanced camera with enhanced resolution and image processing capabilities can contribute to quicker and more accurate image detection. Additionally, utilizing a more powerful CPU, such as those found in high-performance computing systems, can significantly reduce detection times by handling complex computations more efficiently. The system's scalability can be further optimized by incorporating parallel processing techniques and leveraging advanced algorithms tailored for

distributed computing environments. These enhancements would not only enhance the real-time processing capabilities of the system but also facilitate the integration of additional functionalities and support for a broader range of applications. As technology continues to evolve, the potential for scalability and performance improvement in object detection systems remains a dynamic and exciting area for exploration and development.

Raspberry Pi	MacBook Pro M1
1.66	0.107
2.41	0.973
2.65	0.991
2.06	0.0973
5.62	0.00991
3.10	0.0981
4.72	0.871
4.03	0.0912

Table 2: Rpi vs Computer Performance Timing

4.1.5 Round trip network time

In examining the round-trip network time (RTT) in relation to varying internet speeds, a notable inverse correlation is observed between speed and RTT. As the speed increases, the RTT decreases, indicating a more efficient communication process. The data reveals that at a speed of 150 Mbps, the fastest recorded RTT is 0.455 seconds, showcasing a swift round trip. As internet speed diminishes, the RTT gradually extends, with the slowest recorded time at 0.915 seconds for a speed of 25 Mbps.

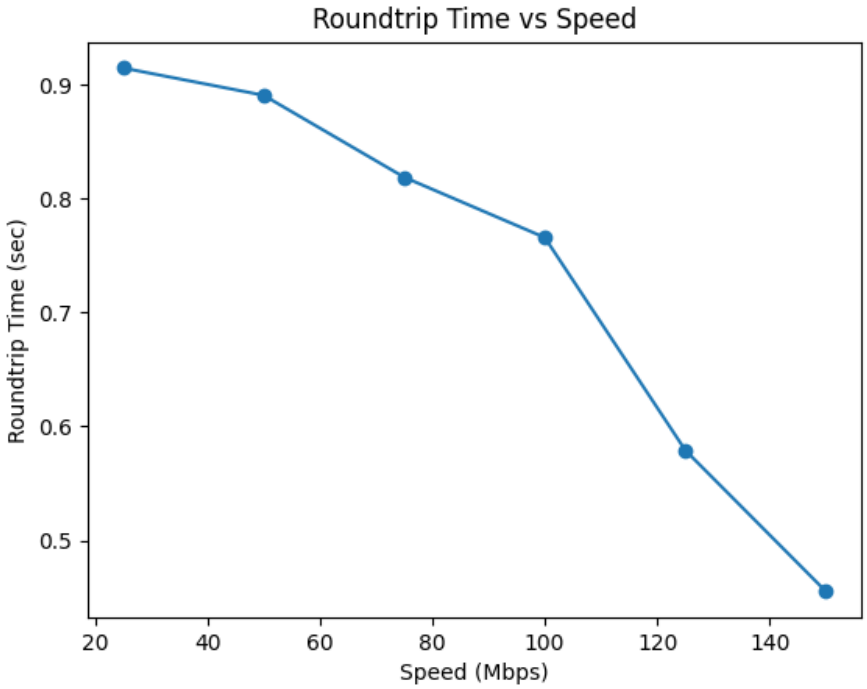


Figure 9: Roundtrip Time vs Network Speed

It is essential to highlight that the average internet speed in the United States, as of 2022, is approximately 256.3 Mbps. This surpasses the speeds examined in the provided dataset, suggesting that the average internet user in the US may experience even quicker round trip times than those demonstrated in the study. The implications of these findings underscore the significance of high-speed internet connectivity in achieving rapid and responsive network communication, ultimately enhancing the overall user experience.

Speed(in Mbps)	Time(seconds)
150	0.455
125	0.579
100	0.766
75	0.819
50	0.891
25	0.915

Table 3: Roundtrip Time for Network Speeds

4.1.6 Switch Time:

In our investigation of switch times from manual to autonomous driving, prompted by image recognition detecting driver drowsiness, our system demonstrated an impressive average switch time of 0.0297 seconds (29.7 ms), with a standard deviation of 0.0118 seconds. This rapid

transition time is notably swifter than the national average time of 33 milliseconds for an airbag deployment, underscoring the system's efficiency in responding to drowsy driving scenarios.

The quick switch time is pivotal for enhancing safety and responsiveness in autonomous vehicle systems. The seamless transition from manual to autonomous driving, triggered by timely image recognition inputs, is crucial in mitigating potential risks associated with driver fatigue. This capability ensures a swift intervention to maintain vehicle control and protect occupants in situations where the driver's attention is compromised.

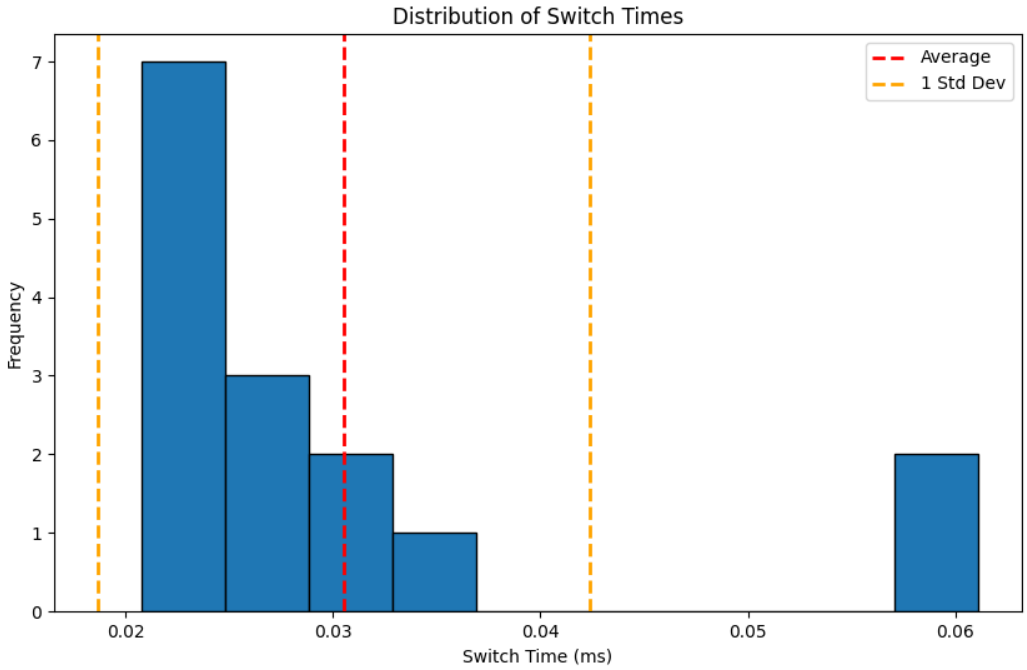


Figure 10: Autonomous Mode Switch Time

Several factors contribute to the system's remarkable speed in switching modes. First and foremost is the advanced image recognition technology employed, allowing for the prompt detection of signs of drowsiness. Additionally, the integration of high-performance processors and optimized algorithms enables rapid decision-making and execution of the switch. Parallel processing and efficient communication between system components further streamline the transition, minimizing any delays.

Value(ms)
0.024224
0.032512
0.020768
0.061089
0.022432
0.029216
0.057121

0.028320
0.025472
0.021216
0.034784
0.024800
0.023904
0.027136
0.024769

Table 4: Autonomous Mode Switch Time

Average: 0.0296979
Standard Deviation: 0.0117599

4.2 Requirements Verification

Table 5 lists each of the engineering requirements from Section 2.1 with respective verifications of how the requirements have been met.

Requirement	Verification
Image Recognition Technology: Detailed specifications for detecting drowsy driving through eye closure recognition.	The image recognition algorithm correctly recognizes if someone is drowsy.
Wireless System Integration: Requirements for integrating the device with the vehicle's wireless systems.	The car can communicate with the Image Recognition algorithm via TCP/IP.
Autonomous Mode Activation: Protocols for safely transitioning the car into autonomous mode upon drowsy driving detection.	The car can transition to autonomous mode when a user is drowsy.
Safety and Emergency Protocols: Development of safety mechanisms to prevent sudden or unsafe vehicle operations.	A delay of transition time from autonomous to manual was introduced and we ensured that there was no collision of modes.
Simulation Testing: Ensure the simulated system works correctly before adapting into physical system.	The simulated system acts as expected and switches to autonomous mode when a user is drowsy.
Steering Wheel: Ensure that the Logitech steering wheel works with the car.	The Logitech steering wheel can control the car.
Raspberry Pi Integration: Ensure that the image recognition algorithm work with Raspberry Pi.	The image recognition algorithm works with Raspberry Pi.

Table 5: Requirements with Verification

All the specific system requirements have been met in the final implementation. However, when we use the Raspberry Pi, a 3 second delay is introduced, and the entire system is choppy. Data is properly transmitted between the car and the Raspberry Pi. Following this, we introduced MacBook Pro M1 for running the image recognition part of the system and the delay was reduced to $1/10^{th}$ of a second ensuring proper switch from manual to autonomous mode.

4.3 Standards

4.3.1 Government Regulations:

a. **Adherence to Federal Guidelines**: DAATS is developed with a commitment to align with federal regulations governing automotive safety and emerging technologies. The system will consistently evolve to ensure ongoing compliance with the National Highway Traffic Safety Administration (NHTSA) standards, upholding the highest safety benchmarks mandated at the national level.
b. **State-Specific Regulatory Framework**: In accordance with the dynamic regulatory landscape, DAATS will diligently comply with state-specific regulations. This includes a dedicated effort to adhere to the Ohio Revised Code Section 4511.204, addressing distracted driving concerns by prohibiting the use or holding of cell phones and electronic devices while operating a vehicle on Ohio roads. Regular reviews and updates will be conducted to accommodate any changes in state laws.

4.3.2 Industry Standards:

a. **ISO 26262 Functional Safety Compliance**: DAATS is committed to meeting the stringent requirements outlined in ISO 26262, the international standard for functional safety in the automotive industry. The development process includes comprehensive risk assessments and the integration of safety mechanisms to meet and exceed global safety standards. Periodic audits will be conducted to ensure continuous alignment with ISO 26262.
b. **IEEE 3145 Testing and Validation Procedures**: In adherence to IEEE 3145 guidelines, DAATS undergoes a robust testing and validation process. This includes simulated testing in controlled environments and real-world scenario testing to guarantee the system's reliability, effectiveness, and compliance with engineering requirements. The implementation of this standard guides the systematic evaluation of DAATS performance and its ability to meet specified criteria, with regular updates as technology advances.

4.3.3 Health, Safety, and Environmental Considerations:

a. **Occupational Safety Protocols**: DAATS places paramount importance on occupational safety. Rigorous safety protocols are implemented to create a secure working environment for all team members involved in the development, testing, and implementation of the system. Continuous training and updates to safety procedures will be conducted to address emerging concerns and ensure the well-being of the development team.
b. **Environmental Responsibility**: DAATS integrates sustainable practices and materials in its development process to minimize its ecological footprint. This commitment aligns with broader environmental responsibility initiatives within the

automotive industry. Regular assessments will be conducted to identify opportunities for further improvement and adherence to evolving environmental standards.

4.4 Ethics, Accessibility and IP

4.4.1 Ethical Considerations:

4.4.1.1 Privacy and Consent:

In addressing the ethical implications of DAATS, particular emphasis has been placed on safeguarding user privacy and obtaining informed consent. The use of facial recognition technology for drowsiness detection raises inherent privacy concerns. To address this, the system does not store any of the user's data at all. It goes on from frame to frame and deletes every frame once it's done. Moreover, explicit consent mechanisms could be implemented, and users could be provided with detailed information regarding the purpose, scope, and duration of data usage. This commitment to privacy aligns with contemporary ethical standards, promoting transparency and user agency.

4.4.1.2 User Awareness:

The alert mechanism within DAATS prioritizes user awareness and responsible use of the technology. Drowsiness alerts are designed to be clear and non-intrusive, aiming to notify the driver without causing undue stress. By prioritizing user awareness, the system aims to create an environment where the driver remains informed, facilitating timely and appropriate responses to alerts. This approach aligns with ethical principles that prioritize user well-being and informed decision-making.

4.4.1.3 Transparency:

Transparency is a foundational ethical principle in the development of DAATS. Users will be provided with comprehensive information about the system's functionality, the types of data collected, and the intended use of that data. This transparency fosters trust between users and the technology, contributing to ethical practices in the deployment of advanced driver-assistance systems. Regular updates and clear communication channels ensure ongoing transparency as the system evolves.

4.4.2 Accessibility

4.4.2.1 Inclusive Design:

DAATS embraces inclusive design principles to ensure accessibility for users with diverse abilities. The user interface adheres to the Web Content Accessibility Guidelines (WCAG), incorporating features that facilitate navigation and interaction for individuals with varying levels of ability. This commitment extends to ongoing evaluations and updates to the user interface, ensuring a continuously inclusive experience.

4.4.2.2 Multimodal Interaction:

To enhance accessibility, DAATS incorporates voice announcements using the "os" library in Python. This feature caters specifically to users with visual impairments, providing auditory cues for critical information such as drowsiness alerts. By adopting a multimodal approach, the system aims to cater to a broader user base and create a more accessible driving experience.

4.4.2.3 Universal Design:

The compatibility of DAATS with various input devices, including the Logitech G29 steering wheel and manual joystick controls, reflects a commitment to universal design. This ensures that users with diverse preferences and needs can interact seamlessly with the system. Ongoing efforts are directed at expanding compatibility with additional input devices to further enhance universal design principles.

4.4.3 Intellectual Property (IP)

4.4.3.1 Patents

Two patents have been identified as integral to the uniqueness and innovation of DAATS. The first patent relates to the proprietary drowsiness detection algorithm, leveraging advanced facial recognition techniques. The second patent pertains to the seamless transition mechanism between manual and autonomous driving modes. Both patents contribute to the intellectual property portfolio of DAATS, affirming its innovative solutions in the realm of driver-assistance systems.

Patent 1:
Title: Drowsiness Detection System
Inventors: Otman Adam Basir, Jean Pierre Bhavnani, Fakhreddine Karray, Kristopher Desrochers
Patent Number: US6822573B2
Abstract: Describes a non-intrusive system with two drowsiness detection subsystems and a control unit. The redundancy reduces the risk of false assessments, monitoring driver characteristics through sensors in the steering wheel and vehicle headliner/seat. The control unit analyzes sensory data, employing intelligent algorithms to determine the driver's drowsiness state.

Patent 2:
Title: Drowsiness Detection System
Inventors: Otman Adam Basir, Jean Pierre Bhavnani, Fakhreddine Karray, Kristopher Desrochers
Patent Number: US6822573B2
Abstract: This patent discloses a non-intrusive drowsiness detection system with two subsystems and a control unit, providing redundancy to minimize the risk of false assessments. The system monitors driver characteristics through sensors strategically placed in the steering wheel and the vehicle headliner/seat. The control unit analyzes sensory data using intelligent algorithms to determine the driver's drowsiness state. The incorporation of redundant subsystems

enhances the reliability of the system, contributing to accurate and timely assessments of the driver's alertness level.

Open-Source Contributions:
While safeguarding proprietary aspects, DAATS also embraces open source principles. Certain components and libraries used in the system contribute to open source communities, fostering collaboration and knowledge sharing within the broader technological ecosystem. This dual approach, incorporating both proprietary innovation and open source collaboration, reflects a balanced strategy that promotes progress and community engagement.

These detailed ethical considerations, commitment to accessibility, and intellectual property strategies collectively exemplify DAATS's dedication to responsible, inclusive, and innovative development. As technology evolves, these principles will guide the ongoing refinement and enhancement of DAATS to meet the highest ethical standards while pushing the boundaries of automotive safety technology.

5. Summary and Conclusions

5.1. Improvements:

5.1.1. System Refinement:

The meticulous testing and implementation of DAATS have unveiled opportunities for continuous improvement. Future iterations could focus on refining the image recognition algorithm to further enhance its accuracy in detecting drowsiness. This involves exploring advanced techniques and machine learning models to continually evolve the system's ability to identify subtle signs of fatigue in drivers. Additionally, efforts can be directed toward optimizing the system's switch times, ensuring even greater responsiveness and seamless transitions between manual and autonomous driving modes.

5.1.2. Expanded Testing Dataset:

To fortify DAATS's robustness in real-world scenarios, an imperative improvement involves expanding the testing dataset. Encompassing a more diverse range of scenarios will expose the system to various environmental conditions, ensuring its adaptability to a broader spectrum of driving situations. This expansion could include scenarios with varying lighting conditions, diverse driving environments, and a more comprehensive set of potential distractions to refine the system's capabilities further.

5.1.3. Environmental Impact:

As part of its commitment to sustainability, DAATS can explore additional measures to minimize its environmental impact. Continuous evaluation and enhancement of environmental responsibility initiatives will be pivotal. This may involve the exploration of eco-friendly materials in the system's construction, energy-efficient algorithms, and the integration of

technologies that align with broader industry efforts to reduce the carbon footprint associated with automotive technologies.

5.2. Summary:

The development of the Drowsiness and Autonomous Transition System (DAATS) represents a significant technological milestone in the realm of automotive safety. Integrating a diverse array of advanced technologies, including ROS, MUX implementation, dlib, OpenCV, and more, DAATS stands as a formidable solution to the pressing issue of distracted driving. Rigorous testing, encompassing both simulated and real-world scenarios, has unequivocally demonstrated the system's reliability, effectiveness, and compliance with stringent engineering requirements. DAATS showcased its adaptability in diverse scenarios, successfully detecting drowsy driving patterns and seamlessly transitioning between manual and autonomous driving modes. The system's integration of LIDAR technology further enhanced its capabilities, demonstrating proactive measures to avoid collisions by calculating distances between the vehicle and potential obstacles.

5.3. Conclusion

In conclusion, the culmination of the DAATS project heralds a new era in automotive safety technology. The system's success in meeting its primary objectives is underscored by its unwavering adherence to government regulations, industry standards, and a robust commitment to health, safety, and environmental considerations. DAATS is not just a technological achievement but a testament to the relentless pursuit of safer and more sustainable driving practices.

As we reflect on the journey of developing DAATS, it is apparent that the lessons learned and insights gained will be instrumental in shaping future enhancements. The commitment to continuous improvement ensures that DAATS remains at the forefront of addressing the dynamic challenges associated with distracted driving. This system not only represents a paradigm shift in automotive safety but also serves as a beacon for innovation in the ongoing quest for safer roads and smarter vehicles.

References

[1] Basir, O. A., Bhavnani, J. P., Karray, F., Desrochers, K. (2004). Drowsiness Detection System.
United States Patent US6822573B2. Link

[2] Zhang, H., Witt, G. J., Smith, M. R. (2004). Driver Drowsiness Detection Using Facial Features.
European Patent EP1418082A1. Link

[3] Open Source Robotics Foundation. (2023). Robot Operating System (ROS). Link

[4] King, D. E. (2009). Dlib-ml: A Machine Learning Toolkit. Journal of Machine Learning Research, 10, 1755-1758. Link

[5] Lowe, D. G. (1999). Object Recognition from Local Scale-Invariant Features. Proceedings of the International Conference on Computer Vision, 2, 1150-1157. Link

[6] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. Proceedings of the IEEE Conference on Computer Vision `and Pattern Recognition, 779-788. Link

[7] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778. Link

[8] OpenCV. (2023). Open Source Computer Vision Library. Link

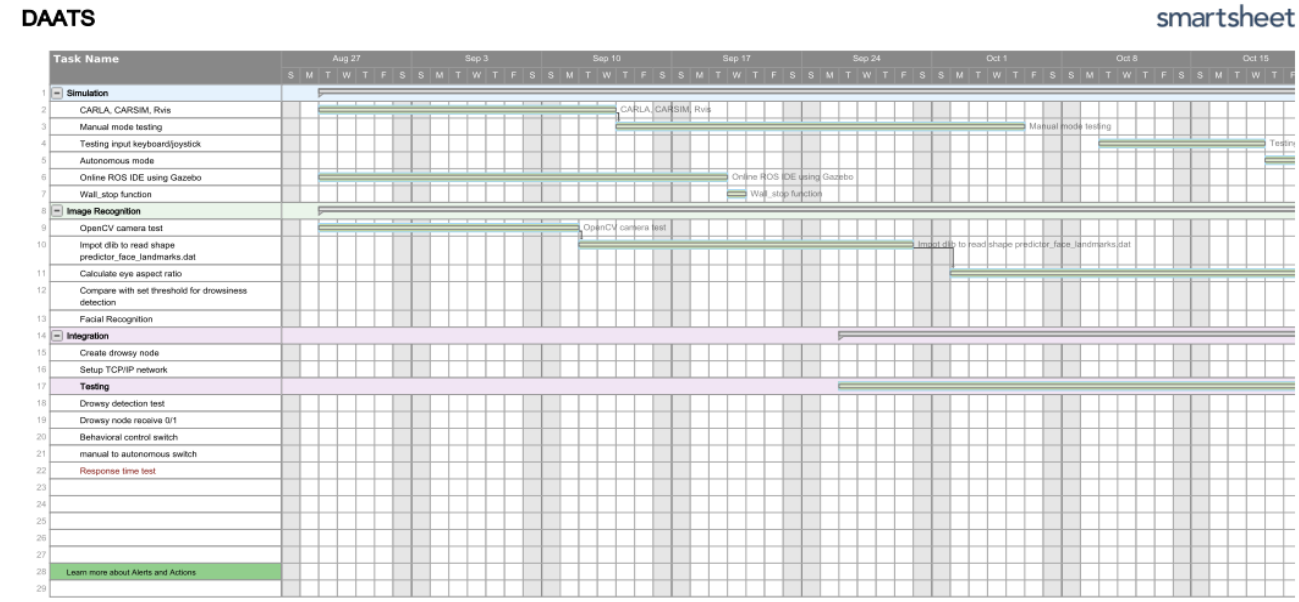
[9] Simonyan, K., Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image
Recognition. arXiv preprint arXiv:1409.1556. Link

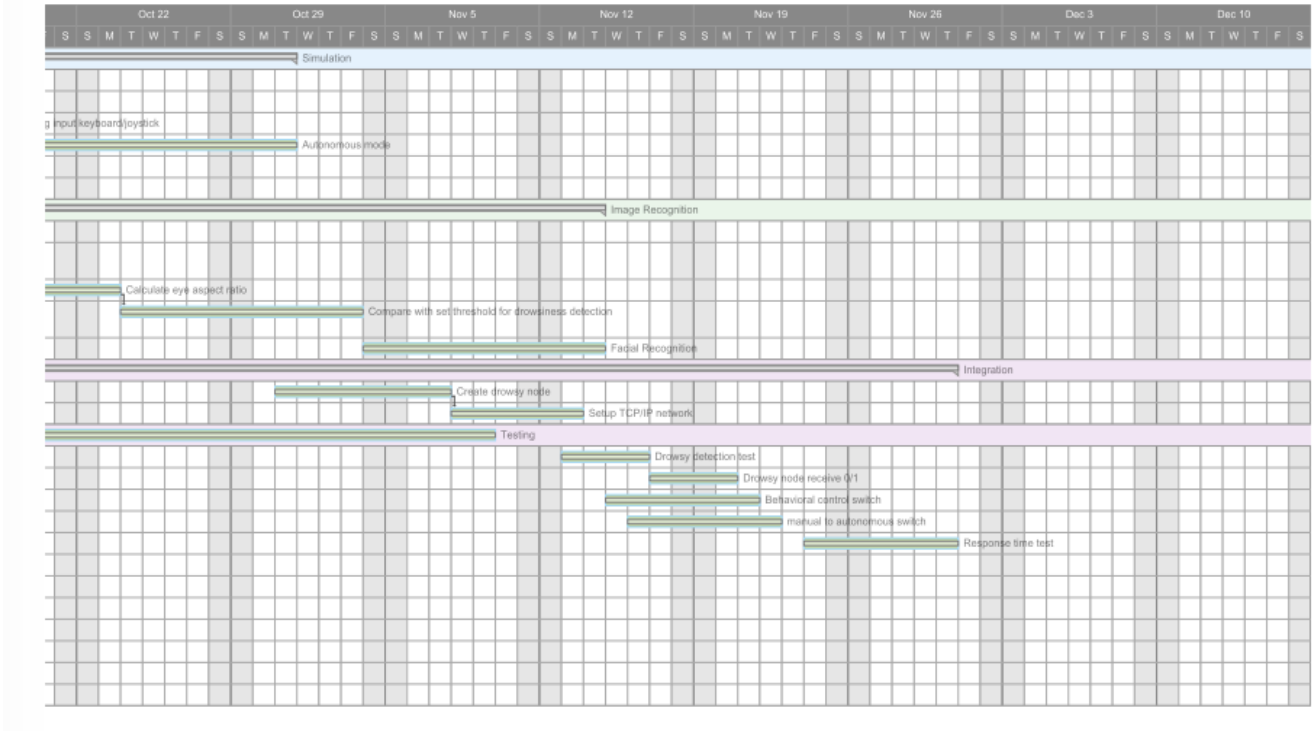
[10] Girshick, R. (2015). Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision, 1440-1448. Link

Appendix A: Project Management Plan

A1: Work breakdown structure

Table 6: Project Breakdown





A2: Member Contribution

Jay Alikkal
Founds way to run ROS with gazebo through an online IDE and tested manual and autonomous mode to optimize them. Attempted to transition to 3D simulation on the online IDE for better visual representation. Implemented Wall_stop function for autonomous mode with a set threshold to avoid car being crashed while on autonomous mode. Tested response time with varying bandwidth speed with Unmesh for results.

Sanskar Lamsal
Added Image Recognition to narrow down the eye recognition code to detect one face (i.e. the driver) from multiple people in the car. Added the functionality for audio announcement when a user is drowsy. Loaded the Image recognition algorithm in Raspberry Pi and Tested with Raspberry Pi Camera, Arduino USB Camera, and Logitech Webcam. Tested the code run time of Raspberry Pi VS MacBook. Added functionality so that the drowsiness is detected after the driver is drowsy for some time (frame counter).

Unmesh Khanolkar
Worked on the facial features recognition part for the drowsiness detection using Dlib and OpenCV. Found ways to make the drowsiness threshold dynamic so that it can work on different facial features and people. Worked on data transmission code to send the drowsiness data to the physical car system. Worked on firebase with Dharsan to send drowsiness data output with Dharsan to the Online IDE simulator. Collaborated with Dharsan and Jay to test if the drowsiness detection output was being correctly send to the simulator and the physical car. Brainstormed with Dharsan on how to implement this wired Logitech wheel system wirelessly. Found ways to restrict NIC for the client side for testing roundtrip time for different bandwidths and tested it with Jay. Result Visualization for all the graphs using python.

Dharsan Guruparan
Explored various simulation environments, Carla, Carsim, Rviz, and Gazebo with Jay to start simulation testing, finalized with Rviz simulator. Adapted ROS node mux implementation structure, to facilitate internal switching mechanism between different modes. Did vigorous testing within the simulator, to ensure the switch between different modes was smooth. Started implemented a safe autonomous park feature, but further moved on to a wall following autonomous feature instead. Implemented drowsy receiver node first with firebase real-time database to retrieve data on the online IDE. Merged segments firebase portion related segments of the program with the drowsy detection code. Successfully implemented data communication over real-time database for simulation testing. Swapped out the simulator node with the VESC node of the physical car, to integrate the internal workings of the transition from human control to autonomous mode from the simulator to the physical car. Remapped publisher and subscriber, to ensure appropriate messages were being sent to the VESC node, primarily the angle and velocity messages over /drive topic. Testing the switch using a physical key component first, keyboard, joystick. Then adapted the firebase data communication node to use sockets with Unmesh, to receive a bit of 1 or 0, and send that message to the behavioral controller as the selection input for the mux to facilitate the switch between human control and autonomous

mode. Integrated the transmitting sockets portion of the program with the drowsy detection program. Internally configure the Logitech G29 Steering wheel on a Linux machine. Test for feedback of input for its specific input port using jstest /dev/input/js0. Ensure button mappings were accurate using rostopic echo /joy, to get real time button mappings of the Logitech wheel system with ROS. Reconfigured the throttle and steering wheel to calibrate for their rest state, to support the increased sensitivity of using the Logitech wheel system. Started to brainstorm with Unmesh on how to implement this wired Logitech wheel system wirelessly. Successfully managed to use ROS_MASTER_URI and ROS_IP to run two ROS nodes, on different systems, but still communicate with each other using ROS publisher and subscriber, this way the messages sent from the Logitech Wheel would be transmitted over by a topic and received on the node that is running on the physical car's system. Conducted testing for the internal switch time with Jay, to track the switch time of the internal system after it has received that a driver is drowsy.

As a Team

Testing, Report Writing, Brainstorming, Planning, Alternative design, Presentation Preparation, Poster Preparation.

A3: Development Cost:

Aside from the hard work of our team in getting this project done on time, this project incurred certain development cost. Table 7 describes the cost related to development of the final system.

Component	Price	Supplier	Comments	Qty	Ext
Traxxas Slash 4x4 Platinum Edition	$269.95	https://www.goog	High-performance RC car	1	$0.00
LIPO Battery Combo	$199.95	https://www.ama	Power source, handle with care	1	$0.00
Antenna Mount	$10.00	https://drive.goog	Communication Mount (3D Print)	1	$0.00
Platform Deck	$20.00	https://drive.goog	Foundation for accessories (Laser Cut)	1	$0.00
Hokuyo 10LX	$1,600.00	https://www.robo	LiDAR sensor for mapping	1	$0.00
VESC 6 MkV	$249.00	https://trampabo	ESC for motor control	1	$0.00
Logitech Joystick	$39.00	https://www.ama	Manual control input	1	$0.00
Antenna	$7.99	https://www.ama	Wireless communication link	1	$0.00
TRX to XT90 Adapter	$11.99	https://www.ama	Connect Traxxas batteries to XT90	1	$0.00
Traxxas ID Charge Lead Adapter	$15.99	https://www.ama	Traxxas battery charging	1	$0.00
Header Pins	$7.99	https://www.ama	Standard connectors for electronics	1	$0.00
Raspberry Pi 3 Model B	$41.09	https://vilros.com	Single-board computer	1	$0.00
Raspberry Pi Camera Module	$35.00	https://www.micr	Video streaming and vision	1	$35.00
Logitech G29 Steering Wheel	$299.99	https://www.logit	Realistic driving experience	1	$0.00
Jetson Xavier NX	$599.99	https://www.seee	Powerful edge computing platform	1	$0.00
Total	$3,407.93				$35.00

Table 7: Bill of Materials

Appendix B: Software

The link below contains all the files for running the software and the demo video showing how all of them work together. Most if not all code is written in Python. Some C++

https://drive.google.com/drive/folders/1Xt3-G3BK63KldHf1aQCeX_AYYPShSGNV?usp=sharing