VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

An Autonomous Institute Affiliated to University of Mumbai

Department of Computer Engineering

Project Report on

Multi-Sensor Fusion for Enhanced Object Detection In Complex Environments

In partial fulfillment of the Fourth Year, Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai Academic Year 2023-24

Submitted by

Kaustubha Sahu (D17A, Roll No. 57)

Aaryan Manawat (D17A, Roll No. 41)

Pranav Ubarhande (D17A, Roll No. 69)

Abhiman Singh (D17A, Roll No. 65)

Project Mentor

Mr. Sanjay Mirchandani

-

(2023-24)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY

An Autonomous Institute Affiliated to University of Mumbai

Department of Computer Engineering

Certificate

This is to certify that ***Kaustubha Sahu, Aaryan Manawat, Pranav Ubarhande, and Abhiman Singh*** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on "***Multi-Sensor Fusion for Enhanced Object Detection In Complex Environments***" as a part of their coursework of PROJECT-II for Semester-VIII under the guidance of their mentor ***Prof. Mr Sanjay Mirchandani*** in the year 2023-24.

This thesis/dissertation/project report entitled ***Multi-Sensor Fusion for Enhanced Object Detection In Complex Environments*** by ***Kaustubha Sahu, Aaryan Manawat, Pranav Ubarhande, and Abhiman Singh*** is approved for the degree of Bachelor of Engineering

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date:

Project Guide: ----------

Project Report Approval

For

B. E (Computer Engineering)

This thesis/dissertation/project report entitled "***Multi-Sensor Fusion for Enhanced Object Detection In Complex Environments"*** by ***Kaustubha Sahu, Aaryan Manawat, Pranav Ubarhande, and Abhiman Singh*** is approved for the degree of Bachelor of Engineering

Internal Examiner

External Examiner

Head of the Department

Principal

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

---	---
Kaustubha Sahu (D17A/57)	Pranav Ubarhande (D17A/69)
---	---
Aaryan Manawat (D17A/41)	Abhiman Singh (D17A/65)

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to the Assistant Professor **Mr. Sanjay Mirchandani** for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr.(Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair ,** for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

Computer Engineering Department

COURSE OUTCOMES FOR B.E PROJECT

Learners will be to,

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solutions for the problem.
CO 4	Able to interpret the data and datasets to be utilized.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one's competency.
CO 11	Able to develop a professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Index

Chapter No.	Title	Page No.
1	**Introduction**	13
1.1	Introduction to the project	13
1.2	Motivation for the project	14
1.3	Problem Definition	15
1.4	Existing Systems	15
1.5	Lacuna of the Existing Systems	16
1.6	Relevance of the Project	16
2	**Literature Survey**	19
A	Brief overview of Literature Survey	19
B	Related Works:	19
2.1	Research Papers a. Abstract of the research paper b. Inference drawn from the paper	19
2.2	Patent search a. Title of the patent and year of the patent b. Summary of the patent c. Link	32
2.3	Comparison with the Existing Systems	33

3.	**Requirement Gathering for the proposed System**	35
3.1	Introduction to Requirement Gathering	35
3.2	Functional Requirements	35
3.3	Non-Functional Requirements	36
3.4	Hardware, Software, Technology and tools utilized	38
3.5	Constraints	41
4.	**Proposed Design**	43
4.1	Block diagram of the system	43
4.2	Modular design of the system	44
4.3	Detailed Design (Flowchart)	45
4.4	Gantt Chart	52
5.	**Implementation of the Proposed System**	53
5.1	Methodology employed for development	53
5.2	Algorithms and flowcharts for the respective modules developed	53
5.3	Datasets source and utilization	59
6.	**Testing of the Proposed System**	60
6.1	Introduction to testing	60
6.2	Types of tests Considered	60
6.3	Various Test Case Scenarios Considered	61
6.4	Inference drawn from the test cases	61
7	**Results and Discussions**	63

7.1	Screenshots of User Interface (UI) for the respective module	63
7.2	Performance Evaluation measures	64
7.3	Input Parameters / Features considered	65
7.4	Graphical and statistical output	68
7.5	Comparison with Existing Systems	69
7.6	Inference drawn	59
8.	**Conclusion**	71
8.1	Limitations	71
8.2	Conclusion	71
8.3	Future Scope	71
	References	72
	Appendix	73
1	**Paper I**	73
a	Paper I	73
b	Plagiarism Report of Paper I	79
c	Project Review Sheets	80

List of Figures:

Fig no.	Heading	Page no.
1	Block diagram of the proposed system	43
2	Modular diagram representation of the proposed system	44
3	Data Flow Diagram of the proposed system	45
4	Flowchart of the proposed system	46
5	State Transition Diagram of the proposed system	48
6	Entity-Relationship diagram of the proposed system	49
7	Gantt Chart	52
8	Grayscaling and Contour Detection	54
9	Data obtained from TF-Luna LIDAR sensor.	56
10	Warping pipeline for curve detection	57
11	Warping Results	57
12	Giving Alert Signals when the object is near	63
13	Lane and object detection on a curved road	63
14	Final Results	64
15	Live distance measurements provided by the TF-Luna LIDAR	68

List of Tables:

Table no.	Heading	Page no.
1	YOLO and MobileNetSSD v2 comparison	58

Abstract

In today's dynamic automotive landscape, enhancing road safety and accident prevention is paramount. Advanced Driver Assistance Systems (ADAS) offer cutting-edge solutions, but their adoption is often limited by cost and accessibility. The project introduces an innovative approach to bridge this gap, aiming to augment even the most affordable vehicles on Indian roads with a cost-effective ADAS Level 0 system. By leveraging a combination of LIDAR and camera sensors, the proposed solution provides collision avoidance and smart surround-view capabilities. This system is designed to retrofit existing vehicles, making advanced safety features accessible to a wider audience. The core methodology involves sensor fusion, integrating data from both LIDAR and camera sensors to perceive surrounding objects in terms of their depth and direction. A Raspberry Pi-based solution, equipped with a 2D LIDAR sensor and a camera, processes real-time data, enabling object detection and distance measurement. The system's modular architecture ensures seamless communication between various components, allowing for non-blocking operation. The project acknowledges the unique challenges of Indian road conditions, including diverse vehicle types, traffic signs, and the presence of animals. Training models with Indian-specific data enhances the system's relevance and adaptability to local scenarios. While the current project focus is on object detection using a 2D LIDAR sensor and camera fusion, future iterations can expand the capabilities to address broader safety concerns.The project aims to democratize ADAS technology by providing an affordable, retrofittable solution for vehicles in India and other similar contexts. Enhancing collision avoidance and smart surround-view capabilities, contributes to improved road safety and accident prevention, ultimately saving lives and reducing accidents on the road.

Chapter 1: Introduction

1.1. Introduction of the project

In today's rapidly evolving technological landscape, the demand for robust and precise object detection systems has surged across various industries, including autonomous vehicles, robotics, surveillance, and industrial automation. However, achieving accurate object detection in complex and dynamic environments remains a formidable challenge. Traditional single-sensor solutions often struggle to cope with the intricacies of real-world scenarios, such as adverse weather conditions, low-light environments, and the presence of occlusions

To address these challenges and enhance the capabilities of object detection systems, the integration of multiple sensors has emerged as a promising solution. The concept of multi-sensor fusion, which amalgamates data from diverse sensors, offers a comprehensive approach to perception and recognition in complex environments. By combining the strengths of different sensor modalities, such as cameras, LiDAR, sensors, etc multi-sensor fusion systems can significantly improve the accuracy, reliability, and robustness of object detection.

Sensor Fusion between LiDARs and RADARs is heavily used in autonomous driving and many other robotics applications.
Using a LiDAR is great for Simultaneous Localization And Mapping, and for depth estimation, that is, estimating the exact distance of any object. A RADAR has a specific technology that can measure the velocities of objects.

LiDAR (Light Detection and Ranging)

- The LiDAR uses infrared sensors to determine the distance to an object.
- A LiDAR is excellent at estimating distances, being a Time-Of-Flight sensor.

Camera

- A Camera is excellent for classifying objects and understanding a scene.

- The camera is a well-known sensor that's used to output bounding boxes, lane line positions, traffic light colours, traffic signs, and many other things

1.2. Motivation for the Project

In an era marked by rapid technological advancement and the proliferation of automation and autonomous systems, the need for precise and reliable object detection in complex and dynamic environments has never been more apparent. The motivation behind the project is driven by several compelling factors that underscore the significance and urgency of developing multi-sensor fusion techniques for object detection.

a) Safety and Autonomous Systems

The advent of autonomous vehicles, drones, and robotics has revolutionized various industries, promising increased efficiency and safety. However, these systems heavily rely on object detection to navigate and interact with their surroundings. In scenarios such as self-driving cars, any lapse in object detection could have severe consequences, including accidents and loss of life. Therefore, the motivation for the project is deeply rooted in the need to enhance the safety of autonomous systems by providing users with more robust and accurate object detection capabilities.

b) Complex and Unpredictable Environments

Modern environments are often characterized by their complexity and unpredictability. Whether it's navigating through crowded urban streets, monitoring construction sites, or patrolling borders, real-world situations pose formidable challenges for object detection systems. Adverse weather conditions, poor lighting, and the presence of occlusions can compromise the effectiveness of traditional single-sensor solutions. The project seeks to address these challenges by embracing the diversity of sensors available to us, enabling us to detect objects with higher accuracy and consistency in a multitude of environmental conditions.

c) Real-world Applications

Beyond the real world of autonomous vehicles, the application of multi-sensor fusion for object detection extends to diverse domains. From surveillance systems that

safeguard critical infrastructure to industrial automation processes and disaster management, the ability to accurately detect and track objects is pivotal. The motivation stems from the potential to contribute to these real-world applications, making users more effective, efficient, and reliable.

d) Research Advancements

The field of multi-sensor fusion for object detection is an exciting area of research with the potential to push the boundaries of what is currently achievable. The project is motivated by the desire to explore cutting-edge technologies and methodologies, contributing to the scientific community's knowledge and helping to shape the future of sensor-based object detection systems.

1.3. Problem Definition

Modern autonomous systems, such as self-driving cars and drones, rely on accurate object detection to ensure safe and reliable operation. However, object detection in complex environments with varying lighting conditions, weather, and occlusions remains a challenge. Single sensors, such as cameras or LiDAR, may provide limited information and suffer from individual sensor limitations. Developing a multi-sensor fusion approach that combines data from multiple sensors can enhance object detection accuracy and robustness.

1.4. Existing System

As of the current state, many vehicles and robotics systems lack advanced collision avoidance and object detection capabilities. Traditional vehicles and non-autonomous systems primarily rely on manual control by human operators, and safety depends on their attentiveness and reaction time. These systems are limited in their ability to proactively detect and respond to potential collisions or obstacles in real time.

Existing collision avoidance systems in some modern vehicles, often referred to as Advanced Driver Assistance Systems (ADAS), are generally integrated into higher-end models due to their complex and costly nature. These systems include features like forward-collision warnings (FCW) and autonomous emergency braking

(AEB), designed to enhance safety. However, these technologies are not universally accessible, especially in low-cost or older vehicles commonly found on Indian roads.

Moreover, most existing ADAS solutions are based on a combination of sensors such as LIDAR, RADAR, and cameras. While these sensors provide valuable data, they are often expensive and not readily available for retrofitting into existing vehicles. This situation underscores the need for more affordable and adaptable solutions that can be implemented on a wider scale, especially in regions like India, where various types of vehicles coexist.

The existing landscape also lacks solutions tailored to the unique challenges of Indian road conditions, including diverse types of vehicles, traffic signs, and even the presence of animals on roads. Existing ADAS and object detection systems are often designed for Western environments and may not be optimized for the specific requirements and road scenarios encountered in India.

The proposed project seeks to address these limitations by developing an ultra-cheap ADAS Level 0 solution that can be retrofitted into a wide range of vehicles, with a focus on Indian road conditions. This solution aims to democratize collision avoidance and object detection technology, making it accessible to a broader segment of vehicle owners, and enhancing road safety in the process.

1.5. Lacuna of the existing systems

1. Limited Environmental Adaptability

One of the primary limitations of existing single-sensor systems is their limited adaptability to varying environmental conditions. For instance: Lighting Conditions: Vision-based systems, like cameras, struggle in low-light or high-glare situations, reducing their reliability during nighttime or in adverse weather. Weather Dependencies: Some sensors, such as LiDAR and radar, are susceptible to interference from rain, snow, and fog, affecting their performance.

2. Data Redundancy and Complementarity

Single-sensor systems can suffer from data redundancy and a lack of information complementarity. Each sensor provides a unique perspective, and not utilizing this diversity of data can be a limitation. Lack of Depth Information: Cameras, while excellent at providing visual data, typically lack depth information, making it challenging to estimate the distance of objects accurately.

3. Vulnerability to Sensor Failures

Dependence on a single sensor makes the system vulnerable to sensor failures. If the primary sensor malfunctions or is damaged, it can lead to a complete breakdown in object detection, potentially compromising safety and system functionality.

4. Inadequate Robustness

In high-stress situations or rapidly changing environments, single-sensor systems may lack the robustness required for real-time decision-making. The inability to reliably detect objects under dynamic conditions can limit the scope of applications, especially in critical contexts such as autonomous vehicles.

5. Processing Time and Computational Load

The real-time processing of sensor data is a critical factor, especially in time-sensitive applications. Single-sensor systems can be computationally intensive, and the processing time may not meet the stringent requirements of some applications, causing delays in decision-making.

1.6. Relevance of the Project

Autonomous Vehicle Safety: Autonomous vehicles rely heavily on accurate object detection to make real-time decisions, navigate, and avoid collisions. The ability to improve object detection through multi-sensor fusion is directly relevant to enhancing the safety of autonomous vehicles on the road.

Reducing Accidents: Autonomous vehicles aim to reduce accidents caused by human error. By improving object detection in complex environments, the solution contributes to reducing the likelihood of accidents, making autonomous vehicles safer for both passengers and pedestrians.

Enhancing Environmental Adaptability: Autonomous vehicles operate in a wide range of environments and weather conditions. The project's relevance lies in its ability to enhance the adaptability of autonomous systems to diverse and challenging environmental scenarios.

Enabling Urban Mobility: In urban settings, autonomous vehicles are expected to play a significant role in transportation. Accurate object detection is essential for navigating crowded streets, handling complex traffic scenarios, and ensuring the safety of passengers and pedestrians.

Efficiency and Productivity: Autonomous vehicles have the potential to improve transportation efficiency and productivity. By reducing accidents and increasing the accuracy of object detection, the project can contribute to the seamless operation of autonomous vehicle fleets, which can be especially relevant in logistics and urban mobility.

Technological Advancement for Autonomous Vehicles: The automotive industry is evolving rapidly towards autonomous vehicles. The project's contributions to multi-sensor fusion are directly relevant to advancing the technology needed for the success of autonomous vehicles.

Chapter 2: Literature Survey

A. Brief Overview of Literature Survey

Kyongsu Yi, Heungseok Chae, and Hojoon Lee [1] introduces a novel sensor fusion system for autonomous vehicles, merging Radar's large Field of View (FOV) with LIDAR's great distance accuracy, utilizing a multiple hypothesis tracking (MHT) technique. Xiaoting Yan, Deguang Li, and Bo Wang [2] examine classification systems for Sensor Fusion, emphasizing its critical role in autonomous vehicles and supporting dependable autonomous driving. An IIT B alumnus and KLA-Tencor CSE propose [3] the significance of Sensor Fusion in overcoming individual sensor limitations for autonomous cars, advocating for reasonably priced driver assistance systems. [4] discusses object identification and location regression for autonomous driving, proposing a LIDAR-camera fusion approach using a Siamese network.Ponn, T., Kröger, T. and Diermeyer in [5] conducts a thorough analysis of object detectors in automated driving using meta-information. Machine learning-based detectors' behavior is modeled using random forests, showing high accuracy in representing detection performance. W. Deng and C. Wang [6] propose an instance segmentation-based approach to lane identification, offering significant advancements in the creation of dependable lane-keeping systems. [7] presents an improved method for real-time object tracking and detection in autonomous transportation systems, advocating for multi-sensor fusion and collaboration between tracking and detection algorithms to enhance applicability and reliability in dynamic situations.

B. Related Works

2.1. Research papers Referred

1. **A Geometric Model based 2D LiDAR/Radar Sensor Fusion for Tracking Surrounding Vehicles, Hojoon Lee, Heungseok Chae, Kyongsu Yi [1]**

a. Abstract of research paper:

This paper presents a novel and efficient sensor fusion system for tracking multiple moving vehicles over Light Detection and Ranging (LiDAR) and Radar in autonomous vehicles. What is important in the sensor fusion using LiDAR and Radar is how well they utilize the characteristics of each sensor. The proposed fusion system improves the estimating accuracy and maximum perception distance for the target vehicles by utilizing LiDAR which has high distance accuracy and Radar which has wide Field of View (FOV) and observability of relative speed. In addition, multiple hypothesis tracking (MHT) based track management and extended Kalman filtering (EKF) based filtering enable multitarget tracking with less computational complexity than particle filter-based methods. Three measurement models are designed according to the types of measurement assigned to each track, and the tracks were updated optimally according to each model. The proposed fusion system is evaluated through the real vehicle test with RT-range.

b. Inference drawn from the paper

The paper presents a novel and efficient sensor fusion system designed for tracking multiple moving vehicles in autonomous vehicles, utilizing both Light Detection and Ranging (LiDAR) and Radar sensors. Recognizing the distinct advantages of each sensor type, the system strategically harnesses the high distance accuracy of LiDAR and the wide Field of View (FOV) and relative speed observability of Radar. This fusion approach not only enhances the estimation accuracy of the positions and motion of target vehicles but also extends the maximum perception distance, allowing autonomous vehicles to detect and track other vehicles at greater distances. The paper adopts a multiple hypothesis tracking (MHT) approach for track management and employs Extended Kalman Filtering (EKF) for more efficient filtering, surpassing particle filter-based methods in computational complexity. Additionally, the system employs three measurement models tailored to each track's assigned measurements, ensuring optimal track updates. Real vehicle testing further underscores the practical applicability and performance of this fusion system,

underlining its potential significance in the development of autonomous vehicle technology.

2. An End-to-End Lane Detection Model with Attention and Residual Block Bo Wang, Xiaoting Yan and Deguang LiSchool of Information Technology, Luoyang Normal University, Luoyang [2]

a. Abstract of the research paper

In autonomous vehicles, Sensor Fusion combines data from multiple sensors, enhancing reliability, redundancy, and safety. It ensures more accurate detection and understanding of the environment. A simple example involves fusing data from a LiDAR and a Camera to improve detection of pedestrians.

There are three main classification approaches for Sensor Fusion:

I - Sensor Fusion by Abstraction Level:

Low-Level Fusion: Combines raw data from multiple sensors, like point clouds from LiDARs and pixels from cameras. This method has potential but was challenging due to the vast data volume.

Mid-Level Fusion: Fuses independently detected objects. A common technique is using a Kalman Filter for combining results from sensors like cameras and radars.

High-Level Fusion: Combines object detections and their trajectories, incorporating predictions and tracking. This approach is a higher level of fusion but can lose information if tracking is inaccurate.

II - Sensor Fusion by Centralization Level:

Centralized Fusion: A central unit handles fusion, typically for low-level data. Each sensor sends its data to the central unit for processing and fusion.

Decentralized Fusion: Sensors process and fuse data locally, forwarding it to the next sensor or unit. This approach can be seen in a distributed sensor network.

Distributed Fusion: Each sensor processes and sends data to the next unit for fusion. This can happen in late fusion scenarios.

III - Sensor Fusion by Competition Level:

Competitive Fusion: Sensors with the same purpose contribute redundant data. For example, combining data from RADAR and LiDAR for pedestrian detection. Redundancy enhances reliability.

Complementary Fusion: Different sensors observing different scenes combine to provide insights not achievable by a single sensor. Examples include creating panoramas from multiple cameras.

Coordinated Fusion: Multiple sensors focus on the same object, creating new scenes. This approach is used in tasks like 3D reconstruction using 2D sensors.

These classification methods provide insight into when, where, and what type of fusion is applied in sensor data processing. Sensor Fusion is critical in autonomous systems for accurate perception and decision-making.

b. Inference of the research paper

The paper categorizes Sensor Fusion into three key classification approaches. "Sensor Fusion by Abstraction Level" discusses the levels at which data fusion occurs. Low-level fusion amalgamates raw data from multiple sensors, promising potential but posing challenges due to data volume. Mid-level fusion focuses on independently detected objects, often employing techniques like the Kalman Filter to combine results from various sensors. High-level fusion advances the concept by considering object trajectories and predictions but risks information loss if tracking is imprecise.

"Sensor Fusion by Centralization Level" classifies fusion based on data processing distribution. Centralized Fusion centers around a central unit handling data fusion, often for low-level data, with each sensor forwarding its data. Decentralized Fusion enables sensors to process and fuse data locally before forwarding to the next sensor or unit. Distributed Fusion represents a similar approach but with each sensor processing and sending data to the next unit.

"Sensor Fusion by Competition Level" explores how sensors contribute data. Competitive Fusion involves similar sensors contributing redundant data, enhancing reliability. Complementary Fusion combines different sensors to provide insights unattainable by a single sensor, such as creating panoramas from multiple cameras. Coordinated Fusion sees multiple sensors focusing on a single object, crucial in tasks like 3D reconstruction using 2D sensors. Overall, the paper underscores the critical role of Sensor Fusion in autonomous vehicles, providing them with an enhanced understanding of their environment and supporting safer and more reliable autonomous driving systems

3. **ADAS Car using Raspberry Pi AI / ML R&D Specialist, KLA-Tencor CSE, IIT B Alumnus | Ex-Vice President — AI & DS, IIA | Ex-TSMC | Reached interview twice for IAS | AI Consultant & Mentor[3]**

   a. Abstract of the research

   This study explores the feasibility of augmenting these functions on older, budget-conscious vehicles.The key focus of the paper is Sensor Fusion, where data from various sensors, including LiDAR, RADAR, and cameras, is skillfully integrated to harness the unique capabilities of each. By combining these data sources, the system compensates for the limitations of individual sensors, facilitating a more comprehensive perception of the vehicle's surroundings.

   The fusion process involves complex tasks, such as the projection of 3D LiDAR point clouds onto 2D images, object detection, and precise matching between LiDAR and camera data. Calibration plays a critical role in aligning coordinate systems, involving rotation, translation, and intrinsic calibration.The research is noteworthy for its practical implementation, which includes both 3D LiDAR-Camera sensor fusion and a more budget-friendly 2D LiDAR-Camera fusion, designed for affordable ADAS implementation. The paper also details the hardware setup and modular architecture, emphasizing the need for cost-effective

ADAS solutions to improve road safety in regions where affordability is a significant concern.

b. Inference drawn from the research paper

The paper underscores the importance of Sensor Fusion, which combines data from LiDAR, RADAR, and cameras to leverage the unique strengths of each sensor. This fusion approach compensates for the individual limitations of these sensors, enhancing the overall perception of a vehicle's surroundings.The modular architecture described in the study facilitates non-blocking system flow, emphasizing practicality. The paper addresses the need for affordable ADAS in regions where budget constraints are a significant factor in enhancing road safety.

4. Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J.,& Whittaker, W. Stanley: The robot that won the DARPA Grand Challenge. Journal of Field Robotics[4]

a. Abstract of the Research Paper

This article describes the robot Stanley, which won the 2005 DARPA Grand Challenge. Stanley was developed for high-speed desert driving without manual intervention. The robot's software system relied predominantly on state-of-the-art artificial intelligence technologies, such as machine learning and probabilistic reasoning.This article describes the major components of this architecture, and discusses the results of the Grand Challenge race

b. Inference Drawn from the Research Paper

This article provides a comprehensive survey of the winning robot of the DARPA Grand Challenge.Stanley, developed by the Stanford Racing Team in collaboration with its primary supporters, relied on a software pipeline for processing sensor data and determining suitable steering, throttle, brake and gear shifting commands.

From a broad perspective, Stanley's software mirrors common methodology in autonomous vehicle control. However, many of the individual modules relied on state-of-the-art Artificial Intelligence

techniques. The pervasive use of machine learning, both ahead and during the race, made Stanley

robust and precise. We believe that those techniques, along with the extensive testing that took place, contributed significantly to Stanley's success in this race.

5. **Real time object detection using LiDAR and camera fusion for autonomous driving Liu, H., Wu, C. & Wang, H. Real time object detection using LiDAR and camera fusion for autonomous driving.[5]**

   a. Abstract of the research Paper

      Autonomous driving has been widely applied in commercial and industrial applications, along with the upgrade of environmental awareness systems. Tasks such as path planning, trajectory tracking, and obstacle avoidance are strongly dependent on the ability to perform real-time object detection and position regression. Among the most commonly used sensors, camera provides dense semantic information but lacks accurate distance information to the target, while LiDAR provides accurate depth information but with sparse resolution. In this paper, a LiDAR-camera-based fusion algorithm is proposed to improve the above-mentioned trade-off problems by constructing a Siamese network for object detection. Raw point clouds are converted to camera planes to obtain a 2D depth image. By designing a cross feature fusion block to connect the depth and RGB processing branches, the feature-layer fusion strategy is applied to integrate multi-modality data. The proposed fusion algorithm is evaluated on the KITTI dataset. Experimental results demonstrate that our algorithm has superior performance and real-time efficiency. Remarkably, it outperforms other state-of-the-art algorithms at the most important moderate level and achieves excellent performance at the easy and hard levels.

   b. Inference drawn from the paper

      In this paper, the authors address the challenge of object detection and position regression for autonomous driving. They propose a

LiDAR-camera fusion algorithm that leverages the strengths of both sensors. By using a Siamese network, raw LiDAR data is transformed into 2D depth images and fused with RGB camera data. The results on the KITTI dataset show that this approach outperforms state-of-the-art algorithms, particularly in moderate-level scenarios, while maintaining real-time efficiency. This fusion technique offers promising improvements for autonomous driving systems, enhancing their ability to handle various driving conditions.

6. Sensor Fusion-Based Approach to Eliminating Moving Objects for SLAM in Dynamic Environments Dang, X.; Rong, Z.; Liang, X [6]

a. Abstract of the Research paper

Accurate localization and reliable mapping is essential for autonomous navigation of robots.Simultaneous Localization and Mapping (SLAM) has attracted widespread attention in recent decades. Based on vision or LiDAR sensors, great efforts have been devoted to achieving real-time SLAM that can support a robot's state estimation. However, most of the mature SLAM methods generally work under the assumption that the environment is static, while in dynamic environments they will yield degenerate performance or even fail. In this paper, first we quantitatively evaluate the performance of the state-of-the-art LiDAR-based SLAMs taking into account different patterns of moving objects in the environment. Through semi-physical simulation, we observed that the shape, size, and distribution of moving objects all can impact the performance of SLAM significantly, and obtained instructive investigation results by quantitative comparison between LOAM and LeGO-LOAM. Secondly, based on the above investigation, a novel approach named EMO to eliminate the moving objects for SLAM fusing LiDAR and mmW-radar is proposed, towards improving the accuracy and robustness of state estimation. The method fully uses the advantages of different characteristics of two sensors to realize the fusion of sensor information with two different resolutions. The moving objects can be efficiently

detected based on Doppler effect by radar, accurately segmented and localized by LiDAR, then filtered out from the point clouds through data association and accurately synchronized in time and space. Finally, the point clouds representing the static environment are used as the input of SLAM. The proposed approach is evaluated through experiments using both semi-physical simulation and real-world datasets. The results demonstrate the effectiveness of the method at improving SLAM performance in accuracy (decrease by 30% at least in absolute position error) and robustness in dynamic environments.

b. Inference drawn from the research paper

The research paper focuses on advancing Simultaneous Localization and Mapping (SLAM) for autonomous robots, particularly in dynamic environments where traditional SLAM methods struggle. It quantitatively evaluates state-of-the-art LiDAR-based SLAM systems, demonstrating their limitations in the presence of moving objects. Leveraging this understanding, the paper proposes a novel approach called EMO, which combines LiDAR and mmW-radar to eliminate moving objects from SLAM data, improving accuracy and robustness in state estimation. EMO detects, segments, and filters out moving objects based on Doppler radar information and LiDAR data, synchronizing the static environment data for SLAM. Experimental results, including semi-physical simulations and real-world datasets, confirm the effectiveness of EMO, showcasing a significant reduction in absolute position error by at least 30%. This research contributes valuable insights and a practical solution for enhancing SLAM performance in dynamic scenarios, a crucial step in advancing autonomous navigation.

7. S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks" [7]

a. Abstract of the research paper

State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet and Fast R-CNN have reduced the running time of these detection networks,

exposing region proposal computation as a bottleneck. In this work, we introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully-convolutional network that simultaneously predicts object bounds and objectness scores at each position. RPNs are trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. With a simple alternating optimization, RPN and Fast R-CNN can be trained to share convolutional features. For the very deep VGG-16 model, our detection system has a frame rate of 5fps (including all steps) on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007 (73.2% mAP) and 2012 (70.4% mAP) using 300 proposals per image.

b. Inference drawn from the research paper

In this paper, the authors introduce a groundbreaking approach to object detection by presenting the Region Proposal Network (RPN). Unlike previous methods that rely on separate region proposal algorithms, the RPN shares convolutional features with the detection network, significantly reducing computational overhead. The RPN is a fully-convolutional network capable of simultaneously predicting object boundaries and objectness scores at various positions. Through end-to-end training, the RPN generates high-quality region proposals that can be seamlessly integrated with the Fast R-CNN detection network. This approach achieves remarkable results, with a very deep VGG-16 model running at an impressive 5 frames per second (fps) on a GPU, while still maintaining state-of-the-art object detection accuracy on PASCAL VOC 2007 and 2012 datasets. It demonstrates the potential for highly efficient and accurate object detection systems in real-world applications.

8. **Wang, C., & Deng, W. (2019). Towards End-to-End Lane Detection: an Instance Segmentation Approach. [8]**

   a. Abstract of the Research Paper

      This paper discusses the increasing integration of driver-assist features, particularly automatic lane-keeping, in modern vehicles. Lane-keeping is essential for positioning a vehicle within road lanes, a fundamental component for both driver assistance and autonomous driving systems. Traditional lane detection methods rely on handcrafted features and heuristics, making them computationally expensive and less adaptable to varying road conditions. More recent techniques employ deep learning models for pixel-wise lane segmentation, but these are typically limited to fixed lane configurations and struggle with lane changes.

      To overcome these limitations, this paper introduces an innovative approach, casting lane detection as an instance segmentation problem. It allows for the detection of multiple lanes and adaptation to lane changes. The proposed method employs learned perspective transformations, conditioned on the image, to ensure robust lane fitting in the presence of changing road planes. The outcome is a fast lane detection algorithm running at 50 frames per second, capable of handling variable lane configurations and lane changes, with competitive results validated on the tuSimple dataset.

   b. Inference Drawn From the Research Paper

      This paper addresses the need for robust and adaptive lane detection in vehicles, essential for both driver assistance and autonomous driving systems. Traditional methods are limited by handcrafted features and heuristics, while recent deep learning approaches can't adapt to changing road conditions or handle variable lane configurations.The paper's innovative contribution is to treat lane detection as an instance segmentation problem, allowing for the detection of multiple lanes and adaptation to lane changes. By incorporating learned perspective transformations conditioned on the image, the proposed approach achieves robust lane fitting even in the presence of changing road planes. The result is a high-speed lane detection algorithm that can accommodate a variable

number of lanes and cope with lane changes, demonstrating competitive performance on the tuSimple dataset. This work provides a significant step forward in the development of reliable and adaptable lane-keeping systems for modern vehicles.

9. A Review on Autonomous Vehicles: Progress, Methods and Challenges Parekh, D.; Poddar, N.; Rajpurkar, A.; Chahal, M.; Kumar, N.; Joshi, G.P.; Cho, W. A Review on Autonomous Vehicles:Progress, Methods and Challenges [9]

a. Abstract of the research paper

Vehicular technology has recently gained increasing popularity, and autonomous driving is a hot topic. To achieve safe and reliable intelligent transportation systems, accurate positioning technologies need to be built to factor in the different types of uncertainties such as pedestrian behavior, random objects, and types of roads and their settings. In this work, we look into the other domains and technologies required to build an autonomous vehicle and conduct a relevant literature analysis. In this work, we look into the current state of research and development in environment detection, pedestrian detection, path planning, motion control, and vehicle cybersecurity for autonomous vehicles. We aim to study the different proposed technologies and compare their approaches. For a car to become fully autonomous, these technologies need to be accurate enough to gain public trust and show immense accuracy in their approach to solving these problems. Public trust and perception of auto vehicles are also explored in this paper. By discussing the opportunities as well as the obstacles of autonomous driving technology, we aim to shed light on future possibilities.

b. Inference drawn from the research paper

In this survey, we have outlined, highlighted, and investigated the technological advancements in autonomous driving. We also examined the recent advances and case studies on autonomous driving. While conducting this review, we discovered and discussed all the future research challenges for further studies in this domain that were mostly

related to implementing the proposed technologies in the real world. Certain aspects of safety, path planning algorithms, security, and privacy have also been reviewed in this paper. Non-technical challenges such as consumer trust, governance, and human behavior towards autonomous driving play a major role in bringing AVs and related technologies to widespread use. This review could contribute with necessary insights into the emerging domains in autonomous driving. We can conclude that autonomous vehicles are the way of the future, and vehicles with Level 3 Automation are ready for commercialization.

10. An Efficient Multi-sensor Fusion Approach for Object Detection in Maritime Environments M.-H. Haghbayan [10]

a. Abstract of the Research Paper

Robust real-time object detection and tracking are challenging problems in autonomous transportation systems due to operation of algorithms in inherently uncertain and dynamic environments and rapid movement of objects. Therefore, tracking and detection algorithms must cooperate with each other to achieve smooth tracking of detected objects that later can be used by the navigation system. In this paper, we first present an efficient multi-sensor fusion approach based on the probabilistic data association method in order to achieve accurate object detection and tracking results. The proposed approach fuses the detection results obtained independently from four main sensors: radar, LiDAR, RGB camera and infrared camera. It generates object region proposals based on the fused detection result. Then, a Convolutional Neural Network (CNN) approach is used to identify the object categories within these regions. The CNN is trained on a real dataset from different ferry driving scenarios. The experimental results of tracking and classification on real datasets show that the proposed approach provides reliable object detection and classification results in maritime environments.

b. Inference drawn from the research paper

This research paper introduces an approach for enhancing real-time object detection and tracking in autonomous transportation systems. The paper

emphasizes the need for cooperation between detection and tracking algorithms in dynamic environments. It proposes a multi-sensor fusion method that combines data from radar, LiDAR, RGB camera, and infrared camera to improve accuracy. The use of probabilistic data association aids in managing uncertainty in sensor data. Object region proposals are generated based on the fusion results, and a Convolutional Neural Network (CNN) is employed for object classification. Real-world experiments in maritime scenarios confirm the approach's reliability and applicability. In essence, the paper offers a solution that combines sensors and deep learning to address the challenges of object detection and tracking in uncertain, dynamic environments for autonomous transportation systems.

2.2 Patent Search

1. **Vehicle imaging processing system and method having obstructed image detection [13].**

   An imaging system for a vehicle is provided. The system includes an image sensor generating a plurality of image frames, and a controller coupled to the image sensor. The controller is programmed to detect an obstructed image frame from the plurality of image frames. The obstruction corresponds to a windshield wiper of the vehicle. In response to detecting an obstructed image, the controller optimizes the obstructed image for display or use by a vehicle system. The controller detects an obstructed image by at least one of object sensing confirmation, edge detection, negative shift detection or headlight illumination detection.

2. **Lane tracking system - GM GLOBAL TECHNOLOGY LLC [14]**

   A lane tracking system for tracking the position of a vehicle within a lane includes a camera configured to provide a video feed representative of a field of view and a video processor configured to receive the video feed from the camera and to generate latent video-based position data indicative of the position of the vehicle within the lane. The system further includes a vehicle

motion sensor configured to generate vehicle motion data indicative of the motion of the vehicle, and a lane tracking processor. The lane tracking processor is configured to receive the video-based position data, updated at a first frequency; receive the sensed vehicle motion data, updated at a second frequency; estimate the position of the vehicle within the lane from the sensed vehicle motion data; and fuse the video-based position data with the estimate of the vehicle position within the lane using a Kalman filter.

3. **Fusion of obstacle detection using radar and camera - GM GLOBAL TECHNOLOGY [15]**

   A vehicle obstacle detection system includes an imaging system for capturing objects in a field of view and a radar device for sensing objects in a substantially same field of view. The substantially same field of view is partitioned into an occupancy grid having a plurality of observation cells. A fusion module receives radar data from the radar device and imaging data from the imaging system. The fusion module projects the occupancy grid and associated radar data onto the captured image. The fusion module extracts features from each corresponding cell using sensor data from the radar device and imaging data from the imaging system. A primary classifier determines whether an extracted feature extracted from a respective observation cell is an obstacle.

2.3 Comparison with the existing system

As of the current state, many vehicles and robotics systems lack advanced collision avoidance and object detection capabilities. Traditional vehicles and non-autonomous systems primarily rely on manual control by human operators, and safety depends on their attentiveness and reaction time. These systems are limited in their ability to proactively detect and respond to potential collisions or obstacles in real time.

Existing collision avoidance systems in some modern vehicles, often referred to as Advanced Driver Assistance Systems (ADAS), are generally integrated into higher-end models due to their complex and costly nature. These systems include

features like forward-collision warnings (FCW) and autonomous emergency braking (AEB), designed to enhance safety. However, these technologies are not universally accessible, especially in low-cost or older vehicles commonly found on Indian roads.

Moreover, most existing ADAS solutions are based on a combination of sensors such as LIDAR, RADAR, and cameras. While these sensors provide valuable data, they are often expensive and not readily available for retrofitting into existing vehicles. This situation underscores the need for more affordable and adaptable solutions that can be implemented on a wider scale, especially in regions like India, where various types of vehicles coexist.

The existing landscape also lacks solutions tailored to the unique challenges of Indian road conditions, including diverse types of vehicles, traffic signs, and even the presence of animals on roads. Existing ADAS and object detection systems are often designed for Western environments and may not be optimized for the specific requirements and road scenarios encountered in India.

The proposed project seeks to address these limitations by developing an ultra-cheap ADAS Level 0 solution that can be retrofitted into a wide range of vehicles, with a focus on Indian road conditions. This solution aims to democratize collision avoidance and object detection technology, making it accessible to a broader segment of vehicle owners, and enhancing road safety in the process.

Chapter 3: Requirement Gathering for the Proposed System

3.1. Introduction to requirement gathering

Functional requirements define what the system should do, encompassing specific actions,behavior, and functionalities it must perform to meet the users' needs. On the other hand,non-functional requirements specify how the system should perform, including aspects such as performance, usability, reliability, security, and scalability. Both types of requirements are essential to gather because they collectively define the scope, capabilities, and quality attributes of the system. Functional requirements ensure that the system meets the users' functional needs, while non-functional requirements ensure that the system meets their quality expectations and performance standards. By gathering both types of requirements comprehensively, the project team can ensure that the resulting system not only performs the desired functions but also meets the users' expectations in terms of reliability, usability, and other critical aspects, thereby enhancing user satisfaction and system effectiveness.

3.2. Functional Requirements

1. Sensor System Integration
    - The system shall integrate a LIDAR sensor, Raspberry Pi Camera, and LED SHIM for data collection and object detection
    - The system shall ensure synchronization between LIDAR and camera data to facilitate sensor fusion.
2. Data Collection and Processing
    - The system shall capture real-time data from the LIDAR sensor, including distance and direction information.
    - The system shall capture real-time images from the Raspberry Pi Camera.
    - Data from the LIDAR and camera shall be calibrated for alignment and

accuracy.

- The system shall preprocess the data to remove noise and outliers.

3. Object Detection
    - The system shall employ object detection algorithms, such as SSD MobileNet, to identify and classify objects within the camera images.
    - Detected objects shall be categorized as vehicles, pedestrians, traffic signs, and other relevant entities.
4. Sensor Fusion
    - The system shall perform sensor fusion to integrate LIDAR data (distance and direction) with camera data (object detection).
    - The fusion process shall provide depth and direction information for objects detected in the camera images.
5. Distance Estimation
    - The system shall estimate the distance from the vehicle to detected objects using fused sensor data.
    - The distance estimation process shall consider the relative orientation of the LIDAR and camera.
6. Data Output
    - The system shall provide real-time visual alerts to the driver.
    - Data on detected objects and distance estimations shall be presented to the driver on a display or interface.
7. System Integration
    - The system shall be designed for potential integration with the vehicle's ability to perform autonomous emergency braking and collision avoidance.

3.3. Non Functional Requirements

1. Performance
    - Real-time Processing: The system shall process sensor data and provide distance estimations and object detection results in real-time.
    - Scalability: The system shall be scalable to accommodate potential future sensor upgrades and additional features.

2. Accuracy
    - Sensor Fusion Accuracy: The sensor fusion process shall provide accurate depth and direction estimations .
    - Object Detection Accuracy: Object detection shall achieve a minimum accuracy rate of 94% for vehicles, pedestrians, and traffic signs.
3. Reliability
    - System Reliability: The system shall have a high level of reliability.
    - Redundancy: The system shall incorporate redundancy measures to ensure uninterrupted operation in case of sensor or component failures.
4. Security
    - Data Security: The system shall ensure the security of user data and sensor data through encryption and access controls.
    - Privacy: The system shall respect user privacy and comply with data protection regulations.
5. User Interface
    - User-Friendly Interface: The user interface shall be intuitive and user-friendly, with clear and concise visual displays.
    - Accessibility: The interface shall be accessible to all users, including those with disabilities.
6. Environmental Conditions
    - Environmental Adaptability: The system shall function effectively under varying weather conditions, including rain, fog, and bright sunlight.
7. Compliance
    - Regulatory Compliance: The system shall comply with relevant automotive safety standards and regulations in India.
    - Data Privacy Regulations: The system shall adhere to data privacy and protection regulations applicable in India.
8. Maintainability
    - Ease of Maintenance: The system shall be designed for ease of maintenance, including software updates and sensor calibration.

- Documentation: Comprehensive documentation shall be provided to assist with system maintenance and troubleshooting.

9. Performance Testing
    - Performance Testing: The system shall undergo rigorous performance testing under various road conditions to ensure its reliability and accuracy.
    - Load Testing: The system shall be tested under load conditions to verify its performance under high data volumes.

3.4. Hardware, Software , Technology and tools utilized

3.4.1 Hardware Requirements:

1. Sensor Integration
    - 2D LIDAR Sensor: The system requires a 2D LIDAR sensor, such as the TF-LUNA Micro LiDAR Distance Sensor for IoT ITS (8M)
    - Raspberry Pi: A Raspberry Pi 4 (4GB RAM) single-board computer for data processing and integration.
    - Raspberry Pi Camera: A compatible Raspberry Pi Camera module, Camera Sensor - Raspberry PI Camera Module 5MP 1080p OV5647 Sensor HD Video Webcam Supports Night Vision for Raspberry Pi Model B/B+ A+ RPi 3/2/1/zero/zero W/Pi 4 with FFC/FPC Cable
2. Power Supply
    - Battery: A 5V 3A 10,000 mAh battery for powering the system, ensuring an uninterrupted power supply(power bank).
3. Mounting and Enclosure
    - Mounting Hardware: Mounting hardware for securely attaching the LIDAR sensor and camera to the system.
    - Protective Enclosure: A protective enclosure for housing the Raspberry Pi and associated components to shield them from environmental factors.
4. Connectivity

- Type C-USB Cable: A Type C-USB cable to connect the power supply and to the Raspberry Pi.
- FFC cable to connect the camera to Raspberry pi.
- JST Jumper 2 cable to connect the Lidar sensor to Raspberry pi.

3.4.2 Software Requirements:

1. Operating System
    - Raspberry Pi OS: The system shall run on a suitable Raspberry Pi operating system, such as Raspberry Pi OS (formerly Raspbian).
    - RealVNC remote viewer
2. Sensor Interface
    - LIDAR Interface: Software for interfacing with and receiving data from the 2D LIDAR sensor via a USB connection.
    - Camera Interface: Software for capturing and processing images from the Raspberry Pi Camera.
3. Data Processing and Sensor Fusion
    - Data Processing: Software for processing and preprocessing sensor data, including noise removal and calibration.
    - Sensor Fusion: Algorithms and software for fusing data from the LIDAR sensor and camera to provide depth and direction information.
4. Object Detection
    - Object Detection Algorithm: The system shall employ object detection algorithms, such as SSD MobileNet, for identifying and classifying objects within the camera images.
5. User Interface
    - User Interface Software: Software for creating a user-friendly interface for the driver, displaying object detection results, and visual alerts.

3.4.3 Techniques utilized till date for the proposed system

1. Sensor Fusion: Previous systems have implemented sensor fusion techniques to combine data from multiple sensors, such as LIDAR, cameras, and radar, to improve object detection and depth estimation.
2. Object Detection Algorithms: Existing systems have employed object detection algorithms like YOLO (You Only Look Once) or SSD (Single Shot Multibox Detector) for real-time detection and classification of objects in camera images.
3. Depth Estimation: Techniques for estimating object distance from sensors, including stereo vision, structure from motion (SfM), and point cloud processing, have been used to provide depth information.
4. Machine Learning: Machine learning models, such as neural networks and deep learning, have been utilized for training object detection and classification algorithms.
5. Calibration and Alignment: Methods for sensor calibration and alignment have been crucial for ensuring accurate sensor fusion and depth estimation.
6. Real-time Processing: Systems have focused on optimizing real-time data processing to provide timely alerts and assistance to drivers.

3.4.4 Tools utilized till date for the proposed system

1. ROS (Robot Operating System): ROS has been used as a middleware platform for managing sensors, communication, and data processing in autonomous vehicle systems.
2. OpenCV: OpenCV, an open-source computer vision library, has been a popular tool for image and video processing, including object detection and camera calibration.
3. TensorFlow and PyTorch: Deep learning frameworks like TensorFlow and PyTorch have been used for training and deploying machine learning models for object detection and classification.
4. LIDAR SDKs: Software Development Kits (SDKs) for LIDAR sensors, provided by manufacturers like Velodyne and RPLIDAR, are used to interface with and process LIDAR data.

5. Object Detection Frameworks: Frameworks such as YOLO (You Only Look Once) and SSD (Single Shot Multibox Detector) have been employed for real-time object detection.
6. MATLAB: MATLAB, with its computer vision and image processing toolbox, has been utilized for various aspects of sensor data processing.
7. Python: The Python programming language is commonly used for software development, data analysis, and scripting in ADAS systems.

3.5. Constraints

1. Sensor Limitations
    - 2D LIDAR Range: The use of a 2D LIDAR sensor limits the system's ability to capture three-dimensional data, which may affect the accuracy of depth estimations.
    - Field of View (FOV): The 2D LIDAR has a restricted FOV, and objects outside this FOV may not be detected.
2. Limited Sensor Fusion
    - Reduced Sensor Fusion: The fusion of sensor data from the 2D LIDAR and camera may provide less comprehensive object depth information compared to a 3D LIDAR system.
    - Limitation on Complex Scenarios: The system's ability to handle complex scenarios, such as those involving occlusions or objects at different heights, may be constrained.
3. Training Data Availability
    - Data Availability: The availability of comprehensive training data, especially for Indian-specific conditions, can be limited, which may affect the system's ability to recognize diverse objects and scenarios.
    - Indian Traffic Conditions: The system may face constraints in adapting to the unique traffic conditions and objects commonly found on Indian roads.
4. Hardware Limitations

- Limited Processing Power: The use of Raspberry Pi and other low-cost hardware components may constrain the system's processing capabilities, potentially affecting real-time performance.
- Power Supply: The reliance on battery power for the system may impose constraints on operating duration and may require frequent recharging or replacement of batteries.

Chapter 4: Proposed Design

4.1. Block Diagram of the System

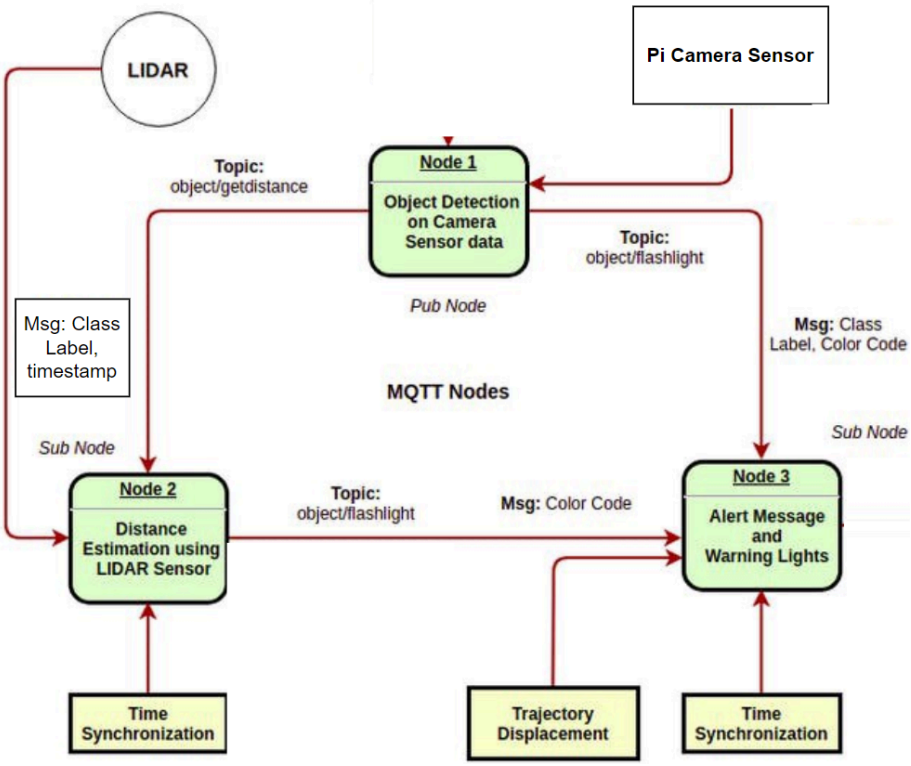


Fig 1: Block diagram of the proposed system

Explanation for the block diagram:

The time synchronization module takes care of the "data relevance factor" for Sensor Fusion. For ADAS, the location of detected objects by 'Node 1' may change, as the objects move. Thus, the distance estimate to the bounding box could go wrong after 2–3 seconds (while the message can remain in the MQTT queue). In order to synchronize, current time = 60*minutes + seconds is appended to the message (to ignore lagged messages).

The model output is sent from Node 1 to Node 2, where the LiDAR-Cam sensor fusion happens, further pushing a message to Node 3. For the system to function, the 3 MQTT nodes should work in tandem, orchestrated by MQTT messages, published, and subscribed on respective topics.

4.2. Modular Design of The System

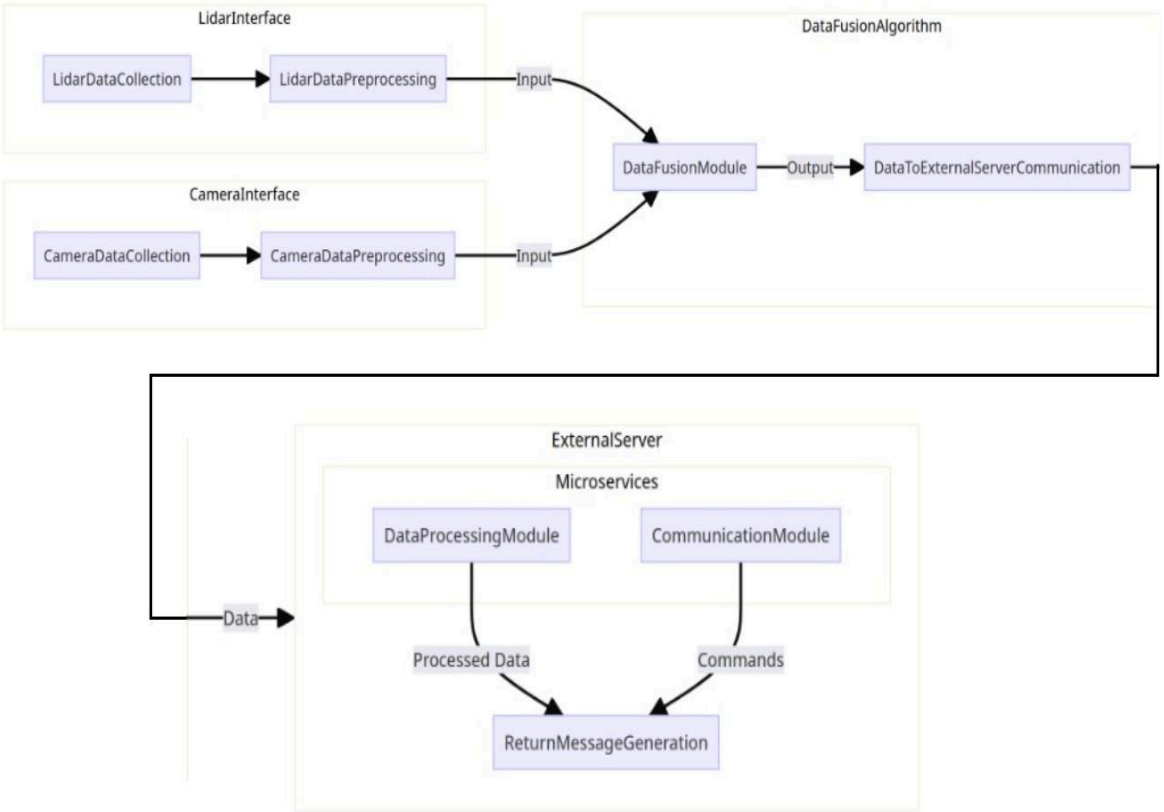


Fig2**:** Modular diagram representation of the proposed system

Explanation for the modular block diagram:

The project augments road safety through an affordable Advanced Driver Assistance System (ADAS) designed for Indian road conditions. By fusing data from a 2D LIDAR sensor and a camera, the system detects objects, measures their distance, and triggers real-time alerts if collision risks are identified. The project's modular architecture ensures seamless operation, while custom calibration enhances data accuracy. Trained with Indian-specific data, including traffic signs and vehicles, this innovative solution seeks to democratize ADAS technology, making it retrofittable into existing vehicles and significantly improving road safety in India.

4.3. Detailed Design

4.3.1 Data Flow Diagram

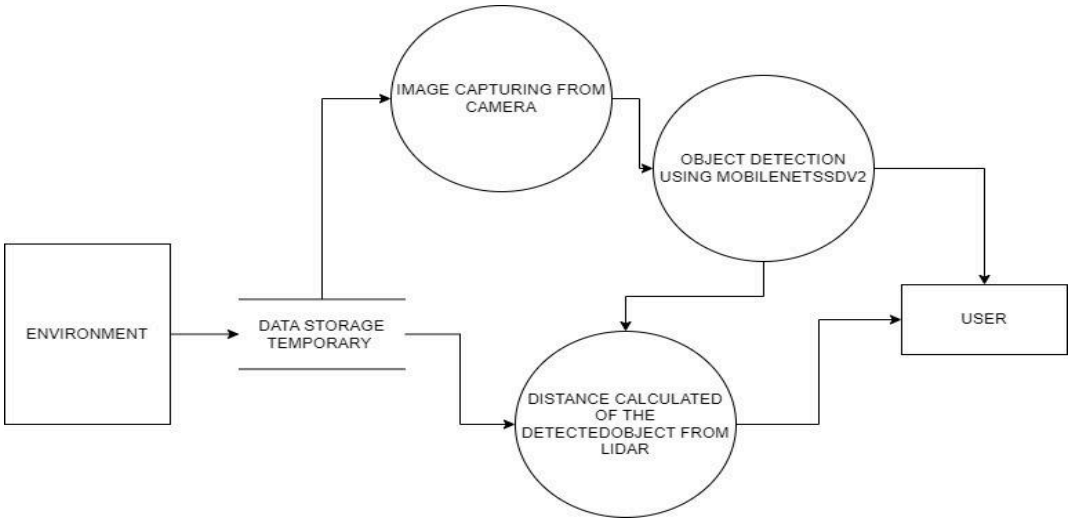


Fig 3: Data Flow Diagram of the proposed system

The DFD outlines the flow of data and processes within the object detection and distance measurement system. It involves the following main components:

User: An external user who interacts with the system.

Raspberry Pi: The hardware platform that hosts the system.

Processes:

- Initialize System: Sets up the system components.
- Capture Image: Captures images with the Raspberry Pi camera.
- Object Detection: Detects objects in captured images.
- LIDAR Measurement: Measures distances to detected objects.
- Display Detected Objects: Shows detected objects to the user
- Display Distance Information: Displays measured distances to the user.
- Evaluate Distance: Checks if objects are at a safe distance.
- Generate Alerts: Issues warnings if objects are too close.

Data Stores:

- Image Data Store: Temporarily holds captured images.
- Detected Object Data Store: Stores information about detected objects.

- Measured Distance Data Store: Contains distance measurements.

Data Flows:

- Capture Image Data: Transfers captured image data for processing.
- Object Detection Data: Conveys information about detected objects.
- Distance Measurement Data: Transfers distance measurements.
- Display Objects Data: Shows detected objects to the user.
- Display Distance Data: Displays measured distances.

4.3.2 Flowchart for the proposed system

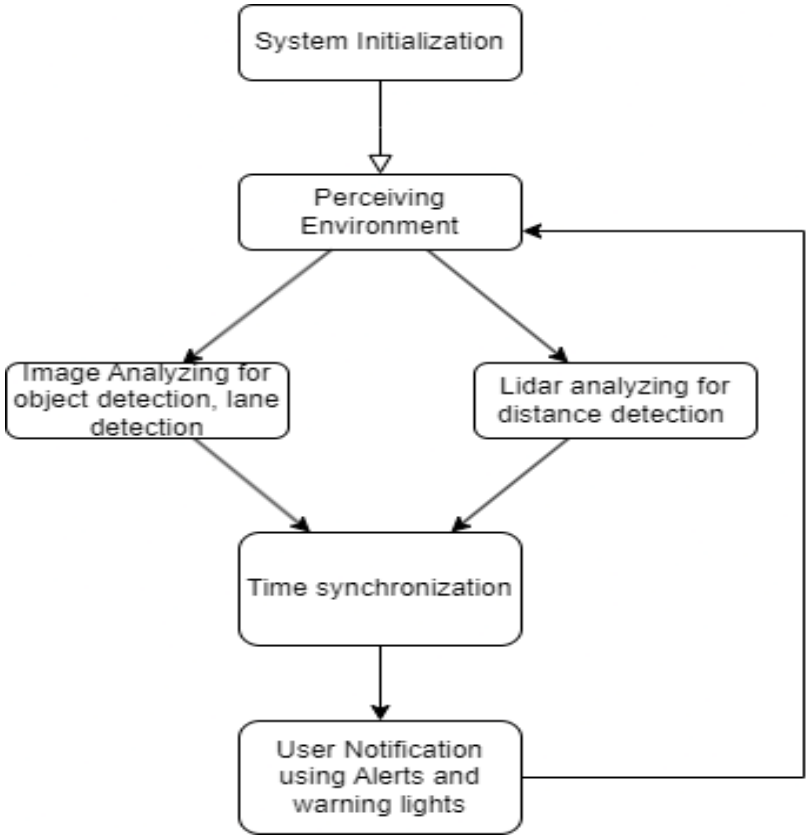


Fig 4: Flowchart of the proposed system

Raspberry Pi acts as the underlying hardware platform for hosting the system. As shown in Figure 4, the system comprises various processes including

initializing system components, capturing images using the Raspberry Pi camera, detecting objects within captured images, and measuring distances to those objects using LIDAR. Additionally, it displays detected objects and their distances to the user, evaluates if detected objects are at a safe distance, and generates alerts if they are too close.

Data is stored in separate data stores: The Image Data Store temporarily holds captured images, the Detected Object Data Store stores information about detected objects, and the Measured Distance Data Store contains distance measurements. Data flows include transferring captured image data for processing, conveying information about detected objects, and transferring distance measurements.

The system operates through different states: Idle, Object Detection, and Distance Measurement. In the Idle state, the system awaits user activation. Upon activation, it transitions to the Object Detection state, which actively identifies and locates objects. If no objects are detected, it returns to the Idle state; otherwise, it moves to the Distance Measurement state to measure distances to the detected objects. If distances are within a safe range, it returns to the Object Detection state. Otherwise, it triggers alert generation due to proximity.

4.3.3 State Transition Diagram/ Activity Diagram

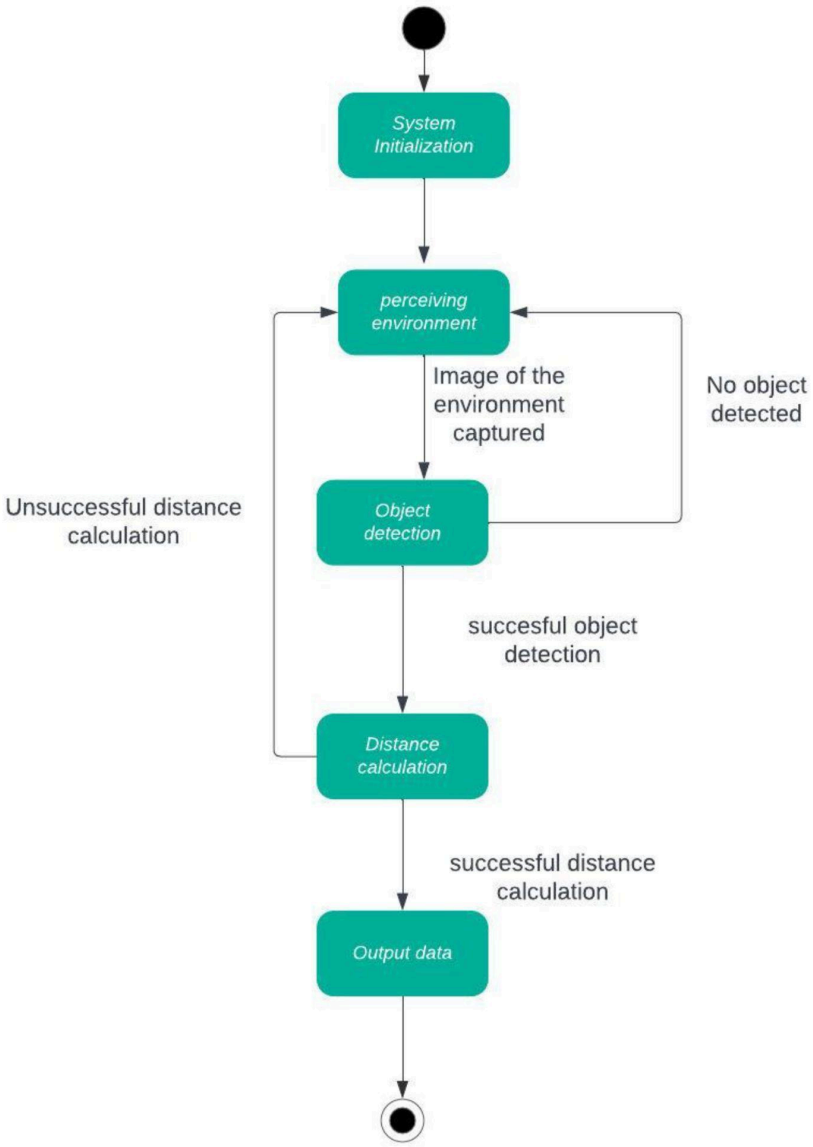


Fig 5: State Transition Diagram of the proposed system

Explanation:

- State 1: Idle

Description: The system is at rest, awaiting user activation.

Transitions:

"On user activation, transition to 'Object Detection':"

User input triggers the system to move to the object detection phase.

- State 2: Object Detection

Description: The system actively identifies and locates objects in the environment.

Transitions:

"If no objects are detected, return to 'Idle':"

If no objects are found, the system goes back to an idle state.

"If objects are detected, transition to 'Distance Measurement':"

Successful detection leads to the measurement of object distances.

- State 3: Distance Measurement

Description: The system measures the distances to detected objects using LIDAR.

Transitions:

"If distances are within a safe range, return to 'Object Detection':"

Safe distances result in a return to the object detection phase.

"If distances are too close, trigger 'Alert Generation':"

Close proximity triggers alert generation.

4.3.4 Entity-Relationship (ER) Diagram

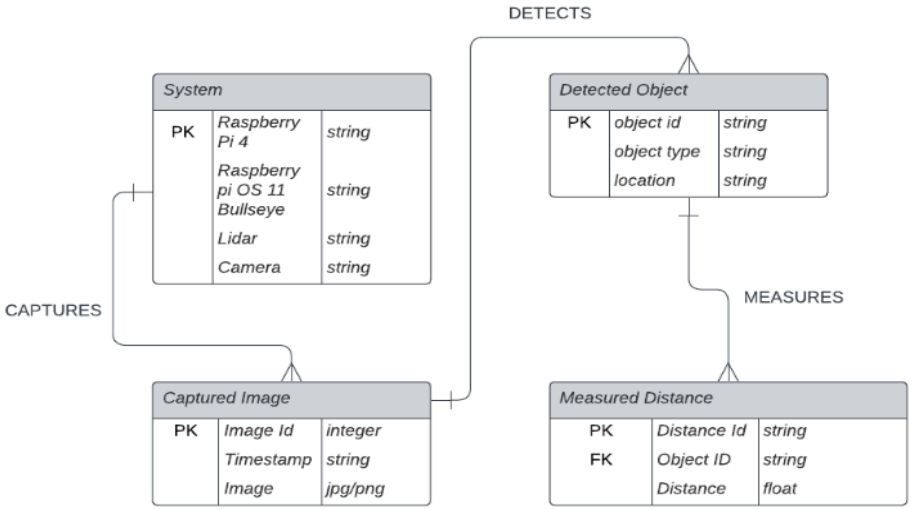


Fig 6: Entity-Relationship diagram of the proposed system

The ER diagram depicts the entities and their relationships within the object detection and distance measurement system. It provides a visual representation of how different components in the system are connected and interact with each other. Here are the main elements:

Entities:

1. System:
    a. Attributes: Raspberry Pi Version (Primary Key): This attribute uniquely identifies each Raspberry Pi version, ensuring a distinct record for each system configuration.
    b. Software Components: This attribute stores a list of software components installed on the Raspberry Pi.
2. Captured Image:
    a. Attributes: Image ID (Primary Key): This attribute provides a unique identifier for each captured image, ensuring each image has a distinct record.
    b. Timestamp: This attribute records the date and time when the image was captured.
3. Detected Object:
    a. Attributes:Object ID (Primary Key): This attribute uniquely identifies each detected object, ensuring a distinct record for each object.
    b. Object Type: This attribute stores the type or category of the detected object.
    c. Location: This attribute represents the coordinates of bounding box information of the detected object.
4. Measured Distance:
    a. Attributes:Distance ID (Primary Key): This attribute provides a unique identifier for each measured distance, ensuring each measurement has a distinct record.
    b. Object ID (Foreign Key): This attribute serves as a reference to the detected object, establishing a link between the measured distance and the associated object.

c. Distance Value: This attribute stores the actual measurement of the distance.

Relationships**:**

1. Captures: This relationship between the "System" entity and the "Captured Image" entity signifies that a system can capture multiple images. It establishes a one-to-many relationship.
2. Detects: This relationship between the "Captured Image" entity and the "Detected Object" entity indicates that a captured image can be used to detect multiple objects. It establishes a one-to-many relationship.
3. Measures: This relationship between the "Detected Object" entity and the "Measured Distance" entity denotes that a detected object can have multiple distance measurements associated with it. It establishes a one-to-many relationship.

4.4. Gantt Chart

The Gantt Chart of our project where we worked for the whole semester to create this model is shown in a timeline pattern. It is the most important part to think and design the planning of your topic and so we planned our work like the gantt chart shown.

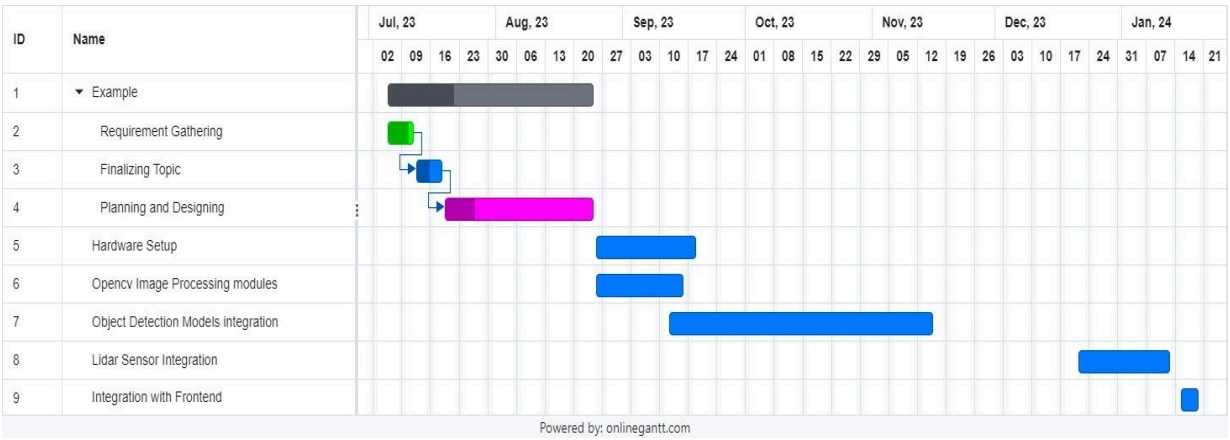


Fig 7. Gantt Chart

Chapter 5: Implementation of the Proposed System

5.1. Methodology employed for development

The methodology employed in the project combines experimental and observational approaches to achieve the research objectives. Data collection primarily involved the deployment of a sensor system, comprising LIDAR and a Raspberry Pi Camera. The data collection process included calibration of sensors, noise removal, and data fusion to integrate LIDAR measurements with camera images for object detection and distance estimation.

Object detection was performed using a pre-trained SSD MobileNet model, while data analysis was carried out using Python, OpenCV, and TensorFlow. Ethical considerations, such as obtaining informed consent and ensuring privacy, were rigorously observed.

Sampling was carried out conveniently, capturing data during various driving scenarios on Indian roads to encompass diverse environmental conditions. The study acknowledges limitations related to sample size and potential biases associated with convenience sampling.

This concise methodology section briefly outlines the key research methods and data collection techniques used in the project. It gives the reader an overview of the approach without going into exhaustive detail.

5.2 Algorithms and flowcharts for the respective modules developed

This study focuses on developing an innovative object detection system through the fusion of 2D LIDAR and camera sensors. By combining cost-effective 2D LIDAR technology with cameras, the system achieves real-time object detection, classification, and depth estimation. The integration of data from the 2D LIDAR sensor and the Raspberry Pi camera aims to create a driver assistance system that enhances road safety and driver awareness. Implementation will involve sensor fusion algorithms, object detection

models, and user-friendly interfaces to provide timely visual alerts to the driver.

Operating System: The system runs on Raspberry Pi OS (formerly Raspbian).

Sensor Interface:

- LIDAR Interface: The software facilitates communication and data reception from the 2D LIDAR sensor via a USB connection. To connect with the Raspberry Pi, the TF-Luna utilizes the Universal Asynchronous ReceiverTransmitter (UART) serial port, specifically, through GPIO pins. Before utilizing the LIDAR sensor with the Raspberry Pi, the mini UART port is enabled in the boot configuration file.
- Camera Interface: Software for capturing and processing images from the Raspberry Pi Camera.

Preprocessing: In this section, essential Preprocessing steps for Camera and LIDAR are outlined:

- Camera Preprocessing –
    - ➢ Grayscale Conversion: As shown in Figure 7, the RGB image is converted to grayscale. This step reduces dimensionality while preserving essential edge and gradient information.

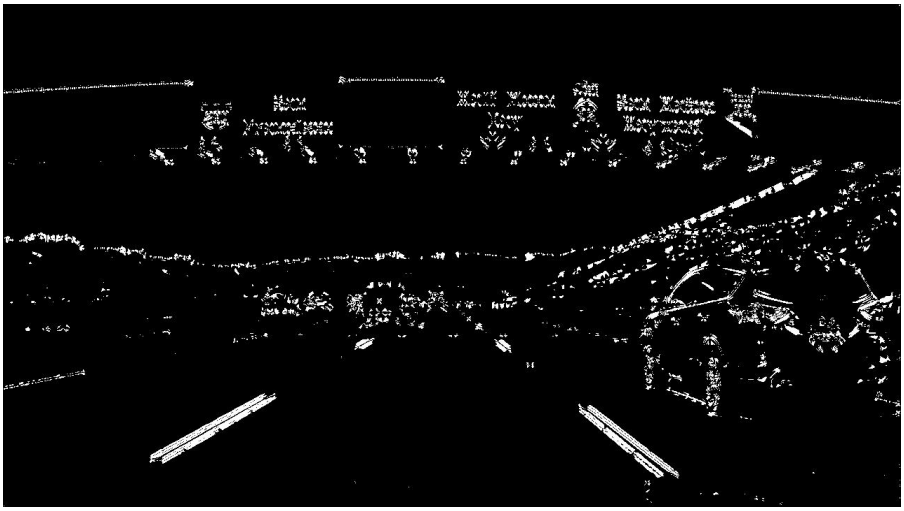


Fig 8: Grayscaling and Contour Detection

- ➢ Undistortion: Camera images undergo undistortion using calibration parameters derived from chessboard images. This correction addresses lens distortions, ensuring an accurate representation of objects.
- ➢ Colour Space Conversion: The RGB colour space is converted to the HLS colour space. This conversion facilitates the extraction of lane line information, which is crucial for object detection. – Gradient Thresholding: Sobel gradient thresholding is applied to the lightness channel (L-channel). This process enhances edge detection, improving the contrast between lane lines and the road surface.
- ➢ Color Thresholding: Colour thresholding is performed on the saturation channel (S-channel). It further isolates lane lines from the background, addressing challenges posed by varying lighting conditions and shadows.
- ➢ Perspective Transformation: Achieved through perspective warping, this step provides a bird's-eye view of the road. Lane lines are rendered as straight lines parallel to the image's vertical axis, simplifying lane detection.

These preprocessing steps collectively refine the quality and representation of camera images, establishing a solid foundation for accurate and reliable object detection in computer vision applications.

- **LIDAR preprocessing:** LIDAR preprocessing for TF-Luna data involves parsing raw sensor output and extracting relevant metrics. The TF-Luna sensor outputs the following data:
  - ➢ Distance: Measured in centimetres by default.
  - ➢ Signal strength (Amp): The reliability of the distance value is affected when the received signal is either overexposed (Amp = 0xFFFF) or too low Amp < 100.
  - ➢ Chip Temperature (Temp): The temperature in degrees Celsius is calculated as Temp/8 - 256°C.

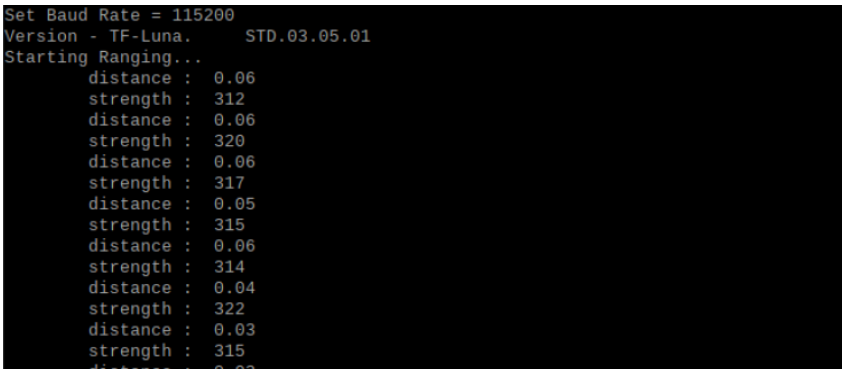


Fig. 9. Data obtained from TF-Luna LIDAR sensor.

As shown in Figure 8, the sensor prints out a test range and signal amplitude at the module's default band rate (115200) and sample rate (100Hz). The processed data is then suitable for obstacle detection, mapping, and object recognition applications.

Lane detection:

The lane detection pipeline initiates with the conversion of the input RGB image to grayscale. This step effectively reduces dimensionality while preserving crucial edge and gradient information. Following this, contour detection is applied to identify potential road surface obstructions or irregularities. Next, a perspective transformation grants us a top-down view of the road scene. This transformation significantly aids in accurate curved lane line detection. Simultaneously, a histogram of pixel intensities along the transformed image's vertical axis is computed. These intensity peaks correspond to lane lines. To iteratively search for lane pixels, the sliding window algorithm is employed within vertical windows of the transformed image. Additionally, polynomial curve fitting models the detected lane pixels, providing a reliable estimation of lane boundaries. Finally, the detected lane lines are visualized on the original undistorted image. This visualization offers a clear depiction of the vehicle's trajectory relative to the lane boundaries. The integrated approach combines various preprocessing and detection techniques, contributing to advancements in autonomous driving technology. Figures 9, 10 represent the results of various steps used in lane detection.

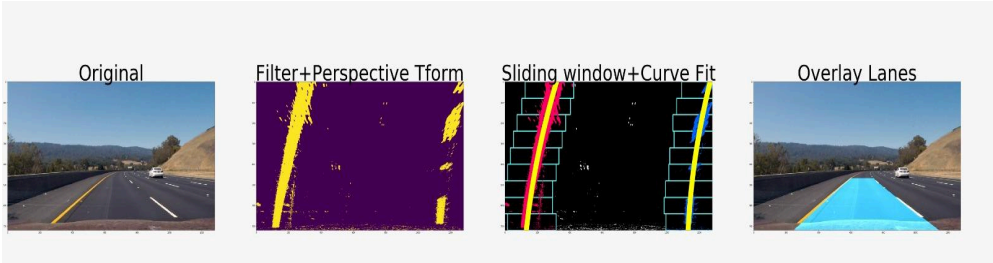


Fig. 10. Warping pipeline for curve detection

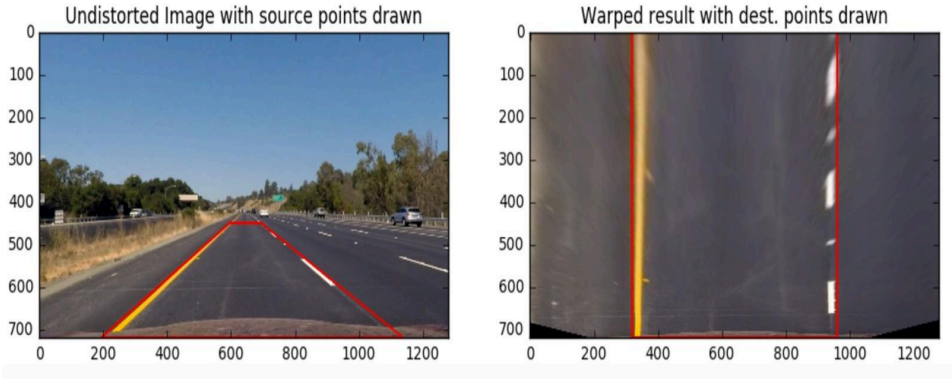


Fig. 11. Warping Results

Object Detection: Object Detection Algorithm:

The system will utilize object detection algorithms, such as SSD MobileNet, to identify and classify objects within camera images. SSD (Single Shot MultiBox Detector) employs a convolutional network that generates bounding boxes and class scores, refined through non-maximum suppression to produce final detections. This network is constructed upon a base architecture designed for high-quality image classification, augmented with layers dedicated to detection. These additional layers, progressively decreasing in size, facilitate multi-scale detection. Each layer generates detection predictions using convolutional filters, with a small kernel predicting parameters for potential detections, yielding either a category score or a shape offset. This process significantly enhances the system's perception and decision-making capabilities, thereby improving overall accuracy. In the realm of computer vision, YOLO (You Only Look Once) and MobileNet SSD (Single Shot

MultiBox Detector) stand out as widely used object detection models. YOLO conducts object detection in a single forward pass by dividing the input image into a grid and making predictions for bounding boxes along with confidence scores. Conversely, MobileNet SSD also performs a single forward pass but employs convolutional filters on feature maps at various scales to detect objects of different sizes. Although YOLO achieves marginally higher mean Average Precision (mAP), MobileNet SSD is optimized for real-time applications, running efficiently even on mobile devices[12]. Thus, we have chosen MobileNet SSD v2, the latest iteration, which sacrifices some accuracy compared to MobileNet SSD v1 but offers improved performance.

Table 1. YOLO and MobileNetSSD v2 comparison

Parameter	**YOLO**	**MobileNetSSD v2**
Params	Typically larger	Typically smaller
Multiply-Adds	Higher	Lower
mAP	Varies (depends on variant)	Generally moderate to high
Hardware Platform	Requires powerful GPU	Suitable for mobile CPUs

Sensor Fusion:

Algorithms and software for fusing data from the LIDAR sensor and camera to provide depth and direction information.

Raspberry Pi acts as the underlying hardware platform for hosting the system. As shown in Figure 7, the system comprises various processes including initializing system components, capturing images using the Raspberry Pi camera, detecting objects within captured images, and measuring distances to those objects using LIDAR. Additionally, it displays detected objects and their distances to the user, evaluates if detected objects are at a safe distance, and generates alerts if they are too close. Data is stored in separate data stores: The

Image Data Store temporarily holds captured images, the Detected Object Data Store stores information about detected objects, and the Measured Distance Data Store contains distance measurements. Data flows include transferring captured image data for processing, conveying information about detected objects, and transferring distance measurements. The system operates through different states: Idle, Object Detection, and Distance Measurement. In the Idle state, the system awaits user activation. Upon activation, it transitions to the Object Detection state, which actively identifies and locates objects. If no objects are detected, it returns to the Idle state; otherwise, it moves to the Distance Measurement state to measure distances to the detected objects. If distances are within a safe range, it returns to the Object Detection state. Otherwise, it triggers alert generation due to proximity.

5.3 Datasets source and utilization

Data is stored in separate data stores: The Image Data Store temporarily holds captured images, the Detected Object Data Store stores information about detected objects, and the Measured Distance Data Store contains distance measurements. Data flows include transferring captured image data for processing, conveying information about detected objects, and transferring distance measurements. The system operates through different states: Idle, Object Detection, and Distance Measurement. In the Idle state, the system awaits user activation. Upon activation, it transitions to the Object Detection state, which actively identifies and locates objects. If no objects are detected, it returns to the Idle state; otherwise, it moves to the Distance Measurement state to measure distances to the detected objects. If distances are within a safe range, it returns to the Object Detection state. Otherwise, it triggers alert generation due to proximity.

Chapter 6: Testing of the Proposed System

6.1. Introduction to testing

Testing plays a critical role in ensuring the reliability, functionality, and performance of any software or hardware system, including driver assistance systems. It involves systematically evaluating the system's capabilities, identifying potential weaknesses or defects, and verifying that it meets specified requirements and standards. The primary objectives of testing are to validate the system's functionality, detect and resolve any issues or bugs, and ultimately enhance its quality and reliability.

In the context of driver assistance systems, testing is especially crucial due to the system's direct impact on road safety and user well-being. The complexity of these systems, which often involve integrating various sensors, algorithms, and hardware components, necessitates comprehensive testing methodologies to ensure their effectiveness in real-world scenarios.

6.2. Types of Tests Considered:

- Functional Testing: Testing the functionality of the driver assistance system to ensure that it performs its intended tasks correctly, such as object detection, lane detection, distance measurement, and collision avoidance.
- Performance Testing: Evaluating the system's performance metrics, including processing speed, accuracy, and responsiveness, under various conditions.
- Integration Testing: Verifying the integration of different components, such as the LIDAR sensor, camera data processing, and collision avoidance algorithms, to ensure they work together seamlessly.
- Robustness Testing: Assessing the system's ability to handle unexpected or adverse conditions, such as sudden changes in lighting, extreme weather, or sensor failures.
- Usability Testing:Evaluating the system's user interface and overall user experience to ensure it is intuitive and easy to use for drivers.

6.3. Various Test Case Scenarios Considered:

1. Object Detection Accuracy Test: Test scenarios to evaluate the accuracy of object detection under different lighting conditions, varying object sizes, and cluttered environments.
2. Distance Measurement Accuracy Test: Test scenarios to verify the accuracy of distance measurement using the LIDAR sensor across different distances and environmental conditions.
3. Lane Detection Test: Test scenarios to assess the system's ability to accurately detect lane boundaries under various road conditions, including curves, intersections, and lane markings.
4. Collision Avoidance Test: Simulated scenarios to test the system's collision avoidance capabilities, including detecting and responding to potential collision threats in real-time.
5. Performance under Adverse Conditions:Test scenarios to evaluate the system's performance under adverse weather conditions such as rain, fog, or snow, and in low-light or high-glare environments.
6. Real-World Road Scenarios: Test scenarios mimicking real-world driving conditions, including urban streets, highways, rural roads, and congested traffic situations.
7. System Integration Test: Testing the integration of hardware and software components to ensure proper communication and functionality between the Raspberry Pi, LIDAR sensor, camera, and collision avoidance algorithms.

6.4. Inferences Drawn from the Test Cases:

1. Object Detection and Lane Detection Accuracy: The system exhibits moderate accuracy 70% to 90%, in object detection and lane detection, with performance varying based on factors such as object size, lighting conditions, and road markings.
2. Distance Measurement Accuracy: The distance measurement accuracy meets the specified error margins within the designated range of the LIDAR sensor, but performance may degrade in certain scenarios, such as low visibility conditions or highly reflective surfaces.

3. Collision Avoidance Performance: The system demonstrates effective collision avoidance capabilities, successfully detecting and responding to potential collision threats in most test scenarios.
4. Robustness under Adverse Conditions:The system shows resilience to adverse weather conditions and low-light environments, although performance may degrade in extreme conditions such as heavy rain or dense fog.
5. Real-World Road Scenario Performance:The system performs well in typical driving scenarios, including urban streets and highways, but may encounter challenges in complex situations such as navigating through dense traffic or negotiating tight spaces.

These inferences provide insights into the strengths and limitations of the driver assistance system, helping to identify areas for improvement and optimization to enhance overall road safety and effectiveness.

Chapter 7: Results and Discussion

7.1. Screenshots of User Interface (UI) for the respective module

Fig. 12. Giving Alert Signals when the object is near

Fig. 13. Lane and object detection on a curved road

Fig. 14. Final Result

- As Figure 11. shows the lane detection and object detection having accuracy of person(56.6%), truck1(89.01%), truck2(55.16%). Since the distance (0.21m) is less than the threshold value an alert is generated.
- As Figure 12. shows the lane detection and object detection having accuracy of person1(41.27%), truck(93%),car(91.49%). The distance is 0.42m.
- As Figure 13. shows the lane detection and object detection accuracy of the car(99.49%). The distance is 2.33m.

7.2. Performance Evaluation Measures

1. Object Detection Model Performance Metrics:

MobileNetV2 + SSDLite refers to the specific architecture used for object detection, consisting of MobileNetV2 as the base convolutional neural network (CNN) for feature extraction and SSDLite for the detection head.

- Model Parameters: Indicates the number of trainable parameters in the model, which affects its complexity and memory requirements.

- Multiply-Adds: Represents the number of multiplication and addition operations performed during inference, providing insights into computational efficiency.
- mAP (Mean Average Precision): This metric measures the average precision across different object categories, considering both precision (the ratio of true positives to all predicted positives) and recall (the ratio of true positives to all actual positives). It quantifies the model's ability to identify objects across different categories precisely.
- Mobile CPU Processing Time: Reflects the time taken by the model to process an input image on a mobile CPU, influencing real-time performance and responsiveness.

2. TF-Luna Micro LIDAR Sensor Performance Metrics:

Error Rate: Indicates the margin of error in distance measurement provided by the LIDAR sensor. It comprises two components:

- ➢ Absolute Error: The actual discrepancy between the measured distance and the true distance, expressed in units (e.g., centimetres).
- ➢ Relative Error: The percentage error relative to the true distance, providing insights into the sensor's accuracy across different distance ranges.

The performance evaluation measures, including the accuracy of the object detection model and error rates of the LIDAR sensor, are fundamental for assessing the reliability and efficacy of the system in real-world scenarios. They enable informed decision-making regarding the technology's suitability for enhancing road safety and driver assistance applications.

7.3. Input parameters/ Features Considered

1. Operating Range of TF-Luna Sensor: The sensor's operational capacity spans from 0.2 meters to 8 meters, defining the distance range within which it accurately measures distances.

2. Error Rate of TF-Luna Sensor: The sensor's error rate is crucial, impacting the precision of distance measurements. Within the distance range of 0 to 3 meters, the error rate is ±6 centimeters, while within the range of 3 to 8 meters, the error rate is ±2%.

3. Serial Communication: Serial communication protocols are employed for data exchange between the TF-Luna sensor and the system. These protocols facilitate the transmission of distance, signal strength, and temperature data from the sensor to the system for further processing and analysis.

4. Data Processing Algorithms: Specialized algorithms are implemented to process the data obtained from the TF-Luna sensor. These algorithms involve filtering, scaling, and offset calculations to ensure accurate and reliable distance measurements under various environmental conditions.

5. Integration with Object Detection System: The TF-Luna sensor is seamlessly integrated with an object detection system, enabling comprehensive environmental perception for enhanced safety and situational awareness. This integration involves data fusion techniques to combine LIDAR data with other sensor inputs, enhancing the system's capability to detect and localize objects in real-time.

6. Utilization of Camera Images: Images captured from the camera are utilized in conjunction with the TF-Luna sensor data to provide visual context for object detection and localization. These images, with dimensions typically ranging from 640x360 pixels to 1280x720 pixels, are processed using advanced computer vision algorithms, including object detection and lane detection, to identify and classify objects in the sensor's vicinity. This visual information enhances the system's ability to perceive and interpret the surrounding environment, contributing to improved decision-making and response capabilities.

7. Graphical User Interface (GUI): A user-friendly GUI is developed to visualize the data acquired from the TF-Luna sensor and camera images. The GUI provides interactive displays of signal strength, distance measurements, and processed camera images, enabling users to monitor and analyze sensor data effectively in conjunction with visual context.

8. Real-time Data Processing: The system incorporates real-time data processing capabilities to handle the continuous stream of data from the TF-Luna sensor and camera. This enables timely decision-making and response to dynamic changes in the environment, ensuring efficient operation in various scenarios.

9. Baud Rate Configuration: Configurable baud rates are utilized to optimize communication between the system and the TF-Luna sensor. By adjusting the baud rate settings, the system can achieve optimal data transmission rates, enhancing overall performance and reliability.

7.4. Graphical and statistical output

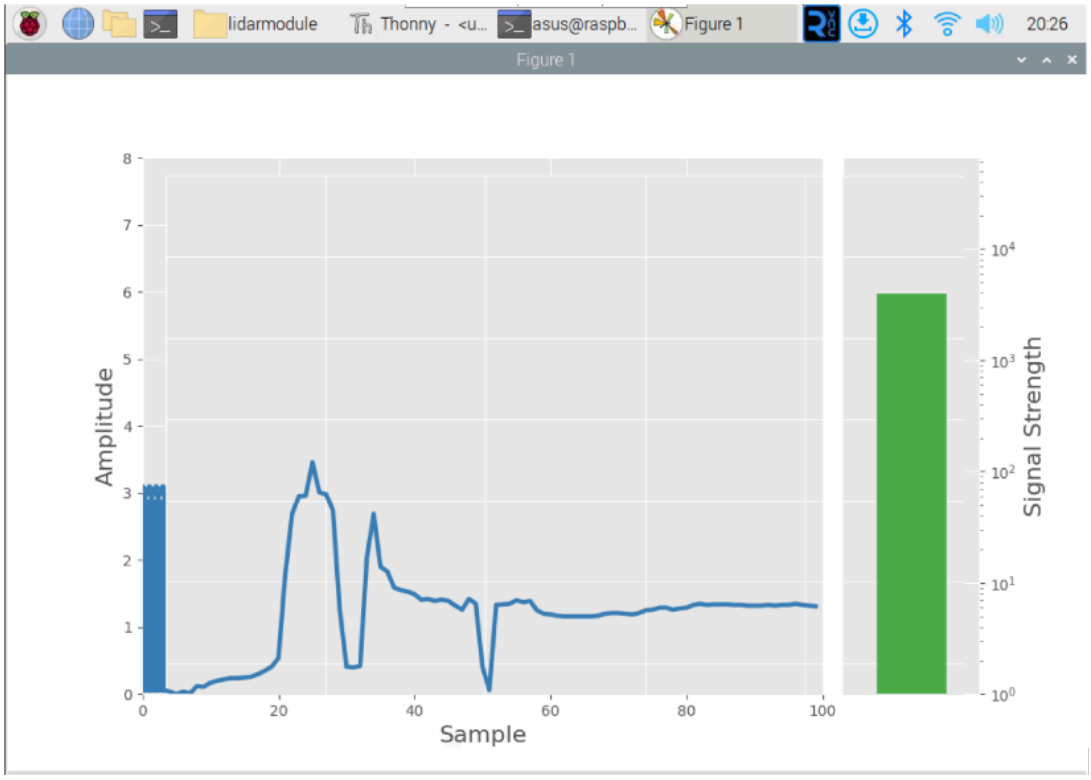


Fig. 15. Live distance measurements provided by the TF-Luna LIDAR

The graphical representation of information from the TF-Luna sensor serves as a valuable tool for users to interpret and analyze data effectively. It comprises two main visualizations:

1. Bar Chart for Signal Strength (Amplitude):
    - This chart presents signal strength, often represented by amplitude, as bars of varying heights.
    - Users can quickly identify instances of weak signal strength or fluctuations in signal intensity, which may indicate potential issues such as obstacles or environmental interference.
    - The bar chart offers a visual overview of signal strength distribution, aiding in the identification of patterns or anomalies.
2. Time-Series Graph for Distance Measurements:

- This graph displays distance measurements over time, typically with time on the x-axis and distance on the y-axis.
- Users can track changes in distance readings over time, observing trends, variations, or sudden changes in the measured distances.
- The time-series graph enables users to analyze the temporal dynamics of distance measurements, facilitating the detection of moving objects, changes in environment, or system performance over time.

Together, these graphical representations provide users with a comprehensive understanding of the data captured by the TF-Luna sensor. By visually presenting both signal strength and distance measurements, users can identify critical information such as weak signals, distance variations, and potential obstacles, empowering them to make informed decisions and take appropriate actions based on the sensor data.

7.5. Comparison with Existing Systems

1. Object Detection Accuracy:

   The object detection algorithm utilized in our system is SSD MobileNet, specifically MobileNet SSD v2. This algorithm employs a convolutional neural network architecture designed to identify and classify objects within camera images. SSD generates bounding boxes and class scores, refined through non-maximum suppression, to produce final detections. Additional layers, progressively decreasing in size, facilitate multi-scale detection. Each layer generates detection predictions using convolutional filters, predicting parameters for potential detections, thereby enhancing the system's perception and decision-making capabilities. While YOLO achieves marginally higher mean Average Precision (mAP), MobileNet SSD v2 is optimized for real-time applications, offering improved performance even on mobile devices. Therefore, MobileNet SSD v2 is chosen for our system, prioritizing improved performance over slightly reduced accuracy compared to previous iterations.

2. Distance Measurement Accuracy:

   The system employs the TF-LUNA Micro LIDAR sensor for distance measurement, with an error margin of ±6 cm for distances less than 3 meters and ±2% for distances between 3 meters and 8 meters.

   The above-mentioned LiDAR is a 2-Dimensional LiDAR sensor, which reduces the overall volume of the data to be processed increasing efficiency without compromising on accuracy with respect to the use case.

3. Computational Efficiency and Resource Utilization:

   Our decision to implement the system on Raspberry Pi was driven by several factors, including cost-effectiveness and modularity. While Raspberry Pi may have limitations in computational resources compared to more powerful hardware platforms used in existing driver assistance systems, it offers several advantages.

   - Cost-Effectiveness: Raspberry Pi is known for its affordability, making it an attractive option for projects with budget constraints. By leveraging Raspberry Pi, we were able to develop a cost-effective solution without compromising on functionality.
   - Modularity: Raspberry Pi's modular design allows for easy integration with various sensors and peripherals, making it suitable for prototyping and experimentation. This modularity enables flexibility in system configuration and facilitates future upgrades or modifications as needed.

   While Raspberry Pi may not offer the same level of computational performance as high-end hardware platforms, its capabilities are sufficient for many driver assistance applications, especially when optimized for efficiency. Additionally, the compact form factor of Raspberry Pi makes it suitable for embedded systems, enabling deployment in vehicles with limited space.

Chapter 8: Conclusion

8.1. Limitations

The driver assistance system faces limitations in object detection accuracy, distance measurement errors, LIDAR sensor range, computational resources, environmental factors, and regulatory considerations.

8.2. Conclusion

The system introduces an innovative approach to object detection, combining 2D LIDAR and camera data. The goal is to create a cost-effective and reliable solution that significantly enhances road safety and driver awareness. By leveraging the capabilities of a 2D LIDAR sensor and a Raspberry Pi camera, the system provides real-time visual alerts to drivers, making driving safer. Sensor fusion techniques and robust object detection algorithms are employed to achieve this. The user-friendly interface ensures seamless interaction for drivers. Despite some limitations, such as the constraints of 2D LIDAR and regulatory compliance gaps, the system strives to offer a valuable, low-cost driver assistance solution.

As the system evolves, it remains committed to fine-tuning its performance across diverse environmental conditions and optimizing real-time data processing. With accessibility and effectiveness at the forefront, the driver assistance system aims to meet the dynamic needs of the automotive industry and contribute to the broader goal of road safety and accident prevention. The system is implemented on a Raspberry Pi, processing real-time data from LIDAR for distance measurement and a camera for object detection. This enables smart surround-view and collision avoidance, significantly enhancing road safety.

8.3. Future Scope

1. Reducing latency of the system.
2. Improving overall performance and accuracy of the system.
3. Building an alert system for collision detection and collision avoidance.
4. Building a dedicated user interface and dashboard for presenting data generated and insights generated from the collected data.

References

[1] Lee, H., Chae, H. and Yi, K. (2019) "A geometric model based 2D LIDAR/radar sensor fusion for tracking surrounding vehicles", IFAC-PapersOnLine, 52(8).

[2] Wang, B., Yan, X. and Li, D. (2022) "An end-to-end lane detection model with attention and residual block", Computational Intelligence and Neuroscience, 2022.

[3] ADAS Car using Raspberry Pi AI / ML R&D Specialist, KLA-Tencor CSE, IIT B Alumnus | Ex-Vice President — AI & DS, IIA | Ex-TSMC

[4] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J.,& Whittaker, W. (2006). Stanley: The robot that won the DARPA Grand Challenge. Journal of Field Robotics, 23(9), 661-692.

[5] Liu, H., Wu, C. and Wang, H. (2023) 'Real time object detection using LIDAR and camera fusion for autonomous driving', Scientific Reports, 13(1).

[6] Dang, X.; Rong, Z.; Liang, X. Sensor Fusion-Based Approach to Eliminating Moving Objects for SLAM in Dynamic Environments. Sensors 2021, 21, 230.

[7] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems (NIPS).

[8] Wang, C., & Deng, W. (2019). Towards End-to-End Lane Detection: an Instance Segmentation Approach. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[9] Parekh, D. et al. (2022) 'A review on Autonomous Vehicles: Progress, methods and challenges', Electronics, 11(14), p. 2162.

[10] Haghbayan, Hashem, et al., "An Efficient Multi-sensor Fusion Approach for Object Detection in Maritime Environments," Advanced Science and Technology Letters,

[11] The Potential of Multi-Sensor Fusion in Diverse Applications

[12] Liu, Wei, et al. "Ssd: Single shot multi-box detector." Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. Springer International Publishing, 2016.

[13] Vehicle imaging processing system and method having obstructed image detection https://patents.google.com/patent/WO2003093857A3/en

[14] Lane tracking system https://patents.google.com/patent/US20130141520/zh

[15] Ramachandra, A.C. et al. (2022) 'Advanced Driving Assistance System for cars using Raspberry Pi', Indian Journal Of Science And Technology, 15(40).

Appendix

1. Paper I

a. Paper I

Revolutionizing Road Safety: Raspberry Pi based driver assistance system for Indian Roads using LIDAR and Camera Sensors

Kaustubha Sahu
Department of Computer Engg.
VES Institute of Technology
Mumbai,India-400074
2020.kaustubha.sahu@ves.ac.in

Pranav Ubarhande
Department of Computer Engg.
VES Institute of Technology
Mumbai,India-400074
2020.pranav.ubarhande@ves.ac.in

Aaryan Manawat
Department of Computer Engg.
VES Institute of Technology
Mumbai,India-400074
2020.aaryan.manawat@ves.ac.in

Abhiman Singh
Department of Computer Engg.
VES Institute of Technology
Mumbai, India-400074
2020.abhiman.rajput@ves.ac.in

Sanjay Mirchandani
Department of Computer Engg.
VES Institute of Technology
Mumbai, India-400074
sanjay.mirchandani@ves.ac.in

***Abstract*—A driver assistance system is implemented on Raspberry Pi which processes real-time data from the LIDAR for distance measurement and camera data for object detection and lane detection, enabling smart surround-view and collision avoidance to enhance road safety. The object detection system is based on MobileNetSSD v2 and achieves a mean average precision (mAP) of 22.1%. The distance calculation is performed using the TF-LUNA Micro LIDAR sensor, which operates within a range of 8 meters and has an error margin of ±6 cm for distances less than 3 meters, and an error margin of ±2% for distances between 3 meters and 8 meters. Thus, the system is designed to handle the unique challenges of Indian roads, to prevent accidents, and ultimately save lives.**

***Index Terms*—Road Safety, Driver Assistance System, LIDAR, Camera, Object Detection, Lane Detection.**

I. Introduction

The need for strong and precise object detection systems has increased across various industries, such as autonomous vehicles, robotics, surveillance, and industrial automation, in today's fast-changing technological landscape. However, it is still a difficult challenge to achieve accurate object detection in complex and dynamic environments. Traditional single-sensor solutions often fail to deal with the complexities of real-world scenarios, such as bad weather conditions, low-light environments, and the presence of occlusions. A promising solution is the integration of multiple sensors, which enhances the capabilities of object detection systems[11][13][15]. The concept of multi-sensor fusion, amalgamating data from diverse sensors, offers a complete approach to perceiving and recognising in complex environments. By combining the strengths of different sensor modalities, such as cameras, LIDAR, sensors, etc., multi-sensor fusion systems can significantly improve the accuracy, reliability, and robustness of object detection. A LIDAR excels at distance estimation, functioning as a Time-Of-Flight sensor. Meanwhile, a camera is adapted to object classification and scene comprehension. Cameras are commonly used to generate bounding boxes, identify lane line positions, recognize traffic light colours, and detect traffic signs, among other applications. The demand for accurate and dependable object detection in complex and dynamic environments has increased in an era of rapid technological progress and the spread of automation and autonomous systems.

The study of the current system is summarized in the related work section II. The methodology section, section III provides a succinct overview of the main research methods and data gathering techniques employed in the project. The implementation of the system is detailed in the Implementation in section IV, which includes an overview of the hardware and software, the integration of sensors and the operation of the entire system. This is followed by a presentation of the results obtained from the system in section V.

II. Related Work

Kyongsu Yi, Heungseok Chae, and Hojoon Lee [1] introduce a novel sensor fusion system for autonomous vehicles, merging Radar's large Field of View (FOV) with LIDAR's great distance accuracy, utilizing a multiple hypothesis tracking (MHT) technique. Xiaoting Yan, Deguang Li, and Bo Wang [2] examine classification systems for Sensor Fusion, emphasizing its critical role in autonomous vehicles and supporting dependable autonomous driving. An IIT B alumnus and KLA-Tencor CSE propose [3] the significance of Sensor Fusion in overcoming individual sensor limitations for autonomous cars, advocating for reasonably priced driver assistance systems. [4] discusses object identification and location regression for autonomous driving, proposing a LIDAR-camera fusion approach using a Siamese network.Ponn, T.,

Kröger, T. and Diermeyer in [5] conducts a thorough analysis of object detectors in automated driving using meta-information. Machine learning-based detectors' behavior is modeled using random forests, showing high accuracy in representing detection performance. W. Deng and C. Wang [6] propose an instance segmentation-based approach to lane identification, offering significant advancements in the creation of dependable lane-keeping systems. [7] presents an improved method for real-time object tracking and detection in autonomous transportation systems, advocating for multi-sensor fusion and collaboration between tracking and detection algorithms to enhance applicability and reliability in dynamic situations.

III. Methodology

This brief methodology section summarizes the key research methods and data collection techniques used in the system.

1) **Sensor Fusion:** Sensor fusion techniques combine data from multiple sensors, such as LIDAR and cameras, to improve object detection and depth estimation.
2) **Object Detection Algorithms:** Utilizing MobilenetSSDv2 for real-time detection and classification of objects in camera images and OpenCV for lane detection.
3) **Depth Estimation:** Techniques for estimating object distance from the sensor (TF-Luna micro LIDAR), including stereo vision, structure from motion (SfM), and point cloud processing, have been used to provide depth information.
4) **Machine Learning:** Utilizing machine learning models, such as neural networks and deep learning, for training object detection and classification algorithms.
5) **Calibration and Alignment:** Methods for sensor calibration and alignment have been crucial for ensuring accurate sensor fusion and depth estimation.
6) **Real-time Processing:** Systems have focused on optimizing real-time data processing to provide timely alerts and assistance to drivers.

IV. Implementation

This study focuses on developing an innovative object detection system through the fusion of 2D LIDAR and camera sensors. By combining cost-effective 2D LIDAR technology with cameras, the system achieves real-time object detection, classification, and depth estimation. The integration of data from the 2D LIDAR sensor and the Raspberry Pi camera aims to create a driver assistance system that enhances road safety and driver awareness. Implementation will involve sensor fusion algorithms, object detection models, and user-friendly interfaces to provide timely visual alerts to the driver[14].

A. Overview of Hardware Components

1) **Sensor Integration**
   - 2D LIDAR Sensor: The TF-Luna as shown in Figure 1, is a Light Detection And Ranging (LIDAR) module. It can accurately measure distances up to 8m.
   - Raspberry Pi: A Raspberry Pi 4 (8GB RAM) single-board computer for data processing and integration.
   - Raspberry Pi Camera: As shown in Figure 2, the camera serves for object classification and scene understanding. This Pi camera supports 1080p resolution with a frame rate of 30 fps with a 5MP sensor.

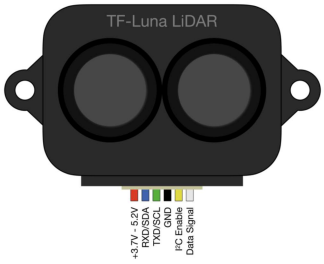


Fig. 1. TF-Luna Mini LIDAR

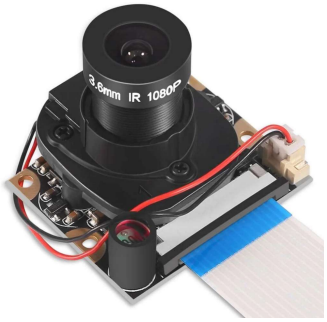


Fig. 2. Raspberry PI Camera Module 5MP 1080p

2) **Power Supply:** A 5V 3A 10,000 mAh battery for powering the system.
3) **Connectivity:**
   - Type C-USB Cable: A Type C-USB cable to connect the power supply to the Raspberry Pi.
   - FFC cable to connect the camera to Raspberry Pi.
   - JST Jumper 2 cable to connect the Lidar sensor to Raspberry Pi.

B. Overview of Software Components

Operating System: The system runs on Raspberry Pi OS (formerly Raspbian).

Sensor Interface

- LIDAR Interface: The software facilitates communication and data reception from the 2D LIDAR sensor via a USB connection. To connect with the Raspberry Pi, the TF-Luna utilizes the Universal Asynchronous Receiver-Transmitter (UART) serial port, specifically, through GPIO pins. Before utilizing the LIDAR sensor with the Raspberry Pi, the mini UART port is enabled in the boot configuration file.

- Camera Interface: Software for capturing and processing images from the Raspberry Pi Camera.

Preprocessing: In this section, essential Preprocessing steps for Camera and LIDAR are outlined:

1) Camera Preprocessing:

- Grayscale Conversion: As shown in Figure 3, the RGB image is converted to grayscale. This step reduces dimensionality while preserving essential edge and gradient information.

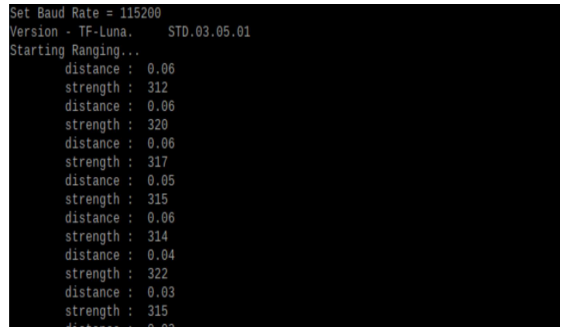
Fig. 3. Grayscaling and Contour Detection

- Undistortion: Camera images undergo undistortion using calibration parameters derived from chessboard images. This correction addresses lens distortions, ensuring an accurate representation of objects.
- Colour Space Conversion: The RGB colour space is converted to the HLS colour space. This conversion facilitates the extraction of lane line information, which is crucial for object detection.
- Gradient Thresholding: Sobel gradient thresholding is applied to the lightness channel (L-channel). This process enhances edge detection, improving the contrast between lane lines and the road surface.
- Color Thresholding: Colour thresholding is performed on the saturation channel (S-channel). It further isolates lane lines from the background, addressing challenges posed by varying lighting conditions and shadows.
- Perspective Transformation: Achieved through perspective warping, this step provides a bird's-eye view of the road. Lane lines are rendered as straight lines parallel to the image's vertical axis, simplifying lane detection.

These preprocessing steps collectively refine the quality and representation of camera images, establishing a solid foundation for accurate and reliable object detection in computer vision applications.

2) LIDAR preprocessing: LIDAR preprocessing for TF-Luna data involves parsing raw sensor output and extracting relevant metrics. The TF-Luna sensor outputs the following data:

- Distance: Measured in centimetres by default.
- Signal strength (Amp): The reliability of the distance value is affected when the received signal is either overexposed (Amp = 0xFFFF) or too low Amp < 100.
- Chip Temperature (Temp): The temperature in degrees Celsius is calculated as Temp/8 - 256°C.



```
Set Baud Rate = 115200
Version - TF-Luna.     STD.03.05.01
Starting Ranging...
        distance :  0.06
        strength :  312
        distance :  0.06
        strength :  320
        distance :  0.06
        strength :  317
        distance :  0.05
        strength :  315
        distance :  0.06
        strength :  314
        distance :  0.04
        strength :  322
        distance :  0.03
        strength :  315
```

Fig. 4. Data obtained from TF-Luna LIDAR sensor.

As shown in Figure 4, the sensor prints out a test range and signal amplitude at the module's default baud rate (115200) and sample rate (100Hz). The processed data is then suitable for applications such as obstacle detection, mapping, and object recognition.

Lane detection: The lane detection pipeline initiates with the conversion of the input RGB image to grayscale. This step effectively reduces dimensionality while preserving crucial edge and gradient information. Following this, contour detection is applied to identify potential road surface obstructions or irregularities. Next, a perspective transformation grants us a top-down view of the road scene. This transformation significantly aids in accurate curved lane line detection. Simultaneously, a histogram of pixel intensities along the transformed image's vertical axis is computed. These intensity peaks correspond to lane lines.

To iteratively search for lane pixels, the sliding window algorithm is employed within vertical windows of the transformed image. Additionally, polynomial curve fitting models the detected lane pixels, providing a reliable estimation of lane boundaries. Finally, the detected lane lines are visualized on the original undistorted image. This visualization offers a clear depiction of the vehicle's trajectory relative to the lane boundaries. The integrated approach combines various preprocessing and detection techniques, contributing to advancements in autonomous driving technology. Figures 5, 6 represent the results of various steps used in lane detection.

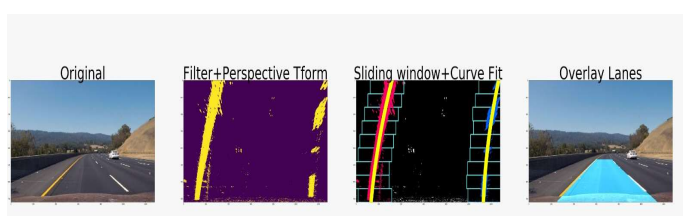


Fig. 5. Warping pipeline for curve detection

Object Detection: Object Detection Algorithm: The system will utilize object detection algorithms, such as SSD

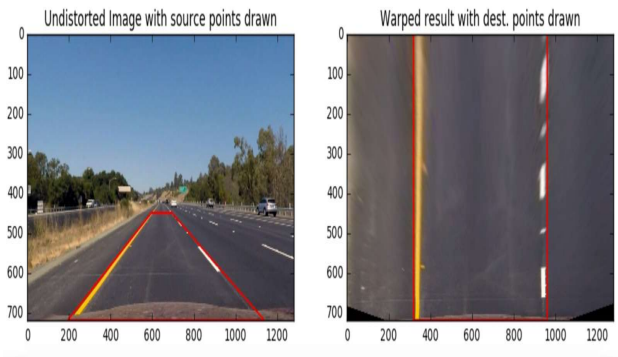


Fig. 6. Warping Results

MobileNet, to identify and classify objects within camera images. SSD (Single Shot MultiBox Detector) employs a convolutional network that generates bounding boxes and class scores, refined through non-maximum suppression to produce final detections. This network is constructed upon a base architecture designed for high-quality image classification, augmented with layers dedicated to detection. These additional layers, progressively decreasing in size, facilitate multi-scale detection. Each layer generates detection predictions using convolutional filters, with a small kernel predicting parameters for potential detections, yielding either a category score or a shape offset. This process significantly enhances the system's perception and decision-making capabilities, thereby improving overall accuracy. In the realm of computer vision, YOLO (You Only Look Once) and MobileNet SSD (Single Shot MultiBox Detector) stand out as widely used object detection models. YOLO conducts object detection in a single forward pass by dividing the input image into a grid and making predictions for bounding boxes along with confidence scores. Conversely, MobileNet SSD also performs a single forward pass but employs convolutional filters on feature maps at various scales to detect objects of different sizes. Although YOLO achieves marginally higher mean Average Precision (mAP), MobileNet SSD is optimized for real-time applications, running efficiently even on mobile devices[12]. Thus, we have chosen MobileNet SSD v2, the latest iteration, which sacrifices some accuracy compared to MobileNet SSD v1 but offers improved performance.

TABLE I
YOLO AND MOBILENETSSD V2 COMPARISON

Parameter	YOLO	MobileNetSSD v2
Params	Typically larger	Typically smaller
Multiply-Adds	Higher	Lower
mAP	Varies (depends on variant)	Generally moderate to high
Hardware Platform	Requires powerful GPU	Suitable for mobile CPUs

Sensor Fusion: Algorithms and software for fusing data from the LIDAR sensor and camera to provide depth and direction information.

C. Flowchart

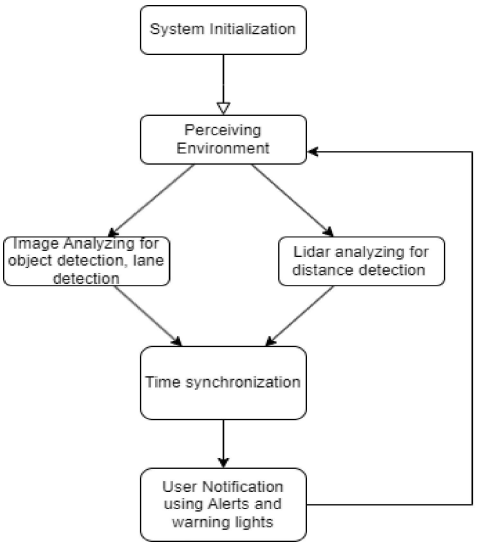


Fig. 7. Flowchart of the system

Raspberry Pi acts as the underlying hardware platform for hosting the system. As shown in Figure 7, the system comprises various processes including initializing system components, capturing images using the Raspberry Pi camera, detecting objects within captured images, and measuring distances to those objects using LIDAR. Additionally, it displays detected objects and their distances to the user, evaluates if detected objects are at a safe distance, and generates alerts if they are too close.

Data is stored in separate data stores: The Image Data Store temporarily holds captured images, the Detected Object Data Store stores information about detected objects, and the Measured Distance Data Store contains distance measurements. Data flows include transferring captured image data for processing, conveying information about detected objects, and transferring distance measurements.

The system operates through different states: Idle, Object Detection, and Distance Measurement. In the Idle state, the system awaits user activation. Upon activation, it transitions to the Object Detection state, which actively identifies and locates objects. If no objects are detected, it returns to the Idle state; otherwise, it moves to the Distance Measurement state to measure distances to the detected objects. If distances are within a safe range, it returns to the Object Detection state. Otherwise, it triggers alert generation due to proximity.

V. RESULTS

The system, which was designed for Indian roads, underwent implementation on different roads. Following this, an analysis was conducted, and the results obtained from the system are presented in this section.

Fig. 8. Real-time object, lane detection and distance estimation on Indian road

1) Camera Outcome Illustration: As shown in Figure 8, the objects detected with accuracy were car (98.01%), motorbike(94.64%), and person(60.29%). The green patch shows the detected lane.

LIDAR Visual Representation

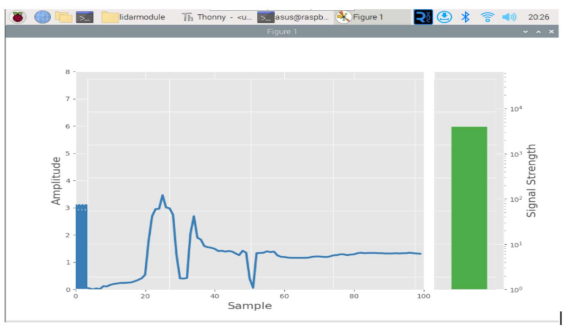


Fig. 9. Live distance measurements provided by the TF-Luna LIDAR

As illustrated in Figure 9, the information obtained from the TF-Luna sensor is presented graphically as signal strength (amplitude) on a bar chart and distance measurements on a time-series graph. Users can discover instances of weak signal strength in certain items or scan routines with the help of this graphical representation.

An alert is given to the user if the distance drops below a predetermined threshold, of 0.40 meters.

Integrated System:

As Figure 10. shows the lane detection and object detection having accuracy of person(56.6%), truck1(89.01%), truck2(55.16%). Since the distance (0.21m) is less than the threshold value an alert is generated.

As Figure 11. shows the lane detection and object detection having accuracy of person1(41.27%), truck(93%), car(91.49%).The distance is 0.42m.

As Figure 12. shows the lane detection and object detection accuracy of the car(99.49%). The distance is 2.33m.

The system code can process video input from a camera or video file, use OpenCV to detect lane lines and train a

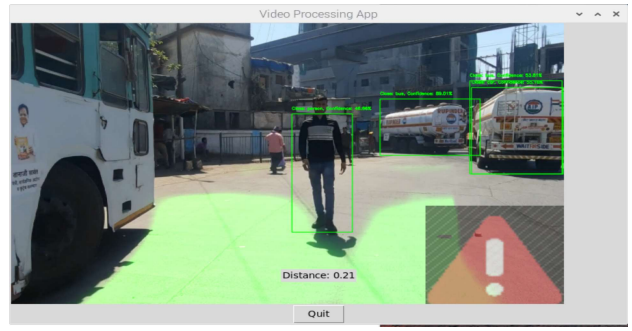


Fig. 10. Giving Alert Signals when the object is near

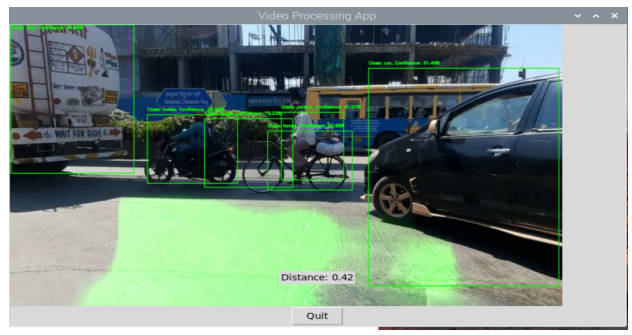


Fig. 11. lane and object detection on a curved road

Fig. 12. Final Result

MobileNetSSDv2 model for object detection. To calculate the distance to the closest object in front of the system, it additionally combines the TF-Luna LIDAR sensor. The processed footage is shown in a graphical user interface window with real-time object and lane line overlays. It also saves the video after processing it into a new file. A real-time plot showing the LIDAR sensor's distance data is also incorporated into the system. An alert icon is incorporated into the video feed to alert the user if the distance drops below a predetermined threshold. The script runs indefinitely to add new data to the plot and update the video feed. Lastly, it can gently close the program and release the video files when you close it.

Object detection model performance metrics:

During road testing in Mumbai, India, the object detection model— MobileNet SSD demonstrated an accuracy range of approximately 70% to 90%.

TABLE II
OBJECT DETECTION MODEL PERFORMANCE METRICS

Model	Params	Multiply-Adds	mAP	Mobile CPU
MobileNetV2 + SSDLite	4.3M	0.8B	22.1%	200ms

The TF-Luna Micro LIDAR sensor has an operating range of 0.2m to 8m. TF-Luna Micro LIDAR sensor error rate:

TABLE III
TF-LUNA MICRO LIDAR SENSOR ERROR RATE

Distance Range (meters)	Error Rate
0 - 3m	± 6cm
3m - 8m	± 2%

VI. CONCLUSION

The system introduces an innovative approach to object detection, combining 2D LIDAR and camera data. The goal is to create a cost-effective and reliable solution that significantly enhances road safety and driver awareness. By leveraging the capabilities of a 2D LIDAR sensor and a Raspberry Pi camera, the system provides real-time visual alerts to drivers, making driving safer. Sensor fusion techniques and robust object detection algorithms are employed to achieve this. The user-friendly interface ensures seamless interaction for drivers. Despite some limitations, such as the constraints of 2D LIDAR and regulatory compliance gaps, the system strives to offer a valuable, low-cost driver assistance solution.

As the system evolves, it remains committed to fine-tuning its performance across diverse environmental conditions and optimizing real-time data processing. With accessibility and effectiveness at the forefront, the driver assistance system aims to meet the dynamic needs of the automotive industry and contribute to the broader goal of road safety and accident prevention. The system is implemented on a Raspberry Pi, processing real-time data from LIDAR for distance measurement and a camera for object detection. This enables smart surround-view and collision avoidance, significantly enhancing road safety.

REFERENCES

[1] Lee, H., Chae, H. and Yi, K. (2019) "A geometric model based 2D LIDAR/radar sensor fusion for tracking surrounding vehicles", *IFAC-PapersOnLine*, 52(8).

[2] Wang, B., Yan, X. and Li, D. (2022) "An end-to-end lane detection model with attention and residual block", *Computational Intelligence and Neuroscience*, 2022.

[3] Ramachandra, A.C. *et al.* (2022) 'Advanced Driving Assistance System for cars using Raspberry Pi', *Indian Journal Of Science And Technology*, 15(40).

[4] Liu, H., Wu, C. and Wang, H. (2023) 'Real time object detection using LIDAR and camera fusion for autonomous driving', *Scientific Reports*, 13(1).

[5] Ponn, T., Kröger, T. and Diermeyer, F. (2020) 'Identification and explanation of challenging conditions for camera-based object detection of Automated Vehicles',*Sensors*, 20(13), p. 3699.

[6] Wang, W., Lin, H., & Wang, J. (2020). "CNN based lane detection with instance segmentation in edge-cloud computing". *Journal of Cloud Computing*, *9*(1)

[7] Haghbayan, Hashem, et al., "An Efficient Multi-sensor Fusion Approach for Object Detection in Maritime Environments," *Advanced Science and Technology Letters*, ISSN: 2287-1233, Vol. 66.

[8] Lee, J. and Hwang, K. (2021) 'Yolo with adaptive frame control for real-time object detection applications', *Multimedia Tools and Applications,* 81(25), pp. 36375–36396.

[9] Parekh, D. et al. (2022) 'A review on Autonomous Vehicles: Progress, methods and challenges', *Electronics*, 11(14), p. 2162.

[10] Dang, X., Rong, Z. and Liang, X. (2021) 'Sensor fusion-based approach to eliminating moving objects for slam in Dynamic Environments',*Sensors*, 21(1), p. 230.

[11] [2] A. Miglani and N. Kumar, "Deep learning models for traffic flow prediction in Autonomous Vehicles: A review, solutions, and challenges,"*Vehicular Communications*, vol. 20, p. 100184, Dec. 2019.

[12] Meyer, G.P.*et al.* (2019) 'LaserNet: An efficient probabilistic 3D object detector for autonomous driving', 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

[13] Korium, M.S., Abdelsalam, A.A. and Perepelkina, S.Yu. (2021) 'Lane detection algorithm', *Izvestiâ vysših učebnyh zavedenij. Priborostroenie,* .

[14] Dai, X. et al. (2020) 'Lane detection combining details and integrity: An advanced method for Lane Detection',2020 3rd International Conference on Algorithms, *Computing and Artificial Intelligence*

[15] Kanchana, B. *et al.* (2021) 'Computer Vision for autonomous driving', 2021 3rd International Conference on Advancements in Computing (ICAC)

b. Plagiarism Report

Revolutionizing Road Safety: Raspberry Pi based driver assistance system for Indian Roads using LIDAR and Camera Sensors

ORIGINALITY REPORT

9%	8%	7%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

Exclude quotes	Off	Exclude matches	< 1%
Exclude bibliography	Off		

c. Project Review Sheets

Review 1:

Inhouse/ Industry_Innovation/Research:

Sustainable Goal: 9

Project Evaluation Sheet 2023 - 24

Class: D17 A/B/C

Group No.: 41

Title of Project: Multi-Sensor fusion for enhanced object detection in complex environment

Group Members: Aaryan Manawat, Kaustubha Sahu, Abhiman Singh, Pranav Ubharhande

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
5	5	4	3	5	2	2	2	2	2	3	3	3	3	1	45

Comments: work on lag time reduction

Dr. Nupur Giri

Name & Signature Reviewer1

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
5	5	4	3	5	2	2	2	2	2	3	3	3	3	1	45

Comments: work on lane change.

Date: 10th february, 2024

Sanjay Nischandani

Name & Signature Reviewer 2

Review 2 :

Inhouse/ Industry
Innovation/Resear

Class: D17 A/B/C

Group No.: 41

Project Evaluation Sheet 2023 - 24

Title of Project: Multisensor Fusion for enhanced object detection in complex environment

Group Members: Kaustubha Sahu, Abhiman Singh, Aaryan Manawat, Pranav Ubarhande

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
4	4	4	2	5	2	2	2	2	2	3	3	3	3	4	45

Comments: work on Research Paper - Refining required.

Dr. Nupur Giri

Name & Signature Reviewer1

Engineering Concepts & Knowledge (5)	Interpretation of Problem & Analysis (5)	Design / Prototype (5)	Interpretation of Data & Dataset (3)	Modern Tool Usage (5)	Societal Benefit, Safety Consideration (2)	Environment Friendly (2)	Ethics (2)	Team work (2)	Presentation Skills (2)	Applied Engg&Mgmt principles (3)	Life - long learning (3)	Professional Skills (3)	Innovative Approach (3)	Research Paper (5)	Total Marks (50)
4	4	4	2	5	2	2	2	2	2	3	3	3	3	4	45

Comments: work is satisfactory.

Date: 9th March, 2024

Sanjay M.

Name & Signature Reviewer 2