Misr University for Science and Technology

College of Engineering

Electronics and Communications Engineering Department

B. Sc. Final Year Project

Advanced Driver Assistance System (ADAS)

Project Team

Name:	ID:
Hatem Abdelhamid Saleh	91145
Haneen Salah Eid	85555
Ahmed Mohamed Fathy	95960
Abdulrhman Amr Mohamed	86519
Abdelrahman Mohamed Abdelrahman	91550
Youssef Mohamed Elmohamady	91293
Omar Hosny Mohamed	75002
Faysal Mohamed Hassan	95887
Toka Hamdy Sayed	91192
Mahmoud Mohamed Abd-El Mohsen	91350

Supervisor

Dr. Waleed Abd-Elshafi Ali Elshazly

ACKNOWLEDGMENT

First and foremost, we thank ***Allah***, the most gracious and most merciful for enabling us to finish this project. We are indebted to our supervisor, **Dr. Waleed Abd-Elshafi Ali Elshazly** for providing us the opportunity to do the project under his guidance and for his invaluable suggestions, patience, and continuous encouragement throughout our project work, without which it would not have been possible to execute our project.

ABSTRACT

Advanced Driver Assistance Systems (ADAS) represent a pivotal advancement in automotive technology, aiming to enhance vehicle safety and overall driving experience. This abstract explores the key components, functionalities, and benefits of ADAS. ADAS integrates a multitude of sensors, cameras, and radar to contribute to the overall goal of reducing accidents, minimizing human errors, and improving road safety. The benefits of ADAS extend beyond safety improvements. By providing drivers with real-time information and assistance, ADAS aims to reduce the cognitive load on drivers, enhancing overall comfort and reducing driver fatigue. Additionally, as these systems continue to evolve, they play a crucial role in laying the foundation for autonomous driving technologies.

Our Project is specifically handling the following systems: Bump Detection, Precise Lane Tracking, Road Sign Interpretation, and Interactive Surface. Bump Detection, a crucial feature of ADAS, utilizes sensitive sensors, including accelerometers and gyroscopes, to detect irregularities and bumps on the road surface. These sensors measure sudden changes in acceleration and vehicle orientation, identifying potential road hazards such as potholes or uneven surfaces. Upon detecting a bump, the system assesses its severity and triggers appropriate responses to mitigate discomfort for passengers and reduce wear on the vehicle. The system can adjust suspension settings or provide real-time feedback to the driver, enhancing overall ride comfort and vehicle durability.

Precise Lane Tracking, an integral component of ADAS, utilizes advanced computer vision algorithms and sensors to monitor the vehicle's position within the lane. Cameras and sensor technologies assess lane markings and surrounding vehicles, ensuring accurate tracking even in complex road conditions. By continuously analyzing this data in real-time, the system provides feedback to the driver about the vehicle's position within the lane. If there is any deviation from the designated lane, visual and auditory alerts are generated, enhancing the driver's awareness of their driving behavior. These alerts are promptly displayed on the interactive surface Touch Screen, enabling immediate corrective action. Road Sign Interpretation, a key capability of ADAS, relies on sophisticated image processing techniques and Machine Learning (ML) algorithms. Through cameras and ML

models, the system comprehensively scans road signs, including speed limits, stop signs, and directional markers.

The system's image recognition technology deciphers these signs in real time, extracting essential information to alert the driver promptly. When a relevant road sign is detected, the information is displayed on the interactive surface, ensuring the driver is informed about the current road conditions and regulations. The interactive surface with a graphical user interface (GUI) represents the pinnacle of user interaction within ADAS. Combining advanced projection technology with intuitive GUI design, this feature projects vital information directly onto the windshield. Drivers receive real-time feedback on ADAS operations, including obstacle detection, road sign interpretation, and navigation alerts. The touch-sensitive interface allows customization of system settings, immediate response to alerts, and seamless control over ADAS functionalities.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

α	The threshold value	
$\Delta\varphi$	The phase difference	
μ_B	mini-batch mean	
$\sigma^2{}_B$	mini-batch variance	
AP	Average Precision	
C	Speed of light (299 792 458 m/s)	m/s
C_s	Speed of sound (343 m/s)	m/s
D	Distance	m
f	Frequency	HZ
FN	False negative	
FP	False positive	
mAP	Mean average precision	
p	The precision	
q	class	
Q	the number of classes	
r	The recall	
t_{LC}	Time to line crossing	
TN	True negative	
TP	True positive	Sec
V_Y	Rate of departure for a vehicle traveling straight	
$\hat{x}_i$	normalize	m/s
y_i	scale and shift	

LIST OF ACRONYMS/ABBREVIATIONS

ACC	Adaptive Cruise Control
ADAS	Advanced Driver Assistance Systems
AI	Artificial intelligence
BSD	Blind Spot Detection
CNNS	Convolutional neural networks
DLC	Distance to line Crossing
GPIO	General Purpose Input/Output
GPU	Graphic processing unit
HID	Human Interface Device
HOG	Histogram of Oriented Gradients
I^2C	Inter-Integrated Circuit
LBP	Local Binary Patterns
LCA	Lane Centering Assist
LDWS	Lane Departure Warning System
LiDAR	Light Detection and Ranging
LKA	Lane Keeping Assist
MCU	Microcontroller unit
ML	Machine learning
MS COCO	Microsoft Common Objects in Context
PWM	Puls Width Modulation
RCNNS	Region with convolutional neural networks
RDM	Road Departure Mitigation
RELU	Rectified linear unit
RESNET	Residual network
SoC	System on Chip
SSD	Single Shot Detector
SPPNET	Spatial pyramid pooling network
TPU	Tensor processing unit
TTLC	Time to Line Crossing

UART	Universal Asynchronous Receiver-Transmitter
VGGNET	Visual geometry group
WHO	World Health Organization
YOLO	You only look once

Chapter (1)

1. INTRODUCTION

1.1 BACKGROUND

With the rapid advancement of technology, the automotive industry has witnessed a significant transformation in recent years. One of the most notable developments in this field is the introduction of Advanced Driver Assistance Systems (ADAS). ADAS refers to a collection of technologies designed to assist drivers in various aspects of their journey, enhancing safety, comfort, and overall driving experience. These systems leverage the power of sensors, cameras, radar, and artificial intelligence (AI) to monitor the vehicle's surroundings, detect potential hazards, and provide timely warnings or interventions to prevent accidents [1].

The primary objective of ADAS is to reduce the likelihood of human error, which is a major contributing factor in most road accidents. According to the World Health Organization (WHO), road traffic injuries are one of the leading causes of death worldwide, and human error plays a significant role in approximately 90% of these accidents. ADAS aims to address this issue by complementing the driver's skills and alertness, providing an additional layer of safety and assistance [1].

ADAS encompasses a wide range of features and functionalities, each serving a specific purpose and addressing different aspects of driving. Some of the most common and well-known **ADAS features include:**

- **Blind Spot Detection (BSD):**

BSD systems use sensors or cameras to monitor the vehicle's blind spots, which are areas that are not directly visible to the driver through the side mirrors or the rearview mirror. When another vehicle enters the blind spot, BSD alerts the driver through visual or auditory warnings, reducing the risk of collisions during lane changes or maneuvers.

- **Lane Departure Warning System (LDW):**

LDW systems use cameras or sensors to monitor the vehicle's position within the lane markings on the road. If the system detects an unintended departure from the lane without the use of turn signals, it alerts the driver through visual, auditory, or haptic feedback. LKA systems work in conjunction with LDW by actively steering the vehicle back into the lane if the driver fails to take corrective action. These features help prevent accidents caused by lane drifting, driver inattention, or drowsiness.

- **Adaptive Cruise Control (ACC):**

ACC is an advanced version of traditional cruise control that automatically adjusts the vehicle's speed to maintain a safe following distance from the vehicle ahead. Using sensors, radar, or lidar, ACC continuously monitors the distance and speed of the lead vehicle, and if necessary, automatically applies the brakes or accelerates to maintain a predetermined gap. This technology enhances driving comfort and reduces the risk of rear-end collisions, particularly in congested traffic conditions.

- **Traffic Sign Recognition (TSR):**

TSR systems utilize cameras and image processing algorithms to identify and interpret traffic signs, such as speed limits, stop signs, and no-entry signs. Once a traffic sign is recognized, the system can provide the driver with relevant information, such as the current speed limit or the presence of a school zone. TSR assists drivers in staying informed about changing road conditions, reducing the possibility of violations, and improving overall road safety.

- **Bump Detection:**

Is an emerging feature in ADAS that focuses on protecting the vehicle and its occupants from potential damage caused by low-speed impacts. Using a combination of sensors and cameras, the system is designed to detect objects or obstructions near the vehicle, such as walls, curbs, or other vehicles. When an imminent collision is detected, the system may activate specific safety measures, such as applying the brakes or providing an audible warning to the driver. Bump Detection systems offer an added layer of protection, particularly during parking maneuvers or in tight spaces where visibility may be limited.

The implementation of ADAS has the potential to transform the driving experience by reducing accidents, injuries, and fatalities. These systems act as a co-pilot, offering valuable assistance to drivers in real-time and helping them make informed decisions. However, it is important to note that ADAS should not be considered a substitute for attentive and responsible driving. Drivers should always remain engaged and vigilant, using ADAS features as aids rather than relying solely on them.

1.2 THE BENEFITS OF ADAS

ADAS technology provides several significant benefits, both for individual drivers and society. Some of the key benefits include:

1) Enhanced Safety:

ADAS features are designed to prevent accidents and minimize the severity of collisions. By continuously monitoring the vehicle's surroundings and providing real-time warnings or interventions, ADAS systems help drivers avoid potential hazards and mitigate risks. This technology is particularly effective in addressing human errors, such as distracted driving, drowsiness, and failure to maintain a safe following distance.

2) Reduced Fatalities and Injuries:

ADAS has the potential to significantly reduce the number of fatalities and injuries resulting from road traffic accidents. By alerting drivers to potential dangers and assisting them in taking appropriate actions, ADAS systems can prevent collisions or mitigate their severity. This can lead to a significant reduction in the loss of life and the physical and emotional trauma associated with accidents.

3) Improved Driving Comfort and Convenience:

ADAS features can enhance driving comfort and convenience by automating certain tasks and reducing driver workload. For example, adaptive cruise control and lane keep assist systems can alleviate the stress of maintaining a constant speed and staying within lanes, especially during long journeys or in heavy traffic. Parking assistance systems simplify the process of parking in tight spaces, reducing the frustration and anxiety often associated with parking maneuvers.

4) Increased Efficiency:

ADAS technology can contribute to improved fuel efficiency and reduced traffic congestion. Features like adaptive cruise control optimize the vehicle's speed and acceleration, promoting smoother and more efficient driving. Traffic sign recognition systems can help drivers adhere to speed limits and other traffic regulations, leading to a more consistent flow of traffic and reduced congestion.

5) Environmental Benefits:

The increased efficiency resulting from ADAS usage can also have positive environmental impacts. By optimizing driving behaviors and reducing unnecessary accelerations or decelerations, ADAS systems can contribute to lower fuel consumption and emissions. This can help mitigate the environmental impact of transportation and support sustainability efforts.

6) Potential for Autonomous Driving:

ADAS technology serves as a crucial steppingstone toward the development and deployment of fully autonomous vehicles. The sensors, cameras, and AI algorithms used in ADAS systems are key components of autonomous driving systems. As ADAS technology continues to advance, it brings us closer to the realization of a future where vehicles can operate autonomously, eliminating the need for human drivers altogether.

1.3 CHALLENGES AND CONSIDERATIONS

While ADAS offers numerous benefits, there are also challenges and considerations that need to be addressed:

1) System Limitations:

ADAS systems have certain limitations that drivers should be aware of. Environmental factors, such as adverse weather conditions or poor road markings, can affect the performance of these systems. Additionally, ADAS features may not be able to detect all potential hazards or anticipate unpredictable human behavior. Therefore, drivers should never rely solely on ADAS and should remain vigilant and actively engaged in the driving process.

2) Driver Understanding and Trust:

ADAS technology is complex, and drivers need to have a clear understanding of its capabilities and limitations to use it effectively. Education and training programs are crucial to ensure that drivers are aware of how ADAS features operate and when and how to intervene if necessary. Building trust in technology is also essential, as drivers need to have confidence in the system's ability to perform reliably.

3) Data Privacy and Security:

ADAS systems rely on collecting and processing vast amounts of data, including personal information about the driver and their surroundings. Ensuring the privacy and security of this data is of utmost importance. Manufacturers and policymakers must establish robust data protection measures and cybersecurity protocols to safeguard against potential risks, such as data breaches or unauthorized access.

4) Regulatory Framework:

The rapid advancement of ADAS technology poses challenges for regulatory bodies that need to develop appropriate standards and regulations. Establishing a comprehensive and harmonized regulatory framework is vital to ensure the safety, reliability, and interoperability of ADAS systems across different manufacturers and vehicle models.

1.4 PROBLEM STATEMENT

`A driver is one of the “best sensors in the vehicle” and is the main responsible for avoiding crashes. But still, a large proportion of crashes are attributed to driver errors. The development and deployment of new in-vehicle technologies to counteract these driver errors and hence to support the driver in preventing crashes is ongoing. ADAS are a group of vehicle technologies that warn drivers timely regarding risky or hazardous situations to avoid crashes.

1.5 PROJECT OBJECTIVE

`ADAS project aims to introduce comfort functions and safety functions. The comfort functions aim to warn the driver by triggering a warning, like a flashing light, sound, vibration, or even a gentle steering suggestion. Safety functions aim to act on the vehicle itself in cases where the driver is not responding to a potentially dangerous situation.

1.6 METHODOLOGY

To achieve the main objective of this project, the following methodology is adopted:

- A literature review of Advanced Driver Assistance Systems Concept.
- Design of Bump Detection system.
- Design of Road Sign Interpretation system.
- Design of Lane Departure Warning system. `
- Design of the interactive surface with a graphical user interface system.
- Identifying equipment and components required.
- Comparing the Different types of available microcontrollers to select the most appropriate.
- Put Project Tasks Timeline.
- Ordering components.
- Setting up Nvidia Jetson Nano Developer Kit and microcontrollers.
- Programming Nvidia Jetson Nano Developer Kit and microcontrollers.
- Connecting complete system.
- Connecting the charging circuit.
- Debugging Nvidia Jetson Nano Developer Kit and integrating Parts together.
- Testing integrated unit.
- Producing Final Prototype.

1.7 PROJECT ORGANIZATION

The rest of project is organized as the following:

- Chapter 2 describes the car movement of ADAS project, System implementation, and Integration of ADAS Functionality.
- Chapter 3 describes the blind spot detection, placement and positioning of sensors, design and working principle, and functional, architectural view.
- Chapter 4 introduces the lane departure system, lane departure warning function, algorithm overview, and advantages and disadvantages.
- Chapter 5 introduces Adaptive Cruise Control system, working principle, controller action, advantages and disadvantages, and future scope.

- Chapter 6 describes traffic sign interpretation, object detection, models for traffic sign recognition, and YOLO models.
- Chapter 7 describes bump detection, problem statement, system design, and image recognition.
- Chapter 8 concludes the work, summarizes the main achievements, and outlines possible future work.

Chapter (2)

2. CAR MOVEMENT IN ADAS

2.1 INTRODUCTION:

In the realm of ADAS, the movement of the vehicle is a fundamental aspect that intertwines with various components and functionalities. This chapter delves into the intricacies of car movement in the context of ADAS project, focusing on its components and operational mechanisms. Furthermore, it sheds light on the integration of ADAS functionalities such as bump detection, TSR, LDW, ACC, and BSD within the project framework.

2.2 CAR MOVEMENT:

The movement of the car is orchestrated by a confluence of mechanical, electrical, and software systems working in harmony. At the heart of the car movement lie the following key components:

2.2.1 Chassis and Drivetrain:

The chassis serves as the foundation upon which the various components are mounted. It provides structural integrity and support for the car. The drivetrain, comprising the motor, gearbox, and wheels, translates electrical energy into mechanical motion as shown in Figure 2-1. In our project, a compact yet efficient drivetrain system is essential to ensure smooth and controlled movement. The chassis is designed to be able to handle the weight of the components. It has a rear wheel drive, and the front wheels are moving right and left for steering to act like real car movement as shown in Figure 2-2.

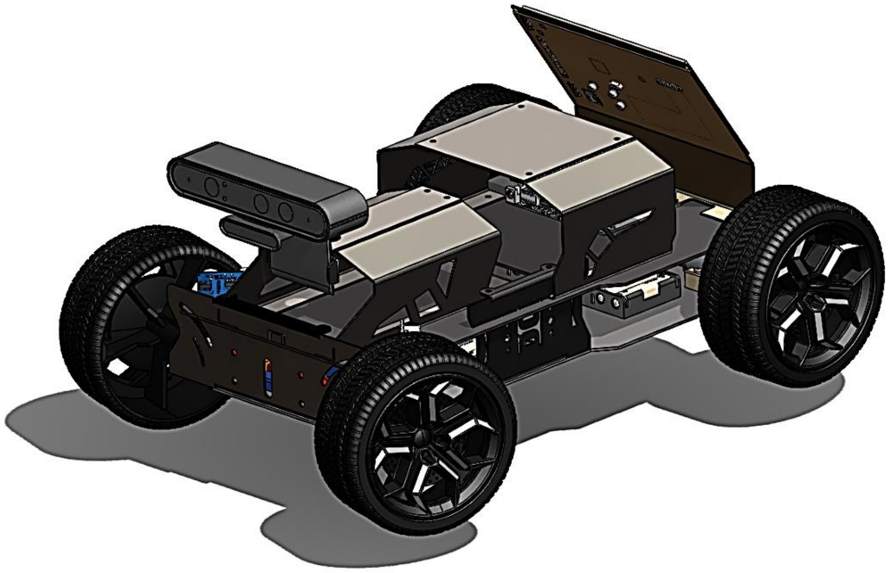


Figure 2- 1: ADAS Chassis.

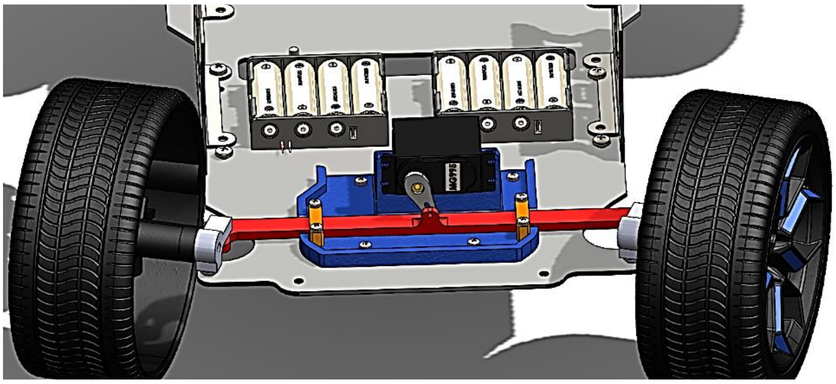


Figure 2- 2: Steering System.

2.2.2 Motor Control System:

The motor control system governs the operation of the motors responsible for propelling the car. It regulates motor speed, direction, and torque, thereby dictating the car's movement dynamics. This system must be adept at responding to real-time input signals from sensors and control algorithms to enable precise and agile maneuverability.

There are two types of motors in our project. DC motor to drive the car forward and backward of rear wheels that acts like engine in real car. Servo motor to steer the front wheels. These motors are chosen wisely to handle the weight of the chassis.

- <u>DC Motor:</u>

The DC Motor that we chose has torque of 8.8kg/cm and 250 rpm. Because of the weight of the car, this torque is more than enough for own needs. Its operating voltage is 12V and the no load current is 250mA. The maximum current that can operate with full load is 6.5A.

Figure 2- 3: DC Motor.

- <u>Servo Motor:</u>

The Servo Motor has torque of 8.5kg/cm and speed of 0.2s/60°. That torque is enough to make smooth steering for the front wheels. Its operating voltage is 5V as the DC motor that we are using.

Figure 2- 4: Servo Motor.

2.2.3 ESP32 MCU:

The brain that controls all the movement is ESP32 microcontroller. The reason for choosing this MCU is that it has over 20 PWM (Pulse Width Modulation) pins, Wi-Fi, Bluetooth connectivity, easy to use, and its cost is very low. The ESP32 acts as slave for the

Master (Nvidia Jetson Nano MCU). The Jetson Nano will give the ESP32 signals of any alarm system in our project, and the ESP32 must take a propriate action to avoid the cushion.

Controlling the car:

Taking advantage of Bluetooth connection in the ESP32, we chose to use PS4 Controller to control the car. Another advantage of using PS4 Controller is that it has PWM buttons that can be used to make fine driving experience. Furthermore, the PS4 Controller has rump motors inside it to give various feedback, and LED panel in front of the controller. We can use these to give all our system's alarm into the PS4 Controller.

Power Supply:

A reliable power supply is imperative to sustain the operation of the car's electrical components. In our project, a compact yet robust power source such as a rechargeable battery pack is employed to energize the motor control system and other ADAS functionalities. Two power suppliers must apply to our project. One to drive the motors and the other to handle the MCU's and other components.

2.3 SYSTEM IMPLEMENTATION:

- **Circuit Diagram:**

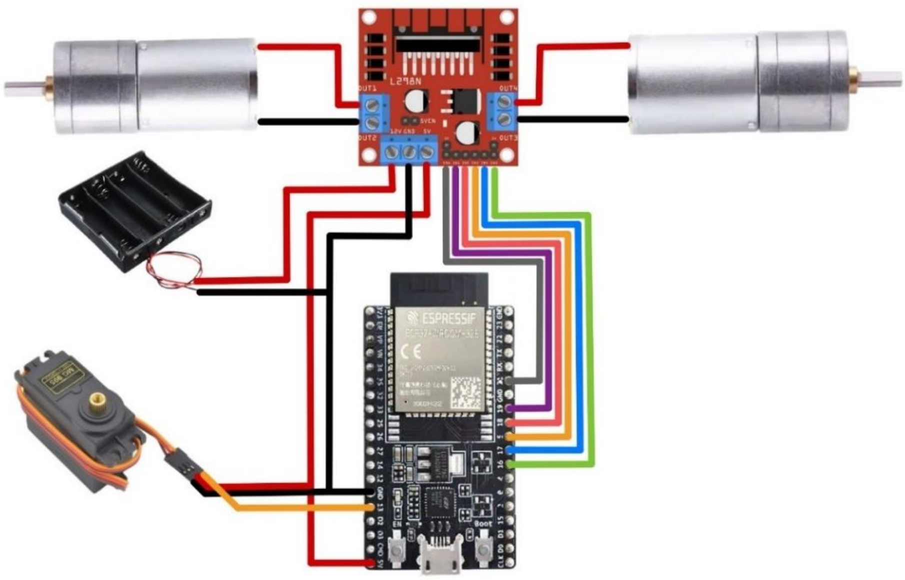


Figure 2- 5: Circuit Diagram of Motors System.

2.4 INTEGRATION OF ADAS FUNCTIONALITY:

The ADAS project transcends mere locomotion; it embodies the essence of innovation by incorporating cutting-edge ADAS functionalities. These systems are designed to enhance safety, efficiency, and convenience for the driver. Let us explore how each ADAS component integrates seamlessly into the project:

- **Bump Detection System:**
  Utilizing depth camera strategically positioned around the car's perimeter, the bump detection system detects obstacles or impediments in the car's path. Upon detection, the system triggers appropriate corrective actions to avoid collisions or mitigate the impact.
- **Traffic Sign Recognition:**
  Equipped with a camera module and image processing algorithms, the traffic sign recognition system identifies and interprets traffic signs such as speed limits, stop signs, and directional indicators. It relays pertinent information to the driver, fostering situational awareness and compliance with traffic regulations.
- **Lane Departure Warning:**

  Leveraging vision-based technologies, the lane departure warning system monitors the car's position within lane markings. In the event of unintentional lane departure, the system alerts the driver through visual or auditory cues, thereby averting potential accidents caused by lane drift.
- **Adaptive Cruise Control (ACC):**

  ACC employs Depth camera to monitor the distance between the car and preceding vehicles. By autonomously adjusting the car's speed and following distance, ACC ensures a safe and consistent driving experience, particularly in congested traffic conditions.
- **Blind Spot System:**

  Mounted on the side mirrors or rear bumper, blind spot monitoring sensors detect vehicles or objects within the car's blind spots. The system alerts the driver to exercise caution during lane changes or maneuvers, and take action if necessary, mitigating the risk of collisions.

Chapter (3)

3. BLIND SPOT DETECTION SYSTEM

3.1 INTRODUCTION:

The blind spot detection (BSD) system is a revolutionary advancement in automotive safety technology that aims to enhance driver awareness and prevent accidents caused by limited visibility in blind spot areas. Blind spots, which are areas around a vehicle that cannot be directly observed by the driver through the rearview or side mirrors, pose a significant risk when changing lanes or making maneuvers on the road. The BSD system utilizes sophisticated sensors and intelligent algorithms to monitor these blind spot areas continuously and provide real-time feedback to the driver.

The heart of the BSD system lies in its sensor technology, which is designed to detect the presence of vehicles or objects in the blind spot zones. Various sensor technologies are employed, including radar, ultrasonic waves, and cameras. Radar-based sensors emit radio waves that bounce off nearby objects, allowing the system to measure their distance and speed accurately. Ultrasonic sensors emit high-frequency sound waves and analyze their reflections to detect objects in blind spot areas. Cameras capture real-time video footage of the surrounding environment, enabling computer vision algorithms to identify vehicles or objects in blind spots.

When the BSD system detects a vehicle in the blind spot, it triggers timely alerts to the driver, effectively mitigating potential collision risks. These alerts can be in the form of visual indicators, such as flashing lights or icons on the side mirrors, auditory cues like beeps or chimes, or haptic feedback, such as vibrations in the steering wheel or seat. By providing multi-modal alerts, the system ensures that drivers are promptly notified of the presence of vehicles in their blind spots, enabling them to make informed decisions and take appropriate actions.

integration of BSD systems with other safety features further enhances their effectiveness. For example, some systems can work in conjunction with LDW systems or adaptive cruise control, enabling a comprehensive view of the vehicle's surroundings and facilitating coordinated responses to potential hazards. This integration creates a synergistic effect, promoting a safer driving experience and reducing the likelihood of accidents caused by blind spots.

While BSD systems offer numerous benefits, it is important to be aware of their limitations. Adverse weather conditions, such as heavy rain or fog, can impact the performance of these systems by reducing the accuracy of the sensors. Drivers should also remember that BSD systems are meant to complement, not replace, traditional mirror usage and visual checks. Maintaining attentiveness and using the system as an additional aid rather than relying solely on it is crucial for safe driving [1].

3.2 OBJECTIVES:

- **Main Objectives:**

The primary goal of this project was to create a BSD system for cars that can identify objects in the blind spot areas and notify the driver to avoid accidents on the road.

- **Specific Objectives:**

The specific objectives of this project were as follows:

1. To define the necessary hardware and software requirements for the Blindspot Detection System.
2. To create logical and physical design models for the System that outline its structure and functionality.
3. To develop a practical prototype of the System by implementing the design specifications [1].

3.3 PLACEMENT AND POSITIONING OF SENSORS:

The placement and positioning of sensors play a crucial role in ensuring optimal coverage and effectiveness of BSD systems as Figure 3-1. The strategic placement of sensors allows for accurate detection of vehicles or objects in blind spot areas, providing timely alerts to drivers and enhancing overall safety on the road. Here are some key considerations for the placement and positioning of sensors in BSD systems [2]:

- **Side Mirrors:**

Many BSD systems utilize sensors integrated into or around the side mirrors. These sensors are positioned to cover the blind spot areas adjacent to the vehicle. Sensors in the side mirrors are typically angled to provide a wide field of view, extending beyond the driver's peripheral vision. The placement of sensors inside mirrors ensures that the system can effectively detect vehicles approaching from the rear and side of the vehicle.

- **Calibration and Alignment:**

Proper calibration and alignment of sensors are essential to ensure accurate detection and reliable performance. Manufacturers provide specific guidelines for sensor calibration and alignment during installation and maintenance. Regular inspection and adjustment of sensor positions may be necessary to maintain optimal coverage and functionality. It's important to note that the placement and positioning of sensors can vary among different vehicle models and manufacturers. Each system is designed to provide the best possible coverage based on the specific vehicle design and intended functionality [2].

Optimal coverage in BSD systems is achieved by carefully considering the blind spot areas, integrating multiple sensor technologies, and ensuring proper calibration and alignment. By strategically placing sensors inside mirrors, bumpers, rear taillights, BSD systems can effectively monitor and provide alerts for potential hazards in blind spot zones, enhancing driver awareness and improving overall safety on the road [2].

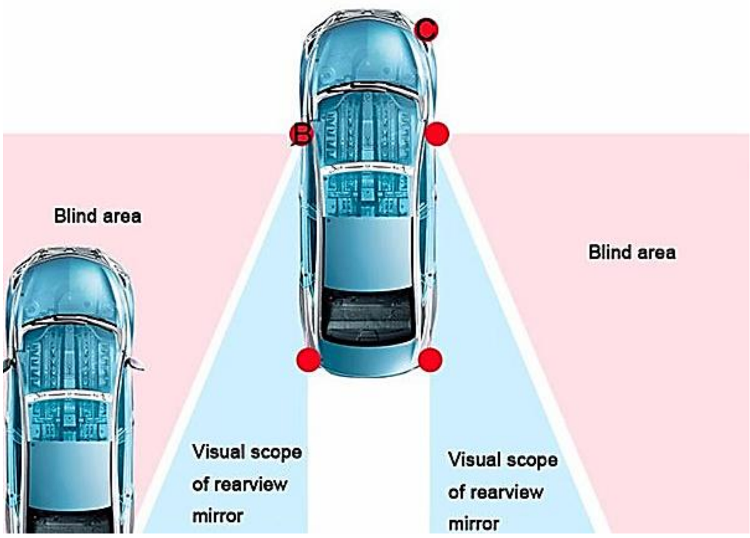


Figure 3- 1: Positioning of Sensors [2].

3.4 DESIGN AND WORKING PRINCIPLE:

The design and working of a BSD system involves various components and processes that work together to detect and alert drivers about objects or vehicles in their blind spot zones. Here's an overview of the typical design and working of a BSD system:

3.4.1 Design Components:

- **Sensors:** BSD systems utilize different types of sensors to monitor the areas around the vehicle. These sensors can include radar, ultrasonic sensors, or cameras. Radar sensors use radio waves to detect objects, ultrasonic sensors emit sound waves, and cameras capture visual information.
- **Data Processing Unit:** The data processing unit is responsible for receiving and analyzing the sensor data. It uses advanced algorithms to process the incoming signals or images and determine the presence and position of objects in the blind spot areas.
- **Alert Generation System:** Once an object is detected in the blind spot, the alert generation system generates warnings or alerts to notify the driver. This system can include visual indicators, auditory cues, or haptic feedback mechanisms.

3.4.2 Working Process:

1. **Sensor Monitoring:**

The BSD system continuously monitors the areas adjacent to the vehicle using the sensors placed strategically around it. These sensors provide real-time information about objects or vehicles present in the blind spot zones.

2. **Object Detection:**

The sensors detect objects within the blind spot areas by emitting signals (radio waves, sound waves, or capturing visual data) and analyzing the returning signals or images. The data processing unit analyzes this information to identify and classify objects.

3. **Object Position Determination:**

The system determines the position, speed, and trajectory of the detected objects relative to the vehicle's position and movement. This information helps assess the potential risk of a collision.

4. **Alert Generation:**

If an object is identified in the blind spot and poses a potential risk, the alert generation system is triggered. It generates alerts to notify the driver about the presence of a vehicle or object in the blind spot.

- **Visual Alerts:**

  Visual alerts can be displayed on the side mirrors, dashboard, or heads-up display. They often involve icons or symbols indicating the presence of a vehicle in the blind spot.

- **Auditory Alerts:**

  Auditory alerts can include beeps, chimes, or spoken messages to draw the driver's attention to the presence of a vehicle in the blind spot.

- **Haptic Alerts:**

  Haptic alerts may involve vibrations in the steering wheel, seat, or other parts of the vehicle, providing a tactile cue to the driver.

5. Driver Response:

Upon receiving the alerts, the driver can take appropriate action. They may delay a lane change, adjust their speed, or check the side mirrors and blind spot areas more carefully before proceeding with the maneuver. The alerts help improve driver awareness and decision-making, reducing the likelihood of blind spot-related accidents.

3.5 FUNCTIONAL ARCHITECTURAL VIEW:

The functional view of the BSD system defines the system's architecturally significant functional elements, the responsibilities of each, the interfaces they offer, and the dependencies between these elements. Figure 3-2 shows the functional architecture of the BSD system [1].

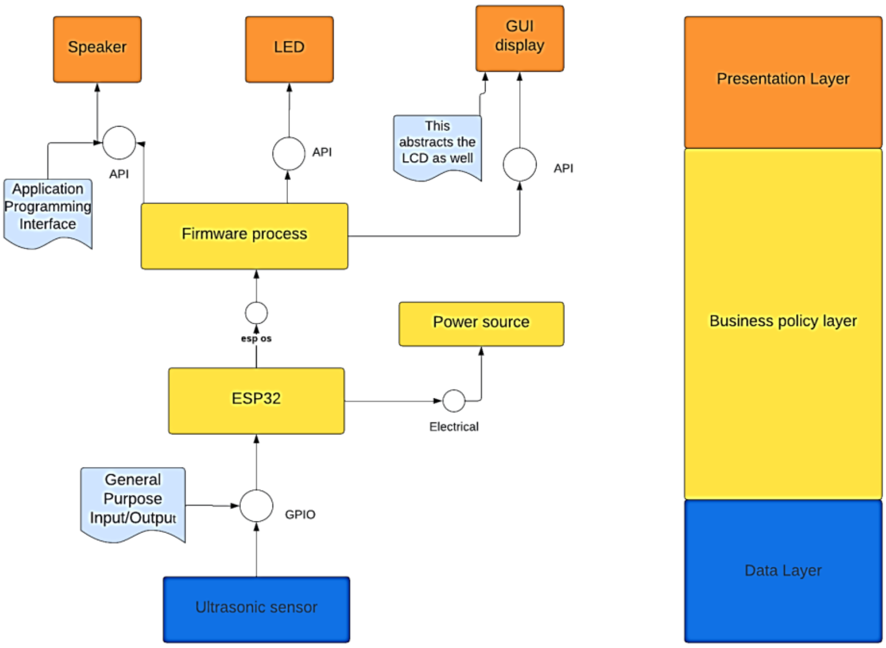


Figure 3- 2: Functional Architecture.

3.6 SYSTEM IMPLEMENTATION:

- **System design:**

the ultrasonic connected with ESP32 as shown in Figure 3-3, the connection in Table 3-1.

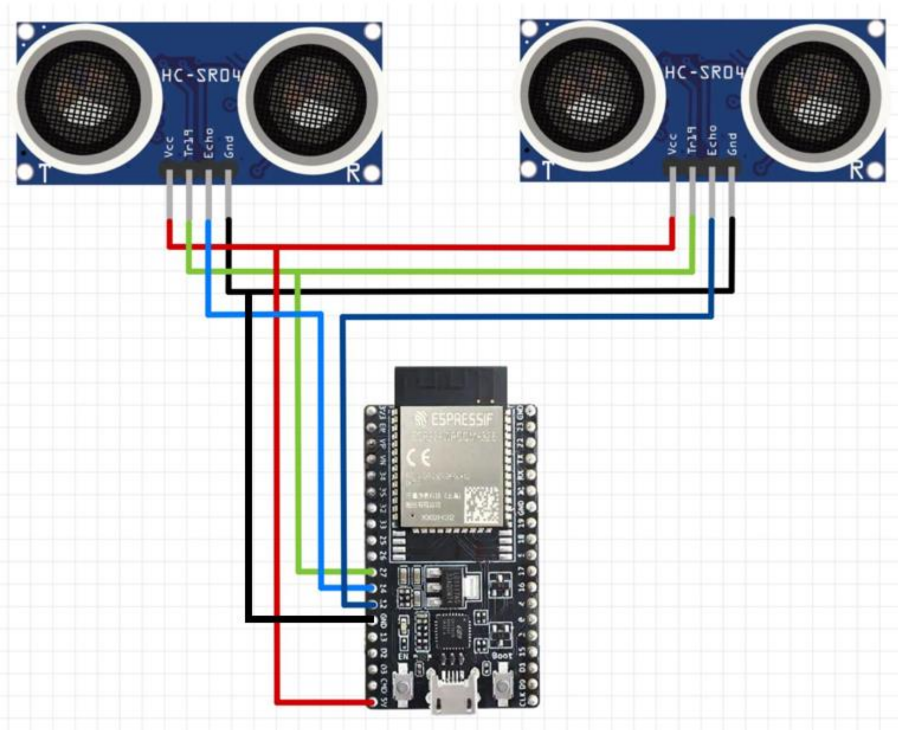


Figure 3- 3: The Circuit Diagram.

Table 3- 1: HC-SR04 device connection with ESP32.

ESP32	Right and Left Ultrasonic sensors
5 V	VCC (Right and Left Sensors)
GPIO 27	TRIG (Right and Left Sensors)
GPIO 12	ECHO (Right Sensor)
GPIO 14	ECHO (Left Sensor)
GND	GND (Right and Left Sensors)

- **System Block Diagram**:

Figure 3-4 shows the core algorithm of the system, and this at a high level describes how the system was implemented to achieve its mission. The description below gives how the operation flows:

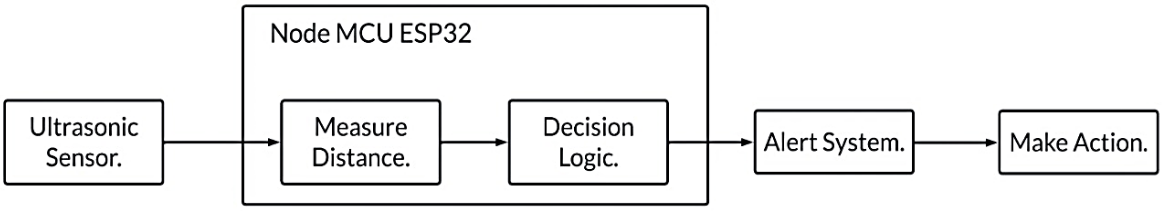


Figure 3- 4: The BSD Block Diagram.

1. The program starts at the system start-up of the ESP32 operating system.
2. The Ultrasonic measuring distance of left and Right sides of blind spot.
3. If the distance is less than 10 m.
4. The alert system makes high rate flashing with high vibration.
5. If the distance is greater than 10m and less than 15m.
6. The alert system makes low rate flashing with low vibration.

3.6.1 The implementation of Blind Spot Detection System:

The system generally has three categories of peripherals, and these are shown in Figure 3-5.

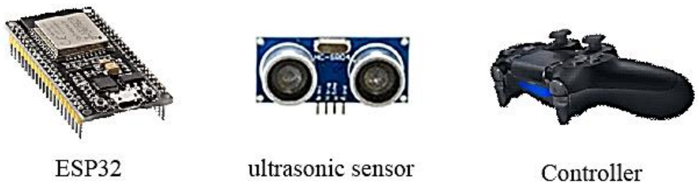


Figure 3- 5: Peripherals.

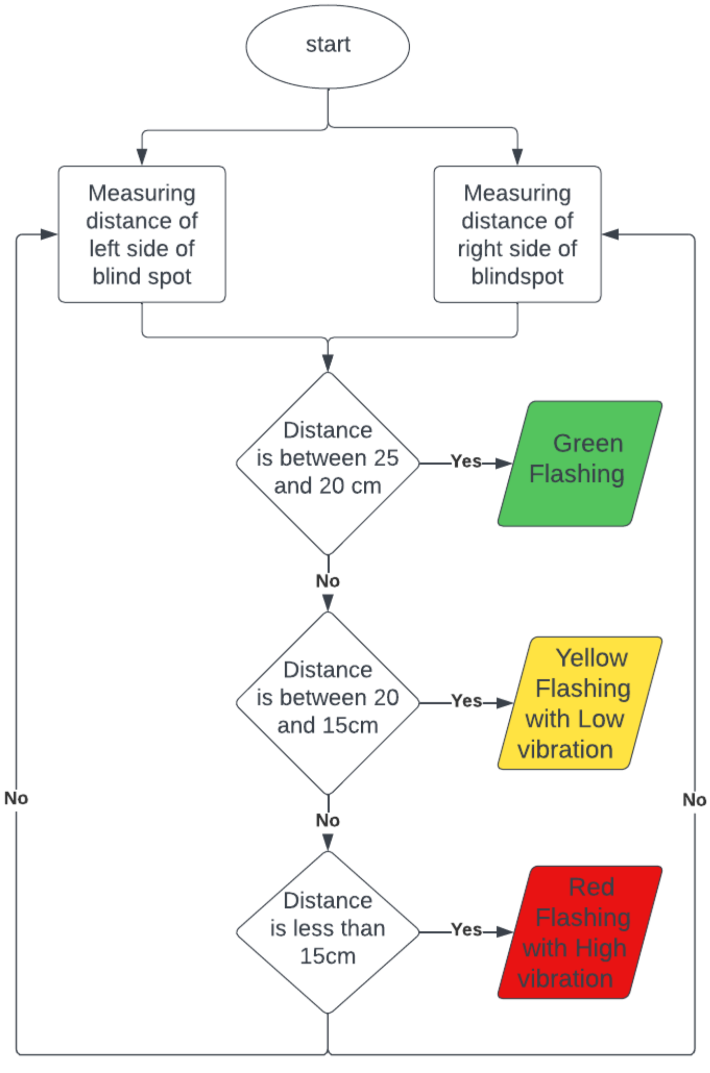


Figure 3- 6: The flowchart.

- **ESP32**:

ESP32 is a low-cost System on Chip (SoC) Microcontroller from Expressive Systems, the developers of the famous ESP8266 Sock It is a successor to ESP8266 SoC and comes in both single-core and dual-core variations of the Ten silica's 32-bit Extensa LX6 Microprocessor with integrated Wi-Fi and Bluetooth. The good thing about ESP32, like ESP8266, is its integrated RF components like Power Amplifier, Low-Noise Receive Amplifier, Antenna Switch, Filters, and RF Balun. This makes designing hardware around ESP32 very easy as you require very few external components. We selected this microcontroller because it has 16 pins for PWM [3].

- **Specifications of ESP32:**
  - Single or Dual-Core 32-bit LX6 Microprocessor with clock frequency up to 240 MHz.
  - 520 KB of SRAM, 448 KB of ROM and 16 KB of RTC SRAM.
  - Supports 802.11 b/g/n Wi-Fi connectivity with speeds up to 150 Mbps.
  - Support for both Classic Bluetooth v4.2 and BLE specifications.
  - 34 Programmable GPIOs.
  - Up to 18 channels of 12-bit SAR ADC and 2 channels of 8-bit DAC
  - Serial Connectivity includes 4 x SPI, 2 x I^2C, 2 x I^2S, 3 x UART.
  - Motor PWM and up to 16-channels of LED PWM.

- **Ultrasonic sensor:**

At its core, the HC-SR04 Ultrasonic distance sensor consists of two ultrasonic transducers. One acts as a transmitter that converts an electrical signal into 40KHz ultrasonic sound pulses. The receiver listens for the transmitted pulses. If it receives them, it produces an output pulse whose width can be used to determine the distance the pulse traveled. The sensor is small, easy to use in any robotics project, and offers excellent non-contact range detection between 2 cm to 400 with an accuracy of 3mm. Since it operates on 5 volts, it can be hooked directly to an ESP32 or any other 5V logic microcontrollers.

- **The theory of working:**

It all starts when a pulse of at least 10 µS (10 microseconds) in duration is applied to the Trigger pin. In response to that, the sensor transmits a sonic burst of eight pulses at 40 kHz. This 8-pulse pattern makes the "ultrasonic signature" from the device unique, allowing the receiver to differentiate the transmitted pattern from the ambient ultrasonic noise as shown in Figure 3-7.

The eight ultrasonic pulses travel through the air away from the transmitter. Meanwhile, the Echo pin goes HIGH to start forming the beginning of the echo-back signal. In case, if those pulses are not reflected then the Echo signal will timeout after 38 mS (38 milliseconds) and return low. Thus a 38 mS pulse indicates no obstruction within the range of the sensor. If those pulses are reflected the Echo pin goes low as soon as the signal is received. This produces a pulse whose width varies between 150 µS to 25 mS, depending upon the time it took for the signal to be received [1].

The width of the received pulse is then used to calculate the distance to the target object using the equation (3.1):

$$Distance = \frac{(speed \times time)}{2}, Speed = 343\ m/s \quad (3.1)$$

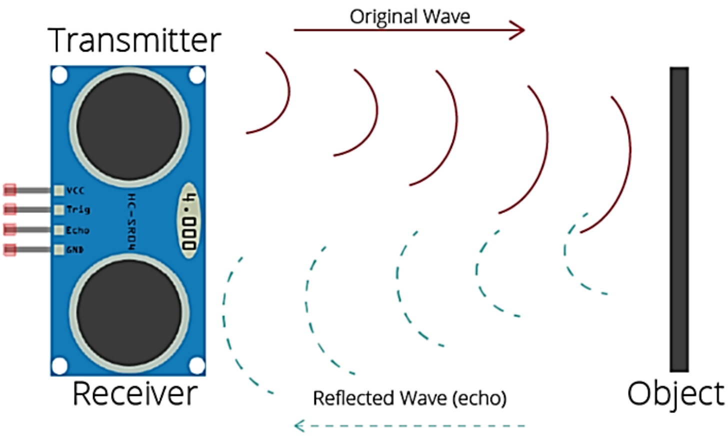


Figure 3- 7: The theory of working.

- **Controller:**

PS4 controller (DualShock) uses Bluetooth for communications. The protocol it uses in HID (Human Interface Device). The pairing of this controller is done by MAC address. The controller has two rumble motors to give vibration feedback. It has a three-axis accelerometer and a gyroscope for rotational movement detection.

Chapter (4)

4. LANE DEPARTURE SYSTEM

4.1 INTRODUCTION:

Lane departure refers to a driving scenario where a vehicle unintentionally deviates from its designated lane without the driver actively steering in that direction. This can occur due to various reasons, such as driver distraction, fatigue, impaired driving, or environmental factors. Lane departure incidents are a significant concern for road safety, as they can lead to accidents, collisions, and potentially serious injuries [4].

Lane departure warning is one important feature in ADAS, which aims to improve overall safety on the road. However, challenges such as inconsistent shadows and fading lane markings often plague the road surface and cause the lane detection system to produce false warnings. Users are aggravated by the warning and tend to disable this safety feature Figure 4-1 the systems described in this chapter aim at improving driving safety by preventing unintended lane departures. In addition, some systems aim at improving comfort by releasing the driver from monotonous tasks on highways and highway-like roads. The support of drivers at roadway sections having temporary or irregular lane markings (such as roadwork zones) is not within the scope of today's available systems [4].

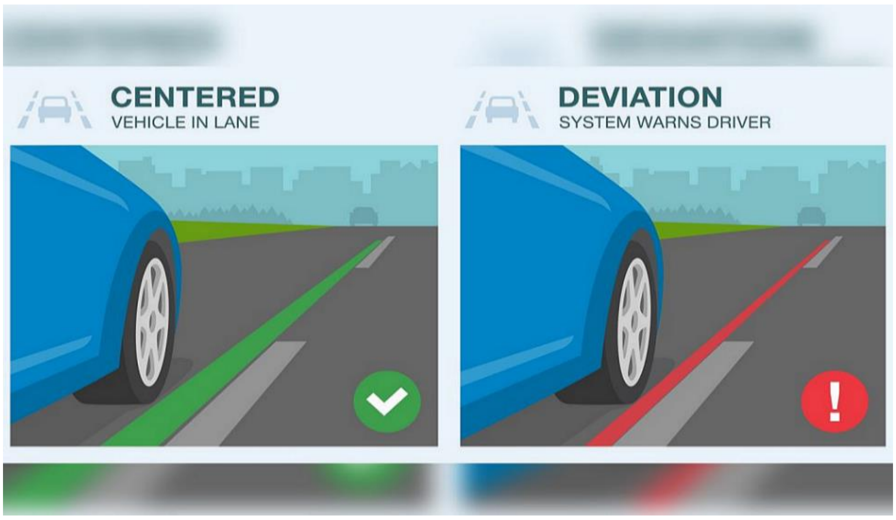


Figure 4- 1: Lane departure deviation from road [4].

4.2 DESCRIPTION:

Lane Departure Warning (LDW) is a feature in ADAS that alerts drivers when they are unintentionally leaving their lane. This system can potentially reduce single-vehicle, sideswipe, and head-on crashes by 11%, and when these types of crashes occur, LDW reduces injury occurrences by 21%.

The LDW system uses forward-facing cameras mounted on the windshield, near the rearview mirror. These cameras monitor lane markings. If the vehicle starts to leave the marked lane while the turning signal is off, the system alerts the driver. A lane departure alert can be an audible alert, a dashboard indicator, or a seat or steering wheel vibration. There are variations to the LDW technology:

- Lane Departure Warning (LDW):

  Provides audible, visual, or rumble warnings when the car goes over or is nearing the lane boundary, when the driver hasn't activated the turning signal.

- Lane Keeping Assist (LKA):

  Adds to LDW system capabilities, applying automatic braking, steering, or both to keep a car within lane and road markings.

- Lane Centering Assist (LCA):

  Focuses on keeping a vehicle centered in its traveling lane, applying automatic steering, braking, or both.

- Road Departure Mitigation (RDM):

  Applies visual and audible LDW to alert the driver. If no steering correction is applied, steering torque is used to keep the car in the intended lane.These advancements in ADAS are crucial for improving road safety and enhancing the capabilities of autonomous vehicles.

4.3 LANE DEPARTURE WARNING FUNCTION:

LDWs are in-vehicle systems that warn the driver of an unintended lane departure on highways and highway-like roads, with one- or two-lane markings depending on the type of system. LDWs have no actors for influencing the vehicle heading. These systems provide a warning to the driver and request him to initiate proper actions to avoid any unintended maneuver. In the simplest case two warning zones are defined. The functional elements of a LDWs are shown in Figure 4-2. The position detection unit recognizes the lane markers and enables the warning system to decide if a warning Should be issued. For this purpose, the lateral deviation from the lane border needs to be extracted. In addition, the rate of departure as well as the curvature of the upcoming road segment might be used but are not required necessarily. Also, it should be noted that LDW function can be operational if just the lane markers of one side are visible. In that case the system.

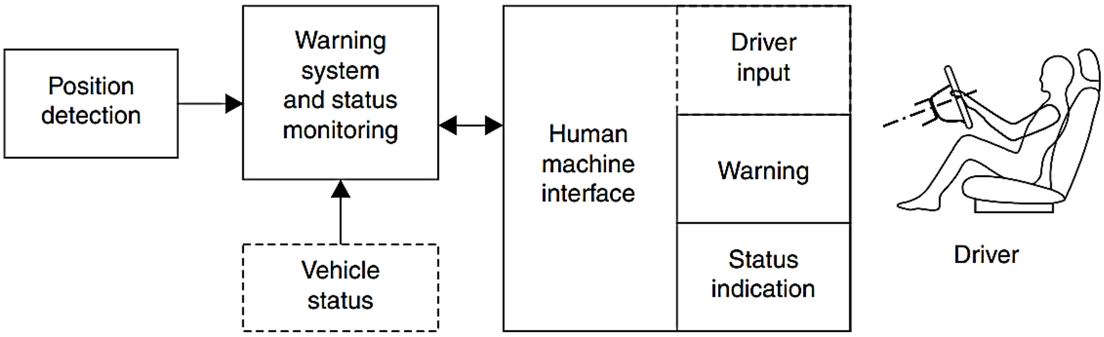


Figure 4- 2: Functional elements of a LDWs [5].

May assume a default lane width to establish a virtual lane border on the opposite side from the visible lane marking. Technically position detection unit could be realized either by a camera with image processing for lane marker recognition or by a laser sensor. The warning system decides, based on the information of the position detection unit, if a warning should be issued. In the simplest case the lateral distance DLC (Distance to Line Crossing) to a lane boundary is used as criterion. In Figure 4-3 a bird's eye view of a vehicle traveling in a lane is shown. The inner zone of the lane is the ''no warning zone'' whereas the warning zones cover the area of the lane boundaries. To cope with detection inaccuracies a ‘‘may warn’’ zone could be defined in between the no warning zone and the warning zones [5].

More elaborate warning systems use predictive criteria like ‘‘Time to Line Crossing’’ TLC (sometimes referred as TTLC) to issue a warning shortly before the lane departure happens. TLC could be defined as:

$$t_{LC} = \frac{D}{V_Y} \tag{4.1}$$

where D is the lateral distance of a defined part of the vehicle to the lane boundary and V_Y is the rate of departure for a vehicle traveling straight.

This simplified formula does not consider curvatures. The geometrical conditions for curved road segments and vehicles driving on a curved trajectory can be found in the literature referenced above. Status monitoring and status indication detect if the LDWS is operational and indicate the status to the driver. Functional aspects of status monitoring include monitoring of on/ off switch (if installed) as driver input and velocity reconditions (if given) via vehicle status monitoring. Activation of LDW function is typically realized using an on/off switch which will keep the status after ignition off [5].

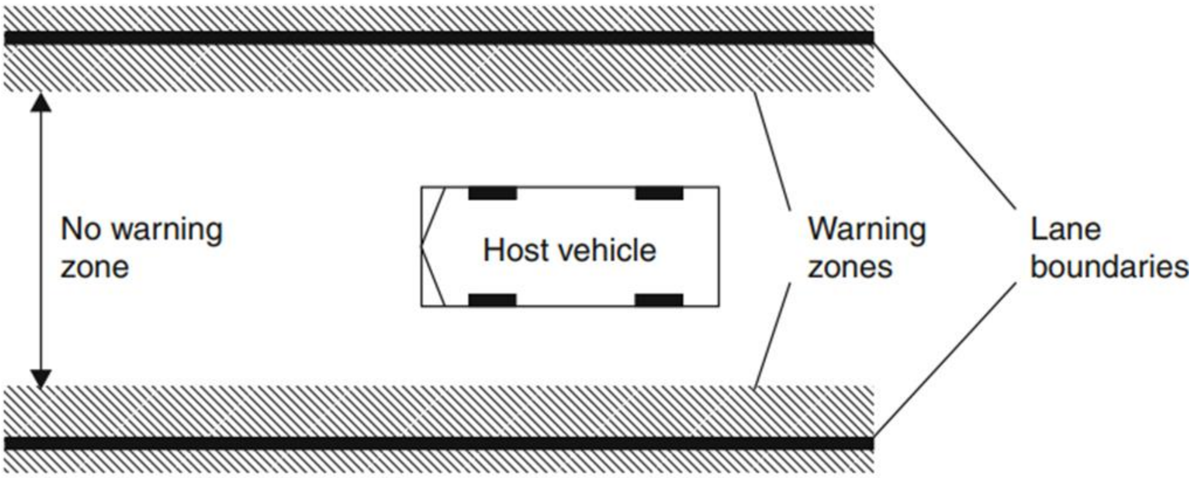


Figure 4- 3: Warning threshold zones and movement of vehicle inside the lane [5].

A basic LDW system can be implemented without vehicle status monitoring, but such a system might suffer from frequent false alarms during intended lane changes (because there will be no detection of turn signal) [5].

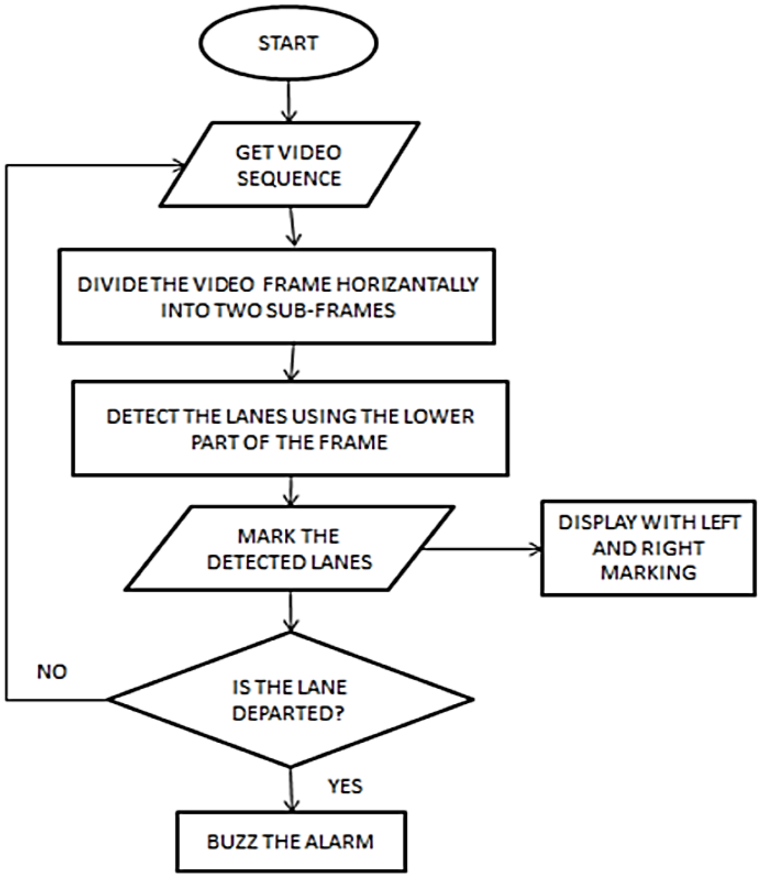


Figure 4- 4: flow chart for lane departure warning [5].

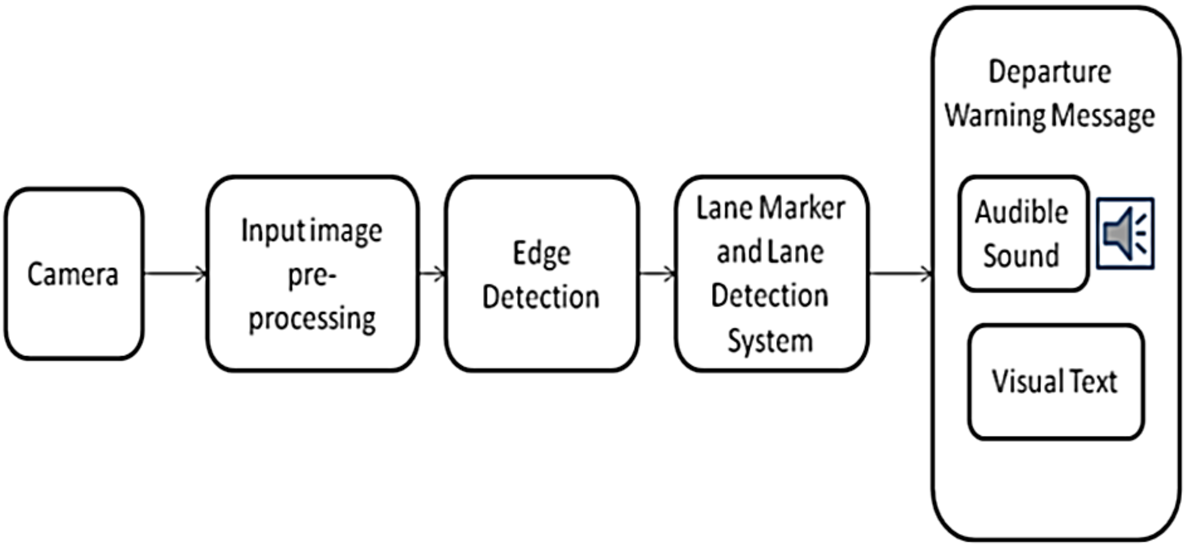


Figure 4- 5: Block diagram of LDW system [5].

This system uses sensors, cameras, or other technologies to monitor the vehicle's position within its lane. If the system detects that the vehicle is drifting out of its lane without the use of turn signals, it provides a warning to the driver. This warning can be visual, audible, or tactile, alerting the driver to take corrective action. Input Video Selection Input video from the Camera mounted on the rear-view mirror of the car is taken with different road sections under different lighting conditions.

4.4 WARNING ELEMENTS:

The warning element is the most important element of the human machine interface of a LDW system. The primary purpose of the warning element is to alert the driver of an imminent lane change and urge him to take corrective actions. First, a warning element needs to reach the driver's attention through one of the human senses. For LDW systems the visual and auditory channels are used.

Table 4- 1: Examples of warning elements used for LDWS.

Warning element	Addressed sense
Beep sound	Auditory
Display indication	Visual

In addition to attracting the driver's attention a warning element should preferably indicate the nature of the danger and if possible, indicate the direction where the danger comes from. For example, a vibrating gas pedal might raise driver's attention but will not indicate a problem with lateral control of the vehicle. Table 4.1 lists warning elements currently used for LDW and some of their properties. A combination of several elements is possible and realized especially for display indication and beep sounds [5].

4.5 OVERVIEW OF ALGORITHM:

The OpenCV methodology is used to fetch frames from video, to convert the frame into matrix.

1. Image Enhancement:

First step involves selecting Region of Interest so that only core part of image is only processed:

- The raw image is blurred using median blur to remove salt and pepper noise.

- Then it is converted to HSV Plane to concentrate only on yellow and white colors in the image.

2. **Edge Detection:**

Edge detection plays a vital role in LDWs, through which lanes are detected. The algorithm adopted for this process is the canny edge detection algorithm. To make the output more precise, morphological operations are carried out over edge detected output in Figure 4-7 edge detection has been applied.

3. **Blurring and Line Fitting:**

Blurring image to remove any noise we don’t want it Figure 4-8 once the edges are detected, the image. These detected lines are fitted back to the original image.

4. **Departure warning:**

Estimation with reference to detected lines, parameters and other factors like car width, Lane width, Position of car, vehicle’s departure is calculated. Once the value exceeds the threshold value, appropriate warnings are issued via HMI.

4.6 STEPS TO LINE DETECTION:

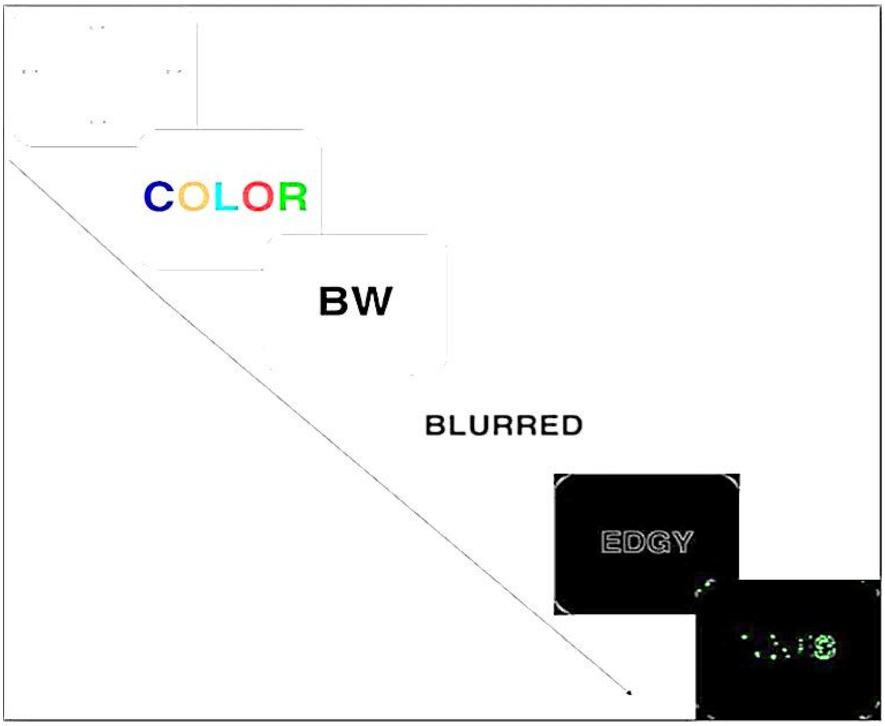


Figure 4- 6: Steps to line detection [6].

1. Edge Detection without Blurring first:

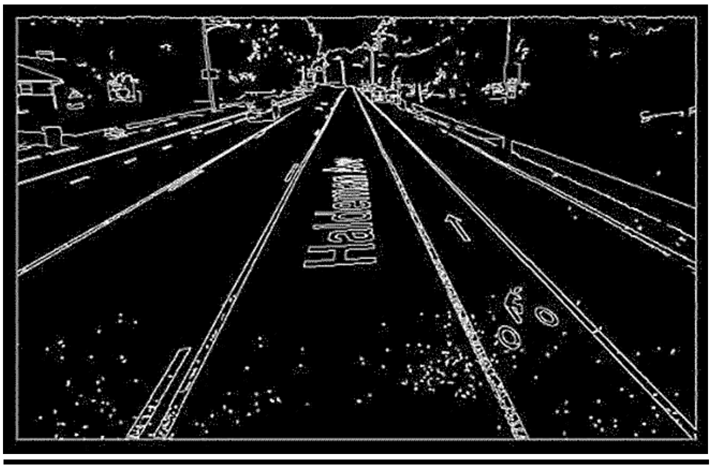


Figure 4- 7: Edge Detection without Blurring [6].

2. Edge Detection after Gaussian Blur applied:

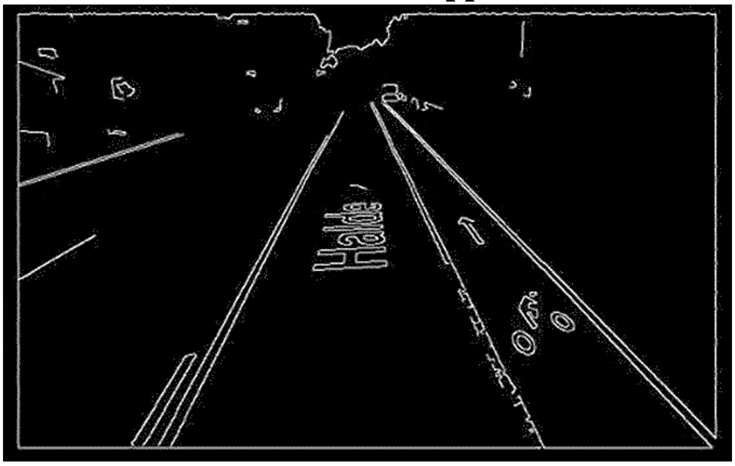


Figure 4- 8: Edge Detection after Gaussian Blur applied [6].

3. Before Crop and Lane detection:

The Hough line detection algorithm will detect many lines on the image. To ensure that only lane markers are detected, we crop out the top of the image and only perform line detection on the bottom.

Figure 4- 9: Before Crop and Lane detection [6].

4. After Crop and Lane detection:

Figure 4- 10: After Crop and Lane detection [6].

We iterate through the set of lines found in the image and pick out the longest line with a positive slope, and the longest line with a negative slope. These correspond to lines representing the right and left lane markers respectively. The left and right lane marker lines may not extend to the top and bottom of the image, so we do some simple math to extend them into the green lines seen in the above image. Using these lane markers, we came up with 3 different ways to detect whether a car was drifting out of its lane. Set static thresholds at certain points of the image. Find the midpoint of the two lines and detect whether either of those two points leave the bounds of the thresholds as shown Figure 4-11.

The two images below show examples of when the vehicle is within lane bounds, and when the vehicle is drifting. Note that in both images below, the yellow and red threshold lines remain in the same position. In the bottom image, the vehicle is drifting to the left as shown Figure 4-12.

5. Vehicle is within lane bounds:

Figure 4- 11: Vehicle is within lane bounds [6].

6. Vehicle drifting slightly. Yellow warning light would turn on:

Figure 4- 12: Vehicle drifting slightly yellow warning light would turn on [6].

Again, we use two boundaries for the yellow warning and red danger signals. This time, we take the lower endpoints of the lane markers and compute the midpoint as shown Figure 4-13 and Figure 4-14. We detect whether the midpoint leaves the bounds set by the thresholds [6].

7. Vehicle is within lane bounds:

Figure 4- 13: Vehicle is within lane bounds [6].

8. Vehicle drifting slightly:

Figure 4- 14: Vehicle drifting slightly [6].

9. Vehicle is within lane bounds:

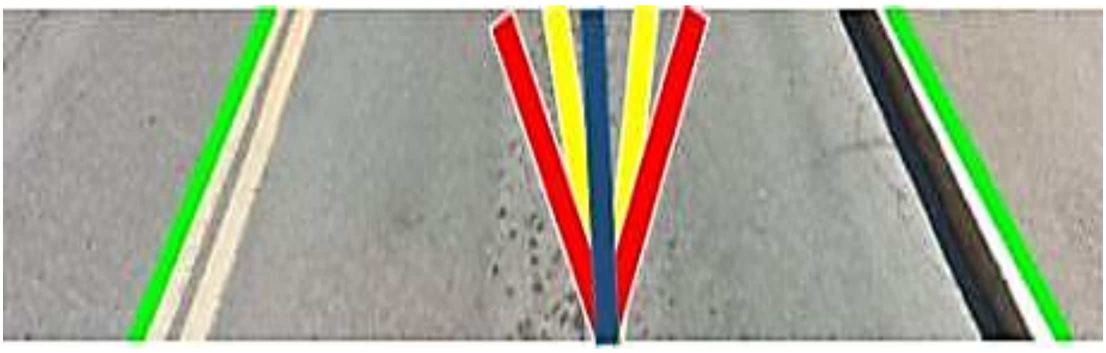


Figure 4- 15: Vehicle is within lane bounds [6].

4.7 ADVANTAGES AND DISADVANTAGES OF LDWS:

Lane Departure Warning (LDW) systems are designed to alert drivers when they unintentionally drift out of their lane. Here are some advantages and disadvantages associated with LDWss:

4.7.1 Advantages of LDWs:

1. **Enhanced Safety:**

   LDW systems can contribute to overall road safety by providing timely warnings to drivers who might be distracted or fatigued, helping them stay within their lane.

2. **Reduced Accidents:**

   By alerting drivers when they unintentionally leave their lane, LDW systems can help prevent accidents caused by lane departure, such as sideswiping or collisions with oncoming traffic.

3. **Driver Awareness:**

   LDW systems can improve driver awareness and attentiveness, acting as an additional layer of safety and reducing the risk of accidents caused by momentary lapses in concentration.

4. **Adaptive Features:**

   Some advanced LDW systems are integrated with lane-keeping assist or lane centering features, which actively intervene to help keep the vehicle within its lane.

5. **Customization:**

   Many LDW systems allow for customization, enabling drivers to adjust the sensitivity of the alerts based on their preferences or driving conditions.

4.7.2 Disadvantages of LDWs:

1. **False Alarms:**

   LDW systems may produce false alarms, especially in situations where the road markings are unclear, faded, or missing. This can lead to driver annoyance and potentially cause distraction.

2. **Dependency Risk:**

Over-reliance on LDW systems may lead to complacency among drivers, if the technology will always prevent lane departure, potentially compromising their attentiveness and reaction times.

3. **Limited Effectiveness in Certain Conditions:**

LDW systems may be less effective in adverse weather conditions, such as heavy rain, snow, or fog, where visibility is reduced, or road markings are obscured.

4. **System Malfunctions:**

Like any electronic system, LDW systems can experience malfunctions or errors, leading to either false alerts or a failure to provide warnings when necessary.

5. **Cost:**

Vehicles equipped with advanced safety features, including LDW systems, may come at a higher cost. Additionally, repairing or replacing these systems in the event of damage can be expensive.

Chapter (5)

5. ADAPTIVE CRUISE CONTROL SYSTEM

5.1 INTRODUCTION:

Automobiles are perhaps the domain that represents the 21st century. vehicles play a major role in our day-to-day life for transportation. The term Cruise Control refers to the concept of assisting drivers in the task of longitudinal vehicle control to avoid any accident or collision. With the increase in the world population the need for automobiles as well as its usage daily has increased drastically. This leads to heavy traffic, rush, collisions, and accidents. The automobile black box is used to analyze the cause of vehicular accidents and prevent loss of life and property from vehicle accidents [7].

5.2 DESCRIPTION:

Adaptive Cruise Control (ACC) is an important feature in ADAS. It automatically adjusts a vehicle's speed when there are slow-moving vehicles ahead, with the aim of maintaining a safe following distance. When the road ahead is clear, ACC automatically accelerates to your pre-set speed. The ACC system uses internal computers and advanced sensors such as radar or laser systems to monitor other vehicles on the road. If the driver's vehicle has insufficient braking distance, the ACC system transmits a signal to the engine or brakes, which causes the vehicle to slow down. When the path is clear, the ACC system returns the vehicle to the driver's selected speed. These advancements in ADAS are crucial for improving road safety and enhancing the capabilities of autonomous vehicles [8].

5.3 WORKING & PRINCIPLE:

Works by detecting the distance and speed of thevehicles ahead by using either a LIDAR system or a RADAR system. The time taken by the transmission and receptionis the key to the distance measurement. The shift in frequency of the reflected beam by Doppler Effect is measured to know the speed. Depending on this speed, the brake and throttlecontrols are done to keep the vehicle in a safe position [9].

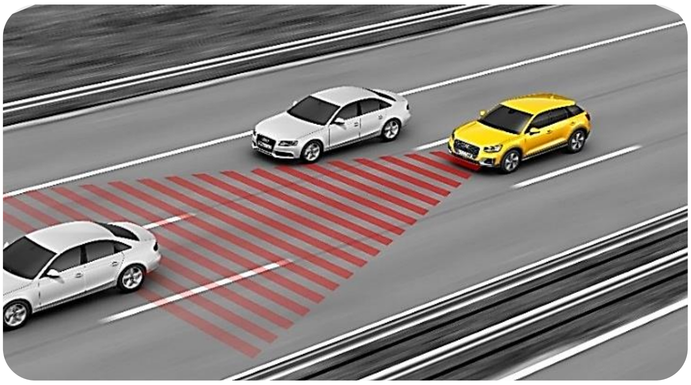


Figure 5- 1: Adaptive cruise control [9].

5.3.1 Working of Adaptive Cruise Control:

- As shown Figure 5-2, The gun transmits the waves at a given frequency toward an incoming car.
- Reflecting waves return to the gun at a different frequency, depending on how fast the car being tracked is moving.
- A device in the gun compares the transmission frequency to the received frequency to determine the speed of the car.
- We can design the chip or ACC having an algorithm such that it will give output only when the input signals are less than the corresponding safe distance value.
- So only when the distance between the car and the object in front of it is less than the same distance value the embedded system will give output to the breaking and the accelerating units.
- Thus, the safe distance will be kept always [9].

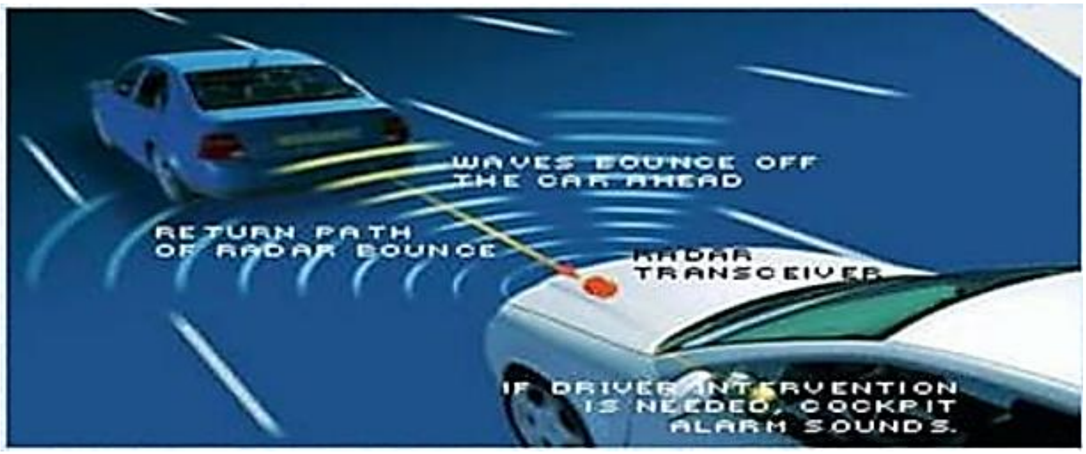


Figure 5- 2: Working of Adaptive Cruise Control [9].

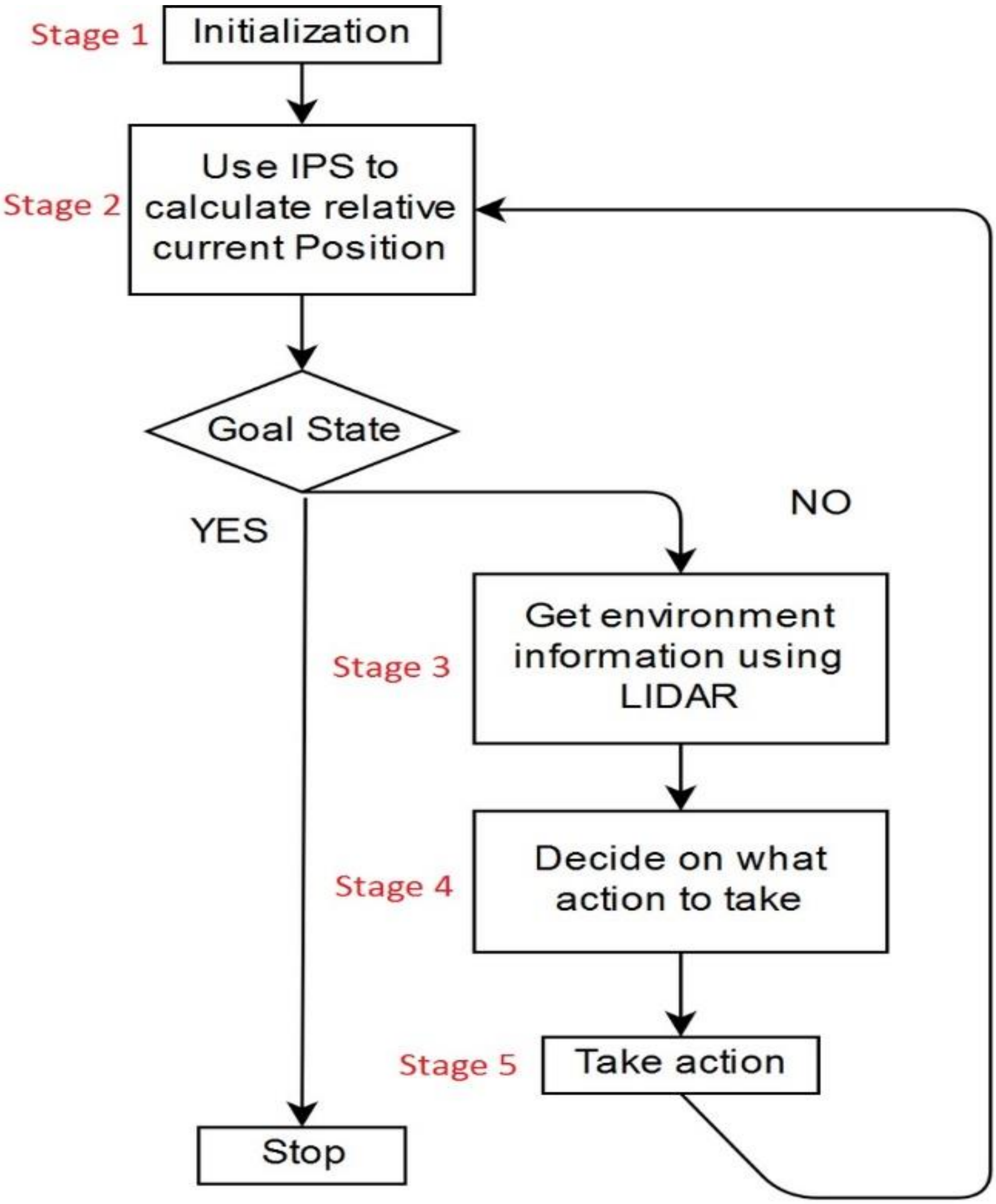


Figure 5- 3: Flow Chart [10].

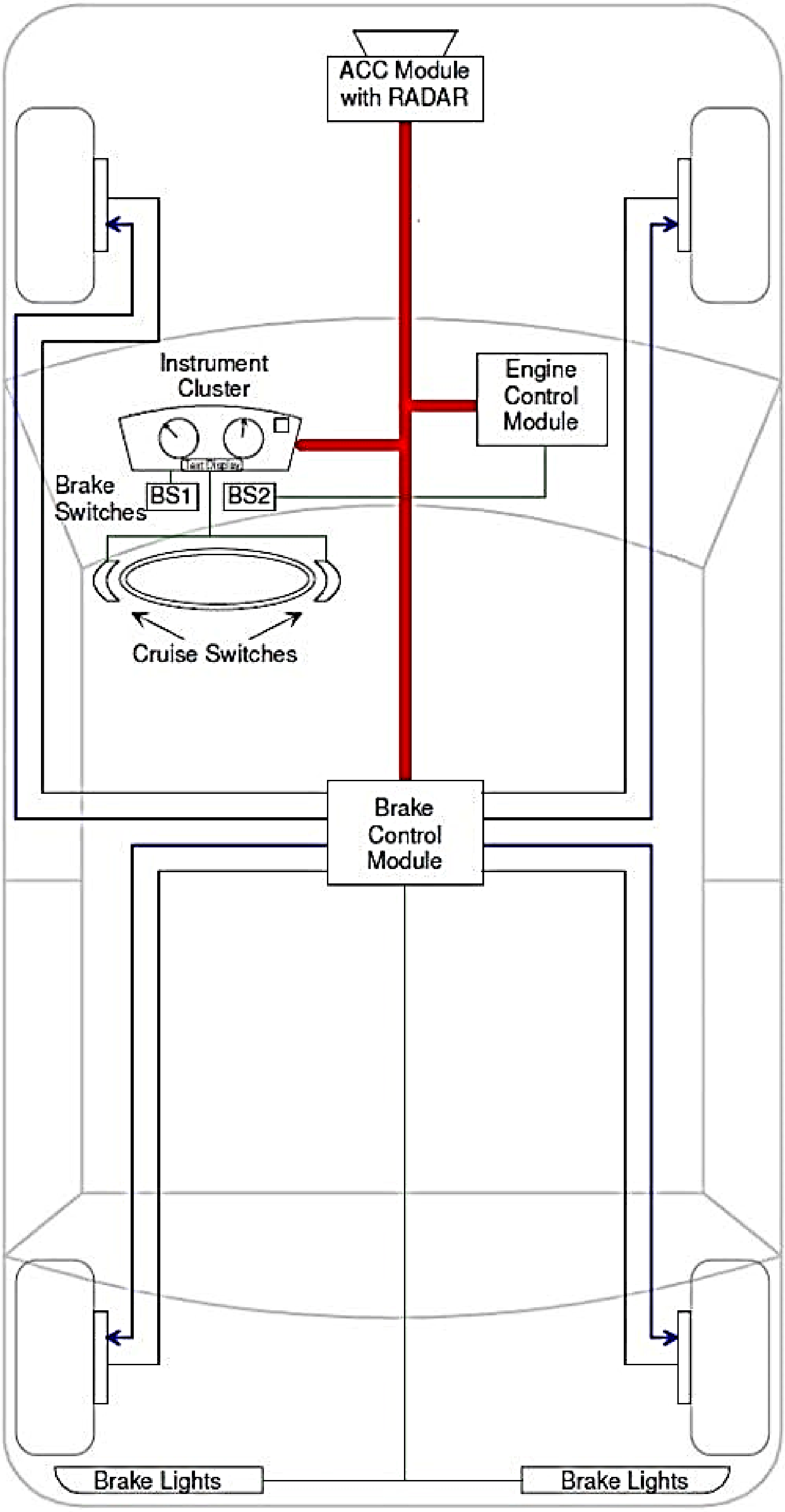


Figure 5- 4: ACC system components [10].

5.4 Block Diagram of ACC:

- ACC uses sensors to monitor the distance between your car and the vehicle ahead, and we can't forget that there are also obstacles on the road, so we used the 3D depth camera to recognize the height and the distance between the obstacles and the car. And by collecting the inputs by the camera and the sensors and sending it to the microcontroller the output will be:
- Audible Alerts: In critical situations, ACC systems can produce audible warnings, such as beeps or alarms, to alert the driver to take action, like braking or steering to avoid a collision by using the Buzzer or sending alerts through the car sound system.
- Visual Warnings: Some ACC systems display visual alerts on the dashboard or instrument cluster. These warnings might include flashing lights, icons, or messages indicating a potential collision or need for driver intervention.
- Acceleration and Braking: ACC can slow down your vehicle by reducing throttle or even apply the brakes if necessary to maintain a safe distance. Once the road ahead is clear, it can accelerate back to the preset speed by the presence of the speed controller as shown in Figure 5-5.

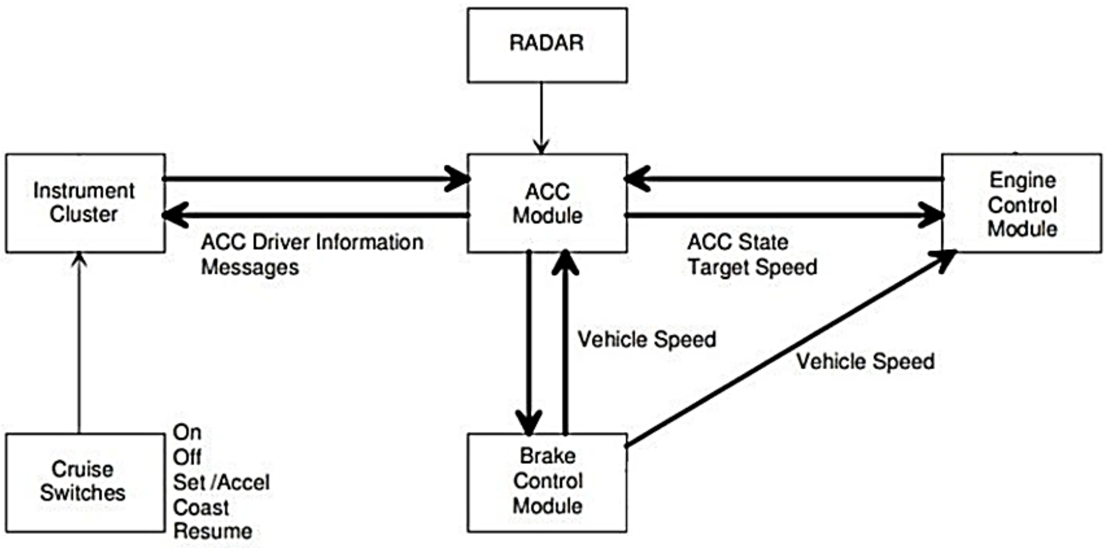


Figure 5- 5: Block diagram of an ACC system [10].

In order to have a functional ACC system, a sensor must be attached to the front of the vehicle, this sensor is used to detect obstacles and vehicles. If a car is moving slowly is detected, the ACC system will deaccelerate, to maintain a safe distance between the ACC vehicle and the other vehicle as shown in Figure 5-6. when the ACC system detects that there is no car in front of it, the car will start gaining speed back to the speed set by the driver control speed. This operation allows the ACC vehicle to automatically deaccelerate and accelerate without the involvement of the driver. The method of the ACC vehicles accelerates is controlled by an engine control unit and limited brake operation [9].

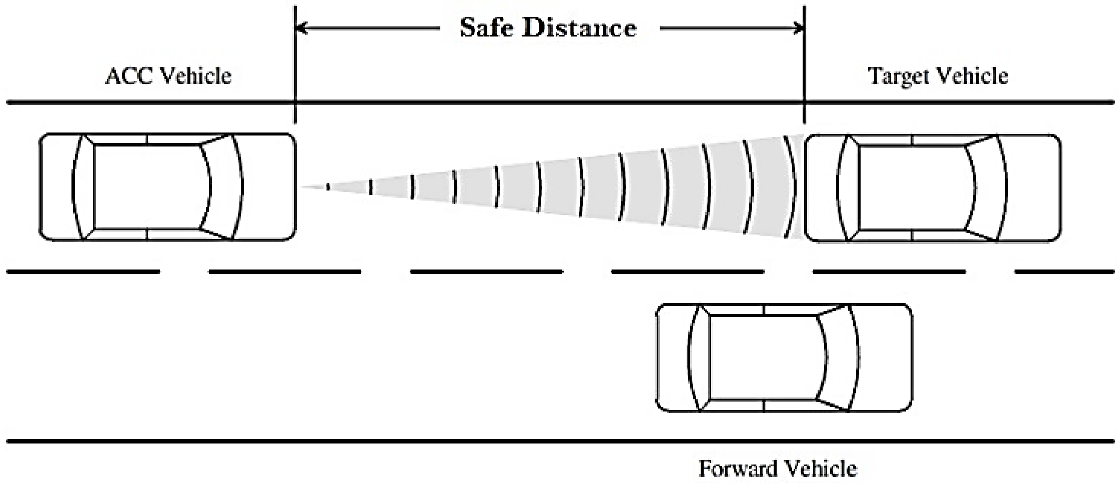


Figure 5- 6: Safe distance [9].

5.5 SENSORS:

SENSOR is a converter that measures physical quantity and converts it into signal which can be read by observer or by instrument. SENSOR responds to an input quantity by generating a functionally related output usually in form of electrical or optical signal.

In this project types of SENSORS can be used:

- LiDAR
- Fusion Sensor (Camera)

These two parts work together to track the car from non-moving objects.

5.5.1 LiDAR:

LIDAR stands for "Light Detection and Ranging". It measures distance by illuminating laser and analyzing the reflected light. LiDAR is a ranging device, which measures the distance to a target. The distance is measured by sending a short laser pulse and recording the time lapse between outgoing light pulse and the detection of the reflected (back-scattered) light pulse as shown Figure 5-7.

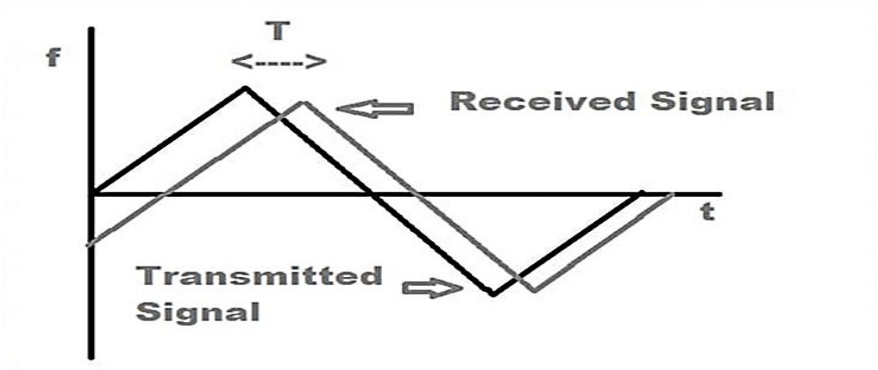


Figure 5- 7: Range Estimation Using Lidar [9].

5.5.2 TF Luna:

TF-Luna is a single-point ranging Lidar based on the TOF principle. With its unique optical and electrical design, it adopts an 850nm infrared light source to achieve stable, accurate, and highly sensitive distance measurements. TF-Luna LiDAR Range Sensor has built-in adaptation algorithms for a variety of application environments and targets and is open to a variety of adjustable configurations and parameters. It can ensure excellent range performance in complex environments and meet the needs of customers in complex application scenarios [11].

- **Specification of TF-Luna:**
  - Ranging distance: - 0.2m~8m@90% reflectivity (Indoor 0Klux)
    - 0.2m~2.5m@10% reflectivity (Indoor 0Klux)
    - 0.2m~8m@90% reflectivity (Indoor 90Klux)
    - 0.2m~2.5m@10% reflectivity (Indoor 90Klux)
  - Accuracy: ±6cm@(0.2m-3m) / ±2%@(3m-8m).
  - Light source: VCSEL.
  - Communication interface: UART/I2C.

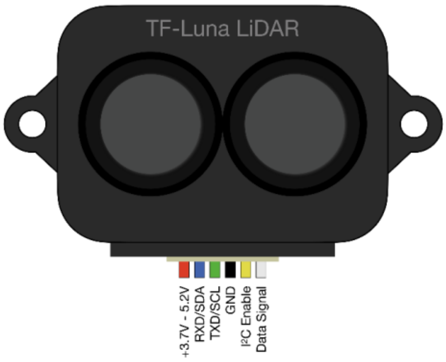


Figure 5- 8: TF-Luna LiDAR [11].

- **Working Principle:**

This product is based on the TOF (Time of Flight) principle. Specifically, the product periodically sends out a modulated wave of near-infrared light, which is reflected when it meets an object. The time of flight is obtained by measuring the phase difference between the modulated wave and the round trip, and then calculating the relative distance between the product and the measured target, as follow Figure 5-9:

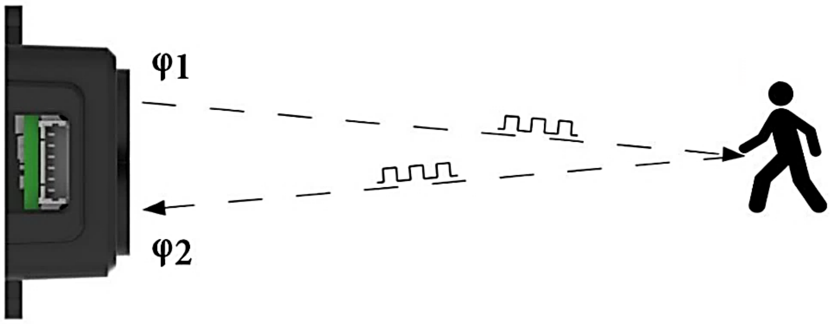


Figure 5- 9: Working Principle [11].

$$D = \frac{C}{2} \times \frac{1}{2\pi f} \times \Delta\varphi, c \text{ is speed of light} \tag{5. 1}$$

5.5.3 Astra Pro Plus Depth Camera:

This ROS real-sensing depth camera comes with a color camera (RGB), IR camera, infrared projector, and depth processor. With dozens of functions such as face recognition, gesture recognition, human skeleton recognition, 3D measurement, environment perception, 3D mapping navigation, etc., it can be widely used in TV, mobile phone, robot, drone, VR/AR, smart home, and other fields. We specially designed an extension bracket for it, which is convenient for users to install on various ROS robots or smart cars [12].

- **Specifications of Astra Pro Plus Depth Camera:**
  - RGB Image Resolution 1920 x 1080 @30fps.
  - Depth Image Resolution 640 x 480 @30fps.
  - Power Supply: USB 2.0.
  - Power Consumption < 2.5 W.
  - Operating Systems: Android / Linux / Windows7/8/10.
  - SDK: Astra SDK or OpenNI 2 or 3rd Party SDK [13].

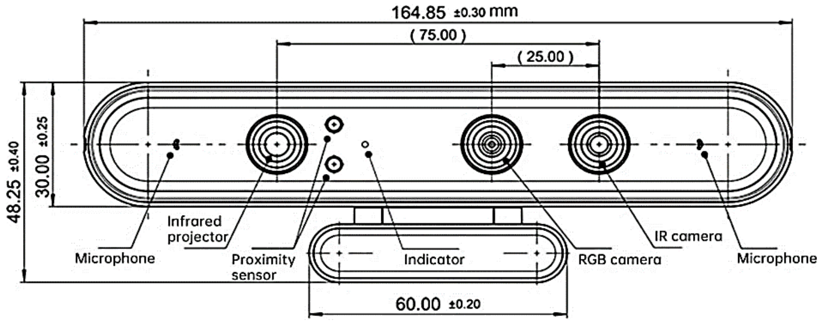


Figure 5- 10: Astra Pro Plus Depth Camera [12].

5.6 CONTROLLER ACTION:

Depending on the present traffic situation two type of controller are possible SPEED CONTROL and HEADWAY CONTROL.

- **SPEED CONTROL**: if there is no vehicle presently in front, then the speed is controlled about a set point just as in conventional cruise control.
- **HEADWAY CONTROL**: to keep a safe distance between the vehicle's headway control is required.

Figure 5- 11: Controller Flow Chart.

5.7 ADVANTAGES OF ADAPTIVE CRUISE CONTROL:

- Driver is relieved from careful acceleration, deceleration and braking in congested traffics.
- Accidents can be reduced.
- It’s very useful for long driving.
- Can avoid unconsciously violating speed limits.
- Increased fuel efficiency.

5.8 DISADVANTAGES OF ADAPTIVE CRUISE CONTROL:

- High Cost.
- Not for heavy traffic.
- Encourages the driver to become careless.
- Dangerous in slippery roads.
- A high market penetration is required if a society of
- intelligent vehicles are to be formed.
- The ACC systems do not respond directly to the traffic.

Chapter (6)

6. TRAFFIC SIGN INTERPRETATION

6.1 INTRODUCTION:

In the intricate landscape of modern transportation, traffic signs emerge as silent sentinels, orchestrating the intricate dance of vehicular movement through essential messages. These universal symbols transcend linguistic barriers, utilizing colors, shapes, and symbols to communicate vital information, fostering safety and efficiency in real-time across diverse transportation networks.

This introduction embarks on a journey through the critical significance of traffic signs in the realm of real-time Traffic Sign Recognition and Interpretation (TSRI). It explores the historical roots of these symbols, their standardized design systems, and their profound impact on road safety, especially in the context of AI-driven systems.

6.1.1 Legal Basis and Regulations:

The legal foundation for traffic signs gains heightened relevance in the realm of AI-driven TSRI. Regulations establish the parameters for conveying restrictions, warnings, and information to road users, with a focus on ensuring rapid interpretation by intelligent systems. The responsibility for these signs, distributed among national and regional highway authorities, becomes pivotal for consistent recognition and interpretation by AI models.

6.1.2 Traffic Signs Manual:

In the context of TSRI, the Traffic Signs Manual serves as a crucial guide, offering advice to optimize the use of traffic signs and road markings. It becomes a valuable resource for complying with mandatory requirements while aligning with the real-time processing capabilities of AI models. The emphasis on reducing sign clutter is particularly pertinent in the context of swift decision-making by intelligent systems.

6.1.3 Definitions and Guidance:

Clear definitions and guidance are imperative in the AI-driven TSRI landscape. The distinctions between legal requirements ("must"), recommended actions ("should"), and permissible options ("may") become crucial for the seamless integration of AI models. The broad scope of "signing" extends to encompass various elements, including traffic signs, carriageway markings, beacons, studs, and traffic signals, all of which require precise interpretation in real-time.

This journey into the world of traffic signs takes on a new dimension as we explore their role in the realm of AI-driven TSRI. It aims to deepen our understanding of these symbols, emphasizing their vital contribution to real-time safety, order, and efficiency on our roads, guided by the intelligence of advanced recognition systems.

6.1.4 Overview:

In the dynamic landscape of modern transportation, the seamless interaction between vehicles and the road environment hinges on the effective interpretation of traffic signs. These ubiquitous symbols serve as a vital communication channel, conveying crucial information to drivers, pedestrians, and automated systems. The advent of intelligent transportation systems places a premium on real-time recognition and interpretation of traffic signs, emphasizing the need for advanced technologies to ensure swift and accurate decision-making. Traffic sign recognition and interpretation have transcended traditional boundaries, evolving into a multidisciplinary field that merges computer vision, AI, and real-time processing capabilities. The overarching goal is to enhance road safety, optimize traffic flow, and contribute to the broader framework of smart mobility.

6.2 BACKGROUND:

The background knowledge and fundamental ideas around convolutional neural networks (CNNs), object detectors, and earlier research on traffic sign detection are introduced in this chapter.

The key layers that comprise a CNN architecture and deep learning are explained in Section 6.2.1. The most widely used CNN architectures that have produced object categorization results at the cutting edge are shown in Section 6.2.2. Concepts related to object detection and the primary CNN-based object detector frameworks are covered in Section 6.2.3.

6.2.1 Deep learning and CNNS:

Artificial neural networks, or neural network-inspired algorithms, are a subset of deep learning. The pioneers and authorities in the field define deep learning, and their intricate and subtle ideas provide a great deal of insight into deeper learning.

As deep learning has developed in the modern era, a deluge of data from all around the world has been collected. Many sources, including social media, online ad networks, e-commerce, and virtual theaters, provide vast volumes of information.

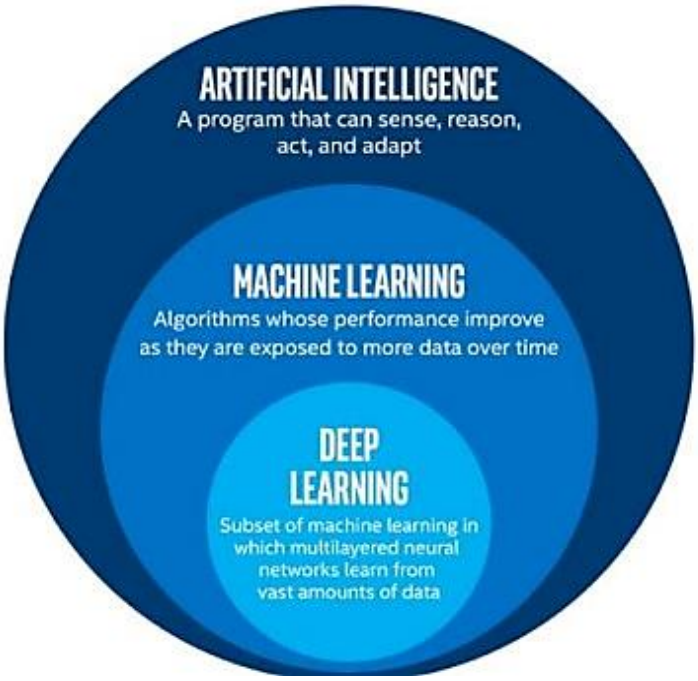


Figure 6- 1: Uses of AI vs ML vs Deep Learning [14].

A neural network is an algorithmic system that mimics the way the human brain functions to find the relationships that underlie a piece of data. Neural networks in this sense relate to artificial or organic neuronal structures. Neural networks can attain optimal results without requiring a redesign of the power output since they are able to adjust to changing inputs. The application of neural networks to the creation of artificially intelligent trading systems is growing quickly.

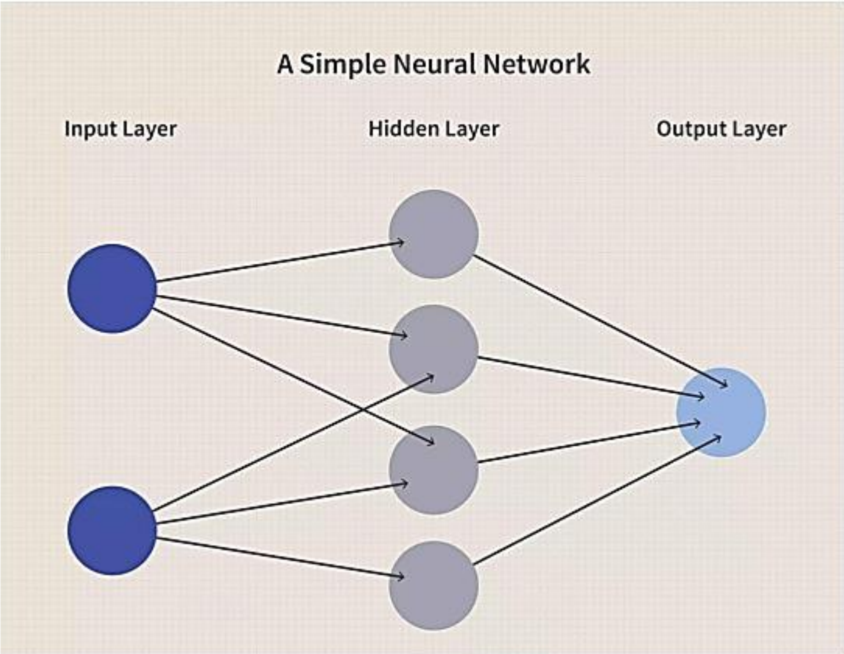


Figure 6- 2: Structure of a Neural Network [15].

A unique kind of neural network called a convolutional neural network (CNN) is made to recognize patterns in input images. CNNs generate feature maps automatically, allowing them to learn directly from the incoming data. CNNs have shown to be incredibly effective at classifying images. and applications for object recognition. Since feature maps are directly learned during the training process, as opposed to traditional object identification systems where features are manually generated, the use of CNNs has expanded recently.

The ability to retrain contemporary architectures for specific applications eliminates the need to train CNNs from start, which is another benefit of CNNS. We refer to this procedure as transfer learning.

CNN architectures are made up of numerous stacked layers, each of which extracts unique information from the input images. Typically, the latter layers learn more complicated features that uniquely characterize the input data, whereas the initial layers learn more fundamental features like edges. An input layer, hidden layers, and an output layer make up a conventional neural network. Each layer has a distinct purpose and carries out a specific task; the most often utilized layers are listed below.

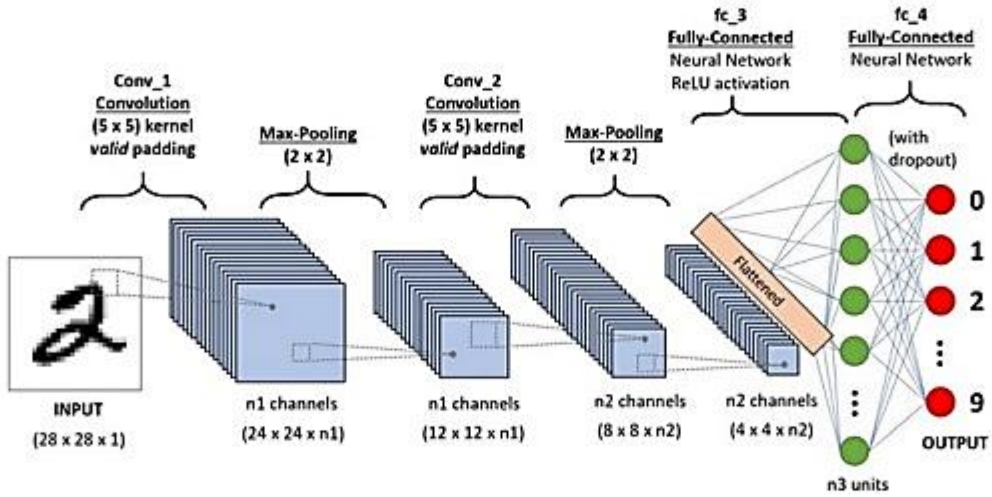


Figure 6- 3: Typical architecture of a convolutional neural network and its different layers [16].

- ## Convolutional layer

The main part of CNN is the convolutional layer. By performing convolution operations between a local area of the input and the related kernel, also known as a filter, it modifies the input data. Construction of high-level input features is the layer's goal. image, which can then be applied to classification or object identification tasks and utilized to recognize patterns like forms Figure 6- 4.

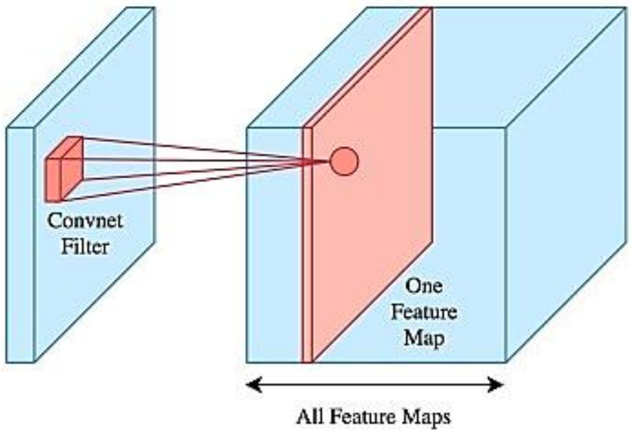


Figure 6- 4: A convolutional layer showing the convolutional operation between its associated filter and the input data [16].

- ## Pooling layer:

To minimize the number of parameters in the finished model and manage overfitting during training, the pooling layer's job is to shrink the feature maps. With a down sampling value of 2, pooling layers are often positioned after the convolutional layer Figure 6- 5.

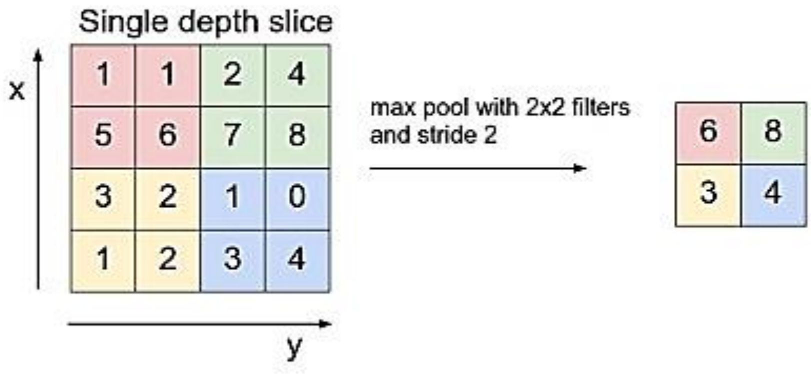


Figure 6- 5: A pooling layer applying a max operation to reduce the size of a feature map [16].

- ## ReLU layer:

Every element of a feature map is subjected to a non-linear thresholding function by the ReLU (rectified linear unit) layer, where the positive values are constant and the negative values are set to zero Figure 6-6.

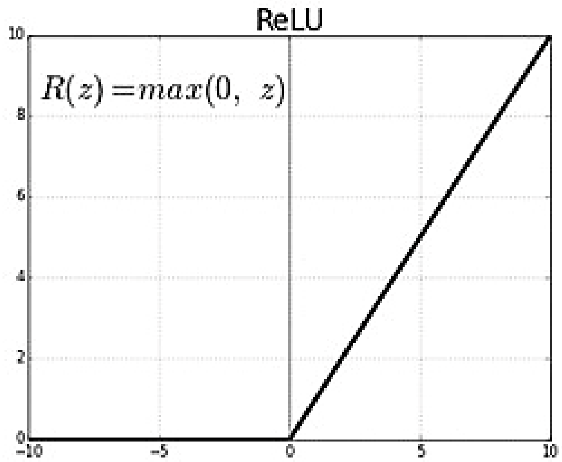


Figure 6- 6: ReLU layer and its associated thresholding function applied to the input data [16].

- ## Batch normalization layer:

It is a technique called batch normalization. By normalizing layer inputs, this method solves the internal covariate shift issue. According to the original study, batch normalization reduces the requirement for Dropout layers which are usually employed to reduce overfitting, by acting as a regularize and enabling the use of greater learning rates. Input: Values of x over a mini-batch: B= {x_ (1…m)}; Parameters to be learned: γ,β
Output: {y_i= BN_(γ,β) (x_i)}

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$
$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2 \quad \textit{// mini-batch variance}$$
$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad \textit{// normalize} \tag{6. 1}$$
$$y_i \leftarrow \gamma \hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i) \quad \textit{// scale and shift}$$

- **L2 Regularization:**

Regularization involves introducing a penalty term to the loss function to prevent overfitting. The most used type of regularization is L2 regularization. It increases the loss function by the square magnitude of each parameter as a penalty: λ P j w 2 j. where λ is the penalty term, which also defines the amount of penalty imposed to the model's weights and is often referred to as the regularization parameter or weight decay.

- **SoftMax classifier:**

SoftMax classifier is a popular multi-class classifier. It uses as activation function the SoftMax function, described in the following equation:

The SoftMax function produces a vector with values between 0 and 1 for each class j from an input vector (z) with arbitrary scores for each class j. The property of the output vector is that the total of its members equals one. Additionally, the SoftMax function's output indicates. the normalized likelihood that features vector z in the input is a member of class j Figure 6-7.

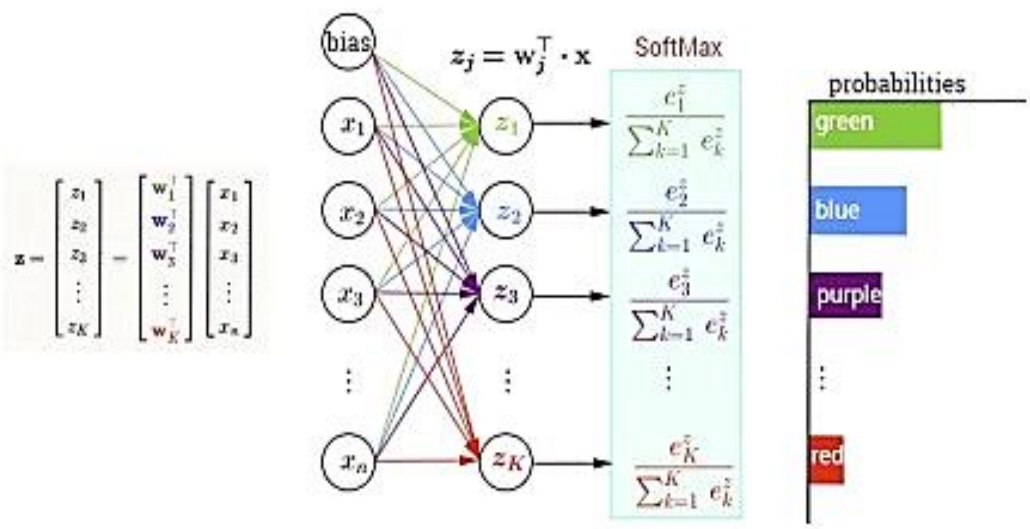


Figure 6- 7: SoftMax classifier. For an input z with arbitrary scores for each class j, the output is a vector with values between 0 and 1 for each class j [16].

6.2.2 CNNs Architecture:

CNNs are increasingly used for tasks like image classification because of advances in computing power and the availability of large datasets. LeNet was the first CNN to be successfully applied in 1998; it was used to recognize machine-printed and handwritten text personalities. The most often utilized CNN architectures in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) for image categorization include.

1. **AlexNet:**

In the 2012 ILSVRC competition, AlexNet created by Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton, earned a top 5 error rate of 16%. AlexNet has remained a mainstay in computer vision literature since its release, serving as an inspiration for other successful systems.

2. **GoogleNet:**

A top 5 error rate of 6.67%, GoogLeNet emerged victorious in the ILSVRC competition in 2014. The inception module, which employs tiny convolution filters to reduce the final model's parameter count, is mostly responsible for this achievement.

3. **VGGNet:**

VGGNet simplicity helped it come second in the 2014 ILSVRC competition. As a feature extractor for other applications like object identification, it is frequently utilized. Approximately 140 M characteristics mean that it needs a lot of memory and processing resources.

4. **ResNet:**

The ILSVRC competition was won in 2015 by ResNet, a system created by Kaiming He et al., with a top error rate of 3.57%. During the training phase, it first introduced the idea of "skip connections," which aid in reducing the vanishing gradient problem.

5. **MobileNet:**

In order to overcome computer power constraints in embedded vision applications, MobileNet was created. MobileNet makes use of a depthwise separable convolution to minimize both the model size and the number of parameters. With fewer parameters, it attains accuracy levels comparable to those of VGG-16 and GoogleNet, enabling quicker training and inference.

6.2.3 object detection:

One of the primary topics of computer vision research is object detection. Its primary goals are to locate objects in an image (object localization) and identify the item's class (object classification) among a preset list of categories Figure 6-8.

For most purposes, a single camera placed in the front of the car will be adequate. Although the hardware of this technology is straightforward, the software required to accurately recognize traffic signs is quite sophisticated, and automakers put a great deal of effort and resources into making them better. Active traffic sign monitoring systems are available from BMW, Mercedes, Tesla, and other manufacturers. These systems indicate the current limits on the car display. For example, the relevant limitations will be displayed on the vehicle display if the vehicle is presently traveling in a 50-mph speed limit no passing zone. Even though most of these technologies are now only informational, they nevertheless aid in improving driver focus while driving. The first and most extensively studied is driverless cars. A car must be able to recognize signs and know what to do when it comes across them to be able to drive itself. The making of an inventory of every traffic sign on the road is another potential use. The technology would map all traffic signs in each region once a vehicle passed past it. Apart from mapping, it can also assess the condition of the traffic sign now and alert the relevant authorities if a traffic sign needs to be changed soon.

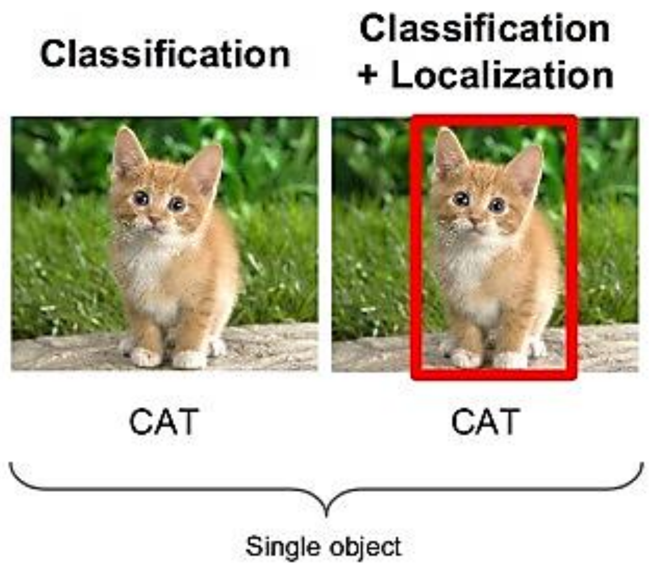


Figure 6- 8: Difference between image classification (left) and object detection (right) [16].

6.2.3.1 Traditional object detection algorithms follow a typical pipeline that consists of:

Region selection: Conventional techniques of object detection in images include scanning the entire image with sliding windows of varying sizes and scales, producing smaller image crops that are then examined one at a time to see if an item is inside the sliding window. Owing to the substantial quantity of examined prospects, this procedure incurs high computing costs.

Feature extraction: We need visual characteristics that provide us with useful information about the picture to examine each candidate that is created throughout the sliding windows procedure. SIFT (scale-invariant feature transform) features, which have the following characteristic, are popular visual characteristics. Haar-like features are employed in face recognition, while HOG (histograms of oriented gradients) features are utilized in human detection due to their invariance to picture size and rotation.

Classification: The next stage is to categorize the picture crops in a target object class and background once we get the feature descriptor vector of each sliding window. SVM (support vector machines) is the most often utilized method.

Non-maxima suppression (NMS): Several candidates are produced as a consequence of the sliding window method, and non-maxima suppression is used to weed out the most important findings. The object detector selects just those results with the greatest scores as the outcome. The popularity of CNN-based object detectors can be attributed to the effectiveness of using CNN designs (such as VGGNet [and ResNet) as feature extractors. These CNN architectures have produced object detectors with state-of-the-art results in terms of sufficient detection speed and precision to be used in mobile devices. CNN networks can identify more intricate patterns in pictures because of their deep structures, which enable them to learn more complicated characteristics. Since we can train the same model using several datasets, CNN architectures are more suited for a wide range of applications due to the features that CNNs learn being more resilient than features that are explicitly developed.

The goal of generic object detectors built on CNNs is to identify the items in a picture and display their location by enclosing them in a rectangular bounding box.

Two categories may be used to group generic CNN-based object identification techniques: two-phase both one-stage and detectors.

6.3 OBJECT DETECTION FOR TRAFFIC SIGN RECOGNITION:

6.3.1 Introduction to Traffic Sign Recognition in Intelligent Transportation System:

Background on Traffic Sign Recognition:

- **Significance of Traffic Sign Recognition:**

In dynamic traffic environments, accurate and swift traffic sign recognition is crucial for human drivers and autonomous vehicles. AI-equipped systems contribute to automating decision-making, reducing cognitive load, and enhancing overall road safety.

- **Challenges Addressed by AI Models:**

Traffic sign recognition poses challenges like lighting variations, diverse sign characteristics, occlusions, and the need for real-time processing. AI models are designed with sophisticated algorithms to adapt to these challenges, making them essential in intelligent transportation systems.

6.3.2 Importance of AI Models for Real-time Applications:

Role of AI Models:

- AI models play a pivotal role in interpreting visual information from road signs, leveraging deep learning architectures.

- These models enable machines to analyze images or video frames, allowing vehicles to respond appropriately to changing road conditions.

Importance of Model Selection:

- The choice of AI model is pivotal, aligning with specific application requirements.

- Different models excel in aspects such as real-time processing, accuracy, adaptability, and efficiency on resource-constrained devices.

- **Considerations for Model Selection:**

  - Real-time Processing: Models like YOLO and SSD, known for real-time capabilities, are ideal for applications requiring timely responses.

  - Accuracy and Precision: Recognizing and classifying traffic signs accurately is paramount for safe and reliable operation.

  - Adaptability: AI models must exhibit adaptability to varying environmental conditions, including lighting, weather, and road scenarios.

  - Resource Efficiency: Models designed for efficiency become essential for optimal performance in edge devices or vehicles.

- **Balancing Real-time Performance and Accuracy:**

  - Achieving a balance between real-time performance and accuracy is crucial for applications such as autonomous vehicles.

  - Swift processing with high accuracy is vital, emphasizing the significance of AI models in intelligent transportation.

6.3.3 Key AI Model:

- Several effective AI models, each with unique strengths, contribute to traffic sign recognition.

- Real-time object detection models like YOLO and SSD have significantly advanced the capabilities of traffic sign recognition systems.

6.4 OBJECT DETECTION FOR TRAFFIC SIGN RECOGNITION:

6.4.1 Introduction object detection:

How Object Detection Works:

Object detection encompasses traditional image processing techniques and modern deep learning networks, with the latter being more robust to occlusion, complex scenes, and challenging illumination. This crucial computer vision task involves identifying specific classes of visual objects, such as humans, animals, and cars, in digital images or video frames. Its significance extends to various computer vision tasks, including segmentation, image captioning, and object tracking, with applications ranging from pedestrian and animal detection to face recognition and number-plate identification.

AI imaging technology offers flexibility, compatible with various cameras, including commercial security and CCTV cameras. Cross-compatible AI software platforms like TensorFlow and Viso Suite eliminate the need for specialized AI cameras, allowing analysis of digital video streams from any camera. Advancements in multi-core processing, GPUs, and AI accelerators like tensor processing units (TPUs) have significantly increased computing power, enabling real-time object detection and tracking in diverse environments. The synergy of deep convolutional neural networks (CNN) and enhanced GPU computing power drives significant strides in computer vision-based object detection [17].

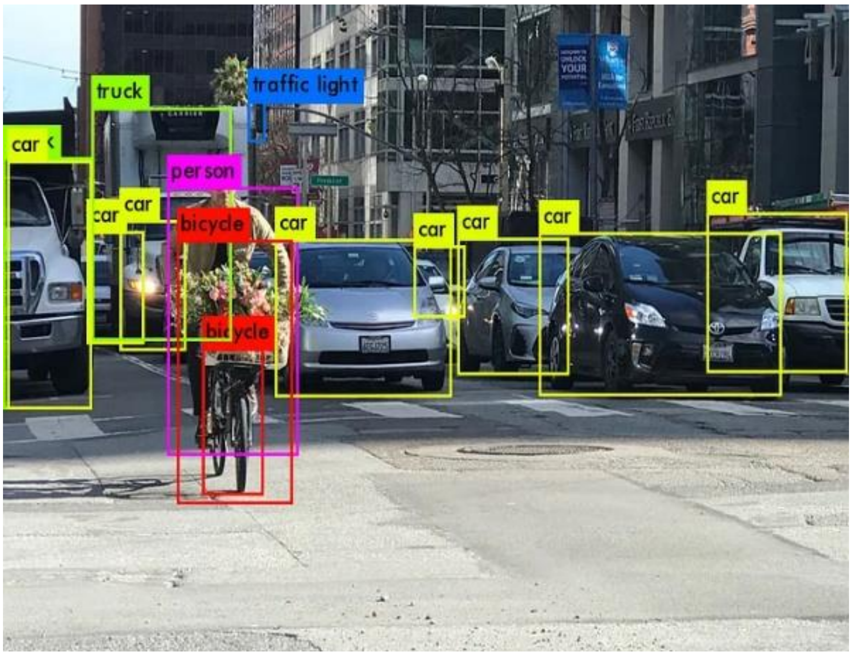


Figure 6- 9: Example of Object Detection [18].

6.4.2 Object Detection: Working Mechanism:

1. Image Processing Techniques:

- **Pros:** No need for historical data for training, and it operates in an unsupervised manner.

- **Cons:** Limited in complex scenarios without a unicolor background, dealing with occlusion, illumination and shadows, and clutter effects. OpenCV is a popular tool for image processing tasks.

2. Deep Learning Methods:

- **Pros:** Significantly more robust to occlusion, complex scenes, and challenging illumination.
- **Cons:** Requires a substantial amount of training data, and the image annotation process is labor-intensive and expensive. Despite challenges, benchmark datasets like MS COCO, Caltech, KITTI, PASCAL VOC, and V5 offer labeled data availability.

Deep learning object detection is widely accepted by researchers and adopted by computer vision companies for building commercial products.

- **Key Metrics:**
  - Metrics, especially mean average precision (mAP), are crucial for fair comparisons between different detectors. Understanding other metrics is essential to grasp the significance of mAP in evaluating object detection performance.
- **2D vs. 3D Detectors:**
  - **2D Detectors:** Provide position information in a 2D plane with bounding boxes.
  - **3D Detectors:** Use data from cameras, LIDAR, or radar to generate 3D bounding boxes, crucial for predicting shape, size, and position. Fusion of 2D and 3D information enhances depth perception. Some methods leverage monocular image-based approaches for 2D to 3D lifting in recent developments in 3D object detection.

6.4.3 Milestones in Object Detection and Comparison of Detection Methods:

Evolution of Object Detection:

Object detection has evolved over 20 years, with distinct periods before and after the introduction of Deep Learning in 2014.

Before 2014 – Traditional Object Detection:

1. Viola-Jones Detector (2001): Pioneering work initiating traditional methods.

2. HOG Detector (2006): Popular feature descriptor in computer vision.

3. DPM (2008): Introduction of bounding box regression.

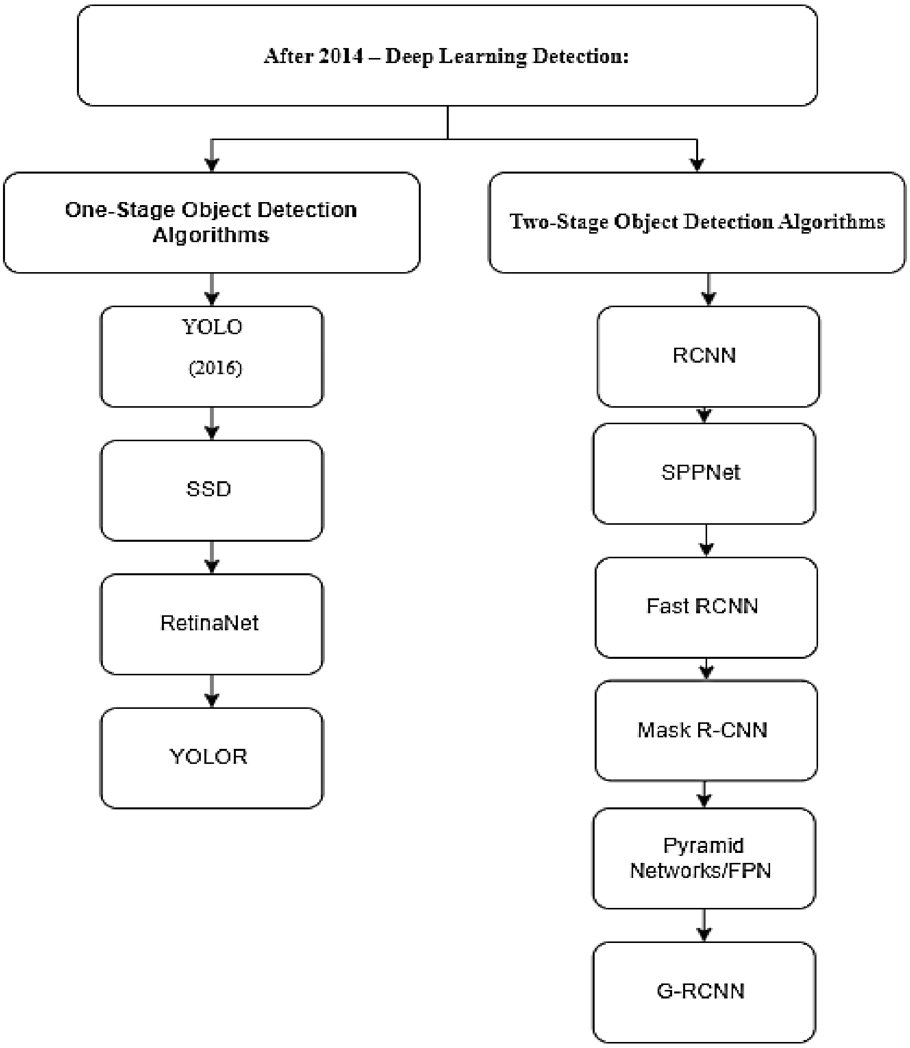


Figure 6- 10: Block Diagram of Examples of One stage and Two stage Object Detection Diagram.

Object Detection Dimensions, One-stage vs. Two-stage:

Object detection involves two tasks: finding objects and classifying them. These tasks are either separated into two stages or combined into one.

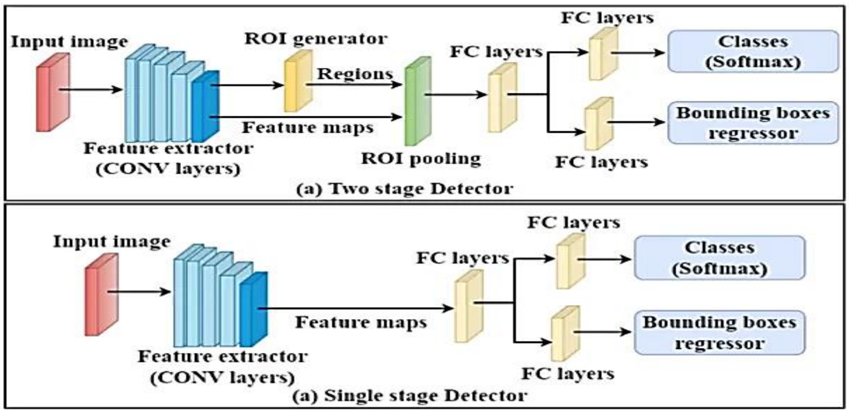


Figure 6- 11: Two-stage vs Single stage detector network diagram [20].

Two-Stage Detectors:

- Propose object regions before classification.
- Examples include RCNN, Faster R-CNN, and G-RCNN.
- Achieve high accuracy but are slower due to multiple inference steps.

One-Stage Detectors:

- Predict bounding boxes directly.
- Examples include YOLO, SSD, and RetinaNet.
- Prioritize speed and efficiency.

Comparison of Object Detection Algorithms:

- Microsoft COCO dataset is a popular benchmark.
- Evaluation based on Mean Average Precision (mAP).

Best Real-time Object Detection Algorithm (Accuracy):

On MS COCO dataset, YOLOv7 is the top performer, followed by ViT, PP-YOLOE, YOLOR, YOLOv4, and EfficientDet.

Fastest Real-time Object Detection Algorithm (Inference time):

In terms of inference time (ms/Frame), YOLOv7 outperforms others, achieving 3.5ms per frame. YOLOv4 has 12ms, YOLOv3 has 29ms, and Mask R-CNN has 333ms.

YOLO Algorithm Comparison (Accuracy and Inference Time):

Comparing YOLO versions, YOLOv8 (2023) demonstrates the best performance in real-time benchmarks, surpassing YOLOv7 and YOLOv6.

6.4.4 Object Detection: Use Cases and Applications:

Diversity in Use Cases:

Object detection is a versatile technology with diverse applications, revolutionizing how computers perceive and understand their surroundings.

Core of Vision-Based AI:

Object recognition forms the backbone of many vision-based AI programs, contributing to scene understanding across various industries.

Advantages and Disadvantages:

Object detectors offer remarkable flexibility for various tasks and custom applications, enabling automatic identification of objects, people, and scenes. This automation proves valuable for tasks like counting, inspection, and verification in business value chains. However, a significant drawback is their computational expense, demanding substantial processing power. When deployed at scale, operating costs can rise, challenging the economic viability of business use cases.

Key Applications:

1. Retail:

- People counting systems strategically placed in retail stores provide insights into customer behavior.
- AI-based customer analysis optimizes store layouts and enhances operational efficiency.
- Detection of queues reduces waiting times, improving customer experience.

2. Autonomous Driving:

- Essential for self-driving cars to recognize pedestrians, traffic signs, and other vehicles.
- Tesla's Autopilot AI heavily relies on object detection to perceive environmental threats.

3. Agriculture:

- Utilized for counting, monitoring, and evaluating the quality of agricultural products.
- Detects damaged produce during processing using ML algorithms.

4. Security:

- Widely used in video surveillance for security applications.
- Detects people in restricted or dangerous areas, aids in suicide prevention, and automates inspection tasks in remote locations.

5. Transportation:

- Vehicle detection with AI for traffic analysis and identifying cars stopping in hazardous areas.
- Enhances safety on crossroads and highways.

6. Healthcare:

- Revolutionizes medical diagnostics by studying images, scans, and photographs.
- Object detection in CT and MRI scans aids in disease diagnosis, such as tumor detection using ML algorithms.

Overall Impact:

Object detection is integral in automating manual tasks, creating new AI-powered products, and enhancing services across various industries, from sports production to productivity analytics.

6.5 AI MODELS FOR TRAFFIC SIGN RECOGNITION:

6.5.1 Overview of Object Detection Algorithms:

Some of AI Models:

- YOLO (You Only Look Once)
- SSD (Single-shot detector)
- Squeezed
- Mobile Net
- YOLOR (You Only Learn One Representation)

Overall Significance:

These algorithms, with their unique features and innovations, cater to diverse needs in computer vision, from real-time applications to specific use cases like autonomous driving and healthcare diagnostics.

6.5.2 AI Models Comparison:

Table 6- 1: Comparison between AI models of object detection.

Model	Explanation	Work	Performance	Power Efficiency	FPS	Process Stages	Accuracy
YOLO	Real-time object detection algorithm	YOLOv4, YOLOv5, YOLOv7, YOLOv8	High accuracy with real-time inference	Generally efficient	High	Single-stage detector	Balanced speed and accuracy across tasks
SSD	Faster single-shot detector	Discretizes output space of bounding boxes	Much better accuracy, especially with smaller input image sizes	Very efficient towards more accurate and faster detection	Higher performance than YOLO, scoring over 74% mAP at 59 fps	One-stage detection	Much better accuracy and faster detection, Low cost
Squeezed	Object detection model for autonomous driving	Specifically developed for autonomous driving	Fast detection with streamlined pipeline	Optimized efficiency	Real-time	Single-stage detector	Balances speed and accuracy for real-time detection
Mobile Net	Single-shot multi-box detection network	Implemented using the Caffe framework	Tracked object data in vector format with efficiency	Optimized for low-power devices	Real-time	Single-stage detector	Accurate detection suitable for mobile deployment
Efficient	Compound scaling for efficient models	Efficient network architectures	Balances accuracy and efficiency with high mAP	Designed for resource optimization	Generally efficient	Multi-level feature pyramid enhances detection	State-of-the-art efficiency, Balances speed and accuracy
CenterNet	Directly regresses the center point of objects	Predicts object centers and regresses bounding boxes	Achieves competitive accuracy with a focus on simplicity	Generally efficient in terms of both accuracy and speed	Processing speed details may vary	Single-stage detection with key point estimation	Accurate localization of object centers, Simplified architecture
DETR	Uses transformers for set-based object detection	Utilizes transformer architecture	Known for handling object detection as a set prediction problem	Efficient design leveraging transformers	Processing speed details may vary	Single-stage detection treating detection as an object set prediction	Handles varying numbers of objects per image, Robust to object occlusion

6.5.3 YOLO Model Review:

The YOLO series revolutionizes real-time object detection. YOLO's unique approach reframes object detection as a regression problem, optimizing the entire pipeline end-to-end. By dividing images into grids, YOLO performs detection and localization efficiently. Despite generating duplicate predictions, Non-Maximal Suppression mitigates noise, retaining high probability bounding boxes. The architecture comprises three components: backbone, neck, and head, leveraging convolutional layers for feature extraction, fully connected layers for predictions, and interchangeable heads for transfer learning.

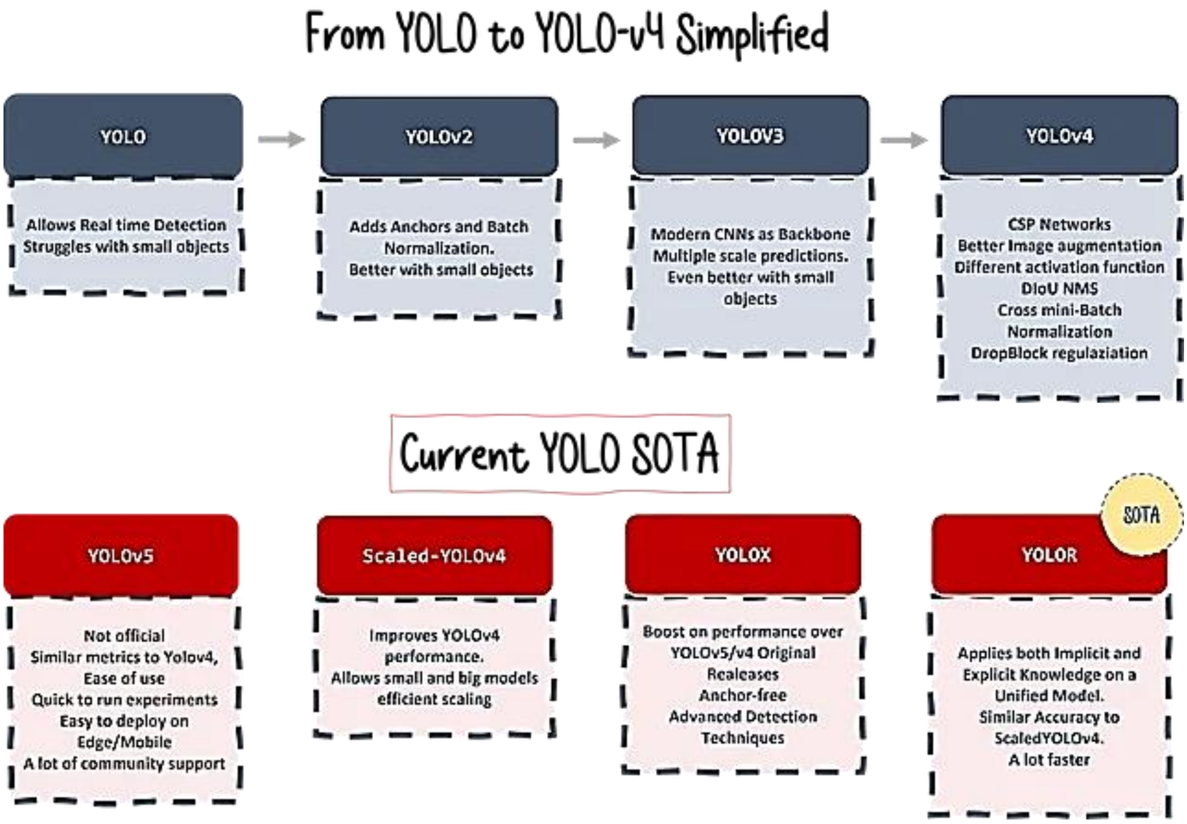


Figure 6- 12: All these models and improvements can be hard to grasp and compare so, a small, simplified summary [20].

In conclusion, YOLOv5 and its derivatives are lauded for their excellence in object detection, specifically in traffic sign recognition. However, the choice of the "best" model depends on specific requirements, dataset characteristics, and computational considerations.

6.5.4 SSD Model Overview:

The Single Shot Multibox Detector (SSD) revolutionizes object detection with its innovative architecture:

1. Multiscale Feature Maps:

- Utilizing feature maps at various scales enables SSD to capture contextual information, facilitating effective detection of objects of diverse sizes.

2. Default Boxes (Priors):

- Integration of default boxes at different aspect ratios and scales enhances SSD's adaptability to varied object dimensions, acting as anchors during detection.

3. Convolutional Filters:

- Strategically designed convolutional layers predict both object class and bounding box coordinates concurrently, streamlining the detection process for improved speed and accuracy.

6.5.5 Model Comparison Review:

- **Comparative Analysis of YOLO and SSD Models:**

In the landscape of object detection, the choice between YOLO and SSD models involves a nuanced consideration of their strengths and trade-offs. While YOLO excels in real-time processing and adaptability, SSD brings unique attributes to the table.

Advantages of SSD:

- Multiscale Detection: SSD's multiscale feature maps contribute to robust detection across a wide range of object sizes.
- Anchor Mechanism: The use of default boxes with various aspect ratios enhances SSD's ability to handle diverse object shapes.
- Simultaneous Prediction: Predicting class probabilities and bounding box coordinates simultaneously streamlines the detection pipeline, contributing to efficiency.

6.5.6 Implementation Considerations

6.5.6.1 Working with AI Models for Object Detection:

To deploy an object detection model on Nvidia Jetson, the following steps can be undertaken:

6. Choose an efficient model tailored for low-power hardware, such as Efficient, SSD-Mobilenet, Tiny-YOLO, or YOLOX.
7. Perform transfer learning on the selected model using custom data.
8. Convert the trained. pb file to Intermediate representation (.xml and .bin) using Model Optimizer.
9. Deploy the hardware-optimized models on Raspberry Pi.

6.5.6.2 Considerations for Real-time Performance and Accuracy:

- YOLO's Real-Time Edge: YOLO's architecture prioritizes real-time processing, making it well-suited for applications where speed is paramount, such as autonomous vehicles.
- SSD's Multiscale Strengths: SSD's ability to detect objects at multiple scales makes it a strong contender for scenarios with varying object dimensions.
- Trade-offs: The choice between YOLO and SSD depends on the specific requirements of the application, balancing factors like real-time performance, adaptability, and detection accuracy.

In the subsequent sections, a more in-depth analysis of YOLO and SSD models, their respective architectures, and performance metrics will be explored to provide comprehensive insights into their applicability and effectiveness in the domain of object detection, with a particular focus on traffic sign recognition.

Table 6- 2: Comparison between different 2D-3D models of object detection [21].

Name	Year	Type	Dataset	mAP	Inference rate (fps)
R-CNN	2014	2D	Pascal VOC	66%	0.02
Fast R-CNN	2015		Pascal VOC	68.80%	0.5
Faster R-CNN	2016		COCO	78.90%	7
YOLOv1	2016		Pascal VOC	63.40%	45
YOLOv2	2016		Pascal VOC	78.60%	67
SSD	2016		Pascal VOC	74.30%	59
RetinaNet	2018		COCO	61.10%	90
YOLOv3	2018		COCO	44.30%	95.2
YOLOv4	2020		COCO	65.70%	62
YOLOv5	2021		COCO	56.40%	140
YOLOR	2021		COCO	74.30%	30
YOLOX	2021		COCO	51.20%	57.8
Complex-YOLO	2018	3D	KITTI	64.00%	50.4
Complexer-YOLO	2019		KITTI	49.44%	100
Wen et al.	2021		KITTI	73.76%	17.8
RAANet	2021		NuScenes	62.00%	

6.5.6.3 Real-time Traffic Sign Recognition with YOLOv5:

The YOLOv5 model stands out as the optimal choice for real-time traffic sign recognition due to its optimized detection speed and accuracy. It strikes a superior balance between the two, outperforming other models. Validation accuracy for traffic sign detection reaches an impressive 99%, showcasing YOLOv5's prowess. Its improved algorithm not only enhances accuracy but also reduces parameters and computational costs, making it well-suited for real-time recognition. Particularly, the lightweight version (YOLOv5s) has demonstrated exceptional efficiency and accuracy in this domain.

- Widely recognized and efficient object detection model.
- Improved versions like YOLOv7 and YOLOv5-based models show advancements in small traffic sign detection.

1. Real-Time Processing:

Engineered for real-time applications, YOLOv5 ensures swift and efficient object detection, a critical aspect for prompt decision-making in traffic scenarios.

2. Single Shot Detection:

YOLOv5's single-shot detection processes the entire image in one pass, making it adept at scenarios like traffic sign detection where quick identification is imperative.

3. Versatility:

YOLOv5's adaptability to various object detection tasks and its capacity to handle diverse object classes render it suitable for recognizing a wide spectrum of traffic signs.

4. Model Size Options:

Offering different model sizes, YOLOv5 accommodates varying requirements, allowing users to strike a balance between speed and accuracy.

5. Training Enhancements:

Advanced training strategies like mosaic data augmentation and cut mix contribute to improved model generalization, crucial for diverse traffic sign recognition scenarios.

6. Efficient Backbone Architecture:

YOLOv5's choice of efficient backbone architectures enhances its capability to capture relevant features from traffic signs of different sizes and shapes.

7. Open-Source and Community Support:

YOLOv5's open-source nature facilitates customization for specific traffic sign datasets, with active community support ensuring continuous development and knowledge sharing.

8. Adaptability to Custom Datasets:

YOLOv5's ability to be trained on custom datasets enhances its adaptability to unique characteristics of traffic signs in different regions.

9. Pre-Trained Models:

Provision of pre-trained models allows YOLOv5 to serve as a starting point for traffic sign recognition projects, with potential for further enhancement through fine-tuning.

In conclusion, YOLOv5's combination of real-time processing, single-shot detection, versatility, and adaptability to custom datasets positions it as a compelling solution for traffic sign recognition. Its efficient architecture and strong community support contribute to its effectiveness in addressing real-world challenges.

6.6 DATASET:

6.6.1 Importance and Role

- **Foundations of Dataset Significance:** Highlighting the crucial role of comprehensive datasets in training and evaluating models for traffic sign recognition.
- **Model Generalization:** Emphasizing how diverse datasets enhance model adaptability and performance in real-world scenarios.
- **Addressing Challenges:** Discussing how datasets encompassing various conditions improve model robustness.
- **Ethical Considerations:** Exploring the importance of unbiased datasets in addressing ethical concerns in model outputs.
- **Dataset Source and Selection:** Guidelines for selecting datasets, including considerations for geographical diversity.

6.6.2 Selection Criteria and Augmentation Techniques

- **Representation of Real-world Scenarios:** Prioritizing datasets with diverse environmental and geographical representations.
- **Inclusion of Diverse Traffic Signs:** Ensuring datasets cover a broad range of traffic sign types.
- **Consideration of Illumination Variations:** Including scenarios with different lighting conditions for robustness.
- **Dataset Size and Scalability:** Discussing the importance of dataset size and scalability for effective model training.
- **Annotation Quality and Consistency:** Emphasizing the need for accurate and consistent annotations.
- **Incorporation of Challenging Scenarios:** Including challenging scenarios for model resilience.

6.6.3 Dataset Selection and Utilization

- **Utilizing Existing Datasets:** Exploring options to leverage curated and annotated datasets from platforms such as Roboflow.

- **Selection Criteria for Traffic Sign Recognition Datasets:** Criteria for selecting datasets to ensure representation of real-world scenarios, diverse traffic signs, illumination variations, dataset size, annotation quality, and incorporation of challenging scenarios.
- **Dataset Augmentation Techniques:** Techniques for enhancing dataset diversity, including synthetic data generation and active learning strategies.
- **Advancements in Annotation Tools:** Discussing future advancements in annotation tools for improved dataset quality.
- **Collaborative Dataset Creation Efforts:** Highlighting the potential benefits of community collaboration in dataset creation.
- **Future Directions in Dataset Development:** Discussing advancements in annotation tools and collaborative dataset creation efforts for improving dataset quality and diversity.

6.7 METHODOLOGY:

6.7.1 Training and Testing AI Models:

In this phase, the suggested AI models will undergo training and testing processes. The training phase involves feeding the models with labeled datasets to learn patterns and features relevant to traffic sign recognition. Subsequently, the trained models will be rigorously tested using separate datasets to evaluate their performance metrics, such as accuracy, precision, and recall.

1. **YOLO:** A fast and accurate real-time object detection model suitable for traffic sign recognition tasks, with optimizations available for deployment on edge devices like the NVIDIA Jetson Nano 4GB.

2. **SSD:** An efficient real-time object detection model known for its speed and accuracy, with lightweight variants that can run efficiently on embedded devices like the Jetson Nano.

3. **MobileNet:** A lightweight CNN architecture optimized for inference on mobile and embedded devices, including the NVIDIA Jetson Nano, making it suitable for ADAS systems with resource constraints.

4. **TinyYOLO:** A compact version of the YOLO model designed for real-time inference on edge devices, including the Jetson Nano, offering a good balance of speed and accuracy for traffic sign recognition.

5. **SqueezeDet:** An ultra-lightweight object detection model optimized for deployment on edge devices, with a focus on efficiency and speed, making it suitable for real-time applications on the Jetson Nano.

These models are compatible with the NVIDIA Jetson Nano 4GB model and offer a range of options suitable for traffic sign recognition tasks on edge devices.

6.7.2 Preprocessing Steps for Real-time Applications:

Preprocessing plays a crucial role in optimizing model performance by preparing input data in a suitable format. Details regarding image preprocessing techniques, such as [1] normalization, resizing, and data augmentation, will be discussed here. Additionally, a diagram illustrating the preprocessing workflow will be provided to facilitate a better understanding of the process.

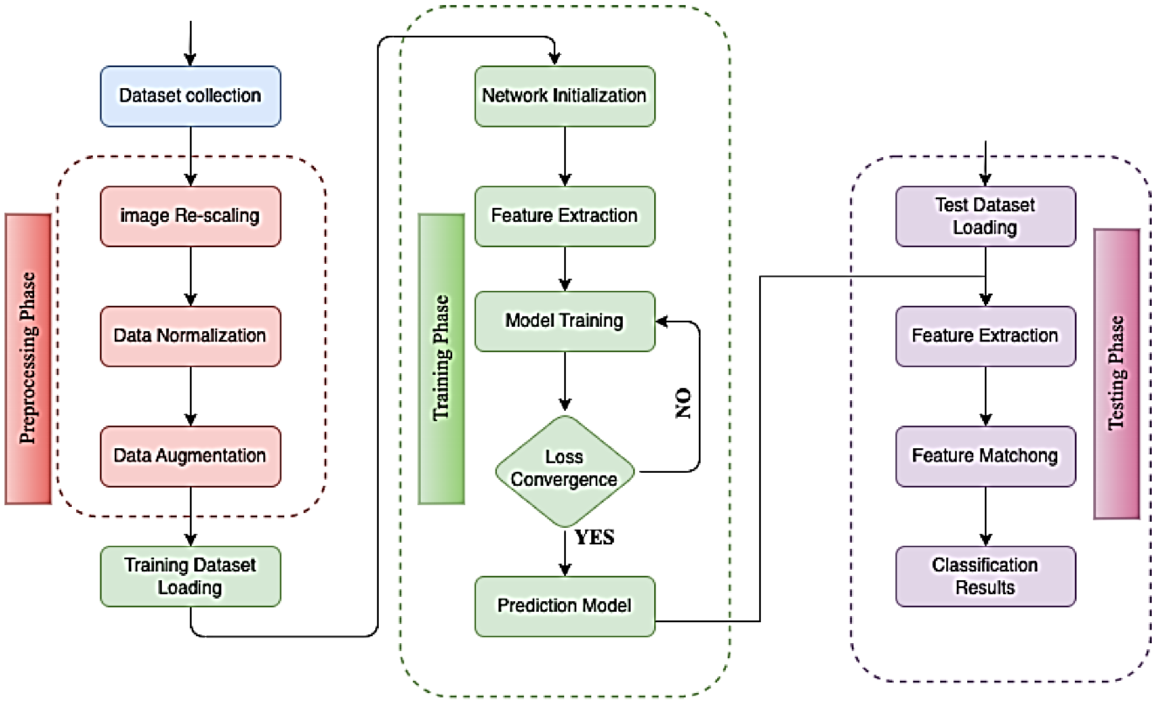


Figure 6- 13: Phases of the Preprocessing Steps for Real-time [19].

Preprocessing Steps for Real-time Applications on NVIDIA Jetson Nano:

1. **Input Acquisition:** Obtain input frames from the camera.

2. **Resize & Normalize:**
   - Resize frames to model-compatible dimensions.
   - Normalize pixel values for standardized input.
3. **Optional Data Augmentation:** Apply random transformations for data diversity.
4. **Convert to Tensor:** Transform preprocessed images into tensors.
5. **Optional Batching:** Group processed frames for parallel processing.

6.7.3 Alerts and Notifications:

Alert and Action Mechanism:

The system responds to different traffic signs, e.g., alerting the driver if a speed sign is detected, and the vehicle is moving above the limit. The goal is to prevent accidents and improve road safety.

Benefits of the System:

- Early detection of potential hazards.
- Real-time feedback for driver adjustments.
- Overall improvement in road safety.

6.8 EXPERIMENTS:

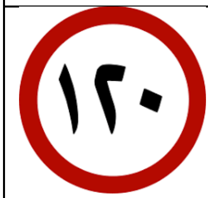
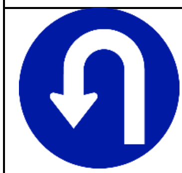
6.8.1 Presentation of Experimental Results:

Table 6-3 of Dataset Comparison between Traffic Signs and Interactions of System: This table provides a comparison of datasets relevant to traffic sign recognition and system interactions, aiding in the selection process.

Table 6- 3: Dataset Comparison between Traffic Signs and Interactions of System [18].

Sign Image	Sign Name	Sign Type	Description	Model Action/Alert
 No entry for vehicular traffic	Do Not Enter	Regulatory	Regulatory sign indicating entry is prohibited.	Display alert to avoid entering.
 No U-turns	Do Not U-Turn	Regulatory	Regulatory sign indicating U-turns are not allowed.	Display alert to avoid making U-turns.
 Traffic signals	Green Traffic Light	Regulatory	Regulatory signal indicating the permission to proceed.	No specific alert (normal driving).
	Mini-Roundabout	Regulatory	Regulatory sign indicating a mini roundabout ahead.	Display alert for upcoming mini roundabout.
	No Parking	Regulatory	Regulatory sign indicating parking is not allowed.	Display alert for parking violation.
	Parking Place	Regulatory	Regulatory sign indicating designated parking area.	Display alert for approaching designated parking area.
	Pedestrian Crossing	Warning	Warning sign indicating a pedestrian crossing ahead.	Display alert to watch for pedestrians.

	Red Traffic Light	Regulatory	Regulatory signal indicating the need to stop.	Display alert to stop the vehicle.
	Speed Limit 20	Regulatory	Circular sign with a black number indicating a maximum speed limit of 20 km/h.	Display alert for exceeding speed limit.
	Speed Limit 30	Regulatory	Circular sign with a black number indicating a maximum speed limit of 30 km/h.	Display alert for exceeding speed limit.
	Speed Limit 40	Regulatory	Circular sign with a black number indicating a maximum speed limit of 40 km/h.	Regulatory | Display alert for exceeding speed limit.
	Speed Limit 50	Regulatory	Circular sign with a black number indicating a maximum speed limit of 50 km/h.	Display alert for exceeding speed limit.
	Speed Limit 60	Regulatory	Circular sign with a black number indicating a maximum speed limit of 60 km/h.	Display alert for exceeding speed limit.
	Speed Limit 70	Regulatory	Circular sign with a black number indicating a maximum speed limit of 70 km/h.	Display alert for exceeding speed limit
	Speed Limit 80	Regulatory	Circular sign with a black number indicating a maximum speed limit of 80 km/h.	Display alert for exceeding speed limit
	Speed Limit 90	Regulatory	Circular sign with a black number indicating a maximum speed limit of 90 km/h.	Display alert for exceeding speed limit

	Speed Limit 100	Regulatory	Arabic Circular sign with a black number indicating a maximum speed limit of 100 km/h.	Display alert for exceeding speed limit
	Speed Limit 100	Regulatory	Circular sign with a black number indicating a maximum speed limit of 100 km/h.	Display alert for exceeding speed limit
	Speed Limit 120	Regulatory	Arabic Circular sign with a black number indicating a maximum speed limit of 120 km/h.	Display alert for exceeding speed limit
	Speed Limit 120	Regulatory	Circular sign with a black number indicating a maximum speed limit of 120 km/h.	Display alert for exceeding speed limit
	U-Turn Allowed	Regulatory	Regulatory sign indicating U-turns are allowed. | Regulatory	Display alert to make a U-turn if needed.
 Traffic signals	Traffic Light - Yellow	Regulatory	Regulatory signal indicating the approaching signal will change.	Display alert to prepare for a change in the traffic light
 Stop and give way 	Stop	Regulatory	Regulatory sign indicating the need to come to a complete stop.	Display alert to stop the vehicle.

	Exclamation Mark	Warning	Warning sign indicating caution or potential danger ahead.	Display alert to proceed with caution.
	Bump	Warning	Warning sign indicating a bump or uneven road surface ahead.	Display alert to prepare for an uneven road surface.

6.8.2 Evaluation Metrics for AI Model Performance:

Key Metrics and Object Detection Dimensions: A Brief Overview Definition of Terms:

- **True Positive (TP):** Correct detection.
- **False Positive (FP):** Incorrect detection.
- **False Negative (FN):** Ground-truth missed by the detector.

IoU Metric: IoU evaluates the overlap between the ground truth and predictions, ranging from 0 to 1. A score of 1 indicates a perfect overlap.

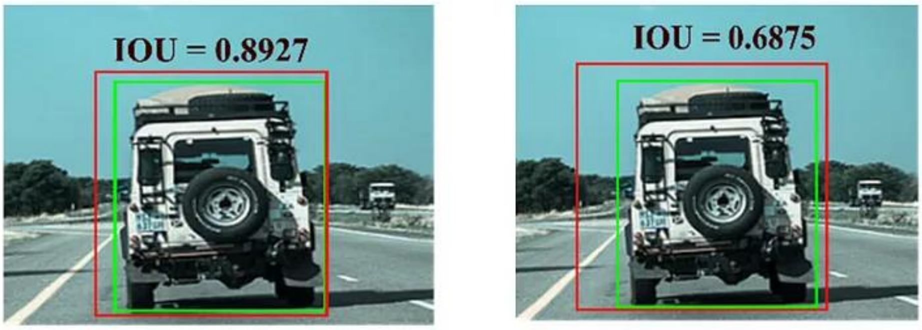


Figure 6- 14: Example of an IOU; green box: ground truth; red box: prediction [20].

Precision and Recall:

- **Precision:** Proportion of correct positive identifications.
- **Recall:** Proportion of actual positives correctly identified.

Equations for Precision and Recall:

In a vehicle object detector, precision answers how many predicted cars were cars, while recall indicates how many cars the model identified.

$$\text{Precision} = \frac{TP}{TP+FP}$$
$$\text{Recall} = \frac{TP}{TP+FN} \tag{6. 2}$$

Average Precision (aP):

Evaluated at an IoU threshold, it represents the area under the precision-recall curve.

$$AP@\alpha = \int_0^1 p(r)dr \tag{6. 3}$$

Mean Average Precision (mAP):

For multi-class object detectors, mAP gives the mean AP across all classes. Recent advancements, mainly due to deep learning, have significantly increased mAP from 30% to over 90% in 2018.

$$maP = \frac{\sum_{q=1}^{Q} AP(q)}{Q} \tag{6. 4}$$

Overall Dataset Summary with Metrics: Presenting a summary of the dataset's performance metrics and characteristics.

6.9 CHALLENGES AND LIMITATIONS:

6.9.1 Identification and Discussion of Challenges:

Developing a custom dataset for traffic sign recognition using YOLOv5 presents several challenges:

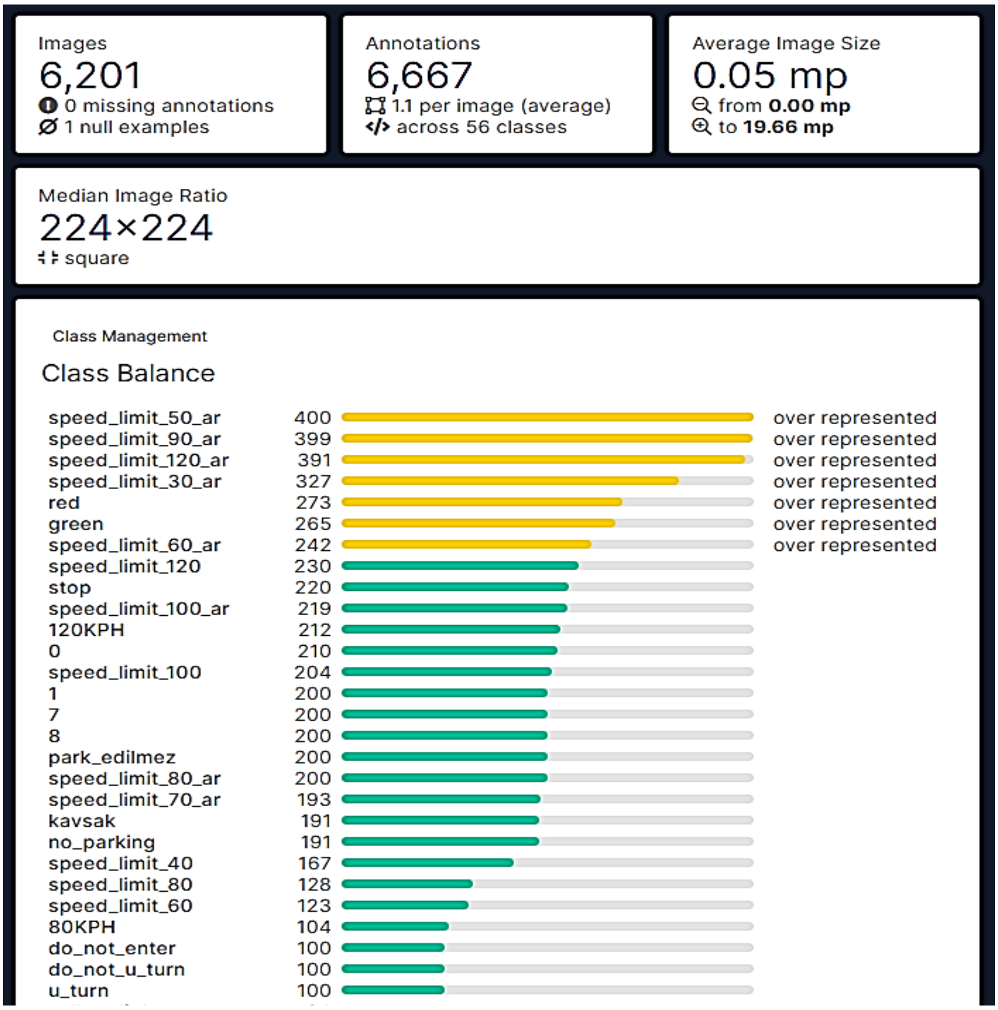


Figure 6- 15: All Classes of The Dataset and the overall of dataset summary [5].

1. **Custom Dataset Creation:**
   - Creating a diverse and accurately annotated dataset is time-consuming.
2. **Annotation Quality:**
   - Ensuring precise and consistent annotations is crucial for model training.
3. **Data Augmentation:**
   - Augmenting the dataset with varied transformations enhances model generalization.
4. **Class Imbalance:**
   - Balancing traffic sign classes minimizes performance biases.
5. **Integration with YOLOv5:**
   - Ensuring dataset compatibility and optimization with the model is vital.

6. **Validation and Evaluation:**
   - Thorough validation and benchmarking against metrics ensure dataset reliability.

6.9.2 Limitations of Existing AI Models:

Considerations regarding YOLOv5 for traffic sign recognition:

1. **Sensitivity to Dataset Quality:**
   - Model performance depends heavily on dataset quality and diversity.
2. **Generalization Across Conditions:**
   - The model may struggle with diverse environmental factors.
3. **Fine-tuning Challenges:**
   - Adapting the model to specific platforms like NVIDIA Jetson Nano requires optimization.
4. **Speed vs. Accuracy Trade-offs:**
   - Optimizing for real-time processing while maintaining accuracy is challenging.

5. **Exploration of Alternatives:**
   - Investigating alternative architectures may lead to further advancements.

Addressing these challenges through systematic experimentation and innovation is essential for advancing ADAS systems in traffic sign recognition.

6.10 FUTURE DIRECTIONS:

6.10.1 Future Enhancements:

The continuous evolution of Traffic Sign Interpretation in ADAS presents opportunities for significant improvements:

1. <u>Integration with Advanced Algorithms:</u>

   - Explore advanced ML and deep learning integration to enhance accuracy and speed.
   - Implement reinforcement learning for adaptive system performance.

2. Multimodal Sensor Fusion:
   - Integrate data from various sensors for a comprehensive road environment understanding.
   - Utilize sensor fusion for improved performance in adverse conditions.
3. Augmented Reality (AR) Integration:
   - Incorporate AR to overlay traffic sign information onto the driver's field of view.
   - Implement AR-based visual cues for intuitive driver interactions.
4. Dynamic Sign Interpretation:
   - Develop capabilities to interpret dynamic or temporary traffic signs for changing road conditions.
   - Implement real-time updates for temporary signs using vehicle-to-infrastructure communication.

10. Customizable Alert Systems:
    - Allow drivers to customize alert preferences based on driving habits.
    - Provide adaptive alerting mechanisms considering individual responsiveness.
11. Cloud-Based Learning and Updates:
    - Implement cloud-based learning for continuous system updates and improvements.
    - Enable over-the-air updates to ensure compliance with the latest regulations.
12. Enhanced Human-Machine Interaction:
    - Introduce natural language processing for voice-based interactions.
    - Improve user interface for clear and concise visual feedback.
13. Collaborative Driving Ecosystem:
    - Explore collaborative driving ecosystems for real-time exchange of traffic sign data.
    - Implement vehicle-to-vehicle communication for enhanced road safety.

6.10.2 The importance of Traffic Sign Interpretation in ADAS for advancing road safety:

Traffic Sign Interpretation in ADAS significantly enhances road safety through:

- **Driver Awareness:** Providing real-time information about road signs maintains driver awareness.
- **Hazard Avoidance:** Assisting drivers in avoiding potential hazards with accurate sign interpretation.
- **Adaptive Driving:** Enabling adaptive driving behaviours based on recognized sign information.
- **Violation Prevention:** Notifying drivers about speed limits and regulations to prevent violations.
- **Accident Reduction:** Contributing to a decrease in accidents by providing essential information.
- **Autonomous Driving:** Crucial for ensuring compliance in autonomous vehicles, enhancing safety.
- **Overall, Road Safety:** Contributes to a safer driving environment for all road users.

In summary, Traffic Sign Interpretation in ADAS plays a vital role in promoting responsible driving, preventing accidents, and advancing road safety through intelligent assistance systems.

Chapter (7)

7. BUMP DETECTION

7.1 INTRODUCTION:

Bump detection stands as a fundamental technology within the realm of ADAS, offering a transformative enhancement to both vehicular safety and passenger comfort. Its inception is rooted in the imperative recognition of the challenges posed by uneven road surfaces, encompassing an array of irregularities such as potholes, speed bumps, and unexpected variations in terrain.

The core motivation behind the development of bump detection technology is to mitigate potential hazards and discomfort by identifying and responding to these road irregularities. Operating at the nexus of sensor technologies, signal processing, and advanced algorithms, bump detection systems employ a spectrum of sensors such as inertial sensors, cameras, LiDAR, and radar to meticulously capture data on the vehicle's motion and its immediate surroundings. Through intricate signal processing techniques, often coupled with ML algorithms, these systems analyze the collected data to discern patterns associated with bumps. The primary objectives of bump detection encompass enhancing safety by facilitating real-time adjustments to the vehicle's suspension system, thereby minimizing the risk of accidents, and optimizing passenger comfort through the proactive adaptation to road irregularities.

Comprising components such as sensors, signal processing modules, algorithms, and actuators that adjust the vehicle's suspension, bump detection systems navigate challenges such as accurate differentiation between normal road variations and significant bumps, adaptability to diverse road conditions, and real-time responsiveness. Calibration and validation processes are integral to refining the system's parameters, ensuring accuracy, and bolstering reliability. With applications spanning across ADAS, autonomous vehicles, and intelligent transportation systems, bump detection technology signifies a critical stride towards redefining the driving experience by prioritizing safety and comfort on the modern roadways.

The car identifies bumps or obstacles in its path. When it detects such obstacles, the onboard computer can reduce the speed of the vehicles or provide feedback to the driver through warnings on an interactive surface [22].

7.2 PROBLEM OF BUMPS IN EGYPT:

In Egypt, most bumps are not built or maintained with public safety in mind. This results in vehicle damage, driver discomfort, and potential loss of control, leading to fatal accidents. Speed bumps are haphazardly installed without proper engineering studies. Figure 7-1 illustrates the fluctuation in speed and extreme decelerations and accelerations before and after speed bumps, even over short distances. This instability in travel speed leads to increased travel time, vehicle damage, passenger discomfort, higher fuel consumption, increased pollution, and pavement deterioration.

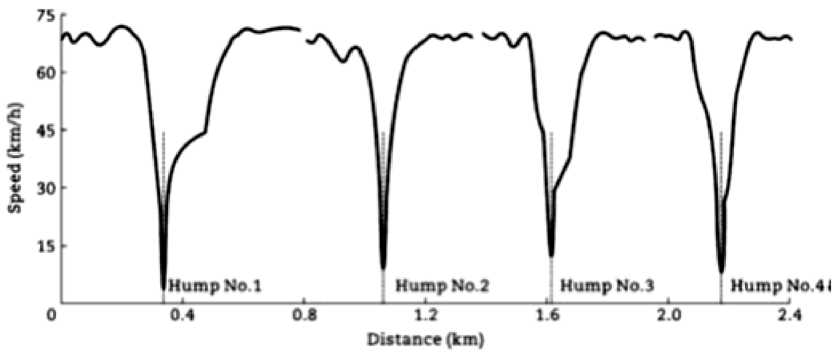


Figure 7- 1: Speed profile for a road section using GPS [23].

Road anomalies have always been a bad sign of road quality. Rough roads, speed bumps, and potholes endanger the safety and comfort of car drivers, so this project focuses on the problem of bumps in Egypt and provide a solution for it [23].

7.3 PROJECT IDEA:

To solve the problem of road bumps in Egypt, a project can be undertaken to implement bump detection using the Astra Pro Plus camera, TF Luna sensor, and NVIDIA Jetson Nano. The following steps can be taken to solve the problem:

- **System Design:** Develop a comprehensive system design that integrates the Astra Pro Plus camera, TF Luna sensor, and NVIDIA Jetson Nano. This design should include the necessary hardware setup and connections.

- **Data Collection:** Collect a diverse dataset of road bump images and sensor readings from various locations in Egypt. This dataset will serve as the training data for the bump detection algorithm.
- **Algorithm Development:** Utilize deep learning techniques and computer vision algorithms to develop an efficient and accurate bump detection algorithm. Train the algorithm using the collected dataset to enable it to recognize different types of road bumps.
- **Sensor Integration:** Integrate the TF Luna sensor with the NVIDIA Jetson Nano to provide additional depth information and enhance the accuracy of the bump detection system.
- **Hardware Implementation:** Install the Astra Pro Plus camera and TF Luna sensor on a robotic car, ensuring proper alignment and calibration. Connect the hardware components to the NVIDIA Jetson Nano for data processing.
- **Real-Time Processing:** Implement real-time data processing capabilities on the NVIDIA Jetson Nano to enable efficient and fast analysis of incoming data from the camera and sensor.
- **Bump Detection and Alert System:** Develop a user-friendly interface that displays real-time bump detection results. Implement an alert system that notifies drivers about the presence of road bumps to ensure timely actions can be taken.

7.4 PROJECT OBJECTIVE:

The objective of this thesis is to develop an algorithm that can be implemented on the onboard computer of a vehicle to accurately detect speed bumps on roads, regardless of various environmental conditions. Successful detection of speed bumps can be utilized to adjust the vehicle's chassis or initiate deceleration when driving with cruise control activated. Additionally, the information about speed bumps can be used to enhance maps or navigation systems. Addressing the significant variations in the appearance of different speed bumps is an essential consideration in this thesis. Some speed bumps are visually highlighted with vibrant colors or even have drawn arrows, as depicted in Figure 7-2(a). Furthermore, speed bumps can be constructed using various materials, such as interlocking pavement, as seen in Figure 7-2(b). In some cases, speed bumps are barely discernible even to human eyes, like the ones shown in Figure 7-3(c). Certain types of speed bumps marked speed humps help control vehicle speeds in areas where pedestrian safety is a concern shown in Figure 7-3(d) [24].


(a)


(b)

Figure 7- 2: (a) speed bump with drawn arrows. (b)speed bump using interlocking pavement.


(c)


(d)

Figure 7- 3: (c)speed bump not discernible. (d)speed bump helps control vehicle speeds in areas where pedestrian safety.

7.5 COMPONENTS AND BLOCK DIAGRAM OF THE SYSTEM:

7.5.1 Components:

- **Nvidia Jetson Nano:**

NVIDIA Jetson Nano Developer Kit is a small as shown in Figure 7-4, powerful computer that lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing. NVIDIA Jetson Nano module revolutionizes edge computing with 472 GFLOPS of accelerated computing, supporting modern neural networks. Ideal for AIoT gateways, cameras, robots, and more, it boasts a powerful Maxwell GPU, 4 GB memory, and diverse IOs. JetPack software ensures swift AI deployment, reducing complexity for autonomous machines [25].

- **Specifications of Jetson Nano:**
  - GPU: 128-core Maxwell™ GPU.
  - CPU: quad-core ARM® Cortex®-A57 CPU.
  - Memory: 4GB 64-bit LPDDR4.
  - Storage: Micro SD card slot (requires an external minimum 16G TF card).
  - Camera: 12 lanes (3x4 or 4x2) MIPI CSI-2 D-PHY 1.1 (1.5 Gb/s per pair).
  - GPIO, I2C, I2S, SPI, UART

Figure 7- 4: Nvidia Jetson Nano [30].

- **Astra Pro Plus Depth Camera:**

The Astra Pro Plus camera is a reliable solution for bump detection. Its high-resolution sensor captures detailed images, enabling precise identification of bumps or irregularities on surfaces. Equipped with advanced algorithms, the camera can analyze the captured images and compare them with predefined patterns or reference surfaces to identify any deviations. Real-time monitoring and integration flexibility further enhance its usability. Whether in manufacturing, automotive, or aerospace industries, the Astra Pro Plus camera provides accurate and efficient bump detection, ensuring quality control and surface inspection, the specification as chapter 5.

- **TF Luna:**

The TF Luna LiDAR is an excellent tool for bump detection due to its laser-based sensing technology. With its ability to measure distances with high accuracy and speed, the TF Luna can detect and analyze surface irregularities, including bumps. The LiDAR sensor emits laser pulses and measures the time it takes for the pulses to bounce back, creating a detailed 3D map of the environment. By comparing the acquired data with a reference surface, the TF Luna can identify and quantify bumps accurately. Its compact size, low power consumption, and compatibility with various platforms make it an ideal choice for applications such as autonomous vehicles, robotics, and industrial automation, the specification as chapter 5.

7.5.2 Block Diagram of Bump Detection System:

The block diagram provided consists of various components and modules that form a Bump Detection System for a car movement system. Here is an explanation of each block:

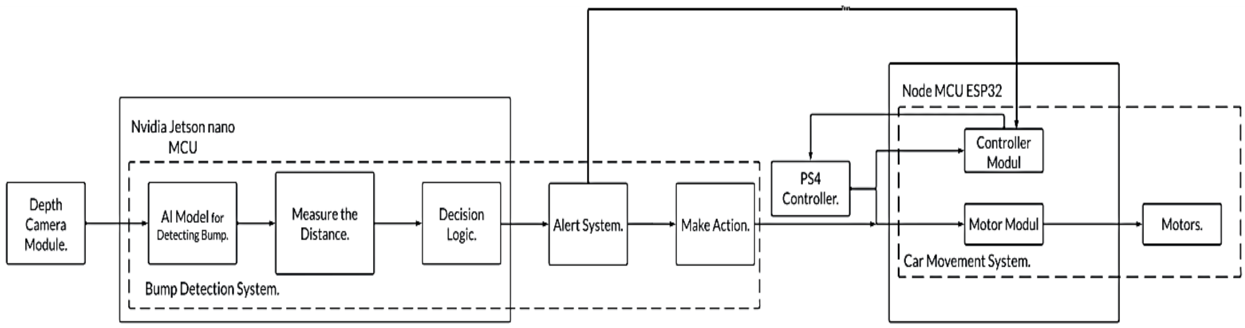


Figure 7- 5: Block Diagram of Bump Detection System.

- **Nvidia Jetson Nano:** It is a small, powerful computer designed for AI and deep learning tasks. It serves as the main processing unit for the system, handling the computation and decision-making processes.
- **MCU (Microcontroller Unit):** This block represents a microcontroller unit, which is a small computer on a single integrated circuit. It is responsible for controlling and monitoring various components of the system.
- **Depth Camera Module:** This module represents a depth camera, which is used to capture the depth information of the surroundings. It helps in accurately detecting bumps or obstacles in the path of the car.
- **AI Model for Detecting Bump:** This block signifies an AI model specifically designed to detect bumps. It processes the data from the depth camera module and identifies potential bumps or obstacles in the car's path.
- **Measure the Distance:** This block indicates the process of measuring the distance between the car and the detected bump or obstacle. It utilizes the depth information provided by the camera module to determine the distance accurately.
- **Decision Logic:** This block represents the decision-making process based on the information gathered from the AI model and distance measurement. It determines the appropriate actions to be taken by the system, such as slowing down, stopping, or changing the direction of the car to avoid the detected bump.

- **Bump Detection System:** This block encompasses all the components mentioned above and represents the overall system designed to detect bumps and ensure the safe movement of the car.
- **PS4 Controller:** This block represents a PlayStation 4 controller, which can be used as an input device for controlling the car movement system. It provides a user-friendly interface to interact with the system.
- **Node MCU ESP32:** This block signifies a microcontroller module based on the NodeMCU ESP32 platform. It serves as a controller module for the overall system, enabling communication and coordination between different components.
- **Alert System:** This block indicates the presence of an alert system, which can include visual or auditory indicators to alert the driver or surrounding individuals about the detected bump or potential danger.
- **Make Action:** This block represents the execution of appropriate actions by the system based on the decision made by the decision logic block. It controls the car's movement or initiates any necessary maneuvers to avoid the detected bump.
- **Motor Module:** This block signifies a motor module responsible for controlling the motors of the car. It receives commands from the system and translates them into motor movements, enabling the desired car movements.
- **Motors:** This block represents the actual motors present in the car. They are responsible for physically driving the wheels or other components of the car, based on the commands received from the motor module.

7.6 IMAGE RECOGNITION:

Image recognition is a vital part of digital advancement. It allows machines to interpret visual information, identifying objects and scenes. It has diverse applications in healthcare, autonomous vehicles, security, and augmented reality. Integration of image recognition brings automation, efficiency, and improved human-computer interactions. Advancements in algorithms and deep learning techniques improve accuracy and speed. Large datasets and computing power enable sophisticated image recognition systems. As the field of image recognition continues to advance, new challenges and opportunities arise. Researchers are constantly exploring novel algorithms and models to improve accuracy and speed. Deep learning techniques, such as convolutional neural networks, have significantly contributed to the progress in image recognition. Additionally, the availability of large-scale datasets and powerful computing resources has accelerated the development of more sophisticated image recognition systems.

7.6.1 Types of algorithms in image recognition:

- **Image Recognition-Based Algorithm**:

1. Data Collection:

Gather a diverse dataset of labeled images representing the objects or patterns you want the algorithm to recognize. Ensure that the dataset is well-balanced and covers various conditions and scenarios [26].

2. Image Preprocessing:

Preprocess the images to standardize their format and enhance relevant features. Common preprocessing steps include resizing, normalization, and augmentation to improve the algorithm's robustness.

3. Feature Extraction:

Extract meaningful features from the images. In traditional computer vision, this might involve handcrafted features like Histogram of Oriented Gradients (HOG) or Local Binary Patterns (LBP). In modern approaches, CNNs are commonly used for automatic feature extraction.

4. Model Selection:

Choose an appropriate image recognition model architecture. CNNs, such as VGG16, ResNet, or Mobile Net, are popular choices for image recognition tasks due to their ability to learn hierarchical features.

5. Model Training:

Train the selected model using the labeled dataset. Use a portion of the dataset for training and another portion for validation to assess the model's generalization. Adjust hyperparameters and optimize the model based on validation performance.

6. Fine-Tuning (Optional):

Fine-tune the pre-trained model on your specific task if a pre-trained model is used. Transfer learning can save computational resources and improve performance.

7. Validation and Testing:

Evaluate the trained model on a separate test dataset to assess its performance and generalization to new, unseen data. Use metrics such as accuracy, precision, recall, and F1 score to measure the model's effectiveness.

8. Thresholding and Decision Making:

Set a decision threshold based on the model's output probabilities. Adjust the threshold to control the trade-off between false positives and false negatives, depending on the specific requirements of your application.

9. Real-Time Inference:

Implement real-time inference on new, unseen images or video frames using the trained model. The algorithm should be capable of making predictions efficiently to support real-time applications.

10. Integration with Systems:

Integrate the image recognition algorithm into larger systems or applications. Depending on the use case, this could involve integration with robotics, surveillance systems, autonomous vehicles, or any other application requiring visual understanding.

11. Continuous Monitoring and Updating:

Implement mechanisms for continuous monitoring of the algorithm's performance in production. If necessary, update the model periodically with new data to adapt to changing conditions and improve accuracy.

12. Edge Cases Handling:

Account for and handle edge cases or scenarios where the algorithm may encounter challenges or produce uncertain predictions. This might involve additional modules or fallback mechanisms.

- **LiDAR-Based Bump Detection Algorithm:**

1. Data Acquisition:

Collect LiDAR point cloud data, which represents the surfaces and objects in the vehicle's vicinity.

2. Preprocessing:

Filter the raw point cloud data to remove noise, outliers, and non-ground points. This helps focus on relevant information for bump detection.

3. <u>Ground Segmentation:</u>

Separate ground points from non-ground points using segmentation techniques. This step helps distinguish the road surface from potential obstacles or bumps.

4. <u>Surface Normal Estimation:</u>

Compute surface normal for the remaining points. Surface normal provide information about the orientation of surfaces in the point cloud.

5. <u>Feature Extraction:</u>

Extract features from the point cloud data that are indicative of bumps. Features could include variations in surface curvature or abrupt changes in height.

6. <u>Thresholding:</u>

Set thresholds for the extracted features to distinguish between normal road variations and potential bumps.

7. <u>Bump Identification:</u>

Identify clusters or regions in the point cloud where the extracted features exceed the predefined thresholds. These clusters may represent areas with potential bumps.

8. <u>Context Awareness:</u>

Consider additional contextual information, such as vehicle speed or GPS data, to enhance the accuracy of bump detection and reduce false positives.

9. <u>Alert Mechanism:</u>

Implement an alert mechanism or interface with the vehicle's control systems to notify the driver or take appropriate actions based on the identified bumps.

10. <u>Integration with ADAS:</u>

Integrate the LiDAR-based bump detection algorithm into the broader ADAS system, ensuring compatibility with other sensors and safety features.

- **Hybrid Image Recognition and LiDAR-Based Algorithm:**

1. Data Collection:

Collect labeled training data that includes both images and corresponding LiDAR point cloud data. Labels should indicate the presence or absence of objects, road conditions, or specific features of interest.

2. Image Processing:

Preprocess the image data by resizing, normalizing, and augmenting images. Train an image recognition model (e.g., Convolutional Neural Network - CNN) on the labeled image dataset to detect and classify objects.

3. LiDAR Data Processing:

Preprocess the LiDAR point cloud data by filtering noise, segmenting ground points, and extracting relevant features such as surface normal or point density. Use LiDAR-specific algorithms (e.g., Point Net) to process and analyze the point cloud data.

4. Feature Fusion:

Combine features extracted from both the image recognition model and LiDAR processing. Fusion techniques can include concatenation, element-wise operations, or merging feature maps at different stages of the neural networks.

5. Joint Feature Representation:

Create a joint feature representation that encapsulates both image and LiDAR information. This combined feature representation is used to capture cross-modal relationships.

6. ML Model Training:

Train a joint ML model (e.g., a neural network) using the combined feature representation. This model learns to integrate information from both modalities for improved performance in tasks such as object detection or scene understanding.

7. Cross-Modal Calibration:

Address any calibration misalignments between the image and LiDAR sensors. Accurate calibration ensures proper alignment of features between different sensor modalities.

8. Validation and Hyperparameter Tuning:

Validate the performance of the model using a separate validation dataset. Fine-tune hyperparameters to optimize the model's accuracy and generalization to new data.

9. Thresholding and Decision Making:

Set threshold values for the model's output to make decisions based on the confidence levels of predictions. This step helps control the trade-off between sensitivity and specificity.

10. Real-Time Processing:

Implement real-time processing of both image and LiDAR data using the trained model. The system continuously analyzes incoming data streams to understand the environment, detect objects, or make navigation decisions.

11. Integration with Control Systems:

Integrate the algorithm with the control systems of a vehicle or a robotic platform. Use the output from the model to inform actions, such as adjusting the trajectory or avoiding obstacles.

12. Adaptability and Continuous Learning:

Design the algorithm to adapt to varying environmental conditions. Implement mechanisms for continuous learning to update the model based on new data and evolving scenarios.

In our project we use the third algorithm which is **Hybrid ML and LiDAR-Based Bump Detection Algorithm**, because it has the lowest error rate and the highest accuracy.

7.6.2 YOLOv5:

YOLO5, short for "You Only Look Once version 5," is an advanced object detection algorithm that has gained popularity in the field of computer vision. Building upon its predecessors, YOLO5 offers improved accuracy and speed in real-time object detection tasks. It utilizes a single neural network to simultaneously predict bounding boxes and class probabilities for multiple objects within an image. YOLO5 achieves this by employing a lightweight architecture, making it efficient for deployment on resource-constrained devices. With its ability to detect objects quickly and accurately, YOLO5 has found applications in areas such as autonomous driving, surveillance systems, and object recognition in videos and images [27].

1. Architecture and Innovation:

The architecture of YOLOv5 revolves around a one-stage object detection approach, where an input image is divided into a grid, and bounding boxes along with class probabilities are predicted directly. This streamlined methodology allows YOLOv5 to achieve remarkable efficiency without compromising accuracy. Noteworthy innovations include the introduction of anchor boxes for bounding box prediction, a feature pyramid network (FPN) for multi-scale feature extraction, and a robust backbone architecture that facilitates accurate object localization and classification.

2. Training Process and Optimization:

Training YOLOv5 involves a series of steps, including data preparation, model configuration, and optimization. The model is trained on labeled datasets, learning to recognize and locate objects within images. The training process emphasizes the importance of well-prepared datasets and hyperparameter tuning to optimize the model's performance. YOLOv5's ability to adapt to diverse datasets and its ease of training contribute to its appeal among researchers and practitioners.

3. Applications and Use Cases:

The versatility of YOLOv5 is evident in its wide range of applications. From the realm of autonomous vehicles, where real-time object detection is crucial for navigation and safety, to surveillance systems that require rapid identification of objects in video feeds, YOLOv5 has proven its efficacy. Its applications extend to healthcare, retail, and industrial settings, showcasing its adaptability across various domains.

4. Performance Metrics and Benchmarking:

The success of YOLOv5 is quantifiable through performance metrics commonly used in object detection evaluations. Precision, recall, F1 score, and mean average precision (mAP) are indicative of its ability to accurately identify and classify objects. YOLOv5 consistently performs well on benchmark datasets, outpacing many of its predecessors and counterparts in terms of both speed and accuracy.

5. Impact on the Computer Vision Landscape:

YOLOv5's impact on the computer vision landscape is profound, as it addresses the need for real-time object detection in resource-constrained environments. Its efficiency has implications not only in research but also in practical applications where rapid decision-making based on visual data is crucial. The open-source nature of YOLOv5 has fostered a vibrant community of developers and researchers contributing to its evolution and adoption across diverse industries.

7.6.3 Image Recognition Using YOLOv5:

In the realm of image recognition, the advent of YOLOv5 (You Only Look Once, Version 5) stands out as a pioneering force, redefining the landscape of object detection and classification. YOLOv5 represents the latest iteration of the YOLO series, renowned for its unparalleled speed and accuracy in processing visual data. Unlike traditional methods that divide an image into multiple regions for analysis, YOLOv5 adopts a holistic approach, enabling it to detect and classify multiple objects within an image in real-time. This introduction delves into the key principles, architecture, and applications of YOLOv5, showcasing its ability to revolutionize image recognition through its efficiency, precision, and adaptability across various domains. From its innovative architecture to its impact on industries such as autonomous vehicles, surveillance, and healthcare, YOLOv5 stands at the forefront of the quest for rapid and accurate visual understanding, offering a glimpse into the future of intelligent image recognition systems.

7.6.4 Preparing Dataset:

The preparation of a dataset for YOLOv5 is a crucial step to ensure the model's effectiveness in image recognition. This process involves meticulous annotation of images, where each object of interest is labeled with bounding boxes to guide the model during training. The annotated dataset serves as the foundation for teaching YOLOv5 to identify and classify objects accurately. Annotators must ensure precise localization of objects, providing the model with a diverse range of examples to learn from. Careful consideration is given to the variety of objects, scales, and orientations present in the dataset to enhance the model's ability to generalize across different scenarios. The quality and representativeness of the dataset directly impacts the model's performance, making thorough annotation and curation essential. As YOLOv5 relies heavily on the richness of the training data, a well-prepared dataset becomes the key to unlocking the model's potential for robust and accurate image recognition in diverse real-world applications [27].

7.6.5 Training process:

The training process in YOLOv5 is a meticulous and crucial phase that transforms the model into a powerful image recognition tool. It begins with the preparation of annotated datasets, where images are paired with bounding boxes indicating the objects of interest. YOLOv5 requires labeled data to learn and generalize patterns effectively. The model configuration involves selecting the appropriate architecture, adjusting hyperparameters, and defining the number of classes to be recognized. During training, the model refines its parameters through iterations, continually fine-tuning its ability to detect and classify objects accurately. YOLOv5 employs optimization techniques, including stochastic gradient descent, to minimize the difference between predicted and ground-truth bounding boxes. The success of the training process heavily depends on the quality and diversity of the training data, the chosen architecture, and the careful tuning of hyperparameters, ultimately producing a robust image recognition model ready to tackle real-world scenarios [27].

Chapter (8)

8. CONCLOUSION AND FUTURE WORK

8.1 CONCLUSION

In conclusion, advanced driver assistance systems (ADAS) have significantly contributed to enhancing car movement safety and improving overall driving experiences. The integration of blind-spot detection systems has effectively reduced the occurrence of accidents caused by lane changes, particularly in scenarios where drivers may have limited visibility. Lane departure systems have proven valuable in preventing unintentional drifts from lanes and reducing the risk of collisions due to driver inattentiveness or fatigue. Traffic sign recognition systems have played a crucial role in improving compliance with traffic regulations. By accurately identifying and displaying relevant traffic signs, these systems help drivers stay informed and make better decisions while on the road.

Overall, the advancements in ADAS technologies have demonstrated their effectiveness in preventing accidents, reducing human error, and enhancing overall road safety. As these systems continue to evolve, they have the potential to significantly reduce the number of collisions and make driving experiences safer and more enjoyable for everyone on the road.

We have successfully designed and built a robotic car. We have implemented a comprehensive control system for the car's movement using a microcontroller (such as an ESP32) in conjunction with a PS4 controller. Additionally, we have integrated two ultrasonic sensors, one on the right and one on the left, at specific angles to detect blind spots for the driver. When an object or another car approaches the blind spot area, an alert is triggered.

8.2 FUTURE WORK

In the future work, we can further enhance the capabilities of the car by implementing reactive measures for significant risks in the blind spot. For instance, if the car is moving in the right or left direction and encounters a high-risk situation indicated by the red color, it can autonomously steer in the opposite direction to avoid a collision.

Future directions in Traffic Sign Interpretation in ADAS offer exciting opportunities for enhancements and integration with advanced algorithms. These include advanced ML and deep learning algorithms, multimodal sensor fusion, augmented reality integration, interpreting dynamic signs, customizable alert systems, cloud-based learning, improved human-machine interaction, and collaborative driving ecosystems.

In the future work, we will be able to showcase the detection of road bumps after developing and successfully testing the system, obtaining satisfactory results with high efficiency in detecting both visible and invisible road bumps using depth cameras to protect drivers .

REFERENCES

[1] Augustine, Stephen Tipa. *Design and Implementation of a Blind Spot Detection and Monitoring System for The Kayoola Buses*. Diss. 2022.

[2] https://www.blindspotmonitor.com/ar/ways-to-fix-a-blind-spot-monitor-in-car/

[3] Espressif. (2023). ESP32 Datasheet. Retrieved from https://www.espressif.com/sites/default/files/documents/esp32_datasheet_en.pdf

[4] Dubizzle. "Lane Departure Warning System." Dubizzle Blog, n.d., [Online]. Available: https://www.dubizzle.com/blog/cars/lane-departure-warning-system/. Accessed 13 Dec 2023.

[5] Azim Eskandarian, *Handbook of Intelligent Vehicles*. First Edition, Springer-Verlag London Ltd. 2012.

[6] Zhang, Jason and Ope Oladipo. "Raspberry Pi Lane Departure Warning System." Cornell University, 21 May 2017, [Online]. Available: https://courses.ece.cornell.edu/ece5990/ECE5725_Spring2017_Projects/jcz28_ooo25/index.html . Accessed 27 Dec 2023.

[7] Marsden, Greg, Mike McDonald, and Mark Brackstone. "Towards an understanding of adaptive cruise control." *Transportation Research Part C: Emerging Technologies* 9.1 (2001).

[8] Piccinini, Giulio Francesco Bianchi, et al. "Driver's behavioral adaptation to Adaptive Cruise Control (ACC): The case of speed and time headway." *Journal of safety research* 49 (2014).

[9] King, Paul John, Michael Julian Richardson, and Daniel Watts. "Adaptive cruise control system." U.S. Patent No. 6,116,369. 12 Sep. 2000.

[10] Bruno Siciliano, et al., *Springer Handbook of Robotics*. Second Edition, Springer Berlin Publishing, Heidelberg, 2016. 1627-1656.

[11] Waveshare. "TF-Luna LiDAR Range Sensor." Waveshare Wiki, n.d., [Online]. Available: https://www.waveshare.com/wiki/TF-Luna_LiDAR_Range_Sensor Accessed 12 Jan 2024.

[12] Yahboom. "Astra Pro." Yahboom Products, n.d., [Online]. Available: https://category.yahboom.net/products/astra_pro Accessed 17 Nov 2023.

[13] Orbbec. "Astra Pro Plus." Orbbec, n.d., [Online]. Available: https://www.orbbec.com/products/archived-products/astra-series/ . Accessed 20 Nov 2023.

[14] Narvaez, Jose Luis Masache. Adaptation of a Deep Learning Algorithm for Traffic Sign Detection. Diss. The University of Western Ontario (Canada), 2019.

[15] j. pryslak, "medium," 23 April 2020. [Online]. Available: https://medium.com/@jpriceless/visuals-depicting-the-relationship-between-ai-and-other-related-terms-8098520be8c4 . Accessed 4 Jan 2024.

[16] M. Atef, "roboflow," Roboflow, 11 January 2023. [Online]. Available: https://universe.roboflow.com/mohamed-atef-dwbkp/traffic-signs-detection-uudww/health . Accessed 20 Des 2023.

[17] Khan, Muneeb A., Heemin Park, and Jinseok Chae. "A Lightweight Convolutional Neural Network (CNN) Architecture for Traffic Sign Recognition in Urban Road Networks." Electronics 12.8 (2023).

[18] P. Azevedo, "What is the best YOLO," 8 Jun 2022. [Online]. Available: https://medium.com/@pedroazevedo6/what-is-the-best-yolo-8526b53414af. Accessed 20 Jan 2024.

[19] Khan, Muneeb A., Heemin Park, and Jinseok Chae. "A Lightweight Convolutional Neural Network (CNN) Architecture for Traffic Sign Recognition in Urban Road Networks." Electronics 12.8 (2023): 1802.

[20] Kukkala, Vipin Kumar, and Sudeep Pasricha, eds. Machine Learning and Optimization Techniques for Automotive Cyber-Physical Systems. Springer Nature, 2023.

[21] "dagshub," 28 6 2022. [Online]. Available: https://dagshub.com/blog/yolov6/.

[22] Devapriya, W., C. Nelson Kennedy Babu, and T. Srihari. "Advance driver assistance system (ADAS)-speed bump detection." 2015 IEEE international conference on computational intelligence and computing research (ICCIC). IEEE, 2015.

[23] Eslam Gamal Elsayed, et al., Localized Bump Detection. BS Thesis. Alexandria University in. Egypt., 2019-2020.

[24] Jindřich Macek, Speed Bump Detection in Automotive Scenarios. BS Thesis. Czech Technical University in. Prague. Computing and information center, 2023.

[25] Nvidia. "Jetson." Nvidia Developer, n.d., https://developer.nvidia.com/jetson. Accessed 11 Nov 2023.

[26] V7 Labs. "Image Recognition Guide." V7 Labs Blog, n.d., [Online]. Available: https://www.v7labs.com/blog/image-recognition-guide. Accessed 2 Jan 2024.

[27] L. Manikandan, et al., “Object Detection Using Yolo V5,’’ Eur. Chem. Bull. 2023, 12 (S3), 6226 – 6233.