Preprints.org

Article

Not peer-reviewed version

Enhancing Autonomous Vehicle Navigation at Complex Junctions Using LIDAR and YOLOv5-Based Detection

Daniyal Khan , Muhammad Talha Zahid , Muddasar Naeem * , Zaib Ullah , Musarat Abbas , Antonio Coronato

Posted Date: 8 May 2025

doi: 10.20944/preprints202505.0619.v1

Keywords: Autonomous Vehicles; Junction Safety; Object Detection; Real-Time Navigation; LIDAR; Ultrasonic Sensors; Sensor Fusion; Vehicle-to-Vehicle Communication; Vehicle-to-Infrastructure Communication; Intel

Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2025

doi:10.20944/preprints202505.0619.v1

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Enhancing Autonomous Vehicle Navigation at Complex Junctions Using LIDAR and YOLOv5-Based Detection

Daniyal Khan [1], Muhammad Talha Zahid [1], Muddasar Naeem [2,*], Zaib Ullah [2], Musarat Abbas [1] and Antonio Coronato [2]

[1] Department of Electronics; Quaid-i-Azam University Islamabad 44000, Pakistan
[2] Research Center on ICT Technologies for Healthcare and Wellbeing, Università Telematica Giustino Fortunato, Benevento 82100 Italy
* Correspondence: m.naeem@unifortunato.eu

Abstract: Vehicle autonomy demand continues to grow because AI developments and sensor technology enable vehicles to navigate schematically while protecting safety conditions. The main obstacle for autonomous vehicles involves handling advanced junctions since their decision-making system needs to remain agile and precise. The study implements a sensing system including LIDAR depth sensors and camera-based traffic signal detections while YOLOv5 is employed as its deep learning identification system. The method employed transfer learning strategies to reduce training duration without compromising detection quality. We use a component integration system to maintain reliable performance across different conditions e.g. low visibility situations. The proposed autonomous vehicle infrastructure has been tested in both simulated and real-world environments. The system demonstrates superior performance in junction and obstacle avoidance operations over traditional autonomous vehicle technologies.

Keywords: autonomous vehicles; junction safety; object detection; real-time navigation; LIDAR; ultrasonic sensors; sensor fusion; vehicle-to-vehicle communication; vehicle-to-infrastructure communication; intelligent transportation systems; deep learning; traffic light detection

1. Introduction

The transportation industry is undergoing a decisive transformation through the emergence of autonomous vehicles (AVs), which aim to enhance automotive safety, operational performance, and environmental sustainability [1]. Recent advances in artificial intelligence (AI), sensor technologies, and real-time data processing have empowered AVs to navigate complex urban environments with minimal human intervention [2].

Despite these technological strides, a large proportion of traffic collisions, particularly in urban areas occur at intersections. Junctions remain inherently challenging due to the convergence of multiple traffic streams that demand rapid and accurate decision making. According to the World Health Organization, over 1.3 million deaths and countless injuries result from road traffic incidents each year, with junction-related accidents forming a substantial portion of this toll [3].

The core challenge in ensuring safe navigation through intersections lies in the vehicle's ability to achieve high environmental awareness and execute immediate, intelligent actions. Although previous research has explored sensor fusion, object detection, and path planning techniques [4], many existing solutions are constrained by limitations such as poor performance in low-visibility conditions, sensor data integration latency, and limited responsiveness in dynamic traffic environments [5]. Vision-only systems, in particular, tend to underperform under adverse lighting or weather conditions, and inter-vehicle communication capabilities are often lacking, which restricts situational forecasting in shared spaces [6].

© 2025 by the author(s). Distributed under a Creative Commons CC BY license.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2025 doi:10.20944/preprints202505.0619.v1

In response to these challenges, this study presents a robust autonomous vehicle framework tailored for junction safety. The system integrates multimodal sensors—including RPLIDAR A1 for spatial data acquisition, an RGB camera for traffic light detection, an IMU and GPS module for localization, and ultrasonic sensors for close-range obstacle avoidance. The YOLOv5 deep learning model is employed for real-time traffic signal recognition, trained on a custom dataset collected under varying lighting conditions to ensure robustness. High-level processing tasks are managed by the Jetson Orin Nano, while low-level motor control is delegated to an Arduino Mega for efficient actuation.

To improve anticipatory decision-making in shared road environments, the system employs LoRa-enabled Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication modules. These enable real-time data exchange, allowing the vehicle to better predict and respond to future road dynamics [7].

The proposed solution is validated through a combination of simulation-based and real-world testing. Results show improved reaction times, higher detection accuracy, and more reliable junction handling under diverse environmental conditions. This work addresses the gaps in existing literature by introducing a scalable architecture that overcomes the limitations of single-sensor dependency and the absence of inter-vehicular communication.

2. Related Work

Researchers have made recent developments in autonomous vehicle technology center on sensor integration along with artificial intelligence perception capabilities and communication systems for intelligent and safe vehicle navigation [8]. The main obstacle in autonomous vehicles exists in achieving precise environmental recognition particularly in traffic regions with multiple complicated elements such as intersections.

The work in [1] stressed that by fusing LIDAR with camera data applications gain better spatial awareness together with diminished false positive occurrences in object detection systems. The research examined highway situations while lacking features that would handle junction-bound traffic attributes or signal detection capabilities which restricted its usage in urban intersection settings.

The study in [4] comprises a detailed review of traffic signal recognition techniques based on deep learning approaches. The research established CNNs as successful in identifying traffic signals yet its examined models required perfect viewing conditions with proper illumination. Our solution trains a YOLOv5 model with customized dataset collected under varied lighting situations to achieve realistic deployment.

YOLOv5 received practical implementation details from [5] through their object detection work. The methodology demonstrated limited capability for executing dynamic decisions with real-time control systems despite showing success in object classification which our AV framework solves by connecting hardware actuators and implementing PID control mechanisms.

The discussions about safety systems through communications have been presented in [6], how Vehicle-to-Everything (V2X) communications can help decrease intersection collisions. The existing system-based solutions require 5G or DSRC infrastructure although it presents service limitations because of incomplete infrastructure delivery. Our system depends on LoRa technology as a V2V and V2I communication method to establish an affordable and extended-range communication system without demanding specific infrastructure.

Furthermore, authors in [9] explored multi-agent reinforcement learning for scheduling in multi-antenna communication systems. Multi-antenna systems are useful to improve the system performance in terms of link reliability, data rates, coverage, and capacity [10]. Their contribution significantly improves spectral efficiency for V2X, but lacks integration with onboard decision-making or low-level motor control, which our work addresses using Jetson Orin Nano and Arduino Mega coordination.

In summary, while previous research has made significant contributions in sensor fusion, deep learning, and vehicular communication, they often address these components in isolation. Our system presents a unified, modular framework that fuses these technologies into a real-time junction safety solution validated in both simulated and real-world scenarios.

3. Technical Background

This section provides a brief overview of the foundational technologies and algorithms employed in the development of the proposed autonomous vehicle system.

3.1. YOLOv5 for Real-Time Object Detection

You Only Look Once version 5 (YOLOv5) is a state-of-the-art object detection framework based on convolutional neural networks (CNNs). It enables real-time identification and classification of objects in a single pass over the image [5]. In this work, YOLOv5 is trained on a custom dataset consisting of labeled traffic light images to detect signal states (red, yellow, green) during vehicle navigation. The model's lightweight architecture and fast inference make it suitable for embedded AI hardware like the Jetson Orin Nano.

3.2. LIDAR for Spatial Mapping

Light Detection and Ranging (LIDAR) is a technology which uses laser beams to collect data from measuring distance-to-target on surrounding landscape to provide 2D profiles. With the proposed system, RPLIDAR A1 is exploited for 2D scanning of the surroundings, generating the space data for obstacle detection and localization. In contrast to vision-based systems, LIDAR provides uniform performance in diverse illumination and is essential for near-range navigation and path planning, As shown in Figure 1 Hector SLAM is utilized on line for mapping and localization whilst path planning is implemented using A* and Hybrid A* [11].

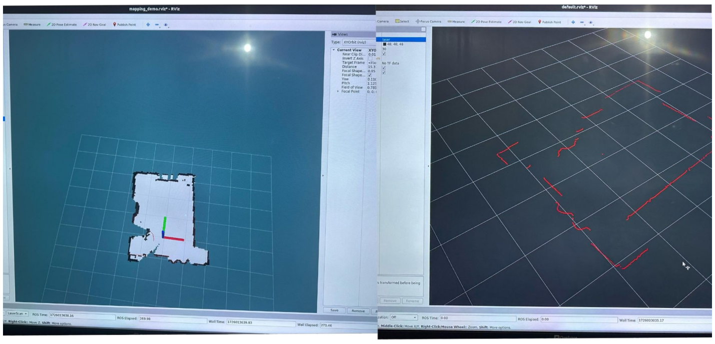


Figure 1. Simultaneous Localization and Mapping.

3.3. PID Controller for Motor Control

Proportional-Integral-Derivative (PID) controllers are widely used in robotics to maintain desired performance by minimizing error over time. In the proposed vehicle, the PID controller adjusts motor speed using feedback from rotary encoders to ensure smooth and accurate motion. The controller calculates output using three terms: proportional (current error), integral (accumulated error), and derivative (rate of error change), allowing precise movement in response to sensor inputs.

3.4. LoRa-Based V2V and V2I Communication

LoRa (Long Range) is a low-power wide-area network (LPWAN) protocol ideal for wireless communication in embedded systems. In this system, LoRa modules are utilized for both Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication, facilitating data exchange with nearby AVs or roadside infrastructure. Unlike conventional V2X protocols that require high-bandwidth networks, LoRa offers a low-cost and scalable alternative suitable for real-time decision sharing in junction environments [6].

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2025 doi:10.20944/preprints202505.0619.v1

3.5. *Simultaneous Localization and Mapping (SLAM)*

SLAM is a technique that allows a robot or autonomous vehicle to construct a map of an unknown environment while simultaneously keeping track of its location within that map. The proposed vehicle uses Hector SLAM, a robust algorithm optimized for lightweight platforms, in combination with LIDAR input to generate and update real-time 2D maps of the operating environment. This aids in reliable localization and path planning, particularly in GPS-denied indoor scenarios [12].

3.6. *Neural Networks*

Neural networks are computational models inspired by the structure and function of biological neural systems. They are composed of layers of interconnected nodes (neurons) that process data by assigning weights and biases during training. Neural networks are particularly effective for tasks involving classification, regression, and pattern recognition [13]. In the context of autonomous vehicles, they are employed to interpret complex sensor data, enabling the vehicle to understand and interact with its environment. Deep neural networks (DNNs), which contain multiple hidden layers, are often used in AVs for high-level decision-making and perception [14].

3.7. *Convolutional Neural Networks (CNNs)*

Convolutional Neural Networks (CNNs) are a specialized type of neural network designed to process grid-like data such as images. CNNs utilize convolutional layers to automatically extract spatial features, making them highly effective for object detection and image classification. Each convolutional layer applies a set of learnable filters that scan the image to detect edges, shapes, and patterns. This hierarchical feature extraction capability allows CNNs to recognize complex visual cues such as traffic lights, pedestrians, and road signs. The YOLOv5 model used in this work is based on a CNN architecture optimized for real-time performance [4].

3.8. *Transfer Learning*

Transfer learning is a machine learning technique where a model developed for one task is reused as the starting point for a model on a second, related task. In the context of this research, a pre-trained YOLOv5 model was fine-tuned using a custom dataset of traffic light images. This approach significantly reduces training time while improving model accuracy, especially when the new dataset is limited in size. Transfer learning enables the model to leverage learned features from large-scale datasets, such as COCO or ImageNet, and apply them to more specific, domain-relevant tasks [5].

4. Methodology

This section provides the detailed approach used to design the Autonomous Vehicle (AV) system with junction safety and obstacle detection, the description of the steps of the hardware and software configurations and sensors incorporated, data acquisition and analytical approaches, and the process of the algorithms that were developed to achieve the expected functionalities. In addition, it includes quality assurance and control, implementation of test cases, and verification phases of the system. The proposed system model is shown in Figure 2.

The AV is engineered to navigate, respond, and operate independently in traffic and congested environments. It can detect and avoid obstacles, recognize and comply with traffic lights, ensuring a smooth and uninterrupted flow of traffic.

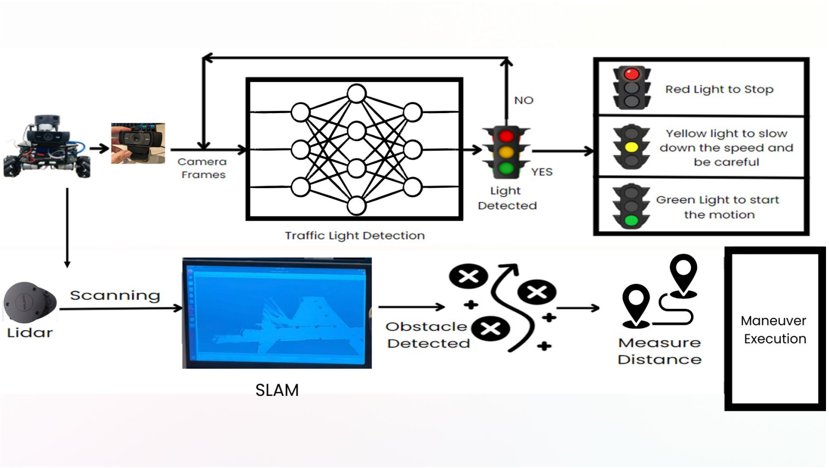


Figure 2. System Architecture.

4.1. Integration of Components

The robot maintains a system of carefully integrated components which function perfectly in dynamic operational environments as demonstrated in Figure 3. The system runs on a 12V 9600 mAh lithium-ion battery that provides steady power to each hardware element. A range of system modules communicate through USB extension boards that ensure stable module connectivity among all system components. The RPLIDAR interfaces with the Jetson Nano through which the Hector SLAM algorithm uses to both scan the environment and produce a 2D representation of surrounding areas. Utilizing OpenCV alongside YOLOv5 assists the system in traffic light detection by processing the video stream from a camera attached to the Jetson Nano. The control system engages with the Mobius motor driver through which it operates both DC and servo motors while receiving instructions from perception modules. The robot becomes equipped to effectively merge mapping with vision and motion control capabilities because of this integration.

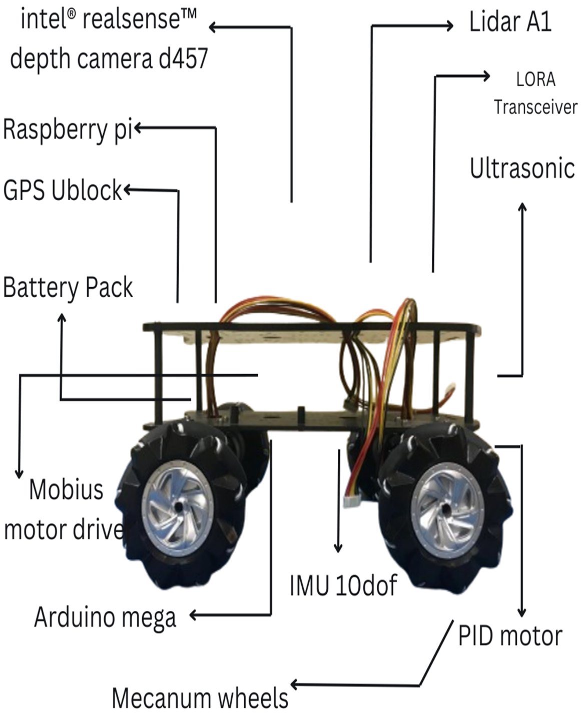


Figure 3. The proposed autonomous vehicle.

4.2. Software Setup

The software setup as presented in Table 1, encompasses various tools, libraries, and algorithms used to control the vehicle and process signals from the sensors. The operating system used is either Raspbian OS / Ubuntu Server OS 20.04 on the Raspberry Pi. Python is the primary programming language for coding control algorithms and integrating components. ROS Noetic is employed for controlling sensors, managing data transactions, and executing control blocks. OpenCV is utilized for image processing and traffic light recognition. YOLOv5 is the algorithm used to predict the position of traffic lights and classify their states. TensorFlow or PyTorch frameworks are used for training and deploying the YOLOv5 model. The Arduino IDE is used for programming the Arduino Mega board to control various components and perform tasks.

Table 1. Software Components of the Autonomous Vehicle System.

Software	Description
Operating System	Raspbian OS / Ubuntu Server OS 20.04 on the Raspberry Pi.
Programming Language	Python, used for coding control algorithms and integrating components.
ROS Noetic	A Robot Operating System (ROS) version utilized for controlling sensors, managing data transactions, and executing control blocks.
OpenCV	Employed for image processing and traffic light recognition.
YOLOv5	An algorithm used to predict the position of traffic lights and classify their states.
TensorFlow or PyTorch	Frameworks for training and deploying the YOLOv5 model.
Arduino IDE	Used for programming the Arduino Mega board to control various components and perform tasks.

4.3. Working Operation

The proposed methodology for operating an autonomous vehicle is demonstrated in Figure 4. The description of each step is given next:

1. **Start by moving the robot forward**
2. **Check for an obstacle using Ultrasonic**
   (a) If no obstacle is detected, continue to move forward.
   (b) If an obstacle is detected, turn left and check for obstacle and, if the path is clear, start moving forward.
   (c) If an obstacle is detected, turn right and check for obstacle and, if the path is clear, start moving forward.
3. **Check for an obstacle using LIDAR**
   (a) If no obstacle is detected, continue to move forward.
   (b) If an obstacle is detected, the robot will plan the path according to the **Slam**.
4. **Check for Traffic Light using Camera**
   (a) If no Traffic light is detected the car moves forward.
   (b) If the Red light is active on the traffic light, **The Vehicle comes to a stop**.
   (c) If the Yellow light is active on the traffic light, **The Vehicle slows down**.
   (d) If the Green light is active on the traffic light, **The Vehicle continues its forward motion**.
5. The process repeats until the robot navigates the path without encountering any obstacles.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2025 doi:10.20944/preprints202505.0619.v1

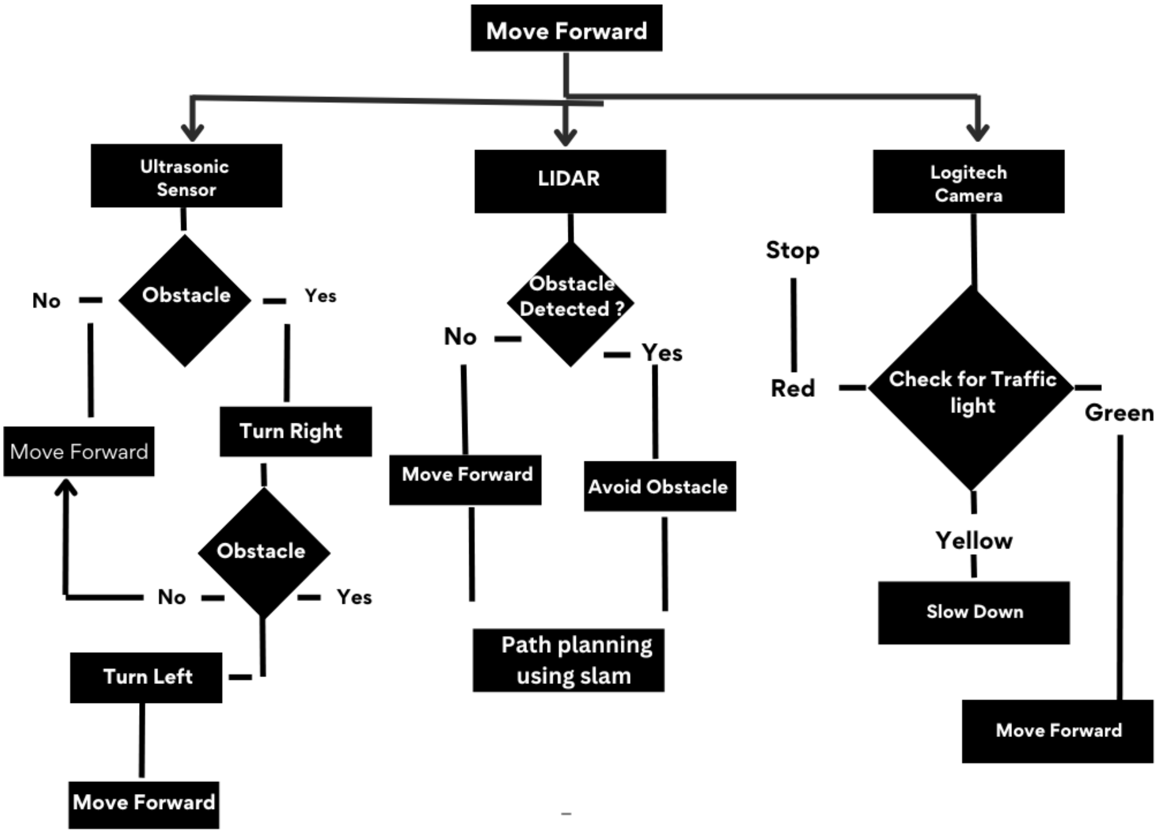


Figure 4. Overview of How The Robot Work.

4.4. Object Detection and Navigation

This sub section presents steps involved in object detection and navigation.

4.4.1. YOLOv5 Training and Optimization

We have developed a custom dataset for the application that fits the Traffic Light detection domain. The dataset comprises three classes: Red, Yellow, Green. These classes are extracted by the model from the camera feed and used to determine the robot's subsequent behavior. The function of each component used is explained below:

Camera and YOLOv5 Model: Used for traffic light detection.
Ultrasonic Sensor: Short range obstacle detection (2-400) cm.
LIDAR: Used for SLAM.
Motor Controller: Responsible for motor movement when desired.
NRF24 Module: Used for vehicle-to-vehicle (V2V) communication.

The system is built by simulating a Traffic junction scene and have multiple traffic lights present in it. The collected videos were extracted framewise with 5 fps by a Python script, where the frames were sampled. The images were manually annotated with the help of the LabelImg tool, which is a graphical image annotation tool that allows accurate marking of objects in the frames. YOLOv5, is then trained with the prepared dataset prepared. YOLOv5 is one of the most advanced and powerful approaches for conducting real-time object detection, making it highly accurate and fast. Based on this, hyperparameter tuning is undertaken in the most optimal manner to ensure improved model performance while achieving the set objectives of classifying classes within our junction scenarios.

4.4.2. Pathfinding and Collision Avoidance Techniques

The path finding and Collision Avoidance Techniques using LIDAR are critically used in the autonomous systems because they involve accurate movements. One of the key methods used here

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2025

doi:10.20944/preprints202505.0619.v1

is at the same time localization and mapping (SLAM) with LIDAR sensors that constantly perform an environment scan and provide data to hector slam [15] to create a map and estimate the robot's location within that map. This data from the LIDAR is employed to detect obstacles and A* and hybrid A* [16] algorithm to decide the best possible path for the operation of the robot in avoiding inflicting havoc on obstacles that are in its path. SLAM makes it easier for the system to simulate map updating in cases where new objects are detected and thus is very useful in interactive environments. This makes it possible to navigate autonomously while avoiding collisions with the barriers and this proves most useful especially where the path is full of barriers of unknown make.

4.4.3. Decision-Making for Movement and Interaction

The YOLOv5 detection influence the robot's decision-making process of movement and interaction. During initial detection the robot scans its surroundings for signs of Traffic Lights. If there is a Traffic Light active ,the robot navigates to the area as indicated. Once the Traffic light color is recogonized, YOLOv5 carries out additional processing to find what color is active and sends the message serially to Arduino Mega's motor shield to control the motors accordingly as demonstrated in Figure 5.

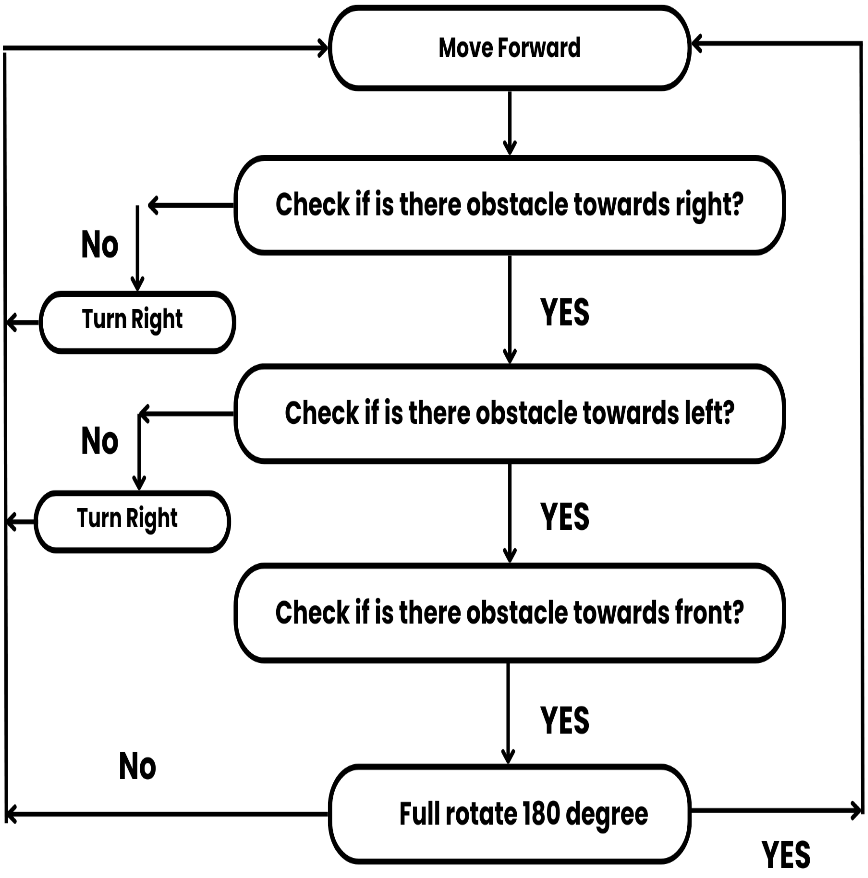

Move Forward
Check if is there obstacle towards right?
No
Turn Right
YES
Check if is there obstacle towards left?
No
Turn Right
YES
Check if is there obstacle towards front?
YES
No
Full rotate 180 degree
YES

Figure 5. Decision flow chart.

5. System Architecture

The vehicle's system architecture consists of three main components. The first component is Perception System that gathers information on the surrounding environment with the help of sensors. While the second component is Decision-Making System which analyzes the sensor information to perform navigational function decisions. The last part of the proposed architecture is Actuation System which is used to ensure implementation of the decisions by steering and regulating the vehicle's movements. We present more detailed functionality of the proposed system in subsequent subsections.

5.1. Perception System Data Collection and Annotation

Perception systems are responsible for taking in account information from the environment using the sensors. The function of each component is give below:
Ultrasonic Sensors: Sense for an obstacle that are in its range to avoid by the vehicle.
LIDAR: Generates very detailed 2D models of the surrounding environment.
Camera with YOLOv5: Recognizes traffic lights, road signs, and other vehicles.

To train the YOLOv5 model, a custom dataset was created by capturing videos of junction scenarios as shown in Figure 6. The following steps were followed:
Scenario Creation: Real-life junction scenarios were staged, and videos were recorded.
Video to Frames Conversion: The recorded videos were converted into frames using a Python script, sampled at 5 frames per second (fps).
Annotation: The frames were annotated using LabelImg software to label traffic lights, road signs, and other relevant objects [17].

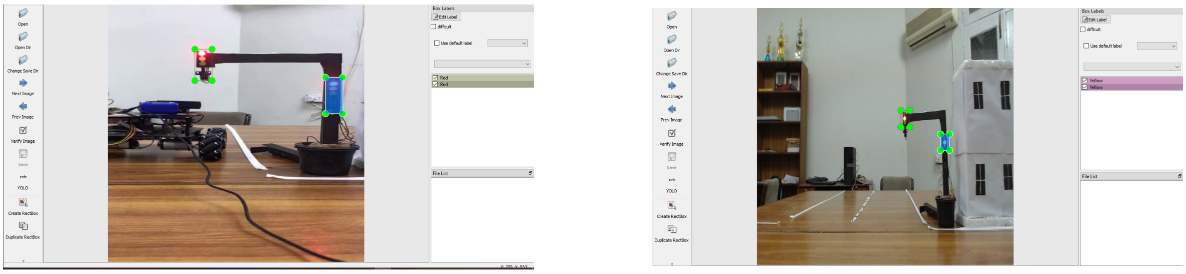


Figure 6. Custom Dataset Creation.

5.2. Communication and Coordination

Effective communication between vehicles is crucial for ensuring safety and coordination. Inter-vehicle communications are essential for achieving safety goals and overall coordination of transport activities. V2X communication was implemented by setting up designated devices for vehicle-to-vehicle (V2V) communication and using the NRF-24 LORA module for vehicle-to-infrastructure (V2I) communication. This setup enabled the sharing of information such as vehicle ID, speed, position, and road conditions, as noted by [18]. Additionally, GPS and IMU data, which provide navigational solutions for cars, were exchanged with neighboring vehicles to help prevent potential accidents at intersections.

5.3. Dataset Collection and Annotation

For the training of YOLOv5 network, a custom dataset was prepared with emphasis on identifying traffic light as a whole and after that to check for the active color. There are total 1060 images in which training folder contains 700 images while the images in validation folder are 360 in total. We created the images shown in Figure 7 and they are closest to the environments from the real-world.

LabelImg application was used for image annotation, there was strict compliance with the required guidelines for bounding the objects of interest. The annotations included three classes:
Red (class 0): Represented by traffic light only when the Red color is active.
Yellow (class 1): Represented by traffic light only when the yellow color is active.
Green (class 2): Represented by traffic light only when the Green color is active.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2025 doi:10.20944/preprints202505.0619.v1

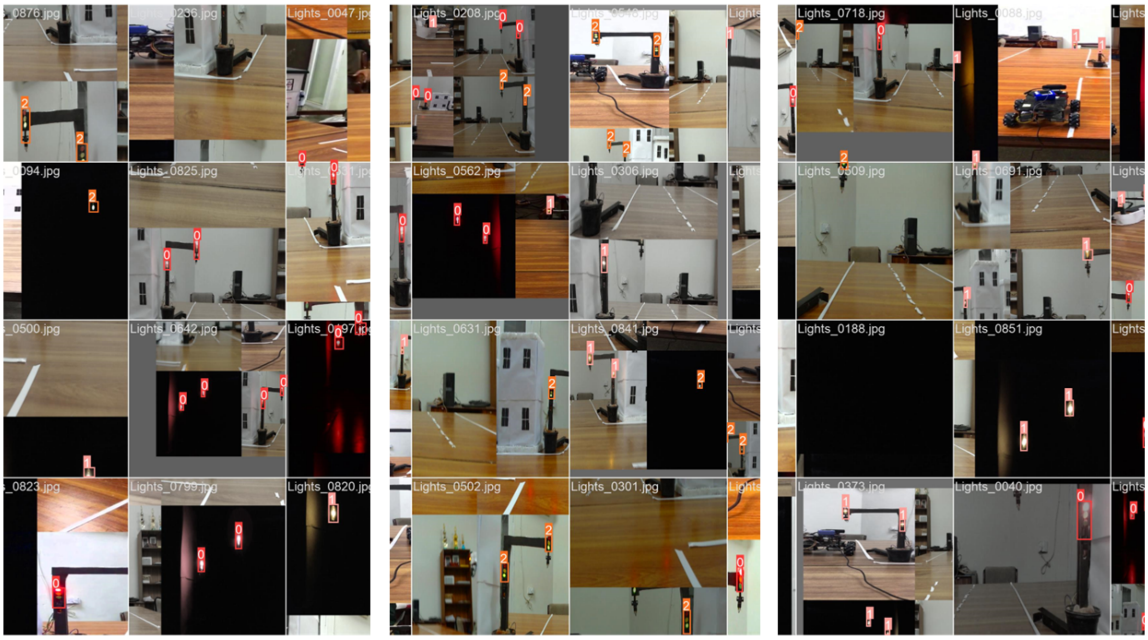


Figure 7. Semantic Segmentation.

5.4. Training Process and Parameters

The YOLOv5 model was trained using custom dataset. The description of steps and parametere are given below:

Data Preprocessing: The images were preprocessed to meet the input requirements of the YOLOv5 architecture. The images were resized to 640 x 640.

Model Configuration: The classes in the configuration file of YOLOv5 were modified to include our classes: Red, Yellow and Green. The path that points at the training folder and validation training folder were also replaced by path that points at our training and validation training folder.

Learning Rate: The parameters were manually tuned in a way that helps to enhance the training performance and also to minimize over-fitted problem.

Epochs: Optimal outcomes were achieved at an overall of ninety seven epochs.

Training Setup: The training process was done.

5.5. Control Logic and Algorithms

Path-finding Algorithm

The AV uses a navigation system in order to navigate from one point to another by using the A* and hybrid A* path planning algorithm. This algorithm utilizes the present position of the vehicle, available obstacles and geometry characteristics of the environment with customized script to give a correct path.

SLAM (Simultaneous Localization and Mapping)

Hector slam with modified script for this robot is used in to help the vehicle construct the map of the environment and at the same time understand where it is in the created map. The LIDAR, camera, and IMUs provide input where the vehicle reconstructs the environment and other moving and non-moving objects in real-time. This enables the vehicle to know its position in the environment hence get around in complex and unfamiliar spaces.

Obstacle Detection and Avoidance

The AV has LIDAR to look for objects in front of the vehicle, ultrasonic to look on both the left and right side of the vehicle and a camera to process images. The vehicle has the ability to compute

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2025 doi:10.20944/preprints202505.0619.v1

the proximity and based on this caused by the existence of an obstacle determines whether to slow down, stop or reverse.

Motion Contro

This is the actual motion control of the necessary paths given by different algorithms in path planning and obstacle avoidance. This includes the speed and direction control of the vehicle's wheels which is regulated through PID controllers. The control system operates the power train, brakes and wheel direction and the established initial path along with the sensors as demonstrated in Figure 8.

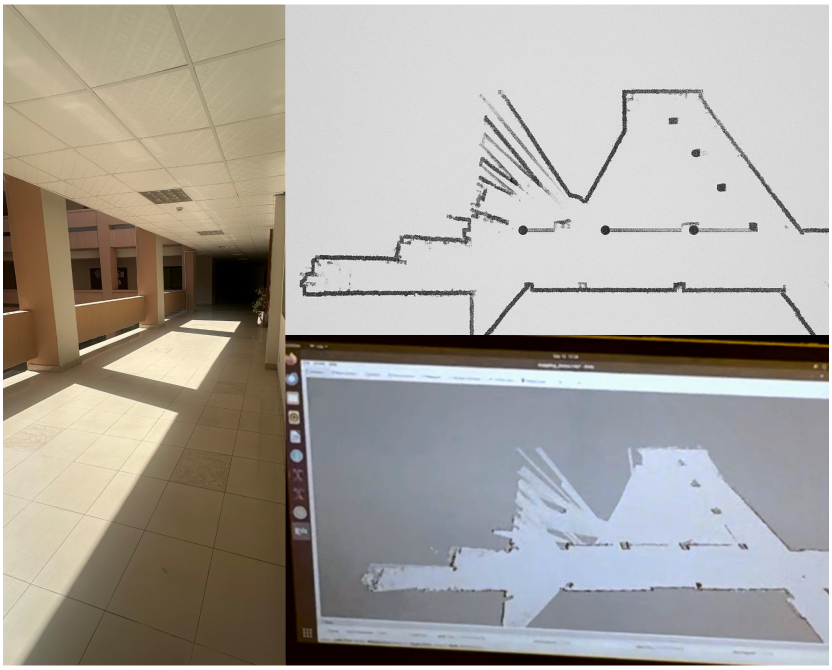


Figure 8. Simultaneous Localization and Mapping.

6. Testing and Results

This section has two parts. The first part presents the testing procedures and performance metrics. While the second part is about results and discussion.

6.1. Testing Methodologies

This section discusses the approaches that are applied to evaluate the system. In particular, the following approaches, tools, and procedures were conducted to assess the system's performance and functionality:
Test Scenarios: The ideas enumerated are explained through different scenarios of a comprehensive strategic planning and realization of changes considering several aspects.
Testing Tools: Techniques and equipment used for testing were simulated on the computerized payroll software as well as other measurement instruments.
Evaluation Metrics: Metrics are used to assess performance metrics including accuracy, response time and reliability.

Testing Environments

For assessing the proficiency and stability of the Autonomous Robot,the testing included a well created physical environment, which is set up to replicates the junctions as they are in the real world. We used thermopole to create 4 buildings, there are traffic lights and lines are demarcated on the road for a better segmentation. The car is introduced to an obstacle first and after that the traffic light is introduced. The traffic light is tested for all the possible outcomes.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2025 doi:10.20944/preprints202505.0619.v1

Testing Procedures

The following procedures are adopted during the test phase:

Preparation: Necessary setup before testing included calibration of sensors and configuration of software.

Execution: Testing involved executing specific steps and commands to evaluate system performance.

Data Collection: Data was collected through logs and real-time monitoring.

Performance Metrics

Performance metrics are important for testing and validation of any system. Next we present the details of each performance metric that we used for the evaluation of our proposed system.

Precision: It answers the question "Out of all the instances that the model predicted as positive, how many were actually positive?". High precision indicates that the model has a low false positive rate.

Recall: It answers the question "Out of all the actual positive instances, how many did the model correctly identify?". High recall indicates that the model has a low false negative rate.

Training Box Loss: Training Box Loss refers to the error between the predicted bounding boxes and the actual ground-truth bounding boxes during the training process.

Training Object Loss: Refers to the error or discrepancy between the predicted object scores (probabilities that a specific region contains an object) and the actual ground-truth labels (indicating whether the region contains an object or is just background) during the training process.

Training Classification Loss: Refers to the error or discrepancy between the predicted class labels for objects within the bounding boxes and the actual ground-truth class labels during the training process.

Validation Box Loss: Validation Box Loss refers to the error between the predicted bounding boxes and the actual ground-truth bounding boxes during the model testing on validation data.

Validation Object Loss: Refers to the error or discrepancy between the predicted object scores (probabilities that a specific region contains an object) and the actual ground-truth labels (indicating whether the region contains an object or is just background) during the model testing on validation data.

Validation Classification Loss: Refers to the error or discrepancy between the predicted class labels for objects within the bounding boxes and the actual ground-truth class labels during the model testing on validation data.

6.2. Results

The results obtained from the testing phase are presented here. Key findings are illustrated with charts, graphs, and tables.

Standard metrics examined performance of the object detection model across 97 training epochs by evaluating Precision, Recall as well as mAP@0.5 and mAP@0.5:0.9. As shown in Figure 9, All performance metrics show a steady increase right after the beginning of training before remaining stable at elevated levels. The metric precision achieves stability at 0.85 and maintains this value following 40 training steps while recall achieves the same benchmark after a similar period. The detection accuracy of the model is reliable because the mean Average Precision (mAP) exceeds 0.85 at IoU threshold 0.5. The performance metric mAP@0.5:0.9 also exhibits a progressive elevation that reaches 0.65 during the period when all other metrics achieve peak stability. This outcome signifies the model's ability to maintain reliable detection accuracy with different intersection-over-union thresholds.

The corresponding loss functions further validate the training effectiveness. As depicted in Figure 10, A considerable decrease appears in every component of loss—the classification loss together with the objectness loss and bounding box regression loss. The validation classification loss together with objectness loss show initial variations up until around 30 epochs before achieving stability at the same time as evaluation metric plateau formation. Between the training and validation losses there exists a steady convergence while the training losses demonstrate minimal overfitting because the

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 May 2025 doi:10.20944/preprints202505.0619.v1

validation results align closely with their training counterparts. The model demonstrates effective generalization capabilities for new data points because of the implemented training method.

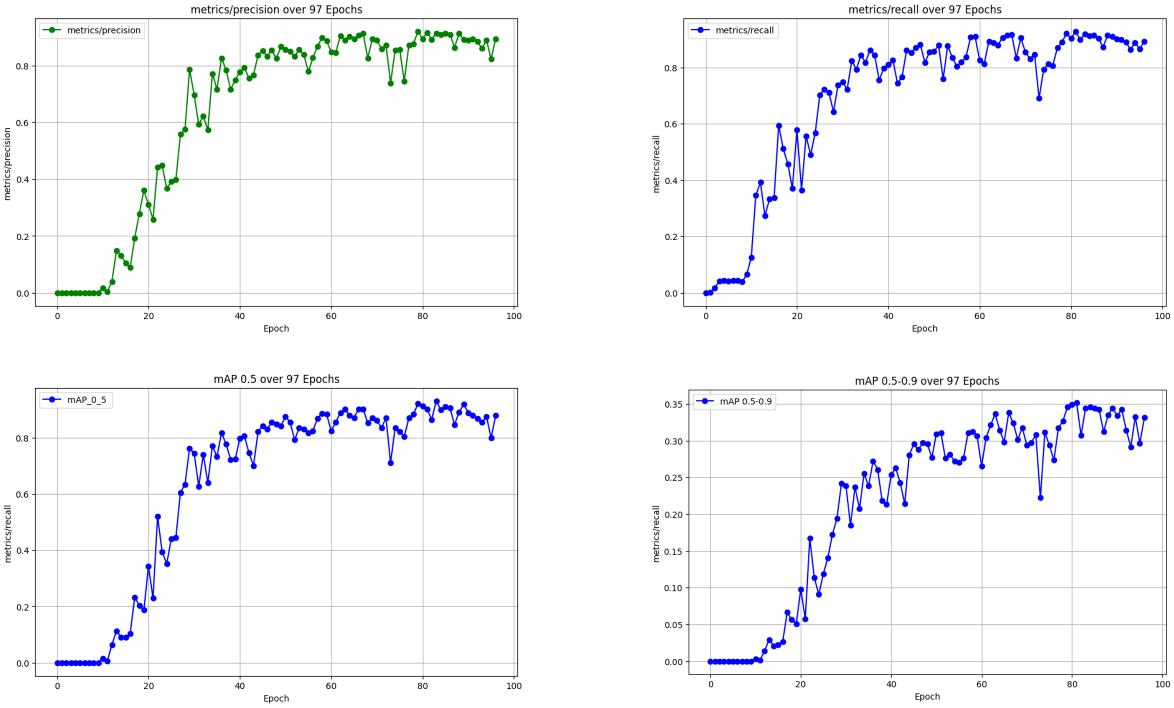


Figure 9. Precision, Recall, mAP 0.5 and mAP 0.5-0.9 over Epochs.

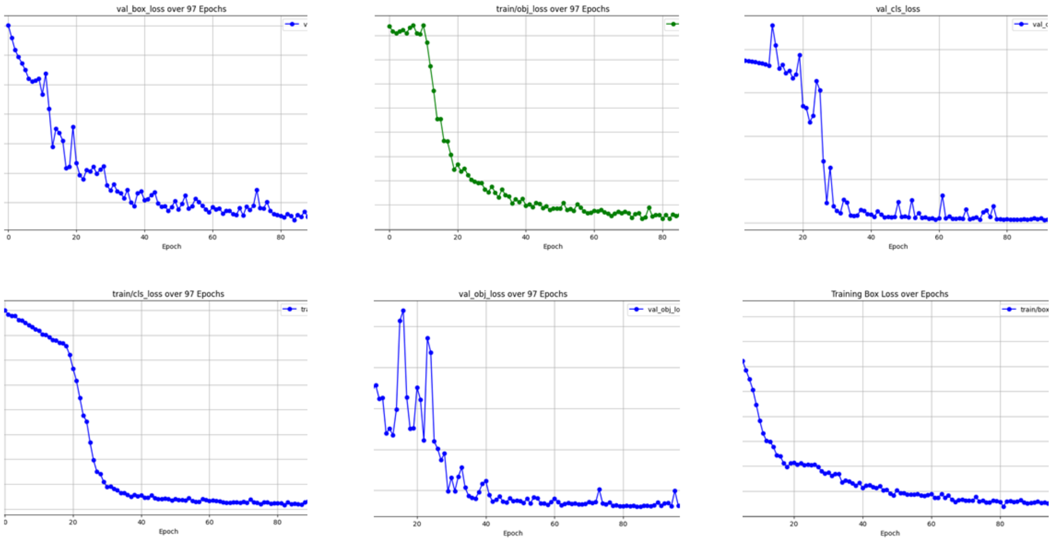


Figure 10. Trends In Losses Over Epochs.

The results shows that the system performed well in most scenarios. However, some challenges such as calibration difficulties were encountered and we addressed them through adjustments. A practical demonstration has been conducted as shown in Figures 11 and 12 and proposed system showed promising results.

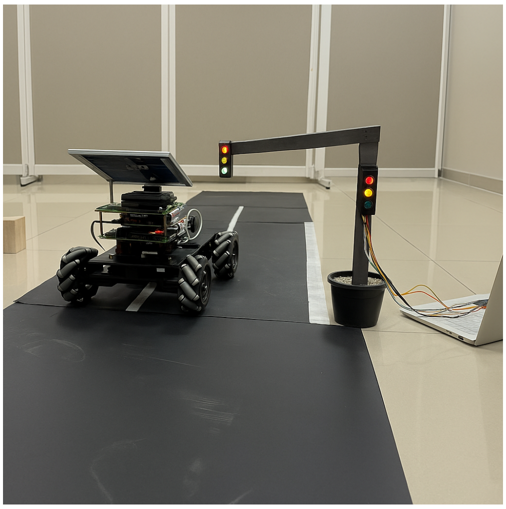
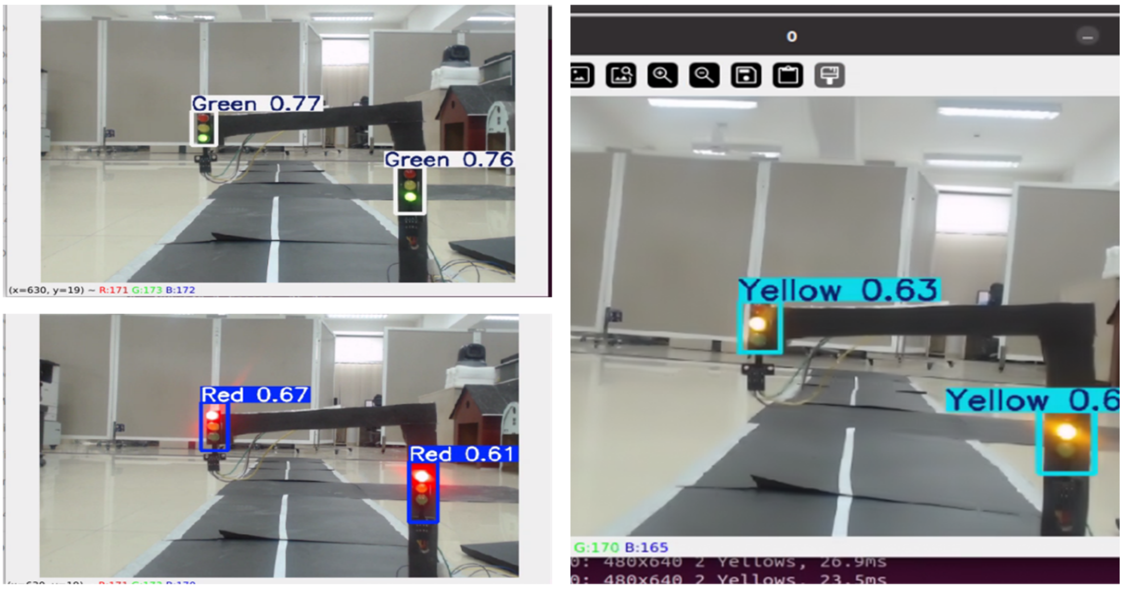


Figure 11. The Final Vehicle.



```
0: 480x640 2 Yellows, 26.9ms
0: 480x640 2 Yellows, 23.5ms
```

Figure 12. Testing YOLOv5.

7. Conclusions

The development and deployment of the autonomous car designed to ensure junction safety highlighted the potential of combining advanced object detection algorithms with robotic platforms to improve traffic flow and reduce accidents. Key findings from the project are outlined below:

Effective Object Detection: The custom-trained YOLOv5 model successfully detected traffic-related objects, achieving high precision and recall. The model's ability to classify traffic lights into Red, Yellow, and Green categories enhanced the robot's capacity to interpret scenarios and contribute to smoother vehicular movement.

Real-Time Performance: The robot demonstrated effective real-time autonomous operation. The integration of SLAM, ultrasonic sensors, and PID control enabled accurate object identification and obstacle avoidance. With YOLOv5's fast and reliable object detection, the robot responded quickly and accurately to its surroundings.

Challenges and Limitations: Some challenges included limited dataset size, hardware constraints, real-time processing limitations, and environmental variations. Addressing these factors will be vital for improving robustness and reliability in real-world applications.

Author Contributions: Conceptualization, M.N. and A.C.; Data curation, M.A.; Formal analysis, Z.U.; Investigation, Z.U and M.A.; Methodology, D.K. and M.T.Z.; Project administration, M.A. and A.C.; Resources, M.N. and Z.U; Software, D.K. and M.T.Z.; Supervision, M.A. and A.C.; Validation, M.N and Z.U.; Writing—original draft, D.K. and M.T.Z.; Writing—review and editing, M.N and Z.U. All authors have read and agreed to the published version of the manuscript.

Funding: The project partially supported by SIAMO (Servizi Innovativi per l'Assistenza Medica a bOrdo) Funded by Ministero delle Imprese e del Made in Italy (MIMIT) wihtn the program "Scoperta Imprenditoriale". Prog. n. F/360124/01/X75

References

1. Bai, W.; Zheng, Y.; Chen, Y.; Yang, S. Towards Autonomous Driving: Sensor Fusion for Self-Driving Cars. *Sensors* **2020**, *20*, 2775.
2. Jamal, M.; Ullah, Z.; Naeem, M.; Abbas, M.; Coronato, A. A hybrid multi-agent reinforcement learning approach for spectrum sharing in vehicular networks. *Future Internet* **2024**, *16*, 152.
3. Organization, W.H. Global status report on road safety. https://www.who.int/publications/i/item/9789241565684, 2022. Accessed: 2025-04-12.
4. Zhou, H.; Wang, J.; Wang, J. A Survey on Deep Learning-Based Traffic Signal Recognition: Challenges and Future Directions. *IEEE Transactions on Intelligent Transportation Systems* **2020**, *22*, 1642–1652.
5. Gonzalez, J.; Smith, J. Object Detection with YOLOv5: A Practical Approach. *Journal of Machine Learning Research* **2021**, *22*, 1–13.
6. Zyss, H. V2X Communication Toward a Zero-Accident Future. *EE Times Europe* **2023**.
7. European Telecommunications Standards Institute (ETSI). ETSI TR 103 439: V2X Standards for Road Safety and Smart Transport. Technical report, ETSI, 2022.
8. Fiorino, M.; Naeem, M.; Ciampi, M.; Coronato, A. Defining a metric-driven approach for learning hazardous situations. *Technologies* **2024**, *12*, 103.
9. Naeem, M.; Coronato, A.; Ullah, Z.; Bashir, S.; Paragliola, G. Optimal User Scheduling in Multi-Antenna Systems Using Multi-Agent Reinforcement Learning. *Sensors* **2022**, *22*, 8278.
10. Naeem, M.; Bashir, S.; Ullah, Z.; Syed, A.A. A near optimal scheduling algorithm for efficient radio resource management in multi-user MIMO systems. *Wireless Personal Communications* **2019**, *106*, 1411–1427.
11. Smith, J. LIDAR-Based Mapping and Localization Techniques. *International Journal of Robotics Research* **2020**, *39*, 450–468.
12. Garcia, L. Hector SLAM: Efficient Mapping and Localization. In Proceedings of the Proceedings of the International Conference on Robotics, 2021, pp. 77–90.
13. Shehzad, F.; Khan, M.A.; Yar, M.A.E.; Sharif, M.; Alhaisoni, M.; Tariq, U.; Majumdar, A.; Thinnukool, O. Two-Stream Deep Learning Architecture-Based Human Action Recognition. *Computers, Materials & Continua* **2023**, *74*.
14. Ullah, N.; Javed, A.; Ghazanfar, M.A.; Alsufyani, A.; Bourouis, S. A novel DeepMaskNet model for face mask detection and masked facial recognition. *Journal of King Saud University-Computer and Information Sciences* **2022**, *34*, 9905–9914.
15. Kohlbrecher, S.; Von Stryk, O.; Meyer, J.; Klingauf, U. A flexible and scalable SLAM system with full 3D motion estimation. *Proc. IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* **2011**.
16. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* **1968**, *4*, 100–107.
17. Tzutalin. LabelImg, 2015. Accessed: 2024-07-22.
18. Gao, W.; et al. Radar Technology and Its Applications in Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems* **2020**, *21*, 233–245.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.