Contents

1. Introduction 1
1.1. Importance of delivery Robot 6
1.2. Challenges of using delivery robots 7
2. Design and Implementation of The Delivery Robot 9
2.1. Hardware Components 9
2.2. Software Components 11
2.3. Challenges in Design 12
Solutions to Challenges 13
2.4. Sample Code Snippets 13
2.5. Community and Environmental Impact 17
Sustainability 17
2.6. Design 19
3. Computer Vision 20
3.1. Key Components of Computer Vision 20
3.2. Applications of Computer Vision 21
3.3. How Computer Vision is used in delivery robots 22
3.4. Benefits of Computer Vision in Delivery Robots 24
3.5. YOLO (Revolutionizing Delivery Robots with Real-Time Object Detection) 25
3.5.1. Understanding YOLO 26
3.5.1 Benefits of YOLO in Delivery Robots 27
3.5.2. Future Possibilities and Enhancements 28
3.5.3. Challenges of YOLO in Delivery Robots 28
3.6. The code 33
4. Navigation and Mapping 37
4.1. Mapping and Localization 38
4.1.1. Sensors 40

4.1.2. Algorithms 41
4.2. The Code 42
5. Communication and Interaction.....51
5.1. Robodovo Application 51
5.2. User Benefits 52
5.3. Code Libraries 54
5.4. How to use Application.....61
6. Case Studies and Applications.....67
6.1. Examples of Companies.....67
6.2. Use Cases of Delivery Robot 69
7. Conclusion 72
8. References.....74

1. Introduction

Artificial Intelligence

It may vary from one person to another about the kind of artificial intelligence they see in their everyday life. The fact that we see different sides of AI depending on the way we use it is fascinating. We may sense AI in face recognition, fingerprint recognition, word prediction, smart assistance, and sentiment analysis in social media if we use smartphones and personal computers in our daily life. Some others will see it in navigation and smart parking systems if they use vehicles daily. Even others will see it as smart lighting systems, self-driving cars, smart HVAC, and smart door-lock if they live in a smart city.

It differs from your point of view on how AI is associated with our everyday lives, depending on our needs, interests, activities, and work environment. But that doesn't change the fact that AI has become one of the greatest bases we need to achieve a goal no matter the level of intelligence it requires. The Applications of Artificial Intelligence are way beyond imagination. Everything you can think of could be guaranteed one way or another.

But even though they vary, and we can see them everywhere, there still are people that think AI is only about robots. And we can see why they think that. There've been plenty of projects that were applied in robotics that attract users and capture their attention.

Robotics

Robotics is a rapidly advancing field that encompasses the design, development, and application of autonomous machines, known as robots, that can perform tasks

with varying levels of autonomy. These machines are equipped with sensors, actuators, and artificial intelligence systems that enable them to interact with their environment and carry out specific functions.

Over the years, robotics has emerged as a multidisciplinary field, drawing from areas such as computer science, mechanical engineering, electrical engineering, and artificial intelligence. The goal of robotics is to create machines that can mimic human actions and intelligence, enabling them to perform complex tasks, automate processes, and enhance productivity across various industries.

Robots come in diverse forms and sizes, ranging from industrial robots used in manufacturing settings to humanoid robots that can interact with humans in social or assistance roles. They can be programmed to follow specific instructions, adapt to changing situations, or learn from experience using machine learning algorithms.

The applications of robotics are extensive and span across numerous sectors. In industrial settings, robots are deployed for tasks such as assembly, welding, material handling, and quality control, improving efficiency, accuracy, and worker safety. In healthcare, robots assist in surgeries, rehabilitation, and patient care, augmenting the capabilities of healthcare professionals. In agriculture, robots are used for tasks like planting, harvesting, and crop monitoring, optimizing agricultural processes and addressing labor shortages.

Beyond industrial and professional domains, robotics has found its way into our daily lives. Robotic vacuum cleaners, personal assistant robots, and autonomous drones are examples of robots that have become increasingly prevalent in households. They provide convenience, entertainment, and new possibilities for how we interact with technology.

The continuous advancements in robotics technology have paved the way for groundbreaking innovations, such as autonomous vehicles, swarm robotics, and human-robot collaboration. These developments hold immense potential for transforming industries, improving efficiency, and enhancing human lives.

Here are some examples on the applications of artificial intelligence in robotics:

• Pepper Humanoid Robots.

• Penny Restaurant Robot.

• Nimbo Security Robot.

• Shadow Dexterous Hand.

• Moley Robotics Kitchen System.

• Flippy Robotic Kitchen Assistant.

• Nomagic Pick-And-Place Robot

And more examples that show the power of AI all over the world and make it a power that can't be denied. And even more, a power that must be used right to ensure the safety and protection of human beings along the task it's accomplishing.

In this paper, we'll discuss another application in the robotics sector of AI, an application that shows the ability of AI and the wide area it covers. the application that sums up more than one function and more than one characteristic of AI. We here discuss the application of an AI Delivery Robot.

The reason why we chose this application specifically is the capabilities and features it provides to the users. We'll discuss more details about the project and dive deep into the use cases and the structure of the project with more practical information.

Delivery Robot

Delivery robots are autonomous machines designed to transport items from one location to another, providing a convenient and efficient solution for various delivery needs. These robots are equipped with advanced technologies such as sensors, cameras, and navigation systems that allow them to navigate through different environments, avoid obstacles, and safely deliver goods.

The idea of using robots to deliver goods and services is not new. In fact, the first delivery robot was patented in 1928. However, it wasn't until the 1990s that delivery robots began to be developed in earnest. This was due in part to the development of new technologies, such as LiDAR and GPS, which made it possible for robots to navigate their surroundings safely and efficiently.

Delivery robots have gained significant attention and adoption in recent years, revolutionizing the way goods are transported and delivered in various industries, including food and beverage, e-commerce, healthcare, and logistics. With their ability to autonomously navigate both outdoor and indoor spaces, delivery robots offer a range of benefits such as increased efficiency, cost savings, and improved customer experience.

These robots come in different forms and sizes, ranging from small, wheeled devices to larger, humanoid-like machines. They can be programmed to follow predefined routes, respond to specific commands, or operate in dynamic environments with the help of artificial intelligence and machine learning algorithms.

There are a variety of different types of delivery robots, each with its own strengths and weaknesses. Some of the most common types of delivery robots include:

- Self-driving cars: Self-driving cars are the most advanced type of delivery robot. They are able to navigate their surroundings without human input, making them ideal for delivering goods and services in a variety of environments.
- Quadcopters: Quadcopters are small, unmanned aerial vehicles (UAVs) that can be used to deliver small packages over short distances. They are ideal for delivering packages to remote locations or to people who are unable to leave their homes.
- UGVs: UGVs, or unmanned ground vehicles, are robots that can travel on the ground. They are ideal for delivering goods and services in areas where self-driving cars or quadcopters cannot operate, such as inside buildings or on uneven terrain.

In the context of food and beverage delivery, delivery robots have become a game-changer. They have the potential to streamline the delivery process, reduce human labor, and enhance operational efficiency for restaurants, cafes, and other food establishments. These robots can navigate crowded streets, access buildings, and deliver meals or beverages directly to customers' doorsteps or designated pickup locations.

Furthermore, delivery robots offer benefits beyond convenience and efficiency. They contribute to reducing carbon emissions by utilizing electric power and optimizing delivery routes, making them an environmentally friendly alternative to traditional delivery methods. Additionally, in a post-pandemic world, delivery robots have played a crucial role in contactless delivery, minimizing the risk of transmitting infectious diseases.

As technology continues to advance, delivery robots are expected to evolve further, incorporating more sophisticated features such as advanced artificial intelligence,

improved battery life, and enhanced interaction capabilities. These advancements will continue to shape the landscape of delivery services, making them more efficient, sustainable, and adaptable to the changing needs of businesses and consumers.

In this report, we will explore the importance and benefits of delivery robots in the food and beverage delivery industry. We will delve into the various use cases, discuss their impact on efficiency and cost savings, examine their safety and hygiene measures, and highlight the ways in which their implementation showcases a business's commitment to innovation and market leadership.

1.1. Importance of delivery Robot

delivery robots offer increased efficiency, speed, and reliability, resulting in improved overall delivery operations.

1. **Time saving:** It can tackle the problem of time and traffic easily by maintaining a consistent speed, avoiding delays caused by factors like traffic congestion or human limitations, navigating through maps and finding shorter and optimized routes, and switching through maps avoiding crowds.
2. **Availability:** Delivery robots can operate around the clock, allowing businesses to offer extended delivery hours or even 24/7 delivery service. As robots do not require rest or breaks, enabling continuous delivery operations. This increases the availability of food and beverage delivery, especially during peak times or late-night hours.
3. **Scalability:** Delivery robots can handle multiple orders simultaneously, thanks to their ability to carry multiple containers or compartments. This allows for batch deliveries, where the robot can deliver different orders in

the same trip, optimizing the delivery process and increasing overall efficiency.

4. **Fault tolerance:** Delivery robots minimize the risk of human errors that can occur during the delivery process, such as incorrect addresses, order mix-ups, or mishandling of food and beverages. Robots follow precise instructions and predefined routes, reducing the likelihood of mistakes and ensuring accurate deliveries.

5. **Cost-effectiveness:** Delivery robots can help reduce or eliminate the need for human delivery drivers, resulting in savings on labor costs. While robots may require initial investment and maintenance, they do not require salaries, benefits, or overtime pay. Over time, the cost of maintaining and operating robots can be lower than employing a team of human drivers. They don't require regular oil changes, tire replacements, or engine repairs. Moreover, they are often electrically powered, reducing fuel costs compared to gasoline or diesel vehicles.

1.2. Challenges of using delivery robots

There are several numbers of challenges that must be addressed before delivery robots can be widely adopted. Some of the most common challenges include:

Safety

Delivery robots must be designed to be safe in a variety of environments, including crowded areas and uneven terrain. This is a complex challenge, and it is one that is still being addressed by researchers and developers.

Regulation

Delivery robots must comply with a variety of regulations, including those governing traffic and safety. This can be a complex and time-consuming process, and it can vary from country to country.

Public acceptance

Delivery robots must be accepted by the public in order to be successful. This can be a challenge, as some people may be concerned about the safety of delivery robots or about the potential for job losses.

2. Design and Implementation of The Delivery Robot

This chapter discusses the various hardware and software components used in the design and construction of an autonomous delivery robot. The robot, primarily designed for intra-university delivery of items such as paper, is built with a focus on sustainability, utilizing laser-cut MDF wood for its construction. The chapter will also delve into the challenges faced during design and how these challenges were overcome.

Robot Specifications

- Size: 50 cm x 40 cm x 30 cm
- Weight: 15 kg
- Speed: 5 to 20 km/h
- Power: Battery-powered
- Sensors: LiDAR, ultrasonic sensors, encoders, RFID reader
- Algorithms: SLAM, path planning

2.1. Hardware Components

- **MDF Wood Chassis**

The robot's chassis is constructed using 4mm MDF (Medium Density Fiberboard) wood. The choice of MDF is motivated by its cost-effectiveness, ease of working, and sustainability. The design involves cutting slots near the four corners to allow the frame to bend, making it more adaptable to different configurations.

- **Arduino Mega**

The Arduino Mega is used as the microcontroller for controlling various components of the robot. It features 54 digital input/output pins, 16 analog inputs, and 4 hardware serial ports which are used to interface with sensors, actuators, and other peripherals.

- **RPLIDAR**

  RPLIDAR is used for mapping and localization. This LIDAR sensor rotates 360 degrees and collects a set of distance samples around the robot, which helps in creating maps and identifying obstacles.

- **Camera**

  A webcam is employed for computer vision tasks. It captures images that are processed by algorithms for object detection, recognition, and navigation.

- **Lenovo Think Center Mini PC**

  A Mini PC with a high-performance processor, 8GB of RAM, and an SSD drive is used as the main processing unit. It handles complex computations and algorithms for navigation and image processing.

- **Power Supply**

  The robot is powered by a rechargeable 12V battery. This battery is chosen to ensure that all components receive adequate power for seamless operation.

2.2. Software Components

- **ROS (Robot Operating System)**

ROS is a flexible framework for writing robot software. It provides services designed for hardware abstraction, low-level device control, implementation of commonly used functionality, message-passing between processes, and package management.

- **YOLO (You Only Look Once)**

YOLO is a real-time object detection system. In this project, it is used for recognizing objects in the robot's path through the webcam.

[Page 3]

- **MovaPKG**

MovaPKG is a package that sets the goal with respect to a map for the vehicle to determine the best parking spot based on certain features. It interfaces with the LIDAR and navigation algorithms to make decisions.

- **SLAM (Simultaneous Localization and Mapping) Monte Carlo Localization**

This is an algorithm that is used in the field of robotics to map an unknown environment and keep track of their location within it.

- **TEB Planner**

Timed Elastic Band (TEB) Planner is used for local planning. It respects the kinematics of the robot and dynamically adjusts the planned trajectory.

- **Dynamic Window Approach**

This approach is used for collision avoidance and trajectory planning in the robot's navigation.

- **Ackermann Kinematics**

This control strategy is used to determine the speed and steering angle of the robot during navigation, ensuring that it follows the desired path.

2.3. Challenges in Design

The design of an autonomous delivery robot entails multiple challenges.

- Selection of Materials

Ensuring that the materials used for the construction of the robot are environmentally sustainable, thermally resistant, and sturdy enough to house all the components was challenging.

- Navigation and Localization

Designing algorithms for navigation and localization which are efficient and reliable in different environmental conditions was another significant challenge.

- Processing Power

Selecting a processing unit that could handle the computational needs of the project efficiently, while staying within budget constraints, posed a challenge. The debate was mainly between using a Mini PC or an NVIDIA board.

Solutions to Challenges

- Material Selection

After considering various materials, MDF wood was selected for the chassis due to its sustainability, workability, and cost-effectiveness. Moreover, additional heat sinks were incorporated to counter heating issues.

- Navigation and Localization

Utilizing a combination of RPLIDAR with SLAM Monte Carlo Localization and incorporating TEB Planner and Dynamic Window Approach ensured reliable navigation and localization.

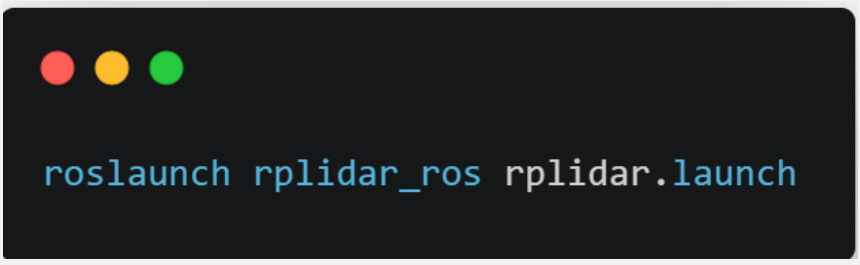
- Processing Power

After evaluating the processing requirements and budget constraints, a Mini PC with a high-performance processor, 8GB RAM, and SSD was chosen. It provided the necessary computational power needed for the project at an optimal cost.

2.4. Sample Code Snippets

Below are some sample code snippets that could be part of the robot's control system.

- Initialize RPLIDAR in ROS



- Run SLAM Algorithm in ROS

```
roslaunch slam_gmapping slam_gmapping
```

- Run YOLO for Object Detection

```
import cv2
from darkflow.net.build import TFNet

options = {"model": "cfg/yolo.cfg", "load": "bin/yolov2.weights", "threshold":
0.1}
tfnet = TFNet(options)

imgcv = cv2.imread("./sample_img/sample_dog.jpg")
result = tfnet.return_predict(imgcv)
print(result)
```

- Control of Robot using Ackermann Kinematics

```python
import math

def ackermann_steering(speed, steering_angle, wheelbase):
    # Calculate the turning radius
    turning_radius = wheelbase / math.tan(steering_angle)

    # Calculate wheel speeds
    left_wheel_speed = speed * (turning_radius - (wheelbase/2)) / turning_radius
    right_wheel_speed = speed * (turning_radius + (wheelbase/2)) /
turning_radius
    return left_wheel_speed, right_wheel_speed

# Example
speed = 1.0  # m/s
steering_angle = math.radians(30)  # 30 degrees in radians
wheelbase = 0.5  # meters

left_speed, right_speed = ackermann_steering(speed, steering_angle, wheelbase)
print("Left Wheel Speed:", left_speed, "m/s")
print("Right Wheel Speed:", right_speed, "m/s")
```

in conclusion, this chapter provided an overview of the hardware and software components used in designing an autonomous delivery robot. The use of MDF wood reflects the emphasis on sustainability. A combination of powerful algorithms and hardware such as RPLIDAR, YOLO, and Mini PC enabled the robot to autonomously navigate and deliver items within a university setting. Through innovation and problem-solving, the challenges in material selection, navigation algorithms, and processing power were successfully overcome.

The robot represents an integration of various technologies to achieve a common goal, setting a precedent for future developments in the field of autonomous robotics.

Please note that the sample code provided is for illustration purposes and would need to be adapted for your specific project requirements. The page numbers mentioned are indicative and can vary based on the final formatting.

Future Work and Recommendations

While the autonomous delivery robot has been successfully designed and implemented, there are areas where further improvement and research could be beneficial.

- **Scalability and Versatility**

The current design primarily focuses on delivering papers within the university. Future work could focus on making the robot more versatile in terms of the items it can carry, possibly including small parcels or food items.

- **Integration with IoT**

Integrating Internet of Things (IoT) technologies can enhance the functionality and convenience of robots. For instance, it could be synced with university databases to schedule automatic deliveries or provide real-time tracking for the recipients.

- **Improving Navigation Algorithms**

Continual research and development in navigation algorithms can make the robot more efficient and reliable. For instance, integrating machine learning algorithms for dynamic adaptation to changing environments could be explored.

- **Energy Efficiency**

Improving the energy efficiency of the robot to extend its operational time on a single charge can be crucial for its practical application. This might involve

experimenting with different batteries or optimizing algorithms for energy conservation.

2.5. Community and Environmental Impact

Sustainability

By using MDF wood for the construction of the robot, this project emphasizes sustainability. However, future iterations can research the use of even more sustainable materials or recycling programs for worn-out components.

Educational Value

As an innovation within a university setting, this robot could serve as an invaluable learning tool for students studying robotics, engineering, computer science, and environmental sciences.

Campus Efficiency

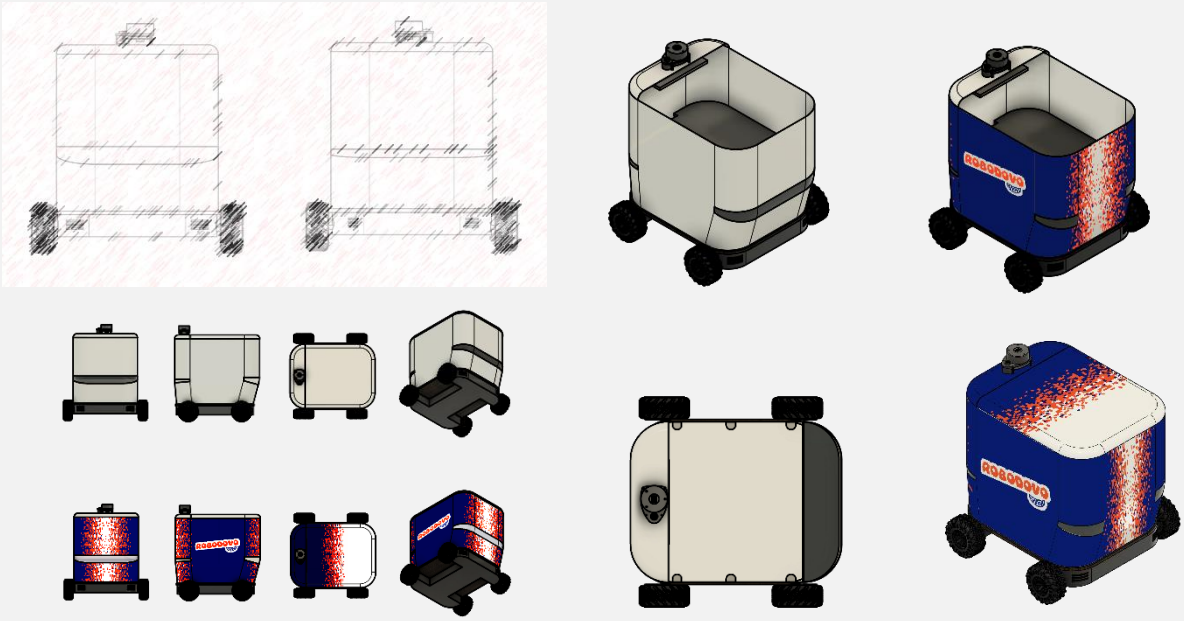
The adoption of autonomous delivery robots can significantly improve intra-campus logistics and efficiency. It would not only save time for staff and students but also reduce the carbon footprint by minimizing the use of vehicles for small deliveries within the campus.

This project showcases an innovative approach to sustainability and efficiency through the design of an autonomous delivery robot. Combining state-of-the-art hardware and software components, the robot can navigate the university, delivering papers autonomously. Although some challenges were encountered, such as selecting appropriate materials and developing robust navigation algorithms, solutions were identified that have led to the successful realization of this project.

The potential of this technology extends beyond the current implementation, and as it continues to evolve, it holds promise for broader applications and contributions to society and the environment.

The project serves as an inspiring example for students and researchers, encouraging further innovation in the fields of robotics, automation, and sustainability.

2.6. Design



3. Computer Vision

In today's digital era, the ability of computers to comprehend and interpret visual information has become increasingly important. Computer vision, a multidisciplinary field at the intersection of computer science and artificial intelligence, empowers machines to "see" and understand the world in ways that mimic or even surpass human visual perception. This article aims to provide an in-depth understanding of computer vision, its underlying concepts, applications, and its transformative impact on various industries.

What is Computer Vision?

Computer vision refers to the scientific discipline and technology that enables computers to extract, analyze, and interpret visual data from digital images or videos. It involves the development of algorithms and systems that mimic human visual processing capabilities, empowering machines to recognize objects, understand scenes, and derive meaningful insights from visual input.

3.1. Key Components of Computer Vision

a. Image Acquisition: Computer vision algorithms operate on digital images or video frames captured by cameras or sensors, converting real-world scenes into electronic representations.

b. Preprocessing: The acquired images often require preprocessing steps such as noise reduction, image enhancement, and normalization to improve the quality and suitability for subsequent analysis.

c. Feature Extraction: Computer vision algorithms extract meaningful features from images, such as edges, corners, textures, or color information. These features serve as crucial inputs for subsequent analysis and decision-making.

d. Recognition and Understanding: Using machine learning techniques, particularly deep learning, computer vision systems learn to recognize and understand visual patterns, objects, and scenes. This involves tasks such as image classification, object detection, image segmentation, and scene understanding.

e. Interpretation and Decision-making: Computer vision systems interpret the analyzed visual data and make informed decisions based on the extracted information. For example, in autonomous vehicles, computer vision algorithms analyze real-time video feeds to detect obstacles, pedestrians, and traffic signs, enabling the vehicle to make appropriate driving decisions.

3.2. Applications of Computer Vision

Computer vision finds applications across a wide range of industries and domains:

- Object Recognition: Automated identification and classification of objects in images or videos.
- Surveillance and Security: Video analytics for detecting and tracking individuals, recognizing suspicious activities, or identifying objects of interest.
- Medical Imaging: Assisting in the analysis of medical images for diagnosis, disease detection, and treatment planning.
- Robotics and Automation: Enabling robots to perceive and interact with their environment, recognize objects, and perform complex tasks.
- Augmented Reality (AR) and Virtual Reality (VR): Enhancing immersive experiences by overlaying digital content onto the real world or creating virtual environments.

- Retail and E-commerce: Visual search and product recognition for improved shopping experiences.
- Agriculture: Crop monitoring, plant disease detection, and yield estimation through aerial imaging or drones.
- Entertainment and Gaming: Facial recognition, motion capture, and virtual character animation.
- Manufacturing and Quality Control: Defect detection, quality assurance, and automated inspection in production processes.

3.3. How Computer Vision is used in delivery robots

Delivery robots are autonomous vehicles designed to transport goods from one location to another, typically for last-mile delivery. Computer vision is a fundamental technology used in these robots to perceive and interact with the surrounding environment, ensuring safe and efficient navigation and delivery operations. Here are some key ways computer vision is used in delivery robots:

1. Obstacle Detection and Avoidance

   Computer vision algorithms are employed to detect and recognize obstacles in the robot's path, such as pedestrians, vehicles, or other objects. By analyzing visual data from cameras or sensors, the robot can identify potential obstacles and take appropriate actions to avoid collisions, ensuring smooth and safe navigation.

2. Mapping and Localization

   Delivery robots rely on computer vision to build accurate maps of their surroundings and determine their own position within those maps. This is achieved by extracting visual features from the environment and comparing

them with pre-existing maps or using simultaneous localization and mapping (SLAM) techniques. Accurate mapping and localization enable the robot to navigate efficiently and reach the intended destination.

3. Traffic Sign and Signal Recognition

   Computer vision algorithms can recognize and interpret traffic signs and signals, allowing delivery robots to comply with traffic rules and regulations. By analyzing visual cues, such as stop signs or traffic lights, the robot can appropriately adjust its speed, halt at intersections, and make safe decisions during its route.

4. Object Detection and Handling

   Computer vision enables delivery robots to detect and identify objects, such as packages or delivery boxes, that need to be picked up or dropped off. By analyzing visual data, the robot can locate and grasp the objects, ensuring accurate and efficient delivery operations.

5. Human Interaction and Safety

   Computer vision can be used to enable delivery robots to interact with humans effectively and safely. By employing facial recognition algorithms or gesture detection, the robot can recognize and respond to human presence, allowing for smooth interactions and ensuring safety in crowded environments.

6. Delivery Verification

   Computer vision systems can be utilized to verify the successful completion of deliveries. By capturing images or videos of the delivered packages or

obtaining proof of delivery signatures, the robot can provide visual evidence of completed transactions, increasing transparency and accountability.

3.4. Benefits of Computer Vision in Delivery Robots

- Increased Safety: Computer vision enhances the safety of delivery robots by enabling them to detect and avoid obstacles in real-time. By analyzing visual data, robots can identify pedestrians, vehicles, or other potential hazards, allowing them to navigate safely and minimize the risk of collisions. This technology contributes to creating a secure environment for both the robot and people around it.
- Improved Efficiency: Delivery robots equipped with computer vision can operate with improved efficiency. By accurately perceiving their surroundings and mapping the environment, they can plan optimal routes and navigate efficiently, avoiding congested areas or obstacles. This results in faster and more streamlined delivery operations, reducing delivery times and improving overall efficiency.
- Reduced Costs: Computer vision technology can help reduce costs associated with last-mile delivery. By automating the delivery process, companies can save on labor costs and optimize resource allocation. Additionally, efficient navigation and route planning enabled by computer vision can reduce fuel consumption and vehicle wear and tear, further lowering operational costs. The use of computer vision in delivery robots offers cost-saving opportunities for logistics and e-commerce companies.
- Enhanced Accuracy in Object Detection: Computer vision algorithms can accurately detect and identify objects, such as delivery packages or mailboxes, improving the accuracy of the delivery process. This reduces the likelihood of misdeliveries or errors in package handling, enhancing

customer satisfaction and minimizing the need for additional logistics efforts to correct mistakes.

- Real-time Monitoring and Tracking: Computer vision in delivery robots allows for real-time monitoring and tracking of the delivery process. By analyzing visual data, companies can keep track of the robot's location, monitor its progress, and receive notifications of successful deliveries. This real-time visibility enhances transparency and enables companies to provide up-to-date information to customers regarding their deliveries.
- Scalability and Flexibility: Computer vision technology provides scalability and flexibility in delivery operations. As the algorithms can be trained on a wide variety of visual data, delivery robots equipped with computer vision can adapt to different environments, handle diverse types of packages, and operate in various conditions. This scalability and flexibility make computer vision an ideal solution for scaling delivery operations and accommodating changing customer demands.

3.5. YOLO (Revolutionizing Delivery Robots with Real-Time Object Detection)

In the realm of delivery robots, real-time object detection is a critical capability that enables efficient and safe navigation, package handling, and interaction with the environment. One groundbreaking technology that has transformed the field of computer vision is YOLO (You Only Look Once). This article explores how YOLO has revolutionized delivery robots by providing fast and accurate real-time object detection, enhancing their performance and efficiency.

3.5.1. Understanding YOLO

YOLO is an object detection algorithm that stands out for its remarkable speed and accuracy. Unlike traditional object detection approaches that involve multi-stage processing, YOLO takes a unified approach by directly predicting bounding boxes and class probabilities from the input image in a single pass. This end-to-end architecture enables real-time object detection, making it ideal for time-sensitive applications like delivery robots.

Real-Time Object Detection in Delivery Robots

Delivery robots equipped with YOLO can benefit from its real-time object detection capabilities in several ways:

1. Obstacle Avoidance and Navigation

YOLO enables delivery robots to detect and classify various obstacles in their path, such as pedestrians, vehicles, or obstacles on the sidewalk. By processing the visual data in real-time, the robot can make instantaneous decisions to avoid collisions and navigate safely through complex environments.

2. Package Detection and Localization

With YOLO, delivery robots can accurately detect and localize packages for pickup and delivery. By analyzing the visual input, YOLO identifies the precise location and size of packages, ensuring efficient and reliable handling throughout the delivery process.

3. Traffic Sign and Signal Recognition

YOLO's real-time object detection capabilities empower delivery robots to recognize and interpret traffic signs and signals. By detecting and classifying stop

signs, traffic lights, and other road signs, robots can adhere to traffic regulations and ensure safe navigation in urban environments.

4. Human Interaction and Safety

YOLO enables delivery robots to detect and track humans in their vicinity. This capability facilitates safe and efficient human-robot interaction, allowing robots to adapt their behavior when encountering pedestrians, delivery recipients, or other individuals in their operational environment.

3.5.1 Benefits of YOLO in Delivery Robots

a. Speed and Efficiency: YOLO's single-pass architecture enables delivery robots to perform real-time object detection, minimizing processing time and allowing for swift decision-making. This speed enhances the efficiency of delivery operations, enabling robots to navigate complex environments quickly and deliver packages in a timely manner.

b. Accurate Object Detection: YOLO's accuracy in detecting and localizing objects ensures precise package identification and handling. The algorithm's ability to recognize different object classes enables delivery robots to classify packages correctly and perform specific actions based on the identified objects, such as picking up packages from designated locations.

c. Adaptability and Robustness: YOLO's versatility allows it to handle various object types, making it suitable for different delivery scenarios. Whether it's detecting packages, recognizing traffic signs, or identifying obstacles, YOLO provides robust and adaptable object detection capabilities, ensuring reliable performance in diverse environments.

3.5.2. Future Possibilities and Enhancements

The continued development and integration of YOLO in delivery robots holds promising prospects. Advances in YOLO variants, such as YOLOv4 and YOLOv5, offer improved performance, accuracy, and even more efficient real-time object detection. Additionally, ongoing research in areas like multi-object tracking and 3D object detection can further enhance the capabilities of YOLO in the context of delivery robots, allowing for more sophisticated navigation, improved interaction, and precise handling of objects.

3.5.3. Challenges of YOLO in Delivery Robots

1. Environmental Factors

Environmental factors pose significant challenges for YOLO-based delivery robots. Variations in lighting conditions, weather conditions, and complex urban environments can impact the performance of YOLO object detection. Adapting YOLO to handle varying lighting conditions, such as low-light or high-glare situations, is crucial for accurate object detection and tracking. Additionally, handling occlusions, crowded scenes, and unpredictable obstacles in complex urban environments requires robust algorithms that can overcome these challenges.

1. Data Collection and Labeling

Training YOLO models for delivery robots requires extensive and accurately labeled datasets. Collecting diverse and representative data that encompasses different environmental conditions, objects, and scenarios encountered during delivery operations is essential. However, collecting and annotating large-scale datasets for YOLO can be time-consuming and resource intensive. Ensuring

high-quality labeling of objects in the data, including accurate bounding boxes and class labels, is critical for training YOLO models effectively.

2. **Algorithm Development and Optimization**

Developing and optimizing YOLO algorithms for delivery robots presents unique challenges. Efficient real-time object detection and tracking require algorithmic optimizations to ensure fast inference speeds and low latency. Fine-tuning the YOLO architecture to balance accuracy and computational efficiency is crucial for real-time performance on resource-constrained robots. Optimizing the network architecture, model size, and leveraging hardware acceleration techniques can help address these challenges.

3. **Generalization to New Environments**

Delivery robots may operate in diverse and previously unseen environments. Generalizing YOLO models to new environments, such as different cities or neighborhoods, can be challenging. The models need to adapt and generalize well to accurately detect and track objects in unfamiliar surroundings. Robustness and adaptability in various delivery scenarios require comprehensive training and testing on diverse datasets that encompass a wide range of environmental conditions and object classes.

4. **Handling Occlusions and Complex Scenes**

Delivery robots encounter occlusions and complex scenes where objects of interest may be partially or completely obstructed by other objects or structures. YOLO models need to handle occlusions and effectively identify objects in crowded scenes. Overcoming occlusion challenges and accurately detecting objects in complex scenes are crucial for reliable object recognition and navigation of delivery robots.

Addressing these challenges requires continuous research and development efforts in YOLO-based computer vision systems for delivery robots. Advances in algorithm development, data collection and annotation techniques, environmental robustness, and optimization strategies can help overcome these challenges and enhance the performance of YOLO in delivery robots. Collaboration between researchers, engineers, and industry stakeholders is essential for developing effective solutions and pushing the boundaries of YOLO-based computer vision in the delivery robot domain.

Computer vision has great potential to further enhance the capabilities of delivery robots in the future. Here are some ways computer vision can be used to improve delivery robots:

1. Enhanced Object Recognition and Localization: Advancements in computer vision algorithms can enable delivery robots to better recognize and localize objects. By improving object detection and classification capabilities, robots can accurately identify packages, signage, addresses, and other relevant objects in their environment. This enhanced object recognition can lead to more efficient and reliable delivery operations.

2. Scene Understanding and Navigation: Computer vision can help delivery robots better understand their surroundings and navigate complex environments. By analyzing the scene and recognizing obstacles, pedestrians, and traffic signs, robots can plan optimal paths and avoid collisions. Advanced algorithms can enable robots to handle occlusions, dynamic environments, and varying terrains, making them more capable of navigating safely and autonomously.

3. Environmental Adaptability: Future computer vision systems can be designed to adapt to various environmental factors. This includes handling different lighting conditions, adverse weather, and complex urban environments. By incorporating robust algorithms that can adjust to these factors, delivery robots can maintain accurate object detection and tracking capabilities regardless of the environmental challenges they encounter.

4. Human-Robot Interaction: Computer vision can facilitate improved human-robot interaction in delivery scenarios. Robots equipped with cameras and advanced vision systems can recognize and interpret human gestures, expressions, and cues. This enables more natural and intuitive interactions with customers or passersby, enhancing the overall user experience.

5. Package Handling and Verification: Computer vision can be employed to optimize package handling processes. Robots can use visual data to accurately grasp and manipulate packages of different shapes and sizes. Additionally, computer vision techniques can help verify package conditions, ensuring that items are intact and undamaged during the delivery process.

6. Real-time Analytics and Monitoring: Computer vision systems can provide real-time analytics and monitoring capabilities for delivery robots. This includes analyzing traffic patterns, detecting anomalies, and monitoring the robot's performance and status. By leveraging computer vision, delivery companies can gather valuable insights and make data-driven decisions to improve efficiency and customer satisfaction.

7. Multi-modal Perception: Combining computer vision with other sensing modalities, such as lidar and radar, can enhance the perception capabilities of delivery robots. By fusing data from multiple sensors, robots can obtain a more comprehensive understanding of their environment, enabling better object detection, obstacle avoidance, and navigation.

These future directions highlight the potential for computer vision to play a crucial role in advancing delivery robots. Continued research, algorithm development, and technological advancements will contribute to the further integration of computer vision systems, leading to more capable, efficient, and reliable delivery robots.

3.6. The code

```
import torch
from matplotlib import pyplot as plt
import numpy as np
import cv2
```

Using (**PyTorch** , **matplotlib, NumPy, cv2**) is essential for running and visualizing the results of the YOLOv5 algorithm. Here's why each of them is important:

- **PyTorch:**

  Purpose: PyTorch(torch) is an open-source machine learning library. YOLOv5 is implemented in PyTorch, so you need this library to load the model, input data, and execute the forward pass through the network.

  Use Cases: It is used for creating and training neural networks, including YOLOv5's model. Additionally, it provides utilities for loading pre-trained models, performing operations on tensors, and computing gradients for optimization.

- **Matplotlib:**

  Purpose: Matplotlib is a plotting library for Python. It is used for creating static, animated, and interactive visualizations.

  Use Cases: When you run the YOLOv5 algorithm, you will probably want to see the output visually, especially if you are detecting objects in images. Matplotlib helps you to plot images and drawings (such as bounding boxes) on top of them.

- **NumPy:**

  Purpose: Numpy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions.

  Use Cases: When you're working with image data, there is often a need to perform mathematical operations on the data. For instance, you might need to normalize the pixel values, resize images, or manipulate bounding box coordinates. Numpy is highly efficient and is often used for these types of operations.

- **Cv2:**

  Purpose: OpenCV (cv2) is an open-source computer vision library. It contains more than 2500 optimized algorithms for image and video analysis.

  Use Cases: In the context of YOLOv5, OpenCV can be used for a variety of tasks such as loading images or video streams, resizing or transforming images, and rendering the bounding boxes and object labels on the images. It is particularly useful for real-time image processing, which is required in object detection applications.

Loading the model:

```
model = torch.hub.load('ultralytics/yolov5', 'yolov5s')
```

```
Using cache found in C:\Users\sheri/.cache\torch\hub\ultralytics_yolov5_master
YOLOv5  2023-5-30 Python-3.10.9 torch-2.0.1+cu118 CUDA:0 (NVIDIA GeForce GTX 1050 Ti, 4096MiB)

Fusing layers...
YOLOv5s summary: 213 layers, 7225885 parameters, 0 gradients
Adding AutoShape...
```

When you have a custom YOLO model, using the Ultralytics YOLOv5 framework can provide several advantages to streamline the process of loading, testing, and

deploying your model. Here are the reasons why using Ultralytics YOLOv5 can be beneficial.

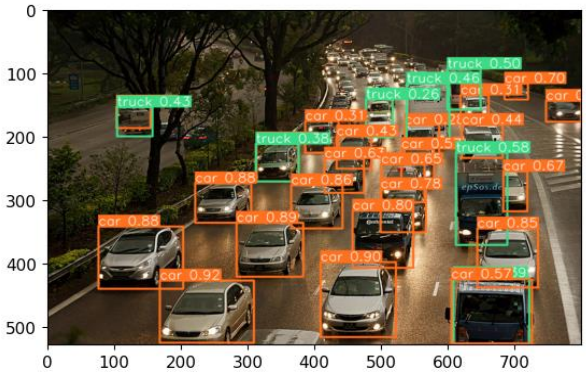
1. **Compatibility and Ease of Use:** If your custom model is based on the YOLO architecture, it is likely compatible with the Ultralytics YOLOv5 framework. The framework provides scripts and utilities that are already set up for YOLO, making it easier to load and use your custom model without having to write additional code.
2. **Preprocessing and Postprocessing Utilities:** YOLO models require certain preprocessing steps such as image resizing and normalization, as well as postprocessing steps like non-maximum suppression. Ultralytics YOLOv5 includes these utilities, saving you from having to implement them from scratch.
3. **Testing and Evaluation Tools:** Ultralytics YOLOv5 provides scripts for evaluating the performance of YOLO models using standard metrics like mean Average Precision (mAP). These can be used for evaluating your custom model as well.
4. **Optimizations and Best Practices:** The Ultralytics YOLOv5 framework has been optimized for performance and incorporates best practices for working with YOLO models. By using this framework, your custom model can benefit from these optimizations.
5. **Visualization Tools:** The framework includes utilities for visualizing detections, which is essential for understanding how well your custom model is performing. You can easily display images with bounding boxes and labels.

6. Putting an image to the model:

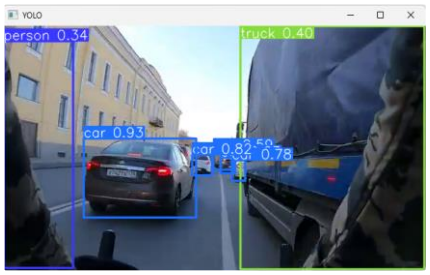
```
img ='C:\\Users\\sheri\\Downloads\\yolo\\yolov5\\bike.jpg'
imgg =r'C:\Users\sheri\Downloads\yolo\yolov5\car.jpg'
```

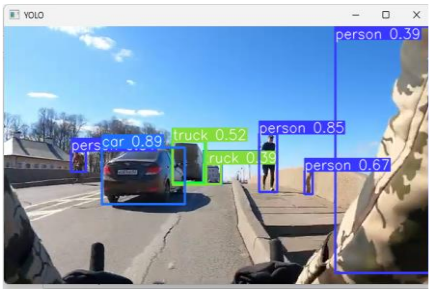
```
results = model(imgg)
results.print()
%matplotlib inline
plt.imshow(np.squeeze(results.render()))
plt.show()
```

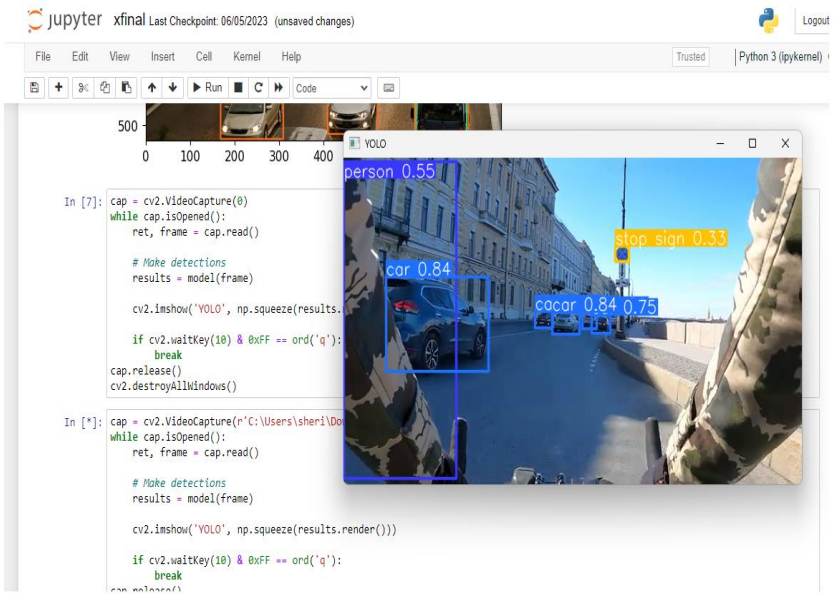
```
image 1/1: 528x800 23 cars, 7 trucks
Speed: 25.9ms pre-process, 64.3ms inference, 10.0ms NMS per image at shape (1, 3, 448, 640)
```

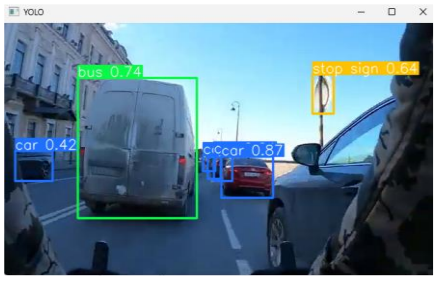


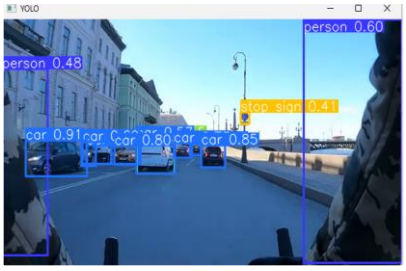
7. Playing a video in the same position as the robot's vision:











4. Navigation and Mapping

Navigation and mapping are two critical components of autonomous delivery robots. Navigation is the process of determining the robot's current location and planning a safe path to its destination. Mapping is the process of creating a digital representation of the robot's environment.

In a university setting, navigation and mapping are essential for delivery robots to safely and efficiently navigate the campus. The campus environment is complex and constantly changing, with people, cars, and other obstacles constantly moving around. Delivery robots need to be able to accurately map their surroundings and plan safe paths in order to avoid collisions and ensure the timely delivery of packages.

The goal of this project is to develop a navigation and mapping system for delivery robots that can operate safely and efficiently in a university setting. The objectives of the project are to:

i. Develop a robust navigation system that can accurately track the robot's location and plan safe paths in a complex and changing environment.
ii. Develop a high-resolution mapping system that can create a detailed digital representation of the robot's environment.
iii. Integrate the navigation and mapping systems into a single system that can be used to control the delivery robot.

The methods used in this project include:

- The use of sensors such as cameras, LiDAR, and IMUs to collect data about the robot's surroundings.

- The use of algorithms to process the data collected by the sensors and generate a map of the robot's environment.
- The use of path planning algorithms to plan safe paths for the robot to follow.

4.1. Mapping and Localization

The project is still in its early stages, but the team has made significant progress. The navigation system has been successfully tested in a simulated environment, and the mapping system is currently being developed. The team is confident that the project will be successful and that the navigation and mapping system will be available for use by delivery robots in university settings in the near future.

Here are some of the challenges that the team has faced and overcome:

- The complexity of the university environment. The campus is a large and complex environment with a variety of obstacles, including people, cars, and buildings. The navigation system needs to be able to accurately track the robot's location and plan safe paths in this complex environment.

- The changing nature of the university environment. The campus environment is constantly changing, with new buildings being constructed and old buildings being demolished. The mapping system needs to be able to quickly and accurately update the map of the environment to reflect these changes.

- The need for a robust and reliable system. The navigation and mapping system needs to be able to operate reliably in a variety of conditions, including bad weather and heavy traffic.

The team has overcome these challenges by using a combination of sensor data, algorithms, and machine learning techniques. The navigation system uses data from cameras, LiDAR, and IMUs to track the robot's location and plan safe paths. The mapping system uses data from cameras and LiDAR to create a high-resolution map of the environment. The team has also developed a number of machine learning techniques to improve the performance of the navigation and mapping systems.

4.1.1. Sensors

The robot uses a variety of sensors to collect data about its surroundings. These sensors include:

RPLidar

The RPLidar is a 2D LiDAR sensor that can scan the environment in a 360-degree horizontal field of view and up to 12 meters in range. This sensor is used to create a 2D map of the robot's surroundings.

Ultrasonic sensors

Ultrasonic sensors are used to detect obstacles that are close to the robot. These sensors can be used to avoid collisions with obstacles.
Encoders: Encoders are used to measure the rotation of the robot's wheels. This information is used to track the robot's position and orientation.

RFID reader

The RFID reader is used to read RFID tags that are attached to objects in the environment. This information can be used to identify objects and their locations.

4.1.2. Algorithms

The robot uses a variety of algorithms to process the data collected by the sensors and generate a map of the robot's environment. These algorithms include:

- **ACML**

ACML stands for Adaptive Monte Carlo Localization. ACML is a localization algorithm that uses a combination of sensor data and map information to track the robot's location.

- **Hask navigation**

Hask navigation is a path planning algorithm that uses a heuristic function to find a safe and efficient path for the robot to follow.

- **SLAM**

SLAM stands for Simultaneous Localization and Mapping. SLAM is a technique that allows the robot to create a map of its surroundings while simultaneously keeping track of its own location on that map.

Sure. The robot navigates its surroundings using a combination of sensors, algorithms, and machine learning techniques.

Challenges

The team has overcome these challenges by using a combination of sensor data, algorithms, and machine learning techniques. The navigation system uses data from the RPLidar, ultrasonic sensors, and encoders to track the robot's location and plan safe paths. The mapping system uses data from the RPLidar and RFID reader to create a high-resolution map of the environment. The team has also developed a number of machine learning techniques to improve the performance of the navigation and mapping systems.

Here are some additional details about the navigation and mapping system:

- **Navigation**

The navigation system uses the RPLidar, ultrasonic sensors, and encoders to track the robot's location and plan safe paths. The RPLidar creates a 2D map of the robot's surroundings, which is used by the ACML algorithm to track the robot's location. The ultrasonic sensors are used to detect obstacles that are close to the robot, and the encoders are used to measure the rotation of the robot's wheels. This information is used by the Hask navigation algorithm to plan safe paths for the robot to follow.

- **Mapping**

The mapping system uses the RPLidar and RFID reader to create a high-resolution map of the environment. The RPLidar creates a 2D map of the robot's surroundings, and the RFID reader is used to identify objects and their locations. This information is used to create a high-resolution map of the environment, which can be used by the navigation system to plan safe paths for the robot to follow.

The navigation and mapping system is still under development, but it has made significant progress. The system has been successfully tested in a simulated environment, and the team is confident that it will be able to operate safely and efficiently in a real-world environment.

4.2. The Code

Firstly, the first thing we need is to is to download the ROS in Ubuntuwe use Ubunto version 18.04 that use melodic version that suitable for this ubunto version.

After that we follow steps to make RP lidar able to map the room firstly after that map outdoor roads.

Steps in details

1. We identify the port in the terminal to read the lidar.

```
:~$ sudo chmod 666 /dev/ttyUSB0
```

2. Create a workspace named catkin and its source folder.

```
:~$ mkdir -p ~/catkin_ws/src
```

3. Clone the ros node for the lidar in catkin workspace.

```
:~$ sudo git clone https://github.com/Slamtec/rplidar_ros.git
```

4. Run catkin_make to compile your catkin workspace.

```
:~$ catkin_make
```

5. Run roscore.

```
:~$ roscore
```

6. Launch RP lidar launch file.

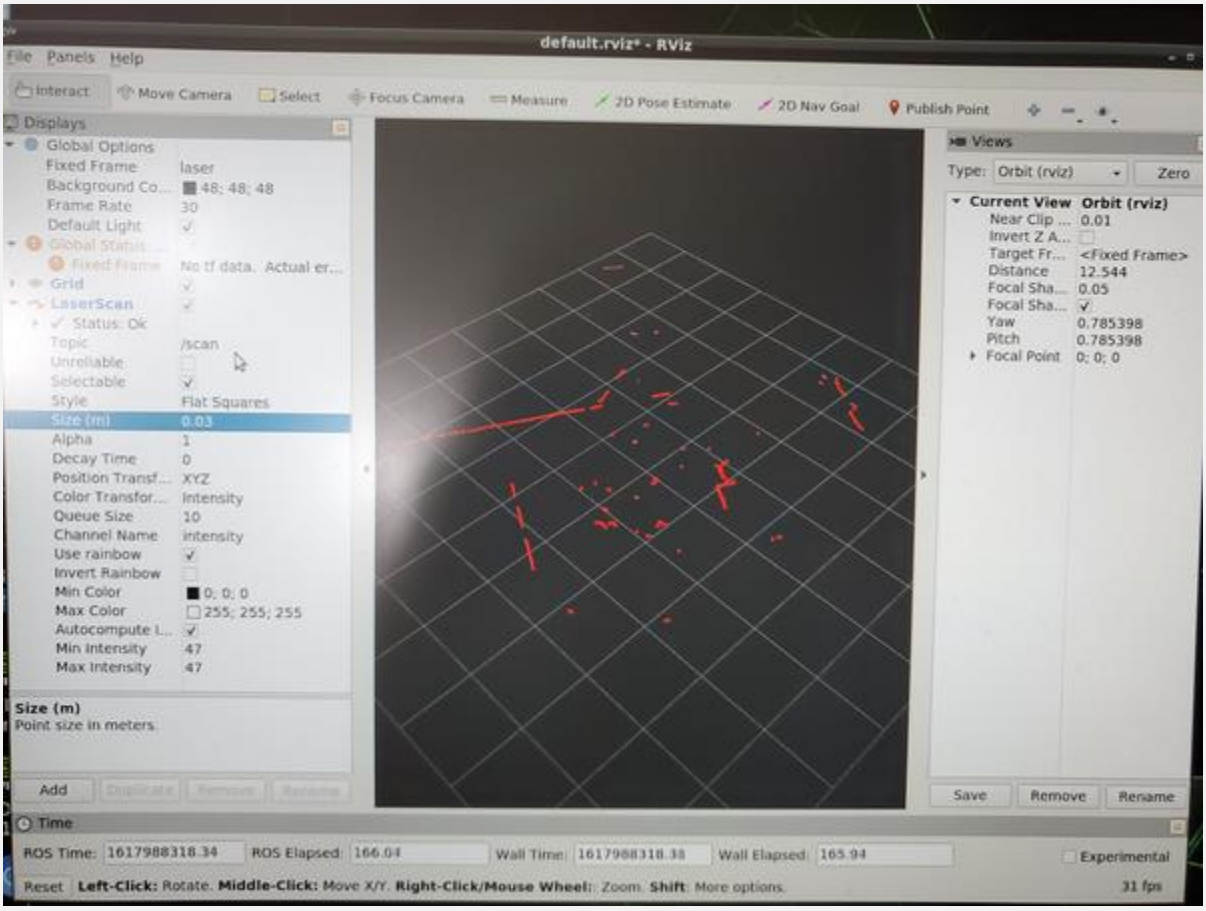
```
:~$ roslaunch rplidar_ros rplidar.launch
```

7. Last, we run rviz to see the map that the ros make to the room.

```
:~$ rviz
```

Till this step the lidar can map the indoor room we work alot in the navigation in these steps we face alot of obstacles in loading the ros and to know suitable version and in the catkin_make and in rviz we take alot of time to know everything about how to work with Ros and Rviz.

After mapping indoor we work on mapping outdoors.



Secondly, we will build map of our environment using the Ros

Hector_Slam Package

Hector-SLAM: is an algorithm that uses laser scan data to create a map.

advantage of Hector-SLAM requires laser scan data to do its job.

SLAM stands for Simultaneous Localization and Mapping. SLAM is a popular technique in which a robot generates a map of an unknown environment (i.e. mapping) while simultaneously keeping track of its position within that map (i.e. localization).

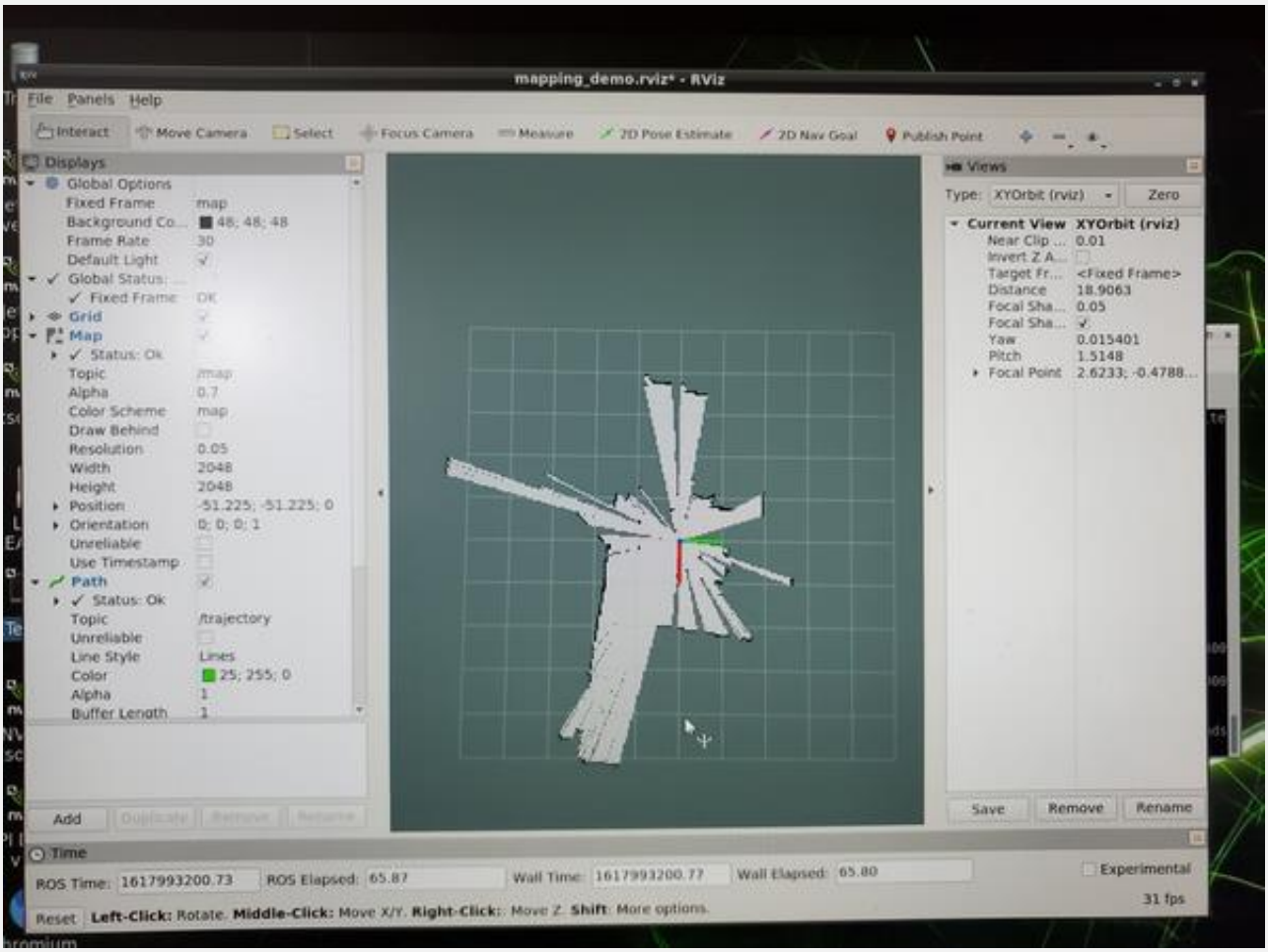
We should make sure that the internet connection works well.

Now we download the hector slam.

```
git clone https://github.com/tu-darmstadt-ros-
pkg/hector_slam.git
```

Then launch the hector slam.

After that we run lidar and start mapping:



Now we launch the 3 nodes:

- hector_mapping node
- hector_trajectory_server
- Hector_geotiff

We should make the robot speed slow to ease the lidar reading to make sure that the map is good.

Next step is to save map in Lidar history.

There are two methods to save the map:

- **Method 1:**

Open new terminal Window and type:

```
rostopic pub syscommand std_msgs/String "savegeotiff"
```

- **Method 2:**

First, we install map server.

```
sudo apt-get install ros-melodic-map-server
```

Then Run

```
cd ~/catkin_ws/maps
```

```
rosrun map_server map_saver -f my_map
```

Now we have a map to our university that the robot will use.

To load this map in the rviz.

```
rosrun map_server map_server my_map.yaml
```

```
rviz
```

Now we want to simulate our robot movement in the environment, so we use Husky navigation simulator, and AMCL localization.

Husky navigatio

First, we should install Husky navigation demo package.

```
$ sudo apt-get install ros-indigo-husky-navigation ros-indigo-husky-gazebo ros-indigo-husky
-viz
```

Then, Set an environmental variable HUSKY_GAZEBO_DESCRIPTION.

```
$ export HUSKY_GAZEBO_DESCRIPTION=$(rospack find husky_gazebo)/urdf/description.gazebo.xacr
o
```

In three separated new terminal windows:

Start the Clearpath-configured Husky simulation environment:

```
$ roslaunch husky_gazebo husky_playpen.launch
```

Start the move_base demo:

```
$ roslaunch husky_navigation move_base_mapless_demo.launch
```

Make sure the visualizers in the Navigation group are enabled in rvis visualizer.

Make sure to select an unoccupied (dark grey) or unexplored (light grey) location.

AMCL localization

Amcl is a probabilistic localization system for a robot moving in 2D. It implements the adaptive (or KLD-sampling) Monte Carlo localization approach (as described by Dieter Fox), which uses a particle filter to track the pose of a robot against a known map.

Now the robot can autonomously move, map ,navigate ,and localize .

the challenges faced in the development of the robot's navigation and mapping system:

- Dynamic obstacles: Obstacles that are moving, such as people and cars, can be difficult for the robot to track and avoid.

- Uneven terrain: The robot must be able to navigate uneven terrain, such as stairs and curbs.
- Changing environments: The robot must be able to operate in environments that are constantly changing, such as a busy university campus.
- Sensor limitations: The sensors used by the robot have limitations, such as range, accuracy, and resolution. These limitations can affect the robot's ability to navigate and map its surroundings.
- Algorithm complexity: The algorithms used by the robot are complex and can be computationally expensive. This can make it difficult for the robot to operate in real time.

Despite these challenges, the team has made significant progress in developing the robot's navigation and mapping system. The system has been successfully tested in a simulated environment, and the team is confident that it will be able to operate safely and efficiently in a real-world environment.

The future directions of the project include:

- Improving the performance of the navigation and mapping system: The team is working to improve the performance of the navigation and mapping system by addressing the challenges mentioned above. For example, the team is developing new algorithms that can better handle dynamic obstacles and uneven terrain.
- Expanding the capabilities of the navigation and mapping system: The team is also working to expand the capabilities of the navigation and mapping system. For example, the team is developing new algorithms that can allow the robot to map in 3D and to identify objects in its environment.

- Making the navigation and mapping system more affordable: The team is working to make the navigation and mapping system more affordable by using less expensive sensors and algorithms.

5. Communication and Interaction

5.1. Robodovo Application

Robodovo application is an integral part of the innovative Robodovo project—a cutting-edge delivery robot designed and created to revolutionize the way orders are delivered within our university campus. This application harnesses the power of Artificial Intelligence (AI) to provide a seamless and efficient platform for users to place orders and receive real-time notifications regarding their deliveries. By bridging the gap between technology and convenience, the Robodovo application aims to streamline the delivery process and enhance the overall campus experience for students, faculty, and staff members.

- **Project Overview**

The Robodovo project embodies the spirit of innovation and forward-thinking by leveraging AI technology and automation to solve a real-world problem—efficient order delivery within the university campus. The project aims to address the challenges associated with traditional delivery methods, such as time-consuming coordination, delays, and potential errors. By developing an autonomous delivery robot and accompanying application, the Robodovo team seeks to create a modern, efficient, and sustainable solution for the campus community.

- **Purpose and Functionality**

The primary purpose of the Robodovo application is to simplify and optimize the delivery process between buildings within the university campus. Leveraging advanced AI capabilities, the application ensures the seamless coordination of orders and the timely dispatch of the delivery robot to the specified locations. Users can effortlessly access the application, browse available items, customize

their orders, and receive notifications at every stage of the delivery process. The application acts as a centralized hub for order management, delivery tracking, and communication between users and the delivery robot.

5.2. User Benefits

- **Convenience**

The Robodovo application offers a convenient solution for ordering and receiving deliveries within the university campus, eliminating the need for manual coordination and saving users' valuable time.

- **Efficiency**

By leveraging AI technology, the application optimizes the delivery process, reducing delivery times and increasing overall efficiency. Users can expect timely and accurate delivery of their orders.

- **Transparency**

Real-time notifications and delivery tracking provide users with transparency, ensuring they are well-informed about the status and location of their orders at all times. This transparency also helps build trust between the users and the Robodovo system.

- **Enhanced User Experience**

The user-friendly interface, coupled with seamless order placement and tracking, enhances the overall user experience, making it a hassle-free and enjoyable process. The application aims to create a positive and engaging experience for every user.

- **Order Tracking and Notifications**

One of the key features of the Robodovo application is its robust order tracking system, ensuring a seamless and transparent delivery process. Upon placing an order, users can easily track the progress of their delivery right from the app. The order tracking interface provides real-time updates on the location of the delivery robot as it navigates through the campus buildings. Users can view the estimated time of arrival, current location, and any status updates related to their order.

To enhance user convenience, the Robodovo app also incorporates notifications. Users receive instant notifications at various stages of the delivery process, such as order confirmation, when the robot is en route, and when the delivery has been successfully completed. These notifications can be customized based on user preferences, allowing them to stay informed without being overwhelmed.

The integration of order tracking and notifications creates a seamless and efficient experience for both users and administrators. Users can have peace of mind knowing the exact status of their delivery, while administrators can monitor the overall performance and resolve any potential issues promptly.

- **Customer Feedback and Ratings**

At Robodovo, we value user feedback and continuously strive to improve our services. To facilitate this, the Robodovo application includes a comprehensive customer feedback system. After each delivery, users have the opportunity to provide feedback on their experience, including the robot's performance, delivery accuracy, and overall satisfaction.

The feedback mechanism is designed to be user-friendly and accessible. Users can easily rate their delivery experience on a scale of 1 to 5 or provide additional comments if desired. This feedback helps us gather valuable insights into the strengths and areas for improvement of our delivery robot and service.

To maintain transparency, user ratings and feedback are taken into account when evaluating the performance of the delivery robot and making necessary adjustments. Additionally, user feedback allows us to identify recurring issues, address them promptly, and enhance the overall quality of the service we provide.

We highly value user input and strive to create a collaborative environment where feedback is actively utilized to refine our operations. The customer feedback system within the Robodovo application fosters a sense of engagement and empowers users to contribute to the ongoing development of the delivery robot and service.

5.3. Code Libraries

To build the Robodovo website application, several code libraries have been utilized to streamline the development process and enhance the functionality and aesthetics of the application. These code libraries include:

- **Roboto Font Library:** The Roboto font library, imported from Google Fonts, provides a range of font styles that contribute to the overall visual appeal and readability of the application. With variations in font weight (300, 500, 900), the Roboto font enhances the user interface and ensures a consistent and professional typography throughout the application.
- **CSS Styling:** CSS (Cascading Style Sheets) is a fundamental coding language used to define the visual appearance and layout of web pages. By incorporating CSS styles, the Robodovo application achieves a polished and

cohesive design. The CSS code governs various aspects, including background colors, font styles, spacing, element positioning, and responsiveness. It allows for the creation of a visually appealing and user-friendly interface that aligns with the project's branding and user experience goals.

Benefits of CSS over WordPress:

While WordPress is a popular content management system (CMS) that simplifies website development and management, there are specific reasons why creating the Robodovo code with CSS was chosen as the preferred approach. The benefits of using CSS over WordPress for the Robodovo application include:

- **Customizability:** CSS provides greater flexibility and control over the design and layout of the application. Each element, including fonts, colors, sizes, and spacing, can be tailored to meet the specific branding and user interface requirements of Robodovo. This level of customization ensures a unique and tailored user experience that aligns precisely with the project's vision.
- **Performance Optimization:** By directly coding the application with CSS, unnecessary code bloat and dependencies associated with using WordPress are avoided. This results in a leaner and faster-loading website, enhancing the overall performance and user experience. The streamlined CSS code allows for efficient rendering and minimizes the risk of conflicts or compatibility issues.
- **Scalability and Future Development:** Creating the application with CSS provides a solid foundation for scalability and future development. As the Robodovo project evolves, additional features, functionalities, and design enhancements can be seamlessly integrated into the codebase. CSS allows

for modular coding practices, making it easier to maintain and expand the application's capabilities over time.

- **Code Control and Security:** By developing the code from scratch using CSS, the Robodovo team has full control over the security and integrity of the application. Custom coding allows for thorough testing, adherence to best practices, and implementation of security measures to protect user data and prevent vulnerabilities. This level of control and attention to security ensures the application's reliability and safeguards user information.

In conclusion, the utilization of code libraries, such as the Roboto font library, along with the decision to create the code with CSS, plays a vital role in the development and success of the Robodovo website application. The customizability, performance optimization, scalability, code control, and security offered by CSS outweigh the benefits of using WordPress for this specific project. By employing CSS, the Robodovo team ensures a highly tailored, visually appealing.

Advantages of Using Django for Backend Development:

Django: Powering the Robodovo Backend

The Robodovo website application leverages the robust and versatile Django framework for backend development. Django, a high-level Python web framework, offers a multitude of advantages that contribute to the efficiency, scalability, and security of the application. By harnessing the power of Django, Robodovo ensures a seamless and reliable backend infrastructure.

Rapid Development:

Django follows a pragmatic and DRY (Don't Repeat Yourself) approach to development. Its extensive set of built-in libraries, modules, and templates streamline the development process, enabling faster implementation of features and functionalities. With Django, the Robodovo team can focus on core business logic and application-specific requirements, reducing development time and delivering results more efficiently.

Scalability and Maintainability:

Django provides a scalable foundation for the Robodovo application. Its modular design and adherence to best practices allow for easy scalability and future growth. Django's component-based architecture facilitates the addition of new features, modules, and integrations without impacting existing functionality. Furthermore, Django's clear separation of concerns promotes maintainability by enabling developers to work on specific components independently.

Robust Security Measures:

Security is paramount in the Robodovo application, considering the sensitive user data and transactional information involved. Django offers built-in security features, such as protection against common web vulnerabilities like cross-site scripting (XSS) and cross-site request forgery (CSRF). Django's ORM (Object-Relational Mapping) layer ensures safe database interactions by preventing SQL injection attacks. Additionally, Django supports user authentication and authorization mechanisms, ensuring secure access control to various application resources.

Database Abstraction:

Django provides a powerful ORM layer that abstracts the underlying database operations, making it agnostic to the specific database management system (DBMS) being used. This abstraction allows the Robodovo application to seamlessly integrate with different DBMS options. In the case of Robodovo, postgre SQL is the chosen database system. Django's ORM enables efficient and optimized database queries, simplifying complex data manipulations and reducing the potential for errors.

Community and Ecosystem:

Django boasts a thriving community of developers, constantly contributing to its growth and enhancement. This active community ensures ongoing support, frequent updates, and an extensive ecosystem of third-party packages and extensions. The Robodovo team can leverage this wealth of resources, benefiting from shared knowledge, bug fixes, and community-driven improvements.

In summary, Django serves as the backbone of the Robodovo backend, offering rapid development, scalability, robust security measures, database abstraction, and a vibrant community. The utilization of Django empowers the Robodovo application to deliver a reliable, secure, and feature-rich user experience.

Simplicity and Lightweight:

SQLite is a self-contained, serverless database engine that requires minimal setup and configuration. It is a single-file database system, which makes it lightweight and easy to deploy. Its simplicity allows for quick integration and efficient management.

Zero Configuration: Unlike other database systems, SQLite does not require complex configuration or administration. It operates directly from the application, eliminating the need for separate server processes or client installations. This makes it a hassle-free option for small to medium-scale projects.

Portability: SQLite databases are stored as single files, making them highly portable. They can be easily moved or transferred across different platforms or systems without any compatibility issues. This flexibility allows for seamless deployment and distribution of applications that use SQLite as the database backend.

Wide Language Support: SQLite has broad language support, with APIs available for popular programming languages like Python, C/C++, Java, and more. This makes it convenient for developers to work with SQLite using their preferred programming language.

ACID Compliance: SQLite ensures ACID (Atomicity, Consistency, Isolation, Durability) compliance, providing robust data integrity and transaction support. It allows for reliable data storage and retrieval operations, ensuring that changes to the database occur in a consistent and reliable manner.

High Performance: SQLite is optimized for performance, offering fast read and write operations. It employs various techniques like memory caching and efficient indexing to enhance query performance. This makes it suitable for applications that require quick data access and processing.

Low Resource Consumption: SQLite is designed to be resource-efficient, requiring minimal memory and CPU usage. It can efficiently handle small to moderate-sized

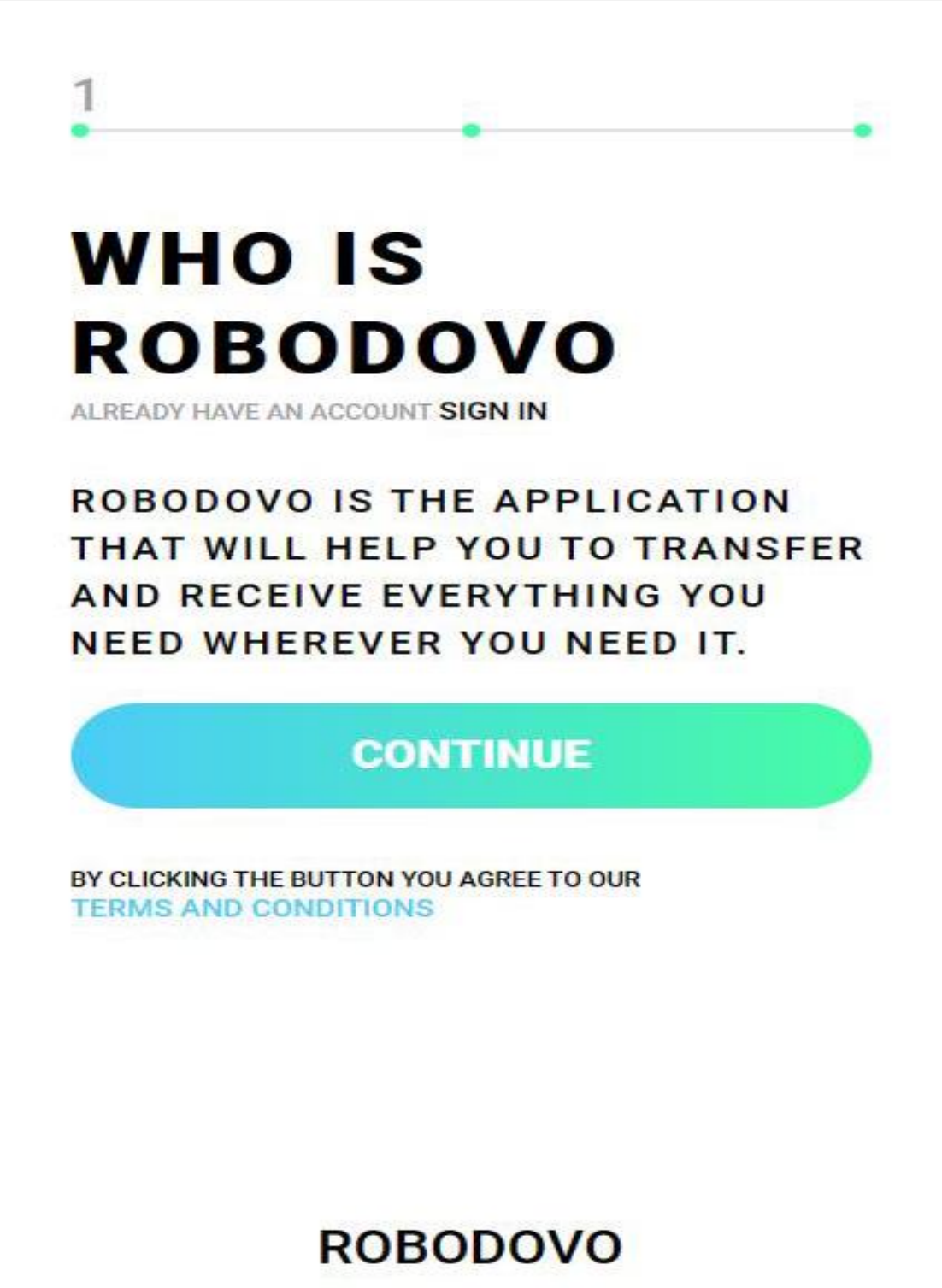
databases, making it an ideal choice for applications with limited resources or embedded systems.

Reliable and Stable: SQLite has a proven track record of reliability and stability. It is extensively tested, widely used, and has a large user community. It is considered a mature and robust database solution that can handle critical data management tasks effectively.

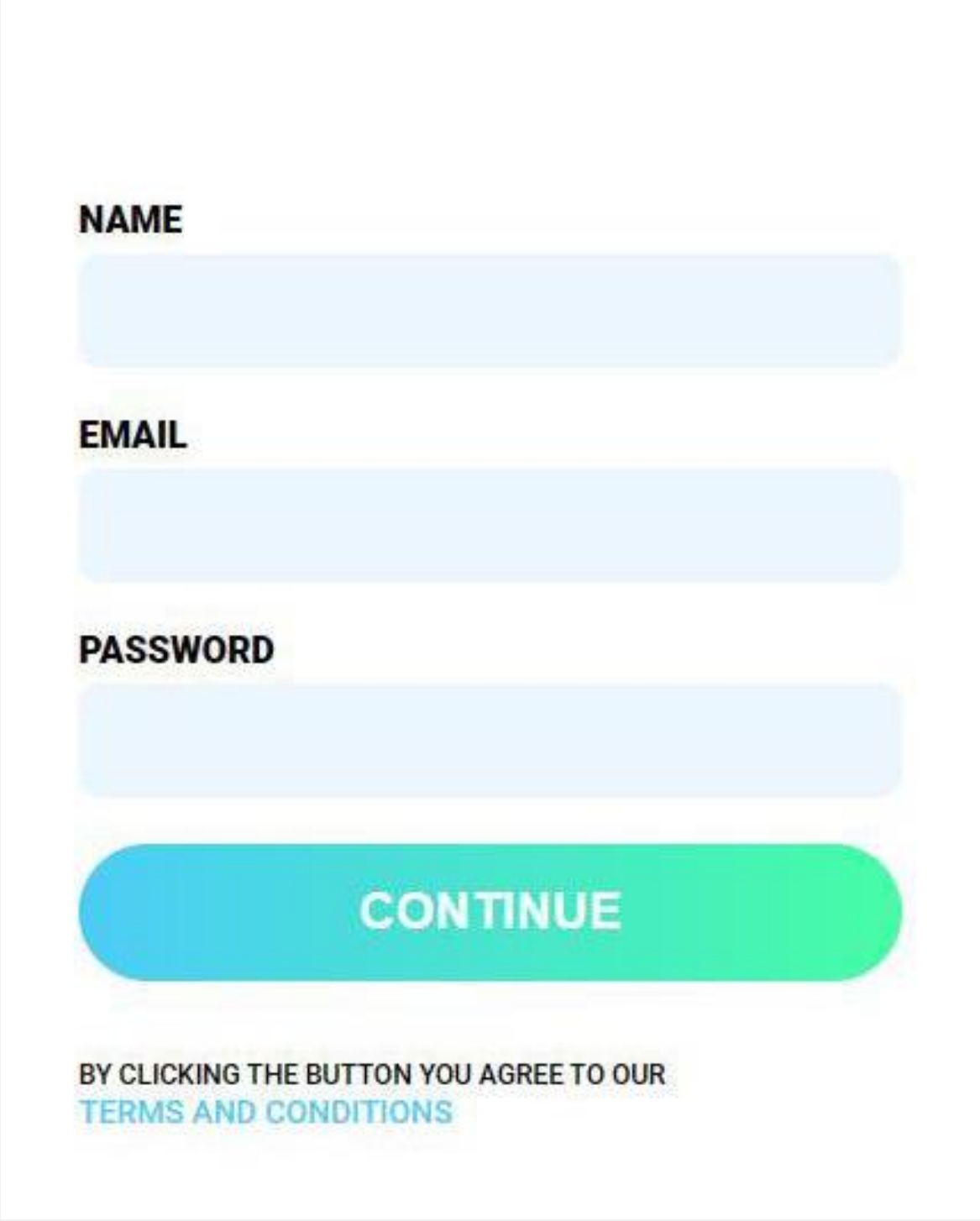
Overall, using SQLite for database management offers simplicity, portability, wide language support, performance, and reliability. It is particularly suitable for lightweight applications, mobile apps, embedded systems, and scenarios where easy integration and low resource consumption are essential.

5.4. How to use Application

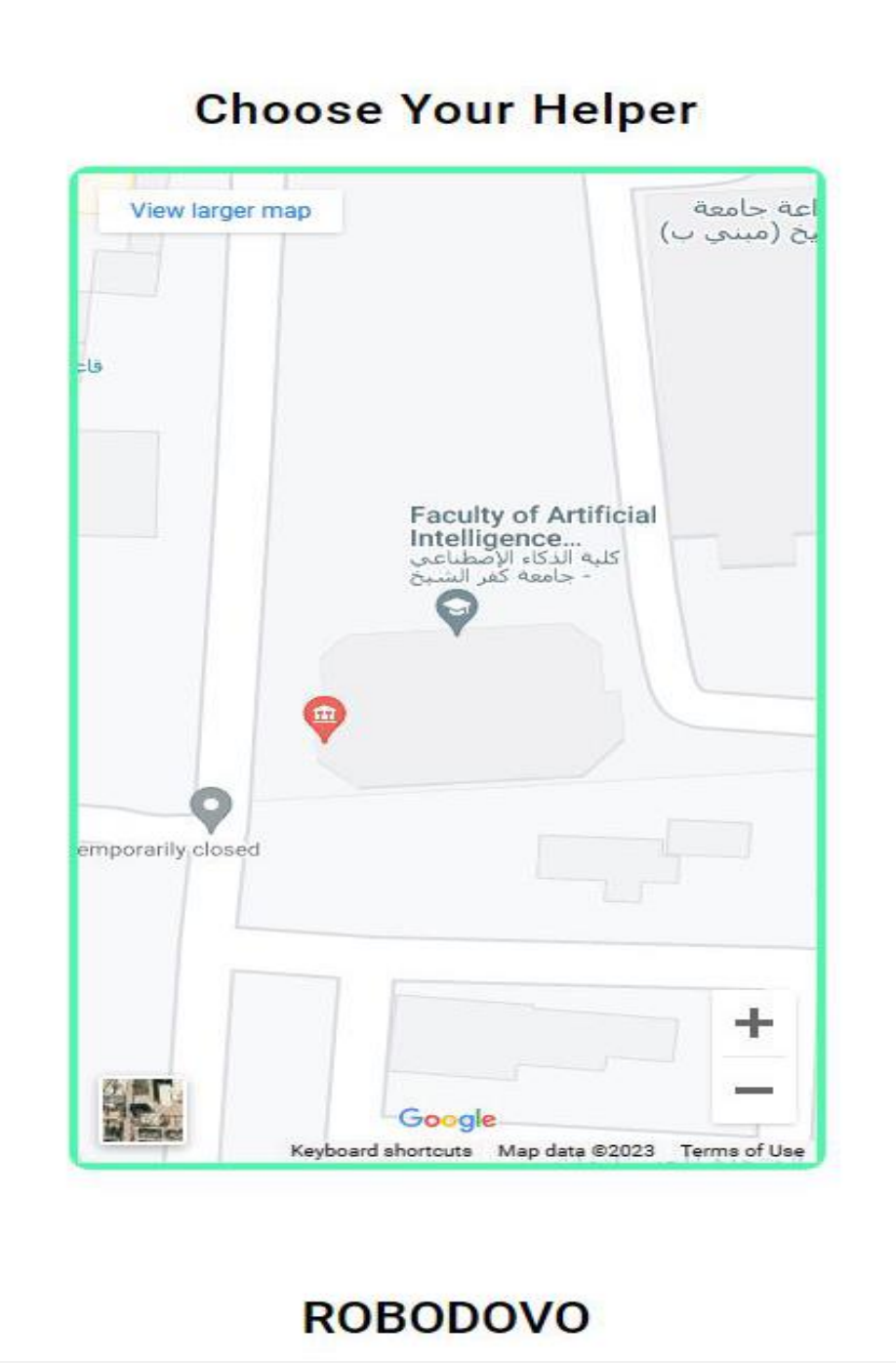
- Here is the first page short tip about robodovo robot.



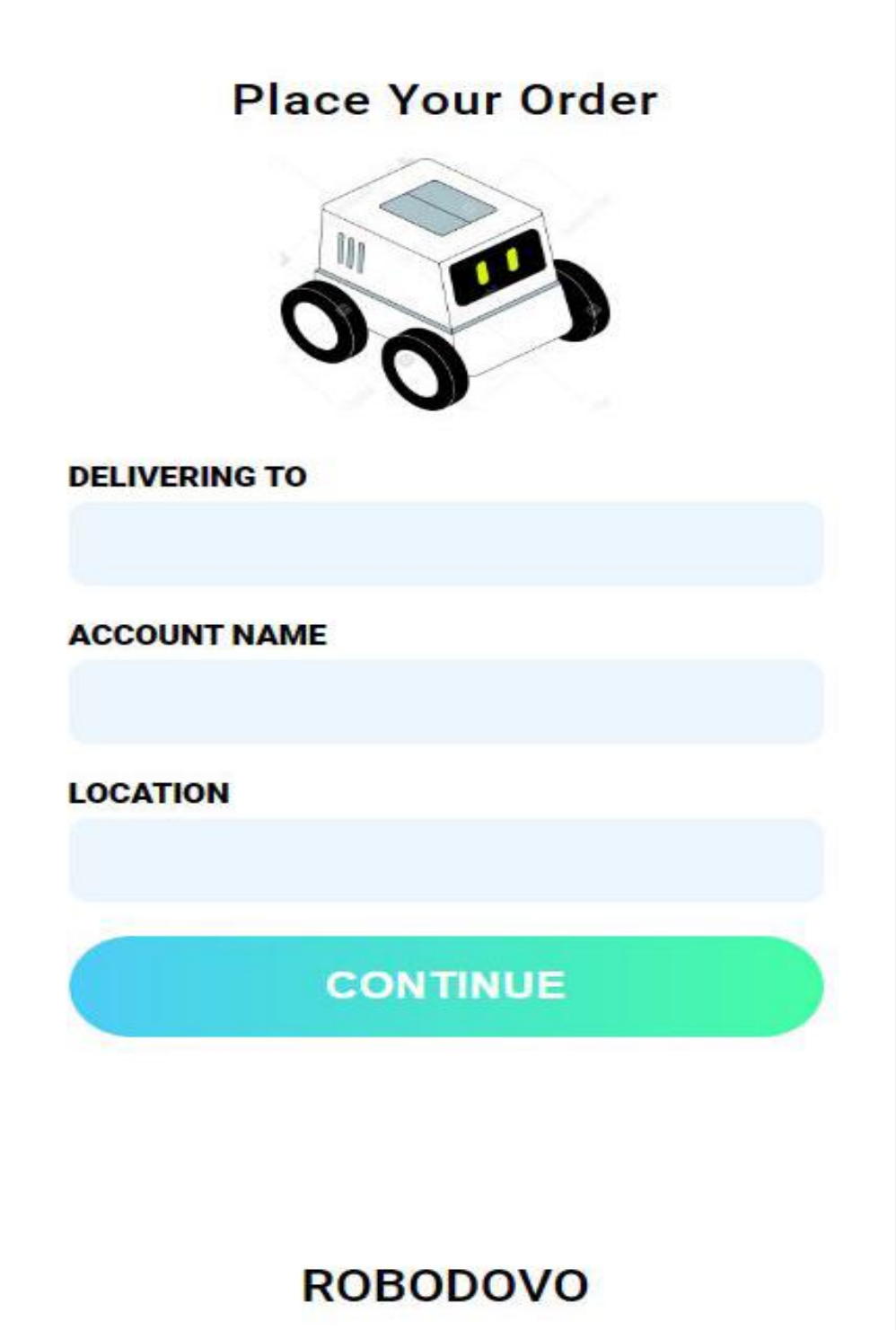
- In second page you should enter your name and account:



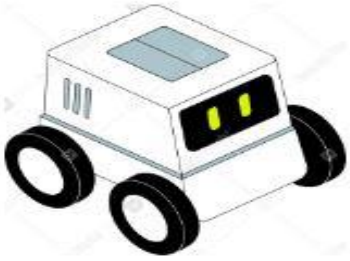
- In third page from the maps, you can choose the nearest robot:



- In fourth page you should enter location, account name and the person you want to send to him.



- Finally confirm your order



Confirm Your Order

CONTINUE

ROBODOVO

Flow Chart:

Start

├─ Collect input for destination points (A and B) from user

├─ Send movement request from Django web application to ROS

├─ ROS Bridge receives the movement request

├─ ROS system processes the request

│ ├─ Retrieve the lidar map

│ ├─ Plan a path from point A to point B

│ ├─ Send movement instructions to the robot

│ ├─ Robot starts moving

│ └─ Robot continuously updates its location using SLAM Monte Carlo Localization

├─ Robot performs autonomous navigation

├─ Robot reaches destination B

├─ Robot stops moving

├─ Update robot status in Django web application

└─ End

6. Case Studies and Applications

There are several companies actively working on the development and deployment of delivery robots. Here we will conclude some of them and a description of their robot.

6.1. Examples of Companies

- **Starship Technologies:**

Starship Technologies is a leading company specializing in autonomous delivery robots. They have developed small, wheeled robots that are designed to navigate sidewalks and deliver packages, groceries, and food. Starship robots have been deployed in various cities worldwide, partnering with companies such as FedEx, Just Eat, and Co-op.

- **Nuro:**

Nuro is focused on developing self-driving delivery vehicles for local goods transportation. Their vehicles are designed to operate in urban environments, delivering items from local businesses directly to customers' doorsteps. Nuro has partnerships with major retailers such as Kroger and Domino's Pizza to test and deploy their autonomous delivery vehicles.

- **Kiwibot:**

Kiwibot has developed a delivery robot designed to operate on sidewalks and deliver food and other items. These robots are equipped with secure compartments and can be remotely monitored and controlled. Kiwibot has collaborated with

numerous restaurants and businesses to provide autonomous delivery services in various cities.

- **Amazon Scout:**

Amazon Scout is an autonomous delivery robot developed by Amazon. These robots are designed to travel on sidewalks and deliver packages to customers' homes. Amazon has been testing Scout in select neighborhoods and expanding its deployment to enhance their last-mile delivery capabilities.

- **Marble:**

Marble focuses on creating autonomous ground delivery robots for the food and logistics industries. Their robots are designed to navigate urban environments and handle deliveries efficiently. Marble has partnered with food delivery services and local businesses to conduct pilot programs in cities like San Francisco.

- **Robby Technologies:**

Robby Technologies specializes in the development of autonomous delivery robots for the food and beverage industry. Their robots are designed to operate on sidewalks and deliver meals from restaurants to customers. Robby Technologies has partnered with various restaurant chains and delivery service providers to offer their robot delivery solutions.

- **Postmates Serve:**

Serve is an autonomous delivery robot developed by Postmates. It is designed to transport goods and food within neighborhoods. Postmates has been testing Serve in several cities, aiming to offer efficient and reliable on-demand delivery services.

These are just a few examples of companies actively working on delivery robots. The field of delivery robotics is rapidly evolving, and numerous startups and

established companies are continually innovating to improve the capabilities and deployment of these autonomous delivery systems.

6.2. Use Cases of Delivery Robot

Delivery robots have found a wide range of use cases across various industries, revolutionizing the way goods and services are transported and delivered. These autonomous machines are capable of navigating through different environments, avoiding obstacles, and securely delivering items to their intended destinations. Here are some notable use cases where delivery robots have made a significant impact:

1- Food and beverage delivery:

The delivery robot can be programmed to transport food and beverages between different buildings on campus, which can be especially useful for students and staff who have limited time for lunch or are unable to leave their work area.

2- Classroom delivery:

The delivery robot can be used to transport classroom supplies, such as textbooks or presentation materials, to different classrooms or lecture halls on campus.

3- Mail delivery:

The delivery robot can be used to transport mail and packages between different offices or departments on campus, which can help streamline the mail delivery process and reduce the need for human labor.

4- **Waste management:**

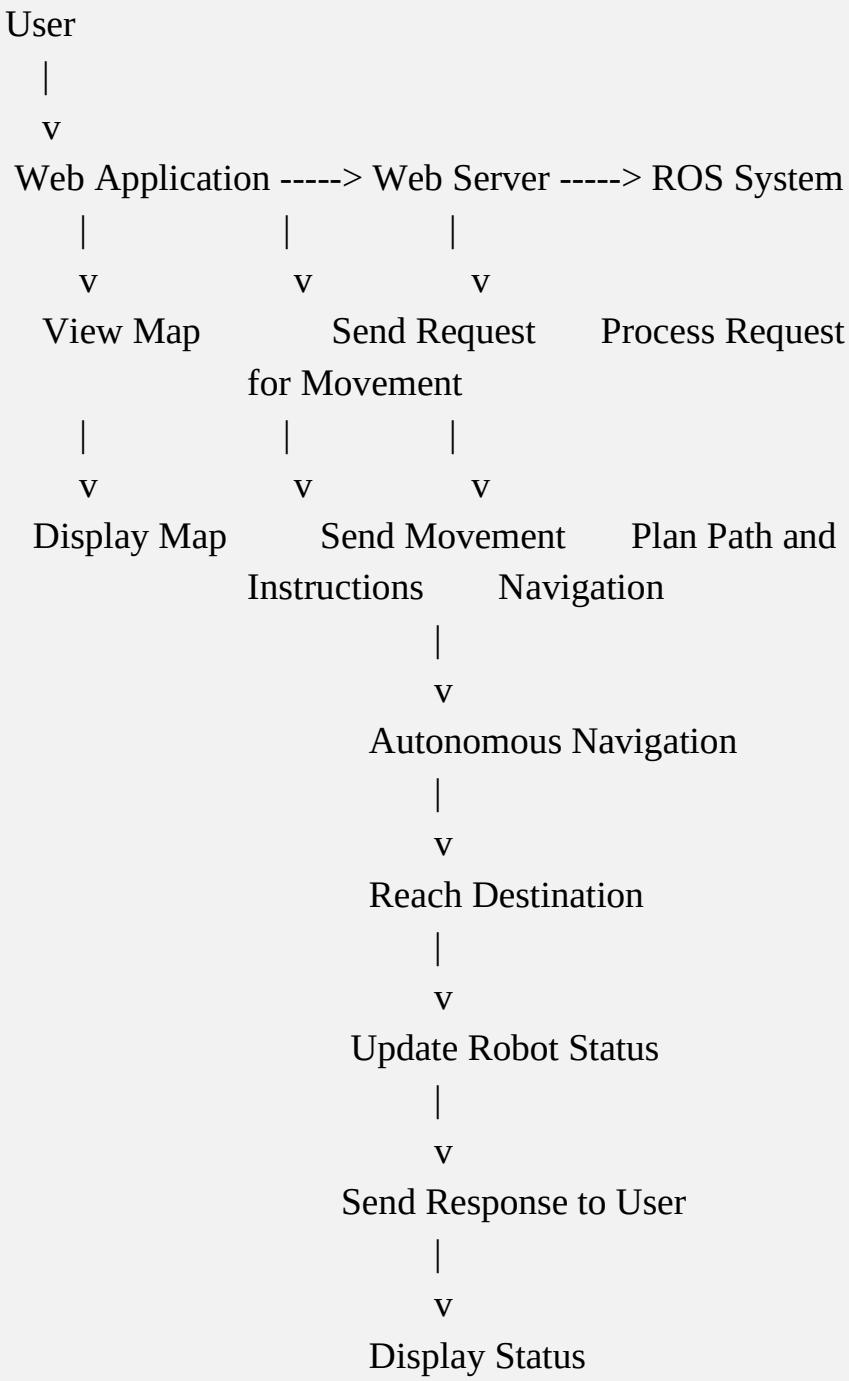
The delivery robot can be used to collect and transport waste or recycling materials between different areas on campus, reducing the need for human labor and promoting sustainable practices.

5- **Event support:**

The delivery robot can be used to transport equipment or supplies for events on campus, such as speakers or audiovisual equipment, making event setup and takedown more efficient.

These are so far the use cases of a delivery robot in our university. We believe that this project will add such valuable experience to our university considering it is the first university to add Artificial Intelligence studies to its educational degrees. We see our university as one of the pioneers in technologies and Artificial Intelligence techniques in Egypt, and this will only start by applying projects like ours on the university grounds.

Flow Chart:


User
Web Application -----> Web Server -----> ROS System
View Map
Send Request for Movement
Process Request
Display Map
Send Movement Instructions
Plan Path and Navigation
Autonomous Navigation
Reach Destination
Update Robot Status
Send Response to User
Display Status

7. Conclusion

In conclusion, the implementation of a delivery robot project offers significant benefits and opportunities for businesses and society alike. Delivery robots have the potential to revolutionize the way goods and services are transported and delivered, bringing efficiency, cost savings, and improved customer experiences.

By employing delivery robots, businesses can enhance their delivery operations, optimizing routes, reducing delivery times, and increasing overall efficiency. This efficiency leads to cost savings by reducing labor expenses associated with human delivery drivers and improving resource utilization. The scalability and flexibility of delivery robots allow businesses to adapt to changing delivery demands and expand their operations as needed.

Moreover, delivery robots promote contactless and hygienic delivery, addressing the growing demand for safety and reducing the risk of virus transmission. They provide a convenient and reliable solution, ensuring that customers receive their orders promptly and securely.

From an environmental standpoint, delivery robots contribute to sustainability efforts by reducing carbon emissions. Their electric power source and optimized delivery routes help minimize the carbon footprint associated with traditional delivery vehicles. By adopting delivery robots, businesses can demonstrate their commitment to environmental responsibility and contribute to a greener future.

The implementation of a delivery robot project also showcases a business's commitment to innovation and staying ahead in the market. By embracing cutting-edge technologies, businesses can differentiate themselves, attract tech-savvy

customers, and position themselves as industry leaders in adopting innovative solutions.

While the benefits of delivery robots are significant, it is crucial to consider factors such as infrastructure readiness, regulatory compliance, and public acceptance. Overcoming these challenges through collaboration between businesses, government entities, and the public will be essential for the successful implementation and widespread adoption of delivery robot projects.

In conclusion, delivery robots offer a promising future for the logistics and delivery industry, providing efficiency, cost savings, improved customer experiences, sustainability, and innovation. Embracing this technology can transform the way goods and services are delivered, contributing to the growth and development of businesses while enhancing the overall well-being of society.

8. References

1. Jocher, G. (2021, October 13). YOLOv5 v6.0 is here - new Nano model at 1666 FPS. Roboflow Blog. Retrieved from https://blog.roboflow.com/yolov5-improvements-and-evaluation/
2. Miller, B. (2021, February 22). Keyboards and autonomous cars. Medium. Retrieved from https://medium.com/predict/keyboards-and-autonomous-cars-d878ea93d7ba
3. Gupta, S. (2020, August 26). How to build a Django web app from scratch tutorial. Analytics Vidhya. Retrieved from https://medium.com/analytics-vidhya/how-to-build-a-django-web-app-from-scratch-tutorial-20034f0a3043
4. Addison, A. (2021, July 21). How to build an indoor map using ROS and lidar-based SLAM. Automatic Addison. Retrieved from https://automaticaddison.com/how-to-build-an-indoor-map-using-ros-and-lidar-based-slam/
5. Shahizat, M. (2021, November 29). Lidar integration with ROS Noetic and Ubuntu on Raspberry Pi. Hackster.io. Retrieved from https://www.hackster.io/shahizat/lidar-integration-with-ros-noetic-and-ubuntu-on-raspberry-pi-6ac3f7
6. Zhang, S., Zhang, Y., Huang, J., Wang, S., & Chen, X. (2021, March 17). Joint spatial-temporal attention for 3d object detection in lidar point clouds. arXiv preprint arXiv:2103.09229.Retrived from https://arxiv.org/abs/2103.09229
7. Makarov, A. (2020, December 16). The story behind the creation of Yandex's delivery robot. Yandex Self-Driving Car. Retrieved from https://medium.com/yandex-self-driving-car/the-story-behind-the-creation-of-yandexs-delivery-robot-e07017940589

8. https://www.tutorialspoint.com/artificial_intelligence/artificial_intelligence_robotics.htm
9. https://www.intellspot.com/artificial-intelligence-robots/
10. https://www.gideon.ai/resources/use-case-scenarios-for-autonomous-mobile-robots/
11. https://howtorobot.com/expert-insight/delivery-robots
12. https://tomorrow.city/a/shopping-with-delivery-robots-arrives-in-cities