Short Circuit: A Visual Approach to Autonomous Racing

Team Short Circuit—Rahul Gondi, Tirthankar Mittra, Paul Motter, Ryan Slocum, Mangala Sneha, and Matt Ward

***Abstract*— We present a 1/10th scale autonomous vehicle capable of navigating hallways and fulfilling a variety of challenges. Utilizing onboard compute, an RGB-D camera, a small infrared distance sensor, and an inertial measurement unit (IMU), we have used existing tools and have written new code to maximize the performance of this vehicle in the navigation of a closed track. In addition to these technical challenges, we have done research on the importance of human-vehicle interaction with advancements in autonomous vehicle technology, as well as the performance benefits available through deep learning in developing similar perception and control algorithms.**

I. Introduction

Autonomous vehicles are currently an extraordinarily popular field of work and research in the robotics community with teams of electrical, mechanical, computer, and systems engineers racing to develop the first viable solutions for autonomy on our roads. Due to the challenging nature of the problems faced in the pursuit of true autonomy, as well as the unpredictability of the environments humans typically drive on, there is still much progress to be made.

This project has sought to tackle some of those same challenges on a scaled-down level in an effort to understand the magnitude of the problem. We have built an autonomous system based on a 1/10th scale vehicle that exhibits some of the qualities needed in real-world autonomous vehicles, including obstacle detection and avoidance and state estimation. In addition to the physical scale of the vehicle being drastically different from the requirements of the real world, the environment the vehicle navigates is also far simpler, with no traffic signals, road markings, pedestrians, cyclists, or even other vehicles.

Through this project, we chose to tackle the following challenges: colliding with an obstacle and recovering, estimating the coefficient of friction between the rear wheels and the ground in real-time, and stopping at a stop sign on the course. These challenges all have applications in the real world (although recovering from a collision by immediately reversing is not directly applicable to or desirable for passenger vehicles). By estimating the coefficient of friction during vehicle transit, a control algorithm could make better decisions on how to navigate the terrain and whether it is currently safe to drive. The application of stop sign detection is clear and could be generalized to include the myriad traffic signs on our roads today.

In pursuit of these objectives, we encountered many challenges and quickly realized the necessity of creating robust algorithms for realistic driving situations. Even in our relatively constrained environment, there were many factors that could throw our autonomous vehicle off course and potentially into a violent collision. Of course, for a full-size autonomous vehicle driving at high speeds on the road, there are millions of situations the controller needs to be prepared for.

In working on this project, we took inspiration from other similar projects that had solved these types of challenges in a more robust manner, as discussed below in Section II. We then discuss the methods by which we arrived at our vehicle's current level of performance in Section III. Next, we analyze the results of the racetrack and the related challenges in Section IV. In Section V, we conclude with learnings from this project, and following that are the appendices on the importance of Human-Robot Interaction with autonomous vehicles and how we could have used Deep Learning to improve our performance.

II. Related Work

Currently, autonomous vehicle industries rely on a variety of sensors to achieve autonomy. The significance of these sensors lies in their ability to help autonomous vehicles perceive their surroundings, make informed decisions, and safely navigate roads. LiDAR, radar, cameras, GPS, and inertial sensors are among the most commonly used sensors in autonomous vehicles. Each type of sensor has its own unique advantages and disadvantages [1]. The Intel RealSense camera is a combination of sensors, including a stereoscopic pair of infrared cameras, an infrared laser projector, and an RGB camera. These cameras can be used in a variety of robotic applications, including object recognition, obstacle avoidance, indoor navigation, and 3D reconstruction. They produce accurate and reliable depth data and compare favorably to other commercially available depth sensors [2].

Visual SLAM is used in robotics to create a map of an unknown environment while simultaneously tracking the robot's location within that environment. [3] reviews and compares various approaches to SLAM and discusses their advantages and limitations. The use of stereo cameras and IMU data fusion and how they positively impact the performance of Visual SLAM is also discussed.

Motion planning is a fundamental problem in autonomous mobile robots that have been tackled by various algorithms. One widely used approach is the sensor-based method, which has demonstrated success in achieving accurate path tracking and obstacle avoidance. For example, [4] proposes a sensor-based method for motion planning of nonholonomic robotic vehicles. The authors use sensors like Lidar and cameras

to track the desired path and avoid obstacles. Another algorithm, presented in [5], employs sensory data to generate safe paths for robots in dynamic environments. The algorithm assigns costs to feasible paths based on factors like distance, clearance, and smoothness. These examples illustrate the importance of sensor-based methods in motion planning and their potential to improve the efficiency and safety of robotics applications. Our approach is to use sensor measurements and the actual dimensions of the course to determine the motion of the vehicle.

However reliable a sensor is, there is always imprecision in measurements to account for. To maintain good movement of the wall following robot, there is a need for a controller. Some of the most commonly used control algorithms are Fuzzy Logic [6] [7], Genetic Algorithm, Neural Network [8], Model Predictive Control [9], PID [10] [11], and a hybrid of these [12] [13] [14]. There are multiple papers focusing on the improvement of the tuning of PID control using algorithms [15] [16] [17]. Our approach is to use a PID controller as well to ensure the movement of the robot is good.

An IMU is used to measure the orientation, position, and motion of an object in 3D space, and it has a wide range of applications. In [18], it is used in conjunction with a 2D laser scanner and a stereo vision system to create a detailed map of the environment, including both geometric and semantic information. [19] uses GPS data to estimate the vehicle's speed and heading, and the magnetometer data(from IMU) to estimate the vehicle's sideslip angle. Using the estimated sideslip angle and other sensor measurements, a model is developed to estimate the tire-road friction coefficient, and [20] proposes an approach that utilizes the IMU data to estimate the vehicle's body sideslip angle and attitude, and the global navigation satellite system data to improve the estimation accuracy. Adaptive Kalman filters are used to estimate the vehicle's sideslip angle and attitude. Our approach is to read accelerometer measurements from the IMU and, through a rough idea of the coefficient of friction of concrete [21], get an estimate of the coefficient of friction of any surface.

III. Methodology

A. Hardware

At the beginning of this challenge, we were provided with the following items: an ODROID XU-4 as our compute, an Intel RealSense D435 as our primary method of perception, a $\frac{1}{10}$th scale Monster Truck chassis with a steering servo and brushed DC motor, a Mini Maestro Servo Controller, an IR Distance Sensor, Phidget 9-axis IMU, a DC buck converter, and a Lithium Polymer (LiPo) battery. While we opted not to use the IR distance sensor for most challenges due to its limited range, we used every other component and fabricated others to complete this car build.

The purpose of this autonomous vehicle is to compete in a timed lap of a rectangular section of corridor located inside the engineering center at the University of Colorado, Boulder. Therefore, the vehicle must be able to navigate such an environment with both speed and reliability while being able to withstand any collisions, accidental or otherwise.

We began by laser-cutting an acrylic plate that was mounted on top of the car, providing a platform to mount much of the hardware, including the RealSense camera near the front, the IMU near the center of gravity, and the ODROID and Mini Maestro towards the back of the vehicle. Our power electronics, including the LiPo battery and DC buck converter, were housed below the acrylic platform in order to maintain a lower center of mass for the car.

All of our electronics are powered by one LiPo battery. This battery is connected in parallel to the Electronic Speed Controller (ESC), which provides power to the brushed DC motor and the steering servo. It is also connected in parallel to the DC buck converter, which steps down the voltage from 10-15 volts to 5 volts, providing power to the ODROID and other low-voltage electronics. Initially, we had made the decision to separate the batteries after we experienced ODROID brown-outs (i.e. the ODROID computing unit did not have enough voltage to function as normal) following voltage sag in the batteries when the motors ran at high speed. However, upon further testing, we found that our batteries had not been not fully charged and that we could easily run the entire system on one LiPo battery.

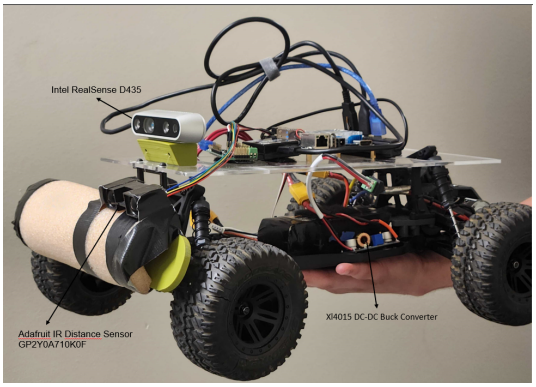


Fig. 1: A side view of our autonomous vehicle

The final build for our car has worked well and has proved durable against crashes during our testing phases. Separating the power electronics into two subsystems has provided extra resilience and allowed us to test the perception and controls separately at times.

B. Reactive Wall-Following Controller

Due to the simplistic nature of the course set out in this challenge, we developed an equally simple controller that allows the vehicle to centralize itself between walls and take corners at high speeds. In order to approach this problem, we needed to first get information from the environment relating to our position on the track. The Intel RealSense RGB-D camera mounted on the front of our vehicle provided a depth image of the space we aimed to navigate, so we primarily rely on this data for our state estimation and reactive control.

To handle noise in the depth data and simplify the logic necessary for our controller to navigate the course, we first

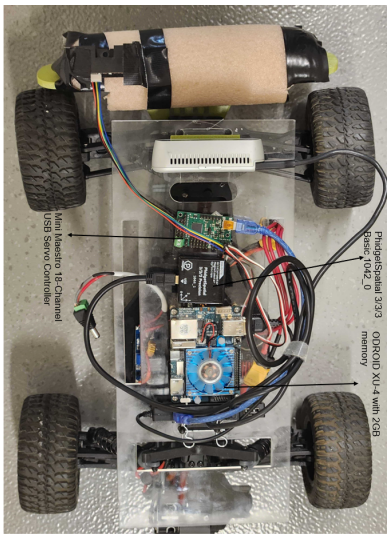


Fig. 2: Top view of our autonomous vehicle (the bumper has since been affixed more securely)

wrote a ROS node that provided distance estimates for the left, right, and center areas of the depth images. This was achieved using a large kernel for the center and two smaller kernels for the left and right sides of the image, over which the depth data was averaged to produce three estimates of distance as in 3. We found that the borders of the depth image were prone to noise and, as such, moved the "averaging regions" for the left and right depths away from the center of the depth image as a whole. After some testing, we settled on a width of 50 pixels for the left and right regions and a width of 100 pixels for the center region to perform this averaging calculation over the entire 848x480 pixel depth image. Our controller then subscribed to this depth data in order to inform its decisions throughout its time on the course.

In addition to using this distance data to inform the decisions of the main controller, we also subscribed to this data with a PID controller that adjusted the steering angle of the front wheels while the car was moving straight. The aim of this PID controller was to keep the vehicle centered in the hallway or just to the right of the center (following a semi-optimal racing line). We used the Simple PID library for Python for the controller implementation [22]. We adjusted the gains of this PID controller to avoid oscillatory behavior, and it was able to reliably keep the vehicle steering straight and avoiding walls on the straightaways of the course.

Our main controller works as a state machine and can initiate a variety of actions, including centering the vehicle straight, turning right, and stopping when an obstacle gets too close to the front bumper. It subscribes to the aforementioned distance information, as well as the PID gain for steering on long straightaways. The controller updates at 10 HZ, allowing it to make decisions based on new data from the

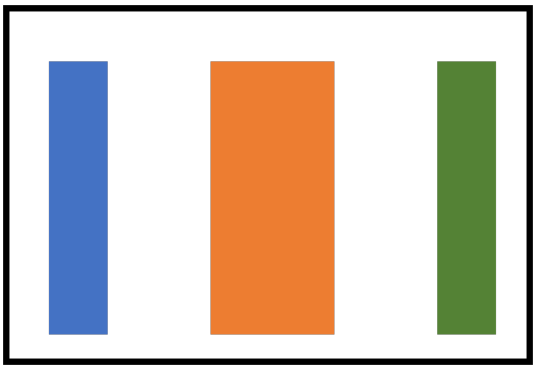


Fig. 3: An approximation of our depth calculation strategy. The entire picture represents the depth image exported by [23]. The left depth is extracted as an average over the blue region, the right over the green region, and the center over the orange region.

camera, which is running at 15 FPS. The main body of the controller is shown below in Algorithm 1: updateState().

Algorithm 1: `updateState()`

```
1  if centerDistance < stopDistance
2      stop();
3  else
4      if centerDistance > turnDistance
5          if centerDistance > slowDownDistance
6              straightFast();
7          else
8              straightSlow();
9      else
10         if (robot.isTurning and
             centerDistance > turnDistance) or
             rightDistance < turnAbortDistance
11             robot.isTurning = False;
12         else
13             turnRight();
14             robot.isTurning = True;
```

Fig. 4: A simplified representation of our state machine

When traversing a straight section of a corridor, the PID controller balances the left and right depth readings to maintain a position central to the width of the corridor as in figure 5, though we found that it was advantageous to bias the "center" of the corridor by scaling the left distance by 90 percent resulting in a vehicle trajectory that more closely followed the right wall of the corridor. While simply gauging the right distance and following the right wall directly would have been a simpler implementation, we observed that incorporating the left distance enabled the vehicle to more accurately center itself when the corridor width was irregular, as in figure 6.

As the racetrack consists of a rough rectangle traversed clockwise, our autonomous vehicle must detect an impending rightwards right angle turn and respond accordingly. This is accomplished by referencing the sensed center depth against a threshold $turnDistance$ and instigating a sharp right turn (full lock, or as sharp as possible) when the center depth falls below that threshold as in line 10 in algorithm 1. The

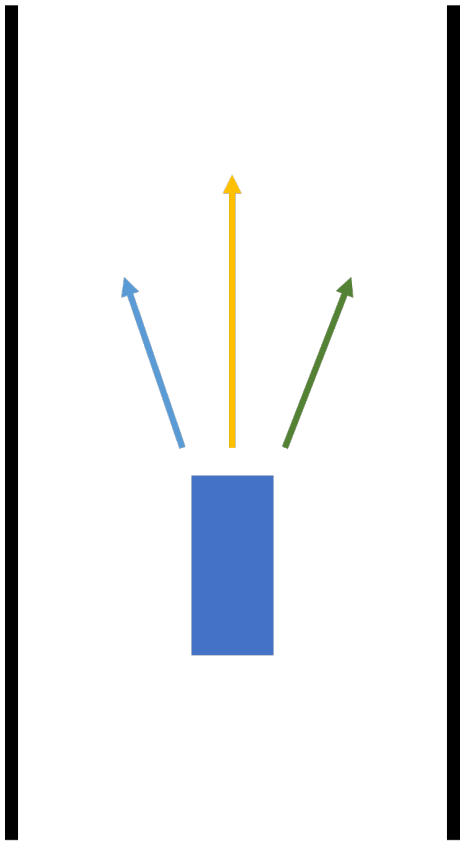


Fig. 5: An approximation of vehicle travel down a straight corridor. The vehicle (represented by the blue box) is balanced in the center of the corridor via PID control referenced by the left (blue) and right (green) distances, maintaining an approximately regular trajectory down the center of the corridor, represented by the orange arrow.

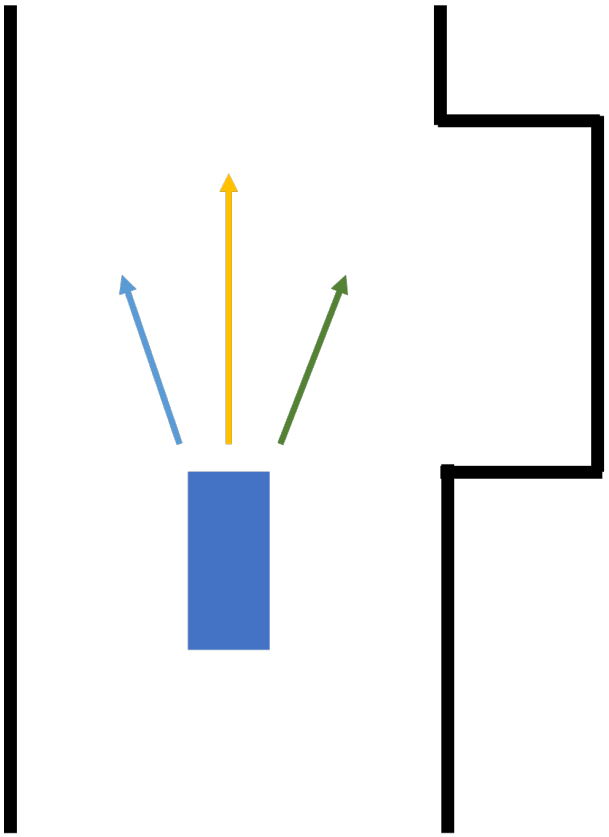


Fig. 6: As in figure 5, the vehicle must be able to continue down the corridor when the boundaries of such are irregular, such as to accommodate a doorway to an adjacent room.

vehicle then returns to corridor-centering PID control when the center distance exceeds the $turnDistance$ threshold after a short duration of full lock leftward correctional turning input as in figure 7. We found that the right turn duration was often longer than desired and that the leftward correction after the completion of the rightward turn enabled the vehicle to resume corridor centering more accurately after a turn.

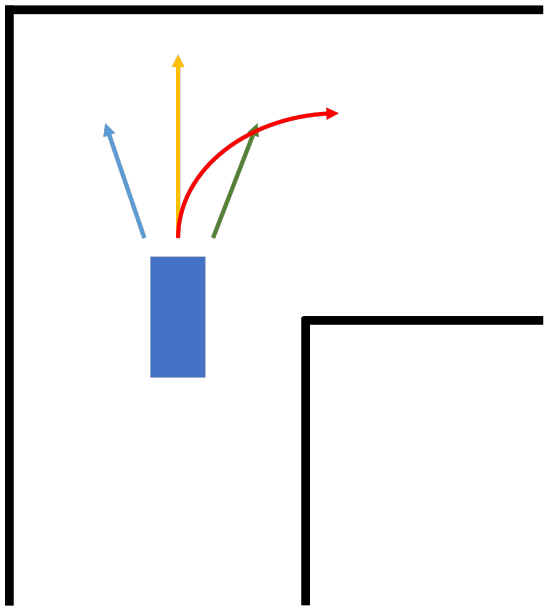
However, the PID controller is not flawless in centering the autonomous vehicle in the corridor. Sometimes, as the vehicle drifts to the right-hand boundary of the corridor, the center depth is sensed as below the $turnDistance$ threshold. This would lead to an unintentional right turn in the vehicle where there is not a turn in the corridor race course itself, followed swiftly by a sharp collision with the right-hand corridor boundary, potentially resulting in damage to the vehicle and certainly resulting in a failure of the vehicle to complete the lap. Therefore, if a turn is initiated and the vehicle immediately senses that the sensed right distance is less than a threshold $turnAbortDistance$, the turn is aborted, and the full-lock correctional turn is applied immediately (see line 10 in algorithm 1) as in figure 8. Anecdotally, we found that this logic is very effective in allowing the vehicle to dodge obstacles in its path.

For each action, we need to set the corresponding joints to the correct position or speed. In this case, we only have two degrees of freedom: the front steering servo motor and the rear brushed DC motor. Each of the actions taken by the controller sets these joints using a ROS node [24] written for the Pololu Mini Maestro [25]. For example, the stop action sets the servo to straight and then stops the motor, the action that steers the robot straight slowly sets the motor to a slow speed and sets the servo to the steering angle provided by the PID controller for wall-following.

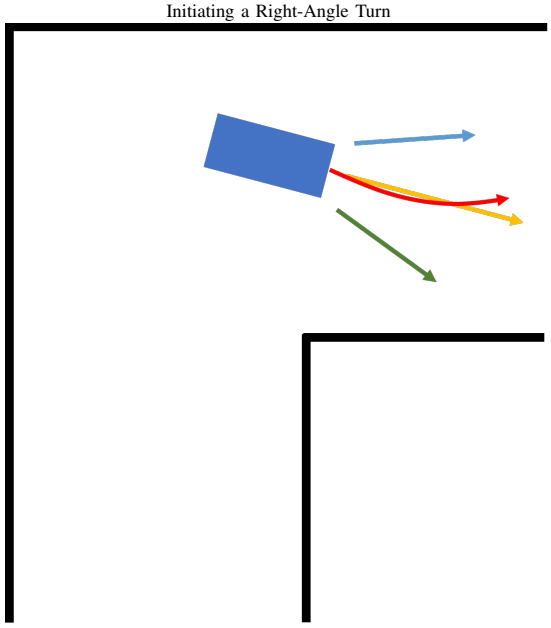
This controller took a great amount of tuning to correctly navigate the course and is not entirely robust. However, given such simple control logic, it is extremely fast and responsive, especially when compared to other algorithms we explored implementing for our relatively limited computational resources, such as SLAM for navigation.

C. Collision Management

The most interesting challenge to our team was that of collision management, or being able to back out of a collision when one occurred. The desired behavior would be to cause a collision with an obstacle or pedestrian, then to have the car autonomously back out of the collision and wait for the obstacle to be removed from the course. Such a collision is instigated by an object suddenly placed in the vehicle's path. In pursuit of this behavior, we primarily utilized the Intel RealSense camera's depth data, as well as the provided Infrared distance sensor. Because the behavior is relatively simple, we opted to use an equally simple state machine to dictate when the car was to reverse and wait for the obstacle



Initiating a Right-Angle Turn



Finishing a Right-Angle Turn

Fig. 7: **Top Image:** When the center distance reading (orange arrow) is below the $turnDistance$ threshold, the vehicle (blue box) initiates a full-lock (i.e. as sharp as possible) right turn. **Bottom Image:** After the center distance exceeds $turnDistance$ following a right turn, a correctional left turn is applied.

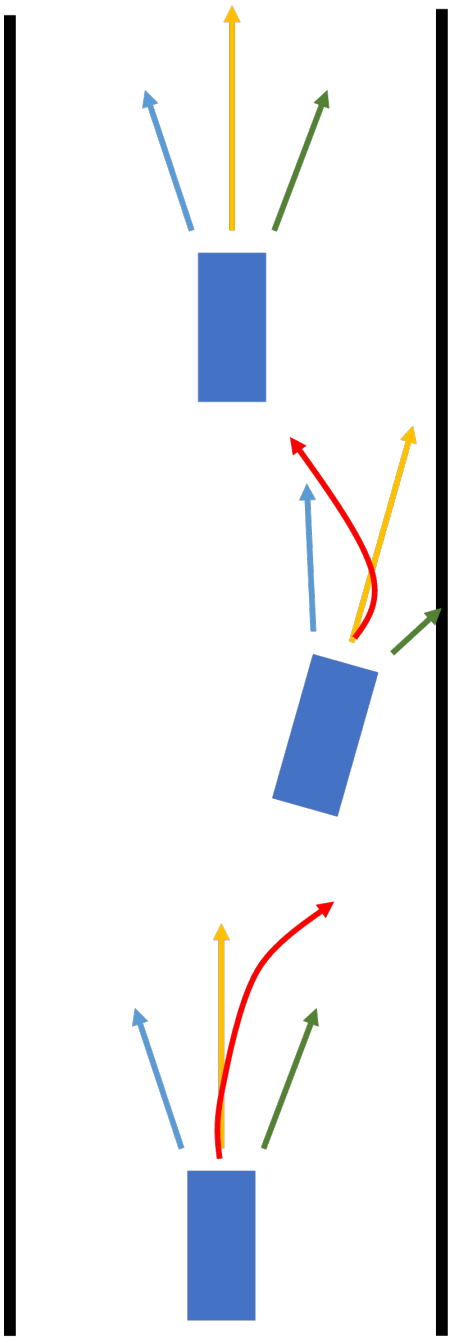


Fig. 8: As the robot travels through the corridor from bottom to top: first, an unintentional right turn is initiated as imperfections in the PID controller orient the vehicle towards the right-hand wall, causing the sensed center distance to stray below $turnDistance$. Second, the vehicle senses that as the sensed right distance is less than $turnAbortDistance$, an unintentional right turn is in progress and immediately initiates a sharp left turn. Third and finally, the vehicle returns to corridor centering state, line 4 in algorithm 1.

to move, as outlined in algorithm 2.

We found that for distances less than about 30 centimeters, the depths returned by the RealSense camera were extremely unreliable. The vehicle would routinely stop about one foot before the obstacle rather than colliding with it. Though this is usually a more desirable outcome than an outright collision, we were challenged with recovering after a collision had occurred. Therefore, once the depth was below the stop threshold as in algorithm 1 line 1, algorithm 2 is applied using only the depths returned by the depth sensor. First, the vehicle barrels at full speed into the obstacle, see 1. Then, once the collision has occurred, the vehicle reverses until it is at a certain distance away from the obstacle; see line 3. After this reverse maneuver, the vehicle waits patiently for the object to be removed in line 5, whence the vehicle exits the collision remediation state algorithm 2 returning to the state machine described in algorithm 2.

Algorithm 2: `onCollision()`

```
while infraredDistance > collisionDistance
    straightFast();
while infraredDistance < stopDistance/2
    straightReverse();
while centerDistance < stopDistance
    stop();
resumeNavigation();
```

D. Stop Sign Detection

In order to fulfill the challenge of stopping at stop signs, we once again used the RGB-D Intel RealSense camera we mounted on our vehicle. This time writing a ROS node to convert the RealSense's RGB image into an OpenCV-compatible format, we used a pre-trained image classifier to accurately detect stop signs on the course and react to their presence in real-time [26].

We first wrote a ROS node that converted the Intel RealSense RGB image into a format that we could use OpenCV to parse easily. While we heavily relied on this article's guidance from Intel's documentation, we were able to display the image using the OpenCV python library [23]. From here, we wrote a ROS python node that used our pre-trained classifier to detect whether there was a stop sign in front of the vehicle or not. This took some tuning to correctly identify stop signs at the correct distance (about 2-3 meters to give the car enough time to stop at an appropriate distance).

Once we had reliable data on whether the vehicle detected a stop sign in its field of view, we were able to publish this information for access in a controller similar to the aforementioned reactive controller from subsection III-B. When a stop sign was detected at a reasonable distance, the controller would execute a control sequence telling it to immediately come to a stop. After a pause of several seconds (to allow for any miniature pedestrians to cross the track), the vehicle would resume its course, ignoring the stop sign until it had passed it.

While not especially sophisticated, we found this stop sign detector to be effective, as well as computationally inexpensive. It was reliable at detecting stop signs and stopping at an appropriate distance, and due to the fact that we were integrating this methodology with our prior reactive controller, we were even able to avoid the stop sign in most cases if it were in the path of the vehicle.

E. Real-time estimation of Coefficient of Friction

For our next challenge, we decided to attempt real-time estimation of the coefficient of friction drag between the vehicle's wheels and the race track. We used a simple model to estimate the coefficient of friction. Equation (1) represents the net acceleration on our car of mass **M**, where F_t is the force applied on our car due to the motor and μ is the coefficient of kinetic friction.

$$Ma = F_t - \mu Mg \tag{1}$$

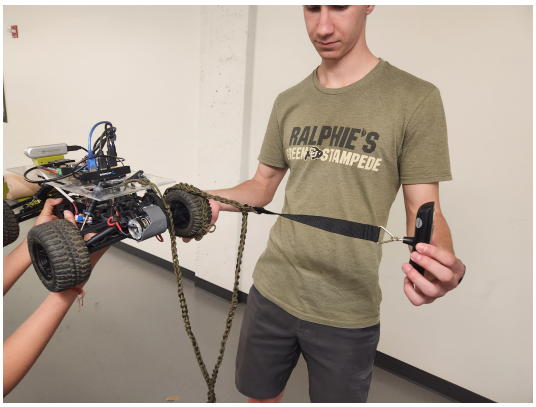


Fig. 9: A team member measures the force applied on a handheld luggage scale by a drive wheel via a cord while another team member secures the opposite drive wheel in a demonstration of why a limited-slip or locking differential is helpful in real-world automotive application.

It is difficult to directly measure the force applied by our car from the motor through the tires. We attempted to measure the wheel torque by wrapping a length of cord around one drive wheel and, while holding the opposite drive wheel stationary, measuring the force applied to the cord with a luggage scale (see figure 9). However, we decided that the potential inaccuracies inherent in this method—such as the losses incurred in the differential and the rest of the driveline as well as the mass of the cord—were not worth the potential gain in accuracy. Instead, we indirectly calculate it by running our car on a surface with a known coefficient of friction. In our case, we used rubber on concrete, which has a coefficient of friction of approximately 0.6 [27]. The actual acceleration of our car is calculated with an IMU mounted on our autonomous vehicle. Since our autonomous vehicle operates on flat terrain, we can calculate the net acceleration on the car from our IMU acceleration readings in the x and y directions.

$$Ma_c = F_t - \mu_c Mg \tag{2}$$

$$Ma_x = F_t - \mu_x Mg \tag{3}$$

Equation (2) and (3) are the two equations of motion, subscript **c** is for the concrete surface (known) and subscript **x** is for the unknown surface, after canceling out F_t from the two equations we get a simplified Equation (4) for calculating the coefficient of friction.

$$\mu_x = \frac{a_c - a_x}{g} + \mu_c \tag{4}$$

We are calculating the coefficient of friction drag online while our car is operating on an unknown surface. To account for noisy results due to slipping, battery power, etc our code maintains a sorted list of 100 recent coefficient of friction calculations and publishes the median value to the appropriate ROS topic. The brushed DC motor runs in two modes: fast and slow. In both these modes, the force F_t applied on the car is different so we have different measurements of acceleration a_c in equation (3), the online calculation of μ_x is done separately for these two modes, but both these modes will ideally produce the same results for the estimates of μ_x.

F. Other attempted challenges

In addition to the aforementioned challenges, as a team, we tried several others. In an effort to further explore the lessons learned from this project, we will outline the attempts we took to make each of these other challenges work in this section.

At the outset of the project, Visual-Inertial SLAM seemed like a reasonable challenge, especially considering the extensive documentation existing on implementing this method with an RGB-D camera such as the RealSense we were provided. However, we ran into a few main problems when attempting this challenge. Firstly, we realized that the lack of reliable odometry data would be problematic in the sensor-fusion step of algorithms such as EKF or UKF SLAM. Without wheel odometry, it would be far harder to reliably estimate the pose of the vehicle. This would have been manageable using landmark detection techniques for localization; however, in the basement track where we were doing our testing and planning to race, the choice of landmarks on the walls and floor was sparse. The walls were entirely blank, and due to the position of the camera near the ground necessitated by the low profile of our autonomous vehicle, much of the field of view of the camera consisted of the floor. In addition to these difficulties, we quickly realized that the ODROID did not provide us with sufficient computing to handle the operations necessary for this real-time localization and mapping.

Once we realized the infeasibility of navigation by a SLAM technique on the hardware supplied from our attempt at Visual-Inertial SLAM, we decided we would try to create a sparse map of our surroundings during a trial run using a Grid-Based FastSLAM library. There were two appealing options: the Hector Mapping ROS package [28], and the Gmapping ROS package [29]. While both of these options addressed our concerns regarding limited compute and lack of visual landmarks, we ran into other difficulties, mostly related to scan-matching and odometry. With both libraries, we had a hard time getting the scans to match from frame to frame as the algorithms lost their notion of keypoints, meaning we could not effectively build a map of the course over time. We believe this was due to noise from RealSense's depth sensor or possibly from excess shaking or bobbing due to the vehicle's movement. In addition to this problem, we still had no effective way to estimate the vehicle's linear velocity, even after trying to address this with various ROS localization libraries meant to filter and fuse IMU data into accurate odometry estimations.

Finally, we had minor successes in getting the vehicle to drift around corners when taking turns. We achieved this by varying the PWM sent to the motors when the vehicle was turning right around the track. By sending the wheels into reverse, we were able to turn with a turning radius of almost zero, making it easier to take turns with a very tight radius. However, we decided the better strategy was to set the wheels to max forward speed for a brief moment when beginning the turn, allowing the wheels to slide out behind the car while not reducing the speed of the vehicle at the same time. We, unfortunately, found that these turns were extremely hard to control reliably (especially on the surfaces of varying grip in the basement we were testing in), and decided that it would be easier to complete the course reliably and quickly if we were to take turns at a speed where we did not lose traction. We realize that the incorporation of sensor readings from the IMU may have aided our completion of this challenge. Given our experiences and difficulties in using IMU readings for SLAM techniques, we decided to attempt other challenges.

As a footnote, we would like to address a challenge that seemed popular with other teams: attempting a jump. This challenge seemed relatively straightforward, as with proper tuning, a vehicle could simply be sent at the launch ramp with a predetermined speed corresponding to a static PWM value while the steering servo is set to dead center. While we found this opportunity tempting, we decided that the risks involved were too great. Our experience tuning the reactive wall-following controller in section III-B showed that various factors could influence a set drive motor PWM value, such as the voltage level remaining in our battery. We also feared the effects of weight imbalances incurred as the result of any additional accessories such as bumpers or sensors we deemed necessary for efficient and repeatable completion of other challenges or the complete failure of the battery attachment mechanism (a hook-and-loop fastener that has seen better days). After watching other teams fling their vehicles into oblivion resulting in broken front protective devices and unpredictable ODROID operation, we are overall satisfied with our decision.

IV. Results

When we tested our car with its reactive wall following algorithm on a representative race track, it was successfully able to make turns and complete the track in a reasonable time range from 40 seconds to one minute. Since the layout of the track was not uniform, we had to fine-tune many of the parameters like coefficients for our PID controller, the distance at which our car makes the turn, motor speed, etc. After parameter tuning, our car was able to complete a full race track with 70-75 percent accuracy; the indeterminacy was due to factors such as battery charge, other challenge participants and their vehicles active in the track and unreliability in the depth sensors. There is some anxiety among the team as to the performance of such a parameter-driven model during the true competition, though we are confident that we have exhausted the limits of our system given the constraints of the competition.

In completing the actual race on a much shorter track, our two successful runs were completed in 12.8 and 10.5 seconds, respectively. These times were below the average at the time and took minimal tuning of the controller to achieve after being moved to the new course.

Once applied as intended, we found that the stop sign detection model was very accurate. With improper parameters used in the detector, we even saw the vehicle detect stop signs as far as seven meters away from the front

of the vehicle. Once we correctly defined the parameter relating to the perceived dimensions of our example stop sign at a reasonable distance, as recorded by the RealSense RGB stream, we were able to reliably stop at a distance of approximately one meter ahead of the stop sign.

While we admit that the system and logic behind our collision mitigation scheme are overly complex and therefore prone to error, we were able to reliably instigate then back out of a collision with a container lid in pre-competition testing. We are excited to see how our autonomous vehicle performs on this challenge in competition on the as-yet-unknown race course.

As the surface of the race track was not uniform, the online coefficient of friction drag on the vehicle was calculated to be approximately between 0.63 to 0.7 during race simulations.

V. Conclusions

While we did not meet all of the objectives we initially set out to achieve during the course of the project, we are able to present a robotic system that successfully navigates a closed course, as well as completes several other adjacent challenges. We applied many of the concepts learned in this Advanced Robotics course, such as the applications of linear control systems and SLAM algorithms to real-world robotics technologies. We were also able to develop hands-on experience working with important tools in robotics software engineering, such as ROS visualization and debugging tools, as well as popular ROS packages.

Going into this project, much of the team did not have hands-on experience with real-world robotics. We learned that robots are prone to make the worst-possible action at the worst possible time, the joys of ceaseless parameter tuning, and the inevitable challenges inherent in a parameter-heavy system. We grew to appreciate the necessity of accurate and abundant sensing equipment. We found that even before a concept of robot operation can be introduced into practice that there is often more effort required to prepare both the hardware and the software stack in order to realize our vision.

At the conclusion of the course of our work, we wonder what would have been possible with different equipment. The ODROID XU-4, while handling our implementations with moderate effectiveness, proved to be inadequate for the task of many modern techniques for image processing and localization. While we were able to obtain workable data from the RealSense D435, we found the depth sensor to be prone to noise and inaccuracy. The competition that inspired ours, the F1Tenth challenge, recommends an NVIDIA Jetson for computing and a sophisticated LIDAR unit [30]. While we appreciate that the challenges on the F1Tenth circuit are in many ways more difficult than what we attempted and that such a setup is likely impractical to our setting, we imagine what we may have been able to accomplish with better equipment. At the same time, we also appreciate that in the elusive "real world" the best possible sensors and computation modules are not always available for every given task. Experience working at the edges of our equipment's ability affords us valuable perspective.

This project was an excellent exercise in gaining familiarity with the entire gamut of applications of real-world robotics, including hardware, controls, perception, and planning. During the course of our work, we were also forced to consider the triviality of what we were trying to accomplish in comparison to the robotics technologies that are being developed in industry and in academia today, as well as the enormity of those extremely difficult challenges. Overall, we feel that regardless of our performance, we learned valuable skills from the project that will help with future robotics challenges.

Appendix: Human Autonomous-Vehicle Interaction

Autonomous vehicles have progressed far beyond Stanley [31]. The automotive industry at large has worked towards marketing autonomous vehicles at varying levels to the general public, and one source estimates that in 2021 the autonomous personal vehicle market was worth 22.2 billion USD and that it will grow to 75.95 billion USD by 2027 [32]. The SAE has developed a taxonomy of autonomous vehicle ability, varying from no automation requiring total human control (level 0) to fully autonomous requiring no human control (level 5) [33] as described in figure 10. Currently, only one company—Mercedes Benz—has offered a vehicle for sale that has been certified to what the SAE considers autonomous driving, where a human must not maintain constant vigilance and be ready to take control at all times [34], though several concerns are attempting to develop fully autonomous taxis to varying levels of controversy and success [35]. Beyond autonomous personal vehicles, autonomous vehicles for construction [36] and food delivery [37] have also been developed. The following is a review of published work to advance these domains and how they interact with human operators, other humans in vehicles, and humans in their environment.

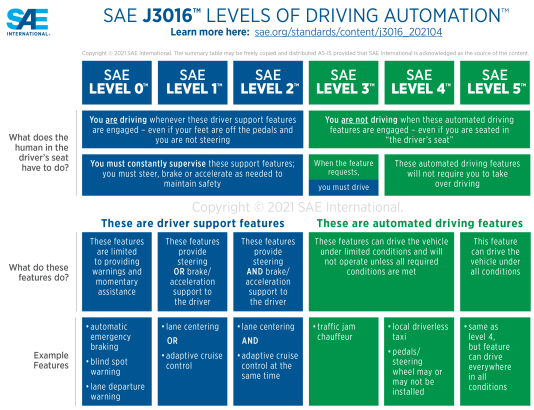


Fig. 10: SAE Autonomy Levels [38]

Up to SAE level three, a human must be in the driver's seat of an autonomous personal vehicle and ready to take control at all times. When a vehicle's system encounters a level of uncertainty, the human must be able to take over and avoid a potentially dangerous situation quickly and with a high level of effectiveness. Kim et al. performed a statistical analysis of four different scenarios at four levels of takeover request alerts and the time required by a human to take avoiding action in four different scenarios [39]. The authors also surveyed participants for qualitative evaluation of the takeover requests. The authors found that a lower time to takeover request was more effective than a faster and that a dynamic time to takeover request generated by driver behavior was more effective than a statically-defined value and that the study participants preferred the dynamic warning. However, what form should the alert take? Alerts are generally presented as visual, auditory, or physical (vibration) stimulus [40]. When takeover requests are audible, it has been found that speech-based alerts combined with electronic tones are more effective than electronic tones alone [41]. Olaverri-Monreal et al. discovered that visual alerts are more effective when the alert corresponds to the level of situation urgency and that visual alerts are more effective for drivers with high rates of smartphone usage [42]. At high levels of automation, Petersen et al. developed a system to increase a human monitor's situational awareness of the driving task that allowed the driver to be more productive in a secondary task [43].

Unfortunately, humans are often prone to attempting a secondary task in a vehicle with autonomous assistance features when their full attention is required for the driving task. There is some question on the part of researchers [44], [45], the industry [46], and other associations [47] on the effectiveness of autonomous vehicle assistance at less than level 4. There have been several high-profile incidents of operators of vehicles with autonomous assistance that require constant human attention becoming highly distracted instead of maintaining awareness of their vehicle's surroundings [48], [49], [50]. California has passed a law attempting to prevent names for autonomous assistance features that may influence a driver to put more trust than warranted in the system and allow himself to become distracted and unable to take control when necessary [51]. Zeeb et al. found that low levels of autonomy can actually lead to higher levels of driver distraction and unsafe action [45] which mirrors findings by AAA [47]. In the face of these shortcomings, several parties question the effectiveness of driver assistance and wonder whether higher levels of autonomy that still require human takeover (such as level 3) are of any value, leading to an increased focus on level 4 and level 5 autonomy [40].

It has long been recognized that in order for an autonomous vehicle to be effective on the road among vehicles operated by human input alone, it must operate at the same level as a vehicle driven by a human [52]. A study in Italy put vehicles with level 2 autonomy (lane-centering and follow distance control) to a Turing test with 550 university students on a loop of local roads. If a passenger could not determine whether the system had been engaged, the system passed the Turing test. The authors found that most systems were able to pass such a Turing test and that steering input was more likely to be detected as controlled by an autonomous system then speed control [53]. If a system acts in a way that is indistinguishable from a human driver from the point of view of a passenger, it is likely that other drivers on the road with an autonomous system will find the system's actions more predictable. To accurately generate such behavior for an autonomous vehicle, Zhang et al. developed a Bayesian game-theory approach that varies an autonomous vehicle's aggression (scaled from "polite" to "aggressive") based on the behaviors of other vehicles on the road. The authors also subjugated their system to a Turing test and found that most participants in the study could not differentiate between the behavior of their autonomous system or a human and

therefore found the system to be generally predictable [54].

When interacting with other human drivers, a human driver often expects informal visual cues from those other drivers. For example, if two drivers meet at a 4-way stop and there is confusion as to who has the right-of-way, one driver will often "wave" the other through the intersection. Similarly, one driver might flash his high beams or make a variety of potentially-potent hand gestures to signal displeasure with another road user. Drivers and even pedestrians rely on establishing eye contact with a human driver to establish that driver's acknowledgement of their presence. Human drivers are also extremely prone to error, and may for example neglect to activate turn signals even when appropriate. In such a case, another human driver in a vehicle behind the driver neglecting to signal is often able to anticipate the forward driver's intentions by observing a gradual shift in the lane towards the intended direction of turn. A standard vehicle with no modifications beyond those absolutely necessary for autonomy is both unable to send such informal signals and unable to interpret signals from human operators. There are certain situations in which simply following existing statutes of road behavior (ex. appropriately using turn signals) will not be sufficient for autonomous-human driver interaction, and an autonomous vehicle resembling those on the road would be unable to make hand signals [55]. Kent Larson's group at MIT developed a prototype vehicle equipped with mechanical "eyes" that swivel and illuminate to simulate eye contact and additional illumination to signal intent of motion [56]. A system applied to a semi truck that illuminates an LED light bar signaling intent to turn and following distance was entered into a competition for internal and external interfaces and won. The developers applied the "show, don't tell" principle of signaling intent to action without explanation of reasoning modeled after human-animal interaction [57]. More complicated displays that can present more information on an autonomous vehicle's intentions have also been developed [58].

Not much work has been published on recognizing gestures from nearby drivers, presumably because sensing a driver through the glass of the windscreen or oft-tinted windows is a difficult obstacle to overcome. Work *has* been published on systems for autonomous cars to recognize traffic control gestures based on a neural network [59]. Geng et al. applied a neural network to infrared image data to enable an autonomous vehicle recognize a multitude of gestures from any nearby humans. Importantly, a infrared camera allows an image processor to easily detect humans against the background [60].

Pedestrians, cyclists, motorcyclists, wheelchair-users, small electric scooter riders, skateboarders, and other road-users that exist outside the safety structures afforded by increasingly-large and heavy vehicles must be able to exist on roadways among autonomous vehicles. These non-car roadway denizens are less uniform than other vehicles and are often more erratic, often lacking basic indicators expected on larger vehicles such as turn signals and brake lights as well as often failing to adhere to road use codes. These difficulties presented to autonomous vehicles and strategies to overcome them have been cataloged by Reyes-Muñoz and Guerrero-Ibáñez [61]. Detection and classification strategies include neural-network approaches such as those proposed by [59], [60] Autonomous-vehicle-to-human interfaces are listed, including visual, acoustic, and anthropomorphic (such as the eyes from [56]) methods. Less-obviously, they also list vehicle-external interfaces for vulnerable road user detection, such as relative location via smartphones [62]. From a pedestrian's perspective, Gronier et al. have developed a questionnaire for pedestrians gauging their interactions with autonomous vehicles to be used by regulatory bodies when approving autonomous vehicle features [63].

An autonomous vehicle's interaction with humans is not limited to driving behavior. Beyond the split-second roadway decisions made to avoid catastrophe, an autonomous vehicle must also interact with humans around it when parked, rolling down a sidewalk, or in a more open environments such as in construction sites, in surveillance applications, and military applications. In the case of small last-mile delivery vehicles, there have been instances of vandalism [64]. For autonomous vehicles to be truly effective, they must be accepted by the general public. An acceptance model for last-mile delivery vehicles has been created to gauge public sentiment [65] and a small survey of public acceptance of last-mile delivery vehicles conducted in Germany found mostly ambivalence, though it is important to note that acceptance almost certainly varies culture-to-culture [66].

Adverse public reaction is not limited only to last-mile delivery autonomous vehicles. Moore et al. performed an interesting study where they conspicuously placed a vehicle made to look obviously autonomous on a loop of road in a public place and recorded interactions with passersby, then repeated this in three different countries. In each country, instances of individuals conspicuously studying the vehicle as well as people willfully obstructing or becoming clearly upset with the vehicle were recorded on camera, which the authors termed "greifing." Pedestrians loitered in front of or turned around and repositioned themselves in front of the vehicle, hurled verbal abuse at the vehicle, and drove erratically in front of the vehicle. A text message obtained from a pedestrian who walked erratically in front of the vehicle for some distance read that the vehicle "[did not] have the courage to destroy [him]." The authors suggest that an autonomous vehicle must have some level of aggression (in contrast to waiting indefinitely at a crosswalk) and must look as inconspicuous as possible to be truly effective and must actively deter "greifing" as a matter of occupant safety [67]. Researchers applying current autonomy methods to a construction site found that most methods for dealing with anomalous situations that arise when teaming with humans and human-operated machinery were not sufficient and that a tailor-made framework must be developed for each specific theatre as most development of such methods is conducted in heavily-controlled environments. They did not find any significant gaps in relation to specific human control of the various systems, however [68].

APPENDIX: DEEP LEARNING

Introduction

As autonomous vehicles proliferate on our roads, they depend on a variety of technology to run effectively and safely. One such technique that could greatly enhance the performance of autonomous vehicles is deep learning, a subset of machine learning. Deep learning algorithms can perform object detection extremely fast relative to other methods. A deep learning architecture, developed by 4 University of Washington alumni, by the name of "You Only Look Once" can process images at 155 frames per second [69]. Their model has outperformed other state-of-the-art detection systems while also proving to be highly accurate at recognizing distinct items, such as vehicles and pedestrians, which may assist the vehicle to avoid collisions. Not only are these deep learning models fast, but they can be effectively applied to a wide variety of situations. The main reason deep learning is being increasingly used in robotics is that it can enable robots to learn from experience and adapt to changing environments. Traditional robotics algorithms are often based on hand-crafted rules and assumptions about the environment, which may not be flexible enough to handle the complexity and variability of real-world scenarios. In contrast, deep learning algorithms can learn from large amounts of data and discover complex patterns and relationships in the data that may be difficult to capture with traditional approaches. One of the main advantages of using deep learning in robotics is that it can help robots to perceive and understand their environment better. For example, deep learning algorithms can be used for object recognition, segmentation, and tracking, which are critical tasks for robotic systems that need to interact with the world around them. Deep learning can also be used for sensor fusion, where data from multiple sensors is combined to provide a more comprehensive view of the environment. Another advantage of using deep learning in robotics is that it can help robots to plan and execute actions more effectively. Deep learning can be used for online path planning, where the robot plans its path in real-time based on its current perception of the environment. It can also be used for control, where the robot learns to perform complex tasks by learning from demonstration or reinforcement learning.

Image Recognition

The first of these applications is that of image recognition. Traffic signs, lane markings, pedestrians, and other aspects of the road can all be recognized and identified using deep learning models. For instance, deep learning-based methods have been used for real-time lane detection and tracking in autonomous vehicles as seen in a report from IEEE [70]. In the report, the authors go on to explain their real-time approach based on deep learning for ego-lane detection which achieves high accuracy and processing speed. The approach involves using a CNN model to learn features of input images and predict lane boundaries. It also uses a novel post-processing method to refine said predictions. Their approach has been shown to outperform several other state-of-the-art methods and has proven effective in a real-world scenario. Additionally, a method for traffic sign recognition in autonomous vehicles using deep learning has been shown to be highly effective [71]. Similar to the lane detection methods, this deep learning stop sign detection is able to outperform the competition. It does so through the use of multiple convolutional layers which extract relevant features from an input image and multiple fully connected layers that are able to classify a street sign from the features it extracts.

Path Planning

Along with image recognition, deep learning can also be useful in path planning by using the present state of the roads and the position of the vehicle. For example, a deep reinforcement learning-based method proposed for path planning in autonomous vehicles by researchers from the University of British Columbia proved that it could be quite useful for path planning [72]. Their model is trained on a large dataset of state-action pairs, where the states represent the current environment and the actions represent the steering and acceleration commands for the vehicle. Using said data, the model then uses its current state to predict an optimal path. We can see another example of deep learning used in the context of path planning in the article titled "Path planning of autonomous vehicles using deep learning" from IEEE's website [73]. This deep learning model uses a dataset of labeled images which represent the environment around the vehicle. It is then able to learn to predict a steering angle and a throttle command given the current image, which can then be used to generate a trajectory for the vehicle.

End-to-End Learning

End-to-end learning is yet another use for deep learning models and is a promising approach for autonomous driving which has the potential to revolutionize the field, allowing systems to learn complex tasks directly from raw inputs. One of such end-to-end learning frameworks for self-driving cars proposed in the article "End to end learning for self-driving cars" uses a convolutional neural network (CNN) to directly map raw pixel inputs to steering commands [74]. While traditional modular approaches that use hand-crafted features work well, the authors show that their approach was able to outperform them. The article titled "Deep learning for autonomous driving: A review" also discusses the use of end-to-end learning in the application of autonomous driving [75]. The authors of this report highlight its potential to improve performance and reduce development time. They also note the need for large amounts of data and the difficulty in interpreting the learned representations. This indicates that while end-to-end learning is a useful method for autonomous driving, it can also have its drawbacks.

Object Detection

Among the many ways that deep learning might enhance autonomous vehicle performance is through object detection. This use of machine learning might be considered to be

the most important by some from an ethical standpoint as deep learning models can be used to detect and identify many environmental items, including barriers, cars, and, most importantly, pedestrians. We can see an example of this in the article "A pedestrian detection method for autonomous vehicles based on deep learning" where the authors propose a pedestrian detection method which, much like many of the other aforementioned deep learning models, involves using a convolutional neural network to learn a mapping between input images and pedestrian locations [76]. That being said, the sheer quantity and variety of objects that one has to train models such as this one on are innumerable due to the different sizes and shapes of people, barricades, and other cars. Even so, the useability of deep learning object detection models has proven useful and even essential in the development of autonomous vehicles.

Vehicle Detection

Vehicle detection is another important task for autonomous vehicles. There is a proposed deep convolutional neural network for vehicle detection in unmanned aerial vehicle imagery [77]. Their model was highly accurate in recognizing various vehicle categories, such as trucks, buses, and cars. A multi-task deep learning approach has been used for vehicle detection in aerial images [78]. They created a novel model that could accurately identify different vehicle kinds while concurrently detecting cars and estimating their orientations. A vehicle detection system is developed based on convolutional neural networks [79]. Their program was able to recognize vehicles in real-time and with excellent accuracy even in challenging scenarios like congested traffic and obstruction. Pedestrian Detection: Pedestrian detection is critical for ensuring the safety of autonomous vehicles. Shi et al. proposed a pedestrian detection algorithm based on deep learning [80]. They implemented a brand-new feature pyramid network in their model, which performed at the cutting edge on numerous benchmarks for pedestrian identification. S. Huang et al. developed a deep learning-based pedestrian detection method for autonomous driving [81]. Their model could detect pedestrians in real-time with high accuracy, even in challenging scenarios such as occlusion and lighting variations. Zhang et al. used transfer learning for real-time pedestrian detection in autonomous vehicles [82]. A huge dataset for pedestrian identification was used to fine-tune a pre-trained convolutional neural network, and they attained high accuracy in practical situations.

Traffic Sign Detection

Traffic sign detection is another important task for autonomous vehicles. A method is developed for traffic sign detection and recognition system based on deep learning [83]. In order to detect and recognize traffic signs in actual situations, their model made use of a multi-scale convolutional neural network. Control Systems: Finally, deep learning can be used to control the systems of autonomous vehicles, such as steering, braking, and acceleration. For instance, an end-to-end deep learning approach was used for steering angle prediction in autonomous vehicles [74]. By learning to regulate the steering angle directly from camera views, the model was able to steer with a high degree of accuracy. Similarly, a method proposed for autonomous driving using deep learning, which achieved high accuracy in steering and speed control [75].

Conclusion

With the ability to precisely detect their surroundings, make defensible decisions, and navigate safely and effectively, deep learning has the potential to greatly enhance the performance of autonomous vehicles. Deep learning has the potential to improve autonomous vehicle functionality and hasten the advancement of this technology through object identification, picture recognition, path planning, and control systems. However, there are also some challenges and limitations associated with using deep learning in robotics, which may explain why it is not used as extensively as in some other domains such as computer vision or natural language processing. One of the challenges of using deep learning in robotics is the need for large amounts of training data, which can be difficult to obtain in some real-world robotics applications. Another challenge is that the use of deep learning models can result in a lack of interpretability or transparency, which can be problematic in situations where it is important to understand how a robot arrived at a particular decision or action. Robotic systems typically require real-time and on-board processing, which can be challenging with deep learning algorithms that are often computationally intensive and may require significant computing resources, plus many robotic systems use online algorithms like RRT* for path planning to account for dynamically changing environment whereas deep learning algorithms like (deep Q learning) DQN will most likely fail when it encounters a situation or training example it has not seen before. Additionally, the stability and safety of a robotic system can be crucial, and the use of deep learning algorithms may introduce additional sources of uncertainty and risk, which need to be carefully managed. Despite these challenges, deep learning is being increasingly used in robotics, and researchers are developing new methods to address these issues and improve the performance, reliability, and safety of deep learning-based robotic systems.

REFERENCES

[1] H. A. Ignatious, Hesham-El-Sayed, and M. Khan, "An overview of sensors in autonomous vehicles," *Procedia Computer Science*, vol. 198, pp. 736–741, 2022, 12th International Conference on Emerging Ubiquitous Systems and Pervasive Networks / 11th International Conference on Current and Future Trends of Information and Communication Technologies in Healthcare. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921025540

[2] V. Tadic, Á. Odry, I. Kecskes, E. Burkus, Z. Király, and P. Odry, "Application of Intel RealSense Cameras for Depth Image Generation in Robotics," *WSEAS Transactions on Computers*, vol. 18, pp. 107–112, 2019.

[3] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual slam algorithms: a survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, p. 16, 2017. [Online]. Available: https://doi.org/10.1186/s41074-017-0027-2

[4] D.-H. Kim, C.-S. Han, and J. Y. Lee, "Sensor-based motion planning for path tracking and obstacle avoidance of robotic vehicles with nonholonomic constraints," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 227, no. 1, pp. 178–191, 2013. [Online]. Available: https://doi.org/10.1177/0954406212446900

[5] I. K. E. Rivlin, "Sensory-based motion planning with global proofs," *IEEE Transactions on Robotics and Automation*, vol. 13, pp. 814 – 822, 1997. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/650160/authors#authors

[6] I. Emmanuel, "Fuzzy logic-based control for autonomous vehicle: A survey," *I.J. Education and Management Engineering*, vol. 2, pp. 41–49, 2016. [Online]. Available: https://www.mecs-press.org/ijeme/ijeme-v7-n2/IJEME-V7-N2-5.pdf

[7] N. P. Varma, V. Aivek, and V. R. Pandi, "Intelligent wall following control of differential drive mobile robot along with target tracking and obstacle avoidance," in *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, July 2017, pp. 85–91.

[8] I. Hammad, K. El-Sankary, and J. Gu, "A comparative study on machine learning algorithms for the control of a wall following robot," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec 2019, pp. 2995–3000.

[9] T. M. Howard, C. J. Green, and A. Kelly, "Receding horizon model-predictive control for mobile robot navigation of intricate paths," in *Field and Service Robotics*, A. Howard, K. Iagnemma, and A. Kelly, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 69–78.

[10] K.-J. Joo, S.-H. Bae, A. Ghosh, H.-J. Park, and T.-Y. Kuc, "Wall following navigation algorithm for a disinfecting robot," in *2022 19th International Conference on Ubiquitous Robots (UR)*, July 2022, pp. 343–346.

[11] M. H. Iqbal and W. S. Aji, "Wall following control system with pid control and ultrasonic sensor for krai 2018 robot," *International Journal of Robotics and Control Systems*, vol. 1, no. 1, pp. 1–14, 2021.

[12] C.-F. Juang and C.-H. Hsu, "Reinforcement ant optimized fuzzy controller for mobile-robot wall-following control," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 10, pp. 3931–3940, Oct 2009.

[13] C.-Y. Chou and C.-F. Juang, "Navigation of an autonomous wheeled robot in unknown environments based on evolutionary fuzzy control," *Inventions*, vol. 3, no. 1, 2018. [Online]. Available: https://www.mdpi.com/2411-5134/3/1/3

[14] R. Braunstingl, J. Mujika, and J. Uribe, "A wall following robot with a fuzzy logic controller optimized by a genetic algorithm," in *Proceedings of 1995 IEEE International Conference on Fuzzy Systems.*, vol. 5, March 1995, pp. 77–82 vol.5.

[15] H. Suwoyo, Y. Tian, C. Deng, and A. Adriansyah, "Improving a wall-following robot performance with a pid-genetic algorithm controller," in *2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Oct 2018, pp. 314–318.

[16] A. Adriansyah, H. Suwoyo, Y. Tian, and C. Deng, "Improving the wall-following robot performance using pid-pso controller," *Jurnal Teknologi*, vol. 81, no. 3, 2019.

[17] H. Suwoyo and F. Harris Kristanto, "Performance of a wall-following robot controlled by a pid-ba using bat algorithm approach," *International Journal of Engineering Continuity*, vol. 1, no. 1, pp. 56–71, Dec. 2022. [Online]. Available: https://ejournal.sultanpublisher.com/index.php/ijec/article/view/39

[18] L. Iocchi and S. Pellegrini, "Building 3d maps with semantic elements integrating 2d laser, stereo vision and imu on a mobile robot," *2nd ISPRS International Workshop*, 2007. [Online]. Available: https://www.isprs.org/proceedings/xxxvi/5-W47/pdf/iocchi_pellegrini.pdf

[19] J.-H. Yoon, S. Eben Li, and C. Ahn, "Estimation of vehicle sideslip angle and tire-road friction coefficient based on magnetometer with gps," *International Journal of Automotive Technology*, vol. 17, no. 3, pp. 427–435, 2016. [Online]. Available: https://doi.org/10.1007/s12239-016-0044-7

[20] L. Xiong, Y. Lu, W. Liu, L. Gao, S. Song, and Z. Yu, "Imu-based automated vehicle body sideslip angle and attitude estimation aided by gnss using parallel adaptive kalman filters," *IEEE Transactions on Vehicular Technology*, vol. PP, pp. 1–1, 03 2020.

[21] B. G. Rabbat and H. G. Russell, "Friction coefficient of steel on concrete or grout," *Journal of Structural Engineering*, vol. 111, no. 3, pp. 505–515, 1985. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%290733-9445%281985%29111%3A3%28505%29

[22] M. Lundberg, "Simple-pid," 2023. [Online]. Available: https://github.com/m-lundberg/simple-pid

[23] Intel® RealSense™, "IntelRealSense/librealsense: Intel® RealSense™ SDK." [Online]. Available: https://github.com/IntelRealSense/librealsense/tree/master

[24] Kalvik, "Pololu_maestro_ros," 2022. [Online]. Available: https://github.com/itskalvik/pololu_maestro_ros

[25] "Pololu - Mini Maestro 18-Channel USB Servo Controller (Assembled)." [Online]. Available: https://www.pololu.com/product/1354

[26] K. K. Rai, "Personal-Python-Projects/Stop Sign Detection at master · kishanrajput23/Personal-Python-Projects." [Online]. Available: https://github.com/kishanrajput23/Personal-Python-Projects

[27] E. Fleming, "Coefficients of friction between rubber tires and concrete road surfaces," in *Proceedings, Highway Research Board*, vol. 14, no. Part I, 1934, p. 214.

[28] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," 2011.

[29] G. Grisetti, C. Stachniss, and W. Burgard, "Gmapping." [Online]. Available: https://openslam-org.github.io/gmapping.html

[30] "Bill of materials," https://f1tenth.org/build. [Online]. Available: https://f1tenth.org/build

[31] S. Thrun, "A personal account of the development of stanley, the robot that won the DARPA grand challenge," *AI Magazine*, vol. 27, no. 4, pp. 69–69, 2006, number: 4. [Online]. Available: https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1910

[32] Autonomous car market size & share analysis - industry research report - growth trends. [Online]. Available: https://www.mordorintelligence.com/industry-reports/autonomous-driverless-cars-market-potential-estimation

[33] (2021) J3016_202104: Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles - SAE international. [Online]. Available: https://www.sae.org/standards/content/j3016_202104/

[34] (2023) Mercedes-benz world's first automotive company to certify SAE level 3 system for u.s. market. [Online]. Available: http://media.mbusa.com/releases/mercedes-benz-worlds-first-automotive-company-to-certify-sae-level-3-system-for-us-market

[35] San Francisco County Transport Autority, "Re: Protest of cruise llc tier 2 advice letter (0002)," Jan 2023.

[36] (2021) Volvo and holcim jointly work in a project to use autonomous electric haulers. [Online]. Available: https://www.volvoautonomoussolutions.com/en-en/news/press-releases/2021/nov/volvo-and-holcim-jointly-work-in-a-project-to-use-autonomous-electric-haulers.html

[37] Starship - ROBOTS: Your guide to the world of robotics. [Online]. Available: https://robots.ieee.org/robots/starship/

[38] (2021) SAE levels of driving automation™ refined for clarity and international audience. [Online]. Available: https://www.sae.org/site/blog/sae-j3016-update

[39] H. J. Kim and J. H. Yang, "Takeover requests in simulated partially autonomous vehicles considering human factors," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 5, pp. 735–740, 2017.

[40] W. Morales-Alvarez, O. Sipele, R. Léberon, H. H. Tadjine, and C. Olaverri-Monreal, "Automated driving: A literature review of the take over request in conditional automation," *Electronics*,

vol. 9, no. 12, p. 2087, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/12/2087

[41] Y. Forster, F. Naujoks, A. Neukum, and L. Huestegge, "Driver compliance to take-over requests with different auditory outputs in conditional automation," *Accident Analysis & Prevention*, vol. 109, pp. 18–28, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457517303408

[42] C. Olaverri-Monreal, S. Kumar, and A. Dìaz-Àlvarez, "Automated driving: Interactive automation control system to enhance situational awareness in conditional automation," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 1698–1703, ISSN: 1931-0587.

[43] L. Petersen, L. Robert, X. J. Yang, and D. Tilbury, "Situational awareness, driver's trust in automated driving systems and secondary task performance," *SAE International Journal of Connected and Automated Vehicles*, vol. 2, no. 2, pp. 129–141, 2019. [Online]. Available: https://www.sae.org/content/12-02-02-0009/

[44] J. R. Clark, N. A. Stanton, and K. M. A. Revell, "Automated vehicle handover interface design: Focus groups with learner, intermediate and advanced drivers," *Automotive Innovation*, vol. 3, no. 1, pp. 14–29, 2020. [Online]. Available: https://doi.org/10.1007/s42154-019-00085-x

[45] K. Zeeb, A. Buchner, and M. Schrauf, "Is take-over time all that matters? the impact of visual-cognitive load on driver take-over quality after conditionally automated driving," *Accident Analysis & Prevention*, vol. 92, pp. 230–239, 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457516301075

[46] D. Paresh, "Google ditched autopilot driving feature after test user napped behind wheel," *Reuters*, 2017. [Online]. Available: https://www.reuters.com/article/us-alphabet-autos-self-driving-idUSKBN1D00MD

[47] E. Edmonds. (2020) AAA finds active driving assistance systems do less to assist drivers and more to interfere. [Online]. Available: https://newsroom.aaa.com/2020/08/aaa-finds-active-driving-assistance-systems-do-less-to-assist-drivers-and-more-to-interfere/

[48] T. Fitzsimons. (2021) Tesla driver slept as car was going over 80 mph on autopilot, wisconsin officials say. NBC News. [Online]. Available: https://www.nbcnews.com/news/us-news/tesla-driver-slept-car-was-going-over-80-mph-autopilot-n1267805

[49] The Associated Press, "A tesla driver is charged in a crash involving autopilot that killed 2 people," *NPR*, 2022. [Online]. Available: https://www.npr.org/2022/01/18/1073857310/tesla-autopilot-crash-charges

[50] S. Davidson. (2022) Tesla driver appears to be asleep going more than 100 km/h on busy ontario highway. CTV News Toronto. Section: Toronto. [Online]. Available: https://toronto.ctvnews.ca/tesla-driver-appears-to-be-asleep-going-more-than-100-km-h-on-busy-ontario-highway-1.6053530

[51] A. Ohnsman. (2023) Tesla calling its cars 'full self-driving' may run afoul of new california law. Forbes. Section: Transportation. [Online]. Available: https://www.forbes.com/sites/alanohnsman/2023/01/04/tesla-calling-its-cars-full-self-driving-may-run-afoul-of-new-california-law/

[52] T. B. Sheridan and W. R. Ferrell, *Man-machine systems; Information, control, and decision models of human performance*, ser. Man-machine systems; Information, control, and decision models of human performance. The MIT Press, 1974, pages: ix, 452.

[53] E. Cascetta, A. Cartenì, and L. Di Francesco, "Do autonomous vehicles drive like humans? a turing approach and an application to SAE automation level 2 cars," *Transportation Research Part C: Emerging Technologies*, vol. 134, p. 103499, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X2100485X

[54] Y. Zhang, P. Hang, C. Huang, and C. Lv, "Human-like interactive behavior generation for autonomous vehicles: A bayesian game-theoretic approach with turing test," *Advanced Intelligent Systems*, vol. 4, no. 5, p. 2100211, 2022. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/aisy.202100211

[55] B. Färber, "Communication and communication problems between autonomous vehicles and human drivers," in *Autonomous Driving: Technical, Legal and Social Aspects*, M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, Eds. Springer, 2016, pp. 125–144. [Online]. Available: https://doi.org/10.1007/978-3-662-48847-8_7

[56] K. Bullis. (2012) How do you know an autonomous vehicle has seen you? MIT Technology Review. [Online]. Available: https://www.technologyreview.com/2012/04/26/256038/how-do-you-know-an-autonomous-vehicle-has-seen-you/

[57] O. Benderius, C. Berger, and V. Malmsten Lundgren, "The best rated human–machine interface design for autonomous vehicles in the 2016 grand cooperative driving challenge," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1302–1307, 2018.

[58] K. Buchholz. (2018) Lights communicate hella's autonomous vehicle messages. SAE International. Last Modified: 2018-02-07T16:26:44-05:00. [Online]. Available: https://www.sae.org/site/news/2018/01/lights-communicate-hellas-autonomous-vehicle-messages

[59] J. Wiederer, A. Bouazizi, U. Kressel, and V. Belagiannis, "Traffic control gesture recognition for autonomous vehicles," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 10 676–10 683, ISSN: 2153-0866.

[60] K. Geng and G. Yin, "Using deep learning in infrared images to enable human gesture recognition for autonomous vehicles," *IEEE Access*, vol. 8, pp. 88 227–88 240, 2020.

[61] A. Reyes-Muñoz and J. Guerrero-Ibáñez, "Vulnerable road users and connected autonomous vehicles interaction: A survey," *Sensors*, vol. 22, no. 12, p. 4614, 2022, publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/22/12/4614

[62] Z. Liu, L. Pu, Z. Meng, X. Yang, K. Zhu, and L. Zhang, "POFS: A novel pedestrian-oriented forewarning system for vulnerable pedestrian safety," in *2015 International Conference on Connected Vehicles and Expo (ICCVE)*, 2015, pp. 100–105, ISSN: 2378-1297.

[63] G. Gronier, L. Schwartz, and T. Latour, "French adaptation and validation of the pedestrian receptivity questionnaire for fully autonomous vehicles (f-PRQF)," *International Journal of Human-Computer Interaction*, pp. 1–15, 2022, publisher: Taylor & Francis. [Online]. Available: https://doi.org/10.1080/10447318.2022.2129735

[64] L. O'Haver, "UT students arrested after damaging food delivery robot," *WATE 6*, 2022. [Online]. Available: https://www.wate.com/news/local-news/ut-students-arrested-after-damaging-food-delivery-robot/

[65] A. M. H. Abrams, P. S. C. Dautzenberg, C. Jakobowsky, S. Ladwig, and A. M. Rosenthal-von der Pütten, "A theoretical and empirical reflection on technology acceptance models for autonomous delivery robots," in *Proceedings of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '21. Association for Computing Machinery, 2021, pp. 272–280. [Online]. Available: https://dl.acm.org/doi/10.1145/3434073.3444662

[66] S. Kapser and M. Abdelrahman, "Acceptance of autonomous delivery vehicles for last-mile delivery in germany – extending UTAUT2 with risk perceptions," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 210–225, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X19309258

[67] D. Moore, R. Currano, M. Shanks, and D. Sirkin, "Defense against the dark cars: Design principles for griefing of autonomous vehicles," in *Proceedings of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '20. Association for Computing Machinery, 2020, pp. 201–209. [Online]. Available: https://dl.acm.org/doi/10.1145/3319502.3374796

[68] J. Czarnowski, A. Dabrowski, M. Maciaś, J. Glówka, and J. Wrona, "Technology gaps in human-machine interfaces for autonomous construction robots," *Automation in Construction*, vol. 94, pp. 179–190, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580517300080

[69] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Jun. 2016, pp. 779–788.

[70] X. Li, J. Zhao, and L. Xiao, "Real-time lane detection and tracking for autonomous driving using deep learning." *IEEE Access*, vol. 6, pp. 25 104–25 112, 2018.

[71] S. Bai, X. Bai, and J. Zhou, "Deep learning-based traffic sign recognition for intelligent vehicles." *IEEE Transactions on Intelligent Transportation Systems*, vol. 19(6), pp. 1838–1850, 2018.

[72] X. Chen, Y. Zhan, B. Zhang, and X. Chen, "A deep reinforcement learning approach for path planning of autonomous vehicles in complex environments." *IEEE Access*, vol. 8, pp. 209 800–209 810, 2020.

[73] S. Chae, M. Lee, and S. Lee, "Path planning of autonomous vehicles using deep learning." *Journal of Institute of Control, Robotics and Systems*, vol. 25(7), pp. 635–642, 2019.

[74] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, and X. Zhang, "End to end learning for self-driving cars." *arXiv*, vol. preprint, p. arXiv:1604.07316, 2016.

[75] H. Chen, C. Wang, and G. Wang, "Deep learning for autonomous driving: A review." *IEEE Transactions on Intelligent Transportation Systems*, vol. 17(12), pp. 3640–3654, 2015.
[76] C. Zhang, Y. Zhu, J. Chen, and H. Liu, "A pedestrian detection method for autonomous vehicles based on deep learning." *IEEE Access*, vol. 6, pp. 45 054–45 061, 2018.
[77] Y. Zhang, S. Liao, S. Zou, and W. Hu, "A deep convolutional neural network for vehicle detection in unmanned aerial vehicle imagery." *Neurocomputing*, vol. 214, pp. 757–768, 2016.
[78] F. Ma, Y. Zhang, L. Zhang, and W. Sun, "A multi-task deep learning approach for vehicle detection in aerial images." *International Journal of Remote Sensing*, vol. 41(1), pp. 215–235, 2020.
[79] Y. Wang, X. Sun, J. Lu, and M. Yang, "Vehicle detection in aerial images based on convolutional neural network with region proposal network." *Neurocomputing*, vol. 398, pp. 47–59, 2020.
[80] Y. Shi, Y. Wang, and H. Wang, "A pedestrian detection algorithm in the complex background based on deep learning." *Journal of Physics: Conference Series*, vol. 1082(3), p. 032073, 2018.
[81] S. Huang, X. You, W. Chen, and X. Xu, "A deep learning-based pedestrian detection method for autonomous driving." *Symmetry*, vol. 12(9), p. 1518, 2020.
[82] L. Zhang, R. Jiang, and Y. Zhang, "Real-time pedestrian detection in autonomous vehicles with transfer learning." *IEEE Transactions on Intelligent Transportation Systems*, vol. 22(2), pp. 776–787, 2021.
[83] X. Guo, J. Liu, F. Wang, and F. Wu, "Traffic sign detection and recognition using deep learning based on improved yolov2." *Neurocomputing*, vol. 266, pp. 141–150, 2017.