Self-Driving Car

Emil Jiang

June 25, 2022

Self-driving cars, a car able to operate itself and perform necessary functions without any human intervention, through the ability to sense its surroundings, has been the subject of interest for many years. Starting in the 1500s, Leonardo Da Vinci created the first autonomous vehicle with the creation of a cart that ran on springs and high tension. Later on, in 1925 Francis Houdini was able to remote control a car with a radio, but he crashed multiple times. Autonomous vehicles continued to develop and in 1990 Carnegie Mellon University utilized neural networks, image processing, and steering controls and drove a car 2797 miles. Now multiple car brands are selling their autonomous vehicles including Ford, Mercedes-Benz, BMW, and most notable, Tesla. Tesla is the closest to fully autonomous, but it still has ways to go to being completely autonomous. There are 6 levels of automation with level 0 being no automation. The driver controls steering and speed and has no assistance at all. Level 1 is limited driver assistance with systems that can control steering and acceleration/deceleration under specific circumstances, but not both at the same time. Level 2 is driver-assist systems that control both steering and acceleration/deceleration. This still requires that person to be attentive at all times. Level 3 is vehicles that can drive themselves in certain situations, such as in dense traffic on divided highways. Human intervention is not needed when the autonomous mode is engaged, but the driver must be ready to take over when the vehicle encounters a situation that exceeds its limits. Level 4 is vehicles that can drive themselves most of the time, but may need a human driver to take over in certain situations. Level 5 is fully autonomous and vehicles drive themselves at all times, under all circumstances. They have no need for manual controls. The importance of fully autonomous vehicles is that computers are safer and more cost-effective. Human error causes most crashes and with computers controlling the vehicle, there will be significantly fewer crashes and deaths related to transportation. This will in turn lead to less money spent on car crash-related damages which is the significance of this project and why it is important to do research in this area. The researcher in this project wanted to see how a model car could be controlled to become fully autonomous.

The researcher brainstormed ideas first and decided on building a model car and controlling it with some sort of device and a camera for vision. The researcher started off by looking online and seeing if other projects have been done similar to what they wanted to do. There were a few projects already done and he took inspiration and wrote an outline. This outline included what the researcher was building, what the car is expected to do, what the functionalities the final product will have, what parts are needed and how they will be put together, and what are the major risks and what can be done to reduce the risks. A model car kit, raspberry pi, USB camera, SD cards, motors, motor board, mouse, keyboard, and wires were the initial material needed.

The first step of the building portion of the car was to set up the raspberry pi. The SD card first had to be formatted using an SD card formatting application. This clears all the data on the SD card and allows it to run Raspbian operating system. The next step was to download NOOBS onto the SD card. NOOBS contains Raspbian and allows the raspberry pi to act as a computer. A mouse, keyboard, micro-HDMI, and power supply were plugged into the raspberry pi and connected to a monitor which allowed the researcher to control the raspberry pi. The researcher thought connecting to the monitor and disconnecting would be too much of a hassle so they installed a VNC viewer on their laptop and set it up connecting it to the raspberry pi. This allows the researcher to code, control, and operate the raspberry pi from a laptop instead of a monitor which allows for more efficiency.

After setting up the raspberry pi, building the car kit was next. The car kit came with two boards, motors, wheels, and screws. The researcher attached the four motors onto one of the boards and screwed up the other board to make a car body. The wheels were then added on. With the car done, the next step was to get the car moving. The researcher connected the red power wire from the battery pack to the motor board. Since the raspberry pi and battery pack both need to be grounded and the motor board only has one ground position, the researcher intertwined the raspberry pi wire with the battery pack wire. After that, the ENA pin of the motor board was connected to physical pin 22 or GPIO25 of the raspberry pi. Input 1 and input 2 of the motor board were connected to physical pins 16 and 18 or GPIO23 and GPIO24 of the raspberry pi. The final step was to connect the motors to the motor board and all the researcher had left to do was connect the webcam to the raspberry pi and position and secure everything on the top of the car.

After the completion of the building section, the researcher was ready to move on to the coding portion. At the start, the researcher was completely unfamiliar with any vision libraries. After reviewing previously similar projects, OpenCV in python was decided on and thought to be the best fit for this project. The researcher looked on youtube and found a multitude of introduction videos and learned the basics in a few hours. The first step was downloading all of the modules needed for the project: OpenCV, RPi.GPIO, Numpy, math, logging, and matplotlib. The researcher then decided that the color blue would be the color of the lanes the car would be following. The reason was that there are not a lot of things that are blue around and blue is different from the floor. Blue painter's tape is also readily available everywhere. The first step in coding was producing a live feed of what the webcam was viewing. The next step was to change BGR colors to HSV colors. This transformation changes all of a specific range to one color.

Different lighting may change a color a bit and this would affect how the car moves so if it's just one color, it would be a lot easier for the car to navigate. The next was to mask all of the colors that are not blue. This allows the lane lines to become white and all other colors to be masked as black. After masking, the goal was to highlight the edges of the now white lane lines. The researcher used the canny edge detection function to find the edges so there are only four distinct lines. After the canny edge detection, the next goal was to isolate everything but what the car should be focused on, which is the lower half of the image. To do this, a white block is compared to the original bottom half of the image. A “and” function is done and only the lines that are already white will show in the bottom half and everywhere else will be black. After isolating the bottom half of the image, the researcher performed a hough line transform which is an OpenCV function that detects line segments. After getting the data for the four-line segments, the researcher wanted to combine them into two lanes. The basic principle behind this is if the slope of the lines is positive, the lane is on the left and if the slope is negative the lane is on the right. Then the slope and intercept of the two left lanes and the slope and intercept of the two right lanes are averaged out into two lanes. Now that the car can detect the lane lines, the middle line the car follows is needed to steer the car. If there are two lanes shown in the image, the two farthest points from the two lanes are averaged to find the middle line. A line from the middle bottom middle is drawn to the averaged point is drawn to draw the middle line. If only one lane is shown, starting from the bottom middle, the same slope is drawn until the end of the image. Next, the angle of the line needs to be determined. 90 degrees would be the car going straight, less than that would mean the car is left turning, and above 90 would mean the car is right turning. To find the angle the tan math function is needed. To move the car, every few milliseconds, the researcher checked if the car was centered. If it was turning left, the right wheel would stop moving and the left would continue to turn the car to the right. If the car was turning right, the left wheel would stop moving and the right would continue to move the car left. This is how the researcher effectively utilizes vision in order to autonomously steer a car.

There were many challenges along the way during this project. An initial challenge was the battery pack that came with the car kit was faulty. Sometimes the voltage did not travel fully across the board. The only solution to the problem was to take out all of the batteries and put them in again. The cause of this problem is still unknown and to improve this project, a new battery pack would be needed to make sure the car is more reliable and does not die from the lack of power in the middle. Another problem with the battery pack was that the battery pack came with an extension. The motor board only takes wire input, but the battery pack came with a circular input. In order to fix this problem, the researcher had to cut off the extension portion and strip the wiring. This led to some problems as the battery pack wiring length was too little which made the motor board and battery pack too close together. Another problem faced during this research was that the raspberry pi and motor board were overheating. Heat would start burning the wiring and the whole system would not work. An early solution was to unplug everything and wait for the system to cool down. This was ineffective and wasted a lot of time so the researcher bought a heat sink for the raspberry pi and motor board in order to control the overheating. After solving that problem, the researcher noticed that the library that was in use had limited capabilities. The motors only had two options which were high and low. This makes turning a lot more difficult. The only fix the researcher could come up with was to check the turning angle every few milliseconds, but this would make the car continuously turn which is not effective. In the future, more research into how to control the motor speed is needed in order to make the car turn and move a lot smoother. The lack of functionality of the python library was a problem in the robot car turning, but the researcher noticed that the robot had a rough time making simple small adjustments in the lanes. Only two motors were in use and the researcher noticed that the front two motors still had the motors in them which made the turning much more difficult. The solution the researcher came up with was to remove the motors from the motor shell and connect the wheels to an empty motor shell. The researcher had to take apart the whole car and rebuild it, but this helped the car do less work to make simple small adjustments. The car also seemed to be going slow. The researcher initially had the battery pack, the heaviest part, in the front and decided to move it to the back. This allowed the front wheels to move more smoothly as there was less weight straining them. Power for the raspberry pi was also a problem as it had to be connected to an outlet. Since the track the researcher used to test was relatively small, an extension cord was only needed to allow the car to move. In the future, another battery pack for the raspberry pi would be needed in order to make the car move around courses without the strain of not having enough wire to power the raspberry pi. Even though the researcher was utilizing a VNC viewer and did not use the monitor, the monitor would still turn off. This caused the VNC viewer to shut off and wasted a lot of time as the monitor and raspberry pi had to all be turned back on and set up. The solution was to remove the third-party monitor and fully be able to turn on/off the raspberry pi from a laptop. This was done by following online directions and typing commands into the terminal. The final problem faced was there was an unknown bug in the program which made the steering angle jump up or down drastically which confused the car and basically stopped the program. The issue is still unknown but the solution was that if the angle jumped up or down more than 10 then only 5 is added or subtracted from the previous angle.

This project helped the researcher develop a broader understanding of how vision works and the struggles of combining hardware and software together into a final product. In the future, the researcher could potentially add better turning mechanics if power levels for the motor are found out which was previously stated above. Machine learning could potentially be another area of study. Having the car stop when a person or stop sign is detected could be further improvement to the car. The area of autonomous vehicles is still in progress and getting better every day and in the future, fully autonomous vehicles are the end goal.