**Vehicle documentation**

Our vehicle is driven by two motors, including a servo motor and a brushless DC motor, referencing the Ackerman steering mechanism. The code of the AI ​​car can be divided into two parts, the motor part and the Intel RealSense depth camera D435i part.

Here is a summary of our vehicle:

• Onboard four 18650 batteries, 10400mAh large capacity, two parallel and two series output current are larger, and the motor power is stronger

• Onboard HY2120+AOD514 lithium battery protection circuit, with anti-overcharge, anti-over-discharge, anti-overcurrent and short circuit protection functions

• Onboard FP5139 automatic buck-boost regulator circuit, which can provide a stable 5V voltage for Jetson Nano

• Onboard 0.91-inch 128×32 resolution OLED, real-time display of vehicle IP address, memory, power, etc.

• Onboard AINA219 acquisition chip, convenient for real-time monitoring of battery voltage and charging current.

Additionally, we imported the Jetson Nano development kit, a computer that allows us to run multiple neural networks in parallel for applications such as image classification or object detection. We believe it is well suited for AI technology and robotics because it is small enough to comply with vehicle regulations.

To mount our depth camera, we added an L-shaped 3D printed bracket to replace the original camera to mount the depth camera at the best angle. The stand is printed from resin, which is strong and stable enough to support the camera. Due to the light weight of the resin, it will not affect the performance of the vehicle during driving.

The motor part adopts brushless DC motor and servo motor. The program code is written in Python. The main idea of ​​vehicle motion is based on Ackermann steering. When our vehicle's camera captures the view of the "traffic lights" (red and green blocks), servo motors turn the vehicle left or right by automatically adjusting the vehicle's front axle. Its acceleration is determined by a programmed brushless DC motor at the rear of the vehicle. The code "car. steering" is used to control the servo motor and "car. throttle" is used to control the DC motor.

For example, if we enter "car. ​​throttle 0.15", the car will move forward at a speed of "0.15", the larger the number, the faster it drives. The range of "car. throttle" is from 0 to 1. For the servo motor, if we enter the code "car. steering 0.2", it will turn right a bit. The range value of steering is from –1 to 1, when the value is negative, the vehicle turns left, and vice versa.

We thought it would be better if our vehicle could be moved closer to the center of the map to reduce the time, as it could reduce the distance traveled. So, we wrote a code to drive our vehicle to travel the Inner circle while we can still finish the tasks at the same time.

Recognizing "traffic lights". Using the Intel RealSense Depth Camera D435i. This is a depth camera that can identify the color of traffic lights the vehicle is facing. In our program, we set the green to range from ([40, 80, 40]) (BGR) to ([102, 255, 255]) (BGR), and the red to range from ([0, 50, 120]) (BGR) to ([10, 255, 255]) (BGR). We train the AI ​​database by feeding thousands of photos of red and green patches from different angles and orientations into the camera database. Therefore, it can have a correct identification. When the camera captures a pre-set range of size of square, it will recognize the color of the square and perform corresponding actions. For example, it turns left when capturing a green block, and vice versa.

In another case, in the sensors section, it shows up in the turn maneuver. During the turning process, if the camera sees the traffic light, it will stop the turning function for 2 seconds, but it will perform the function of recognizing the color of the traffic light, and the steering current sensor will sense the traffic light. When the distance between them is less than or equal to 10 cm, the vehicle will turn in the correct direction. For example, when the vehicle is about to turn, the right sensor senses more than 2000cm, rotates 90 degrees clockwise, and the direction of the servo motor is 0 (front motor). During the turning movement, the camera sees the red light and stops for 2 seconds, but the servo motor rotates 90 degrees counterclockwise, and the direction is 180 degrees. After passing the red light, the vehicle will check the value again and restore the balance. In addition, the camera also has a depth camera function, which can check the distance between the vehicle and the wall. When the camera senses that the object in front of it is black (say, a wall), it immediately turns in the right direction and keeps moving.

When we were taking pictures, we decided to use a handle of PS4 to drive the car to take pictures and train AI. We think it's more convenient because we can still control the vehicle ourselves. We think it is extremely important for our vehicle's database to do high-precision motion, so it does take a lot of time for us to take pictures. When doing this, we need to maintain a high degree of concentration and take useful photos. We focus on quality over quantity. After we have trained the AI, we must combine our database with our program code to make the correct action when faced with "traffic lights".

During this process, we encountered some problems. For example, when an error occurs in the program. To find out what the problem is, we need to study the datasheet again to find the error. If we can't find it, we will search online, find reference materials, and track and debug. It is important to revise the datasheet frequently because we can find wrong solutions. "Real engineering is about learning or creating solutions and sharing them with the community to advance the whole idea". We appreciate engineers who share their ideas for our consideration.

To sum up, our vehicle adopts Intel RealSense depth camera D435i, Jetson Nano development kit, motor matching with all equipment, and self-judgment between traffic lights to ensure smooth and accurate driving of the vehicle. It is an autonomous artificial intelligence vehicle that can complete tasks on its own.