Smart Mobile Robot (July 2022)

Walter Humphrey, Gerardo Peña, Javier Piña, José Manuel Neri

Students from Instituto Tecnológico y de Estudios Superiores de Monterrey

A01701730

A01701474

A01701478

A01706450

This project was made for the Intelligent Robotics Implementation class (Gpo 502) with the help of Manchester Robotics, who provided material to carry out this project..

SUMMARY The following report presents the results obtained in the development of a differential mobile robot, in which a line follower was developed to stay on the track and achieve the route, while continuously checking the traffic lights on the road. In addition, a neural network that could recognize between different traffic signs and make decisions based on recognition was generated and trained. The methods used for each of the parts that make up the project will be explained, as well as comparisons with other possible solutions for this challenge. The findings obtained during the project are also presented.

OBJECTIVES Implement a line follower node that is capable of following paths including curves. Implement a traffic light detector node that is capable of recognizing if it is green, red and yellow. Implement a convolutional neural network that is capable of predicting traffic signals effectively. Finally, implement a master node that is capable of unifying the previous nodes, resulting in intelligent and autonomous behavior for the puzzlebot, a small mobile robot.

KEYWORDS ROS, python, opencv, tensorflow, convolutional neural networks, image processing, autonomous vehicle.

I. INTRODUCTION

In this project, we worked with a differential mobile robot, which had the objective of crossing a track following a line to complete the route while recognizing each traffic signal on its way with the help of a neural network, as well as recognizing the traffic lights and identifying the amount of light in the hsv spectrum.

II. MATERIALS AND METHODS

To carry out this project we had the help of Manchester Robotics, who provided us with the necessary materials and advice for the development of the project. The robot used in the project is a Jetson Edition puzzlebot from the Manchester Robotics company. The puzzlebot has two motors with integrated encoders, a 2GB Jetson Nano, a hackerboard to control the motors and a camera to visualize the track.

The track where the puzzlebot makes its way can be seen in image 1.1. The robot must navigate autonomously within it, starting from the area where the Stop sign is located and returning to it, after making the journey obeying the signs and traffic lights, without any human intervention.

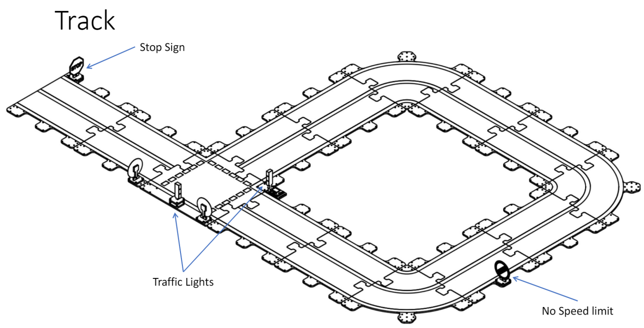

image 1.1. Drawing of the track on which the robot

The track has four traffic signals: Stop, Go Ahead, Turn Right and No Limit. At the intersection there are two traffic lights. The floor of the track is marked with three black lines that run along the ground as a guide for the robot, these are interrupted only at the intersection.

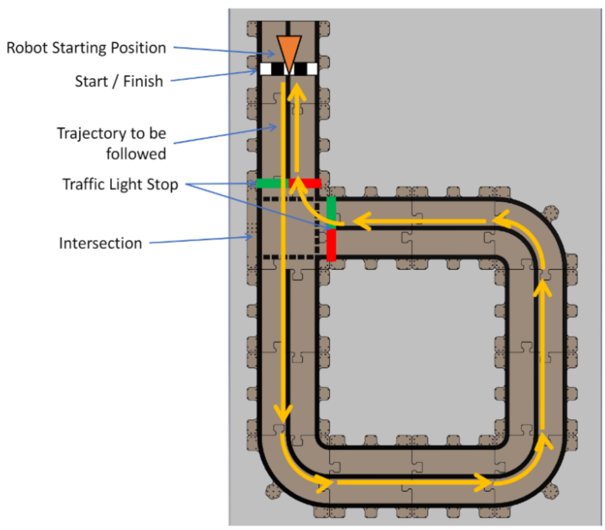

image 1.2. puzzlebot path

The presented solution consists of four ROS nodes in which it is located: a neural network, a semaphore detection code, a line follower and a master node that obtains the information from the other nodes to control the response of the puzzlebot. The most relevant libraries we used for signal and semaphore detection were OpenCV and TensorFlow.

Neural Network (CNN)

We generate and train a convolutional neural network for the detection of traffic signs. We use a RELU function as activation function, we use softmax as activation function in the final layer to be able to normalize the values to a percentage, said percentage tells us the probability of belonging of the image to each one of the signals, once imported the model, we use probabilities to decide which signal the robot is looking at and based on that it performs certain routines.

The network was trained on a database containing hundreds of images of the four signals present on the runway. The photos were taken from real traffic signs and photos of the signs on the road were added to improve network training.

Traffic Light Detection

For the identification of traffic lights we had to process the image to limit the space where a traffic light could appear and determine the colors that this space presents from the image captured by the camera. Filters such as fastNlMeansDenoisingColored were applied, which allows us to reduce the noise in the image to later be able to convert to HSV format and thus be able to differentiate the colors based on the part of the color spectrum in which the image is found. Three color masks were created, one for the detection of green and two for detecting both spaces of red within the color spectrum. We perform a conversion of the image to binary format to highlight the corresponding color and then determine what color it is by comparing it with a certain number of lit pixels in the binary image. In this way we determine if the puzzlebot is facing a traffic light and the color of the light.

Line follower

As for the line follower, we had to limit the area of the image to prevent adjacent objects from causing noise, in addition to only following the line located in the center. We convert the image to grayscale so that we can transform the image to a binary format and highlight the dark line. We applied a Gaussian Blur with a kernel of (5.5) to smooth the image and then did 4 iterations of erode and dilation filters to maximize blacks and whites respectively. We made the sum of each of

the pixel columns and thus determine in which part of the image the darkest line is located (the column that had a lower sum value).
A proportional control represented by *equation 1.1* was used,

$$vel_w = k_w(CenterOfCamera - xDarkPixel)$$

Equation 1.1. Proportional robot control

which helps us keep the center of the image close to the darkest pixel detected by the vision system.

Master Node
To communicate (image 1.1), each of the nodes generated a python script that worked as a master node, which is responsible for receiving the data published by each node and then decides what speeds to publish to the cmd_vel topic, which makes the robot move.
The master node checks the color of the traffic lights to know if the robot can move forward, if it is in front of a signal the master executes certain routines to perform the movement required by every road sign.

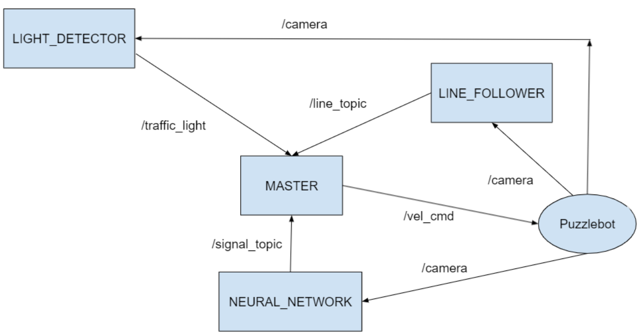

image 1.3. Communication scheme between nodes

III. RESULTS

In the tests carried out with the latest version of the robot, the complete circuit was completed in 1:15 minutes, the traffic signals were fully identified, in addition to the traffic light detector working at all times.
Regarding the detection of traffic lights, it was possible to isolate the color of the light to be able to differentiate the state of the traffic light, as we can see in image 2.1, the masks used work very well when highlighting the desired color.

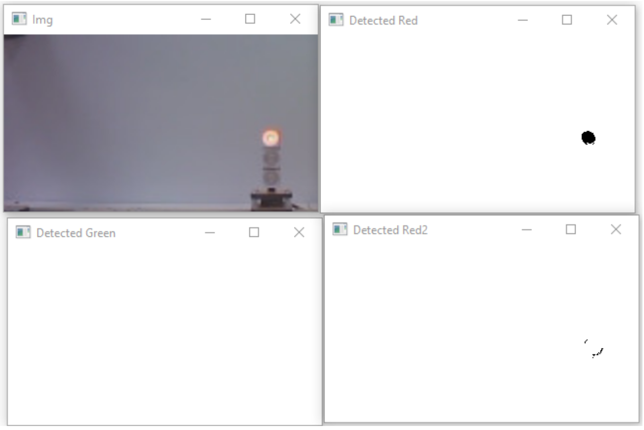

Image 2.1. Traffic light identification

As mentioned before, for the line follower the image was cropped to prevent the robot from trying to follow dark objects off the track, and as shown in image 2.2, it was possible to focus the vision just in the area where the line to follow is found, in this way the robot always stays on the path.

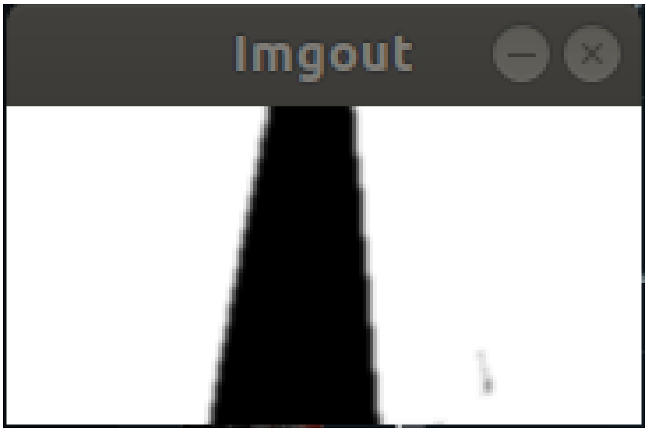

Image 2.2. Image of the line seen by the robot

The generated neural network was able to successfully identify each of the signals arranged on the test track.

Here is a video demonstration of the robot operation:
https://youtu.be/58Kq-_zgGJ4

IV. DISCUSSION OF RESULTS

Gerardo: We obtained good results in the final tests, since it was shown that the robot can obtain information from its own environment, correctly detecting traffic signs, as well as adequately identifying the colors of traffic lights. Some flaws that we found in the process were with respect to the routines that must be carried out when facing certain signals, such as the case of the "turn right", which had to advance for a certain time and then turn, when doing this process the robot sometimes collides with the traffic light, this is thanks to the fact that the signal is not always detected at the same moment, therefore sometimes the robot starts the routine ahead of time causing it to crash. This problem could be fixed by improving the neural network model, allowing the signal to be identified with even less error and limiting a signal to only be heeded if it is close enough. Another area of opportunity is that we could make the robot move faster, for this we need to calculate new values for the gains in the control system, and possibly improve the whole control, maybe go for a PID controller that is able to maintain over the line to the robot even at higher speeds.

Despite these areas of improvement in our system, I consider that it was possible to provide the robot with a certain degree of autonomy, since with the help of all the nodes mentioned above it is possible to complete the route.

Among the main challenges that we find in the development of this project, it stands out having to couple the algorithms to the robot's environment, such as the natural light that exists in the environment, specifically in the detection of traffic lights, since the variations in light change the color ranges that must be considered to differentiate the status of the traffic light.

Manuel: The demonstration of what our project was came out as we expected, it worked correctly, this is because the robot managed to follow the line correctly, identify the traffic signals and act respectively with the traffic lights, but the union of what it was all this it was not easy at all in what was the case of the neural network which simply needed the image of the signal, if the circle was missing a piece or even the image exceeds what the circumference is, it generated noise and could not identify the signal 100% due to the database we use, which requires using an image similar to the training ones, another detail that we noticed is the latency that exists between the Jetson and the computer for sending an image great, but this problem was solved by lowering the resolution of the image, while the color detection also caused a big problem due to the lighting of what were the traffic lights, like it changes the light of the environment or the noise that came to generate the outside of the track, all this was solved by modifying the values and testing what our new nodes were.

Walter: The operation of the robot was adequate according to the proposed objectives. The traffic light detector was able to distinguish if it was facing a green, red, yellow traffic light or there was no traffic light, thus giving the order to the master node to continue with the movement. On the part of the line follower, the robot remained within the track at all times, thanks to the fact that the darkest point was correctly obtained and it was ensured that it was the one on the line to follow. Finally, the convolutional neural network was able to correctly predict the 4 signals along their path, thus ensuring that the master node will correctly indicate the movement to follow. Thanks to these results, the desired degree of intelligence and autonomy was obtained. The main challenges faced during this project were making the necessary adjustments in terms of computational vision with the use of opencv in order to ensure a correct processing of our environment. We realized that even light changes or the presence of people near the runway were enough disturbances to prevent the proper functioning of our system. Another great challenge we faced was the transfer speed between our nodes, due to the computational power of the jetson and the computer where we ran the neural network. Many times, although the code was correct, this speed of information transfer prevented the robot from acting in time and generated unwanted behavior. correctly. To solve this we chose to reduce the quality and size of the images that we were processing in order to lighten the load and transfer of data.

Javier: The implemented solution allowed the fulfillment of the challenge proposed by Manchester Robotics. The puzzlebot was able to make the route on the track while detecting the signs and traffic lights on its way. The neural network proved to be well enough trained to detect all the signals on each run. This also shows that the image processing that was done to cut out the area of the image where circles were detected, the areas where the signals appeared, was correct. In the final implementation the speed of the robot was somewhat slow but at no time did it deviate from the reference line, so we can argue that the control of the robot was adequate despite being a simple proportional gain. Regarding traffic light detection, some errors do occur due to the constant change of lighting inside the room. To solve this problem, before each test, the range of color pixels detected within the traffic light detection must be configured. Once the parameters are configured appropriately for the current lighting, the robot does not show errors, until the lighting changes again. It is appropriate to mention that for the codes to run well, the neural network node must be executed within a computer that supports ROS smoothly. Otherwise the programs start to crash due to the delay in the robot camera.

V. CONCLUSIONS

Manuel: Autonomous robots are becoming more and more a reality where little by little they are entering our lives, this is because they can facilitate a task as complex as driving, they will not help reduce accidents. by simple things such as blind spots, distractions, tiredness, among various factors. This is due to the algorithms that these bring, to the sensors which are responsible for collecting information in real time which becomes highly valuable to preserve the life of the driver.

Gerardo: In general, simple algorithms were developed that managed to meet each of the objectives set for the robot. Much importance was given to image processing, such as the use of filters in real time, use of masks to isolate colors, among other methods currently widely used in computer vision, this to facilitate the work of signal detection, traffic light detection and line tracking.
A possible area of improvement is to make the robot reach higher speed, for this the values of the gains should be changed and possibly the control system should be changed considering a more robust control that allows the robot to be kept on track.

Javier: The functionality of the robot was successful in being able to complete the route of the track identifying the signs and traffic lights. Robot autonomy was achieved using neural networks, computer vision, control, and ROS. Despite being an implementation of an autonomous vehicle, being small-scale, it is limited to the ideal conditions of the test space. To achieve a substantial improvement, the light detection would have to be tuned to ensure that the color detection comes from a traffic light. Control can also be enhanced by PID or even fuzzy control.

Walter: The robot managed the course by always recognizing traffic lights and signs along it, as well as following the established path. This degree of autonomy was achieved thanks to the fact that all the nodes fulfilled their function in a timely manner, thus ensuring that the robot will complete the circuit with all the variables that it involved. In the future I consider that there is much room for improvement in our system, the traffic lights can be detected with greater precision, the line follower can detect the lines with more image processing to ensure the trajectory and the controller of our robot could be more robust to increase control over movement. Giving a more specific explanation of the improvements that could be made would be to include the detection of circles together with the detection of colors in order to avoid noise and false information when observing the lights of a traffic light and the general environment where the robot is.

REFERENCES

API documentation : Tensorflow Core v2.9.1. TensorFlow. (n.d.). Retrieved June 18, 2022, from https://www.tensorflow.org/api_docs

Autopilot. Autopilot | Tesla México. (2022). Retrieved June 18, 2022, from https://www.tesla.com/es_MX/autopilot

Nvidia. (2022, April 13). Nvidia Drive Solutions for autonomous vehicles. NVIDIA Developer. Retrieved June 18, 2022, from https://developer.nvidia.com/drive

OpenCV modules. OpenCV. (n.d.). Retrieved June 18, 2022, from https://docs.opencv.org/4.x/

Wiki. ros.org. (n.d.). Retrieved June 18, 2022, from http://wiki.ros.org/ROS/Tutorials

ANNEXES

1. Diagrama de comunicación en ROS: https://docs.google.com/drawings/d/1BFyG15PDkxO3gXqaw5EMEh1MIcEPTi3nCbyzRMxQfOA/edit?usp=sharing
2. Códigos del proyecto: https://github.com/Javier-P-C/Autonomous-Navigation-Robot